**Chem-Bio Informatics Journal, Vol. 3, No. 3, pp.114-129(2003)**

# Development of the parameter-fitting module for web-based biochemical reaction simulator BEST-KIT

Jun YOSHIMURA[1*], Tadahiro SHIMONOBOU[2], Tatsuya SEKIGUCHI[3], Masahiro OKAMOTO[1]

[1]*Graduate School of Bioresource and Bioenvironmental Sciences, Kyushu University, 6-10-1, Hakozaki, Higashi-ku, Fukuoka, 812-8581, Japan*
[2] *Department of Biochemical Engineering & Science, Faculty of Computer Science & Systems Engineering, Kyushu Institute of Technology, Iizuka-city, Fukuoka, 820-8502,Japan*
[3] *Department of Information Engineering, Faculty of Engineering, Maebashi Institute of Technology, 460-1, Kamisadori, Maebashi-city, Gunma, 371-0816, Japan*

**E-mail: jun@brs.kyushu-u.ac.jp*

(Received November 9, 2002; accepted July 30, 2003; published online September 10, 2003)

## Abstract

**We have implemented an efficient, user-friendly, and web-based "biosimulator" named BEST-KIT (Biochemical Engineering System analyzing Tool-KIT: http://www.best-kit.org) for analyzing large-scale nonlinear reaction networks such as metabolic pathways. The BEST-KIT mainly consists of a module, "MassAction++," that can construct and analyze a reaction scheme represented by both mass action law (mass balance) and approximated velocity functions of enzyme kinetics at steady state. This module was developed in Java applet style and can be carried out on "any" platform machine through a web browser. In this study, we developed the parameter-estimation module for MassAction++. This module can estimate the values of unknown kinetic parameters based on the experimentally observed time-course data of state variables. We adopted three optimization techniques, the modified Powell method, the genetic algorithm, and the Hybrid method, which incorporates the genetic algorithm into the modified Powell method. The user can use an appropriate method for each purpose.**

**Key Words:** Web-based simulator, computer simulation, optimization, metabolic pathway, bioinformatics

**Area of Interest:** Bioinformatics and Bio Computing


 

## 1. Introduction

In molecular biology, we often see various kinds of technical terms which involve the suffix "-ome" or "-omic", such as genome, proteome, metabolome, and so on. The "-ome" does not represent an individual component but, rather, a functional set or group of them. For example, a genome is a set of genes that is related to some functional property. Why are the "-ome" or "-omic" studies so widedly distributed? One of the answers is that the so-called "some functional property" results from the interplay of individual components or modules, which cannot be represented as an individual component; we should have a holistic and systemic view. Systems biology is the discipline that aims to integrate data generated by genomics, proteomic, and metabolomic analysis in order to gain a holistic view of living cells. If we focus on the metabolic pathways in the cell, however, a mere collection of static datasets on metabolites and enzymes will not provide us with a sufficient view. Inevitably, the dynamic behavior, that is a timed series of responses of target metabolites to external perturbation, should be incorporated to develop kinetic models of the selected pathway.

Since the phenomenon that new properties can arise in a system of nonlinearly interacting components, has been described by theoretical and mathematical biologists, a variety of biochemical network modeling packages as a tool-kit for analyzing a system's nonlinearity have been developed over the years and used by researchers interested in understanding the functional properties of metabolic pathways or cellular networks; Some of them are as follows: A-Cell [1], BioSpice [2], DBSolve [3], E-Cell [4], Genomic Object Net [5], Gepasi [6], JDesigner [7], MIST [8], PLAS [9], Virtual Cell [10], and WinSCAMP [11]. Recently, we also have implemented an efficient, user-friendly "biosimulator" named BEST-KIT [12][13] (Biochemical Engineering System analyzing Tool-KIT) for analyzing large-scale nonlinear reaction networks such as metabolic pathways. The main module of BEST-KIT, "MassAction++ [14]," can be used now from "any" platform machine through the web browser. Once the model and mathematical equations are generated, numerical integration can be accomplished to simulate a sequence of nonlinear biochemical reactions such as those found in metabolic pathways. However, because of a lack of information, the actual kinetics and parameter values could not be used. Without actual information on kinetic parameter values, there is no way to confirm that the simulated dynamics actually happened in the real metabolic pathways; and such simulations rarely convince molecular biologists. An efficient parameter estimation method or parameter-fitting function combined with the use of an experimentally observed timed series of data on the concentrations of target metabolites in the selected pathway should be incorporated into a network modeling package.

Among numerical optimization techniques, the directed-search method, which needs not to evaluate the gradient, is most suitable to analyze the dynamics of complex nonlinear control systems. The Powell method [15] and the modified Powell method [15][16] are well known to have an ultimate fast convergence among direct-search methods, especially where the cost function is well approximated by a quadratic form of target parameters. Those conventional optimization methods, however, show a strong power only when the initially guessed target parameter value is very near to the optimal solution; most of the conventional numerical optimizations can easily be trapped in local optima (minima). An efficient method for numerical optimization should be developed, which would show fast convergence even if the optimization were to start in a region of the parameter space far from the optimal solution. In this paper, to overcome this difficulty, we describe a newly developed hybrid optimization method, which is a unique combination of the Real-coded Genetic Algorithm [17] and the modified Powell method, followed by the incorporation of this method into the MassAction++ module in BEST-KIT.

## 2. Concept of MassAction++

In the development of MassAction++, we have considered some remarkable concepts as shown in Figure 1. The users need not be conscious of either troublesome equations or numerical calculations. Cumbersome simultaneous nonlinear differential equations of a constructed reaction scheme can be automatically produced without the need to write troublesome equations. Mathematical modeling is represented by using the mass action law (mass balance) and the approximated velocity function related to steady state enzyme kinetics. These equations are calculated by the Gear method [18] as one of the most efficient numerical calculation methods for "stiff" differential equations. As shown in Figure 1, MassAction++ has been designed as a "client-server system," where the heavy duty work for numerical calculation of the constructed scheme, which might require long cpu-time in a client machine, can be carried out in the server machine (a virtual cpu-server having high-performance cpu-capability) through the Internet, and the calculated numerical results can be sent back to the client and visualized as graph on there. Since we have adopted a dynamic allocation of memory for solving simultaneous differential equations, the maximum number of reactants that can be dealt with on MassAction++ depends only on the size of the main memory of the cpu-server. Since the cpu-server issues a unique task ID number for each calculation request from a client, many users can login and request calculations at once.

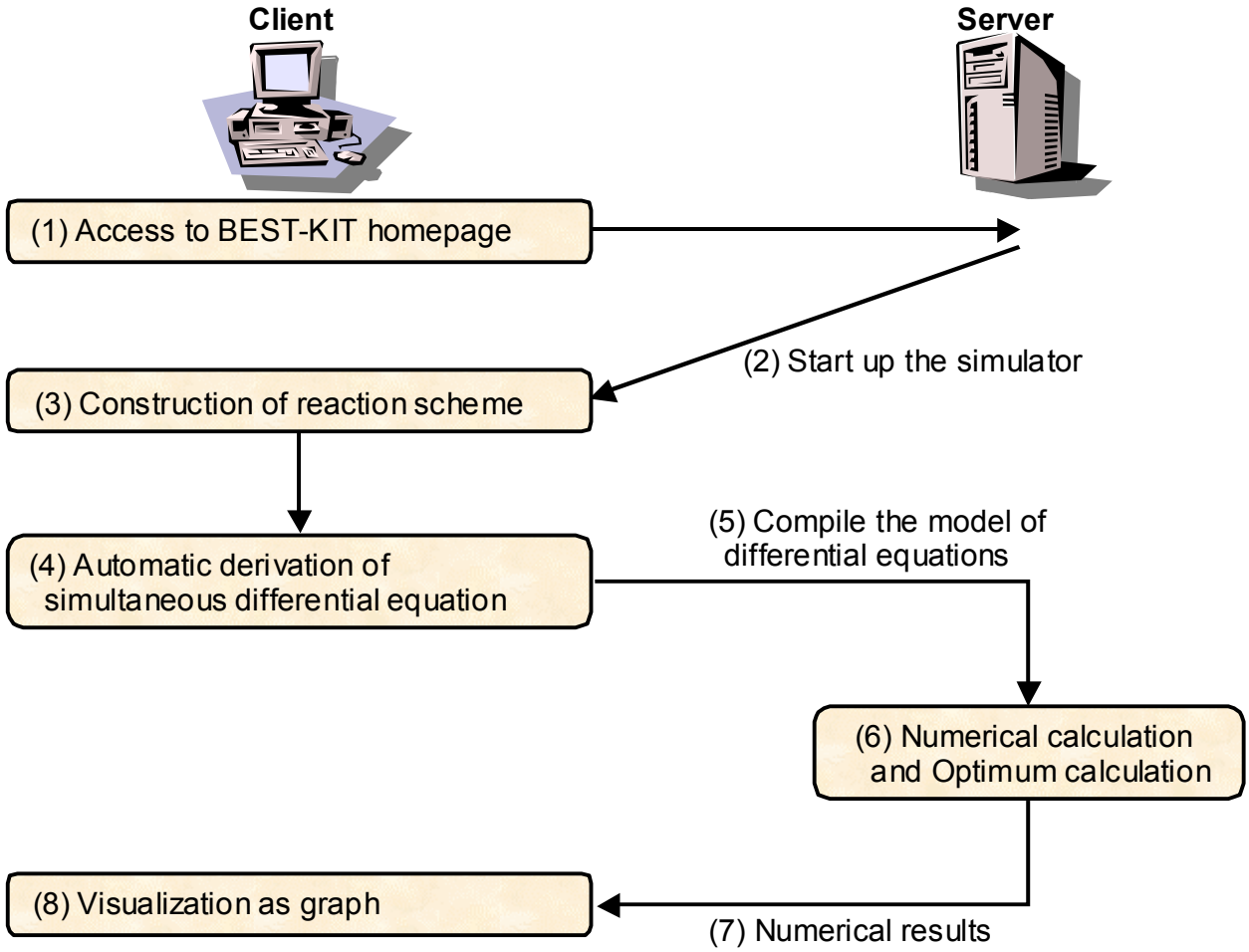


**Figure 1.** Procedure for simulation in MassAction++

MassAction++ has an efficient GUI (Graphical User Interface) as shown in Figure 2. By using the "mouse," the users can easily construct and update a reaction scheme in the editing window (working area) through the GUI, even when the number of reaction components (state variables) is relatively large. In this module, each reactant and reaction step comprising a reaction scheme is represented as a "symbol," and the users can construct reaction schemes by connecting the symbols with lines. This GUI provides the following four areas: (1) Menu bar area, (2) Working area, (3) Choice area, and (4) Input area. In the Menu bar area, there are several menus such as File (related to file handling such as save, upload, load), Edit (related to editing the constructed scheme), Calculate (related to numerical calculation) and Graph (related to visualization). The user can construct the reaction scheme within the Working area by selecting a suitable kinetic mechanism of a steady state enzymatic reaction appearing in the Choice area and by setting the initial concentrations and kinetic constants in the Input area.

We have developed the GUI of MassAction++ in Java applet style by using JDK (Java Developer's Kit) version 1.1.3, so that all users can run this simulator from "any" platform machine. There is a requirement, however, that the user's machine has to be equipped with web browser executable JDK1.1 (for example, at least Netscape Navigator 4.0 or Internet Explorer 4.0, etc) in order to use this simulator.

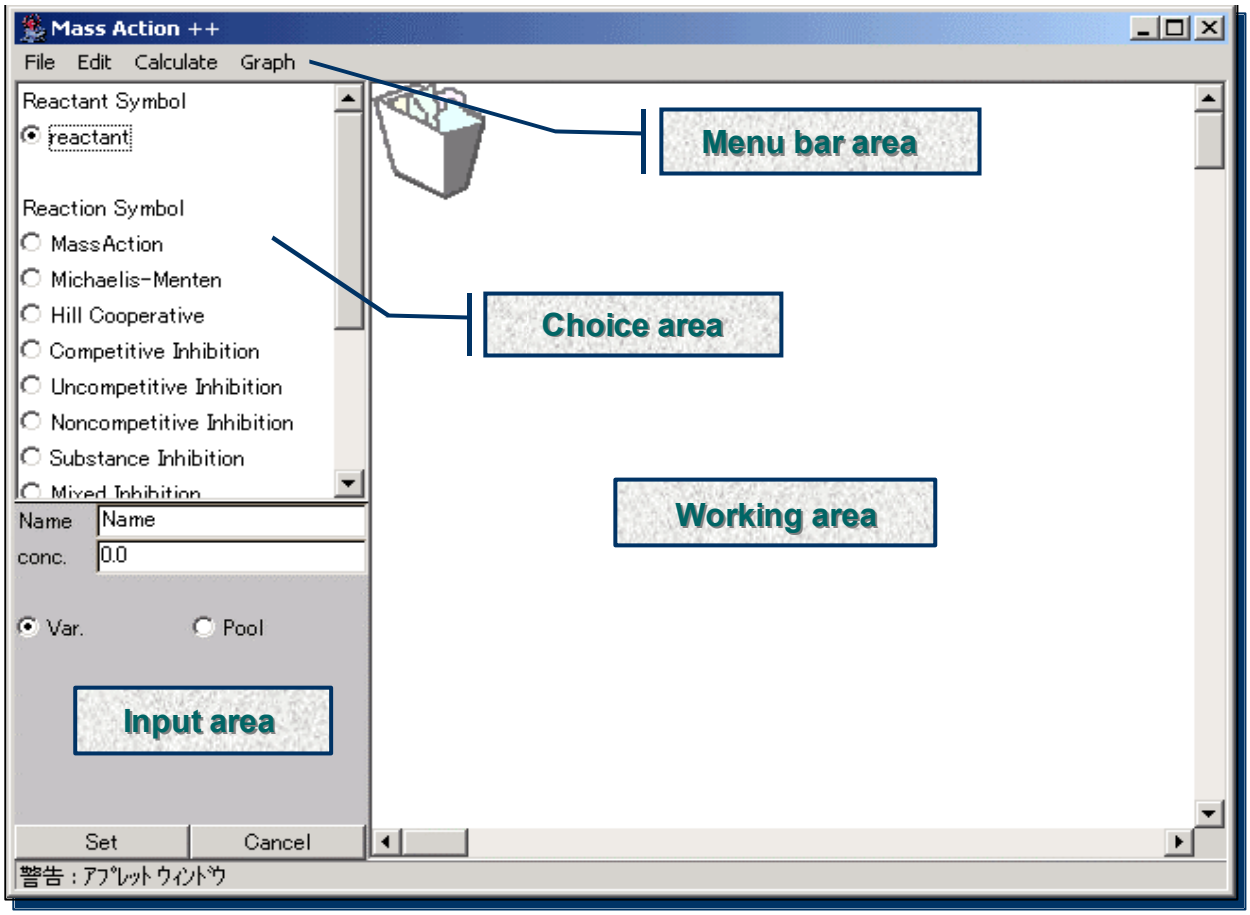


**Figure 2.** Snapshot of MassAction++

Figure 3 shows the procedures for the numerical calculation and optimization in BEST-KIT. (1) Cumbersome simultaneous nonlinear differential equations describing the scheme can be automatically produced without having to write troublesome equations. At this time, if the users select "MassAction" in Reaction Symbol, the differential equations produced will be based on mass action law (mass balance). If the users select a reaction symbol other than MassAction, differential equations will be produced based on a selected approximated equation of steady state enzyme kinetics. (2) The derived differential equations and the initial parameters are packed into a data class. The data class is then sent to the server through the Internet by using HORB [19] as a communication package that extends Java for distributed object computing to enable data communication between a client and the server through the Internet. HORB is the world's first Java ORB (Object Remote Broker) and is 100% compatible with Sun's Java language specification, interpreter, and Java classes. Of course, HORB includes the CORBA IDL complier and CORBA IIOP protocol and supports a very fast and functional proprietary protocol as well. The processing speed of HORB is very fast, reportedly twice the speed of other ORBs. (3) The data class received by the server is saved as a file by using a Java application program. The server produces a C-language source code for the mathematical representation of differential equations; (4) and the server compiles and links it to the numerical calculation program or the optimization program. (5) After the numerical calculation or the optimization has been executed, numerical results are saved as a file. (6) The result file is loaded by the Java application program, after which it is packed into a result data class. (7) These data classes are sent back to the client via the Internet by means of HORB. (8) These are visualized in graphic form on the client's machine.

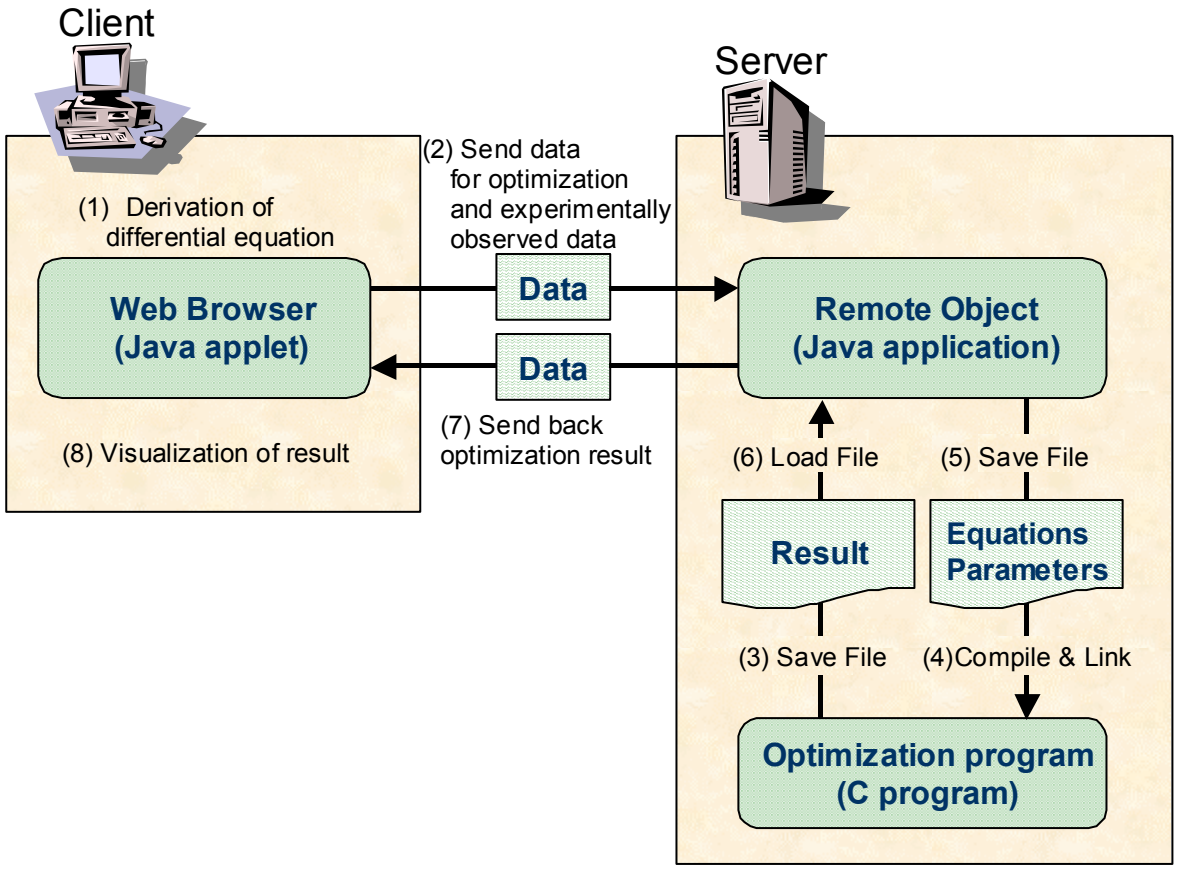


**Figure 3.** Procedures for the optimization in client-server

## 3. Parameter fitting

By using a parameter-fitting function in MassAction++, the users can estimate the values of unknown kinetic parameters based on the observed time-course data of reactants. Figure 4 represents the situation where both the kinetic values of inhibition constant (Ki1) in the competitive enzymatic reaction producing the product B from the substrate A (A → B), and the Michaelis constant (Km2) in the Michaelis-Menten reaction producing the product C from the substrate B (B → C) are supposed to be unknown, and two sets of experimental time-course data of reactants A, B, and C shown in Figure 5 are given. By assigning "unknown" to the estimated parameters and setting initial guess values for these parameters, the simulator estimates the values of the kinetic parameters that can realize the experimental time-course data by using nonlinear numerical optimization techniques such as the modified Powell method (MP), genetic algorithm (GA), or the Hybrid method (GA + MP). In this case, as shown in the upper side of Figure 6, the estimated values are 0.09017 for Ki at A → B, 0.133685 for Km at B → C, and the mean squared relative error between the calculated values and experimentally observed ones is 4.518585%. The lower side figures of Figure 6 represent the time courses of A, B, and C calculated by using estimated kinetic values and two sets of experimentally observed data (Figure 5).

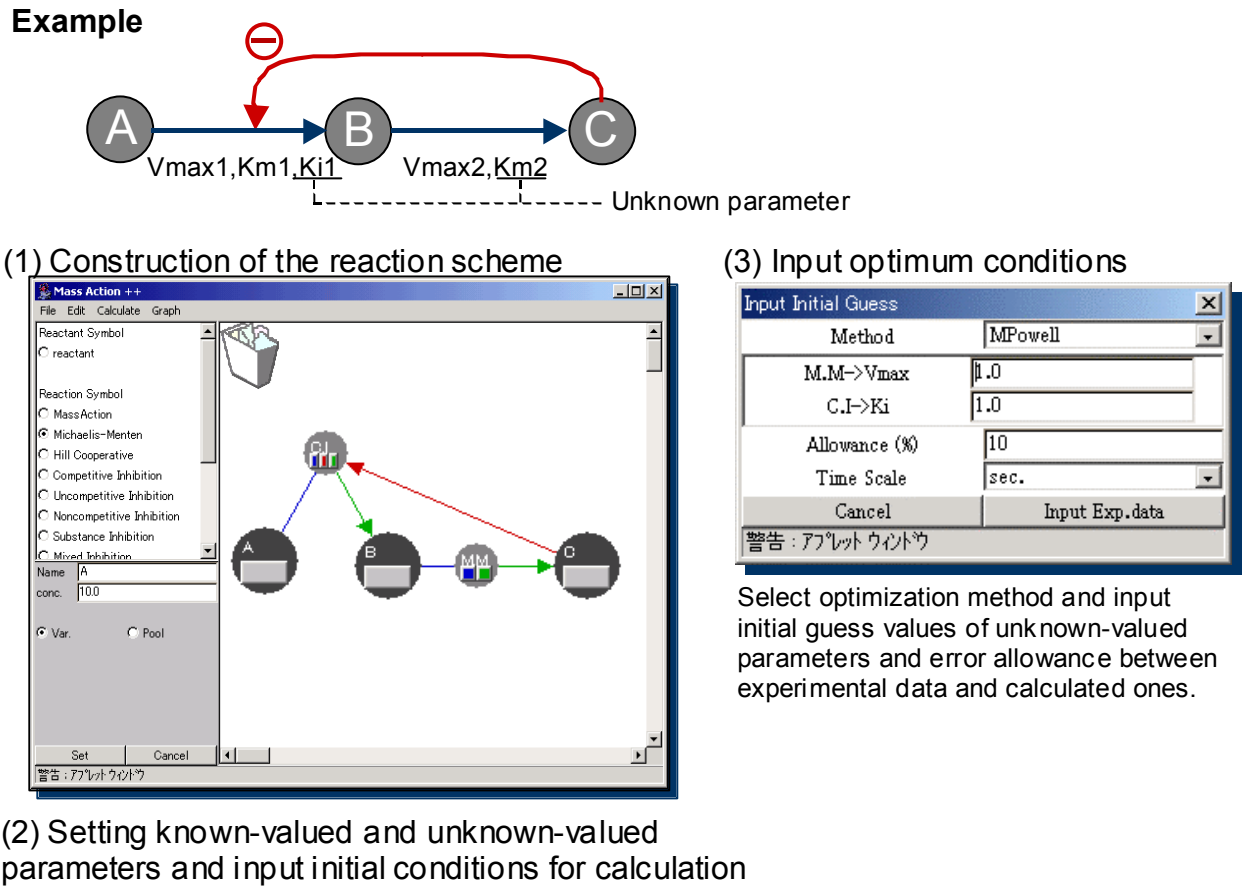


**Figure 4.** Parameter-fitting module

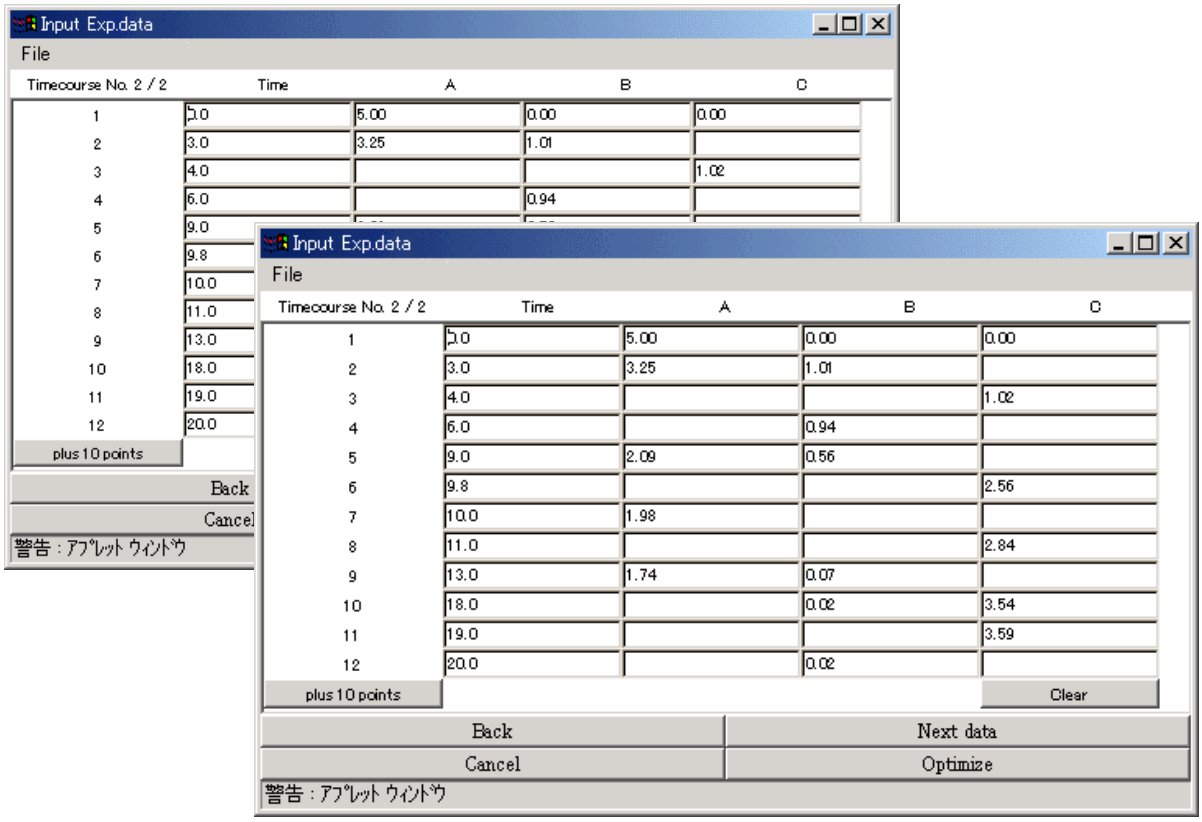


**Figure 5.** Experimentally observed time-course data

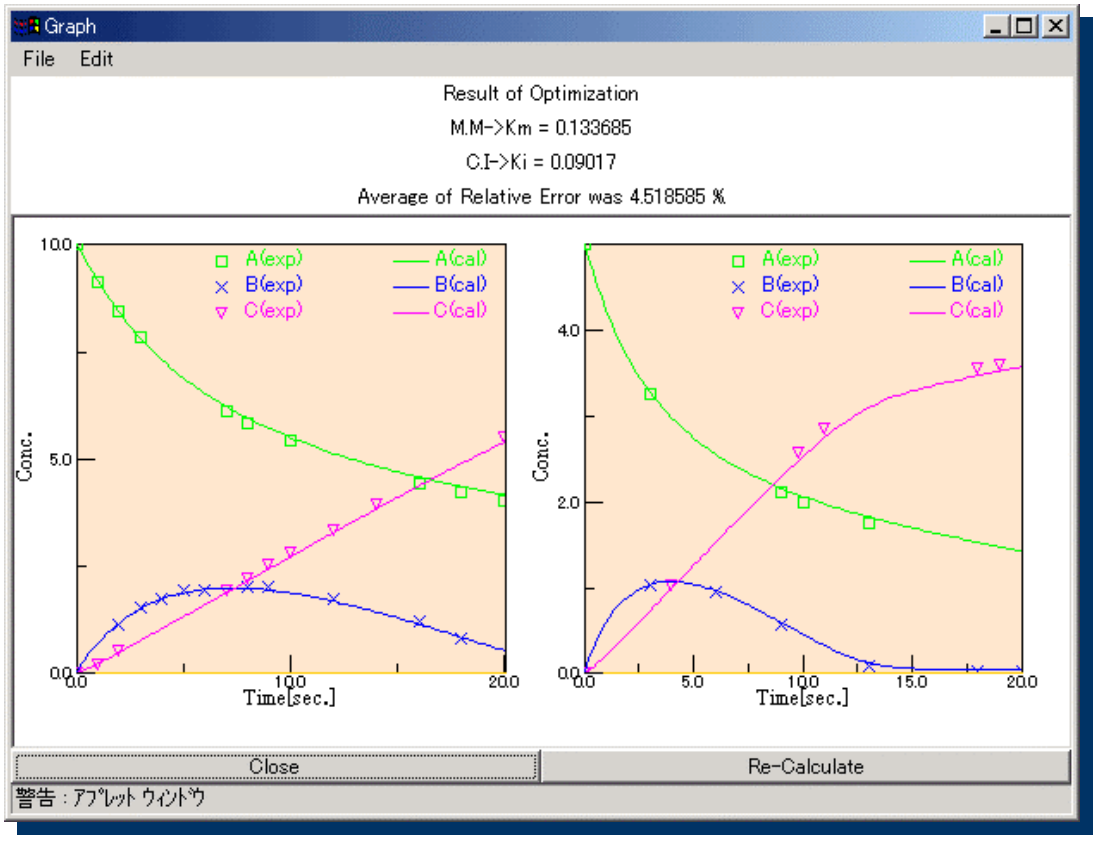


**Figure 6.** Result of parameter-fitting

# 4. Optimization techniques

MassAction++ adopts three optimization techniques for estimating the values of unknown kinetic parameters, the modified Powell method, the genetic algorithm, and the Hybrid method; and the users can use it properly according to their purpose. The features of each technique are described below.

## 4.1 Modified Powell method (MP)

The MP is well known to have an ultimate fast convergence among direct-search methods without using the derivative of the objective function, especially where the cost function is well approximated by a quadratic form of the parameters to be estimated; the MP has enough power to accelerate the dropping to a minimum point in quadratic form. This method has a critical disadvantage, however, that the minimum points obtained may not be the global minimum; most of the conventional conjugated gradient methods can easily be trapped in local optima (minima) by constraints in a region of the parameter space far from the optimal solution.

## 4.2 Genetic Algorithm (GA)

The GA is known as one of the heuristic algorithms that can seek out the global minimum; by escaping from being trapped in the local minimum, the "best" solutions will be found in regions of the parameter space containing a relatively high proportion of "good" solutions, and these regions can be explored and exploited by several genetic operations such as selection and crossover. The GA has the major disadvantage, however, that there is a considerably large computational cost (cpu time) for the large numbers of runs of the solution candidates. The actual procedure of a typical or conventional GA is as follows:

1) *Generation of Initial Population*
   Randomly generate an initial population of size *N*.
2) *Selection for Reproduction*
   Select a pair of individuals to become parents from the population.
3) *Generation of Offspring*
   Generate offspring by applying crossover and mutation to the selected pair of individuals in step 2).
4) *Selection for Survival*
   Choose two individuals from the family containing the parents and their offspring.
5) *Repeat the procedures between step 2 and step 4 until a certain condition for termination is satisfied.*

We adopted MGG [20] (Minimal Generation Gap) as the generation-alternation model shown in Figure 7.

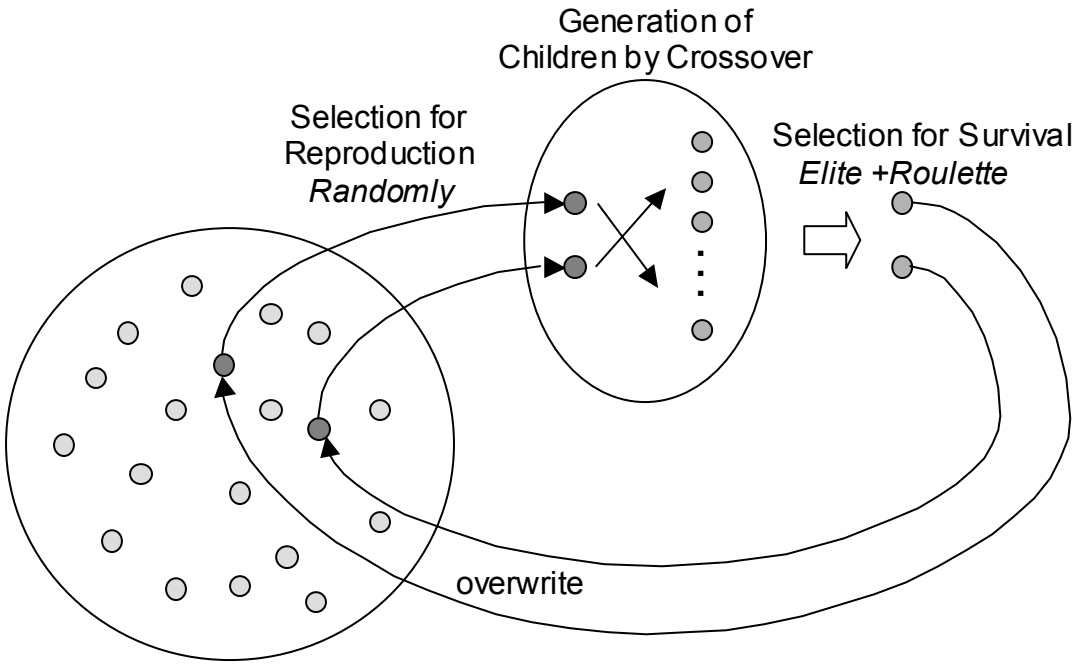


**Figure 7.** MGG (Minimal Generation Gap)

The MGG is a new generation alternation model that does not consider mutation. It is the most desirable model that can avoid early convergence and suppress evolutionary stagnation. The algorithm of the MGG is described as follows.

1) *Generation of Initial Population*
   Make an initial population that is composed of random real number vectors.

2) *Selection for Reproduction*
   Select a pair of individuals by random sampling without replacement from the population. The selected pair of individuals becomes parents of offspring.

3) *Generation of Offspring*
   Generate offspring by crossover between the selected pair of individuals.

4) *Selection for Survival*
   Select two individuals from the family comprising the parents and their offspring; one is the best individual (Elite individual) and the other is selected by rank-based roulette wheel selection (Roulette individual). Replace the parents chosen in Step 2) in the population with the two individuals.

5) *Repeat the procedures between step 2 and step 4 until a certain condition for termination is satisfied.*

In this study for the generation of children by crossover in Figure 7, UNDX [21] (Unimodal Normal Distribution Crossover) is applied as a crossover operation.

The UNDX generates offspring from a normal distribution area defined by three parents, as shown in Figure 8. Offspring are made around the line segment connecting two parents, Parent 1 ($\boldsymbol{p}_1$) and Parent 2 ($\boldsymbol{p}_2$). The mathematical representation of UNDX is as follows:

$$\boldsymbol{c}_1 = \boldsymbol{m} + z_1\boldsymbol{e}_1 + \sum_{k=2}^{l} z_k \boldsymbol{e}_k, \quad \boldsymbol{c}_2 = \boldsymbol{m} - z_1\boldsymbol{e}_1 - \sum_{k=2}^{l} z_k \boldsymbol{e}_k \qquad \boldsymbol{m} = (\boldsymbol{p}_1 + \boldsymbol{p}_2)/2$$

$$z_1: N(0, \ \sigma_1^2), \qquad z_k: N(0, \ \sigma_2^2) \quad (k=2, \ldots, l)$$

$$\sigma_1 = Ad_1, \sigma_2 = Bd_2/\sqrt{l}$$

$$\boldsymbol{e}_1 = (\boldsymbol{p}_2 - \boldsymbol{p}_1)/|\boldsymbol{p}_2 - \boldsymbol{p}_1|, \quad \boldsymbol{e}_i \perp \boldsymbol{e}_j \ (i \neq j) \quad (i,j = 1,2, \ldots, l),$$

where $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ are children, $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are parents, and $\boldsymbol{m}$ is the middle point of parents. Further, $d_1$ is the distance between two parents, and $d_2$ is the distance between the third parent $\boldsymbol{p}_3$ (randomly selected) and the line connecting $\boldsymbol{p}_1$ to $\boldsymbol{p}_2$; and $z_1$: $N(0, \ \sigma_1^2)$ and $z_k$: $N(0, \ \sigma_2^2)$ ($k = 2,\ldots,l$: number of estimated parameters) are normally distributed random numbers. A and B are constants given by the user.

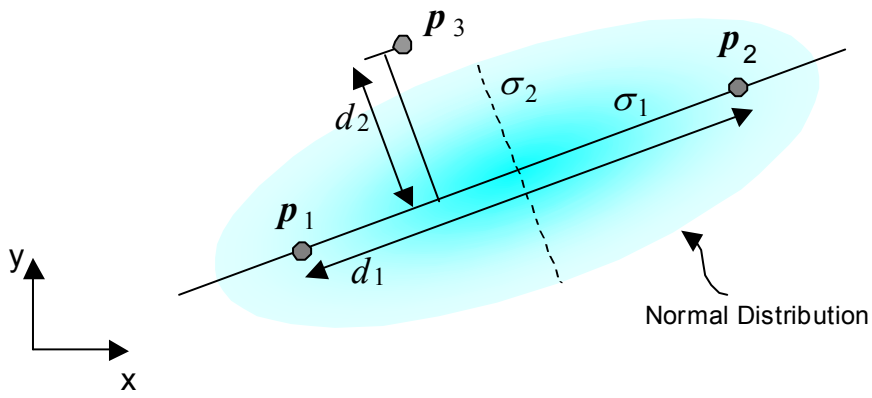


**Figure 8.** Searching region by using UNDX (Unimodal Normal Distribution Crossover) for the estimations of 2 parameter-values (x, y).

### 4.3 Hybrid method (GA+MP)

There are many variations on how to incorporate the GA into the MP [16][22]. As shown in Figure 7, in the MGG, two individuals are selected from the family comprising the parents and their offspring after the genetic operation; one is the best-fitted individual (elite individual) and the other is randomly selected. Since the MP can accelerate the dropping to the minimum point in quadratic form as shown in Figure 9, we have applied the MP to the selected two individuals. By using this unique hybrid method, fast convergence can be expected without the loss of effectiveness of the GA (MGG+UNDX shown in Figure 7 and Figure 8). The following is the actual procedure as is diagrammed in Figure 10.

1) *Generation of Initial Population*
   Make an initial population that is composed of random real number vectors.

2) *Selection for Reproduction*
   Select a pair of individuals by random sampling without replacement from the population.

3) *Generation of Offspring*
   Generate offspring by applying the UNDX to the selected pair of individuals.

4) *Selection for Survival*
   Select two individuals from the family comprising the parents and their offspring; one is the best individual (Elite individual) and the other is selected by rank-based roulette wheel selection (Roulette individual).

5) ***Apply the modified Powell method***
   Apply the modified Powell method to each of the two selected individuals in step 4) After applying the MP, replace the parents selected in Step 2) in the population with the two newly obtained individuals.

6) *Repeat the procedures from step 2 to step 4 until a certain condition for termination is satisfied.*

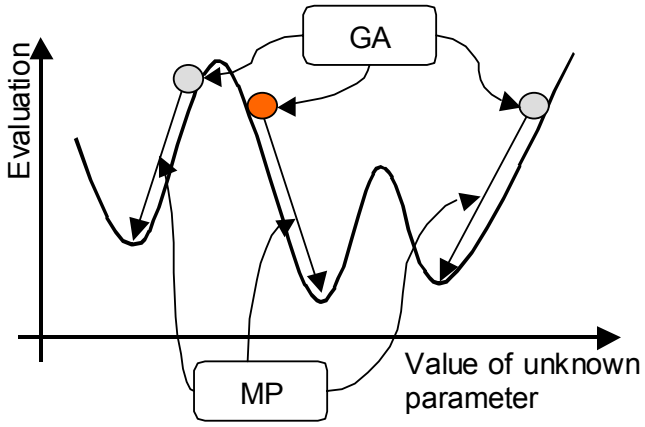


**Figure 9.** The conceptual diagram of the Hybrid method

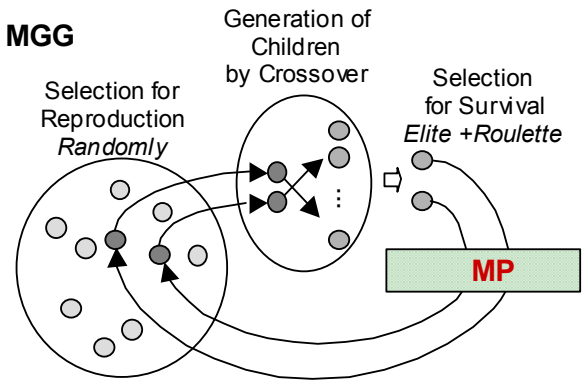


**Figure 10.** Procedure of the Hybrid method

## 5 Numerical Experiments

We examined the searching efficiency of three optimization methods: the MP, the GA, and the Hybrid method (GA+MP).

We assumed a Michaelis-Menten type reaction with two enzyme-substrate complexes, which is shown in the upper part of Figure 11. The initial concentrations of the reactants are as follows: [X1] = $1.0\text{x}10^{-6}$(M), [X2] = $1.0\text{x}10^{-4}$ (M), [X3] = [X4] = [X5] = 0.0 (M). The standard values for rate constants in the corresponding steps are fixed at k1 = $3.0\text{x}10^{7}$ $M^{-1}sec^{-1}$, k-1 = $3.0\text{x}10^{2}$ $sec^{-1}$, k2 = $3.0\text{x}10^{5}$ $sec^{-1}$, k-2 = $3.0\text{x}10^{4}$ $sec^{-1}$, k3 =7.2 $sec^{-1}$. The calculated time-course data for reactants under the standard parameter set are shown in the lower right side in Figure 11. These time-course data are treated as a given set of observed data for optimization in the following case study. As the condition for optimization, the targeted error allowance (tolerance) per sampling point is 5%. The population size of individuals is 100, and the number of children generated at each crossover is 100.

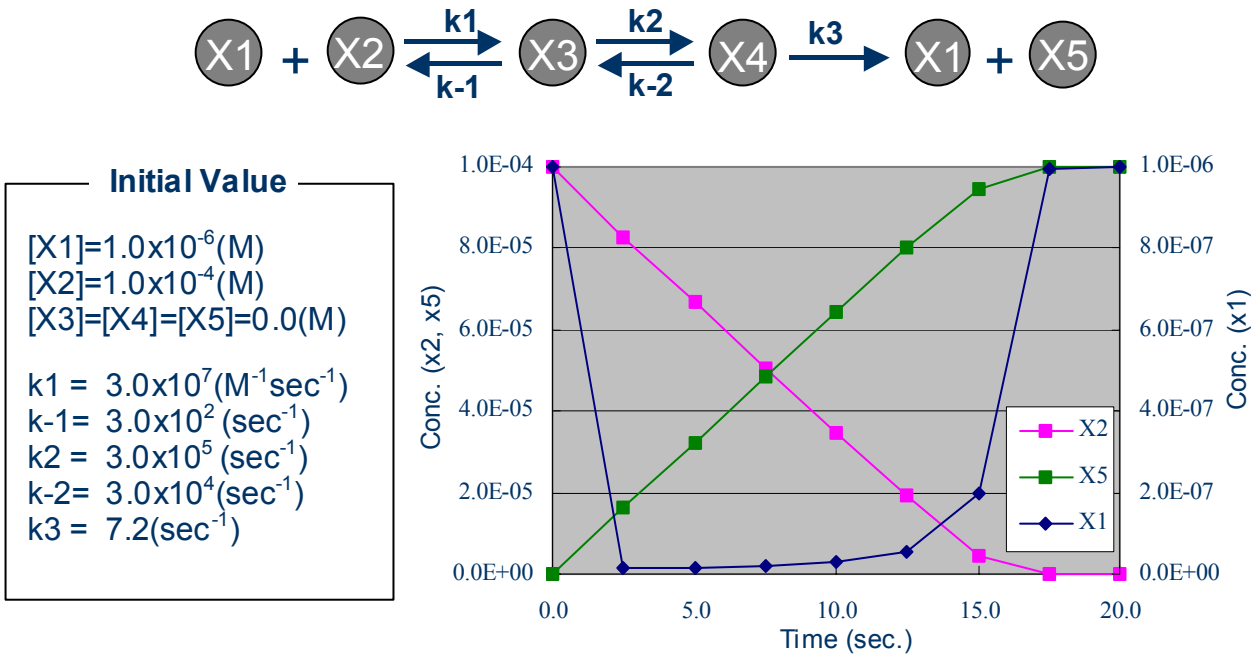


**Figure 11.** Case study

### 5.1 Case Study 1

As case study 1, assume the values of kinetic rate constants k1 and k3 are unknown. By using each optimization method, the task is to estimate those values that can realize the experimentally observed time course shown in Figure 11. By the GA and the Hybrid method, the searching region for k1 is estimated to be $[0.0,\ 1.0\times10^{8}]$ and [0.0, 15.0] for k3.

By applying the MP to this task, the following results are obtained (Table 1).

**Table 1.** Optimization results for the modified Powell method (the MP)

| Initial Guess | | Estimated Value | | Error (%) | cpu time* (sec.) |
|---|---|---|---|---|---|
| k1 | k3 | k1 | k3 | | |
| $1.0\times10^{7}$ | 7.0 | $3.2\times10^{7}$ | 7.14 | 3.70 | 7.1 |
| $1.0\times10^{7}$ | 9.0 | fail to estimate | | 27.36 | ——— |
| 1.0 | 6.5 | $2.8\times10^{7}$ | 7.25 | 3.31 | 7.8 |
| 1.0 | 6.0 | fail to estimate | | 19.03 | ——— |

*Ultra SPARC IIi 333MHz

As for k1, the optimal solution can be estimated from a wide distribution of initial guesses. Contrary to this, for k3, the optimal solution can be found only from an initial guess value that is very close to the solution; starting with initial guesses from the searching region [6.5, 7.0] leads to success and otherwise does not.

The optimization results for the GA and the Hybrid method are shown in Table 2.

**Table 2.** Optimization results for the GA and the Hybrid method within 20 trials (case study 1)

| | GA | Hybrid method (GA+MP) |
|---|---|---|
| No. of successes | 20/20 | 20/20 |
| No. of Evaluations | $2896 \pm 1638$ | $760 \pm 448$ |
| Average cpu time* (sec) | $173.2 \pm 99.6$ | $45.0 \pm 28.6$ |

*Ultra SPARC IIi 333MHz

The GA and the Hybrid method require more cpu time for optimization than does the MP; however, even when the searching region for k3 is very wide, such as [0.0, 15.0], these two methods could estimate the k3 value correctly. When these two methods are compared, we see that the Hybrid method can terminate optimization about 4 times faster than the GA, and within 20 trials the Hybrid method has a smaller standard deviation for average cpu time than does the GA.

### 5.2 Case Study 2

The previous case showed the superiority in searching efficiency of the Hybrid method. In case study 2, we increased the number of estimated parameters; the target parameters are k1, k2, and k3. Since we have to estimate the value of k2 in this case, the optimization was evaluated from the experimentally observed time courses of X1, X2, X3, and X5 as shown in Figure 12. In this case, we set the searching regions for the three parameters to be estimated as follows: $[\,0.0,\ 1.0\times10^{8}\,]$ for k1, $[\,0.0,\ 1.0\times10^{6}\,]$ for k2, and $[\,0.0,\ 15.0\,]$ for k3. We executed 15 trials by using the GA and the Hybrid method. The result is shown in Table 3.

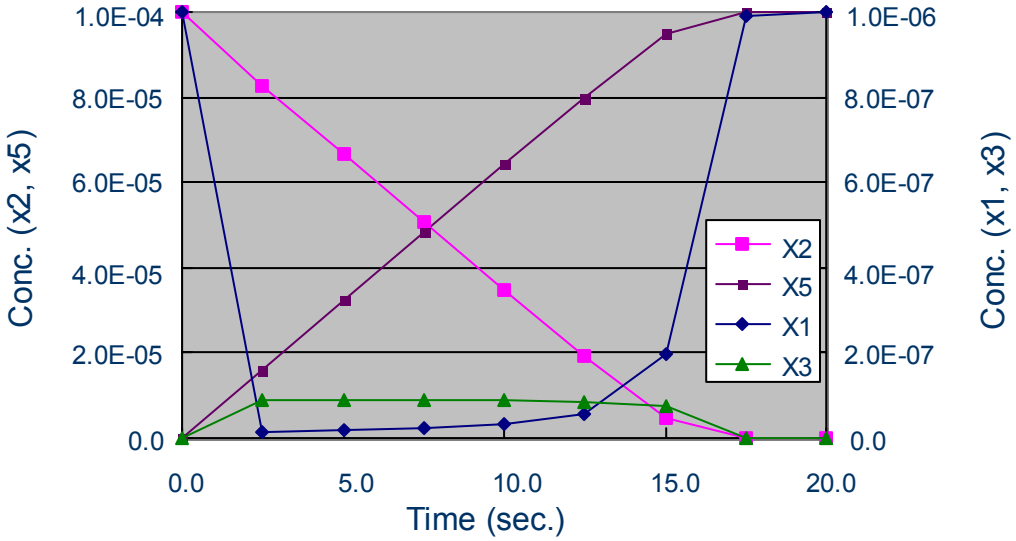


**Figure 12.** The experimentally observed time course data

**Table 3.** Optimization results for the GA and the Hybrid method within 15 trials (case study 2)

| | GA | Hybrid method (GA+MP) |
|---|---|---|
| No. of successes | 15/15 | 15/15 |
| No. of Evaluations | $1.91\times10^{4}\pm1.11\times10^{4}$ | $5.65\times10^{3}\pm4.53\times10^{3}$ |
| Average cpu time* (sec) | $2179.7\pm1220.7$ | $702.8\pm879.6$ |

*Ultra SPARC IIi 333MHz

Both methods succeeded in searching for optimal solutions of k1, k2, and k3; however, the average cpu time for optimization by the Hybrid method is about one third of that required by the GA.

## 6. Discussion

In this study we proposed a new optimization method, which incorporates the genetic algorithm into the modified Powell method. The genetic algorithm we adopted in this study belongs to the real-coded genetic algorithm (RCGA), which has attracted attention as a numerical optimization method for nonlinear systems. One of the crossover operators for RCGA called the unimodal normal distribution crossover (UNDX), which we adopted as shown in Figure 8, performed well in the optimization of various functions including multi-modal ones and benchmark functions with epistasis among the parameters [20]. The UNDX generates a new population lying on some ponds or along some valleys in order to focus the search on promising areas from the viewpoint of searching efficiency. Especially when the function has epistasis among its parameters, namely valleys that are not parallel to the coordinate axis, the UNDX can efficiently optimize it. When a genetic algorithm (GA) is applied to optimization problems, it is important that characteristics are preserved in designing coding/crossover and diversity maintained in designing generation alternation. Generation alternation models are independent of problems, while coding/crossover depends on problems. The simple GA is one of the well-known generation alternation models; however, the simple GA has two problems. One is early convergence in the first stage of searching and the other is evolutionary stagnation in the last stage. A new generation alternation model called minimal generation gap (MGG) was proposed to overcome the above problems [20][23]. The MGG

has all of the advantages of the conventional models as well as the ability to avoid early convergence and to suppress evolutionary stagnation. In this study, the GA (MGG+UNDX) became a more powerful searching method by the incorporation of the modified Powell method (MP), which can accelerate the dropping to the minimum point in quadratic form. Although the Hybrid method is less powerful than the MP at convergence speed when the initial guess of the estimated parameter is located very near to the optimal solution (see Tables 5 and 6), the Hybrid method has a great capability for discovering the optimal solution even when the initial guess of an estimated parameter is far from the optimal solution. Also in comparison with the GA (MGG+UNDX), the Hybrid method has an additional advantage in its convergence speed.

To summarize this study, the GA can find the solution within a comparatively large searching range, but it has very slow convergence, because the GA does not have an efficient local searching function. The MP can find the solution very quickly but only when optimization is started from very near the solution; the MP does not have the function of escaping from a local minimum. Thus the Hybrid method offers all of the advantages of both optimization techniques while offsetting their disadvantages; the proposed procedure (MGG+UNDX+MP) can not only help the algorithm to extricate the solutions from local minima but also seek out the global minimum with very fast convergence.

## 7. Conclusions

We have developed a parameter-fitting module for the web-based BEST-KIT. By using this module, the users can easily estimate unknown kinetic parameters based on the observed time-course data of reactants. This module adopted three optimization techniques, the modified Powell method, the genetic algorithm, and the Hybrid method, which we proposed. We carried out computational experiments to investigate the respective the features of these three optimization techniques. The MP can get the optimal solution very quickly only in the case where optimization starts from a point very near to the optimal solution. In contrast, the GA and the Hybrid method can find the optimal solution even when optimization starts far from the optimal solution. Especially, the Hybrid method showed superiority over the GA with respect to the speed of convergence.

This work was partially supported by a Grants-in-Aid for Scientific Research on Priority Areas, “Genome Information Sciences” (No.12208008) from the Ministry of Education, Culture, Sports, Science and Technology in Japan (MO).

## References


[1] K. Ichikawa, *Bioinformatics*, **17**(5), 483-484 (2001).
[2] Home Page of BioSpice, http://www.lbl.gov/~aparkin/.
[3] Home Page of DBSolve, http://homepage.ntlworld.com/igor.goryanin/.
[4] Home Page of E-Cell, http://www.e-cell.org/.
[5] Home Page of Genomic Object Net, http://GenomicObject.Net.
[6] Home Page of gepasi, http://www.gepasi.org.
[7] Home Page of JDesigner, http://www.cds.caltech.edu/~hsauro/JDesigner.htm.
[8] Home Page of MIST, http://cse.ogi.edu/DISC/projects/mist/.
[9] E. O. Voit, Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists (Cambridge University Press), (2000).
[10] Home Page of Virtual Cell, http://www.nrcam.uchc.edu/.
[11] Home Page of WinSCAMP, http://www.cds.caltech.edu/~hsauro/Scamp/scamp.htm.
[12] M. Okamoto, Y. Morita, D. Tominaga, N. Kinoshita, J-I. Ueno, Y. Miura, Y. Maki, Y. Eguchi, Pacific Symposium on Biocomputing '97 (PSB97), 304-315 (1997).
[13] M. Okamoto, K. Sakuraba, J. Yoshimura, K. Tanaka, J-I. Ueno, M. Mori, T. Shimonobou, T. Sekiguchi, *Chem-Bio Informatics Journal*, **2**, 1-17 (2002).
[14] Home Page of MassAction++ in BEST-KIT, http://www.best-kit.org/~bestkit/bestkit/MassAction.html.
[15] W. H. Press et al., Numerical Recipes (Cambridge University Press), (1992).
[16] M. Okamoto, T. Nonaka, S. Ochiai, D. Tominaga, *Applied Math. & Comput.*, **91**, 63-72 (1998).
[17] T. Ueda, N. Koga, I. Ono, M. Okamoto, Proc. of The Seventh Int. Symp. on Artificial Life and Robotics (AROB 7th '02), **1**, 290-293 (2002).
[18] C. W. Gear, Numerical initial value problems in ordinary differential equations, (1971).
[19] S. Hirano, *World wide Computing and Its Applications, Springer Lecture Notes in Computer Science*, **1274**, 29 (1997).
[20] H. Sato, I. Ono, S. Kobayashi, *J. of Japanese Society for Artificial Intelligence,* **12**(5), 734-744 (1997).
[21] I. Ono, S. Kobayashi, Proc. 7th ICGA, 246-253 (1997).
[22] C. T. Hsiao et al., *J. of Computational Physics*, **173**, 433-454 (2001).
[23] I. Ono, T. Nijo, N. Ono, Proc. of the Genetic and Evolutionary Computation Conf. (GECCO2000), 203-210 (2000).