Original Paper

Two-phase Search (TPS) Method: Nonbiased and High-speed Parameter Search for Dynamic Models of Biochemical Networks

Kazuhiro Maeda[†1] and Hiroyuki Kurata[†1]

Dynamic simulations are essential for understanding the mechanism of how biochemical networks generate robust properties to environmental stresses or genetic changes. However, typical dynamic modeling and analysis yield only local properties regarding a particular choice of plausible values of kinetic parameters, because it is hard to measure the exact values *in vivo*. Global and firm analyses are needed that consider how the changes in parameter values affect the results. A typical solution is to systematically analyze the dynamic behaviors in large parameter space by searching all plausible parameter values without any biases. However, a random search needs an enormous number of trials to obtain such parameter values. Ordinary evolutionary searches swiftly obtain plausible parameters but the searches are biased. To overcome these problems, we propose the two-phase search method that consists of a random search and an evolutionary search to effectively explore all possible solution vectors of kinetic parameters satisfying the target dynamics. We demonstrate that the proposed method enables a nonbiased and high-speed parameter search for dynamic models of biochemical networks through its applications to several benchmark functions and to the *E. coli* heat shock response model.

1. Introduction

Computer simulations enable one to capture the dynamic behavior of complex biochemical networks. In principle, both molecular network architecture and the values of kinetic parameters determine the dynamic behavior of systems. In biology, molecular network structures are being built, but it is still hard to measure the accurate values of kinetic parameters *in vivo* due to experimental complexity. The values of kinetic parameters vary with time and environment and the measured values *in vitro* are often different from those *in vivo*. In many studies, a particular set of local kinetic parameters has been determined for convenience so that dynamic models reproduce target data. Thus, the simulated results often depend on the values of kinetic parameters, or reflect only a local view of the system. There have been only a few simulation methods that extensively investigated how a systematic change in the parameter values alters the prediction of dynamic behaviors [1)–5)].

To compare some performances of alternative mathematical models, Alves, et al. [1)–3)] statistically searched their parameter values so that they make the other dynamic properties the same. Stelling, et al. [4)] studied dynamic properties linked to network structure in the *per-tim* feedback loop model by systematically investigating the two dimensional parameter space and suggested some influential process determining the oscillator features. These previously presented random or systematic searches are a great step for approaching to global analysis, but they restricted the search space of parameters or the size of models due to calculation complexity.

In mathematical analysis for robustness (MAR), we proposed the evolutionary method that explores possible solution vectors of kinetic parameters satisfying the target dynamics and extracts the global mechanism of how changes in kinetic parameters alter the robustness of a circadian oscillator [5)]. The employed evolutionary method tries to search all possible solutions in a greatly extended space, but it still remains to be improved or to be verified in terms of search performance. The critical requirement for obtaining global results is the nonbiased search in large parameter space. One must prevent the search from being intensively performed on local regions.

To overcome the problems of nonbiased search for a large parameter space, we developed a novel search algorithm, the two-phase search (TPS) method that smoothly combines a random search with an evolutionary algorithm to achieve both nonbiased and high-speed searches. To demonstrate the feasibility of this method, we apply it to benchmark problems and reveal its search performance in terms of the calculation efficiency and solution distributions. Finally, the effectiveness of the TPS method is verified through the parameter search of the *E. coli* heat shock response model [6),7)].

[†1] Department of Bioscience and Bioinformatics, Kyushu Institute of Technology

2. Methods

2.1 Dynamic Model to be Optimized

Generally a dynamic model for biochemical networks is formulated by differential-algebraic equations (DAEs):

$$\mathbf{0} = \mathbf{G}(\mathbf{x}, \mathbf{y}, \mathbf{P}) \tag{1}$$

$$\dot{\mathbf{y}} = \mathbf{F}(t, \mathbf{x}, \mathbf{y}, \mathbf{P}) \tag{2}$$

where t is time, and $\mathbf{P}$ is the kinetic parameter vector. The algebraic equation (1) shows the binding reaction for complex formation (e.g., Eqs. (A1)–(A19) in Table 4), while the differential equation (2) indicates the synthesis, conversion and degradation of molecules (e.g., Eq. (12) and Eqs. (A20)–(A28) in Table 5). In Eq. (1), $\mathbf{x}$ is the dependent variable vector that indicates the concentration of free molecules and binding complexes (e.g., $[\sigma^{70}], \ldots, [\sigma^{32}\!:\!DnaK\!:\!FtsH]$ in Table 6), while, $\mathbf{y}$ is independent variable vector that consists of the total concentration vector of each molecule (e.g., $[Protein]$ in Eq. (12) and $[P_{fold}], \ldots, [P_{unfold}]_{total}$ in Table 7). In Eq. (2), $\mathbf{x}$ is set to be the independent variable vector and $\mathbf{y}$ is the time-dependent variable vector.

Numerical optimization for a dynamic model is used to estimate the values of kinetic parameters so that the model reproduces the behaviors of the existing experimental data [8)–10)]. A certain fitness function is necessary to characterize the degree to which the model reproduces the target experimental behaviors.

2.2 Two-phase Search (TPS) Method

Since biological data contain different types of errors, it is meaningless to seek the global minimum for the fitness function defined for a given dynamic model. The objective in this study is not to find such a global minimum, but to explore all possible plausible solutions of kinetic parameter vectors that produce the target dynamics.

The TPS method is proposed that combines a random search with a search by genetic algorithms (GAs), as shown in **Fig. 1**. Details of its computation algorithm are shown in Appendix Fig. 10. First, the random search explores a large parameter space without any biases to find a coarse solution showing a good fitness value. In this phase, it is not necessary to find any solutions providing lowest fitness values. The resultant coarse solution is employed to generate the initial populations for the subsequent search by GAs. Second, after the initial population is created around the coarse solution vectors, use of GAs intensively searches all plausible solution vectors that show a low fitness value or provide the target features. This two-phase search is iterated to obtain the sufficient number of the plausible solutions (Fig. 1 A). The i-th resultant solution vector of kinetic parameters $\mathbf{P}_i$ is given by:

$$\mathbf{P}_i = (p_{i,1}, p_{i,2}, \ldots, p_{i,N}), \tag{3}$$

where $p_{i,j}$ is the value of the j-th parameter of the i-th solution vector and N is the number of search parameters.

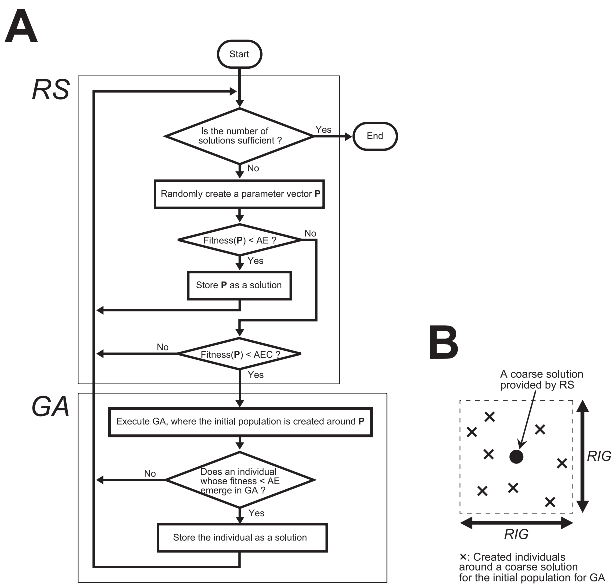

A
RS
GA
Start
Is the number of solutions sufficient ?
Yes
End
No
Randomly create a parameter vector P
Fitness(P) < AE ?
No
Yes
Store P as a solution
No
Fitness(P) < AEC ?
Yes
Execute GA, where the initial population is created around P
No
Does an individual whose fitness < AE emerge in GA ?
Yes
Store the individual as a solution
B
A coarse solution provided by RS
RIG
RIG
×: Created individuals around a coarse solution for the initial population for GA

Fig. 1 Schematic diagrams of the TPS method. A: A flow chart for TPS that consists of a random search (the first phase) and a search by GAs (the second phase). B: How to create the initial populations for the second phase search by GAs. AE indicates the allowable error, AEC the allowable error for the coarse solution, and RIG the region of the initial population for the search by GAs.

The TPS method has two critical control parameters: the allowable error for the coarse solution (AEC) obtained by a random search in the first phase and the region of the initial population for the search by GAs (RIG) in the second phase. The end condition of the first phase search is provided by:

$$\text{Fitness}(\mathbf{P}) < AEC. \tag{4}$$

The end condition of the second phase search for a final solution is given by:

$$\text{Fitness}(\mathbf{P}) < AE, \tag{5}$$

where AE is the allowable error of the plausible solution ($AE < AEC$). The initial population for the GA search is randomly generated within the hypercube whose edge length is RIG and whose centroid is set to the coarse solution in the first phase (Fig. 1 B). We also define the relative value of AEC to AE and the relative value of RIG to the search region of each parameter, which are respectively provided by:

$$RAEC = AEC/AE, \tag{6}$$

and

$$RRIG = RIG/(p^U - p^L), \tag{7}$$

where p^U is the upper bound for the value of kinetic parameter and p^L the lower bound.

2.3 Characterization of the Solutions

Three standards are defined to characterize the search results, the number of evaluations necessary for obtaining a given number of the final solutions (EVA), the centroid vector (CRV) and standard deviations vector (SDV) for the solution distributions. The CRV and SDV characterize the statistical features for the solution distributions, defined by:

$$\mathbf{CRV} = (c_1, c_2, \ldots, c_N), \tag{8}$$

$$c_j = \frac{1}{M}\sum_{i=1}^{M} p_{i,j}, \tag{9}$$

$$\mathbf{SDV} = (sd_1, sd_2, \ldots, sd_N), \tag{10}$$

$$sd_j = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(p_{i,j} - c_j)^2}, \tag{11}$$

where M is the number of solutions. A small value of EVA indicates an efficient or high-speed search. A search can be regarded nonbiased, when two standards of CRV and SDV are close to those in a random search. To obtain detailed information for a solution distribution, the frequency distributions of the solutions with respect to each search parameter are presented.

Table 1 Benchmark functions. AE is the allowable error.

	Objective function ($n = 2$)	Search region	AE
Sphere	$f(\mathbf{P}) = \sum_{i=1}^{n} p_i^2$	$-5.12 < p_i < 5.12$	0.333
Rosenbrock	$f(\mathbf{P}) = \sum_{i=1}^{n-1}(100(p_{i+1} - p_i^2)^2 + (p_i - 1)^2)$	$-2.048 < p_i < 2.048$	0.676
Rastrigin	$f(\mathbf{P}) = 10n + \sum_{i=1}^{n}(p_i^2 - 10\cos(2\pi p_i))$	$-5.12 < p_i < 5.12$	6.00
Schwefel	$f(\mathbf{P}) = 418.9828873n + \sum_{i=1}^{n} p_i \sin\sqrt{\lvert p_i \rvert}$	$-512 < p_i < 512$	215
ANFM	$f(\mathbf{P}) = \left\lvert \frac{-p_2+\sqrt{p_2^2+4p_1p_2}}{2} - 1 \right\rvert$	$0.02 < p_1 < 200$ $0.01 < p_2 < 100$	10^{-4}

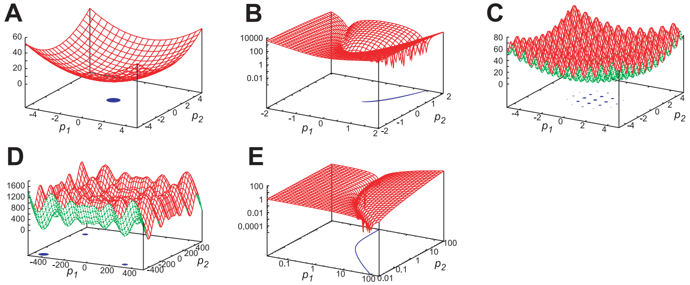


Fig. 2 Landscapes of the benchmark functions and their solution spaces. (A) Sphere, (B) Rosenbrock, (C) Rastrigin, (D) Schwefel, (E) ANFM. Blue regions are the solution spaces.

2.4 Benchmark Functions

In this paper, the parameter search of a dynamic model of biochemical networks indicates finding all possible values of the parameters that satisfy specific objective functions. To investigate the search capability of the TPS method, we employ several generally-used benchmark functions as shown in **Table 1**. The landscapes of the benchmark functions and the solution spaces are shown in **Fig. 2**, where p_i is the i-th kinetic parameter. A Sphere function is unimodal and its variables are separable. The Rosenbrock function is unimodal, while its variables are non-separable. The Rastrigin function and the Schwefel function are multimodal and their variables are separable. In the former, the solutions distribute around the center of search region; in the latter, they are distributed far from the center. The experiments were performed on the two dimensional space. Their allowable errors (AEs) were determined so that 1% of the randomly generated sets of the parameter values provide a fitness value of less than the AEs (Table 1).

Next, we add the benchmark function of a typical biological model with an autogenous negative feedback model (ANFM):

$$\frac{d[Protein]}{dt} = k\frac{K}{K + [Protein]} - [Protein] \tag{12}$$

where $[Protein]$ is the protein concentration, k the synthesis rate constant and K the binding constant. The protein suppresses its own synthesis. The steady-state concentration $[Protein]_{ss}$ is analytically provided by:

$$[Protein]_{ss} = \frac{-K + \sqrt{K^2 + 4kK}}{2} \tag{13}$$

The target behavior is set to steady-state concentration $[Protein]_{ss}$ of 1. Therefore, the ANFM benchmark function is defined by:

$$f(k, K) = \left| \frac{-K + \sqrt{K^2 + 4kK}}{2} - 1 \right| \tag{14}$$

Two kinetic parameters (k, K) are explored in logarithmic space, where the basis parameter vector is set to $(k, K) = (2, 1)$ that provides one to the steady state concentration of $[Protein]_{ss}$ and AE is set to 10^{-4}. In our parameter searches, the value of each basis parameter was 10^{-2} to 10^{2}-fold varied in logarithmic space. k is provided as p_1 and K is as p_2.

2.5 Dynamic Model of the *E. coli* Heat Shock Response

A biochemical network map of the *E. coli* heat shock response is shown in **Fig. 3** A[6),7)]. Heat shock denatures or unfolds proteins, compromising cellular function. To counter heat shock, heat-shock proteins (hsps), chaperones and protease, are produced to refold the denatured proteins to their native state and to degrade them. The regulation of the synthesis, degradation, and activity of the σ^{32} factor plays a major role in heat shock. The time course of the σ^{32} concentration is shown in Fig. 3 B. The level of σ^{32} increases rapidly at heat shock, achieves a sharp peak, and then reduces to a new steady state.

The mathematical model was constructed by the CADLIVE GUI network constructor[11),12)] and the CADLIVE dynamic simulator[13)]. Appendix A.1 indicates the equations, initial conditions, lists of molecular concentration variables and kinetic parameters. The simulation is performed for 100 minutes. Heat shock occurs at 60 min and is implemented through an increase in the rate constant for protein denature (Table 8).

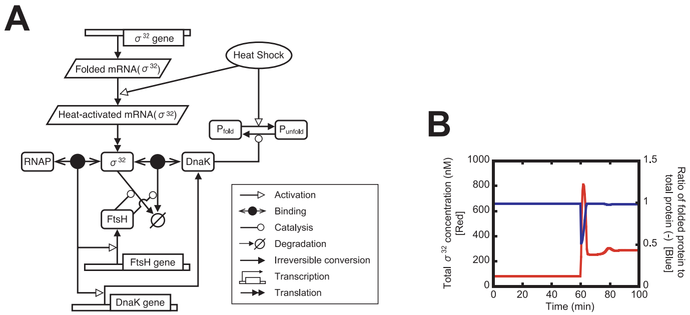


Fig. 3 An *E. coli* heat shock response model. A: A network map. The notation of CADLIVE[11)–13)] is used for simplifying the diagram. B: Time evolution of the σ^{32} concentration (red, $[\sigma^{32}]_{total}$ in Table 7) and the ratio of the folded protein concentration to the total protein concentration (blue, $[P_{fold}]/([P_{fold}]+[P_{unfold}]_{total})$ in Table 7). Heat shock occurs at 60 min and is implemented through an increase in the rate constant for protein denature.

To capture the typical dynamic features of the heat shock response, the fitness function is defined by:

$$Fitness = 2 - ratio1 - ratio2 + penalty1 + penalty2, \tag{15}$$

where $ratio1$ is the ratio of the folded protein concentration to the total protein concentration before heat shock and $ratio2$ that after heat shock (Fig. 3 B). $penalty1$ and $penalty2$ impose penalty on the fitness when the level of σ^{32} does not behave as mentioned above. This fitness function mainly targets the protein refolding ability. $ratio1$ and $ratio2$ are presented by:

$$ratio1 = \frac{[P_{fold}]_{\mathrm{at59min}}}{[P_{fold}]_{\mathrm{at59min}} + [P_{unfold}]_{total,\mathrm{at59min}}}, \tag{16}$$

$$ratio2 = \frac{[P_{fold}]_{\mathrm{at100min}}}{[P_{fold}]_{\mathrm{at100min}} + [P_{unfold}]_{total,\mathrm{at100min}}}. \tag{17}$$

As $ratio1$ and $ratio2$ become close to 1, the fitness approaches to zero, which is the best value. $penalty1$ and $penalty2$ are presented by:

$$penalty1 = 1 - \frac{[\sigma^{32}]_{total,\text{at100min}} - [\sigma^{32}]_{total,\text{at59min}}}{[\sigma^{32}]_{total,\text{at59min}}}, \tag{18}$$

$$penalty2 = 1 - \frac{\max\left([\sigma^{32}]_{total}\right) - [\sigma^{32}]_{total,\text{at100min}}}{[\sigma^{32}]_{total,\text{at100min}}}, \tag{19}$$

where $\max\left([\sigma^{32}]_{total}\right)$ is the maximum value for the total σ^{32} concentration in the simulation. If *penalty*1 is less than zero, *penalty*1 is set to zero. *penalty*2 is set in the same manner. The kinetic parameter solutions are estimated that satisfy the condition: $Fitness < AE = 2 \times 10^{-2}$, indicating that the means between the ratio of the folded proteins to the total proteins before heat shock and that after heat shock are 0.99.

A very small value of AE is not practical, because experimental data generally contain considerable errors. Adversely, a large value of AE cannot reproduce the target behaviors of the heat shock model. It is still hard to mathematically or precisely estimate a best value of AE by considering how the biological errors are generated, because the errors are not quantified. In this model, we choose an appropriate value of AE that is small enough for the model to indicate a biologically successful refolding of heat-denatured proteins.

3. Results and Discussion

3.1 TPS Application to Benchmark Functions

In order to demonstrate the feasibility of TPS, we applied it to the benchmark functions (Table 1) to search all possible solution vectors that give a smaller fitness value than a defined AE and investigated the search performance of EVA, CRV, and SDV. Three types of the searches: a random search (RS), a search by GAs (SGA), and TPS, were iterated until the number of the solutions reached to 10,000. RS and SGA were used as controls. In SGA, one search was stopped when a solution was obtained or the search reached to the maximum generation. For the next search, the initial population was newly generated. The unimodal normal distribution crossover (UNDX)[14] and minimal generation gap (MGG)[15] were employed as crossover and selection, respectively. Mutation was not employed. The maximum generation was 100, the population size was 10, and the number of

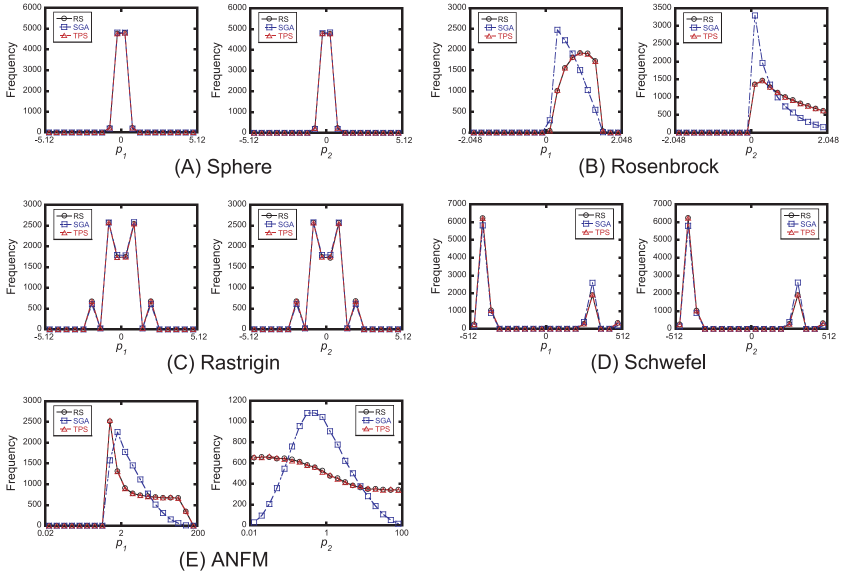
Table 2 Characterization of the search performance of RS, SGA, and TPS. The presented data by TPS are best cases in our experiments. For Schwefel, TPS did not simultaneously achieve a lower EVA than RS and the same CRV and SDV as RS. If $|c_i| < 10^{-2}$, c_i is set to zero.

			CRV		SDV	
		EVA	c_1	c_2	sd_1	sd_2
Sphere	RS	1.00×10^6	0.00	0.00	2.89×10^{-1}	2.89×10^{-1}
	SGA	5.67×10^5	0.00	0.00	2.76×10^{-1}	2.76×10^{-1}
	TPS ($RAEC = 1.01, RRIG = 0.4$)	9.93×10^5	0.00	0.00	2.88×10^{-1}	2.88×10^{-1}
Rosenbrock	RS	9.64×10^5	8.68×10^{-1}	8.62×10^{-1}	3.30×10^{-1}	5.69×10^{-1}
	SGA	9.71×10^5	6.57×10^{-1}	5.33×10^{-1}	3.20×10^{-1}	4.78×10^{-1}
	TPS ($RAEC = 1.02, RRIG = 0.4$)	9.54×10^5	8.67×10^{-1}	8.60×10^{-1}	3.31×10^{-1}	5.70×10^{-1}
Rastrigin	RS	1.07×10^6	0.00	0.00	1.03	1.03
	SGA	6.07×10^5	0.00	0.00	1.00	1.00
	TPS ($RAEC = 1.01, RRIG = 0.4$)	1.06×10^6	0.00	0.00	1.03	1.02
Schwefel	RS	9.12×10^5	-2.33×10^2	-2.34×10^2	3.27×10^2	3.26×10^2
	SGA	3.25×10^6	-1.89×10^2	-1.88×10^2	3.42×10^2	3.42×10^2
	TPS ($RAEC = 1.01, RRIG = 0.4$)	9.06×10^5	-2.31×10^2	-2.33×10^2	3.28×10^2	3.27×10^2
ANFM	RS	3.07×10^8	$10^{0.717}$	$10^{-0.283}$	$10^{0.636}$	$10^{1.12}$
	SGA	5.61×10^6	$10^{0.516}$	$10^{-0.176}$	$10^{0.408}$	$10^{0.716}$
	TPS ($RAEC = 100, RRIG = 0.002$)	4.00×10^6	$10^{0.718}$	$10^{-0.282}$	$10^{0.637}$	$10^{1.12}$

the children generated by the crossover per selection was 10. In TPS, the second phase employed the same setting as SGA. Since the TPS method aims at both high-speed and nonbiased searches, TPS is expected to achieve a smaller EVA than RS and to provide the same CRV and SDV as RS.

The search performance of TPS is summarized in **Table 2**. The performance values of EVA, CRV and SDV are the average values for 10-time trials. We compared these values between TPS and the controls (RS and SGA) (Appendix A.3). In terms of EVAs, TPS was a fast algorithm in ANFM, while it was not so fast for the other benchmark functions. The EVA value by the well-designed TPS was 1.3% of that of RS for ANFM, while the EVA values by TPS were roughly 99% of those of RS in Sphere, Rosenbrock, and Rastrigin. These 1% reductions were statistically significant (Appendix A.3), although they are no use for practical purpose. For Schwefel, the EVA by TPS was not significantly reduced compared with that by RS. In terms of solution distributions, the CRV and SDV by TPS were statistically the same as those of RS, suggesting that TPS presents nonbiased search. On the other hand, the CRV and SDV by SGA were significantly different from those of RS for all benchmarks, indicating that the solutions by SGA are biased.

To further investigate the solution distributions by TPS in comparison with RS and SGA, we plotted the frequency distribution of the solutions with respect to each search parameter in **Fig. 4**. The frequency distributions by TPS are the same as those by RS for all benchmark functions, as expected from Table 2, where RS is assumed as a non-biased search. On the other hand, the frequency distribution by SGA is clearly different from those by RS for Rosenbrock and ANFM, indicating that the solutions by SGA are biased. Note that the solution distributions by RS and SGA are statistically different in the other benchmarks of Sphere, Rastrigin, and Schwefel. Rosenbrock and ANFM have the typical landscapes with a thin and long solution space (Fig. 2), which may make the search space by SGA notably biased. The differences in the CRV and SDV in Table 2 are confirmed as the differences in these solution distributions. In summary, when AEC and RIG were well-designed, TPS provided the same solution distributions as RS in all the benchmarks, indicating that TPS is a non-biased search. For ANFM, TPS is demonstrated to be a efficient and non-biased search algorithm.



(A) Sphere

(B) Rosenbrock

(C) Rastrigin

(D) Schwefel

(E) ANFM

Fig. 4 Frequency distributions for each solution parameter searched by RS, SGA and TPS in the benchmark functions.

3.2 Characterization of the Two Control Parameters in Benchmark Functions

TPS has two control parameters critically responsible for search performance. Effects of AEC and RIG on the search performances of EVA, CRV and SDV were investigated, as shown in **Fig. 5**. In a broad range of RIG, EVA decreased with the increase in AEC and then increased for the benchmarks except Schwefel, indicating there is an optimal or minimum value for EVA. This tendency is clearly observed in ANFM, while it is judged from the numerical data in the other functions. A change in RIG affected the performance of TPS at a large value of AEC, while it did not at a small AEC. At an adequately small value of AEC ($RAEC \approx 1$), the EVA, CRV and SDV were almost the same as those by RS, indicating that TPS performs in the same manner of RS at a small AEC value. At large values of AEC and RIG, the EVA, CRV and SDV were almost

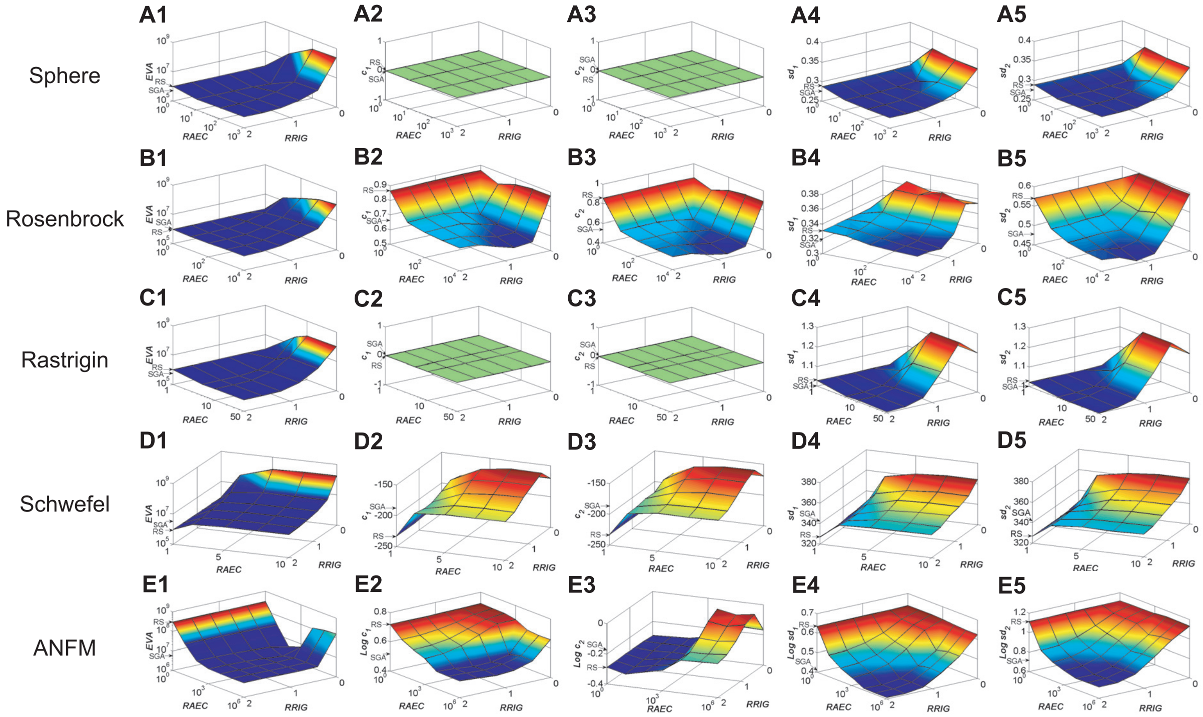
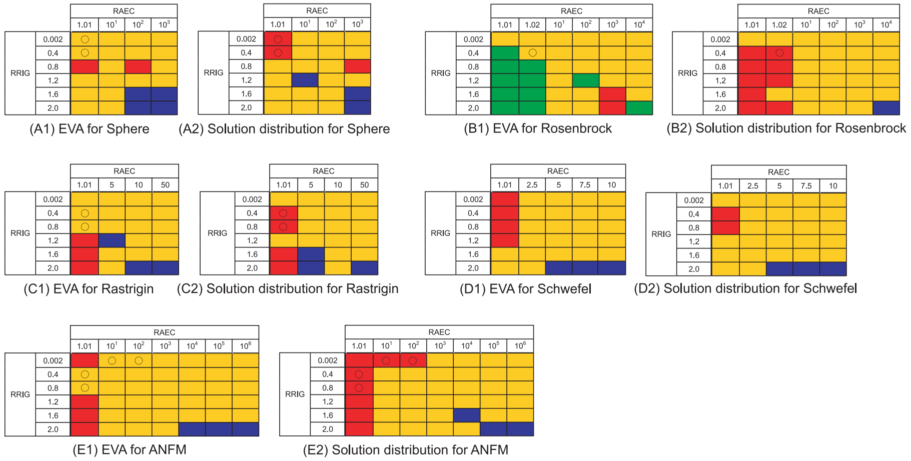


Fig. 5 Characterization of the search performance by the TPS method. The EVAs, CRVs and SDVs are calculated for different values of the control parameters (AEC and RIG). (A) Sphere, (B) Rosenbrock, (C) Rastrigin, (D) Schwefel, and (E) ANFM. If $|c_i| < 10^{-2}$, c_i is set to zero.

the same as those by SGA, indicating TPS with adequately large AEC and RIG behaves like SGA. The combination of a large AEC and a small RIG caused massive EVA and the eccentric CRV and SDV that are far away from those of both RS and SGA. This combination forced the second phase search of TPS to perform in a narrow region surrounding unreasonable coarse solution.

Here, we statistically investigated the feasible regions in which TPS achieved a lower EVA than RS and the same solution distribution as RS, as shown in **Fig. 6**. For the typically used benchmarks, TPS achieved the same solution distribution as RS only at a small value of AEC ($RAEC \approx 1.01$). Since TPS with a small AEC became close to RS, TPS did not effectively reduce EVA. In ANFM, by contrast, TPS achieved the same solution distribution as RS at a wide range of AEC ($1.01 \leq RAEC \leq 10^2$). Since a large value of AEC readily finds a coarse solution in the first phase search (RS) and the solution is quickly obtained in the second search by GAs, TPS achieves a much smaller EVA than RS.

This indicates that TPS would effectively solve the ANFM benchmark function, a typical biological equation, rather than the other typical benchmark functions. This difference in the search performance between the ANFM model and the others is probably caused by the landscape of the function. The landscape of the biological equation (ANFM) seems very different from that of the Sphere, Rosenbrock, Rastrigin and Schwefel benchmark functions (Fig. 2). The biological



(A1) EVA for Sphere (A2) Solution distribution for Sphere (B1) EVA for Rosenbrock (B2) Solution distribution for Rosenbrock

(C1) EVA for Rastrigin (C2) Solution distribution for Rastrigin (D1) EVA for Schwefel (D2) Solution distribution for Schwefel

(E1) EVA for ANFM (E2) Solution distribution for ANFM

Fig. 6 Characterization of the EVA and solution distributions by the TPS method. The characterization was performed using the Wilcoxon rank sum test and the goodness-of-fit test. In (A1-E1), the EVA by TPS is statistically the same as that by RS (red), the same as that by SGA (blue), or neither close to RS nor SGA (yellow). The EVAs by TPS are the same as those both by RS and SGA (green), i.e., the EVAs of TPS, RS, and SGA are all the same. In (A2-E2), the solution distribution by TPS is the same as that by RS (red), the same as that by SGA (blue), or neither close to RS nor SGA (yellow). (○) indicates that TPS simultaneously provides a smaller value of EVA than RS and the same solution distribution as RS, indicating that TPS is a best choice for an efficient and non-biased search.

benchmark has a long and thin solution space, while the other typical benchmark functions have scattered or locally condensed spaces.

For all benchmarks, there were some regions in which TPS achieved statistically the same EVA and solution distribution as RS (Fig. 6). These regions where TPS shows the same performance (EVA, CRV and SDV) as RS are distributed around $RAEC \approx 1.01$ (red and green marked spaces). On the other hand, the regions in which TPS achieved statistically the same EVA and solution distribution as SGA are distributed around a RRIG of 2.0 and the adequately large AEC that approximately equals the maximum value for each benchmark (blue and green marked spaces).

3.3 Summary of the TPS Performance in Benchmark Functions

The search performance of TPS is summarized in **Fig. 7**, although the size and shape of RS-like, SGA-like and the feasible region for TPS are dependent on benchmarks. The TPS method with a small AEC tends to approach to RS. In this case, the value of EVA is large, while nonbiased search is performed. When both the values of AEC and RIG are adequately large, the performance of TPS becomes approximately close to that of SGA. Thus, TPS achieves a small EVA, but the solution distributions are different from those of RS. The combination of a large AEC and a small RIG moves the TPS method far away from both RS and SGA. This search spends an enormous computational time, which is sometimes larger than that of RS and provides a different solution distribution from those of both RS and SGA.

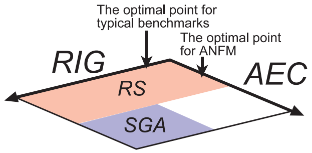

The optimal point for typical benchmarks
The optimal point for ANFM
RIG
AEC
RS
SGA

Fig. 7 Recommendation values for the two control parameters of TPS.

If AEC and RIG are set to an appropriate (optimal) value, the TPS method achieves high-speed and nonbiased search (Fig. 7). Note that an appropriate value of RAEC must be very close to 1 for Sphere, Rosenbrock, and Rastrigin, while it can be set to a large value (> 100) for ANFM. In the former three functions, since an appropriate value of RAEC is close to 1, i.e., TPS is close to RS, EVA is not reduced greatly. In ANFM, however, since an appropriate value of AEC can be set to a large value, TPS performs a high-speed search without any bias. TPS is demonstrated to be suitable for the ANFM function, a typical biological model, although it is not practically effective in generally used benchmark functions.

It is still hard to automatically determine the optimal values of AEC and RIG for different functions, because they depend on the landscape of the functions (Fig. 2). To determine the value of AEC and RIG, we recommend that one performs a random search in small size to capture a rough landscape of the function. Next, estimate AEC based on the landscape. If AEC needs to be very small, TPS becomes close to RS, i.e., TPS is not effective. If one finds an appropriate or large value of AEC, one investigates the effect of RIG on the function in small size. For example, in ANFM an AEC value of 0.01 ($RAEC =$

Table 3 Characterization of the EVA and calculation time by RS, SGA and TPS in the heat shock response model. The searches were performed on a single core of Intel DualCore Xeon 5160 3 GHz × 2 with 32 GB RAM.

	EVA	Execution time (hour)
RS	2.45×10^5	261
SGA	1.75×10^5	202
TPS	9.22×10^4	159

100) is reasonable and a value of RRIG is 0.002.

3.4 Application to the Heat Shock Response

The heat shock response model has 43 kinetic parameters. Out of 43 parameters, 11 binding constants (Kb_1-Kb_{11} in Table 8) are searched so that the fitness value becomes less than AE. In the searches, the basis values for the search parameters are set to 10^7 and 10^{-4} to 10^4-fold varied in logarithmic space, while the other 32 parameters are fixed. RS and SGA are employed as controls. The searches are iterated until the number of solutions reaches to 1,000.

SGA and the second phase in the TPS method employed UNDX and MGG. The control parameters for SGA were as follows: the maximum generation was 100, the population size 10, and the number of the children generated by the crossover per selection 10. The control parameters for TPS were designed based on the results for benchmark experiments. After a random search was performed to find a coarse solution with $Fitness < AEC = 1$ ($RAEC = 50$), the GA search was performed as follows: the maximum generation was 20, the population size 10, the number of the children generated by the crossover per selection 10, and RIG 4 ($RRIG = 0.5$). A RIG value of 4 indicates that the values of the coarse solutions are 10^{-2} to 10^2-fold varied to generate the initial population for the second phase.

The EVA of TPS is summarized in **Table 3**. The TPS method reduced the EVA value to 38% of that of RS and 53% of SGA, indicating that TPS is faster than RS and SGA. In real time, TPS reduced the calculation time to 61% of that of RS and 79% of SGA. The statistical properties of the solution spaces searched by TPS are shown in **Fig. 8**, where the CRVs and SDVs were calculated in logarithmic space. The CRVs were approximately the same for three searches (Fig. 8 A). The SDVs of SGA were smaller than those of RS (Fig. 8 B), indicating that SGA was biased, while the SDV by the TPS method was approximately the same as that of RS. This suggests that TPS presents a non-biases search in the same manner as RS. To further investigate the solution distribution, the frequency distribution with respect to each search parameter was shown in **Fig. 9**. TPS presents relatively the same distribution as RS, while SGA does not. This supports that the solution distribution by TPS is non-biased. Those of RS and TPS are not exactly the same, probably because the number of the solutions is rather small in terms of the extensive 11-dimensional search space.

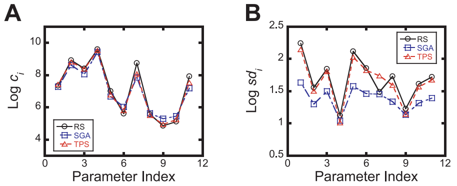


Fig. 8 Statistical properties of the solution spaces searched by RS, SGA and TPS in the heat shock response model. (A) Centroid vector (CRV), (B) standard deviations vector (SDV).

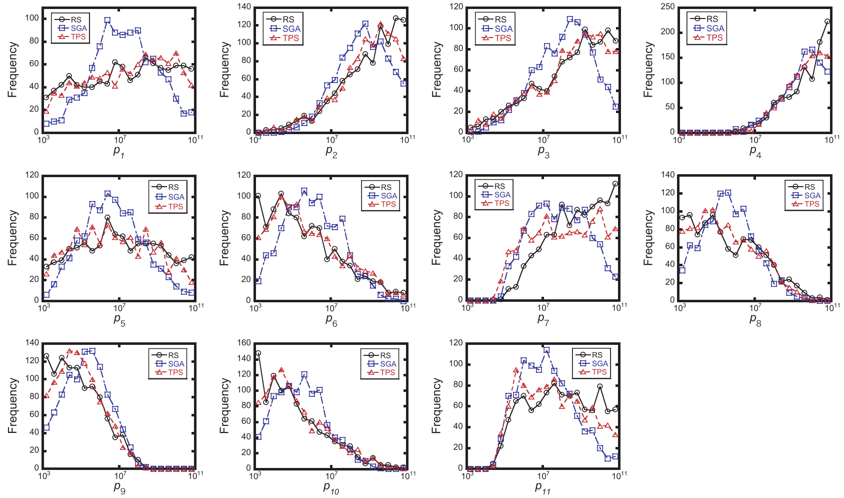


Fig. 9 Frequency distributions for each solution parameter in the heat shock response model.

The heat shock response model and the ANFM benchmark function are built based on biochemical kinetics. The search performances (calculation efficiency and solution distributions) for both the biological models are much better than those of the typical benchmark functions of Sphere, Rosenbrock, Rastrigin, and Schwefel. The TPS method is suggested to take advantage in the biochemical problems built based on molecular kinetics.

4. Conclusion

We propose the TPS method that consists of a random search and an evolutionary search to effectively explore all possible solution vectors of kinetic parameters satisfying the target dynamics, which greatly enhances the search efficiency without any biases in biological problems. The proposed method enables one to approach to global and firm analyses that consider how the changes in parameter values affect the results. We investigated the effects of two critical control parameters, AEC and RIG, of the TPS method on search performance. When the appropriate values of AEC and RIG are selected, which depends on the landscape of target functions, TPS achieves both high-speed and nonbiased searches. The TPS method does not show so high performance for typically-employed benchmark functions, but provides a great advantage in biochemical models.

Acknowledgments This work was supported by KAKENHI (Grant-in-Aid for Scientific Research) on Priority Areas "Systems Genomics" from the Ministry of Education, Culture, Sports, Science and Technology of Japan and partially by KAKENHI (Grant-in-Aid for Scientific Research (B), 2006, 18300098).

References

1) Alves, R. and Savageau, M.A.: Systemic properties of ensembles of metabolic networks: Application of graphical and statistical method to simple unbranched pathways, *Bioinformatics*, Vol.16, No.6, pp.534–547 (2000).
2) Alves, R. and Savageau, M.A.: Comparing systemic properties of ensembles of biological networks by graphical and statistical methods, *Bioinformatics*, Vol.16, No.6, pp.527–533 (2000).
3) Alves, R. and Savageau, M.A.: Extending the method of mathematically controlled comparison to include numerical comparisons, *Bioinformatics*, Vol.16, No.9, pp.786–798 (2000).
4) Stelling, J., Gilles, E.D. and Doyle, F.J.3rd: Robustness properties of circadian clock architectures, *PNAS*, Vol.101, No.36, pp.13210–13215 (2004).
5) Kurata, H., Tanaka, T. and Ohnishi, F.: Mathematical identification of critical reactions in the interlocked feedback model, *PLoS ONE*, Vol.2, No.10, p.e1103 (2007).
6) El-Samad, H., Kurata, H., Doyle, J.C., Gross, C.A. and Khammash, M.: Surviving heat shock: Control strategies for robustness and performance, *PNAS*, Vol.102, No.8, pp.2736–2741 (2005).
7) Kurata, H., El-Samad, H., Iwasaki, R., Ohtake, H., Doyle, J.C., Grigorova, I., Gross, C.A. and Khammash, M.: Module-based analysis of robustness tradeoffs in the heat shock response system, *PLoS Computational Biology*, Vol.2, No.7, p.e59 (2006).
8) Banga, J.R.: Optimization in computational systems biology, *BMC Systems Biology*, Vol.2, No.1, p.47 (2008).
9) Kimura, S., Ide, K., Kashihara, A., Kano, M., Hatakeyama, M., Masui, R., Nakagawa, N., Yokoyama, S., Kuramitsu, S. and Konagaya, A.: Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm, *Bioinformatics*, Vol.21, No.7, pp.1154–1163 (2005).
10) Mendes, P. and Kell, D.B.: Non-linear optimization of biochemical pathways: Applications to metabolic engineering and parameter estimation, *Bioinformatics*, Vol.14, No.10, pp.869–883 (1998).
11) Kurata, H., Inoue, K., Maeda, K., Masaki, K., Shimokawa, Y. and Zhao, Q.: Extended CADLIVE: A novel graphical notation for design of biochemical network maps and computational pathway analysis, *Nucleic Acids Research*, Vol.35, No.20, p.e134 (2007).
12) Kurata, H., Matoba, N. and Shimizu, N.: CADLIVE for constructing a large-scale biochemical network based on a simulation-directed notation and its application to yeast cell cycle, *Nucleic Acids Research*, Vol.31, No.14, pp.4071–4084 (2003).
13) Kurata, H., Masaki, K., Sumida, Y. and Iwasaki, R.: CADLIVE dynamic simulator: Direct link of biochemical networks to dynamic models, *Genome Research*, Vol.15, No.4, pp.590–600 (2005).
14) Ono, I. and Kobayashi, S.: A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover, *Proc. 7th Int Conf on Genetic Algorithms*, pp.246–253 (1997).
15) Satoh, H., Yamamura, M. and Kobayashi, S.: Minimal generation gap model for GAs considering both exploration and exploitation, *Proc. 4th Int Conf on Soft Computing*, pp.494–497 (1996).

Appendix

A.1 Mathematical Model for the *E. coli* Heat Shock Response

The *E. coli* heat shock response model employed in this paper consists of 28 equations (**Table 4** and **Table 5**), 28 variables, 49 constants (43 kinetic param-

Table 4 Mathematical equations (1).

Equation	Eqn No.
$0 = [\sigma^{70}]_{total} - ([\sigma^{70}] + [\sigma^{70}\colon RNAP] + [Pg\colon \sigma^{70}\colon RNAP] + [D\colon \sigma^{70}\colon RNAP])$	A1
$0 = [RNAP]_{total} - ([RNAP] + [\sigma^{70}\colon RNAP] + [Pg\colon \sigma^{70}\colon RNAP]$ $+[RNAP\colon \sigma^{32}] + [Ph\colon RNAP\colon \sigma^{32}] + [D\colon \sigma^{70}\colon RNAP] + [D\colon RNAP\colon \sigma^{32}]$ $+[RNAP\colon D])$	A2
$0 = [\sigma^{32}]_{total} - ([\sigma^{32}] + [RNAP\colon \sigma^{32}] + [Ph\colon RNAP\colon \sigma^{32}] + [\sigma^{32}\colon DnaK]$ $+[\sigma^{32}\colon FtsH] + [D\colon RNAP\colon \sigma^{32}] + [\sigma^{32}\colon DnaK\colon FtsH])$	A3
$0 = [FtsH]_{total} - ([FtsH] + [\sigma^{32}\colon FtsH] + [\sigma^{32}\colon DnaK\colon FtsH])$	A4
$0 = [DnaK]_{total} - ([DnaK] + [\sigma^{32}\colon DnaK] + [P_{unfold}\colon DnaK]$ $+[\sigma^{32}\colon DnaK\colon FtsH])$	A5
$0 = [P_{unfold}]_{total} - ([P_{unfold}] + [P_{unfold}\colon DnaK])$	A6
$0 = [Pg]_{total} - ([Pg] + [Pg\colon \sigma^{70}\colon RNAP])$	A7
$0 = [Ph]_{total} - ([Ph] + [Ph\colon RNAP\colon \sigma^{32}])$	A8
$0 = Kb_1[\sigma^{70}][RNAP] - [\sigma^{70}\colon RNAP]$	A9
$0 = Kb_2[Pg][\sigma^{70}\colon RNAP] - [Pg\colon \sigma^{70}\colon RNAP]$	A10
$0 = Kb_3[RNAP][\sigma^{32}] - [RNAP\colon \sigma^{32}]$	A11
$0 = Kb_4[Ph][RNAP\colon \sigma^{32}] - [Ph\colon RNAP\colon \sigma^{32}]$	A12
$0 = Kb_5[\sigma^{32}][DnaK] - [\sigma^{32}\colon DnaK]$	A13
$0 = Kb_6[\sigma^{32}][FtsH] - [\sigma^{32}\colon FtsH]$	A14
$0 = Kb_7[P_{unfold}][DnaK] - [P_{unfold}\colon DnaK]$	A15
$0 = Kb_8[D][\sigma^{70}\colon RNAP] - [D\colon \sigma^{70}\colon RNAP]$	A16
$0 = Kb_9[D][RNAP\colon \sigma^{32}] - [D\colon RNAP\colon \sigma^{32}]$	A17
$0 = Kb_{10}[RNAP][D] - [RNAP\colon D]$	A18
$0 = Kb_{11}[\sigma^{32}\colon DnaK][FtsH] - [\sigma^{32}\colon DnaK\colon FtsH]$	A19

eters and 6 constant concentrations) (**Table 6**, **Table 7**, and **Table 8**).

A.2 Detailed Algorithm of Two-phase Search (TPS)

The TPS method combines a random search with a search by GAs in order to achieve a high-speed and nonbiased search. **Figure 10** is a detailed algorithm of TPS.

A.3 Statistical Method to Examine the Significant Difference in EVA, CRV, and SDV

We employed the Wilcoxon rank sum test to examine the significant difference in EVA by RS, SGA, and TPS. In the section of TPS application to benchmark functions, we performed 10 times of trial for all searches. Therefore, we conclude that there is a significant difference in the two mean values for EVA, when the Wilcoxon rank sum test using 20 values of EVA for the two searches provides p-value < 0.05.

Table 5 Mathematical equations (2).

Equation	Eqn No.
$\frac{d[P_{fold}]}{dt} = kp_4[mRNA(Protein)] - kx_2[P_{fold}] + kx_3[P_{unfold}\colon DnaK]$ $-kpd_5[P_{fold}]$	A20
$\frac{d[mRNA(\sigma^{32})]}{dt} = km_1 \frac{[Pg\colon \sigma^{70}\colon RNAP]}{[Pg]_{total}}[G] - kmd_1[mRNA(\sigma^{32})]$	A21
$\frac{d[mRNA(DnaK)]}{dt} = km_2 \frac{[Ph\colon RNAP\colon \sigma^{32}]}{[Ph]_{total}}[G] - kmd_2[mRNA(DnaK)]$	A22
$\frac{d[mRNA(FtsH)]}{dt} = km_3 \frac{[Ph\colon RNAP\colon \sigma^{32}]}{[Ph]_{total}}[G] - kmd_3[mRNA(FtsH)]$	A23
$\frac{d[mRNA(Protein)]}{dt} = km_4[G] - kmd_4[mRNA(Protein)]$	A24
$\frac{d[\sigma^{32}]_{total}}{dt} = kp_1[mRNA(\sigma^{32})] - kpd_1[\sigma^{32}] - kpd_8[RNAP\colon \sigma^{32}]$ $-kpd_9[Ph\colon RNAP\colon \sigma^{32}] - kpd_{10}[\sigma^{32}\colon DnaK] - kx_1[\sigma^{32}\colon FtsH]$ $-kpd_{11}[\sigma^{32}\colon FtsH] - kpd_{14}[D\colon RNAP\colon \sigma^{32}]$ $-kx_4[\sigma^{32}\colon DnaK\colon FtsH] - kpd_{16}[\sigma^{32}\colon DnaK\colon FtsH]$	A25
$\frac{d[FtsH]_{total}}{dt} = kp_2[mRNA(FtsH)] - kpd_2[FtsH] - kpd_{11}[\sigma^{32}\colon FtsH]$ $-kpd_{16}[\sigma^{32}\colon DnaK\colon FtsH]$	A26
$\frac{d[DnaK]_{total}}{dt} = kp_3[mRNA(DnaK)] - kpd_3[DnaK] - kpd_{10}[\sigma^{32}\colon DnaK]$ $-kpd_{12}[P_{unfold}\colon DnaK] - kpd_{16}[\sigma^{32}\colon DnaK\colon FtsH]$	A27
$\frac{d[P_{unfold}]_{total}}{dt} = kx_2[P_{fold}] - kpd_4[P_{unfold}] - kx_3[P_{unfold}\colon DnaK]$ $-kpd_{12}[P_{unfold}\colon DnaK]$	A28

We employed the goodness-of-fit test to examine the significant difference in the solution distribution among RS, SGA and TPS. After the searches by RS, SGA and TPS, we calculated the frequency distribution of the solution with respect to each kinetic parameter in the same manner as Fig. 4, and then applied the goodness-of-fit test to the frequency distributions. We conclude that there is a significant difference between the two frequency distributions by RS and TPS or between those by SGA and TPS, when the goodness-of-fit test using the frequency distributions of the solution presents p-value < 0.05.

Intrinsically the goodness-of-fit test requires the “control” frequency distribution. We employed the solution distribution obtained by RS as the control frequency distribution to explore significant differences between RS and TPS. Next, we employed the solution distribution obtained by SGA as the control frequency distribution to explore significant differences between SGA and TPS.

Table 6 A list of biochemical parameters (1).

State in Eq. (1) and Eq. (2)	Component	Definition	Initial Concentration (M)
x	$[\sigma^{70}]$	σ^{70}	6.5339×10^{-10}
x	$[RNAP]$	RNA polymerase core	2.7818×10^{-10}
x	$[\sigma^{32}]$	σ^{32}	5.6533×10^{-11}
x	$[FtsH]$	FtsH (protease)	9.8674×10^{-7}
x	$[DnaK]$	DnaK (chaperone)	1.0117×10^{-6}
x	$[P_{unfold}]$	Unfolded proteins	2.4860×10^{-5}
x	$[Pg]$	Housekeeping gene promoters	8.5974×10^{-6}
x	$[Ph]$	HSP gene promoters	7.5019×10^{-8}
x	$[\sigma^{70}\colon RNAP]$	Holoenzyme of RNAP-bound σ^{70}	1.8176×10^{-10}
x	$[Pg\colon\sigma^{70}\colon RNAP]$	σ^{70}:RNAP-bound promoter Pg	1.5627×10^{-6}
x	$[RNAP\colon\sigma^{32}]$	Holoenzyme of RNAP-bound σ^{32}	1.5728×10^{-11}
x	$[Ph\colon RNAP\colon\sigma^{32}]$	RNAP:σ^{32}-bound promoter Ph	1.1798×10^{-9}
x	$[\sigma^{32}\colon DnaK]$	σ^{32}-bound DnaK	5.7193×10^{-10}
x	$[\sigma^{32}\colon FtsH]$	FtsH-bound σ^{32}	5.5783×10^{-9}
x	$[P_{unfold}\colon DnaK]$	Unfolded protein-bound DnaK	2.5150×10^{-5}
x	$[D\colon\sigma^{70}\colon RNAP]$	σ^{70}:RNAP-bound D	1.8558×10^{-8}
x	$[D\colon RNAP\colon\sigma^{32}]$	RNAP:σ^{32}-bound D	2.1448×10^{-7}
x	$[RNAP\colon D]$	D-bound RNAP	3.2827×10^{-6}
x	$[\sigma^{32}\colon DnaK\colon FtsH]$	σ^{32}:DnaK-bound FtsH	5.6434×10^{-8}

Table 7 A list of biochemical parameters (2).

State in Eq. (1) and Eq. (2)	Component	Definition	Initial Concentration (M)
y	$[P_{fold}]$	Folded proteins	5.0300×10^{-3}
y	$[mRNA(\sigma^{32})]$	mRNA of σ^{32}	1.5627×10^{-8}
y	$[mRNA(DnaK)]$	mRNA of DnaK	3.9327×10^{-8}
y	$[mRNA(FtsH)]$	mRNA of FtsH	1.5731×10^{-9}
y	$[mRNA(Protein)]$	mRNA of proteins	7.6200×10^{-6}
y	$[\sigma^{32}]_{total}$	Total σ^{32}	8.2395×10^{-8}
y	$[FtsH]_{total}$	Total FtsH	1.0487×10^{-6}
y	$[DnaK]_{total}$	Total DnaK	2.6218×10^{-5}
y	$[P_{unfold}]_{total}$	Total P_{unfold}	5.0010×10^{-5}
Constant	$[G]$	Molar concentration of one molecule per cell	2.54×10^{-9}
Constant	$[D]$	Nonspecific DNA binding sites	1.18×10^{-2}
Constant	$[\sigma^{70}]_{total}$	Total σ^{70}	1.778×10^{-6}
Constant	$[RNAP]_{total}$	Total RNAP	5.08×10^{-6}
Constant	$[Pg]_{total}$	Total Pg	1.016×10^{-5}
Constant	$[Ph]_{total}$	Total Ph	7.62×10^{-8}

Table 8 A list of kinetic parameters. The increase in the values of kx_2 and kp_1 emulates heat shock and the response, respectively.

Parameter	Definition	Unit or Value
Kb_1	Binding constant between RNAP and σ^{70}	M^{-1}
Kb_2	Binding constant between Pg and σ^{70}:RNAP	M^{-1}
Kb_3	Binding constant between RNAP and σ^{32}	M^{-1}
Kb_4	Binding constant between Ph and RNAP:σ^{32}	M^{-1}
Kb_5	Binding constant between DnaK and σ^{32}	M^{-1}
Kb_6	Binding constant between FtsH and σ^{32}	M^{-1}
Kb_7	Binding constant between DnaK and P_{unfold}	M^{-1}
Kb_8	Binding constant between D and σ^{70}:RNAP	M^{-1}
Kb_9	Binding constant between D and RNAP:σ^{32}	M^{-1}
Kb_{10}	Binding constant between D and RNAP	M^{-1}
Kb_{11}	Binding constant between FtsH and σ^{32}:DnaK	M^{-1}
kx_1	Degradation constant of FtsH-bound σ^{32}	5 min^{-1}
kx_2	Unfolding rate constant of folded proteins	75 → 150 min^{-1}
kx_3	Refolding rate constant of unfolded proteins	15000 min^{-1}
kx_4	Degradation rate constant of FtsH-bound σ^{32}:DnaK	5 min^{-1}
kp_1	Translation rate constant for σ^{32}	20 → 80 min^{-1}
kp_2	Translation rate constant for FtsH	20 min^{-1}
kp_3	Translation rate constant for DnaK	20 min^{-1}
kp_4	Translation rate constant for proteins	20 min^{-1}
kpd_1-kpd_5, kpd_8-kpd_{12}, kpd_{14}, kpd_{16}	Protein degradation rate constant	0.03 min^{-1}
kpd_6, kpd_7, kpd_{13}, kpd_{15}	Protein degradation rate constant	1 min^{-1}
km_1	Transcription rate constant for σ^{32}	20 min^{-1}
km_2	Transcription rate constant for DnaK	500 min^{-1}
km_3	Transcription rate constant for FtsH	20 min^{-1}
km_4	Transcription rate constant for proteins	1500 min^{-1}
kmd_1-kmd_4	mRNA degradation constant	0.5 min^{-1}

```
$N \leftarrow 0$
While $N <$ Required number of solutions:
    Randomly generate parameter vector $\mathbf{P}$
    If Fitness($\mathbf{P}$) $< AE$:
        Store $\mathbf{P}$ as a solution
        $N \leftarrow N + 1$
    End
    Else if Fitness($\mathbf{P}$) $< AEC$:
        As an initial population, create $n_p - 1$ individuals randomly in
        the hypercube whose center is $\mathbf{P}$ and the edge length is $RIG$
        Add $\mathbf{P}$ to the initial population
        $Generation \leftarrow 1$
        While $Generation \leq n_g$:
            If the minimum value of fitness in population $< AE$ :
                Store the individual that gives the minimum fitness
                as a solution
                $N \leftarrow N + 1$
                Break
            End
            Execute selection, crossover, and mutation
            $Generation \leftarrow Generation + 1$
        End
    End
End
```

Fig. 10 Detailed algorithm of TPS. n_p is the population size and n_g is the maximum generation.

(Received August 8, 2008)
(Accepted November 12, 2008)
(Released March 24, 2009)

(Communicated by *Susumu Goto*)

Kazuhiro Maeda was born in 1984. He received his M. in Information Engineering from Kyushu Institute of Technology in 2008. Since 2008 and now he is a doctor course student at Department of Bioscience and Bioinformatics, Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology. His research interests are in systems biology. He is a student member of IPSJ and JSBi.

Hiroyuki Kurata received his Ph.D. in Chemical Engineering from The University of Tokyo in 1993. He was an assistant professor of Department of Chemical Engineering there from 1993 to 1994. From 1994, he was visiting scholar in Department of Plant Pathology, University of California, Davis (Research fellow of Japan Society for the Promotion of Science). From 1996 to 2000, he was an assistant professor in Department of Chemistry and Biotechnology, The University of Tokyo. Then he has moved to Department of Bioscience and Bioinformatics, Kyushu Institute of Technology as an associate professor in 2000. Since 2006, he has been a professor there. He is a member of The Society of Chemical Engineers, The Society for Bioscience and Bioengineering, The Molecular Biology of Japan, and The Japanese Society for Bioinformatics.