Bioprocess Biosyst Eng (2011) 34:433–446
DOI 10.1007/s00449-010-0486-7

ORIGINAL PAPER

An integrative and practical evolutionary optimization for a complex, dynamic model of biological networks

Kazuhiro Maeda · Yuya Fukano · Shunsuke Yamamichi · Daichi Nitta · Hiroyuki Kurata

Received: 23 August 2010 / Accepted: 4 November 2010 / Published online: 27 November 2010

Abstract Computer simulation is an important technique to capture the dynamics of biochemical networks. Numerical optimization is the key to estimate the values of kinetic parameters so that the dynamic model reproduces the behaviors of the existing experimental data. It is required to develop general strategies for the optimization of complex biochemical networks with a huge space of search parameters, under the condition that kinetic and quantitative data are hardly available. We propose an integrative and practical strategy for optimizing a complex dynamic model by using qualitative and incomplete experimental data. The key technologies are the divide and conquer method for reducing the search space, handling of multiple objective functions representing different types of biological behaviors, and design of rule-based objective functions that are suitable for qualitative and error-prone experimental data. This strategy is applied to optimizing a dynamic model of the yeast cell cycle to demonstrate the feasibility of it.

Keywords Dynamic simulation · Biochemical network · Multi-objective optimization · Model fitting · Cell cycle

Electronic supplementary material The online version of this article (doi:10.1007/s00449-010-0486-7) contains supplementary material, which is available to authorized users.

K. Maeda · Y. Fukano · S. Yamamichi · D. Nitta · H. Kurata (✉)
Department of Bioscience and Bioinformatics, Kyushu Institute of Technology, 680-4 Kawazu, Iizuka, Fukuoka 820-8502, Japan
e-mail: kurata@bio.kyutech.ac.jp

Introduction

Systems biology aims at constructing a large-scale, dynamic model of complex biochemical networks in silico and at understanding the mechanism of how such systems generate a variety of cellular functions. Advances in molecular biology and omics technology have extensively revealed details of biochemical reactions and gene interaction networks, enabling drawing biochemical network maps in various cellular systems such as apoptosis, cell cycles, differentiation, metabolic networks, and stress responses. A major problem for dynamic modeling is to know the values of kinetic parameters in vivo, but it is very hard to measure the exact values of them due to experimental complexity [1–3]. As an alternative way, numerical optimization is presented to estimate the values of kinetic parameters so that they reproduce the behaviors of existing biological data.

A general strategy for dynamic modeling has been proposed that combines reverse engineering with forward engineering [4–7]. Ideally, we would like to gain access to the activities of all important molecular species including complexes and modified molecule, while it is hard to know the molecular details of all biochemical reactions. There is a strong need for methods that can handle complicated molecular systems at an abstract level without going all the way down to biochemical reactions with exact kinetic parameters [8]. Forward engineering builds the mathematical models that directly reflecting the essential network architecture; reverse engineering explores their associated kinetic parameter values so that the models can reproduce experimental data. From this viewpoint, the model would focus on capturing the intrinsic architecture of molecular networks rather than their detailed kinetics. In the reverse engineering, it is a challenging task to find optimal

solutions out of a huge search space that can explain the experimental data. The current deterministic methods for global optimization of non-linear dynamic models are too expensive in terms of computational cost. In contrast, stochastic methods including evolutionary (genetic) algorithms can provide high-quality solutions in less computational cost [2, 9, 10]. At present, evolutionary searches are widely used to optimize a dynamic model of biological systems.

The performance for optimization typically depends on the cost or objective functions chosen. In many cases, (weighted) least squares estimators or maximum likelihood estimators are used with respect to time course data of molecular components [1, 2, 11–14]. While most of the proposed algorithms pursue numerically or theoretically rigorous ways to find a global solution, a serious question raises if such rigorous approaches are really achieved or effective in biology, because experimental data have uncertainty and considerable errors due to the inherent properties of molecular components and immature measurement techniques [15, 16]. In general, the time course data for each molecular concentration are used as a reference or experimental model to estimate the kinetic parameter values, but few kinetic data and few quantitative values are measured in vivo. Furthermore most of experimental observations and biological information are qualitative and fragmental, because they are measured under different experimental and genetic conditions. The typical status of the experimental data is exemplified by a cell cycle network: Growth is delayed when a specific gene is removed, while DNA replication stops when another gene is removed. Under such data quality it does not seem a serious issue to find a global solution. The important thing is to explore the plausible values of kinetic parameters that can explain the dynamic behaviors derived form biological information and experimental data, but there are few reports that challenge such an intrinsic and real problem.

Since the dynamics of cultured cells are described or featured at different abstract levels from molecular kinetics to physiological behaviors under a variety of environmental or genetic conditions, the cells can have multiple objective functions to satisfy those features. In engineering fields, multiple objective optimization techniques have extensively been presented [17–19]. The weighted-sum method is one of the most widely used solutions, which converts a multi-objective optimization problem into a set of single objective problems and defines a weighted sum of them as the unique objective. On the other hand, the set of all Pareto solutions, known as the Pareto frontier, is effective in finding globally optimal solutions [18, 20]. Recently, in biology, multi-objective optimization has shown some benefits compared to single objective approaches [21].

In addition to the above problems, the size of networks is a serious obstacle for optimization. An increase in network size explosively increases the search space of kinetic parameters, thereby making optimization hard. Divide and conquer algorithms have been proposed to be effective in solving this problem in various fields [22]. Decomposition of molecular networks into subsystems is one of the promising solutions for optimizing a dynamic model [23–27].

The objective of this paper is to develop a practical, comprehensive, versatile framework to simultaneously solve the above problems, how to optimize a dynamic model with multi-objective functions by using error-prone, fragmental, qualitative data. The proposed algorithm consists of a module decomposition and integration method, evolutionary optimization with respect to a multi-objective function, and design of the scoring rules to evaluate the dynamic model, named Integrative and Practical Evolutionary Search (IPES). IPES presents a new strategy to challenge the intrinsic and real problems accompanied by the sparsity and uncertainty of biological data, and can be distinguished from the widely used methods that definitely require the reference or nominal time course to pursue a global solution. To demonstrate the feasibility of IPES, it is applied to a cell cycle network whose dynamics are characterized by a variety of experimental data for wild-type and genetic mutants.

Methods

Dynamic model

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

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{F}(t, \mathbf{x}, \mathbf{y}, \mathbf{P}) \tag{1}$$

$$0 = \mathbf{G}(\mathbf{x}, \mathbf{y}, \mathbf{P}) \tag{2}$$

where t is time, and $\mathbf{P}$ is the kinetic parameter vector. The differential equation (1) shows the conversion, degradation and synthesis of molecules, while the algebraic equation (2) indicates the binding reaction for complex formation. In Eq. 1, $\mathbf{x}$ is the independent variable vector that indicates the binding complex concentrations solved by Eq. 2, $\mathbf{y}$ is the time-dependent variable vector that consists of the total concentration vector of elementary and modified molecules. In Eq. 2, $\mathbf{x}$ is the dependent variable vector, while $\mathbf{y}$ is the independent concentration given by Eq. 1. This type of DAEs is generated by the two-phase partition (TPP) method that applies the quasi steady-state approximation to the differential equations describing complex formation

[5, 28]. TPP divides the differential equations into fast and slow reaction. Fast reactions such as complex formation are converted into algebraic equations.

Integrative and practical evolutionary search for optimization of a dynamic model

Since it is difficult to optimize a large-scale model because of a huge space of search parameters, a full model is divided into subsystems with a relatively small number of search parameters. After roughly optimizing each module, all the modules are assembled together to provide the full models. Next, Distributed Cooperation model of Multi-Objective Genetic Algorithm (DCMOGA) is presented to simultaneously optimize different types of experimental data [29]. The scoring rules are designed to numerically evaluate the simulated dynamics in comparison with the reference behaviors built based on biological knowledge and experimental data. These scoring rules enable one to construct a dynamic model based on qualitative and fragmental experimental data.

The integrative and practical evolutionary optimization (IPES) consists of the two major processes: module decomposition and integration, and optimization for multi-objective functions, as shown in Fig. 1. The scoring rules are designed as an if–then rule to define the objective functions. Note that module decomposition is not used in the process of DCMOGA. Details of the procedure are given as follows:

1. Module decomposition and coarse optimization for each module.
   1.1. Decomposition of a full mathematical model into subnetworks (modules) in terms of biological functions and temporal orders.
   1.2. Coarse optimization of each module by Genetic Algorithms (GAs).

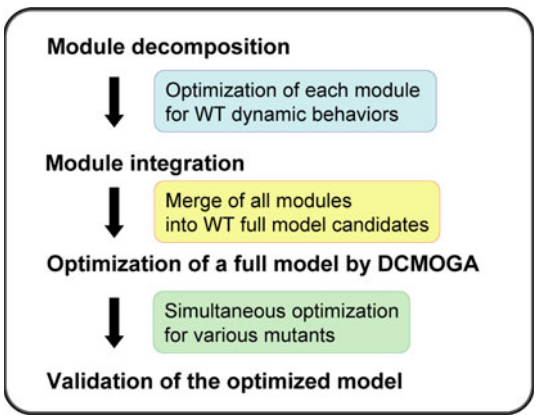

Fig. 1 A procedure for an integrative and practical evolutionary search for optimizing a dynamic model with multi-objective functions

   1.3. Module integration: All suboptimal modules are merged into full model candidates, where the overlapped parameters among modules are averaged.
2. Optimization of the full model with respect to multi-objective functions.

Distributed Cooperation model of Multi-Objective Genetic Algorithm is applied to optimization of the dynamic model. The full model candidates provided by the module integration are employed to make the initial populations.

Module decomposition and integration

Choosing modules is a non-trivial task. A number of strategies can be adapted to ease this task, but we do not address them here [23–26]. In this article, module decomposition can be performed in terms of biological functions and temporal orders, as exemplified in Fig. 2. Ideally, the network may be divided so that each module has fewer external molecules and the size of each module is small enough to optimize. All the molecules within a network can conveniently be separated into M modules, while the interactions from the neighboring modules are not neglected. The mathematical equations within a module can be affected by the molecular components in its neighboring modules.

Mathematical equations are decomposed into M interacting modules (subnetworks): S_k ($k = 1, 2, \ldots M$):

$$\frac{d\mathbf{y}_{S_k}}{dt} = \mathbf{F}_{S_k}(t, \mathbf{x}_{S_k}, \mathbf{y}_{S_k}, \mathbf{y}_{S_j}, \mathbf{P}) \quad (j \neq k)$$
$$0 = \mathbf{G}_{S_k}(\mathbf{x}_{S_k}, \mathbf{y}_{S_k}, \mathbf{y}_{S_j}, \mathbf{P}) \tag{3}$$

$\mathbf{y}_{S_k}$ is the concentration vector for the molecules that belong to module S_k and $\mathbf{y}_{S_j}$ is the concentration vector for the molecules that belong to the external modules $S_j (j \neq k)$. The time course of $\mathbf{y}_{S_j}$ is independently given as specific time functions: $\mathbf{y}_{S_j} = \mathbf{h}(t)$ assumed based on experimental data and biological knowledge. The external ones are regarded as the input signal to the module, which is the same idea as "dependent input" [3, 26]. In the differential equations $\mathbf{x}_{S_k}$ is the independent variable vector given by the algebraic equations and $\mathbf{y}_{S_k}$ is the time-dependent variable vector. In the algebraic equations, $\mathbf{x}_{S_k}$ is the variable vector to be solved, where $\mathbf{y}_{S_k}$ is given by the differential equations.

The kinetic parameters of each module can be optimized with respect to an objective or fitness function, assuming the time courses of the external molecular concentrations, as exemplified in Fig. 3. The important thing at this stage is not to obtain a highest fitness value for each module, but to

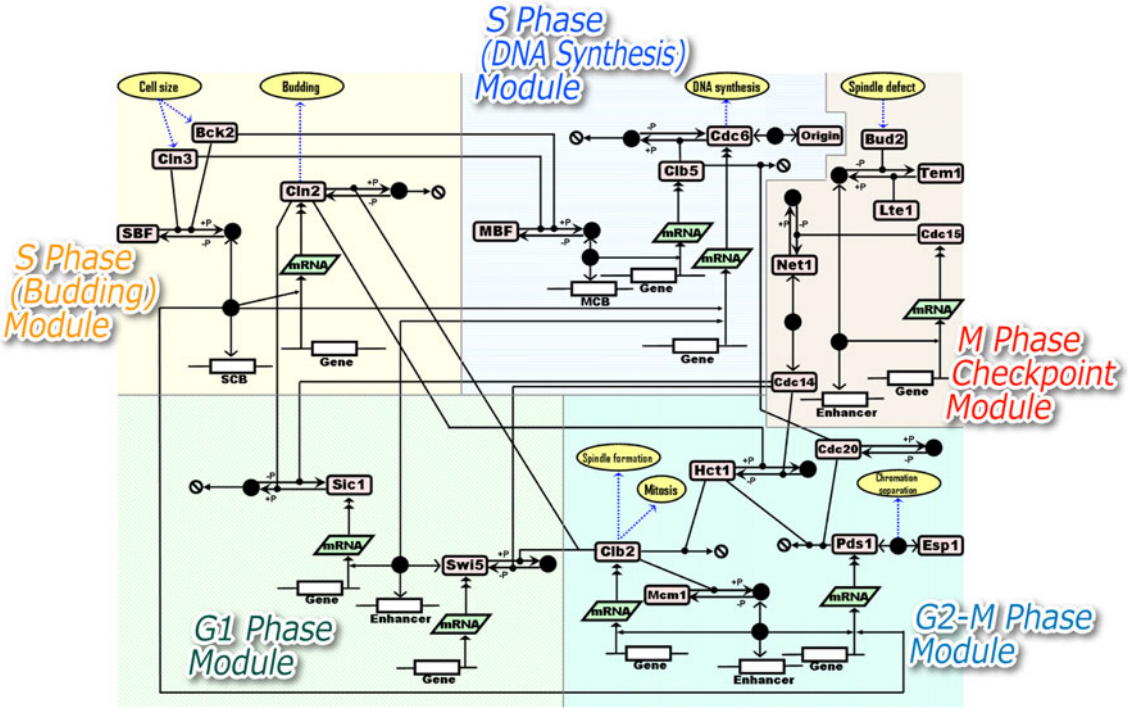

Fig. 2 Modular architecture of a budding yeast cell cycle network map. Five modules are named S phase (budding) module, S phase (DNA synthesis) module, G1 phase module, G2-M phase module, and M phase checkpoint module

provide the initial population used for the subsequent optimization of the full model.

Several suboptimal solutions for each module are combinatorially merged as full model candidates to avoid local minima as much as possible, where the values of overlapped parameters among modules are averaged. These candidates are used as the initial population for the subsequent DCMOGA.

Optimization for multi-objective functions

Pareto solution for multi-objective functions

Optimization of the mathematical model that reproduces different types of dynamic features can be a multi-objective problem. Generally it can be transformed into a single objective problem by linearly combining multiple objective

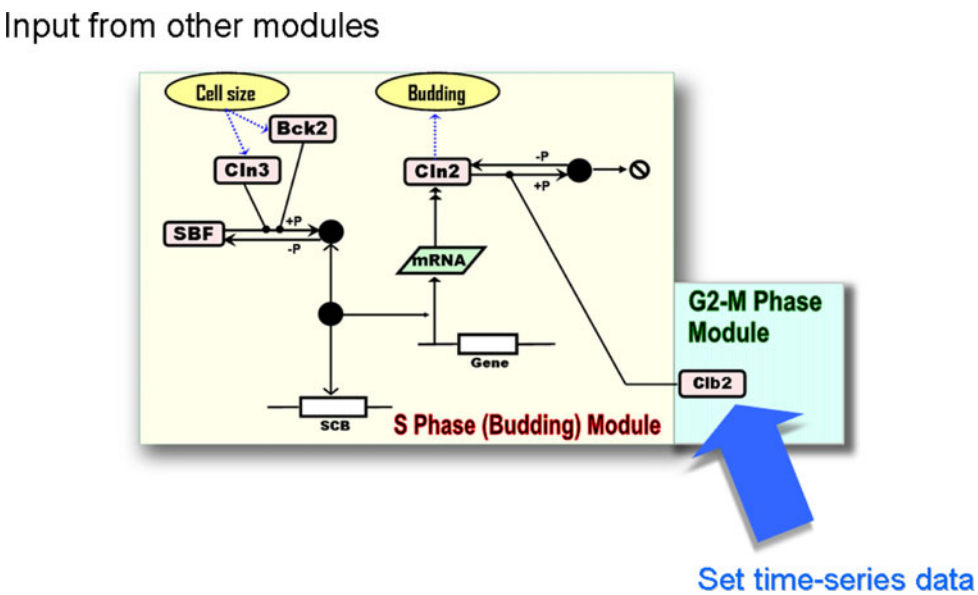

Fig. 3 Method of optimization for each module. Each module is optimized under the condition that the time courses of the external components are provided. In this figure, the S phase (budding) module is optimized, assuming the time course of the Clb2 (external component or dependent input) concentration

functions with different weight coefficients. However, since tradeoff can be obtained among objective functions, it is not satisfactory to optimize such a transformed single objective function. Instead, the Pareto optimality is presented to consider a tradeoff relationship among N objective functions: $\mathbf{f}(\mathbf{p}) = (f_1(\mathbf{p}), \ldots, f_N(\mathbf{p}))$. Here, we define a dominance relation between two solutions. $\mathbf{p}_1$ dominates $\mathbf{p}_2$ when the following condition is satisfied:

$$f_i(\mathbf{p}_1) \geq f_i(\mathbf{p}_2) \quad \forall i \in 1, \ldots, N \quad \text{and}$$
$$f_j(\mathbf{p}_1) > f_j(\mathbf{p}_2) \quad \exists j \in 1, \ldots, N.$$

The Pareto optimal solutions $\mathbf{p}*$ are defined as the solutions that are not dominated by any other solutions. Among the Pareto optimal solutions, there can be a tradeoff relation in which if one objective function is going to improve, another one deteriorates.

DCMOGA

DCMOGA is employed for handling multiple objective functions: $\mathbf{f}(\mathbf{p}) = (f_1(\mathbf{p}), \ldots, f_N(\mathbf{p}))$, as shown in Fig. 4 [29]. DCMOGA searches for the Pareto-optimum solutions

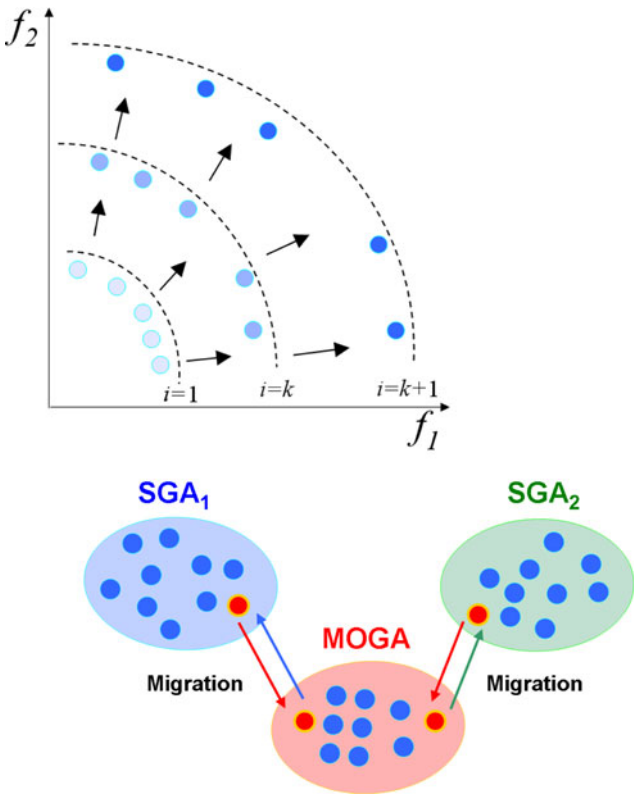

Fig. 4 Illustrations of Pareto solutions and the islands in DCMOGA. **a** Advancement of the Pareto front. *Circles* indicate individuals and *dotted lines* indicate the Pareto fronts. i is the number of generation. **b** Each SGA island has its own objective function. In the MOGA island, the individuals are ranked according to the Pareto ranking method for all the objective functions. At a regular interval some individuals are exchanged between MOGA and SGAs

$\mathbf{p}^*$, where the advancement of the Pareto front is done at every generation [20] (Fig. 4a). In DCMOGA, $N + 1$ subpopulations (islands) are created with respect to N objective functions (objects) (Fig. 4b). One of these islands is the population for evaluating all objective functions called a MOGA group. The other N groups are the population for finding an optimum of each objective function, called SGA groups.

DCMOGA finds the Pareto solutions while maintaining variations in solutions, as follows:

1. Initial population setting

Since the combinatorially possible number of the full model candidates is large, the kinetic parameter vectors that provide higher fitness values are selected from each module and are combined to make the initial population.

2. Optimization in the MOGA and SGA islands

The individuals in the MOGA group are optimized with respect to multiple objective functions by using GAs. On the other hand, each SGA group is independently optimized just for one objective function ($f_i(\mathbf{p})$) during the intervals of the migration time.

3. Migration among the islands

The Pareto-front solutions in the MOGA group are exchanged for the parameter sets with a highest fitness value in each SGA group. The solution with the highest fitness in each SGA is immigrated to MOGA and is substituted for the solution with the lowest rank there. Simultaneously, the solution with the highest rank in MOGA is immigrated to each SGA and is substituted for the solution with the lowest fitness there. The parameter set with the highest fitness in each SGA group is sent to the MOGA group, while the Pareto-front in the MOGA group is sent to all the SGA groups.

4. Repetition of optimization and migration

Return to (2) until the number of termination or the desired solutions are obtained.

Normal MOGA

As a control method, the normal multi-objective genetic algorithm (MOGA) method is used. The MOGA method has just one MOGA island without any SGA islands, where the Pareto-optimum solutions are searched with respect to multiple objective functions.

Genetic algorithms

Instead of the conventional binary- or Gray-expression, the real-coded GA uses the real numerical value expression as

a gene. One individual with the search kinetic parameters expresses one model. Genetic algorithms are performed as follows.

1. Initialization

Generate initial population and assign uniform random numbers within the search space to each parameter of all individuals.

2. Evaluation and selection

The fitness value for each individual is calculated. The individuals with the higher fitness (the elite individuals) are kept without any evolutionary operation, while the remaining ones are sent to evolutionary operation (3). Optimization is completed when the number of generations reaches the upper limit.

3. Evolutionary operation: crossover

Unimodal normal distribution crossover (UNDX) is used as the crossover method to make the offspring individuals from the parent individuals [30, 31]. In this study, mutations are not used.

4. Back to (2).

Pareto ranking

Selection in the MOGA group is based on the dominance relation. If **p** dominates N solutions, the rank of **p** is given as $r(\mathbf{p}) = 1 + N$. The high-ranking solutions are kept as the elite individuals.

Scoring rules for calculating a fitness value

Since experimental data can be obtained under different conditions by a variety of experimental techniques involving molecular biology or omics technology, they are often qualitative, fragmental and heterogeneous. Thus, it is hard to present the exact time course of molecular concentrations or to define mathematically rigorous objective functions, e.g., a sum of the square of the difference between the simulated time course and the associated reference or experimental curve. The important thing is to capture typical features of dynamics, not to simulate the exact behavior of the reference model.

To effectively use such biological data, the score functions are proposed to consist of if–then rules that evaluate the simulated time course of specific molecular components and events. For example, a scoring rule for the ith component is determined as follows:

$$
\begin{aligned}
&s_i = 0\\
&\text{FOR} \quad t = t_{start} \quad \text{to} \quad t_{final} \quad \text{DO}\\
&\quad \text{IF} \quad t_1 < t < t_2 \quad \text{THEN}\\
&\quad\quad \text{IF} \quad c_i(t) > \theta_i \quad \text{THEN}\\
&\quad\quad\quad s_i = s_i + a\\
&\quad\quad \text{ENDIF}\\
&\quad \text{ENDIF}\\
&\text{ENDFOR}
\end{aligned}
$$

where t is the time, θ_i is the threshold, c_i is the concentration for the ith component, s_i is the score for the ith component, and a is the points that are empirically determined based on experimental data. Such empirical scoring rules calculate objective (fitness) functions. Details of how to design an objective function according to the scoring rules are illustrated in the following section (Supplementary Table S3). The objective (fitness) function for a dynamic model is defined as the sum of all the scores:

$$Objective(Fitness)\_function = \sum_i s_i$$

Application to a yeast cell cycle model

A biochemical network map

A network map of the budding yeast cell cycle is shown in Fig. 2, using the CADLIVE notation [32, 33]. This map is one of the most sophisticated images of the whole system of an yeast cell cycle, which is consistent with the previous work [34]. Details of the reactions are summarized elsewhere [33, 34]. In the map, at Start, a series of events is initiated in rapid succession: SBF turns on Cln2 and Clb5 levels, Sic1 disappears, Hct1 turns off, and DNA synthesis and bud emergence commence. Shortly thereafter, Clb2 level rises, and a spindle starts to form. At Finish, active Cdc20 turns on Hct1 by overwhelming the inhibition exerted on Hct1 by Clb2. When Hct1 turns on, Clb2 is degraded and the control system switches to the G1 state, in which the enemies of Clbs (Hct1 and Sic1) are active.

Rule-based dynamic model

When the exact values of kinetic parameters are not measured in vivo and details of reaction networks are hard to fully identify, the dynamic model is built mainly based on qualitative features and experimental observations. Most of

the kinetic parameters are provided nominal values so that the model reproduces most of experimental observations. Thus, the model would be constructed at the coarse-grained level using the formalism of rules, rather than at the level of the exact and full kinetics. It uses temporal logic or rules to specify qualitative and quantitative system behaviors.

Assuming that binding reactions are much faster than gene expression, the TPP method is employed to make the dynamic model, resulting in differential–algebraic equations (Supplementary Table S1). The binding reactions are described by algebraic equations and the others are represented by ordinary differential equations. Details of the conversion method are described elsewhere [5]. The generated dynamic model has 116 differential–algebraic equations with 138 kinetic parameters. The reactions employed in our dynamic model are almost the same as those employed in Chen's model, while the mathematical equations are not consistent with their model, because TPP is used for mathematical modeling. The reason for use of TPP is that it can automatically convert reaction networks into mathematical equations [5], leading to our final goal of automatic modeling of biochemical networks.

Logical rules that express a change in biological events are described by the Hill type equation instead of if–then rules, as shown Supplementary Table S1D. A high number of the Hill coefficient provides the switch-like behavior as well as if–then rules.

Thirty-four search parameters are listed in Supplementary Table S2 and the other kinetic parameters are assumed or estimated, as shown in Supplementary Table S1. Note that there are few kinetic data and few quantitative data in vivo due to experimental complexity, while a series of the cell cycle reactions and events have intensively been investigated. Thus, nominal values are assigned to kinetic parameters. For example, they are set so that transcription and translation occur on the minute order, and binding among proteins and DNAs occurs on the second order. The concentrations of proteins within a cell are given as the nano-order molar concentration.

Module decomposition

As shown in Fig. 2, the dynamic model is decomposed into three temporal modules: the G1, S, and M phases, since the cell cycle networks look cascade reactions. Furthermore, the S phase was divided into the budding and DNA synthesis modules, and the M phase into the spindle formation and spindle checkpoints. Finally, the network consists of these five modules. In each module, the search parameters are optimized by GAs (population number: 100, maximum generation: 50), while the time courses of the external molecules that act on the intra-modular molecules are provided/assumed (Fig. 3). The fitness function, consisting of a sum of score functions, is set to each module, which are designed by assuming the dynamics of major components of wild type (Cln2, Clb2, Clb5, and Sic1, and three events of Cln3 activity, spindle formation, and origin replication).

Module integration

Before performing DCMOGA, the initial population is generated by merging the independently suboptimized modules into a full model. Out of the coarsely suboptimal solutions for each module, four solutions are picked up. The four solutions are exhaustively combined among all the five modules, resulting in $4^5 = 1{,}024$ kinetic parameter sets. The objective (fitness) function for wild-type is simulated for 1,024 parameter sets and 50 sets showing a high fitness value are selected as the initial population of DCMOGA.

Full model optimization for multi-objective functions

DCMOGA starts for the five objective functions assigned to wild type and four mutants. The objective function for each mutant is built based on its biological behaviors, as shown in Supplementary Table S3. Each objective function is assigned to the SGA island, while the Pareto solutions for all the five objective functions are explored in the MOGA island. Maximum generation is set to 50 and the population number of each island is set to 50. The 50 parameter solutions with higher fitness, obtained by module integration, are set as the initial population for the MOGA island. The search parameters are the same as employed by the module decomposition and integration method. GAs are employed, where UNDX is used without mutation and the migration interval is set to 4.

Score evaluation

The scoring rules provide the score for the simulated time course. They check whether the simulated data correspond to the reference behaviors at the regular time steps, and add or subtract a score according to the if–then rules. If the simulated results agree the reference model, the score is added, otherwise it is subtracted. For wild-type, the simulated time course of Cln2, Clb5, Clb2, and Sic1, and three events of Cln3 activity, origin replication, and spindle formation are evaluated. The progress in each event is represented as the numerical index. The size of time step is fixed to 100 steps per minute in the simulation. In every step, the differential–algebraic equations (Table S1A), event functions (Table S1D) and the scoring functions (Table S3) are evaluated. The fitness function is calculated over two cycles, because it is necessary to judge whether the cycle restarts

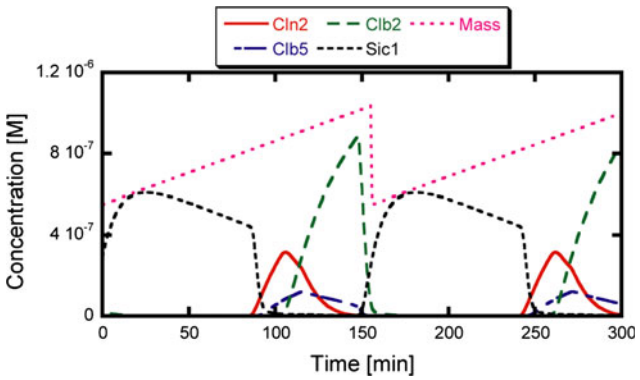

Fig. 5 Reference curves of the cell cycle model. The reference curves are drawn so that it provides the maximum score calculated from the scoring rules

after M phase. Details of the scoring rules are described in Supplementary Table S3. The fitness values, the values of the objective function, are normalized by 100 of the maximum score where the simulated behaviors are completely consistent with the reference behaviors. A time course curve that presents a score of 100 is shown in Fig. 5.

Implementation

The optimization programs are written in C language. Calculation is carried out on Dell-Optiplex 755 (Intel Core2 Duo 2.33 GHz with 2.00 GB RAM).

Results and discussion

Module decomposition and integration

In terms of biological functions and temporal order of reactions, the wild-type network of the budding yeast cell cycle is divided into five modules, as shown in Fig. 2. The search parameters (Supplementary Table S2) are optimized by GAs for the objective (fitness) function for wild-type (Supplementary Table S3A). The fitness values for all the modules are simulated as shown in Fig. 6. The fitness values are normalized by setting the highest fitness that satisfies all the scoring rules to 100. The fitness value increases toward 100 for all the modules. At a generation of 50, four parameter solutions with high fitness values are picked up out of each module, and merged combinatorially to form the kinetic parameter vectors for the full model, resulting in 1,024 sets of kinetic parameter vectors. The fitness values of 1,024 models are distributed from 80 to 92 (data not shown). As a control, the non-decomposed or full model is ten times optimized over a generation number of 1,000, where the search parameters and their search space are set to the same as those employed in the decomposition and integration method. The fitness values for all the optimization trials are <60. The decomposition and integration method is shown to greatly increase the fitness value compared with the non-decomposition method.

Full model optimization for multiple objectives

Next, the generated full model is further optimized with respect to various genetic mutants (multi-objective functions) by using DCMOGA. An objective function is assigned to each knockout mutant (Sic1, Cln2, Clb2, Clb5) and wild type. Fifty parameter solutions showing higher fitness values out of the resultant 1024 simulations are set as the initial population for the MOGA island. The time evolution of the fitness values on the Pareto front are plotted with respect to each objective function during 50 generations, as shown in Fig. 7. The highest fitness score was picked up from the Pareto front for each generation. As a control method, the fitness values on the Pareto front are simulated by the normal MOGA method, which explores the solutions just within the MOGA island without any SGA islands. Note that the calculation cost of DCMOGA is approximately twice as high as that of the normal MOGA method. DCMOGA uses one MOGA and four SGA islands to be calculated for each generation. In the MOGA island, five batch simulations are carried out with respect to wild type and four genetic mutants. In each SGA, one batch simulation is performed for each mutant. On the other hand, the normal MOGA method utilizes just one MOGA island. Thus, two generations for the normal MOGA method is corrected so as to correspond to one generation in Fig. 7. DCMOGA increases the fitness value for all the objective functions compared with the normal MOGA, indicating that DCMOGA is effective in optimization of multiple-objective functions. The fitness values move up and down. It indicates a tradeoff among the objective functions, where an increase in the fitness value for one objective function causes the decrease for another function. Each module optimization needs 10 h and DCMOGA requires 50 h. The total time required for optimization is approximately 100 h ($10 \times 5 + 50$).

Validation of an optimized model

To demonstrate the validity of the dynamic model optimized by the IPES method, the models of wild-type and mutants are simulated, as shown in Fig. 8. The optimized model well reproduces the qualitative behaviors of key molecules as follows. In a Sic1 knockout mutant, cell cycle events are slightly advanced [35]. In a Cln2 knockout mutant, an increase in Sic1 is observed during G1 phase, while an increase in Cln2 and Clb5 does not occur, indicating that cell cycle stops in S phase [36]. In the process of DCMOGA, the fitness value for Cln2 knockout mutant

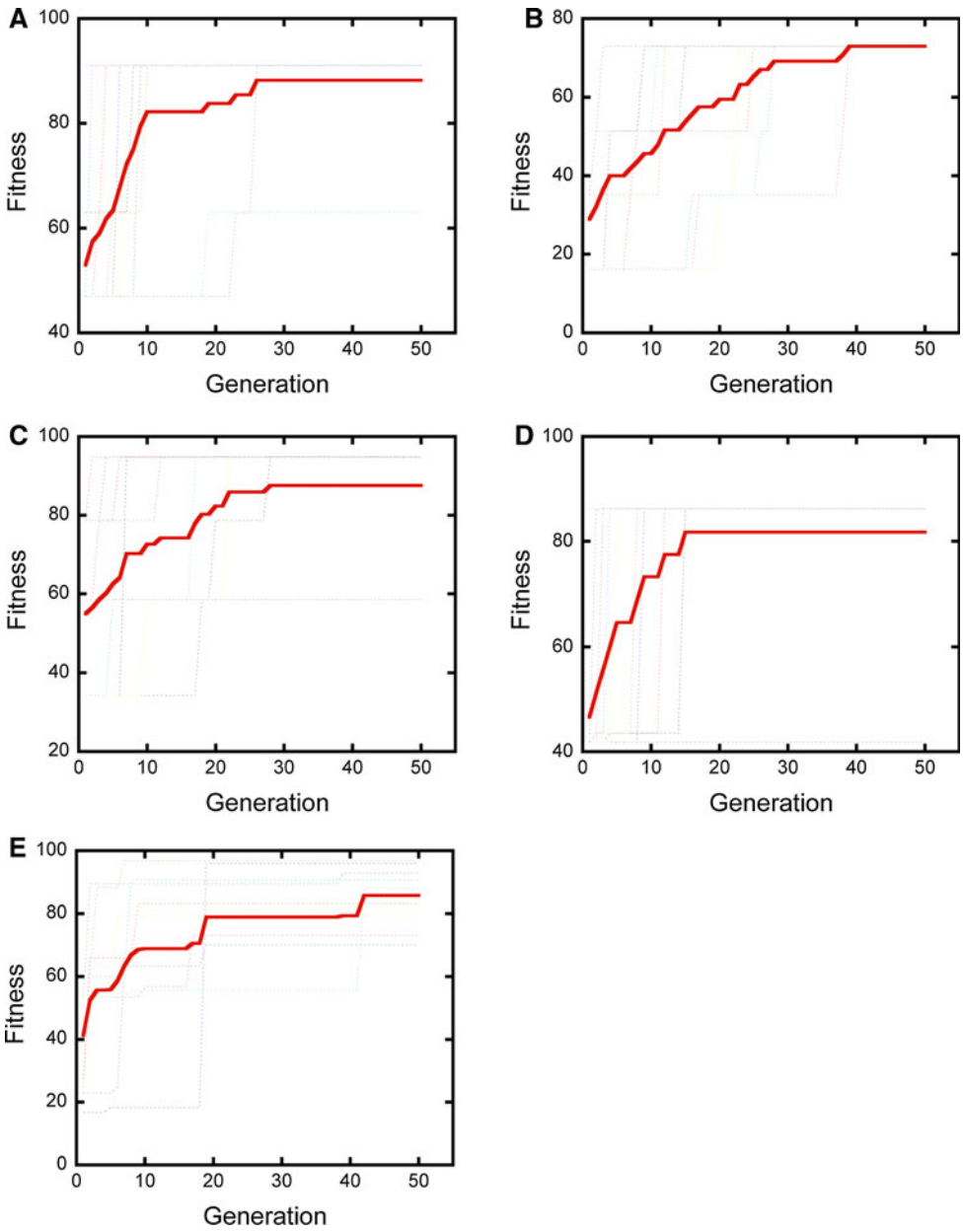

Fig. 6 Evolution of the fitness values of each module. Five modules are optimized: S phase (budding) module (**a**), S phase (DNA synthesis) module (**b**), G1 phase module (**c**), G2-M phase module (**d**), and M phase checkpoint module (**e**). The *thick lines* are the mean of the ten trials of simulations (*dotted lines*)

would decrease as the tradeoff of the fact that other fitness values increase (Fig. 7c). Although the fitness is low, the simulated behavior that the cycle halts at S phase is consistent with experimental data. In a Clb2 knockout mutant, Cln2 and Clb5 increase during S phase, while an increase in Clb2 is not observed during G2 phase, indicating that cell cycle stops at G2 phase [37]. In a Clb5 knockout mutant, an increase in Clb2 is delayed at 300 min, indicating that the initiation of M phase [38].

Furthermore, to evaluate the validity of the estimated model, we examined if the model can predict the behaviors of the mutants that were not used in the optimization: Cln3/Bck2, Cdc14, Swi5, and Hct1 knockout mutants, as shown in Fig. 9. In a Cln3/Bck2 double knockout mutant, cell cycle stops in G1 phase [39]. In a Cdc14 knockout mutant, cell cycle does not exit from M phase [40]. In a Swi5 knockout mutant, the period of the cell cycle is slightly shortened compared with wild type (Fig. 8a) [41]. In a Hct1 knockout mutant, the exit from M phase is very delayed [42]. These simulated results capture experimentally observed behaviors, demonstrating the feasibility of the optimized model.

Robustness of optimized models to perturbations to scoring rules

It is important to know how robust the optimized models are with respect to some perturbations to the scoring rules. As additional experiments, we changed the threshold

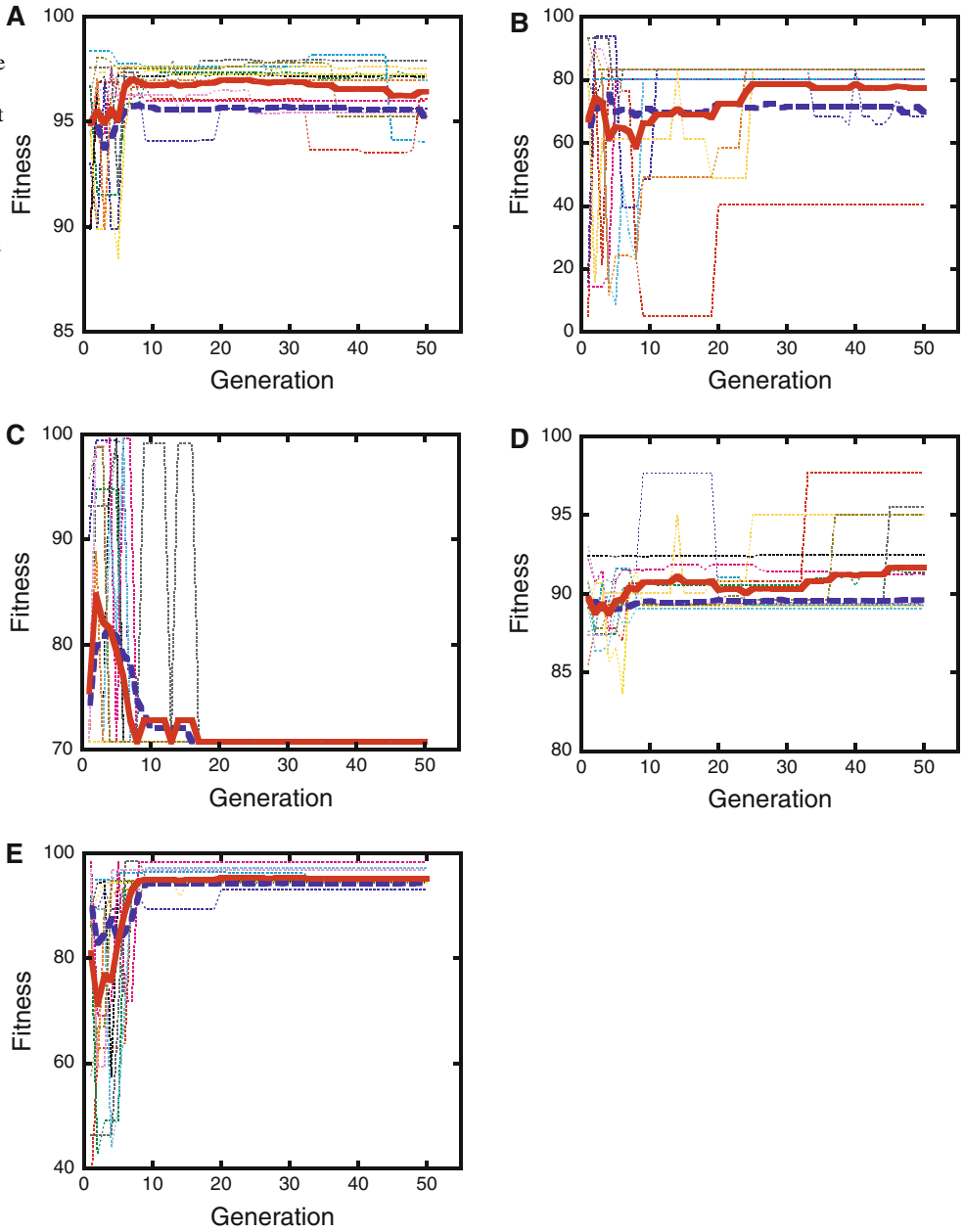

Fig. 7 Evolution of the fitness values for the full models of wild type and four mutants. Five objective functions for wild type (**a**), Sic1 knockout mutant (**b**), Cln2 knockout mutant (**c**), Clb2 knockout mutant (**d**), and Clb5 knockout mutant (**e**) are simultaneously optimized by using DCMOGA. The *thick solid lines* are the means of the simulated time courses by DCMOGA (*thin dotted lines*). The *thick dotted lines* are those by the normal MOGA (control method)

values and the values of the score points up to 50%, and then optimized the full model for wild type (the original scoring rules are shown in Supplementary Table S3A). As expected, the optimized models presented different values for numerical time course (Supplementary Figure S1). However, the qualitative behaviors, i.e., the tendency of an increase or decrease in the molecular concentrations of interest, are the same. This suggests that the results of optimization are robust with respect to a change in the scoring rules as far as the qualitative behaviors are concerned.

Conclusions

Practically useful optimization

Advances in molecular biology and omics technology produce a variety of biological data to construct biochemical network maps. Many biochemical reactions and gene regulations have rapidly been revealed, while kinetic data in vivo are extremely shortage due to experimental complexity. Therefore, a dynamic model is required to be optimized under the constraint that there are few quantitative data.

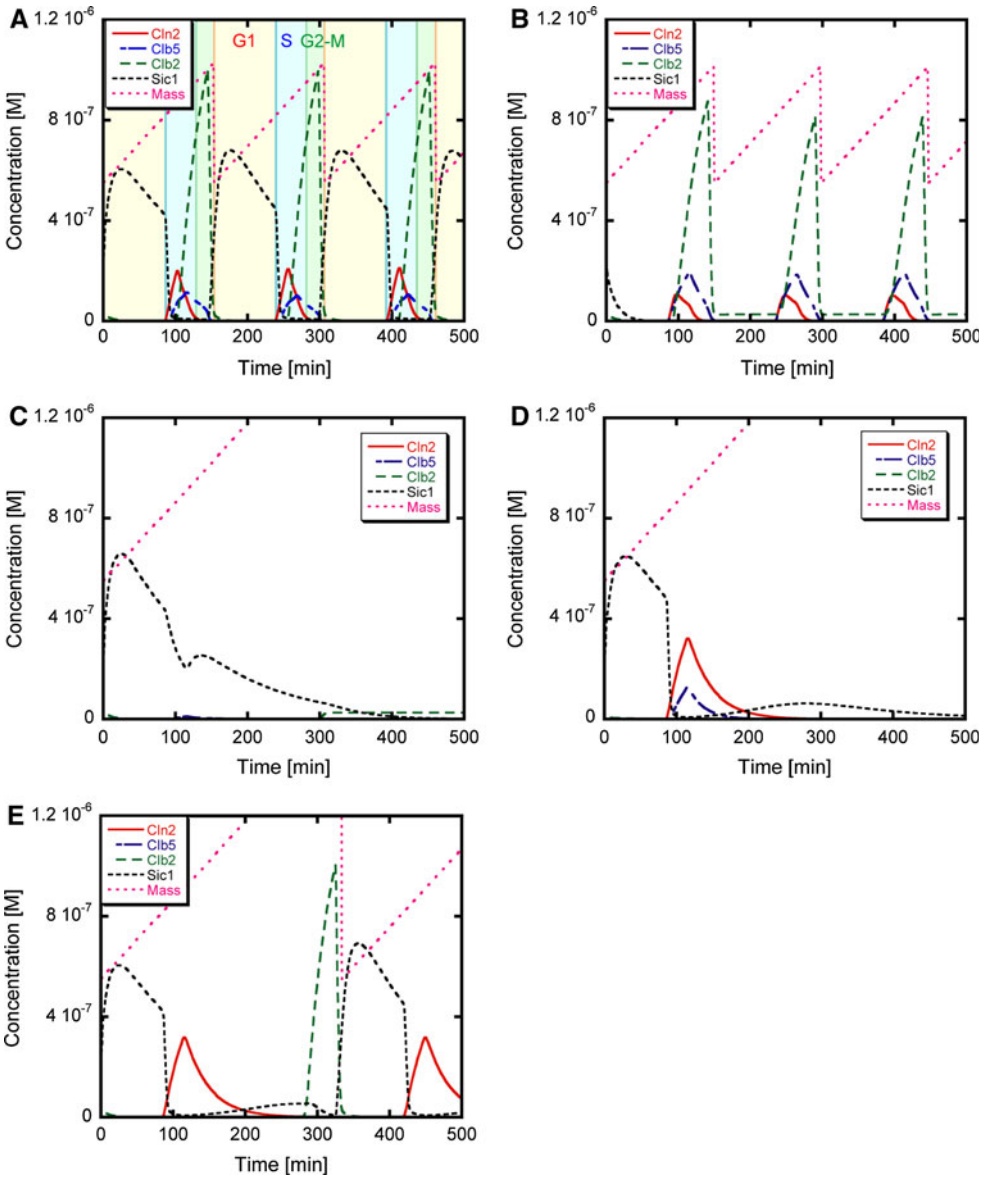

Fig. 8 Validation of the dynamic behaviors of wild type and gene knockout mutants. The time course of the major molecular components and events are plotted for wild type (**a**), Sic1 knockout mutant (**b**), Cln2 knockout mutant (**c**), Clb2 knockout mutant (**d**), and Clb5 knockout mutant (**e**)

Many elegant algorithms have been proposed for exploring a global solution for dynamic models that well reproduce biological behaviors [1, 2, 11–14]. They intensively develop a numerically or theoretically rigorous method by simplifying the optimization problem, e.g., by minimizing the square means of the difference between experimental data and the simulated time course data. However, the real problem is much more complicated, where we must handle qualitative and error-prone data for dynamic modeling [15, 16]. The important thing is not to pursue the global solution for the mathematically rigorous objective functions, but to build a dynamic model that qualitatively explains various experimental data and biological knowledge. This model can have many plausible solutions, but it is quite reasonable, considering the quality of given experimental data. This study would focus on providing such a qualitative or plausible dynamic model rather than on constructing the unique model with exact kinetic parameter values, allowing for a broader modeling paradigm [43, 44].

The proposed algorithm

An IPES is proposed to estimate the kinetic parameter values of a complex dynamic model by using qualitative and error-prone biological data. The key technologies are the divide and conquer method for reducing the search space, handling of multiple objective functions representing different types of biological behaviors, and design of the rule-based evaluation of fitness or objective functions. To demonstrate the feasibility of IPES, it is applied to an optimization problem of a yeast cell cycle model.

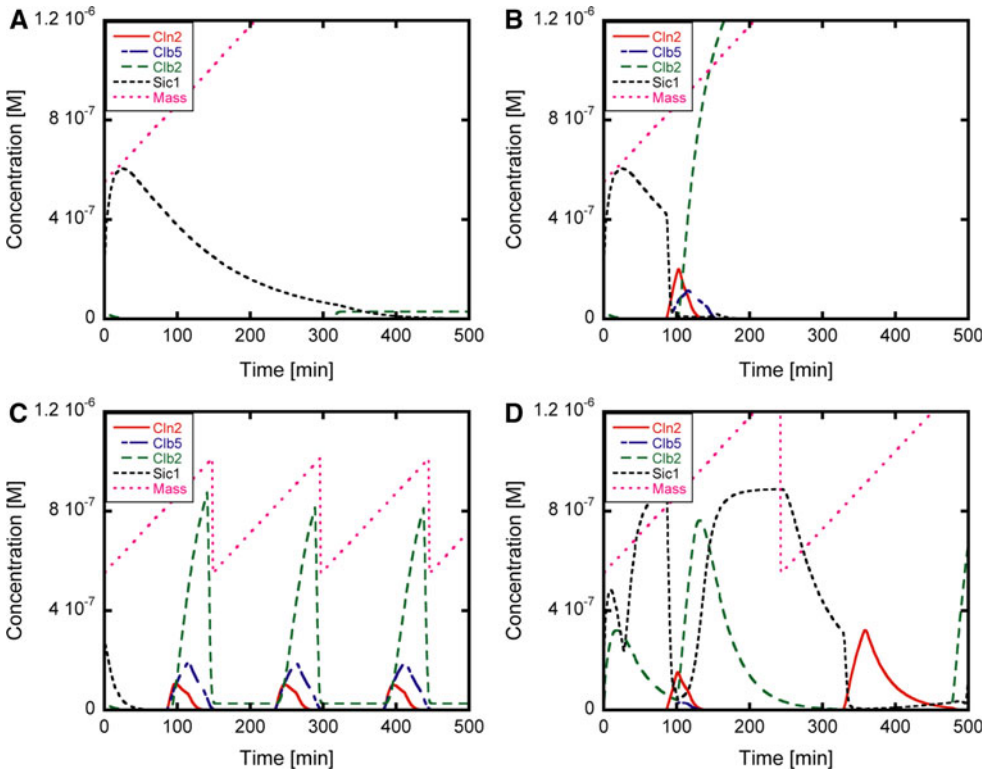

Fig. 9 Prediction of the dynamic behaviors of the knockout mutants that are not used for the optimization by DCMOGA. The time courses of the major molecular components and events are plotted for Cln3/Bck2 double knockout mutant (**a**), Cdc14 knockout mutant (**b**), Swi5 knockout mutant (**c**), and Hct1 knockout mutant (**d**)

First, to transform a network into a set of independent modules, the module decomposition and integration method is performed in terms of the temporal order of reactions and biological functions. To deal with the components being interacted from the external components belonging to the neighboring modules, specific time functions are assumed and assigned to the external ones so that the time functions well represent their biological behaviors. Then, coarsely optimized solutions are obtained for each module, and the resultant ones are merged to provide the solution candidates for the full model. These candidates are used as the initial population of the subsequent optimization for multiple objective functions.

Second, DCMOGA is employed to simultaneously optimize multiple dynamic behaviors of the cells cultured under different genetic an environmental conditions [29]. DCMOGA implements the Pareto front search, because the objective functions show a tradeoff relationship, i.e., an increase in the fitness value for an objective function causes a decrease in that for another function. An SGA island is assigned to a specific objective function and the MOGA island explores the Pareto-optimum solutions. DCMOGA is effective in finding the Pareto solutions while maintaining variations in solutions.

Third, the scoring rules are created for evaluating the degree of how the simulation time course of molecular and event components reflect biological knowledge and experimental data. "Trial and errors" are needed to determine appropriate points with respect to the evaluation of each dynamic feature of biological models. Thus, they are basically empirical, but practically useful for an intelligible description of the dynamic behaviors with few quantitative data. Further investigations are now performed to automatically or readily design the scoring rules from a variety of experimental data and biological knowledge.

Note that the IPES strategy can generate many plausible or local solutions, because the given constraints are evidently loose compared with the model size. To narrow the space of plausible solutions, more biological data or system-based criteria such robustness and stability may be added to objective functions.

Toward one click modeling

Dynamic modeling leads to an understanding of the mechanism of how biochemical networks generate particular cellular functions, but it is hard for ordinary biologists to construct complex dynamic models, because the modeling requires expert knowledge and mathematical techniques. A final goal for our study is to develop one-click modeling that enables any biologists to conveniently simulate dynamic models. If modeling is done by one-click, dynamic simulation for biochemical networks will be very popular. The one-click modeling requires the automatic generation of a dynamic model with tuned kinetic parameters without any manual operations, from a given

biochemical network map. While the automatic converter from a biochemical map to its associated mathematical equations has been presented [5, 16, 33], an automatic optimizer has not been proposed yet due to process complexity. The IPES method is the first and critical step for developing the standard technology for the automatic optimization of large-scale networks.

Acknowledgments This work was supported by Grant-in-Aid for Scientific Research on Priority Areas "Systems Genomics" and partially by Grant-in-Aid for Scientific Research (B) (22300101, 2010) from the Ministry of Education, Culture, Sports, Science and Technology of Japan. KM was supported by Research Fellowships from the Japan Society for the Promotion of Science for Young Scientists.

References

1. Mendes P, Kell D (1998) Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. Bioinformatics 14:869–883
2. Moles CG, Mendes P, Banga JR (2003) Parameter estimation in biochemical pathways: a comparison of global optimization methods. Genome Res 13:2467–2474
3. van Riel NA (2006) Dynamic modelling and analysis of biochemical networks: mechanism-based models and model-based experiments. Brief Bioinform 7:364–374
4. Banga JR (2008) Optimization in computational systems biology. BMC Syst Biol 2:47
5. Kurata H, Masaki K, Sumida Y et al (2005) CADLIVE dynamic simulator: direct link of biochemical networks to dynamic models. Genome Res 15:590–600
6. Zak DE, Gonye GE, Schwaber JS et al (2003) Importance of input perturbations and stochastic gene expression in the reverse engineering of genetic regulatory networks: insights from an identifiability analysis of an in silico network. Genome Res 13:2396–2405
7. Kremling A, Fischer S, Gadkar K et al (2004) A benchmark for methods in reverse engineering and model discrimination: problem formulation and solutions. Genome Res 14:1773–1785
8. D'Haeseleer P, Liang S, Somogyi R (2000) Genetic network inference: from co-expression clustering to reverse engineering. Bioinformatics 16:707–726
9. Vilela M, Chou IC, Vinga S et al (2008) Parameter optimization in S-system models. BMC Syst Biol 2:35
10. Tanaka S, Kurata H, Ohashi T (2006) Effective and fast optimization for a dynamic model of the Drosophila circadian oscillator. In: Proceedings of the IEEE international conference on systems, man, and cybernetics, pp 3596–3601
11. Rodriguez-Fernandez M, Mendes P, Banga JR (2006) A hybrid approach for efficient and robust parameter estimation in biochemical pathways. Biosystems 83:248–265
12. Rodriguez-Fernandez M, Egea JA, Banga JR (2006) Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems. BMC Bioinformatics 7:483
13. Kim CS (2007) Bayesian orthogonal least squares (BOLS) algorithm for reverse engineering of gene regulatory networks. BMC Bioinformatics 8:251
14. Balsa-Canto E, Peifer M, Banga JR et al (2008) Hybrid optimization method with general switching strategy for parameter estimation. BMC Syst Biol 2:26
15. Maeda K, Kurata H (2009) Two-phase search (TPS) method: nonbiased and high-speed parameter search for dynamic models of biochemical networks. IPSJ Trans Bioinformatics 2:2–14
16. Kurata H, Tanaka T, Ohnishi F (2007) Mathematical identification of critical reactions in the interlocked feedback model. PLoS One 2:e1103
17. Fonseca CM, Fleming PJ (1995) An overview of evolutionary algorithms in multiobjective optimization. Evol Comput 3:1–16
18. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans Evol Comput 3:257–271
19. Fleming PJ, Purshouse RC, Lygoe RJ (2005) Many-objective optimization: an engineering design perspective. Lect Notes Comput Sci 3410:14–32
20. Das I, Dennis JE (1998) Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems. SIAM J Optim 8:631–657
21. Handl J, Kell DB, Knowles J (2007) Multiobjective optimization in bioinformatics and computational biology. IEEE/ACM Trans Comput Biol Bioinform 4:279–292
22. Cormen TH, Leiserson CE, Rivest RL (2000) Introduction to algorithms. MIT Press, Boston
23. Ho SY, Hsieh CH, Yu FC et al (2007) An intelligent two-stage evolutionary algorithm for dynamic pathway identification from gene expression profiles. IEEE/ACM Trans Comput Biol Bioinform 4:648–660
24. Koh G, Teong HF, Clement MV et al (2006) A decompositional approach to parameter estimation in pathway modeling: a case study of the Akt and MAPK pathways and their crosstalk. Bioinformatics 22:e271–e280
25. Kimura S, Ide K, Kashihara A et al (2005) Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm. Bioinformatics 21:1154–1163
26. van Riel NA, Sontag ED (2006) Parameter estimation in models combining signal transduction and metabolic pathways: the dependent input approach. Syst Biol (Stevenage) 153:263–274
27. Tanaka S, Kurata H, Ohashi T (2004) Optimization of *E. coli* heat shock response parameter tuning using distributed and integrated genetic algorithms. In: Proceedings of the IEEE international conference on systems, man and cybernetics, pp 1243–1248
28. Kurata H, Taira K (2000) Two-phase partition method for simulating a biological system at an extremely high speed. Genome Inform Ser Workshop Genome Inform 11:185–195
29. Hiroyasu T, Miki M, Okuda T et al (2001) Distributed cooperation model of multi objective genetic algorithms. The Science and Engineering Review of Doshisha University, pp 129–140
30. Ono I, Kobayashi S (1997) A real-coded genetic algorithm for function optimization using unimordal normal distribution crossover. In: Proceedings of 7th international conference on genetic algorithms, pp 246–253
31. Okamoto M, Nonaka T, Ochiai S et al (1998) Nonlinear numerical optimization with use of a hybrid Genetic Algorithm incorporating the Modified Powell method. Appl Math Comput 91:63–72
32. Kurata H, Inoue K, Maeda K et al (2007) Extended CADLIVE: a novel graphical notation for design of biochemical network maps and computational pathway analysis. Nucleic Acids Res 35:e134
33. Kurata H, Matoba N, Shimizu N (2003) CADLIVE for constructing a large-scale biochemical network based on a simulation-directed notation and its application to yeast cell cycle. Nucleic Acids Res 31:4071–4084
34. Chen KC, Calzone L, Csikasz-Nagy A et al (2004) Integrative analysis of cell cycle control in budding yeast. Mol Biol Cell 15:3841–3862
35. Schneider BL, Yang QH, Futcher AB (1996) Linkage of replication to start by the Cdk inhibitor Sic1. Science 272:560–562

36. Richardson HE, Wittenberg C, Cross F et al (1989) An essential G1 function for cyclin-like proteins in yeast. Cell 59:1127–1133
37. Surana U, Robitsch H, Price C et al (1991) The role of CDC28 and cyclins during mitosis in the budding yeast *S. cerevisiae*. Cell 65:145–161
38. Schwob E, Nasmyth K (1993) CLB5 and CLB6, a new pair of B cyclins involved in DNA replication in *Saccharomyces cerevisiae*. Genes Dev 7:1160–1175
39. Di Como CJ, Chang H, Arndt KT (1995) Activation of CLN1 and CLN2 G1 cyclin gene expression by BCK2. Mol Cell Biol 15:1835–1846
40. Fitzpatrick PJ, Toyn JH, Millar JB et al (1998) DNA replication is completed in *Saccharomyces cerevisiae* cells that lack functional Cdc14, a dual-specificity protein phosphatase. Mol Gen Genet 258:437–441
41. Toyn JH, Johnson AL, Donovan JD et al (1997) The Swi5 transcription factor of *Saccharomyces cerevisiae* has a role in exit from mitosis through induction of the cdk-inhibitor Sic1 in telophase. Genetics 145:85–96
42. Schwab M, Lutum AS, Seufert W (1997) Yeast Hct1 is a regulator of Clb2 cyclin proteolysis. Cell 90:683–693
43. Schaub MA, Henzinger TA, Fisher J (2007) Qualitative networks: a symbolic approach to analyze biological signaling networks. BMC Syst Biol 1:4
44. Bosl WJ (2007) Systems biology by the rules: hybrid intelligent systems for pathway modeling and discovery. BMC Syst Biol 1:13