Budker Institute of Nuclear Physics
SB RAS

A.D. Bukin

NEW MINIMIZATION STRATEGY FOR NON-SMOOTH FUNCTIONS

Budker INP 97-79

Novosibirsk

New Minimization Strategy

for Non-smooth Functions

A.D.Bukin

Budker Institute of Nuclear Physics SB RAS
630090 Novosibirsk, Russia

Abstract

New minimization strategy is suggested for non-smooth functions. Main element of this strategy is a combination of Simplex method and modified Newton's one. Probability of reaching the "true" minimum point is highly increased by successive minimization runs from different starting points.

This algorithm is implemented in Fortran-77 in subroutine COMBI. Comparison of minimization results obtained with COMBI and MINUIT programs is performed on the set of functions with complicated profile.

1. Introduction

Function minimization is a very typical problem in data processing in high energy physics experiments. For interactive work with data the code MINUIT [1] is generally used, providing very good convergence to the minimum point of function and nice service (statistical errors evaluation, contour plot, fixing and releasing some of the function parameters etc.) User manual of MINUIT has a special warning that this program "...is not intended for the repeated solution of identically parametrized problems (such as track fitting in a detector) where a specialized program will in general be much more efficient ...".

Trying to apply some new approach for kinematic reconstruction of events [2] to the experimental data of SND-detector [3], the author encountered this very problem: what general purpose minimization code can be used? Despite the mentioned warning the program MINUIT was used first. The trouble was that sometimes a message like "...arithmetic fault, floating divide by zero ..." was issued from one of MINUIT subroutines. It was very difficult to override this trouble, and if SIMPLEX mode was used instead of MIGRAD then troubles disappered but minimum was found with less probability and hence selection efficiency was significantly decreased. These circunstances gave rise to the search of some minimization strategy, which would be reliable enough (as compared to MINUIT) even is slower than MINUIT.

2. Description of the new algorithm

The main idea of this new algorithm is to combine two well-known algorithms: SIMPLEX [4] and Newton's method (which is rarely used for function minimization because in general case it does not provide convergence to the minimum point; one of the possible implementation is discussed in [5]). This combination proved to be rather efficient, but nevertheless rather often minimization is stopped far from the "true" minimum point. In order to increase the probability to find "true" minimum point several minimizations by SIMPLEX method are performed from different start points. Achieved minimum points of differents minimization runs are stored in an array, and the information about these minimum points influences on the choice of the next start point and is used to make a decision when to stop the minimization. At any moment not more than four results of previous minimization runs are stored, and extra minimum points are rejected by a special rule.

Here is a brief description of the algorithm.

2.1. Start of new SIMPLEX minimization

The choice of the starting point of the new SIMPLEX minimization depends on the number of SIMPLEX run:

1. The starting point of the first SIMPLEX run is supplied by a user in the input parameters.

2. The starting point $\mathbf{R}_{start}$ of the second run of SIMPLEX minimization is derived by the following rule:

$$\mathbf{R}_{start} = \mathbf{R}_2 + R_{jump} \times \frac{\mathbf{R}_2 - \mathbf{R}_1}{|\mathbf{R}_2 - \mathbf{R}_1|} \quad (1)$$

   where $\mathbf{R}_1$ is an initial point supplied by the user and $\mathbf{R}_2$ is a minimum point found by the first SIMPLEX run, R_{jump} is a variable which for the second run is set to

$$R_{jump} = 0.01 \times |\mathbf{R}_2 - \mathbf{R}_1| + 0.1,$$

   and for all successive runs is varied according to results of the current minimization. If the new minimization result is better than any previous one, then R_{jump} is increased by a factor of 3, otherwise it is divided by 2. The latter assures the convergence of the algorithm.

3. For all other SIMPLEX runs the starting point is chosen along the "parabolic" curve

$$\mathbf{R}_{start} = \mathbf{R}_2 \cdot (1+t) - \mathbf{R}_1 \cdot t + t(t+1)\mathbf{e} \quad (2)$$

   where $\mathbf{R}_2$ is the best minimum point of all the previous SIMPLEX runs, and $\mathbf{R}_1$ is the *most distant minimum point* to $\mathbf{R}_2$ of all points, stored at this moment in the array of minimum points ("history" array). The vector $\mathbf{e}$ is orthogonal to the vector $(\mathbf{R}_1 - \mathbf{R}_2)$ and provides minimum sum of squared deviations of this curve from the minimum points $\mathbf{R}_i$, stored in the "history" array:

$$\mathbf{e} = \frac{\sum [\mathbf{R}_i - \mathbf{R}_2 - t_i \cdot (\mathbf{R}_1 - \mathbf{R}_2)]\, t_i \cdot (t_i + 1)}{t_i^2 (t_i+1)^2} \quad (3)$$

   where

$$t_i = \frac{(\mathbf{R}_i - \mathbf{R}_2)(\mathbf{R}_1 - \mathbf{R}_2)}{(\mathbf{R}_1 - \mathbf{R}_2)^2}$$

   If the absolute valuee of this vector exceeds the current value of R_{jump} or $|\mathbf{R}_1 - \mathbf{R}_2|$ then the length of this vector is shortened to the least of these two values. If there are only two minimization results and thus there is an uncertainty in the definition of the vector $\mathbf{e}$ then it is set to zero. Parameter t in formula (2) is chosen so that the distance $|\mathbf{R}_{start} - \mathbf{R}_2|$ is equal to R_{jump}.

The initialization SIMPLEX step H_s is set to $R_{jump} \times 0.1$ (for the first SIMPLEX run R_{jump} is set to unit).

2.2. Initialization of SIMPLEX

First, the initialization step in each coordinate x_i is set to $h_i = H_s$, then the preliminary descent along this coordinate is performed by function evaluation at the points $x_i \pm h_i$. If the function value in some direction is less than that at the initial point, then h_i is multiplied by 1.5 and the function check is proceeded from this new point. Since the function value at the points $x_i \pm h_i$ is greater than that at the point x_i the preliminary descent along this coordinate is stopped and h_i is divided by 2.

Thus, we obtain an improved starting point $\mathbf{R}_{start}$ with n coordinates $x_i, i = 1, \ldots, n$. Then we construct a set of $(n+1)$ points which we shall call further as *simplex*. One point is the starting point $\mathbf{R}_{start}$, all other points are obtained increasing all coordinates by h_i in turn. Among these $(n+1)$ points there is one with the lowest value of function. This point will be called the "best point". Similarly there is the "worst point" with the greatest value of the function. SIMPLEX minimization is performed by successive change of the worst point to a new better one.

2.3. Choice of a new point for the simplex

Let us denote the points of the simplex as $\mathbf{R}_j$, the best point as $\mathbf{R}_{best}$ and the worst one as $\mathbf{R}_{worst}$. First, the "center of gravity" $\mathbf{R}_c$ is calculated:

$$\mathbf{R}_c = \frac{1}{n} \sum_{\mathbf{R}_j \neq \mathbf{R}_{worst}} \mathbf{R}_j \qquad (4)$$

Now four new points in turn are tested as a replacement for the worst point. If any of these points has a function value less than that of the worst point, then the worst point is replaced by this new point, and the new best and worst points are determined.

New points are searched along a straight line going through the center of gravity and the worst point, defined by a parameter ρ:

$$\mathbf{R} = (1+\rho) \times \mathbf{R}_c - \rho \times \mathbf{R}_{worst} \qquad (5)$$

The first point is tested at $\rho = 2$, the second one at $\rho = 1$. If none of these points is better than the "worst" point, then the function is evaluated at $\rho = -0.5$. The parabolic interpolation by the least squares' method is done for the four points: three new points with different ρ and the "worst" point. At the minimum point of this parabola the function is also evaluated and the final decision is made. If one of these points is better than the "worst" point in a simplex, then the latter one is replaced, and the new "worst" point in simplex is determined. If the replacement is impossible, then initialization parameter H_s is multiplied by 0.2 and simplex initialization is repeated around the best point of the simplex (jump to 2.2).

In parallel with simplex minimization all points and the function values are used to estimate the minimum point by the modified Newton's method.

2.4. Modified Newton's method

The base of the Newton's method is reconstruction of the quadratic form coefficients in the n-dimensional space using function values in $N_q = (n+1)(n+2)/2$ points. Since the coefficients of the quadratic form are obtained, one can derive the minimum point $\mathbf{R}_q$ and predict the minimum function value f_q.

Since we often deal with functions strongly differing from the quadratic form and the chosen points are not optimal for the evaluation of quadratic form coefficients, it is more reasonable to use least squares method for the approximation of the function surface with the quadratic form with much more points than N_q.

In this algorithm all the successive points and function values during simplex minimization are used to store the appropriate sums for the least squares method. If the number of used points exceeds $(3N_q+5)$ then the coefficients of the quadratic form are evaluated, which are used to predic the minimum point $\mathbf{R}_q$ and minimum function value f_q. The stored sums are reset to zero.

If the actual function value at the predicted point $f(\mathbf{R}_q)$ is less than the best function value of the simplex, then we go to 2.2 to start a new simplex around $\mathbf{R}_q$.

2.5. End of simplex minimization

To stop the simplex minimization a user defined parameter ϵ is used (desired accuracy). Simplex minimization is stopped if the difference between the function values at the best and worst points of the simplex is less than $5 \cdot 10^{-3} \times \epsilon$.

The results of simplex minimization are stored in a special array ("history" array). The number of remembered results is limited to 4, so when the number of minimization results exceeds 4, the new result substitutes one of the previous ones. This substitution is always done if a new minimum value is less than all previous minimum values. Otherwise substitution is done if a distance from the new minimum point to the global minimum point is less than that of any minimum point. This most distant minimum point is deleted from the array.

In a case when the new minimum point is the most distant point from the global minimum, the substitution is not done, but the parameter R_{jump} is multiplied by 0.5 so that the next minimum point is more likely to be close to the global minimum.

This leap size R_{jump} is multiplied by 3 if the new minimum point is better than all previous points and the distance from the new minimum to the previous one is greater than $0.5R_{jump}$. This can help to move along the "valley" with an increasing step.

Simplex minimization runs are repeated according to 2.1 while R_{jump} is greater than 10^{-13} and the difference between the "worst" minimum value in the array of minimum points and the achieved global minimum value is greater than 0.01ϵ.

2.6. End of global minimization

As it was mentioned, the minimization is usually stopped when the difference between the minimum values in the "history array" becomes less than 0.01ϵ. But sometimes an estimation of the minimum point $\mathbf{R}_q$ by Newton's method is considered to be good enough to be the global minimum. It occurs when the difference between the estimated minimum function value f_q and actual function value $f(\mathbf{R}_q)$ is less than $0.01\cdot\epsilon$.

Minimization is also stopped if the number of function evaluations exceeds the user defined limit. In all cases the best point is output as a minimum point.

The subroutine COMBI performing minimization according to this algorithm was written in Fortran–77.

3. Test of the algorithm

For program debugging and test of the algorithm the following functions of two parameters were used:

$$f_1(x,y) = \left[(x-y)^2-4\right]^2 + 100\cdot\left[6(x^2+y^2)+8xy-4\right]^2$$
$$\min f_1(x,y) = 0 = \begin{cases} f_1(1,-1) \\ f_1(-1,1) \end{cases}$$

$$f_2(x,y) = 100\cdot\left[y-0.01x^2+1\right]^2 + 0.01\cdot(x+10)^2$$
$$\min f_2(x,y) = f_2(-10,0) = 0$$

$$f_3(x,y) = 100\cdot\left[y-\cos x\right]^2 + (y-x-1.5\pi)^2$$
$$\min f_3(x,y) = f_3(-1.5\pi,0) = 0 \approx f_3(-4.712388980385,0)$$

$$f_4(x,y) = 100y^2 + 0.01\cdot|x+10|$$
$$\min f_4(x,y) = f_4(-10,0) = 0$$

$$f_5(x,y) = 100\cdot|x+10| + 0.01y^2$$
$$\min f_5(x,y) = f_5(-10,0) = 0$$

$$f_6(x,y) = 100\sqrt{|y-0.01x^2|} + 0.01|x+10|$$
$$\min f_6(x,y) = f_6(-10,1) = 0$$

$$f_7(x,y) = 100\cdot\sqrt{|25+xy|} + 100\cdot\sqrt{|x+\exp(y)-\exp(5)+5|}$$
$$\min f_7(x,y) = 0 = \begin{cases} f_7(-5,5) \\ f_7(142.574,-0.17535) \end{cases}$$

$$f_8(x,y) = 1000\cdot|y^2+x^2-800| + |y+x+40|$$
$$\min f_8(x,y) = f_8(-20,-20) = 0$$

$$f_9(x,y) = 1000\left(x-5y-y^2\right)^2 + |y+x+9|$$
$$\min f_9(x,y) = f_9(-6,-3) = 0$$

$$f_{10}(x,y) = 1000\left(x^2+20|x|+y^2-270\right)^2 + |3x+y+30|$$
$$\min f_{10}(x,y) = 0 = \begin{cases} f_{10}(-7,-9) \\ f_{10}(-9,-3) = 0 \end{cases}$$

$$f_{11}(x,y) = 1000\sin^2(x-y) + (x+5)^2 + (y+5)^2$$
$$\min f_{11}(x,y) = f_{11}(-5,-5) = 0$$

$$f_{12}(x,y) = 1000|x+5-\rho\cos\rho| + 1000|y+5+\rho\sin\rho| + \rho$$
$$\text{where } \rho = \sqrt{(x+5)^2+(y+5)^2}$$
$$\min f_{12}(x,y) = f_{12}(-5,-5) = 0$$

$$f_{13}(x,y) = \rho + 100\sin^2(10\rho-\varphi),$$
$$\text{where } \rho = \sqrt{(x+3)^2+(y-0.5)^2},\ x = -3+\rho\cos\varphi,\ y = 0.5+\rho\sin\varphi$$
$$\min f_{13}(x,y) = f_{13}(-3,0.5) = 0$$

$$f_{14}(x,y) = 1000|y-0.001x^3| + |y+x+11|$$
$$\min f_{14}(x,y) = f_{14}(-10,-1) = 0$$

$$f_{15}(x,y) = 1000|y+x^2+10x-25| + 0.1\cdot|y+10x+75|$$
$$\min f_{15}(x,y) = 0 = \begin{cases} f_{15}(-10,25) \\ f_{15}(10,-175) \end{cases}$$

$$f_{16}(x,y) = 1000\cdot|(y+x-10)(3y-x+10)(3x-y+10)| + |y+x+10|$$
$$\min f_{16}(x,y) = f_{16}(-5,-5) = 0$$

$$f_{17}(x,y) = 1000\cdot|(y+2x-10)(3y-x+10)(3x-y+10)| + |y+x+10|$$
$$\min f_{17}(x,y) = 0 = \begin{cases} f_{17}(-5,-5) \\ f_{17}(20,-30) \end{cases}$$

$$f_{18}(x,y) = 1000\cdot|(y+15x+80)(y-21x-100)(100x+y-100)| + |y+17x+90|$$
$$\min f_{18}(x,y) = 0 = \begin{cases} f_{18}(-5,-5) \\ f_{18}(2.28916,-128.91566) \end{cases}$$

$$f_{19}(x,y) = 1000 \cdot |y - x^2 + 10| + 0.1 \cdot |y - x - 62|$$

$$\min f_{19}(x,y) = 0 = \begin{cases} f_{19}(9,71) \\ f_{19}(-8,54) = 0 \end{cases}$$

$$f_{20}(x,y) = 1000 \cdot (y - 5x - 9)^2 + 0.1 \cdot (4y + x + 6)^2$$

$$\min f_{20}(x,y) = f_{20}(-2,-1) = 0$$

As one can see, all functions have a minimum value equal to zero. The initial point was always $x = 1, y = 1$, initial steps were equal to 0.1. While using COMBI the required accuracy of function minimization was equal to 0.01, a search for minimum by MINUIT program was performed at default conditions (MINUIT release 95.03 was used at VAX/3600 station).

Table 1 demonstrates the achieved minimum points from the main MINUIT algorithms and COMBI with a limit on the number of function calls equal to 10^5. The remaining distance to the real minimum point $r^{(0)}$ from the estimated one r equals $\Delta r = \sqrt{\sum_i \left(r_i - r_i^{(0)}\right)^2}$. The "Abend" instead of the results of minimization means that no results were obtained because of some error message, something like "... arithmetic fault, floating divide by zero ...".

First of all, these results show the well-known fact that the Minuit modes MIGRAD and MINIMIZE work much better than the modes SEEK and SIMPLEX, therefore let us compare only MIGRAD or MINIMIZE with COMBI. For all 20 functions in Table 1 COMBI successfully found the minimum point, whereas Minuit has reached the minimum point in 6 cases. On the other hand, in these 6 cases COMBI used for the search 2 to 10 times more function evaluations than Minuit (the last function is extremely simple and is used only to test the correct work of the modified Newton's method).

It is interesting of course to check the algorithm with functions of more parameters. Table 2 demonstrates similar minimization results for twelve different functions F_i of 4 variables. It is more difficult to construct some interesting functions in the 4-dimensional space, so the first 10 functions $F_i(x_1, \ldots, x_4)$ were constructed of the former 2-dimensional functions $f_k(x,y)$ in the following way

$$F_i(x_1,\ldots,x_4) = f_{2i-1}(x_1,x_2) + f_{2i}(x_3,x_4) + f_{2i-1}(x_1,x_2) \cdot f_{2i}(x_3,x_4) \quad (6)$$

The function number 11 is defined as follows

$$F_{11}(x_1,x_2,x_3,x_4) = \rho + 100 \sin^2(10\rho - \varphi_1 - 2\varphi_2 - 3\varphi_3) \quad (7)$$

Table 1: Results of minimization of 20 two-dimension functions $f_i(x_1, x_2)$ by COMBI algorithm and MINUIT. Δr is a distance to the "true" minimum point, N_{cal} equals the number of function evaluations, which were used to reach a minimum point

Function Number	f_{min} $\Delta r/N_{cal}$				
	MINUIT command				COMBI
	SEEK	SIMPLEX	MIGRAD	MINIMIZE	
1	$5.8 \cdot 10^{-5}$ 0.0/21265	$5.0 \cdot 10^{-2}$ 0.2/32	16.0 1.5/45	16.0 1.5/45	$7.4 \cdot 10^{-4}$ 0.1/550
2	$2.5 \cdot 10^{-4}$ 0.0/36631	0.86 9.3/87	$1.7 \cdot 10^{-6}$ 0.0/280	$1.7 \cdot 10^{-6}$ 0.0/280	$4.0 \cdot 10^{-8}$ 0.0/1801
3	$6.9 \cdot 10^{-5}$ 0.0/31298	9.8 3.3/48	9.77 3.1/99	9.77 3.1/99	$2.4 \cdot 10^{-32}$ 0.0/10038
4	$3.6 \cdot 10^{-4}$ 0.0/21320	$4.6 \cdot 10^{-4}$ 0.0/33	100.1 11.0/46	Abend	$7.7 \cdot 10^{-7}$ 0.0/551
5	$5.5 \cdot 10^{-4}$ 0.0/15100	0.18 0.7/47	989.0 9.9/56	$2.2 \cdot 10^{-6}$ 0.0/210	$1.5 \cdot 10^{-6}$ 0.0/522
6	0.56 11.1/12014	0.22 19.0/103	0.28 19.1/146	0.22 19.1/223	$3.8 \cdot 10^{-5}$ 0.0/27643
7	26.5 0.0/51453	0.21 0.0/138	1309.0 20.4/97	1287.5 20.4/202	$3.8 \cdot 10^{-5}$ 0.0/54474
8	78.7 56.0/22053	69.3 52.6/77	68.6 52.4/190	68.6 52.4/190	$6.2 \cdot 10^{-6}$ 0.0/16405
9	$1.8 \cdot 10^{-4}$ 0.0/23639	1.8 1.4/216	$5.4 \cdot 10^{-6}$ 0.0/598	$5.4 \cdot 10^{-6}$ 0.0/598	$3.4 \cdot 10^{-7}$ 0.0/4621

Table 1: continued

Function Number	f_{min} $\Delta r/N_{cal}$ MINUIT command: SEEK	SIMPLEX	MIGRAD	MINIMIZE	COMBI
10	57.5 18.4/24941	60.4 19.9/65	$8.7 \cdot 10^{-8}$ 0.0/5222	$8.7 \cdot 10^{-8}$ 0.0/5222	$1.1 \cdot 10^{-6}$ 0.0/51504
11	$5.2 \cdot 10^{-4}$ 0.0/24105	$8.1 \cdot 10^{-8}$ 0.1/55	$1.0 \cdot 10^{-26}$ 0.0/35	$1.0 \cdot 10^{-26}$ 0.0/35	$5.7 \cdot 10^{-12}$ 0.0/353
12	5.44 5.3/24218	10.1 10.1/72	13.9 13.9/125	13.9 13.9/206	$1.0 \cdot 10^{-6}$ 0.0/14308
13	$4.4 \cdot 10^{-2}$ 0.0/26914	4.4 4.4/32	5.1 5.1/60	5.1 5.1/60	$1.7 \cdot 10^{-4}$ 0.0/54158
14	10.2 9.2/22110	22.0 20.1/55	26.2 19.8/112	21.7 19.8/229	$2.5 \cdot 10^{-5}$ 0.0/8512
15	9.9 26.4/17720	9.6 26.8/60	9.6 26.8/159	9.6 26.8/159	$2.6 \cdot 10^{-5}$ 0.0/20075
16	6.4 4.7/18445	8.0 6.3/81	4.8 3.8/187	4.8 3.8/187	$1.1 \cdot 10^{-6}$ 0.0/15449
17	26.0 11.3/17425	24.1 19.0/84	15.0 11.2/156	15.0 11.2/191	$1.4 \cdot 10^{-5}$ 0.0/31227
18	11903.8 8.5/25779	107.8 8.5/90	110.9 8.1/95	107.3 8.1/145	$2.6 \cdot 10^{-5}$ 0.0/40544
19	6.4 54.3/16197	6.4 54.2/58	6.4 53.8/179	6.4 53.8/179	$3.6 \cdot 10^{-5}$ 0.0/18054
20	$1.4 \cdot 10^{-8}$ 0.0/14165	$7.0 \cdot 10^{-8}$ 0.0/62	$8.4 \cdot 10^{-28}$ 0.0/45	$8.4 \cdot 10^{-28}$ 0.0/45	$2.4 \cdot 10^{-21}$ 0.0/24

where

$$\rho = \sqrt{(x_1+1)^2 + (x_2+1)^2 + (x_3+1)^2 + (x_4+1)^2},$$
$$\varphi_1 = arctg \frac{\sqrt{(x_2+1)^2+(x_3+1)^2+(x_4+1)^2}}{x_1+1},$$
$$\varphi_2 = arctg \frac{\sqrt{(x_3+1)^2+(x_4+1)^2}}{x_2+1},$$
$$\varphi_3 = arctg \frac{x_4+1}{x_3+1}$$
$$\min F_{11} = F_{11}(-1,-1,-1,-1) = 0$$

The last function F_{12} is a quadratic form:

$$\begin{aligned} F_{12}(x_1,x_2,x_3,x_4) &= (x_1+x_2+x_3+x_4+4)^2 + \\ &+100 \cdot (x_1 - 2x_2 + 3x_3 - 4x_4 - 2)^2 + \\ &+100 \cdot (x_1 + x_2 - 2x_3 - 2x_4 - 2)^2 + \\ &+100 \cdot (x_1 + 2x_2 + 2x_3 - 3x_4 + 2)^2 \end{aligned} \quad (8)$$

with $\min F_{12} = F_{12}(-1,-1,-1,-1) = 0$.

The limit on the number of function calls was set to 10^5 for all algorithms except for the random search SEEK of MINUIT program, where parameter *maxcalls* was set to 10^4. A starting point for minimization was always equal to $(1,1,1,1)$. Here the program COMBI found the "true" minimum point in 8 cases from the total of 12 functions. Minuit found the minimum point in 3 cases (the failture of COMBI was always followed by that of Minuit). In those cases when both programs failed, the minimum function value of COMBI was usually considerably lower than that of Minuit. Again, for successful minimizations Minuit works $(4 \div 10)$ times faster than COMBI.

The table 3 shows results of the minimization of 8 functions R_i of 8 parameters. The first 6 functions are constructed from the previous 12 functions of 4 parameters:

$$R_i(x_1,\ldots,x_8) = F_{2i-1}(x_1,x_2,x_3,x_4) + F_{2i}(x_5,x_6,x_7,x_8) \quad (9)$$

Table 2: Results of minimization of 12 four-dimension functions F_i by COMBI algorithm and MINUIT. Δr is a distance to the "true" minimum point, N_{cal} equals the number of function evaluations, which were used to reach a minimum point

Function Number	F_{min} $\Delta r/N_{cal}$ — MINUIT command: SEEK	SIMPLEX	MIGRAD	MINIMIZE	COMBI
1	$8.9 \cdot 10^{-2}$ 2.9/32235	5.74 23.6/129	16.0 1.5/476	16.0 1.5/476	$3.1 \cdot 10^{-4}$ 0.1/3814
2	0.15 13.7/19275	10.97 10.8/75	$5.3 \cdot 10^{-9}$ 0.0/1049	$5.3 \cdot 10^{-9}$ 0.0/1049	$9.2 \cdot 10^{-7}$ 0.0/3939
3	1434.1 15.9/11803	0.298 21.0/184	0.312 18.9/279	0.195 18.8/408	$2.3 \cdot 10^{-5}$ 0.0/36570
4	$2.5 \cdot 10^{+7}$ 29.1/43114	$8.8 \cdot 10^{+4}$ 53.3/276	Abend	Abend	$1.2 \cdot 10^{-4}$ 0.1/84782
5	370.4 18.9/37398	1099.8 24.4/148	60.8 19.5/1318	60.8 19.5/1318	$6.4 \cdot 10^{-5}$ 0.0/100009
6	449.5 9.4/17312	809.4 13.2/200	438.1 9.8/181	5.15 5.0/488	$8.9 \cdot 10^{-4}$ 0.0/47901
7	61.6 11.4/17268	125.4 19.7/338	171.7 20.5/193	116.5 20.5/338	3.2 3.2/24986
8	$1.2 \cdot 10^{+7}$ 28.0/33588	$1.2 \cdot 10^{+7}$ 28.1/266	92.6 27.5/326	90.9 27.5/519	78.1 58.2/40931
9	$1.6 \cdot 10^{+10}$ 12.5/11319	2691.6 20.4/310	Abend	Abend	107.8 8.5/89885
10	101.8 54.1/29667	3143.7 56.4/192	801.7 54.8/388	6.56 54.2/650	6.61 56.3/18726
11	4.04 4.0/10276	3.99 4.0/32	$1.2 \cdot 10^{-8}$ 0.0/6823	$1.2 \cdot 10^{-8}$ 0.0/6823	$1.2 \cdot 10^{-3}$ 0.0/88470
12	0.138 0.0/29407	$9.2 \cdot 10^{-8}$ 0.0/164	$1.7 \cdot 10^{-20}$ 0.0/93	$1.7 \cdot 10^{-20}$ 0.0/93	$3.3 \cdot 10^{-19}$ 0.0/47

Table 3: Results of minimization of 8 eight-dimension functions R_i by COMBI algorithm and MINUIT. Δr is a distance to the "true" point of minimum, N_{cal} equals the number of function evaluations, which were used to reach a minimum point

Function Number	R_{min} $\Delta r/N_{cal}$ — MINUIT command: SEEK	SIMPLEX	MIGRAD	MINIMIZE	COMBI
1	14.0 16.5/14584	28.7 26.2/268	16.0 1.6/1798	16.0 1.6/1798	$2.0 \cdot 10^{-5}$ 0.0/29573
2	$2.3 \cdot 10^{+5}$ 60.6/66877	$1.9 \cdot 10^{+5}$ 61.4/564	Abend	Abend	78.4 58.1/100346
3	1387.5 23.6/22257	1430.9 25.2/979	54346.3 34.5/1313	53523.4 33.0/3086	112.1 20.4/108464
4	$1.3 \cdot 10^{+7}$ 30.6/21796	$1.2 \cdot 10^{+7}$ 32.3/592	Abend	Abend	219.9 32.4/100601
5	$2.8 \cdot 10^{+9}$ 54.9/16458	$1.0 \cdot 10^{+7}$ 255.1/100000	Abend	Abend	1828.7 57.7/100799
6	5.67 4.4/57156	13.3 4.6/268	$1.0 \cdot 10^{-2}$ 0.0/11929	$8.6 \cdot 10^{-4}$ 0.0/12813	2.9 2.9/101047
7	19.9 16.7/40834	$1.8 \cdot 10^{+5}$ 16.7/1502	1.69 16.9/1943	1.69 16.9/1943	1.66 16.6/34378
8	39.6 0.6/17793	114.5 1.2/427	$9.8 \cdot 10^{-6}$ 0.0/240	$9.8 \cdot 10^{-6}$ 0.0/240	$2.2 \cdot 10^{-9}$ 0.0/136

One more function is defined by the following formula:

$$\begin{aligned}R_7 &= 1000 \times \Big[(x_1 + 1 - \rho\cos(5\rho))^2 \\ &+ (x_2 + 2 - \rho\sin(5\rho)\cos(6\rho))^2 + \\ &+ (x_3 + 3 - \rho\sin(5\rho)\sin(6\rho)\cos(7\rho))^2 + \\ &+ (x_4 + 4 - \rho\sin(5\rho)\sin(6\rho)\sin(7\rho)\cos(8\rho))^2 + \\ &+ (x_5 + 5 - \rho\sin(5\rho)\sin(6\rho)\sin(7\rho)\sin(8\rho)\cos(9\rho))^2 + \\ &+ (x_6 + 6 - \rho\sin(5\rho)\sin(6\rho)\sin(7\rho)\sin(8\rho)\sin(9\rho) \times \\ &\times \cos(10\rho))^2 + \\ &+ (x_7 + 7 - \rho\sin(5\rho)\sin(6\rho)\sin(7\rho)\sin(8\rho)\sin(9\rho) \times \\ &\times \sin(10\rho)\cos(11\rho))^2 + \\ &+ (x_8 + 8 - \rho\sin(5\rho)\sin(6\rho)\sin(7\rho)\sin(8\rho)\sin(9\rho) \times \\ &\times \sin(10\rho)\sin(11\rho)\cos(12\rho))^2\Big] + 0.1\rho \end{aligned} \quad (10)$$

where

$$\rho = \sqrt{\sum_{i=1}^{8}(x_i + i)^2}$$

And the last function is a quadratic form:

$$\begin{aligned}R_8 &= (x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + 8)^2 + \\ &+200 \times (x_1 - x_2 + 2x_3 + 2x_4 + 2x_5 + 2x_6 + 2x_7 + 2x_8 + 8)^2 + \\ &+150 \times (x_1 - 2x_2 + 3x_3 - 3x_4 + 3x_5 - 3x_6 + 2x_7 - 2x_8 + 8)^2 + \\ &+300 \times (x_1 - 3x_2 + 2x_3 - 2x_4 + 4x_5 + 2x_6 + x_7 - 3x_8 + 8)^2 + \\ &+100 \times (x_1 - 4x_2 + x_3 + 5x_4 - 6x_5 + 7x_6 - 8x_7 + 9x_8 + 8)^2 + \\ &+100 \times (x_1 + 2x_2 - 3x_3 + 4x_4 - 5x_5 + 6x_6 - 7x_7 + 8x_8 + 8)^2 + \\ &+400 \times (x_1 + 3x_2 - 4x_3 + 3x_4 - 2x_5 + x_6 + 3x_7 - 4x_8 + 8)^2 + \\ &+250 \times (x_1 + 4x_2 - 5x_3 - 4x_4 + 3x_5 - 2x_6 - x_7 + x_8 + 8)^2 \end{aligned} \quad (11)$$

In this check there is only one function where both programs had success, but since it is an exact quadratic form very convenient for COMBI, we can't use it to compare the convergence time. The only case when Minuit performed successful minimization (function number 6) appeared to be very difficult for COMBI. Minimization stopped due to a function calls limit. If this limitation is removed, then COMBI reaches a minimum point after 347813 function evaluations (that is 30 times more than Minuit). Among the other four variants where COMBI stopped by N_{cal} limitation, waiving this limitation has not helped. So the final score of searched minimum points for 8 dimensions is: 3 found by COMBI (without limit on N_{cal}) and 2 found by Minuit for 8 different functions. When both programs failed to reach the "true" minimum point, estimation of COMBI was usually much better than that of Minuit.

All these test functions look artificial and exotic. So the last trial of the algorithm is performed with a function typical for data processing in high energy physics experiments. Let us fit the "experimental data" with a resonance curve defined by a simple formula:

$$\rho(W) = \frac{N_m p^3(W)\Gamma^2 M^2}{p^3(M)\cdot[(W^2 - M^2)^2 + \Gamma^2 M^2]} + b \quad (12)$$

where

$$p(W) = \begin{cases}\sqrt{(W/2)^2 - m^2}, & W > 2m \\ 0, & W < 2m\end{cases}$$

Here we have 5 free parameters: M, Γ, m, N_m, b. Let us choose such "true" values for them: $M = 1020$, $\Gamma = 4$, $m = 490$, $N_m = 1000$, $b = 10$. Then for 21 energy points $W_i = 1010, 1011, 1012, \ldots, 1029, 1030$ let us calculate the "experimental" numbers of events: $n_i = \rho(W_i)$. For the first test let us use the fractional number of events that provides the known minimum value of the likelihood function and optimal parameters estimation.

The log-likelihood function can be used in the following form:

$$L = \begin{cases}\sum_{i=1}^{21}[\rho(W_i) - n_i + n_i\ln(n_i/\rho(W_i))] \\ 10^{10} \times (1 + 2m - M), \text{ if } M < 2m \\ 10^{10} \times (1 - b), \text{ if } b < 0 \\ 10^{10} \times (1 - \rho(W_k)), \text{ if any } \rho(W_k) < 0\end{cases} \quad (13)$$

The Table 4 shows the results obtained with different minimization programs. Starting point is always $M = 1015$, $\Gamma = 3.5$, $m = 450$, $N_m = 900$, $b = 1$. Initial steps are equal to 0.1.

The table 5 presents results of the fit, when the "experimental" data are smeared according to Poisson distribution. Of course here we do not know the "true" minimum point and minimum function value.

In order to test the program with two times greater number of free parameters let us do the simultaneous fit of "experimental" data with three decay modes of the resonance. Let the "true" values of the resonance parameters be $M = 1020$, $\Gamma = 4$, $m_1 = 490$, $N_{m1} = 1000$, $b_1 = 10$, $m_2 = 490$, $N_{m2} = 1000$, $b_2 = 10$, $m_3 = 0$, $N_{m3} = 500$, $b_3 = 20$. After evaluation

Table 4: Results of minimization of log-likelihood function with different minimization routines. Number of "experimental" events are equal to average value of events with "true" parameters of the resonance curve

Minimization routine	MINUIT				COMBI
	SEEK	SIMPLEX	MIGRAD	MINIMIZE	
min L	6.60	11.92	$3.7 \cdot 10^{-7}$	$3.7 \cdot 10^{-7}$	$3.7 \cdot 10^{-8}$
Number of FCN calls	85466	160	352	352	769
M	1020.0	1020.0	1020.0	1020.0	1020.0
Γ	4.3	4.5	4.0	4.0	4.0
m	486.9	487.3	490.0	490.0	490.0
N_m	922.8	900.1	1000.0	1000.0	1000.0
b	5.7	1.2	10.0	10.0	10.0

Table 5: Results of minimization of log-likelihood function with different minimization routines. Number of "experimental" events are smeared around the average value and rounded

Minimization routine	MINUIT				COMBI
	SEEK	SIMPLEX	MIGRAD	MINIMIZE	
min L	28.724	32.511	21.919	21.919	21.919
Number of FCN calls	100001	191	275	275	871
M	1019.96	1019.96	1019.98	1019.98	1019.98
Γ	4.263	4.367	4.032	4.032	4.032
m	493.06	491.90	491.49	491.49	491.48
N_m	920.87	900.40	999.53	999.53	999.52
b	2.42	1.38	2.11	2.11	2.10

of all expected average "experimental" number of events, let us "smear" them in accordance with Poisson distribution. Results of the minimization of the likelihood function over 10 free parameters are shown in Table 6 (parameter m_3 is fixed at the known value 0). The starting point is $M = 1015$, $\Gamma = 3.5$, $m_1 = 450$, $N_{m1} = 900$, $b_1 = 1$, $m_2 = 450$, $N_{m2} = 900$, $b_2 = 1$, $N_{m3} = 600$, $b_3 = 1$. Initial steps are equal to 0.1.

Table 6: Results of minimization of log-likelihood function with different minimization routines. Number of "experimental" events of the three decay modes of resonance are smeared around the average value and rounded

Minimization routine	MINUIT				COMBI
	SEEK	SIMPLEX	MIGRAD	MINIMIZE	
min L	114.97	108.926	41.115	41.115	41.115
Number of FCN calls	31187	275	764	764	2810
M	1020.09	1019.97	1019.97	1019.97	1019.97
Γ	4.159	4.312	3.941	3.941	3.941
m_1	450.80	491.53	492.39	492.39	492.38
N_{m1}	906.57	900.03	1010.37	1010.37	1010.36
b_1	7.47	1.00	4.44	4.44	4.43
m_2	454.71	488.55	490.69	490.69	490.68
N_{m2}	904.86	899.94	971.17	971.17	971.15
b_2	9.52	0.95	10.62	10.62	10.61
N_{m3}	586.52	599.96	500.05	500.05	500.06
b_3	7.97	0.99	21.60	21.60	21.60

4. Conclusions

Minimization strategy suggested in the present paper can be an alternative to the Variable-metric method in the cases when a minimized function has no derivatives or has a very complicated profile.

The main feature of this strategy is a combination of Simplex method and modified Newton's one. The probability of finding the "true" minimum point is increased by successive minimization from different starting points until the function values in the found minimum points coincide within the desired accuracy. This algorithm is implemented in the code COMBI, written in Fortran.

For smooth functions the time of convergence of COMBI to the minimum point is close to that of the MIGRAD algorithm of the well known program MINUIT.

The general purpose minimization routine COMBI was designed for use in event processing, so it has no such a brilliant service for interactive work and writes no messages to SYS$OUT float. Tested on thousands events from the SND detector [3], it has demonstrated very reliable work without arithmetic faults of the computer.

I would like to thank E.V.Pakhtusova and S.I.Eidelman for fruitful discussions.

References

[1] *F.James.* Function Minimization and Error Analysis. Reference Manual. CERN Program Library Long Writeup D506, March 1994

[2] *A.D.Bukin.*
On the Kinematic Reconstruction of Multiparticle Events. Reported at the "Computing in High Energy Physics" Conference, April 7-11, 1997, Berlin.
Optimal Rotation Procedure. Preprint Budker INP 97-50, Novosibirsk, 1997

[3] *V.M.Aulchenko et al.* SND — Detector for VEPP-2M and ϕ-factory. Proceedings of the Workshop on Physics and Detectors, Frascati, April 9-12, 1991, pp.605-613.
M.N.Achasov et al. Status of the experiments with SND detector at e^+e^- collider VEPP-2M. Preprint Budker INP 96-47, Novosibirsk, 1996.

[4] *Nelder J.A. and Mead R.* A simplex method for function minimization. Comput. J. 7 (1965) 308.

[5] *A.D.Bukin and S.I.Eidelman.* Computer in Planning and Data Processing of Experiment. Manual for students of the Physics Faculty of the Novosibirsk State University, Publishing Department of NSU, Novosibirsk, 1995 (in Russian).

[6] PAW Physics Analysis Workstation. An Introductory Tutorial. CERN Program Library Long Writeup Q121, CERN Geneva, Switzerland, 1995

A. Simple test program

The following simple test program was written to demonstrate how to use subroutine COMBI:

```
c ====
c                   Test of minimization routine COMBI:
c  Combined Simplex algorithm + modified Newton's method
c       INP RAS, Novosibirsk, Russia, September 10, 1997
c                                          A.D.Bukin
c ====
      implicit  none
      integer *4 NPAR,NCAL,ISG(2)
      real *8 Fmin,X(2),DFM
      integer i
      external TesFun
c NPAR is a number of the function arguments
      NPAR=2
c DFM is a desired accuracy
      DFM=0.01
c Ncal is a limit for the number of function evaluations
      Ncal=100000
      do i=1,NPAR
c ISG(i) is a flag of variable parameter:
c ISG(i)=1 means that parameter is variable
c ISG(i)=0 means that parameter is fixed
        ISG(i)=1
c X(i) is initial value of parameter
        X(i)=1.
      end do
      print 10,DFM,Ncal,(i,X(i),ISG(i),i=1,2)
10    format(t5,'Test of minimization routine COMBI:'/
     *       t5,'===================================='//
     *       ' Desired accuracy DFM=',f8.3/
     *       ' Limit for the number of function evaluations=',i5//
     *       ' Par.No.',t10,': Init.Value',t25,': Minim.flag'/
     *       1x,36('-'),t10,':',t25,':'/
     *       2(i5,t10,':',f8.2,t25,':',i8/))
      call COMBI(TesFun,NPAR,X,ISG,DFM,Ncal,Fmin)
      print 20,Ncal,Fmin,X
20    format(' Minimum searched after',i6,
     *       ' function evaluations'/
     *       ' Minimum function value Fmin=',1p,d10.3/
```

```
*       ' found at the point X1=',0p,f8.3,', X2=',f8.3)
      end
c --------------------------<========<<
      subroutine TesFun(NPAR,F,Xp)
      implicit none
      integer *4 NPAR
      real *8 F,Xp(NPAR),x,y,r,W
      parameter (W=5.d+0)
      x=Xp(1)+3.d+0
      y=Xp(2)+4.d+0
      r=sqrt(x**2+y**2)
      F=r+1.d+1*((x-r*cos(W*r))**2+(y-r*sin(W*r))**2)
      return
      end
c --------------------------<=======<<
include 'COMBI.FOR'
```

The profile of the minimized function used in this example (drawn with PAW code [6]) is shown in Fig. 1. The output listing of the test program is the following:

```
   Test of minimization routine COMBI:
   ==================================

Desired accuracy DFM=    0.010
Limit for the number of function evaluations=100000

Par.No. : Init.Value    : Minim.flag
--------:---------------:------------
   1    :    1.00       :     1
   2    :    1.00       :     1

 Minimum searched after 28993 function evaluations
 Minimum function value Fmin= 2.214D-06
 found at the point X1=  -3.000, X2=  -4.000
```

In Fig. 2 one can see the development of minimization. The path from starting point to the estimated minimum point for every Simplex minimization run is shown with an arrow. Dashed line connects the end of every arrow with the beginning of the next arrow. The stopping points for MINUIT program are shown by special symbols. In the modes MIGRAD, MINIMIZE and SIMPLEX the program stopped almost near the start point making the decision that the minimum point is found. SEEK mode obtained better point of minimum but it is also far from the "true" point. The minimization process

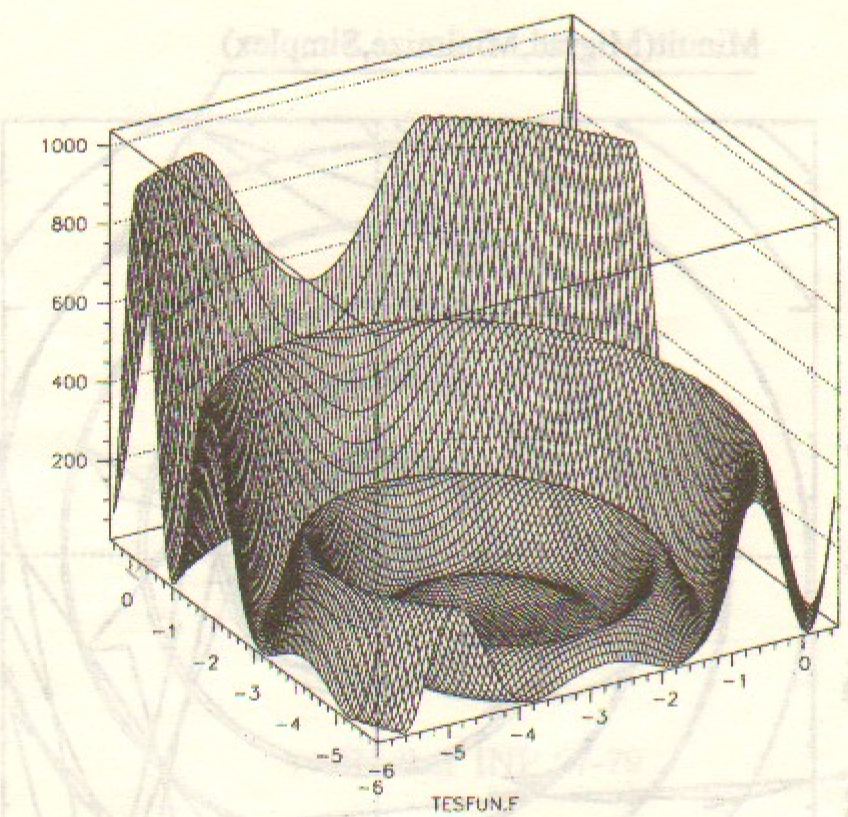

Figure 1: Profile of the test function

of COMBI algorithm seems to be almost chaotic, but it cannot get in the infinite loop and converges anyhow.

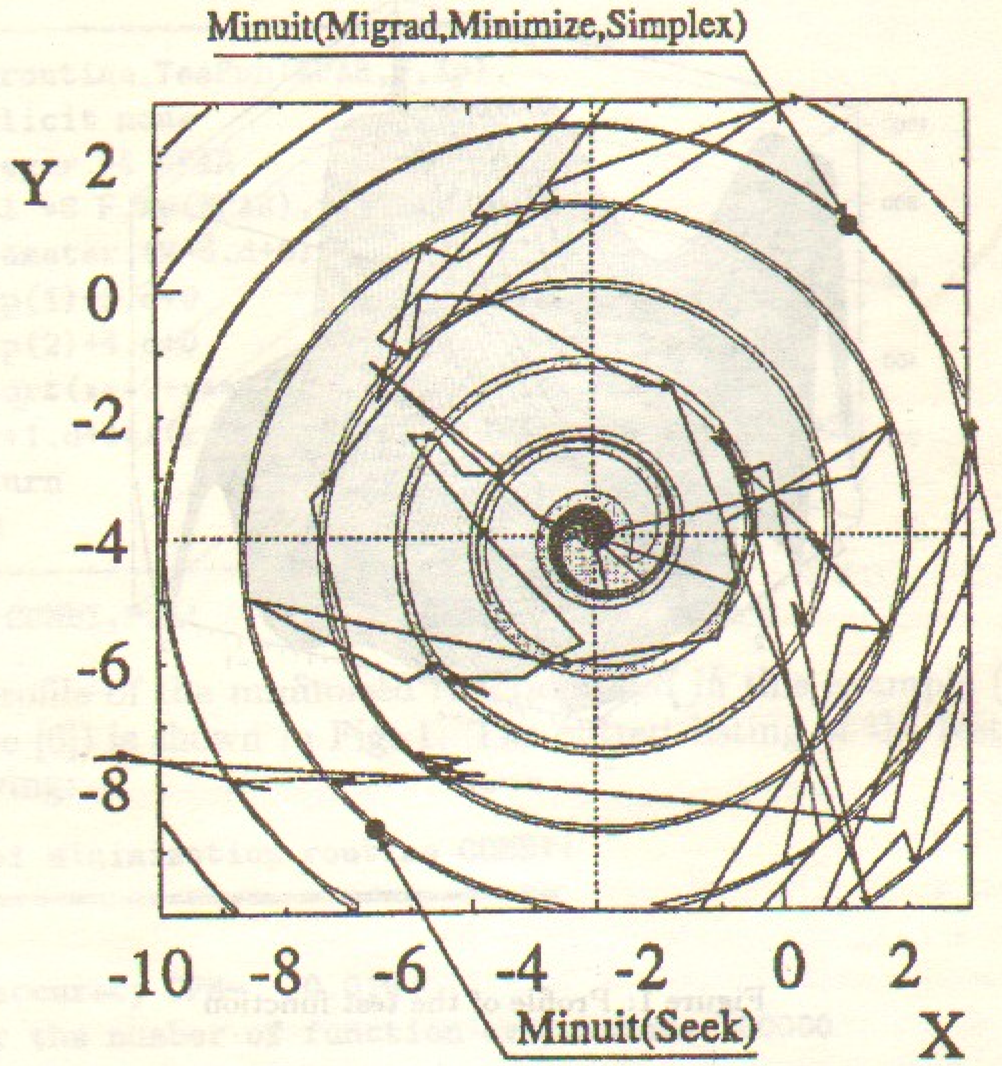

Figure 2: Schematic view of the development of minimization

A.D. Bukin

New minimization strategy

for non-smooth functions

Budker INP 97-79

Ответственный за выпуск А.М. Кудрявцев
Работа поступила 15.10. 1997 г.

Сдано в набор 15.10.1997 г.
Подписано в печать 15.10.1997 г.
Формат бумаги 60×90 1/16 Объем 1.7 печ.л., 1.4 уч.-изд.л.
Тираж 100 экз. Бесплатно. Заказ N° 79

Обработано на IBM PC и отпечатано на
ротапринте ИЯФ им. Г.И. Будкера СО РАН,
Новосибирск, 630090, пр. академика Лаврентьева, 11.