

libRCGA: 動力学モデルの高速なパラメータ推定のための遺伝的アルゴリズムライブラリ

前田 和勲^{1,2} Fred C. Boogerd³ 倉田 博之^{2,4}

概要: 生化学システムの動的な振る舞いを知るためには動力学モデルの構築が不可欠である。しかし、モデルを構築するには、モデルに含まれる動力学パラメータを推定する必要がある、この部分がモデル構築のボトルネックになっている。我々は、この問題を解決するために、遺伝的アルゴリズムのC言語ライブラリ libRCGA を開発した。このライブラリの特徴は、並列計算可能であること、確率的ランキングが使えることの2点である。本稿では、libRCGA の特徴を紹介するとともに、libRCGA が既存のライブラリよりも高い性能を持つことを示す。libRCGA は、<http://kurata21.bio.kyutech.ac.jp/maeda/index.html> あるいは GitHub で入手できる。

1. はじめに

動力学モデルの構築は、生化学システムの動的な振る舞いを解析するのに不可欠である [1]。解析に用いることができる現実的な動力学モデルを構築するには、モデルの挙動が実験データと一致するように動力学パラメータの値を推定する必要がある。これは逆問題の1つであり、パラメータ推定問題と呼ばれる [2]。現在でもパラメータ推定は動力学モデル構築のボトルネックとして残されている。

パラメータ推定には高い計算コストがかかる。動力学モデルは常微分方程式 (Ordinary Differential Equation: ODE) で表現されることが多い。モデルの挙動と実験データの一致性を定量化するためには、ODE を数値積分する必要がある、これには大きな計算コストがかかる [3]。また、パラメータ推定問題には強い非線形性があり、多数の局所解が存在する (多峰性である) ことが多い [2]。このため、遺伝的アルゴリズム (Genetic Algorithm: GA) のような集団に基づく確率的最適化アルゴリズムが適している。このようなアルゴリズムは、一般に大きな計算コストがかかる。従って、パラメータ推定には高性能な計算環境が不可欠である。

動力学モデルが現実的であるとみなされるにはパラメータ推定の実験データ以外にも基本的な生物学的な制約を満

たす必要がある。例えば、生物ではほとんどの代謝物濃度は 10 mM 以下であるし、分子の結合解離反応は酵素反応よりも遥かに速い。制約付き最適化問題の定式化を使うことで、このような生物学的知見をパラメータ推定問題に組み込むことができる [4]。しかし、多くの最適化アルゴリズムは制約なし最適化問題を念頭に設計されている。

動力学モデルのパラメータ推定を高速化するために、我々は、実数値遺伝的アルゴリズム (Real-coded genetic algorithm: RCGA) の C 言語ライブラリ libRCGA を開発した [5]。libRCGA では、Unimodal Normal Distribution Crossover/Minimal Generation Gap (UNDX/MGG) と Real-coded Ensemble Crossover star/Just Generation Gap (REX^{star}/JGG) の2つが Message Passing Interface (MPI) を使って並列実装されている。このため、libRCGA を用いることで高性能計算環境上で高速なパラメータ推定を実現できる。また、確率的ランキングを使うことで制約付き最適化問題も効率的に解くことができる。本稿では、libRCGA の特徴を紹介するとともに、libRCGA が既存のライブラリよりも高い性能を持つことを示す。

2. 方法

2.1 パラメータ推定問題

パラメータ推定問題は、次の制約付き最適化問題として扱うことができる。

$$\text{minimize } f(\mathbf{x}), \quad (1)$$

$$\text{subject to } \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \quad (2)$$

$$\mathbf{x}^{lb} \leq \mathbf{x} \leq \mathbf{x}^{ub} \quad (3)$$

¹ 九州工業大学 若手研究者フロンティア研究アカデミー

² 九州工業大学 情報工学部 生命情報工学科

³ Department of Molecular Cell Biology, Faculty of Science, VU University Amsterdam, the Netherlands

⁴ 九州工業大学 バイオメディカルインフォマティクス研究開発センター

ここで, $\mathbf{x} = (x_1, x_2, \dots)$ は設計変数ベクトル, f は目的関数, $\mathbf{g} = (g_1, g_2, \dots)$ は制約関数ベクトルである. $g_i > 0$ のとき, i 番目の制約が満たされていないことを示す. $\mathbf{x}^{lb}$ と $\mathbf{x}^{ub}$ は, それぞれ設計変数の下限と上限を決めるベクトルであり, これらによって探索空間が決まる. 式 2 を満たす空間は可能領域 (feasible region) と呼ばれる. 制約なし問題では式 2 は省略される.

2.2 遺伝的アルゴリズム

遺伝的アルゴリズム (GA) は, 生物の進化を模倣した最適化アルゴリズムの 1 つである [6]. 基本的な手順は次の通りである.

- (1) 初期集団を生成する. ここで, 集団内の各個体は設計変数ベクトルである.
- (2) 集団内からいくつかの個体を選び出す. 選び出された個体は親個体と呼ばれる.
- (3) 親個体をもとにして子個体を生成する.
- (4) 良い子個体を選択する. 制約付き最適化においては, 良い個体とは, 小さい目的関数値と制約関数値を与える個体のことを指す.
- (5) 選択した子個体で集団内の親個体を置き換える.
- (6) 終了条件が満たされたら終了する. そうでなければ, ステップ (2) に戻る.

多くの場合, 十分に良い個体が得られた場合や, 事前に決められた最大世代数に達した場合に GA は終了する. 動力学モデルのパラメータ推定では, RCGA が用いられる. この場合, それぞれの個体は 1 組の動力学パラメータを表す実数値ベクトルとなる. そして, ステップ (4) における良い個体とは, 実験データとの良い一致を与える動力学パラメータ値のベクトルである.

2.3 libRCGA

パラメータ推定問題は計算コストがかかるので, スーパーコンピュータのようなマルチコア-マルチノード計算環境の計算性能を引き出すことが重要である. これを達成するために, 我々は libRCGA を開発した [5]. libRCGA を用いることで, C 言語や並列計算, RCGA の深い知識がなくても高性能計算環境で高速なパラメータ推定を行うことができる. libRCGA の全体像を図 1 に示す. libRCGA は現在, UNDX/MGG と REX^{star}/JGG という 2 つの RCGA を実装している. 制約付き最適化問題のために, 確率的ランキングを使用できる. RCGA は, 逐次あるいは並列プログラムとして実行できる. libRCGA は MPI 以外に外部のソフトウェアやライブラリを必要としないので, Linux/UNIX だけでなく Windows や macOS でも利用できる. また, スーパーコンピュータに限らず多くのコンピュータで利用できる.

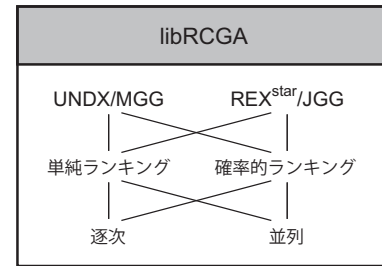


図 1 libRCGA の全体像.

2.3.1 UNDX/MGG

UNDX/MGG は, 交叉法に UNDX[7], 世代交代法に MGG[8] を使う RCGA である. UNDX/MGG は, 動力学モデルの構築で長く使われてきた [9], [10], [11], [12], [13], [14]. UNDX は, 親個体の性質を子個体にうまく伝えることができる. MGG は, 緩やかな世代交代によって探索が局所解に陥ることを防ぐ. UNDX/MGG は, 局所解に陥りにくいが探索が遅い.

2.3.2 REX^{star}/JGG

REX^{star}/JGG は, 交叉法に REX^{star}, 世代交代法に JGG を使う RCGA である [15]. REX^{star}/JGG は, 最近では代謝モデルのパラメータ推定 [16] や遺伝子ネットワークの推定 [17] に用いられている. REX^{star} は, 親個体の情報をもとにして, より良い個体を得られそうな領域に子個体を生成する. JGG は, 子個体の生成に使われた親個体全てを子個体で置き換える. このため, REX^{star}/JGG は素早い探索が行える.

2.3.3 制約付き最適化問題の取り扱い

UNDX/MGG と REX^{star}/JGG は, 制約なし最適化問題を解くためのアルゴリズムである. ここでは, UNDX/MGG と REX^{star}/JGG を使って制約付き最適化問題を扱う方法について説明する. 制約が満たされているか調べるために, 一般にペナルティ関数 ϕ が用いられる.

$$\phi(\mathbf{x}) = \sum_{i=1}^n [\max\{0, g_i(\mathbf{x})\}]^2 \quad (4)$$

ここで, $\mathbf{x}$ は設計変数ベクトルであり, g_i は i 番目の制約関数, n は制約関数の数である. i 番目の制約が満たされているとき $g_i \leq 0$ となる. 従って, $\phi = 0$ のとき全ての制約が満たされていることを示す. 制約付き最適化問題では, この ϕ と目的関数 f (式 1 参照) を同時に小さくする必要がある. 問題は, この 2 つの指標を使ってどのように個体をランキングするかである.

最も単純なランキング方法は次のようなものである. 次の 2 つの条件のうち 1 つでも満たせば, 個体 $\mathbf{x}_1$ が個体 $\mathbf{x}_2$ よりも良い個体とみなされる [18].

$$\phi(\mathbf{x}_1) < \phi(\mathbf{x}_2), \quad (5)$$

$$\phi(\mathbf{x}_1) = \phi(\mathbf{x}_2) \quad \text{AND} \quad f(\mathbf{x}_1) < f(\mathbf{x}_2) \quad (6)$$

このランキング方法では, 常に ϕ が f よりも重要とみなさ

表 1 制約なしベンチマーク問題 [15] における探索成功回数. それぞれの最適化アルゴリズムを各問題に対して 100 回適用した. 改良 (μ, λ) -ES には libSRES, UNDX/MGG と REX^{star} /JGG には libRCGA を用いた.

	設計変数の数	悪スケール性	変数間依存性	多峰性	探索成功回数		
					改良 (μ, λ) -ES	UNDX/MGG	REX^{star} /JGG
Sphere	100	なし	なし	なし	100	100	100
Scaled Sphere	100	あり	なし	なし	100	0	100
Ellipsoid	100	あり	なし	なし	100	0	100
Cigar	100	あり	なし	なし	100	0	100
k -tablet	100	あり	なし	なし	100	0	100
MM Benchmark	100	あり	あり	なし	100	0	100
Rosenbrock (star)	50	なし	あり	なし	0	100	91
Rosenbrock (chain)	100	なし	あり	なし	87	0	100
Ackley	50	なし	あり	あり	13	100	100
Bohachevsky	50	なし	なし	あり	0	92	100
Rastrigin	20	なし	なし	あり	0	73	87
Schaffer	20	なし	あり	あり	0	0	100
Schwefel	10	なし	なし	あり	21	77	54

れている. 本稿では, このランキング法を単純ランキングと呼ぶことにする. この方法の問題点は, 探索初期において $\phi > 0$ の個体が集団内から淘汰されてしまい, 探索空間の限定された領域しか探索されなくなることである [19].

確率的ランキングは f と ϕ の重みのバランスをとることで単純ランキングの問題を解決する [19]. 確率的ランキングでは, 2つの個体の比較は f と ϕ どちらかに基づいて行われるが, 実際にどちらに基づくかは比較ごとにランダムに決められる. f が使われる確率 (P_f) が ϕ が使われる確率よりも若干低くなるように設定した場合, 確率的ランキングは効果的に働くことが知られている. 確率的ランキングは, バブルソートに似たアルゴリズムによって実装できる [19].

2.4 libSRES (既存のライブラリ)

本稿では, libRCGA の性能を評価するために libSRES との比較を行う. libSRES は, Ji らによって開発された確率的ランキング進化戦略 (Stochastic Ranking Evolution Strategy: SRES) の C 言語ライブラリである [20]. SRES は, 改良 (μ, λ) -進化戦略 (ES) と確率的ランキングを組み合わせたものである. 改良 (μ, λ) -ES では, 交叉ではなく, 親個体の変異によって子個体が生成される. これまでに SRES が動力学モデルのパラメータ推定において高い性能を示すことが報告されている [21]. このため, いくつかのソフトウェアが libSRES をパラメータ推定エンジンとして採用している [22], [23], [24], [25].

3. 結果と考察

本稿で示す全ての計算機実験には, スーパーコンピュータ Shirokane3 の Intel Xeon E5-2670 v3 (2.3 GHz) を用いた.

3.1 ベンチマーク問題

libRCGA の性能を評価するために, 表 1 の 13 個の制約なしベンチマーク問題に対して libRCGA の UNDX/MGG と REX^{star} /JGG を適用した. $f \leq 10^{-6}$ となる設計変数ベクトルが得られた場合に探索が成功したとみなした. 比較対象として libSRES の改良 (μ, λ) -ES もベンチマーク問題に適用した.

表 1 に示すとおり, 改良 (μ, λ) -ES は, Rosenbrock (star) と多峰性の問題 (Bohachevsky, Rastrigin, Schaffer) において全ての試行で探索に失敗した. 一方, UNDX/MGG は悪スケールの問題 (Scaled Sphere, Ellipsoid, Cigar, k -tablet, MM benchmark) と Rosenbrock (chain), Schaffer において, 全ての試行で探索に失敗した. これに対して, REX^{star} /JGG は全ての問題で探索に成功した.

次に, 表 2 の 13 個の制約付きベンチマーク問題に対して改良 (μ, λ) -ES, UNDX/MGG, REX^{star} /JGG を適用した. $\phi = 0$ かつ f が既知の最小値から 1% 以内となる設計変数ベクトルが得られた場合に探索が成功したとみなした.

改良 (μ, λ) -ES と確率的ランキングを組み合わせた場合, 全ての問題に対して探索に成功した. これに対して, UNDX/MGG と REX^{star} /JGG では, 確率的ランキングを組み合わせた場合でも少なくとも 1 つの問題で探索に失敗した.

以上の結果をまとめると, 制約なしベンチマーク問題において, REX^{star} /JGG は改良 (μ, λ) -ES や UNDX/MGG よりも優れている. 一方, 制約付きベンチマーク問題に関しては, 改良 (μ, λ) -ES が最も優れている.

3.2 パラメータ推定問題

次に, 実際のパラメータ推定問題に対する libRCGA の性能を評価するために, Three-step モデルと HIV モデルとい

表 2 制約付きベンチマーク問題 [19], [20] における探索成功回数. それぞれの最適化アルゴリズムを各問題に対して 100 回適用した. 改良 (μ, λ) -ES には libSRES, UNDX/MGG と REX^{star}/JGG には libRCGA を用いた. "単純"は単純ランキング, "確率的"は確率的ランキング ($P_f = 0.45$) を表す. 確率的ランキングを組み合わせた改良 (μ, λ) -ES は SRES として知られている.

	設計変数の数	制約数	探索成功回数					
			改良 (μ, λ) -ES		UNDX/MGG		REX ^{star} /JGG	
			単純	確率的	単純	確率的	単純	確率的
ex1	13	9	100	99	8	2	99	99
ex2	20	2	12	13	1	0	4	5
ex3	10	1	93	100	100	100	0	100
ex4	5	6	100	100	100	100	100	100
ex5	4	5	39	100	100	100	21	0
ex6	2	2	100	100	100	100	100	100
ex7	10	8	100	100	100	100	100	100
ex8	2	2	100	100	100	100	100	100
ex9	7	4	100	100	100	100	100	100
ex10	8	6	100	100	93	0	100	43
ex11	2	1	100	100	100	100	100	100
ex12	3	1	100	100	100	100	100	100
ex13	5	3	0	84	0	100	0	11

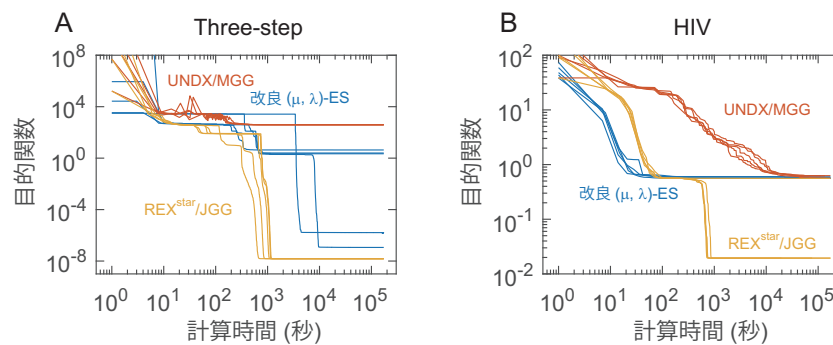


図 2 目的関数 f の収束曲線. (A) Three-step モデル, (B) HIV モデル. 試行回数はアルゴリズムごとに 5 回である. ランキングには確率的ランキング ($P_f = 0.45$) を用いた. 従って, ここで示した改良 (μ, λ) -ES は, SRES として知られるものと同じである.

う 2 つの動力学モデルのパラメータ推定を行った. この 2 つのモデルは, 最適化アルゴリズムの性能評価のために広く用いられているものである [20], [21], [26], [27], [28], [29]. 目的関数 f は, モデル予測と実験データの自乗誤差の総和である.

Three-step モデルは, 3 つの酵素反応からなる直鎖状代謝経路の動力学モデルである. このモデルは 8 つの ODE, 36 の未知動力学パラメータを持つ. パラメータ推定の目的は, 16 個の時系列実験データをモデルが再現できるようにすることである. 本来 Three-step モデルのパラメータ推定は制約なし最適化問題だが, 本稿では, 3 つの酵素が似た性質を持つことが既知であると仮定し, 24 個の制約を持つ制約付き最適化問題とした.

HIV モデルは, HIV プロテイナーゼ反応とその阻害の動力学モデルである. このモデルは 9 つの ODE, 20 の未知パラメータを持つ. パラメータ推定の目的は, 5 つ時系列

データをモデルが再現できるようにすることである. 本来 HIV モデルのパラメータ推定は制約なし最適化問題だが, 本稿では, 阻害剤が基質よりも強く酵素に結合するといった一般的な生物学的知識を制約として組み込むことで, 12 の制約を持つ制約付き最適化問題とした.

改良 (μ, λ) -ES, UNDX/MGG, REX^{star}/JGG を Three-step 問題と HIV 問題に適用した. 計算機実験は Shirokane3 上で行い, 31 個の計算コアを使用した. Three-step 問題において, REX^{star}/JGG は $f < 10^{-3}$ を 17 分以内で達成した (図 2A). 一方, 改良 (μ, λ) -ES は, $f < 10^{-3}$ を達成するのに少なくとも 63 分かかった. また, UNDX/MGG では $f = 397$ が最小であった. HIV 問題において, REX^{star}/JGG は, $f \approx 0.02$ を 15 分以内で達成できた (図 2B). 一方, 改良 (μ, λ) -ES と UNDX/MGG で得られた最小の f はそれぞれ 0.559 と 0.562 であった.

以上をまとめると, REX^{star}/JGG は, パラメータ推定

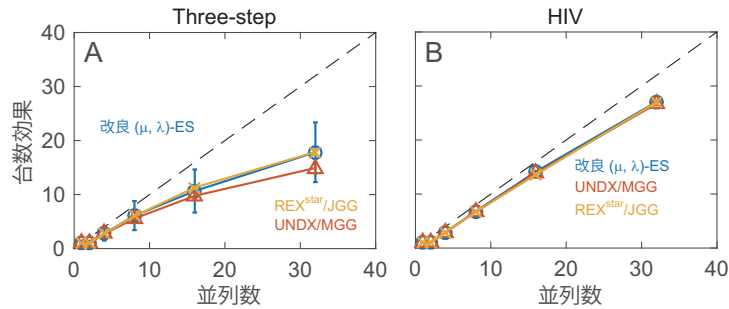


図3 並列数と台数効果. (A) Three-step モデル, (B) HIV モデル. 試行回数はアルゴリズムごとに5回である. シンボルは平均値, エラーバーは±標準偏差である. ランキングには確率的ランキング ($P_f = 0.45$) を用いた. 従って, ここで示した改良 (μ, λ) -ES は, SRES として知られるものと同じである.

問題において改良 (μ, λ) -ES と UNDX/MGG よりも優れている.

3.3 並列化による計算速度向上の評価

一般に, 逐次プログラムを X 個のプロセスに並列化した場合, 計算速度の向上は X 倍よりも低くなる. これはプログラム中に並列化できない部分があったり, 並列化数が増えるほどプロセス間の通信が増えたりするためである. そこで, 並列化によって libRCGA の UNDX/MGG と REXstar/JGG の計算速度が実際にどれほど向上しているのか調べるための計算機実験を行った. 具体的には, 上記の Three-step 問題と HIV 問題における計算時間をもとにして台数効果を計算した. 台数効果は次の式で計算できる.

$$\begin{aligned} \text{台数効果} &= \frac{\text{並列プログラムの計算速度}}{\text{逐次プログラムの計算速度}} \\ &= \frac{\text{逐次プログラムの計算時間}}{\text{並列プログラムの計算時間}} \end{aligned} \quad (7)$$

台数効果 = X の場合, 逐次プログラムに比較して並列プログラムの計算速度が X 倍であることを示す.

Three-step 問題と HIV 問題の両方において並列数が増えるほど台数効果が向上した (図3). Three-step モデルにおいて, 32 並列の場合の台数効果は, 改良 (μ, λ) -ES で 18 ± 5.5 , UNDX/MGG で 15 ± 1.0 , REXstar/JGG で 18 ± 0.5 であった (平均 ± 標準偏差). HIV 問題では, 改良 (μ, λ) -ES, UNDX/MGG, REXstar/JGG の全てにおいて, 32 並列の場合の台数効果は 27 ± 0.3 であった.

Three-step モデルで台数効果が低い理由は次のように説明できる. libRCGA の RCGA はマスターワーカーモデルを使って並列化されている. 並列数が n の場合, m 個の個体の評価 (目的関数 f と制約関数 g の計算) を $n-1$ 個のワーカープロセスで分担する. 各ワーカーは, それぞれ $m/(n-1)$ 個の個体を評価するので, 各ワーカーに配分される計算コストは一見すると等しいように思える. しかし, パラメータ推定問題において, 個体の評価には ODE の数値積分が含まれる. このため, 同じ数の個体が与えられても

実際の計算コストはワーカーごとに異なる. 計算の終わっていないワーカーを待つ時間は, 計算の終わったワーカーにとっては空き時間になってしまう. 特に, Three-step モデルは動力学パラメータの探索範囲が広く, 動力学パラメータ値によって数値積分にかかる計算コストに大きなばらつきがある. このため, Three-step モデルでは台数効果が向上しにくい. ワーカーにタスクを一括して渡すのではなく, 計算の終わったワーカーに優先的にタスクを割り振ることで台数効果がさらに改善できるかもしれない.

4. おわりに

本稿では, C 言語ライブラリ libRCGA を紹介し, その性能評価を行った. libRCGA を用いることで, 高性能計算環境において高速なパラメータ推定が行える. また, 確率的ランキングを使うことで制約付き最適化問題も効率的に解くことができる.

我々は, 動力学モデルのパラメータ推定を高速化する目的で libRCGA を開発したが, libRCGA は汎用的な RCGA のライブラリであるので, パラメータ推定以外の最適化問題にも適用できる.

参考文献

- [1] Kitano H (2002) Systems biology: a brief overview. Science 295, 1662-1664.
- [2] Banga JR (2008) Optimization in computational systems biology. BMC Syst Biol 2, 47.
- [3] Shiraishi E, Maeda K, Kurata H (2009) A gradual update method for simulating the steady-state solution of stiff differential equations in metabolic circuits. Bioprocess Biosyst Eng 32, 283-288.
- [4] Lee WP, Hsiao YT (2012) Inferring gene regulatory networks using a hybrid GA-PSO approach with numerical constraints and network decomposition. Inform Sciences 188, 80-99.
- [5] Maeda K, Boogerd FC, Kurata H (2018) libRCGA: a C library for real-coded genetic algorithms for rapid parameter estimation of kinetic models. IPSJ Transactions on Bioinformatics
- [6] Sun JY, Garibaldi JM, Hodgman C (2012) Parameter

- Estimation Using Metaheuristics in Systems Biology: A Comprehensive Review. *IEEE/ACM Trans Comput Biol Bioinform* 9, 185-202.
- [7] Ono I, Kobayashi S (1997) A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover. *Proc of 7th Int Conf on Genetic Algorithms*, 246-253.
- [8] Satoh H, Yamamura M, Kobayashi S (1997) Minimal generation gap model for GAs considering both exploration and exploitation. *Proc of Int Conf on Fuzzy Logic, Neural Networks and Soft Computing*, 494-497.
- [9] Maeda K, Minamida H, Yoshida K, Kurata H (2013) Flux module decomposition for parameter estimation in a multiple-feedback loop model of biochemical networks. *Bioprocess Biosyst Eng* 36, 333-344.
- [10] Yoshimura J, Shimonobou T, Sekiguchi T, Okamoto M (2003) Development of the parameter-fitting module for web-based biochemical reaction simulator BEST-KIT. *Chem-Bio Informatics Journal* 3, 114-129.
- [11] Masaki K, Maeda K, Kurata H (2012) Biological Design Principles of Complex Feedback Modules in the *E. coli* Ammonia Assimilation System. *Artif Life* 18, 53-90.
- [12] Ueda T, Ono I, Okamoto M (2002) Development of System Identification Technique Based on Real-Coded Genetic Algorithm. *Genome Informatics* 13, 386-387.
- [13] Maeda K, Fukano Y, Yamamichi S, Nitta D, Kurata H (2011) An integrative and practical evolutionary optimization for a complex, dynamic model of biological networks. *Bioprocess Biosyst Eng* 34, 433-446.
- [14] Jahan N, Maeda K, Matsuoka Y, Sugimoto Y, Kurata H (2016) Development of an accurate kinetic model for the central carbon metabolism of *Escherichia coli*. *Microb Cell Fact* 15, 112.
- [15] Kobayashi S (2009) The Frontiers of Real-coded Genetic Algorithms. *J Jpn Soc Artif Intell* 24, 147-162.
- [16] Tohsato Y, Ikuta K, Shionoya A, Mazaki Y, Ito M (2013) Parameter optimization and sensitivity analysis for large kinetic models using a real-coded genetic algorithm. *Gene* 518, 84-90.
- [17] Kimura S, Sato M, Okada-Hatakeyama M (2015) An Effective Method for the Inference of Reduced S-system Models of Genetic Networks. *Inform Media Tech* 10, 166-174.
- [18] Mezura-Montes E, Coello CAC (2011) Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation* 1, 173-194.
- [19] Runarsson TP, Yao X (2000) Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation* 4, 284-294.
- [20] Ji X, Xu Y (2006) libSRES: a C library for stochastic ranking evolution strategy for parameter estimation. *Bioinformatics* 22, 124-126.
- [21] Moles CG, Mendes P, Banga JR (2003) Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome Res* 13, 2467-2474.
- [22] Zi Z (2011) SBML-PET-MPI: a parallel parameter estimation tool for Systems Biology Markup Language based models. *Bioinformatics* 27, 1028-1029.
- [23] Hoops S, Sahle S, Gauges R, Lee C, Pahle J, Simus N, Singhal M, Xu L, Mendes P, Kummer U (2006) COPASI- a COmplex PATHway SIMulator. *Bioinformatics* 22, 3067-3074.
- [24] Schmidt H, Jirstrand M (2006) Systems Biology Toolbox for MATLAB: a computational platform for research in systems biology. *Bioinformatics* 22, 514-515.
- [25] Hirmajer T, Balsa-Canto E, Banga JR (2009) DOTcvpSB, a software toolbox for dynamic optimization in systems biology. *BMC Bioinformatics* 10, 199.
- [26] Mendes P (2001) Modeling Large Biological Systems From Functional Genomic Data: Parameter Estimation. In: Kitano H, (ed) *Foundations of Systems Biology* MIT Press, Cambridge, MA.
- [27] Miro A, Pozo C, Guillen-Gosalbez G, Egea JA, Jimenez L (2012) Deterministic global optimization algorithm based on outer approximation for the parameter estimation of nonlinear dynamic biological systems. *BMC Bioinformatics* 13, 90.
- [28] Rodriguez-Fernandez M, Egea JA, Banga JR (2006) Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems. *BMC Bioinformatics* 7, 483.
- [29] Mendes P, Kell D (1998) Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics* 14, 869-883.