

生物シミュレーションに対する
目的関数を二つ持つ遺伝的アルゴリズム

田中 信

目次

1. 序論	1
2. 生物シミュレーションの最適化問題	4
2. 1. 対象問題：大腸菌の熱ショック応答	6
2. 2. 遺伝的アルゴリズムを用いた大腸菌熱ショック応答の最適化	8
2. 2. 1. 遺伝的アルゴリズム	10
2. 2. 2. 実数値遺伝的アルゴリズム (Real-coded GA)	12
2. 2. 3. 交叉	13
2. 3. 大腸菌熱ショック応答における，これまでの最適化計算と問題点	15
3. 2つの目的関数をもつ最適化問題	16
3. 1. 大腸菌熱ショック応答における2つの目的関数	17
3. 2. 2つの目的関数を対象とする遺伝的アルゴリズム	19
3. 2. 1. 重み付き線形結合 GA	19
3. 2. 2. 分散 GA	20
3. 2. 3. 分散統合 GA	21
3. 2. 4. 分散協力 GA	23
3. 3. 既存手法を用いた場合での大腸菌熱ショック応答の最適化	25
3. 3. 1. 線形結合 GA を用いた場合の結果	26
3. 3. 2. 分散 GA を用いた場合の結果(個体群ごとに目的関数が異なる場合)	28
3. 3. 3. 分散 GA を用いた場合の結果(重み関数を共通の目的関数とする場合)	30
3. 3. 4. 分散統合 GA を用いた場合の結果	33
3. 3. 5. 分散協力 GA を用いた場合の結果	36
3. 4. 従来手法を用いた最適化計算の結果と問題点	39

4. 重み関数に依存しない2つの目的関数を対象とした最適化手法の提案：生存率 GA...	40
4. 1. 生存率 GA: Survival Ratio GA.....	41
4. 2. 2つの目的関数最適化問題に対応する GA.....	43
4. 2. 1. 生存率 GA.....	43
4. 2. 2. 固定寿命 GA.....	45
4. 2. 3. 個体の寿命を考慮した GA.....	46
4. 3. 寿命を導入した GA での大腸菌熱ショック応答の最適化.....	47
4. 3. 1. 固定寿命 GA を用いた場合.....	47
4. 3. 2. 個体の寿命を考慮した GA を用いた場合の結果.....	49
4. 3. 3. 生存率 GA を用いた場合の結果.....	51
4. 4. 生存率 GA の生存率パラメータに関する考察.....	53
4. 5. 既存手法との比較.....	55
4. 5. 1. 最適化に要した平均計算回数と成功確率.....	55
4. 5. 2. エリート個体の分布推移.....	57
4. 5. 3. 個体分布図.....	58
4. 5. 4. 各最適化手法における平均計算回数と成功確率の関係.....	62
4. 6. 単位の異なる2つの評価基準を持つ大腸菌熱ショック応答の最適化.....	64
4. 6. 1. 単位の異なる2つの評価基準を持つ大腸菌熱ショック応答シミュレーション	65
4. 6. 2. 単位の異なる2つの目的関数最適化を各 GA 手法で解いた結果.....	67
4. 7. 生存率 GA による熱ショック応答の最適化計算効率の向上.....	69
5. 2つの評価基準を持つショウジョウバエ概日リズムシミュレーションの最適化.....	70
5. 1. ショウジョウバエの概日リズムシミュレーション.....	71
5. 1. 1. 概日リズムにおける2つの評価基準.....	73
5. 1. 2. 概日リズムの最適化問題と熱ショック応答との比較.....	75
5. 2. 概日リズムにおける最適化実験の結果.....	76

5. 3. 概日リズムシミュレーションの広域探索.....	79
5. 3. 1. 1 段階の生存率 GA を用いた概日リズムシミュレーションの広域探索	80
5. 3. 2. 概日リズムの広域探索における 2 段階最適化と 1 段階生存率 GA での結果	81
5. 3. 3. 最適化計算によって発見された局所解の個数について.....	85
5. 3. 4. 1 段階探索と 2 段階探索における探索経路	90
5. 4. 生存率 GA のみの 1 段階探索による概日リズムの最適化効率の向上.....	92
6. 大腸菌の窒素同化システムでの最適化計算.....	94
6. 1. 大腸菌の窒素同化システムとモジュール分割.....	95
6. 2. モジュール分割を利用した窒素同化システムの最適化.....	97
6. 3. 生存率 GA を用いた窒素同化のモジュール分割法での最適化結果と性能の向上.	99
7. 結論.....	102
8. 謝辞.....	106
9. 参考文献.....	107
10. 付録.....	110
単一目的関数最適化問題における生存率 GA	110

1. 序論

近年、バイオインフォマティクスの発展により、大規模な生物シミュレーションを扱う研究が盛んに行われるようになってきた。この分野の研究では、細胞の生命分子ネットワークを微分代数方程式で記述し、シミュレーションすることによって、生命現象の原理を追求している。しかし、計算機上に細胞のダイナミクスを再現することは容易ではない。なぜなら、反応に関わる動力学パラメータは実験などの直接的な測定が困難だからである。そこで、さまざまな学術的な生物実験の結果に基づいて、動力学パラメータ・経路を推定することにより、実際の生命のダイナミクスと同じ挙動を示すモデルを探索すること(REVERSE ENGINEERING)が必要である。

しかし、生物シミュレーションでは一般に推定すべき動力学パラメータの数が多く、また、それらが相互に強い依存関係を持つため、従来の解析的な最適化手法は適用できない。そこで、確率的な最適化手法である遺伝的アルゴリズム(Genetic Algorithms, GA)を用いて最適化計算を行うことが多い。しかし、GAによる最適化では、シミュレーションの実行を繰り返す反復計算を行うため、計算に膨大な時間がかかる。また、従来手法のGAを用いた場合、最適化の成功確率は低く、最適解を得るためには複数回の試行が必要である。さらに、生物のダイナミクスを検証する際には、生物を取り巻く環境条件をさまざまに変化させても、実際の挙動と同じ結果を示すかを検証しなければならず、この最適化計算が研究全体のボトルネックとなる。

最適化計算に時間がかかり、成功確率が低くなる原因の1つは、探索空間の形状が複雑(多峰性で多くの局所解を持つ)であり、GAでの探索が十分行われる前に局所解に収束してしまうことである。そこで、本論文では、最適化問題を2つの評価基準で定義し、2つの探索軸を利用して、広い探索空間の中から解の絞込みを行う。一方、従来のGAは1つの評価基準に対し最適化を行う手法であり、2つの評価基準を同時に満たす探索は出来ない。この問題に対し、2つ以上の多目的関数の最適化にも対応できる手法も提案されているが、これらを利用して十分な最適化性能を得る

ことは出来なかった。

そこで、2 つの最適化基準を持つ問題をより効率的に解くために、生存率という概念を導入した生存率 GA(Survival Ratio GA)を定義し、最適化計算を行った。生存率を GA に導入する事により、2 つの評価基準を共に満たす解を効率的に探索することができる。また、本論文ではただ最適化計算で解を得るだけではなく、他の最適化問題に本手法を用いる際の参考になるように、本手法を用いた場合の結果やその性質・特性などの検証もあわせて行っている。

また、GA を用いて効率的な最適化を行うためには、ユーザが GA の条件パラメータを適切な値に設定しなければならない。条件パラメータの値が適切でなければ、GA は本来の最適化性能を最大限発揮することができない。しかし、この作業は経験的な知識が要求させるために、初めて GA を使うユーザが適切な設定を即座に決定することは困難であり、試行錯誤を繰り返して条件パラメータの設定を行っている。より複雑な問題を解くために高度な GA 手法が提案されるが、GA が高度になればなるほど、ユーザが設定しなければならない条件パラメータの数が多くなるため、適切な条件パラメータの設定はさらに困難となる。実際の研究場面においては、最適化計算が必要な研究者の多くは自分の専門分野の研究に対して GA を道具として使いたいと思う「GA 初心者」である場合が多い。そのため、初心者ユーザにも使いやすい手法であるためには、設定が必要な条件パラメータの数が少なく、そのパラメータの設定により最適化性能が影響を受けにくい手法でなければならない。生存率 GA では、ユーザが設定すべきパラメータ数が従来手法よりも少ない2つだけ(GA の個体数と生存率)であり、ユーザの条件パラメータ設定の負担を軽減にもなっている。

本論文の 2 章では、生物シミュレーションにおける最適化計算の必要性について、一過性の反応挙動を示す大腸菌の熱ショック応答最適化問題を対象として、従来の最適化手法を用いた際の問題点について述べる。3 章ではこの熱ショック応答に対して 2 つの評価基準を設定し、従来手法で解く場合について評論する。4 章では、2 つの評価基準を持つ最適化問題をより効率的に解くために提案した生存率 GA を説明し、この手法を用いた場合の最適化計算の結果について述べる。また、5 章では別

のタイプの生物シミュレーションとして恒常的な周期挙動を示すショウジョウバエの概日リズムにおける 2 つの目的関数の最適化問題に生存率を用いた場合の結果について述べる. 更に6章では, モジュール分割可能な反応系である大腸菌の窒素同化システムに対して, 生存率 GA を効果的に用いる方法について述べる.

2. 生物シミュレーションの最適化問題

計算機の性能向上により、大規模な計算を短時間で行えるようになってきた。特にバイオインフォマティクスの分野においては、この情報処理技術を生命現象の膨大で複雑な解析に用いることで、生物現象を計算機上で再現してその真理を追究しようとする研究が盛んに行われるようになった。

一般的に、生物の現象を計算機上にシミュレーションとして再現する場合は、実験や過去の知見から得られた結果を数式で記述して、計算機に計算させればよいと思われるかもしれない。しかし、生物の反応を正確に再現するには既知の情報が不十分である。なぜなら、実際の生物実験などにおいては、物質量が非常に微量であり、連続的な測定や実験が物理的に困難である場合が多いからである。そのため、生物反応の反応経路や全体の仕組みは実験結果により得る事が出来るが、それらの反応挙動を司る各種定数の値は測定できずに未知の場合が多い。そのため、計算機シミュレーションを用いた Reverse engineering を用いた再現実験が行われている。

Reverse engineering では、これまでの研究で解明されている内容からシミュレーションを構築し、未知のパラメータなどは変数として与える。これらの複数の変数の値を調節して、シミュレーションの結果が実際の反応挙動と一致すれば、調節を行った変数の値は、実際の値の候補であるといえる。このように、物理的な実験で測定できないパラメータをシミュレーションの立場から推定することで、実験では得られない値を推定しようとする研究である。しかし、調整が必要な変数の数が多く、それらが互いに相互依存関係を持つため、解析的な最適化手法は適用できない。そこで、確率的な最適化計算が試みられてきた。特に遺伝的アルゴリズム(Genetic algorithms, GA)[1]を用いた最適化では、従来手法よりも良好な結果が得られる事が多く報告され、よく使用されている。

しかし、GA での最適化は多くの反復計算が必要であるため、最適化完了までに多くの時間が要求される。また、1つのシミュレーションに関して、各種環境要因を変化

させたときでも実際の生物反応を同一の挙動を示すか検証する必要があり、複数回の最適化計算が必要である。そのため、この生物反応の Reverse engineering に関する研究においては、パラメータの最適化計算が進行のボトルネックになっている。つまりは、この研究の進行を妨げているパラメータ最適化計算をより効率的に行うことが出来れば、全体の研究進度も向上することになる。そこで、本論文では、生物の動的シミュレーションの最適化を対象として、より効率的な最適化計算が行えるような手法の提案と検証を行っている。

2. 1. 対象問題：大腸菌の熱ショック応答

まず、生物シミュレーションの中から対象問題としたのは、大腸菌の熱ショック応答シミュレーションである。この反応系は全体の経路が比較的小さく、この反応を対象とする研究も盛んに行われているため、物理的に得られる知見も多い[2], [3]。この反応挙動は一過性のピーク挙動を持つため、実際の挙動と一致する解を発見する前に、ピーク挙動を示す解候補を発見し、探索空間の絞込みを行わなければならない。自然界において、大腸菌は環境温度が摂氏 37 度であるとき盛んに増殖する。しかし、環境温度が摂氏 42 度以上になると折りたたまれている蛋白質 (Folded protein) が変性し蛋白質の立体構造が破壊される(Unfolded protein)。この Unfolded protein の蓄積が細胞死を引き起こす原因になる。そこで、この細胞死を防ぐ防御反応が「熱ショック応答(Heat Shock Response)」である。大腸菌の熱ショック応答の反応経路は遺伝子発現ネットワークで表現すると(図 2-1)のように表現される。この反応系の全てを二相分離法を用いて偏微分方程式と代数方程式で記述して逐次的な解析を行うことで、各物質濃度の時系列変化を計算することができる。[4], [5]

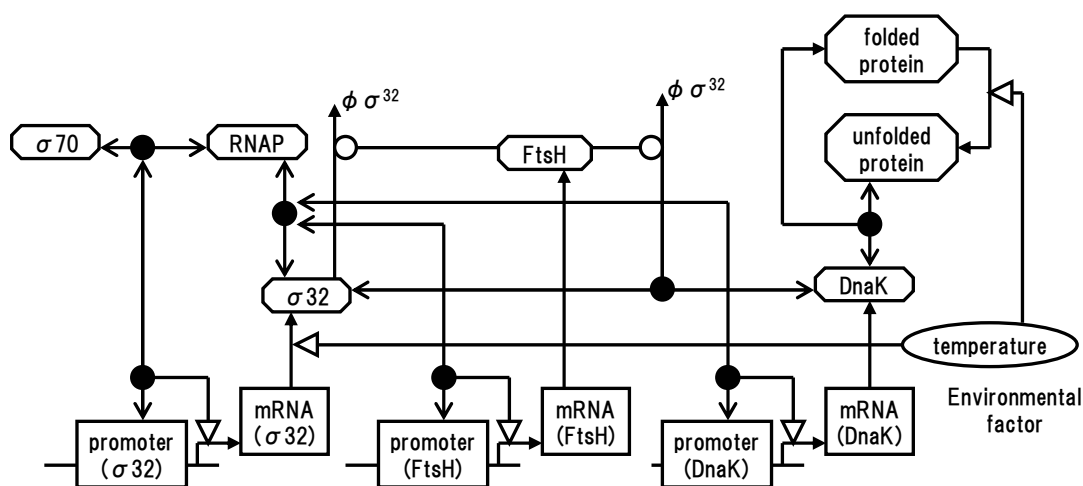


図 2-1. 大腸菌熱ショック応答のネットワークマップ

環境温度が42度上昇すると細胞内の蛋白質が変性を始める。シャペロンであるDnaKは、 σ^{32} から離れ、変性蛋白質に結合し、巻き戻しをはじめる。遊離した σ^{32} はRNAポリメラーゼ(RNAP)と結合して熱ショック蛋白質(HSP)の発現を促す。HSPはシャペロン(DnaK など)とプロテアーゼ(FtsH など)のセットである。こうして、合成されたDnaKはUnfolded proteinに結合して正常な状態に巻き戻す。蛋白質の巻き戻しが終わると、過剰に合成されたDnaKは再び σ^{32} と結合し、 σ^{32} がRNAPと結合しにくくなって、HSPの転写が抑制される。さらに、FtsHが σ^{32} を分解することによってHSP合成が抑制される。こうして新たな定常状態となる。

2. 2. 遺伝的アルゴリズムを用いた大腸菌熱ショック応答の最適化

本問題では熱ショック応答の特徴である σ^{32} が一過性のピーク挙動を正確に再現することが求められる。今回の検証では、反応系への影響が強い9つの蛋白質や酵素の結合定数を未知として、これらの値の最適化を行う(図 2-2)。このとき、9個のパラメータの値を変えるとシミュレーションの結果はさまざまな挙動を示す(図 2-3)。最適化が必要なこれらのパラメータは互いに依存関係を持ち、探索範囲が広いため、解析的な手法では最適化が困難であった。そこで、確率的な最適化手法でGAを用いた。目的関数として、本反応において特徴的な挙動を示す蛋白質 σ^{32} 濃度の時間変化とシミュレーションによる時間変化曲線との二乗誤差をとり、その二乗誤差を最小化することによって最適化を行う。このようなフィードバックなどの相互作用のある複雑系はロバスト制御であるため、値が多少変化しても系への影響は小さい。そのため GA における突然変異などの乱数効果、特に近傍探索の検索効果が反映されにくくなる。そこで、今回は結合定数の対数値を遺伝子とすることで、探索空間を縮小すると同時に、上述の問題点も解消している(図 2-4)。

Keq[1]	$\text{RNAP} + \sigma^{70} \rightarrow \sigma^{70}:\text{RNAP}$
Keq[2]	$\text{RNAP} + \sigma^{32} \rightarrow \sigma^{32}:\text{RNAP}$
Keq[3]	$\text{RNAP} + \text{DNA} \rightarrow \text{DNA}:\text{RNAP}$
Keq[4]	$\sigma^{32} + \text{FtsH} \rightarrow \sigma^{32}:\text{FtsH}$
Keq[5]	$\sigma^{32} + \text{DnaK} \rightarrow \sigma^{32}:\text{DnaK}$
Keq[6]	$\text{mRNA}(\sigma^{32}) + \text{DnaK} \rightarrow \text{mRNA}(\sigma^{32}):\text{DnaK}$
Keq[7]	$\text{Punfold} + \text{DnaK} \rightarrow \text{Punfold}:\text{DnaK}$
Keq[8]	$\text{Promoter} + \sigma^{70}:\text{RNAP} \rightarrow \sigma^{70}:\text{RNAP}:\text{Promoter}$
Keq[9]	$\text{Promoter} + \sigma^{32}:\text{RNAP} \rightarrow \sigma^{32}:\text{RNAP}:\text{Promoter}$

図 2-2. 本問題でコーディングした熱ショック応答のパラメータ

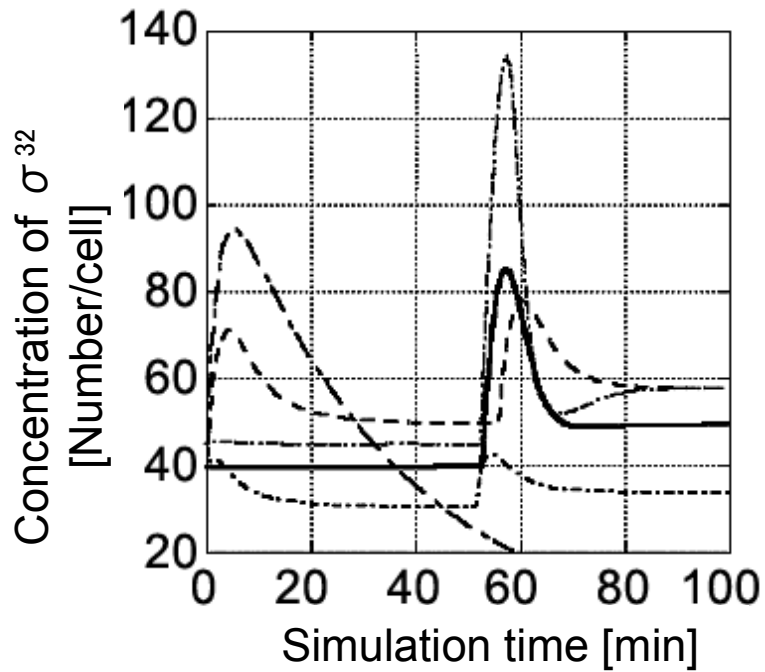


図 2-3. 調節蛋白質 σ^{32} の挙動変化

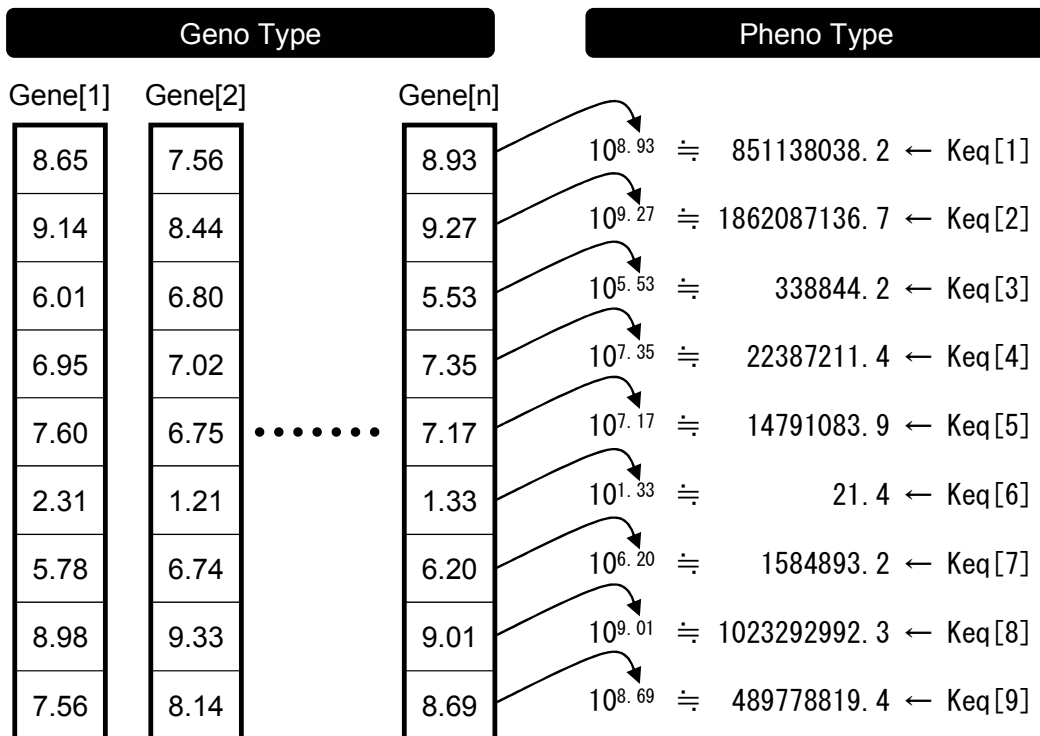


図 2-4. 実数値での遺伝子コーディングの例

2.2.1. 遺伝的アルゴリズム

遺伝的アルゴリズム(Genetic Algorithm, GA)[1]は生物の進化過程を模倣した確率的数値最適化手法である。自然界において、実際の生物は各世代を形成する個体の集団(個体群)を構成し、環境への適応度が高い個体は高い確率で選択され、交叉や突然変異を経て次世代の個体群が形成される。この繰り返しにより生命は進化を続け、環境に適応した個体が生み出されていく。GAはこのダーウインの進化論に基づく生物の進化の過程を模倣して考案された学習的アルゴリズムである。

GAにおいて、最も単純な世代交代モデルはSimple GA(SGA)であり、図2-5で示される。SGAでは、初期個体を生成し、各個体が環境に対する適用度(評価値, Fitness)を計算し、その値に基づき交叉や突然変異を行うことで、評価値の高い個体の性質を継承する子個体群を生成する。これまでの操作を1世代として、世代を繰り返していくことで優秀な個体性質を獲得しようとするものである。以下にGA各操作の具体的な手順を示す。

1. 初期化(Initialization): GAでは、各個体は複数の遺伝子(gene)の集まり(染色体)で構成される。よって、1つの個体は複数個の数値データを有する。遺伝子は個体の性質を定義し、初期化ではランダムな染色体を持つN個の個体(個体群)を生成する。
2. 評価(Fitness): 各個体に対して、任意の評価式に従った適応度(評価値)を計算する。適応度の最も高い個体(エリート個体)は以下の③～⑤の操作は行わず、無条件で次世代に保存する(エリート戦略)。この操作により最良個体の消失を防ぐ。エリート個体の適応度が十分高いと判断できる時は、その個体を最適解として計算を終了する。
3. 淘汰・選択(Selection): 個体集団の中から適応度に依存した一定の規則で個体を選択する(選択)。選択規則には個体の適応度順位に依存する「ランク選択」や各個体が全体の適応度に占める割合に依存する「ルーレット選択」などがある。

この選択により適応度の低い個体のいくつかは淘汰され、適応度の高い個体が増殖する。交叉と合わせて行われる。

4. 交叉(Crossover): 選択された個体に対し、ある交叉確率で遺伝子の入れ換えや補間などの操作を行い新しい性質(遺伝子)を持つ個体を生成する。
5. 突然変異(Mutation): 全ての遺伝子に対し一定の確率で突然変異を起こし、遺伝子(データ値)を変更する。この操作により、個体群の多様性が維持される。

こうして生成された新個体に対し③～⑤の操作を繰り返す。

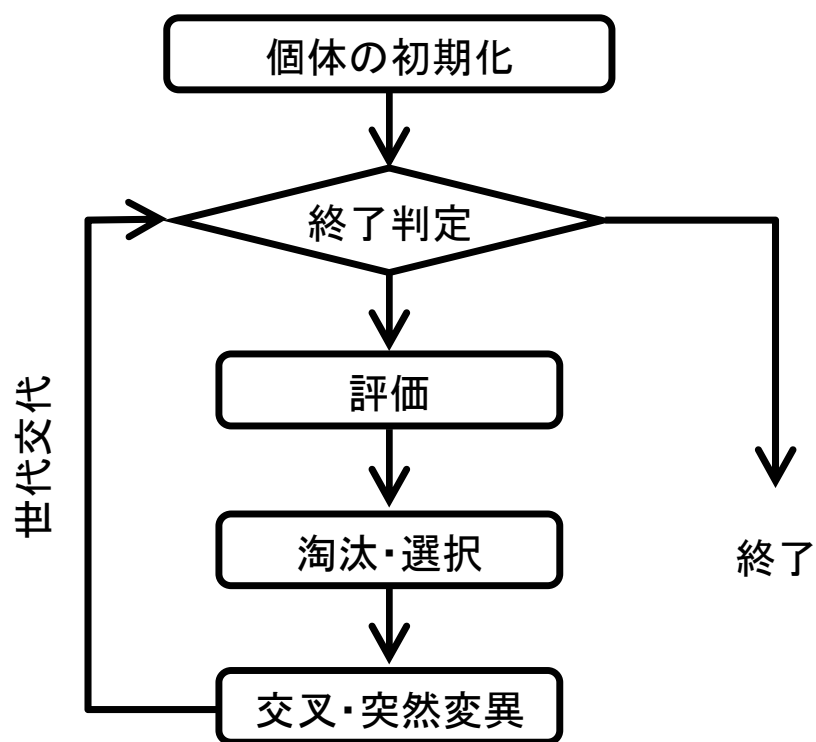


図 2-5. 遺伝的アルゴリズムの流れ

2. 2. 2. 実数値遺伝的アルゴリズム(Real-coded GA)

GA において、個体遺伝子の表現にはさまざまな手法が提案されている。遺伝子を2進数で表現するビット型 GA では、個体遺伝子を0か1のビット列で記述し、これを生物の遺伝子配列にみたて、交叉や突然変異では、このビット列の入れ替えや反転を行うことで優良遺伝子への収束を行う。これに対し、遺伝子を実数値で表現する GA は実数値遺伝的アルゴリズム(Real-coded GA)[6]と呼ばれている。実数値 GA では、個体の遺伝子を実数値配列で表現される。また、交叉は、これらの値を数値的に更新することで子個体の遺伝子を生成する。GA の最適化性能を大きく左右する交叉操作においては様々な手法が研究・検討されており、各種の比較実験により良好な最適化を与える方法やそのパラメータ設定が提案されている。

2.2.3. 交叉

交叉は、遺伝的アルゴリズムの最大の特徴である。この操作で生成される子個体は親個体の性質を受け継ぐ事で、探索点の分布を望ましい方向に導く操作である。交叉の手法は数多く提案されているが、今回の最適化では、実数値 GA との相性が良く、高い安定性と最適化性能に定評がある UNDX[7]を用いる。

UNDX (単峰性正規分布交叉)では、両親を結ぶ直線上に第3の親との距離を基にした正規分布の近傍に子個体が生成される(図 2-6)。

UNDX のアルゴリズムは以下の通りである。

1. 3 個の親を x_1, x_2, x_3 とする。
2. 親 x_1, x_2 の中点を $x_p = (x_1 + x_2)/2$ とする。
3. 親 x_1, x_2 の差ベクトルを $d = x_1 - x_2$ とする。
4. 親 x_1, x_2 を結ぶ直線を主探索直線と呼び、親 x_3 から主探索直線までの距離を D とする。
5. 子 x' を以下の式に従って生成する。

$$x' = x_p + \xi d + \sum_{i=1}^{n-1} \eta_i D e^i,$$

$$\xi \sim N(0, \sigma_\xi^2), \quad \eta_i \sim N(0, \sigma_\eta^2)$$

ここで、 n は探索空間の次元を、 $N(0, \sigma^2)$ は平均 0、分散 σ^2 の正規分布を、 e_i は主探索直線に直交する部分空間の正規直交基底ベクトルをそれぞれ表す。上記のアルゴリズムのパラメータの経験的推奨値は $\sigma_\xi = 1/2$ 、 $\sigma_\eta = 0.35/\sqrt{n}$ であり、このとき UNDX は設計指針を比較的よく満たす事が理論解析によって示されている。

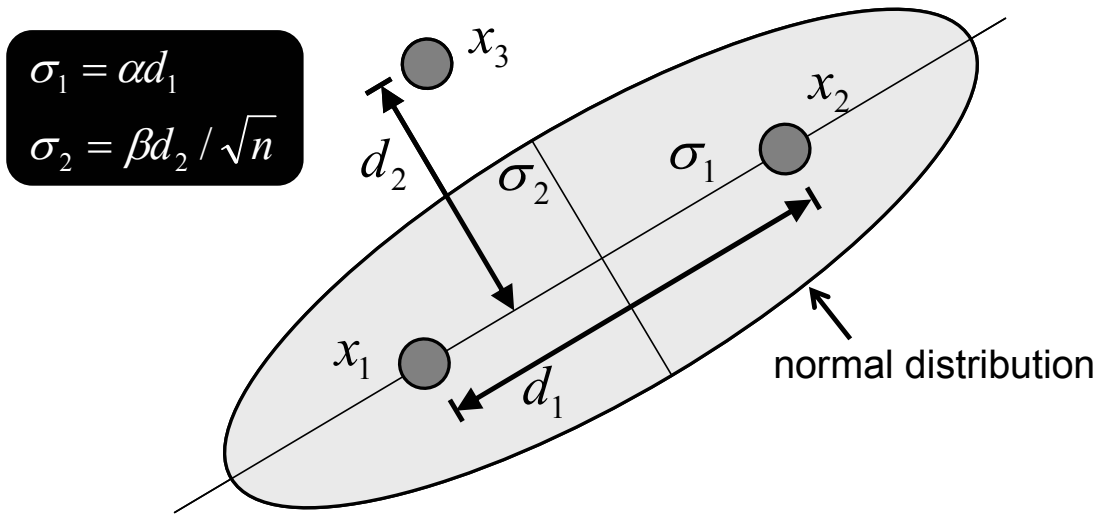


图 2-6. UNDX (单峰性正规分布交叉)

2. 3. 大腸菌熱ショック応答における、これまでの最適化計算と問題点

これまでの研究でも、大腸菌熱ショック応答の最適化をより効率的に行うために、GAを拡張した分散統合 GA[8]や優性方向優先探索法[9]などを提案し、効率化を実現してきた。提案手法を用いた検証実験では、最適化成功確率は2~3倍に、平均計算回数も1/2~1/10となり、全体の研究進度のボトルネックとなっている最適化計算の問題を軽減する事ができた。

しかし、実際の研究の現場における最適化では、最適解の値も未知であれば、どこまでの範囲を探索すれば最適解が得られるかも未知である。そこで、実際の現場での使用を考慮して、パラメータ探索の探索範囲を広くした場合での最適化実験を行ったところ、一過性のピーク挙動を示す解候補を見つけ出すことが困難になり、最適化成功確率も10%未満にまで落ちてしまった。ピーク挙動を示す解が存在する範囲が探索範囲に全体に対して非常に狭いため、最適化の成功確率は低下する。

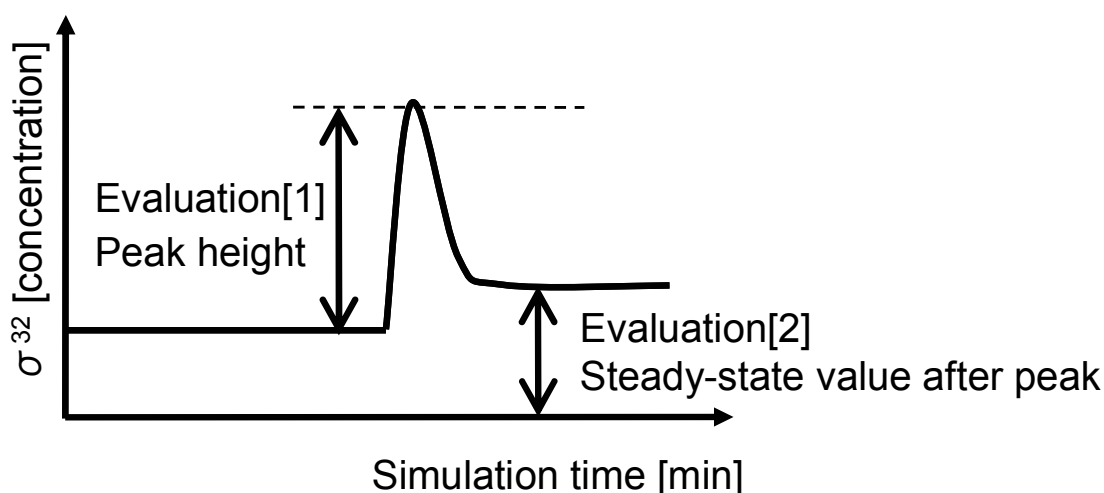
3. 2つの目的関数をもつ最適化問題

大腸菌の熱ショック応答最適化問題において、動力学パラメータの探索範囲を大きくすると、特徴的な σ^{32} の過渡応答(鋭いピークを持つ時間変化)の再現が難しくなった。そこで、本研究では、熱ショック応答システムの過渡応答のダイナミクスに関する最適化問題を効率的に解くために、ピークを表現する2つの目的関数で定義することにした。

単一評価関数でのGAでは、特に探索範囲が広い場合は、個体群が局所最適解に収束しやすいため、GAの探索性能が低下し、目的とする解を発見することが困難になる場合がある。そこで、2つの評価基準を用いることで、2つの探索軸から目的とする解の探索を試みる。しかし、評価基準が2つある問題においては、目的とする解は唯一の最適解ではなく、各評価基準における許容条件(終了条件)を共に満たす解をできるだけ多く見つけ出すことが重要となる。特に生物シミュレーションなどを対象としたときは、値の異なる複数の解が同一の挙動を示す場合が多くみられる。つまりは複数の局所解が存在することが一般的である。そのため、少ない計算回数と高い成功確率で、条件を満たす解を見つけることも重要であるが、それと同時に、2つの評価基準における条件を満たす局所最適解をできるだけ多く見つけ出すことも大切である。こうして発見された複数の局所解の結果は、反応経路を専門的に取り扱っている研究者にフィードバックすることで、反応経路を特定する有益な情報や知見を提供することもできるようになる。

3. 1. 大腸菌熱ショック応答における2つの目的関数

図3-1は、大腸菌の熱ショック応答シミュレーションにおける2つの評価基準を示している。本系では蛋白質 σ^{32} が一過性のピーク挙動を示し、このピークの形状を実際の反応挙動との類似性判定に用いている。今回の実験で定義した評価基準は「ピークの高さ」と「ピーク後の定常値の値」である。ピークの高さは σ^{32} の濃度が 40.0 に近いほど、ピーク後の定常値の値は 50.0 に近いほど実際の挙動に近い。最適化計算の終了判定は、ピークの高さが 38.0~42.0 の範囲に収まり、かつ定常値の値が 49.0~51.0 の範囲に収まったときに最適化成功とした。(表 3-1)



< Peak height >

$$Evaluation[1] = \frac{|\sigma^{32}[\text{simulation}] - \sigma^{32}[\text{target}]|}{|\sigma^{32}[\text{acceptable value}] - \sigma^{32}[\text{target}]|}, \quad \begin{matrix} (0.0 \leq \text{Peak height} \leq 200.0) \\ (0.0 \leq Evaluation[1] \leq 1.0) \end{matrix}$$

< Steady – state value after peak >

$$Evaluation[2] = \frac{|\sigma^{32}[\text{simulation}] - \sigma^{32}[\text{target}]|}{|\sigma^{32}[\text{acceptable value}] - \sigma^{32}[\text{target}]|}, \quad \begin{matrix} (0.0 \leq \text{Steady – state} \leq 100.0) \\ (0.0 \leq Evaluation[2] \leq 1.0) \end{matrix}$$

図3-1. 大腸菌熱ショック応答シミュレーションにおける2つの目的関数 (評価1:ピークの高さ, 評価2:定常値)

表 3-1. 大腸菌熱ショック応答最適化問題における評価値の許容範囲と終了条件
(ピークの高さ・定常値)

目的関数	目標値	許容範囲	終了条件
ピークの高さ [個/cell]	40.0	0.0~200.0	40.0 ± 2.0
定常値 [個/cell]	50.0	0~100.0	50.0 ± 1.0

3. 2. 2つの目的関数を対象とする遺伝的アルゴリズム

一般的には, GA は 1 つの目的関数を最適化するアルゴリズムであるため, 2 つ以上の目的関数を持つ場合に対しては, 工夫が必要である. 目的関数を重み付き線形結合により 1 つに統合する手法(重み付き線形結合 GA=Linear Combination GA)や, 複数の個体群のそれぞれに目的関数を割り当てる分散 GA を拡張した手法(分散協力 GA, Distributed Cooperation model of Multi-Objective Genetic Algorithm, DCMOGA)などが提案されている.

3. 2. 1. 重み付き線形結合 GA

重み付き線形結合 GA (Linear Combination GA)は, 重み係数を用いて 2 つの目的関数を 1 つに統合する手法である(図 3-2). 最もシンプルな手法で, 2 つ(または 2 つ以上)の目的関数を対象とする問題において, よく使われている. 線形結合評価において最適化が上手く進むかは重み係数の設定に大きく依存している. この重み係数の設定が適切でない場合には, 片方の評価基準が重視され, バランスよく双方の性質を持つ解が発見しづらくなる可能性がある. また, 適切な重み係数を設定するには, 2 つの目的関数の変数強度やオーダ 評価値の分散や偏差なども考慮する必要がある.

$$\boxed{\text{Weight function}} = w \times \boxed{\text{Evaluation 1}} + (1-w) \times \boxed{\text{Evaluation 2}}$$

(w: weight coefficient, $0.0 \leq w \leq 1.0$)

図 3-2. 線形結合評価での重み関数と重み係数

3. 2. 2. 分散 GA

分散 GA (Distributed GA)では、複数の個体群を用い、それらの間で個体の入れ替えを行うことで、単一個体群 GA の局所収束による探索能力の低下を軽減し、個体性質の多様性も得る [10]. 各個体群で世代交代を行い、一定世代間隔ごとに移住率に従い個体の入れ替え(移住操作)を行う. 複数の個体群を用いるため、単一個体群の GA に比べて、1 世代あたりの計算負荷は倍増するが、単一個体群での局所収束による探索能力低下の問題点を軽減している. 2 つの目的関数への対応としては、個体群を同一の目的関数で評価する手法や、各個体群に別々の目的関数を割り振る手法などがある(図 3-3).

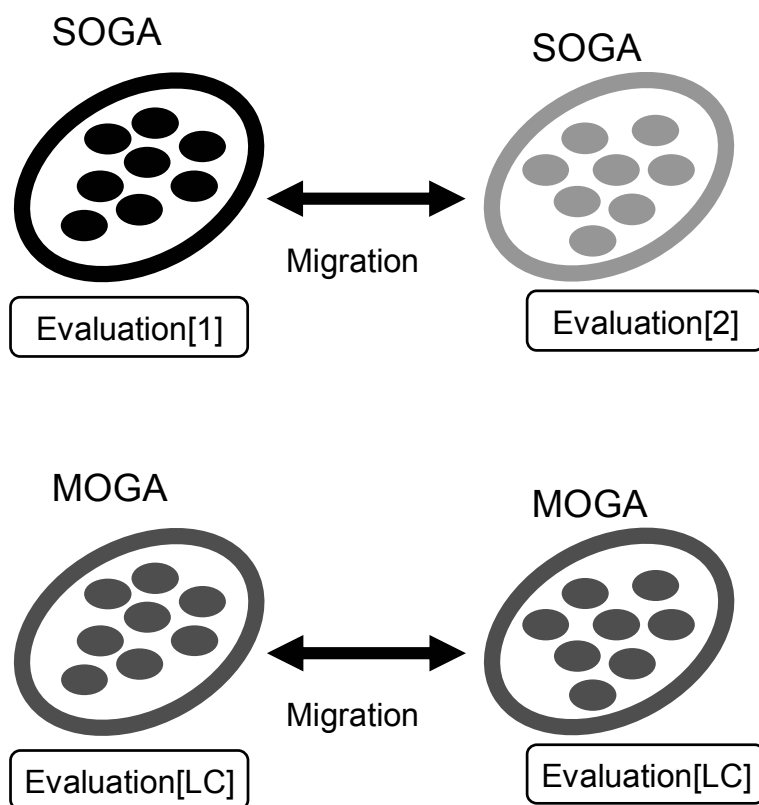
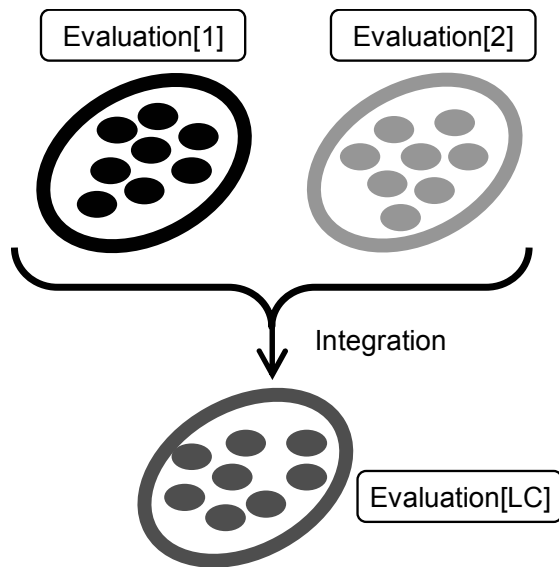


図 3-3. 分散 GA のアルゴリズムの概要. 各個体群に別々の目的関数を用いる手法(上図)や共通の目的関数を用いる手法(下図)がある.

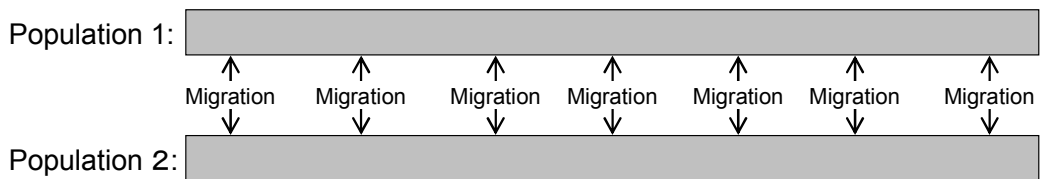
3. 2. 3. 分散統合 GA

分散統合 GA (Distributed and Integrated GA)は、田中らが 2000 年に提案した手法であり、分散 GA の個体群分割を応用した世代交代手法である(図 3-4)「DIGA」。分散 GA での移住において、移住率や移住間隔などの設定が不適切であるときは、移住操作により個体性質の攪拌が起こりすぎて、ある個体群の性質が隣接する個体群の性質に引き寄せられ、類似性が高くなることで、重複的な過剰探索により、最適化計算が非効率化する場合があった。また、移住操作のパラメータは最適化性能にも大きく影響を与えるため、適切な値を決定することはユーザの負担にもなっていた。

そこで、分散統合 GA では、移住操作は行わずに各個体群が独立して世代交代を行う。その後、各個体群から個体を抽出して新しい個体群を作り、この個体群のみに世代交代を行う(統合操作)。この操作により、統合前に各個体群(各目的関数)で探索範囲の絞込みを行い、統合後は、各個体群を頂点とする空間の内部、および周辺を再探索する。また、統合後は探索する個体数を減らすことで、局所探索における重複的な計算ロスを軽減する。



<Distributed GA, DGA>



<Distributed and integrated GA, DIGA>

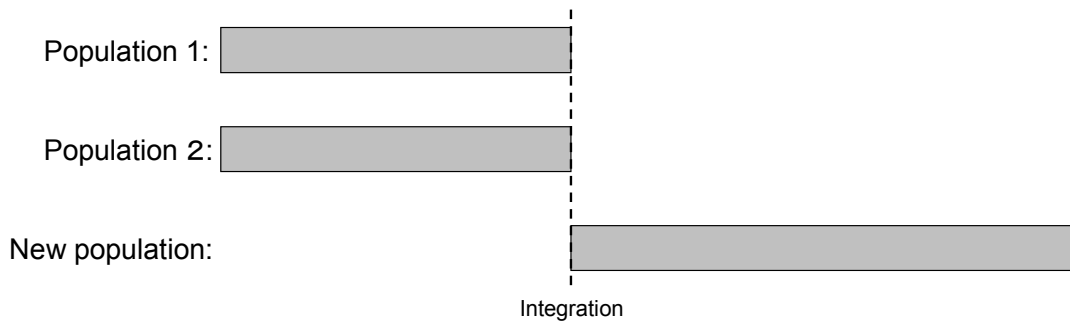


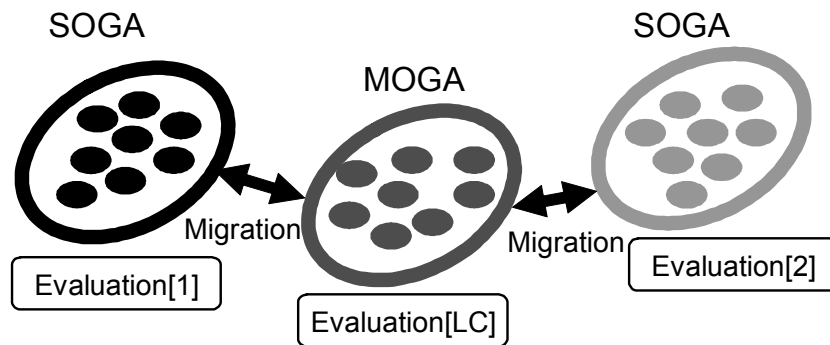
図 3-4. 分散統合 GA の概要(上図)と分散GA との計算負荷の比較(下図)

3. 2. 4. 分散協力 GA

分散協力 GA (Distributed Cooperation model of Multi-Objective Genetic Algorithm, DCMOGA)とは、複数の個体群を用いる分散 GA の考え方を基礎として、単一母集団線形結合 GA での局所収束の欠点を軽減した手法である(図 3-5)[11].

n個の目的関数を評価基準とする問題の場合、n個の個体群(SOGA 個体群)と、n個の目的関数を包括する評価基準を用いる個体群(MOGA 個体群)、合計(n+1)個の個体群を用いて、これらの中で個体の入れ替えを行う。SOGA 個体群ではそれぞれの目的関数に対し優れた性質の探索を行い、MOGA 個体群では、各 SOGA 個体群から移住によって持ち込まれる目的関数全体に適した性質を評価して、多目的関数の最適化を進める。

使用に際しては、MOGA 個体群で必要な重み係数だけでなく、移住操作に関する移住率や移住間隔などをユーザが適切に設定する必要がある。



<DCMOGA>

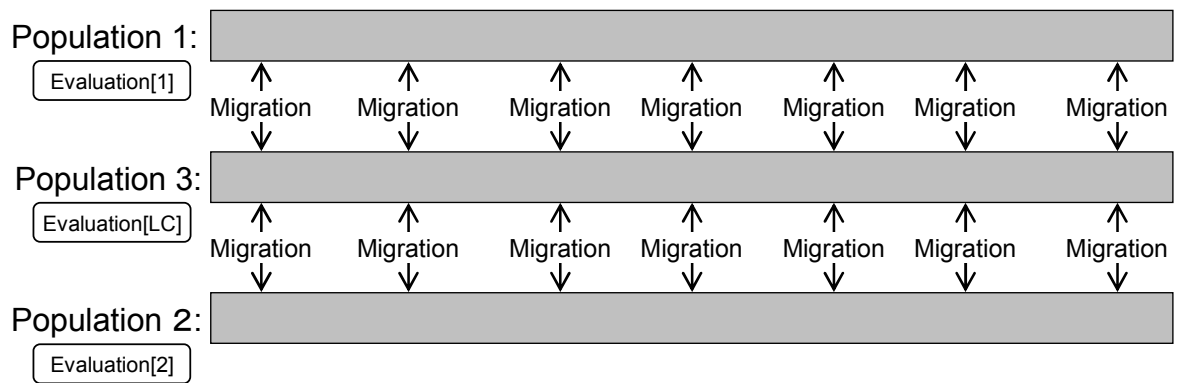


図 3-5. 分散協力 GA (DCMOGA)の概要(上図)とその計算負荷(下図)

3. 3. 既存手法を用いた場合での大腸菌熱ショック応答の最適化

既存手法を用いて、2つの評価基準を持つ大腸菌熱ショック応答のパラメータ最適化問題を解き、最適化完了までの評価値計算回数と成功確率を比較した。計算回数とは、最適化完了までに個体の評価値を計算した回数である。最大評価計算回数を100,000回(100個体のGAで1000世代分に相当)とし、その間に双方の評価値が終了条件を満たしたときに、最適化成功として計算を終了する。設定条件で1世代当たりの計算負荷が異なるため、この値を最適化性能の比較に用いた。またGAは確率的な最適化手法であるため、それぞれに対し20回の試行を行い、その統計で比較を行った。

3. 3. 1. 線形結合 GA を用いた場合の結果

図 3-6 は、線形結合 GA を用いて、熱ショック応答シミュレーションの最適化計算を行ったときの平均計算回数と最適化成功確率を示している。棒グラフは、それぞれの手法を用いて最適化計算を行ったときに要した平均計算回数(20 試行分)であり、棒グラフはそれぞれにおける最適化の成功確率を示している。個体群内の個体数を 100, 200, 500 とした各場合において、重み係数の値を 0.0~1.0 まで 0.1 刻みで変化させた各条件で、20 回の試行を行い平均計算回数と最適化成功確率を測定した。平均計算回数は少ないほど、最適化成功確率は高いほど良好な最適化性能を示しているといえる。また、GA の設定条件に対して、平均計算回数と成功確率の変化が少ないほど、GA の最適化性能が設定条件に依存しにくい手法であり、利用者は各試行の設定条件は表 3-2 の通りである。

重み関数では 2 つの目的関数を線形的に 1 つに統合しているため、各目的関数のオリジナルな特徴が結合する事で各評価の重要性が弱まり、お互いの中間的な解しか発見できない場合もみられた。また重み係数の設定が適切な場合では成功確率も高くなっているが、その値から離れると成功確率も低くなっていく。

(表 3-2) 線形結合 GA の条件設定

線形結合 GA
個体数:100, 200, 500 個体
重み係数:0.0~1.0 (0.1 刻み)
交叉:UNDX
試行回数:20 試行

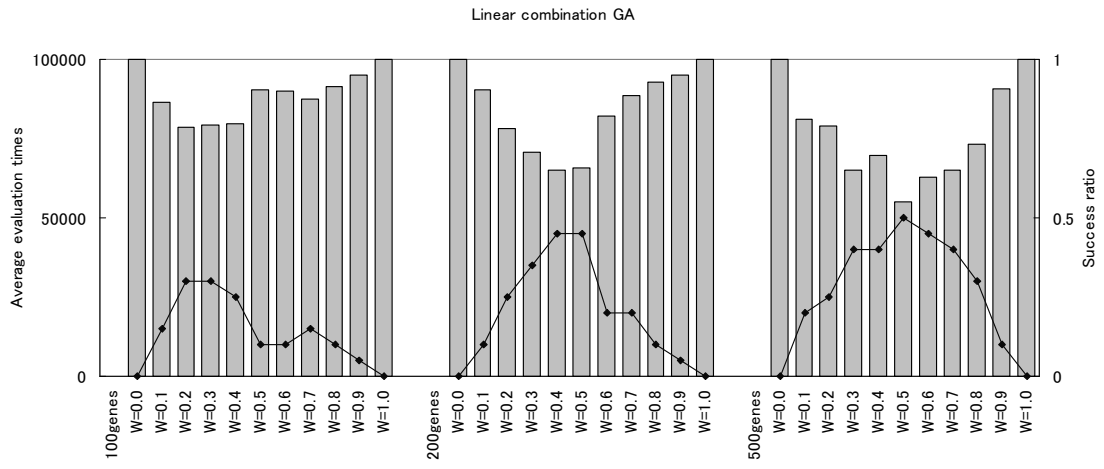


図 3-6. 熱ショック最適化問題における線形結合 GA での平均計算回数と成功確率

3. 3. 2. 分散 GA を用いた場合の結果(個体群ごとに目的関数が異なる場合)

分散GAでは、単一個体群GAにおける個体の初期分布の偏りによる影響を考慮して、複数の個体群を生成し、移住操作により個体性質の多様性を保とうとしている。本手法では探索の開始時に 2 つの個体群を用意し、それぞれの個体群を別々の目的関数で評価する(図 3-6)。また、一定世代おきに個体群間で個体の入れ替え操作(移住)を行うことで、個体群に他方の目的関数で評価された個体を流入させることで、個体性質の多様性を与えられることを期待している。各試行の設定条件は表 3-3 の通りである。各個体群内の個体数を 50, 100, 200 個体とした各場合において、移住操作に関するパラメータ(移住間隔, 移住率)の値を変えた各組み合わせにおいて最適化実験を行った。図 3-7 は、分散 GA を用いて、熱ショック応答シミュレーションの最適化計算を行ったときの平均計算回数と最適化成功確率を示している。

分散 GA で 2 つの個体群に別々の目的関数で評価を行った手法では、各個体群がそれぞれに世代交代を行い、各自の目的関数に対して優秀な解は発見できた。個体の初期分布が偶然的に良い試行では、2 つの目的関数を共に満たす解を発見することができたが、その他の多くの試行では 2 つの目的関数を共に満たす解の発見は困難であった。

<Distributed GA>

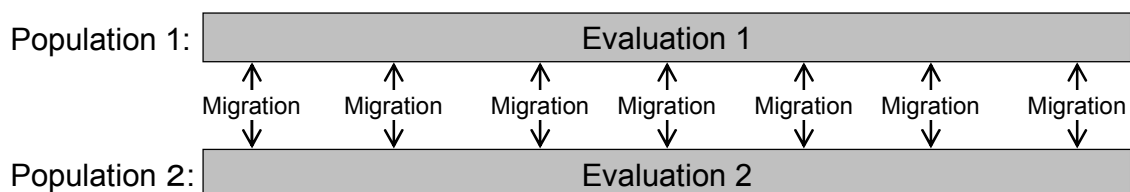


図 3-6. 分散 GA における 2 つの目的関数での個体評価(個体群ごとに目的関数が異なる場合)

表 3-3. 分散 GA の条件設定(個体群ごとに目的関数が異なる場合)

分散 GA(目的関数 1 : 目的関数 2)

個体群内の個体数:50, 100, 200 個体

移住間隔:10, 20, 50 世代おき

移住率:10, 20, 50%

交叉:UNDX

試行回数:20 試行

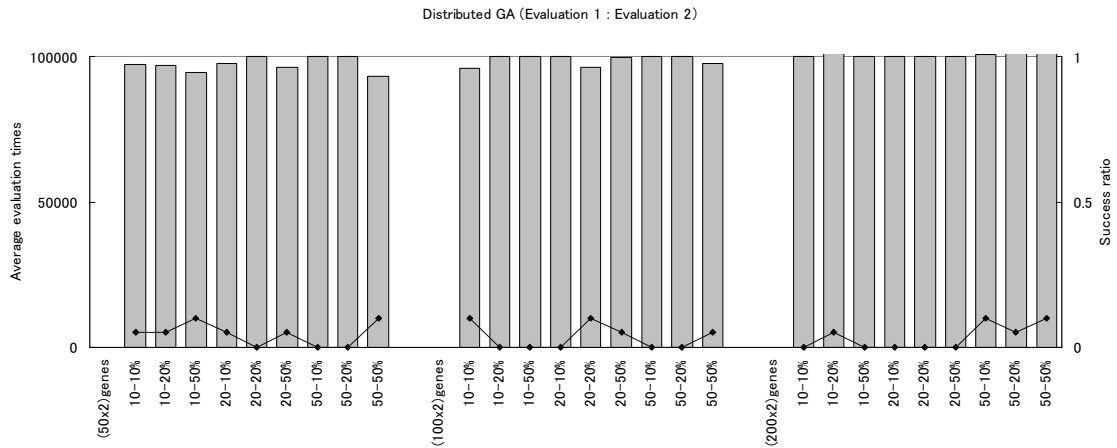


図 3-7. 熱ショック最適化問題における分散 GA での平均計算回数と成功確率(個体群ごとに目的関数が異なる場合)

3. 3. 3. 分散 GA を用いた場合の結果(重み関数を共通の目的関数とする場合)

分散GAでは、単一個体群GAにおける個体の初期分布の偏りによる影響を考慮して、複数の個体群を生成し、移住操作により個体性質の多様性を保とうとしている。本手法では、探索の開始時に 2 つの個体群を生成し、共に重み関数で個体を評価し、移住操作を繰り返す(図 3-8)。各試行の設定条件は表 3-4 の通りである。各個体群内の個体数を 50, 100, 200 個体ときた各場合において、移住操作に関するパラメータ(移住間隔, 移住率)の値を変えた各組み合わせにおいて最適化実験を行った。図 3-9 は、分散 GA を用いて、熱ショック応答シミュレーションの最適化計算を行ったときの平均計算回数と最適化成功確率を示している。

分散 GA で目的関数を重み関数とした場合では、重み係数の値を単一個体群の時に良好な結果を示した $w=0.5$ に固定して、移住に関するパラメータ(移住率, 移住間隔)の組み合わせを変えて実験を行った。単一個体群と比べ、分散 GA では複数の個体群を並列的に世代交代するため、1 世代あたりの計算回数が多くなる。そのため、同じ世代数で最適化研鑽が終了しても、全計算回数は多くなる。また、移住操作の設定値が適切でない場合は、移住により、ある個体群の性質が他の個体群に流入しすぎて、2 つの個体群の性質が類似してしまい、局所の過剰探索により計算回数を浪費してしまうこともある。逆に移住操作が有効に作用しなかったときは、2 つの個体群をそれぞれ別々に世代交代しているだけとなり、成功確率は単一個体群の場合と同程度であるが、平均計算回数が増大してしまう。分散 GA では、移住操作の設定と同時に重み係数の設定も必要である。

<Distributed GA>

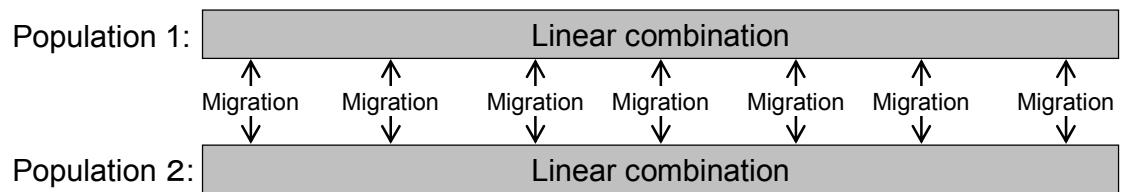


図 3-8. 分散 GA における 2 つの目的関数での個体評価(全ての個体群の評価を重み関数で行う場合)

表 3-4. 分散 GA の条件設定(全ての個体群の評価を重み関数で行う場合)

分散 GA (重み関数: 重み関数)
個体群内の個体数: 50, 100, 200 個体
重み係数: 0.5
移住間隔: 10, 20, 50 世代おき
移住率: 10, 20, 50%
交叉: UNDX
試行回数: 20 試行

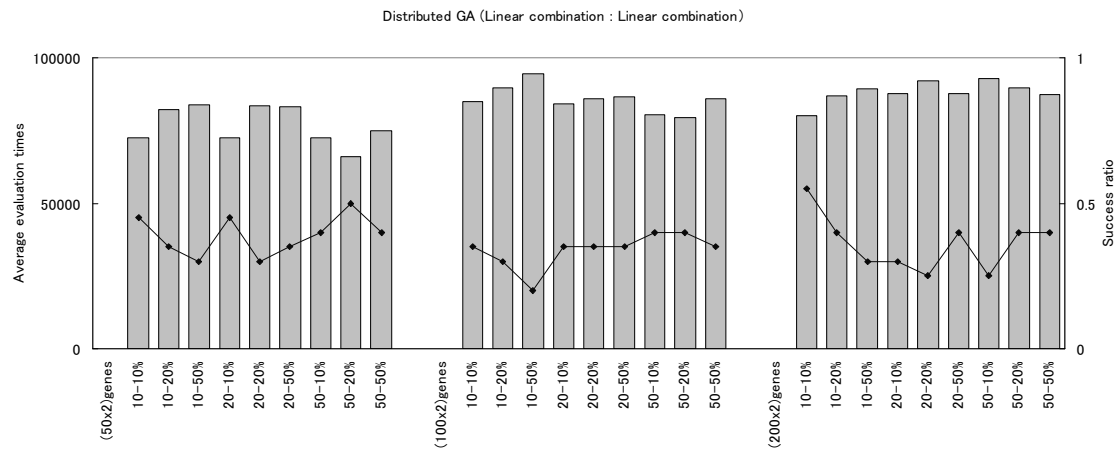


図 3-9. 熱ショック最適化問題における分散 GA での平均計算回数と成功確率(全ての個体群の評価を重み関数で行う場合)

3. 3. 4. 分散統合 GA を用いた場合の結果

分散統合GAでは、単一個体群GAにおける個体分布の偏りによる影響を軽減するために複数の個体群を使用する。また、分散GAの移住操作における個体性質の過剰な攪拌と、母集団間の類似性が高まるために起こる局所の過剰探索による計算負荷の浪費を軽減するために、個体群間で移住操作は行わずに、ある世代数において各個体群から個体群を抽出して新しい個体群を作り、統合前までに絞り込まれた範囲の再探索を行う手法である(図 3-10)。本問題では目的関数の数が2つなので、探索開始時に2つの個体群を生成し、別々の目的関数で評価を行う。個体進化はそれぞれ独立して行い、統合世代において各個体群から抽出された個体を用いて新しい個体群を生成する。この新しい個体群では重み関数を用いた評価を行う。各試行の設定条件は表 3-5 の通りである。各個体群の個体数を50, 100, 200 個体とした各場合において、統合世代数を10~200と変えて最適化実験を行った。統合後の新しい個体群で用いる重み関数の重み係数の値は、これまでの実験で良好な値を示した $w=0.5$ で実験を行った。実際の研究現場で使用する際には、ユーザが重み係数の値も適切に設定しなければならない。図 3-11 は、分散統合 GA を用いて、熱ショック応答シミュレーションの最適化計算を行ったときの平均計算回数と最適化成功率を示している。

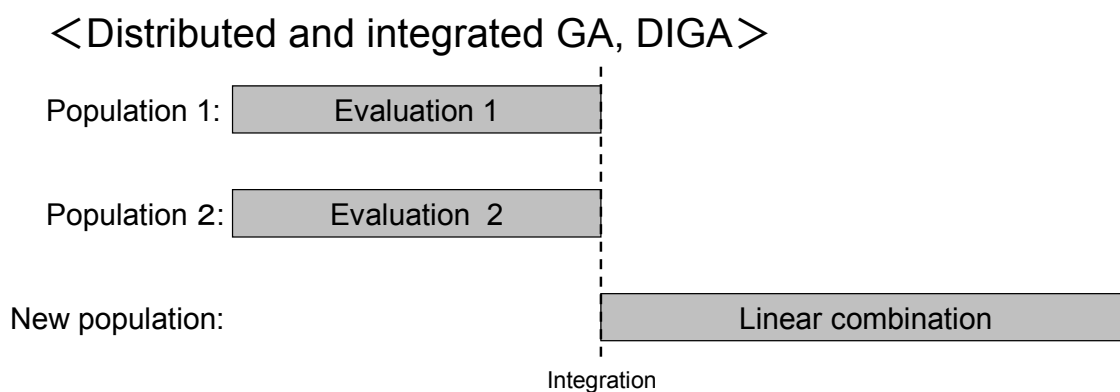


図 3-10.分散統合 GA における各個体群における個体評価式

表 3-5. 分散統合 GA の条件設定

分散統合 GA

個体群内の個体数:50, 100, 200 個体

重み係数:0.5

統合世代数:10, 20, 30, 40, 50, 100, 200

交叉:UNDX

試行回数:20 試行

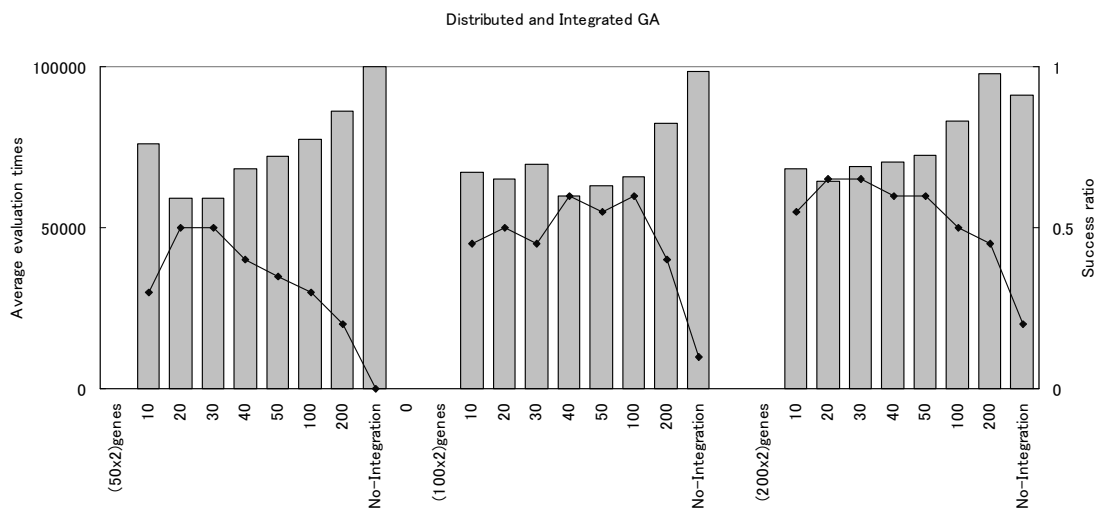


図 3-11. 熱ショック最適化問題における分散統合 GA での平均計算回数と成功確率

分散統合 GA では、分散 GA における探索中期以降の過剰な局所集中探索を軽減するために、複数の個体群から個体を抽出し、新しい個体集団を作る統合操作により、評価計算を必要とする1世代あたりの個体数を減少させる事で、余分な計算回数の蓄積を軽減している。今回は、統合前の各個体群ではそれぞれの目的関数で評価を行い、統合後の個体群では重み関数での評価を採用した。平均計算回数は成功確率が同程度の分散 GA の結果よりも少なくなっている。また、分散統合 GA でユーザが設定するパラメータは個体数と統合世代数だけであり、移住操作を行わない

ことで、ユーザが設定すべきパラメータ数を減らし、設定依存の負担も軽減している。移住を行わずに各個体群が独立して世代交代を行うことで、統合後の個体群には、各評価基準に優れた個体が集められる。そして、これらの個体を親として子個体を生成し、統合前までに絞り込まれた範囲とその内部の再探索を行う。統合世代数が10~50世代では、良好な結果を示したが、統合世代数が100世代以上となると施工確率が低下する傾向がみられた。これは、統合前に複数の個体群が存在する期間が長くなるために、その分の計算負荷が蓄積する原因と、統合前の個体群で探索範囲の絞り込みが進みすぎて、統合後の個体群に多様性を十分に確保できずに局所収束が起りやすいことが原因である。

3. 3. 5. 分散協力 GA を用いた場合の結果

分散協力 GA では、各目的関数で評価される SOGA 個体群と多目的関数で評価される MOGA 個体群を用いる。本問題は 2 つの目的関数を持つので、分散協力 GA では合計 3 個の個体群を使用し、移住操作を行いながら MOGA 個体群に各 SOGA 個体群の個体性質が流入することで、多様性の維持を試みている(図 3-12)。各試行の設定条件は表 3-6 の通りである。各個体群内の個体数を 50, 100, 200, 500 個体とした場合に、重み関数を 0.0~1.0 まで変えて最適化実験を行った。移住操作に関するパラメータは事前の予備実験で良好な結果を示した値(移住間隔:20 世代, 移住率 20%)を使用した。実際の実験現場においては、ユーザがこれらの移住間隔, 移住率の値を適切に設定する必要がある。図 3-13 は、分散協力 GA を用いて、熱ショック応答シミュレーションの最適化計算を行ったときの平均計算回数と最適化成功確率を示している。

<DCMOGA>

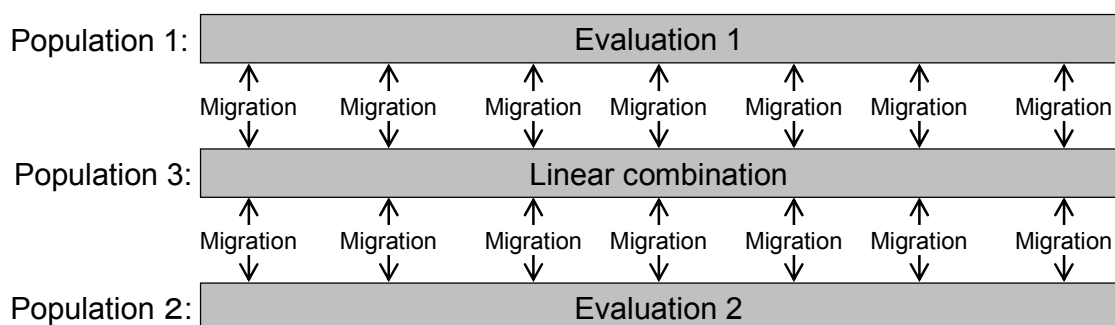


図 3-12. 分散協力 GA における各個体群での個体評価式

表 3-6. 分散協力 GA の条件設定

分散協力 GA(目的関数 1:目的関数 2:重み関数)

個体群内の個体数:50, 100, 200, 500 個体

重み係数:0.0~1.0 (0.1 刻み)

移住間隔:20 世代おき

移住率:20%

交叉:UNDX

試行回数:20 試行

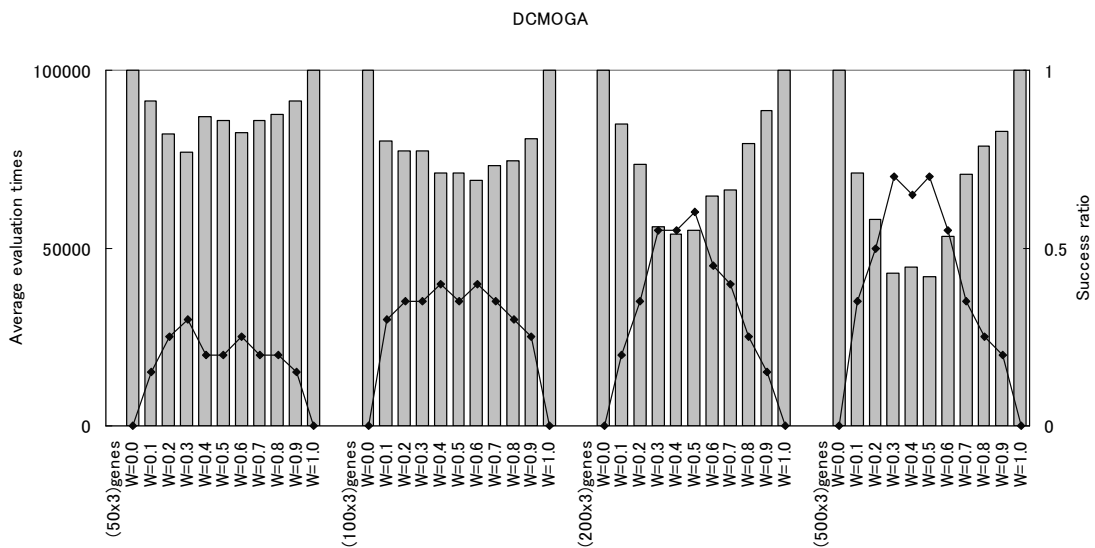


図 3-13. 熱ショック最適化問題における分散協力 GA での平均計算回数と成功確率

分散協力 GA は、重み関数と移住パラメータを設定する必要がある。これまでの実験結果より、移住パラメータの設定よりも重み係数の設定が最適化性能に与える影響が大きいことがわかるため、移住間隔を 20 世代、移住率を 20% に固定して、重み関数を変化させて最適化実験を行っている。分散協力 GA では、各目的関数で評価される個体群と重み関数で評価される個体群を用意し、それぞれの間で移住操作を行

う事で、線形結合 GA における局所収束と、分散 GA における個体群性質の偏りを軽減している。そのため、成功確率は他手法よりも高く、計算回数も少なくなっている。しかし、やはり重み係数の設定次第では、成功確率が大きく影響を受けている。分散協力 GA は他手法よりも、良好な結果を示したが、ユーザは個体数、重み係数、移住率、移住間隔などの多くの値を適切に設定しなければならない。

3. 4. 従来手法を用いた最適化計算の結果と問題点

従来手法を用いて、2つの目的関数を持つ大腸菌熱ショック応答シミュレーションの最適化計算を行った結果、最適化成功確率は70%近い結果を示す手法も見られたが、重み関数の重み係数の設定次第で、成功確率や平均計算回数は大きく影響を受けた。これらの手法では重み関数を用いて2つの目的関数を1つに統合して評価を行っているため、各目的関数の性質は探索に反映されにくくなる。また、重み係数の設定次第では、片方の目的関数の影響を強く受けてしまうことがあるため、他方の目的関数の性質を満たすことが困難になる。実際の使用に際しては、最適化性能に依存の大きい重み係数の設定の他にも、移住間隔や移住率、統合世代数などの多くのGAパラメータを適切に設定しなければならず、これがユーザの負担にもなっている。

4. 重み関数に依存しない 2 つの目的関数を対象とした最適化手法の提案：生存率 GA

前節では、既存の手法を用いて 2 つの目的関数を持つ大腸菌熱ショック応答の最適化問題を解いてきた。その結果、重み係数の設定に最適化性能が強く依存する場合が多くみられた。重み係数の決定においては、各目的関数のオーダ変数強度、偏差などのバランスを考慮して設定しなければ、良好な最適化結果を得ることは困難となり、この設定はユーザの負担にもなっている。

そこで、本研究では重み関数を用いずに、2 つの目的関数の最適化を行う手法として、生存率 GA を用いた手法を提案する。基本原理としては、1 世代おきに目的関数を入れ替えることで、子個体は双方の目的関数で評価された親個体の性質を継承する。この世代交代を繰り返す事で、両方の目的関数を満たす解の探索を試みる。しかし、一般的な GA で、目的関数の入れ替えを行っても、個体群の性質は、2 つの目的関数の間で振幅を繰り返すだけになってしまう。そこで、GA における各個体に寿命という概念を与え、寿命のルールに従って個体群に数世代にわたり生き残るアルゴリズムを定義する。寿命を導入して、個体が数世代にわたり個体群に生き残り親として選択されることで、新たに生成される子個体は、これまでの探索で発見された双方の目的関数の性質（個体群内に生き残っている数世代前の個体の性質、過去の探索履歴）を継承することができるようになる。

4. 1. 生存率 GA: Survival Ratio GA

個体に寿命の概念を用いることは、以前にも提案されている。エリート個体が個体群内に残留し続けることで起こる局所収束を、個体に寿命を設けることで軽減することが目的であった。しかし、生存率 GA は、それとは異なる目的で寿命という概念を GA に付加する[12]。個体群内の各個体は生存率により、個体は次世代に一定の確率で生き残るようにする(図 4-1)。その結果、個体群内には、過去の世代の個体が年代層を形成する。この年代層を含む個体を親にもつ子個体はそれらの性質を継承する。つまり、過去の探索履歴をデータベースのように個体群内に保持し、それらを参照した探索が可能となる。世代を繰り返すことで過去の探索個体が個体群内に蓄積されていくが、全個体数は生存率に従い次第に飽和する。ある世代の個体群について、初期個体数と同数の探索点(子個体)が生成され、残りは個体群内に蓄積された過去の探索履歴の数を示している。また、GA1 世代当たりの計算負荷は各世代で新たに生成される子個体の評価計算だけを行うので、生存率の設定に関わらず一定であり、従来の GA の計算負荷に等しい。一般的な GA では、個体群の優良性質を交叉操作などで失わないために、最優良個体を次世代に保存するエリート保存戦略[13]を用いることが多い。しかし問題によっては、保存された最優良個体(エリート個体)の性質に個体群全体の性質が収束してしまう場合がある。エリート戦略の中には、エリート個体の保存期間に制限をつけたりする手法も提案されているが、それらの設定もユーザに任されている。生存率 GA では、エリート保存戦略は用いないが、個体寿命の設定によって最優良個体は数世代にわたり個体群内に生き残り、その後は次第に淘汰されていくため、生存率パラメータの 1 つを設定するだけで、過去探索履歴の保存と優良個体の保存の両方を行うことができる。

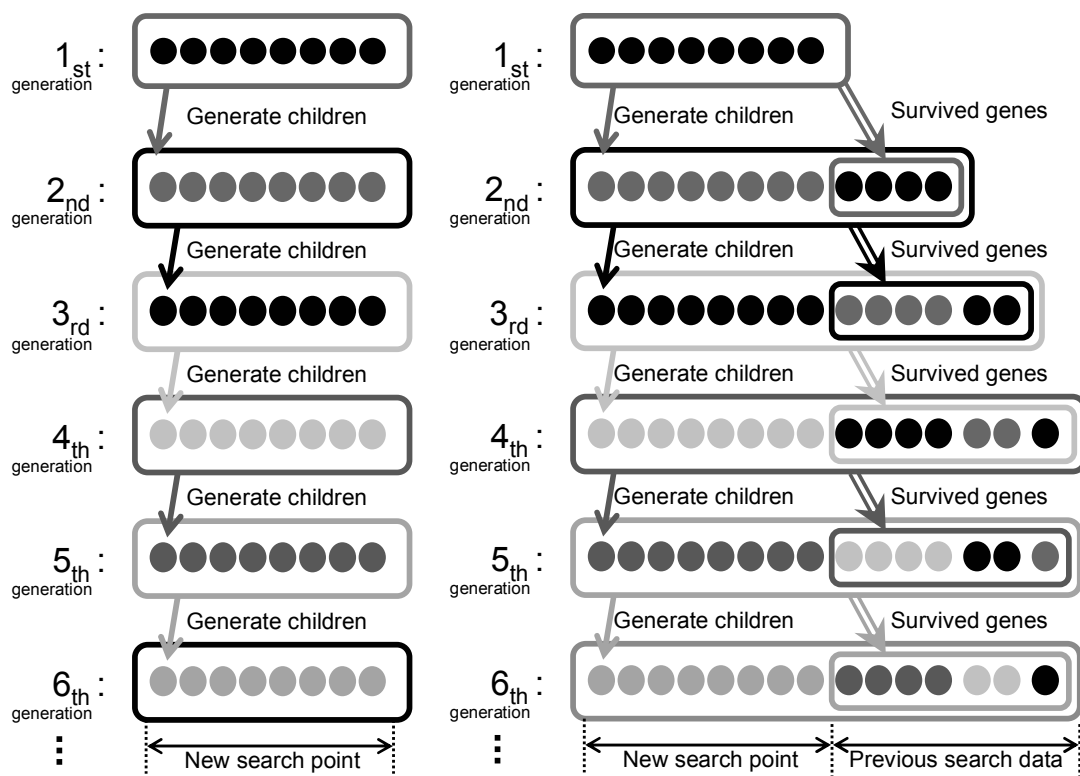


図 4-1. 従来 GA のアルゴリズム(左)と生存率 GA のアルゴリズム(右). 例として生存率 50%の場合

4. 2. 2つの目的関数最適化問題に対応する GA

4. 2. 1. 生存率 GA

2つの目的関数を同時に最適化するために、1世代ごとに2つの目的関数を入れ替えて個体の評価を行う(図 4-2)。すると、個体群内には、双方の目的関数で評価された個体が保持され、子個体は、過去の双方の個体性質を継承することが出来る。個体群内には、目的関数 1、目的関数 2 で評価された個体が存在しており、親個体はこれらの組み合わせで決定されるため、子個体群は「共に関数 1」「共に関数 2」「関数 1 と関数 2」のような組み合わせた性質を与えられることになる。つまりは、1つの個体群の中で、評価 1 を良くしようとするベクトル、評価 2 を良くしようとするベクトル、評価 1 と評価 2 を共に良くしようとするベクトルの 3 つの探索方向を持つことになる。これらの子個体は生存率で数世代間は個体群内に残留し、次世代の探索のヒントとなる。

また、この手法では、2つの目的関数をそのまま使っているので、Linear Combination GA のときのような重み係数の設定をする必要がない。親個体の選択も評価値ではなく、ランクで行うため、2つの目的関数の評価値のオーダ依存強度などを考慮する必要がない。Survival Ratio GA で設定すべき GA パラメータは個体数と生存率だけであり、これまでの他の手法に比べて設定すべきパラメータが少ないため、GA の最適化性能がパラメータ設定に依存することが少なく、ユーザの負担を軽減している。

GA に寿命の概念を取り入れる手法は、以前にも提案されている。しかし、その手法では、個体群内に局所最適解の性質を持った個体が生き残り続けることで、その他の性質の探索が十分に行えなくなる問題を軽減することを目的としている。つまり寿命により、個体の排除を指定している。そのため、過去の探索履歴も保持するが、母集団の個体数が増えることなく一定であるため、寿命の設定しだいで母集団内の探索個体数が次第に減少していく。本手法では、同じ寿命でも個体の生存を定義している点で、既存手法とは基本的な概念が異なり、探索個体は寿命の設定に関わらず一定の個数を保持している。

2つの目的関数を取り扱い、GAの個体に寿命を導入する手法は、他にもいくつか考えられる。そこで、本研究では比較のために以下のような世代交代ルールを定義した遺伝的アルゴリズムでの最適化も行った。

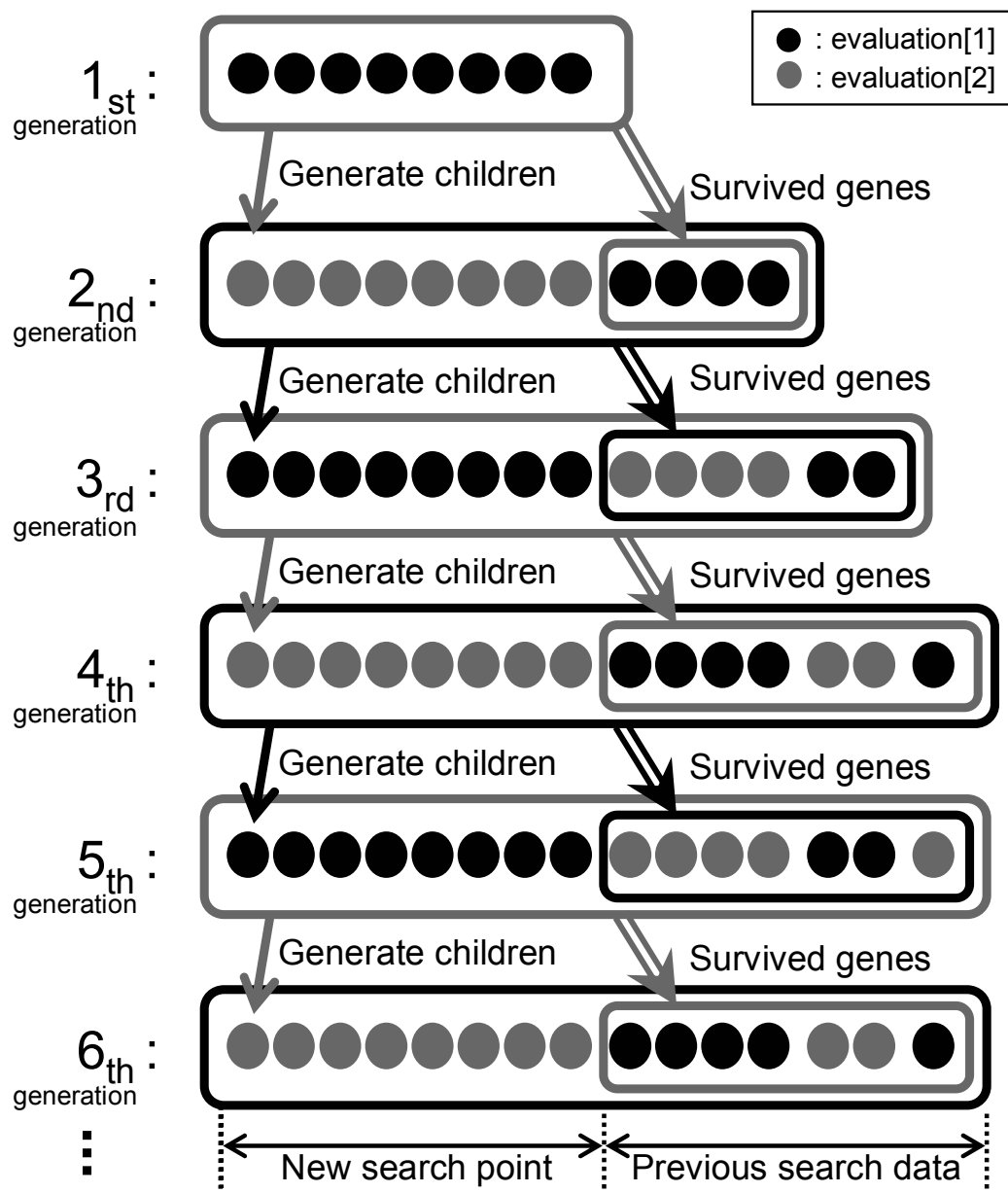


図 4-2. 2つの目的関数を持つ最適化問題への生存率 GA の対応. 例として生存率 50%の場合

4. 2. 2. 固定寿命 GA

本手法では、個体の寿命を生存率ではなく生存可能な世代数(寿命)で定義する。各個体は寿命の世代数だけ生き残り、子個体に性質を継承していく(図 4-3)。個体群の個体数は制限されておらず、寿命の許す限り個体群内に残留する。同時に、子個体は毎世代同じ数だけ生成される。そのため、探索開始から個体群内の個体数は増加し、その後ある値(個体数×寿命世代数)で飽和する。母集団は過去の探索履歴を保持し、子個体はこれらを継承することで、多様な性質を獲得する。

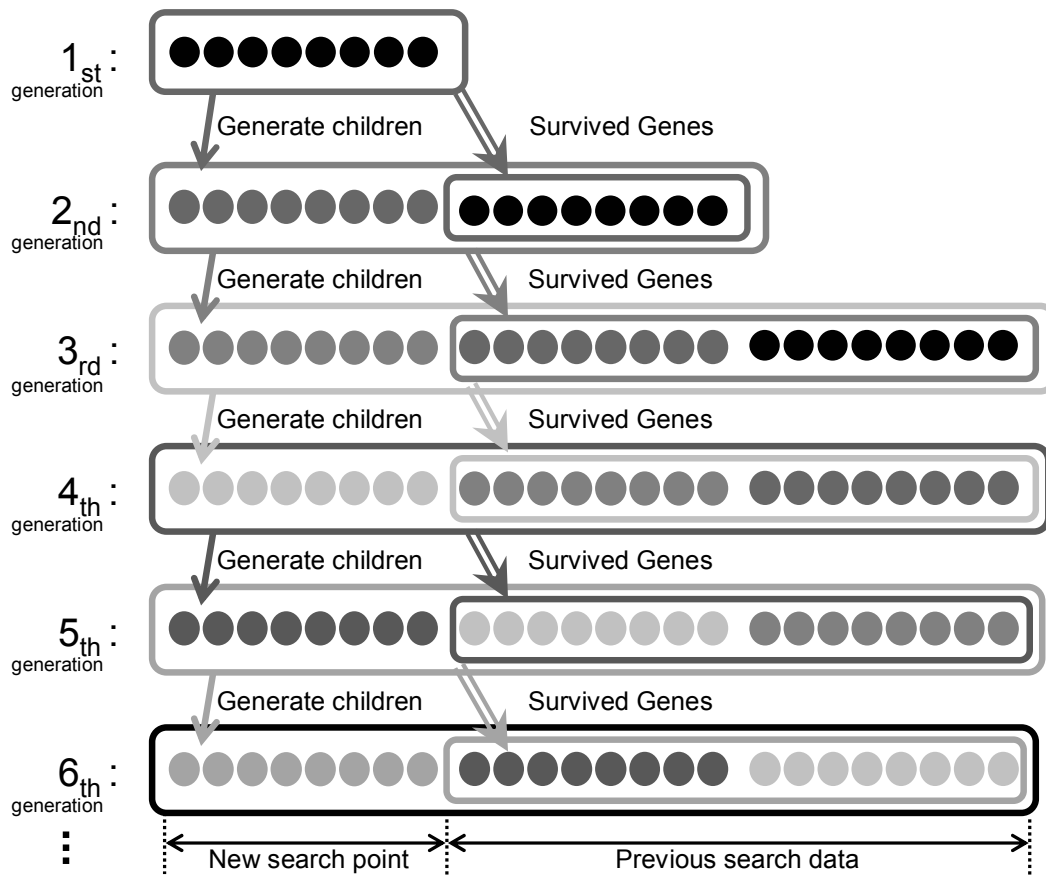


図 4-3. 固定寿命 GA のアルゴリズム(例として個体寿命が 3 世代のとき)

4. 2. 3. 個体の寿命を考慮した GA

この手法は、エリート個体が母集団内に生き残り続けることで起こる局所収束を軽減するために提案された手法であり、各個体に生存可能世代数を寿命で定義する(図4-4)[14]. 探索開始から個体群内では各個体は寿命の許す限りの世代だけ残留する. 淘汰や寿命で消滅した個体数だけ新しい子個体を生成し、これらが次の世代の探索点となる.

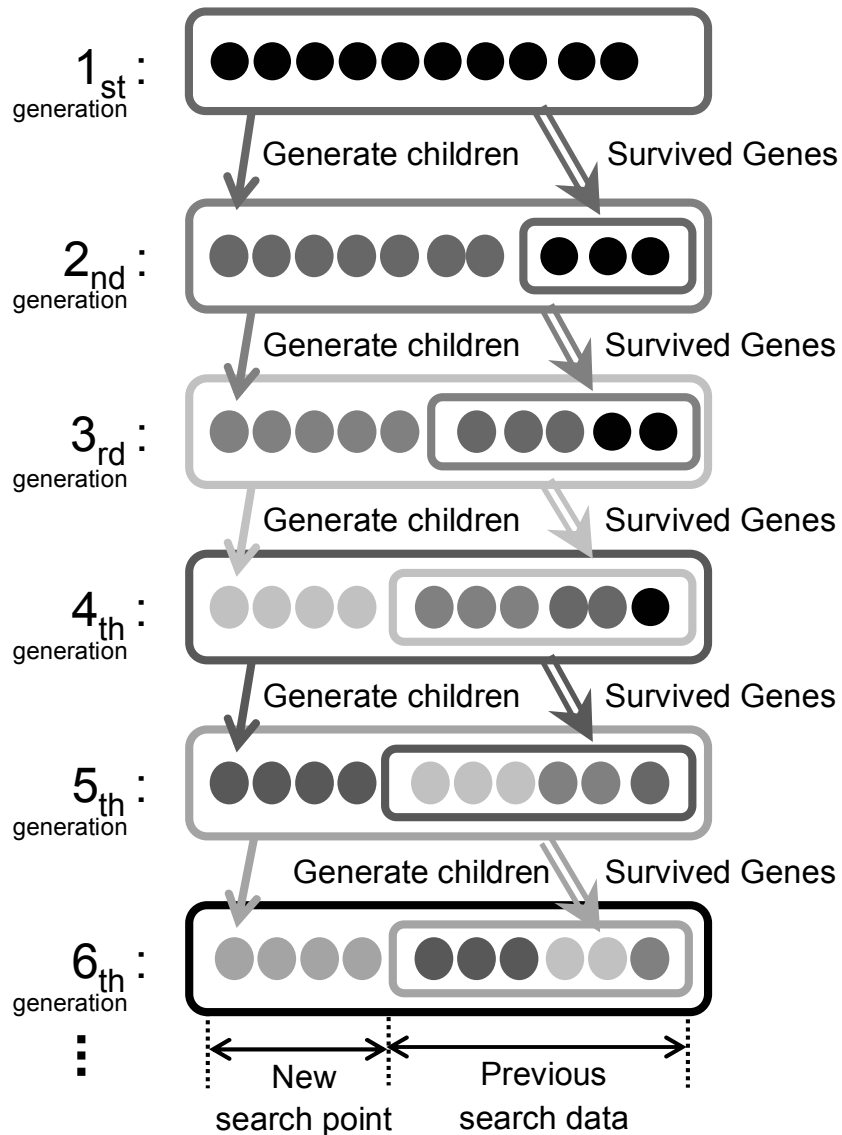


図 4-4. 個体の寿命を考慮した遺伝的アルゴリズムの流れ

4. 3. 寿命を導入した GA での大腸菌熱ショック応答の最適化

寿命を導入した遺伝的アルゴリズムの各手法を用いて、2 つの評価基準を持つ大腸菌熱ショック応答のパラメータ最適化問題を解き、最適化完了までの平均計算回数と成功確率の比較を行った。

4. 3. 1. 固定寿命 GA を用いた場合

図 4-5 は、固定寿命 GA を用いて、熱ショック応答シミュレーションの最適化計算を行ったときの平均計算回数と最適化成功確率を示している。個体群内の個体数を 50, 100, 200 としたとき、個体の寿命を 0~10 世代と 1 世代刻みで設定した各場合において最適化実験を行っている。各試行の設定条件は表 4-1 の通りである。

固定寿命では、寿命が 2~3 世代のときは、個体群内に過去の探索履歴が蓄積され、子個体はそれらを継承することで、寿命を用いない手法(寿命が 1 世代の場合に等しい)よりも、最適化成功確率は高くなった。しかし、寿命が 5 世代以上になると、個体群に過去の探索履歴が過剰に蓄積されるため、子個体が優秀な性質を継承できる可能性が低くなる。そのため最適化成功確率は低下した。

表 4-1. 固定寿命 GA の条件設定

線形結合 GA

個体数:50, 100, 200 個体

寿命:0~10 世代 (1 刻み)

交叉:UNDX

試行回数:20 試行

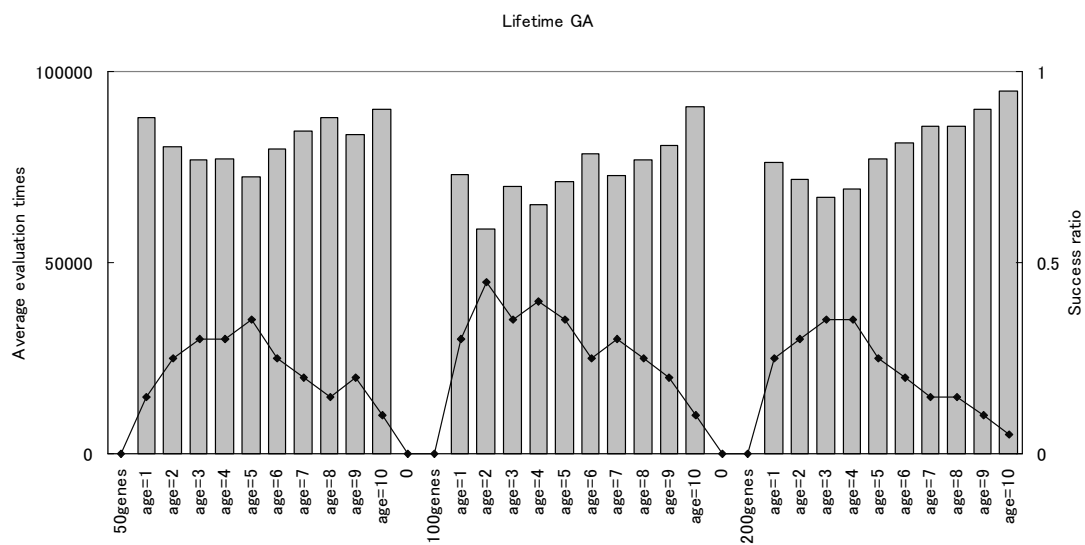


図 4-5. 熱ショック最適化問題における固定寿命 GA での平均計算回数と成功確率

4. 3. 2. 個体の寿命を考慮した GA を用いた場合の結果

図 4-6 は、個体の寿命を考慮した GA を用いて、熱ショック応答シミュレーションの最適化計算を行ったときの平均計算回数と最適化成功確率を示している。個体群内の個体数を 50, 100, 200 としたとき、個体の寿命を 0~10 世代と 1 世代刻みで設定した各場合において最適化実験を行っている。各試行の設定条件は表 4-2 の通りである。ある個体が次世代に生き残るルールは提案手法の文献[14]を使用した。

- 個体が設定した年齢に達したら、その個体を淘汰する。
- 各個体の適応度 f_i に対し、個体 i が次世代に生き残る確率 E_i を以下の式で定義する。

$$E_i = N \left(\frac{f_i}{\sum_{j=1}^N f_j} \right), \quad N: \text{個体数}, \quad i=1,2,\dots,N$$

- E_i の整数部分 b_i から、各個体の増殖数を決める。
- b_i の合計は一般的に個体数 N よりも少ないので、 E_i の小数部分を大きい順に並べて、 N から b_i の合計を引いた数 c だけ上から順に切り上げて 1 とする。
- すなわち、 $N = \sum_{i=1}^N b_i + c$ が成立し、個体数は一定に保たれる。
- その後、交叉を行う。

「寿命を考慮した GA」では、過去の個体も母集団に残留するが、全個体数を固定しているため、過去の履歴が残るほど、新たに生成される子個体数は減少し、十分な空間探索が困難になっていく。GA の設定で初期個体数を増やしても、個体選択と淘汰のバランス次第では、探索初期に子個体数が急激に減少する。個体の初期分布が悪く、初期探索で良好な結果が得られなければ、その後の局所探索に支障をきたす場合がある。

ため、大域探索から局所探索に推移するスピードをユーザが適切に調整しなければ

ならない。

表 4-2. 個体の寿命を考慮した GA の条件設定

個体の寿命を考慮した GA

個体数:50, 100, 200, 500 個体

寿命:0~10 世代 (1 刻み)

交叉:UNDX

試行回数:20 試行

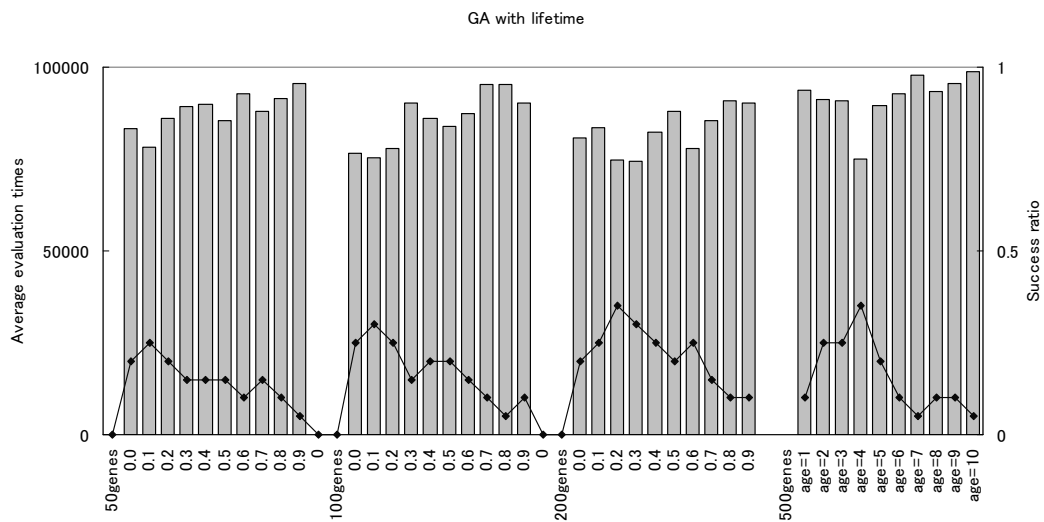


図 4-6. 熱ショック最適化問題における、個体の寿命を考慮した GA での平均計算回数と成功確率

4. 3. 3. 生存率 GA を用いた場合の結果

図 4-7 は、生存率 GA を用いて、熱ショック応答シミュレーションの最適化計算を行ったときの平均計算回数と最適化成功確率を示している。個体群内の個体数を 50, 100, 200 としたとき、個体の寿命を 0~10 世代と 1 世代刻みで設定した各場合において最適化実験を行っている。各試行の設定条件は表 4-3 の通りである。

生存率 GA では、各個体が生存率に従って個体群内に生き残っていくと同時に、生存率に従って個体群内から淘汰されていくことで、現在の世代に近い個体ほど親として選択されやすい世代層を形成する。また、2 つの目的関数を交互に繰り返す事で、これまでの両者の探索履歴を子個体が参照するため、両者の性質を組み合わせた性質が継承される。

生存率 GA の成功確率は、他手法と比べて高い結果を示した。また、最適化完了までの平均計算回数も少ない。生存率の値は高ければ高いほど、過去の探索履歴を多く保持できる。しかし、高すぎると過剰に探索履歴を保持しすぎて、子個体が優秀な性質を継承できる可能性が低下する。

表 4-3. 生存率 GA の条件設定

生存率 GA
個体数: 50, 100, 200 個体
生存率: 0.0~1.0 (0.1 刻み)
交叉: UNDX
試行回数: 20 試行

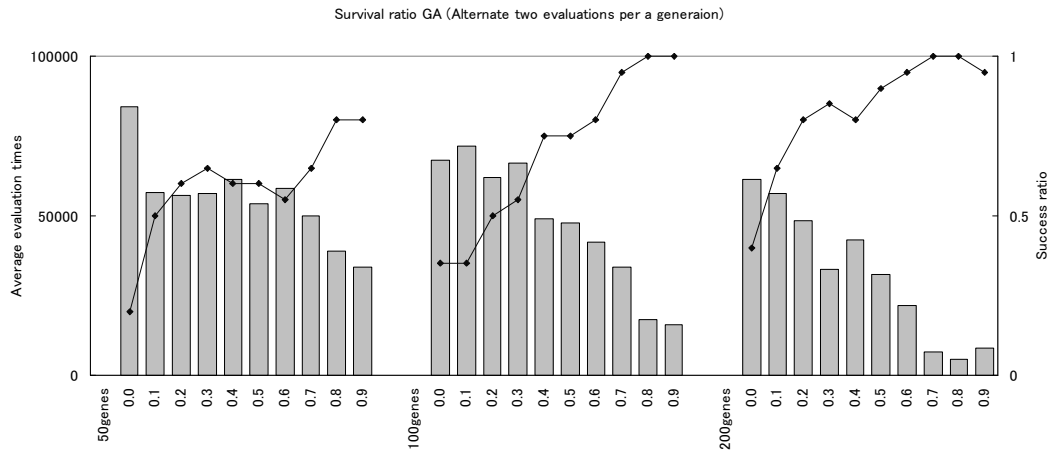


図 4-7. 熱ショック最適化問題における生存率 GA での平均計算回数と成功確率

4. 4. 生存率 GA の生存率パラメータに関する考察

生存率の設定が低すぎる時は、個体群内に十分な過去探索個体を蓄積しないため、過去の散策履歴や、双方の混合格質を継承できる子個体が生成されにくくなる(図 4-8). 逆に生存率の設定が高すぎると、個体群内に過剰な個体が蓄積され、子個体が優秀な親個体の性質を継承できる可能性が低くなる(図 4-9)

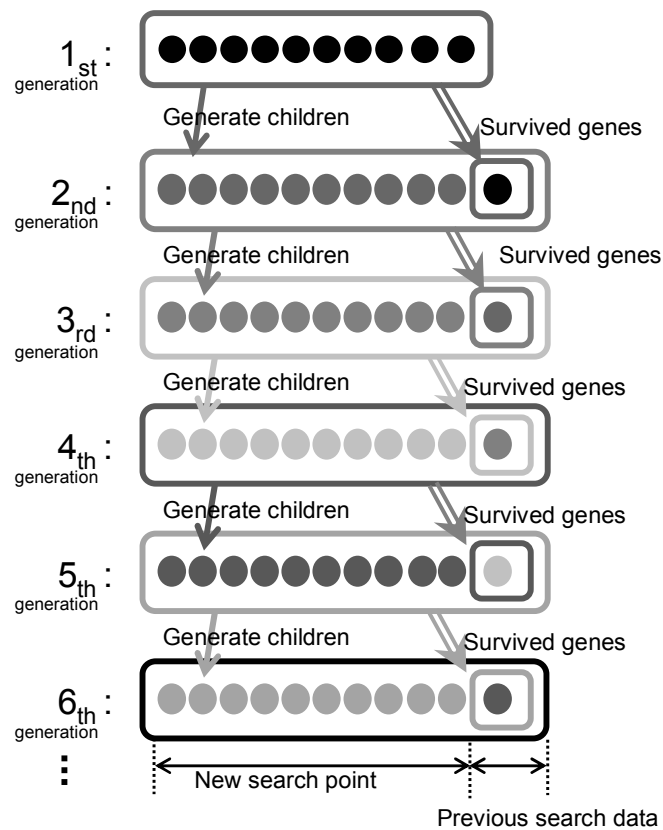


図 4-8. 生存率 GA において、生存率の設定が低すぎる場合
(生存率 0.0 の場合は従来の Simple GA と等しくなる)

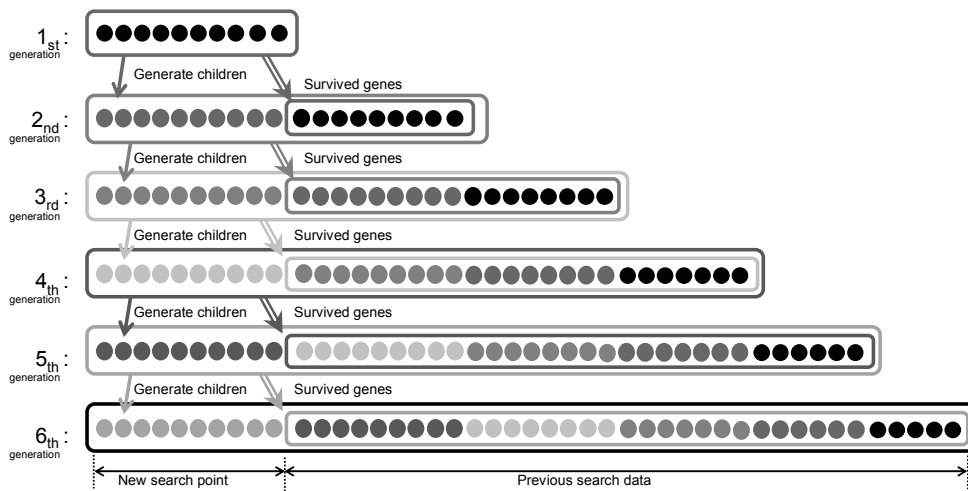


図 4-9. 生存率 GA において、生存率の設定が高すぎる場合

4. 5. 既存手法との比較

4. 5. 1. 最適化に要した平均計算回数と成功確率

2つの目的関数を持つ大腸菌の熱ショック応答シミュレーション最適化問題を、従来手法の線形結合 GA と分散協力 GA, 提案手法の生存率 GA を用いて解いた場合の最適化成功確率と平均計算回数の比較を行った。

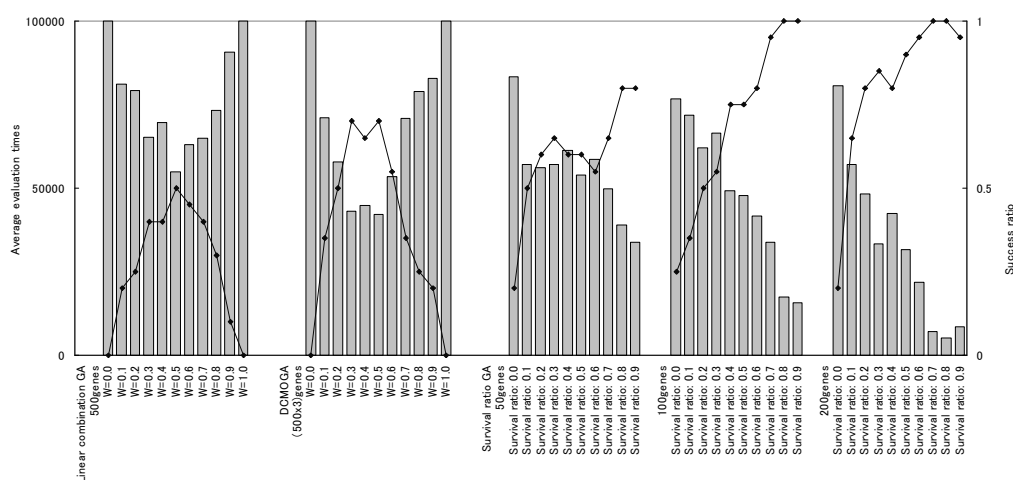


図 4-10.大腸菌熱ショック応答最適化問題における各手法の成功確率と平均計算回数 (評価 1:ピークの高さ, 評価 2:定常値)

(図 4-10)は、各 GA による探索が終了条件を満たすまでにかかった平均評価計算回数と成功確率のグラフである。個体の評価に重み関数を用いる線形結合 GA と分散協力 GA では、重み係数の設定で成功確率が大きく影響を受けている。また、重み係数の値の設定が良好な場合では、50%以上の成功確率を示し、平均計算回数も少なくなっているが、重み係数の値が適切でなければ、成功確率は急激に低下し、計算回数も増大している。また、DCMOGA では、複数の個体集団を使用するため、個体の多様性を維持し、それぞれの目的関数において得られた優秀な個体を移住させることで、重み関数で評価される個体群での探索を支援している。そのため、複数個体群のため、GA の 1 世代あたりの計算負荷は線形結合 GA に比べて倍増して

しまうが、高い成功確率と少ない計算回数を示した。

Survival Ratio GA の平均計算回数は他手法と同程度あるいは少なく、最適化成功確率は他手法よりも高い結果となった。生存率パラメータの設定においても、生存率が 0.2 以上では成功確率が 50%以上となっている。個体数 100 個体と 200 個体の場合で、生存率が 0.8 以上のときは、100%近い成功確率を示し、平均計算回数も従来手法よりも少なくなる結果を示し、良好な最適化性能を示した。

4. 5. 2. エリートの個体分布推移

図 4-11 は生存率 GA における，個体群内の最優良評価個体をもつ 2 つの評価値の世代推移を示している．一般的な GA では 1 つの評価基準に従い最適化が進むため，単調減少のグラフとなる(最適化問題が評価値の最小化の場合)．これに対して生存率 GA では，1 世代ごとに 2 つの評価基準を交互に入れ替えて個体の評価・並び替えを行うため，評価値の世代推移曲線は図 4-11 のような振動を繰り返す．2 つの評価基準がある場合，両方の評価値が共に満たす性質をすぐに発見することは困難であるため，片方ずつの性質を交互に改善しながら探索が進む．

個体群には過去の探索履歴として，これまでに発見された双方の性質が蓄積されているので，次世代で新しく生成される子個体がこれらを継承することで，それぞれの評価値を徐々に改善していく．このとき，子個体は親の選択組み合わせにより，同一の評価個体の性質を受け継ぐ場合と，双方の評価個体の混合格質を受け継ぐ場合がある．こうして，Survival Ratio GA の中では，3 通りの性質を持つ個体が生成されることによって多様な個体性質を保持している．

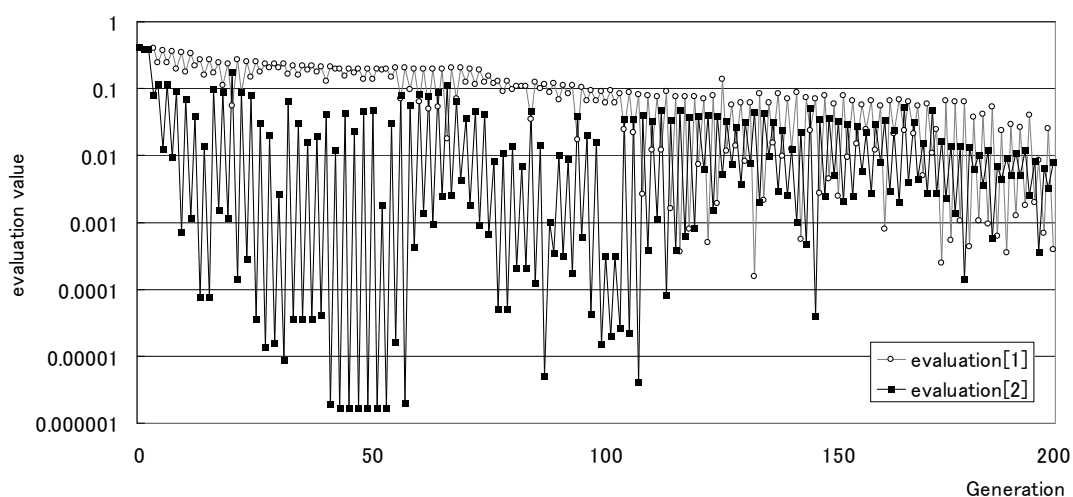


図 4-11. 2 つの目的関数の世代推移 (例として初期個体数:100, 生存率 0.7 の試行)

4. 5. 3. 個体分布図

(図 4-12)は 2 つの目的関数の評価値を軸としたときの、全個体分の布図である。(最適化開始から完了までのデータをプロットしている)。また、図 4-13 は個体群内の最優良個体の評価値に対する世代推移である。

線形結合 GA では、重み係数の値を適切に調整することは困難であるため、2 つの評価基準のうち、個体の初期分布によって、まず片方の評価基準を改善する方向に探索が進む。そして、その後他方の評価基準も満たすように個体性質の変化が起こる傾向が多くみられた。個体群が片方の評価基準を満たす性質を獲得したとき、もう片方の評価基準を良好に満たす性質は異なる空間に位置するため、この時点で個体群が局所に完全収束してしまう前に他方の探索を行わなければならない。しかし、単一個体群である場合や重み係数の設定が適切でないときは、個体群がその局所収束している状態を脱却して、その他の範囲を探索することが困難となるため、一方の評価基準しか満たさない性質のまま個体群が収束してしまい探索が終了する。

この問題点を軽減するのが複数個体群を用いた分散協力 GA である。重み関数を評価とする個体群では、個体性質が局所に収束してしまう前に、それぞれの評価基準で評価された個体を各母集団から取り込み、個体の多様性の確保を図っている。個体分布のグラフでも、SOGA 個体群はそれぞれの目的関数を満たすように探索を進め、これらの性質を重み関数での評価が行われている MOGA 個体群に移住させることで、多様性を獲得し、MOGA 個体群は 2 つの探索軸の間を保管するような探索を進める。しかし、分散協力 GA も重み関数を用いるため、まず、個体の分布によってどちらかかの性質に収束し、その後他方の評価基準を満たすように探索が進む傾向がみられる。しかし、SOGA が絶えず各評価基準における優秀な個体の提供を受けられるため、単一個体群の線形結合 GA よりは、個体群が局所収束して探索能力が著しく低下する場合も少なく、このため、最適化成功確率も線形結合 GA よりは高くなっている。

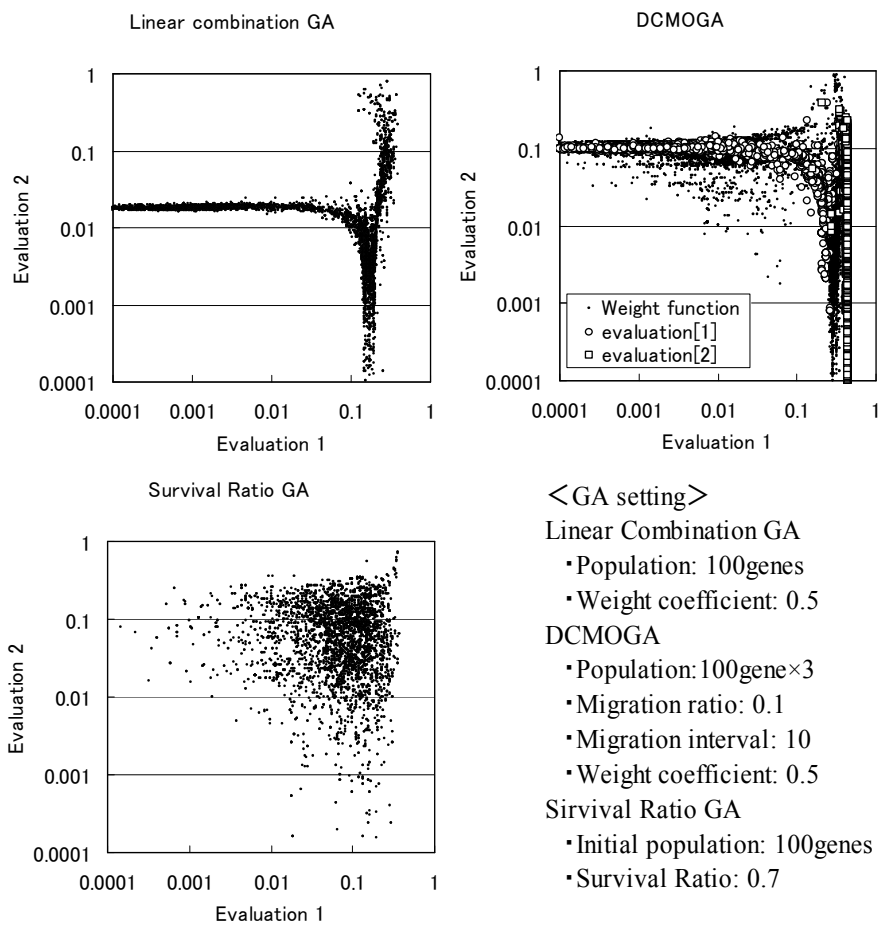


図 4-12. 各 GA において, 2 つの目的関数の評価値を軸としたときの個体分布

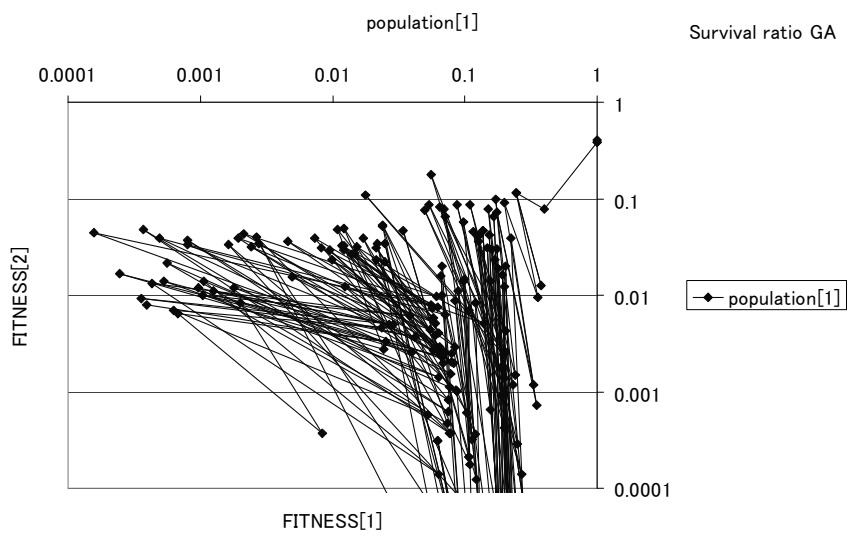
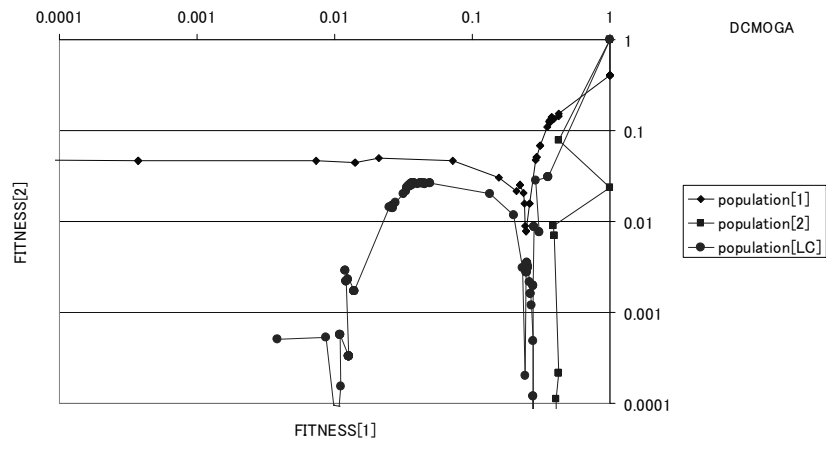
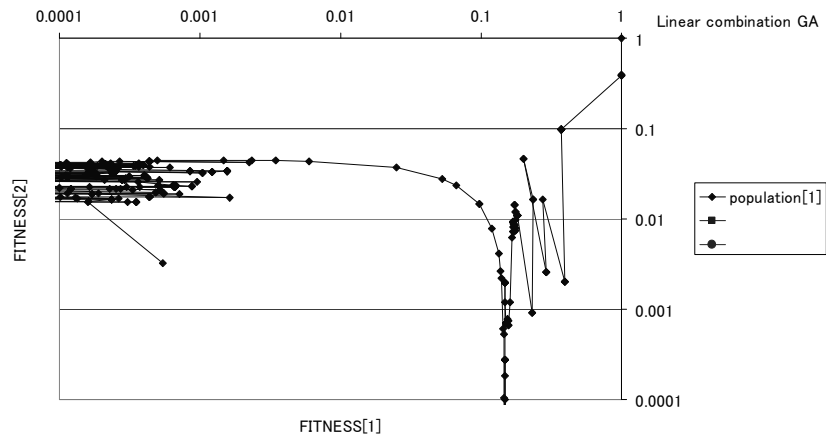


図 4-13. 各手法におけるエリート個体の目的関数の世代推移

生存率 GA では、前世代の個体が母集団内に蓄積され、新たに生成される子個体はこれらの過去の探索結果も継承した性質を得ることができる。また 2 つの目的関数に対応して、個体の評価方法を 1 世代おきに入れ替えることで、個体群内にはこれまでに各評価基準で評価された個体が蓄積し、子個体はそれらの双方を継承し多様な個体性質を保持する。また、1 世代おきに評価関数が置き換わるため、個体群の探索方向(個体進化を進める方向)も毎世代変わり、個体群は 2 つの探索軸を往来する挙動を示す。生存率 GA の手法では、各評価関数をそのまま使用しているため、この往来によりそれぞれの評価基準に対する良好な性質を探索すると共に、新しく生成される子個体は双方の性質を組み合わせた性質を得ることができるため、個体性質の多様性を十分に保つことができ。従来手法よりもより広い範囲を探索し、より多様な性質の解を見つけ出すことができている。

4. 5. 4. 各最適化手法における平均計算回数と成功確率の関係

(図 4-14)は、これまでに用いた手法それぞれにおける最適化成功確率と、最適化に成功した試行の平均計算回数の関係をプロットした分布図である。各手法において、GA の設定条件を変えて最適化実験を行い、それぞれの条件での成功確率と計算回数をプロットしている。GA の設定条件は前節までの検証実験で用いたものに等しい。最適化計算手法としては、成功確率が高く、平均計算回数が少ない手法ほど良好な最適化結果を示すといえる。つまり、このグラフにおいては左上にあるプロットほど良好な結果であるといえる。また、プロットが分散せずに集中している手法ほど、GA の設定条件が最適化計算の性能に与える影響が少ないといえる。一般的な反復的最適化手法では、平均計算回数と最適化成功確率はトレードオフとなる場合が多いが、生存率 GA では、計算回数も少なく成功確率も高い結果を示している。生存率パラメータの値の設定により、成功確率は影響を受けているが、全体的にみても、成功確率は他手法よりも高く、平均計算回数は手法と同程度あり、大腸菌熱ショック応答最適化問題において良好な最適化性能を示している。

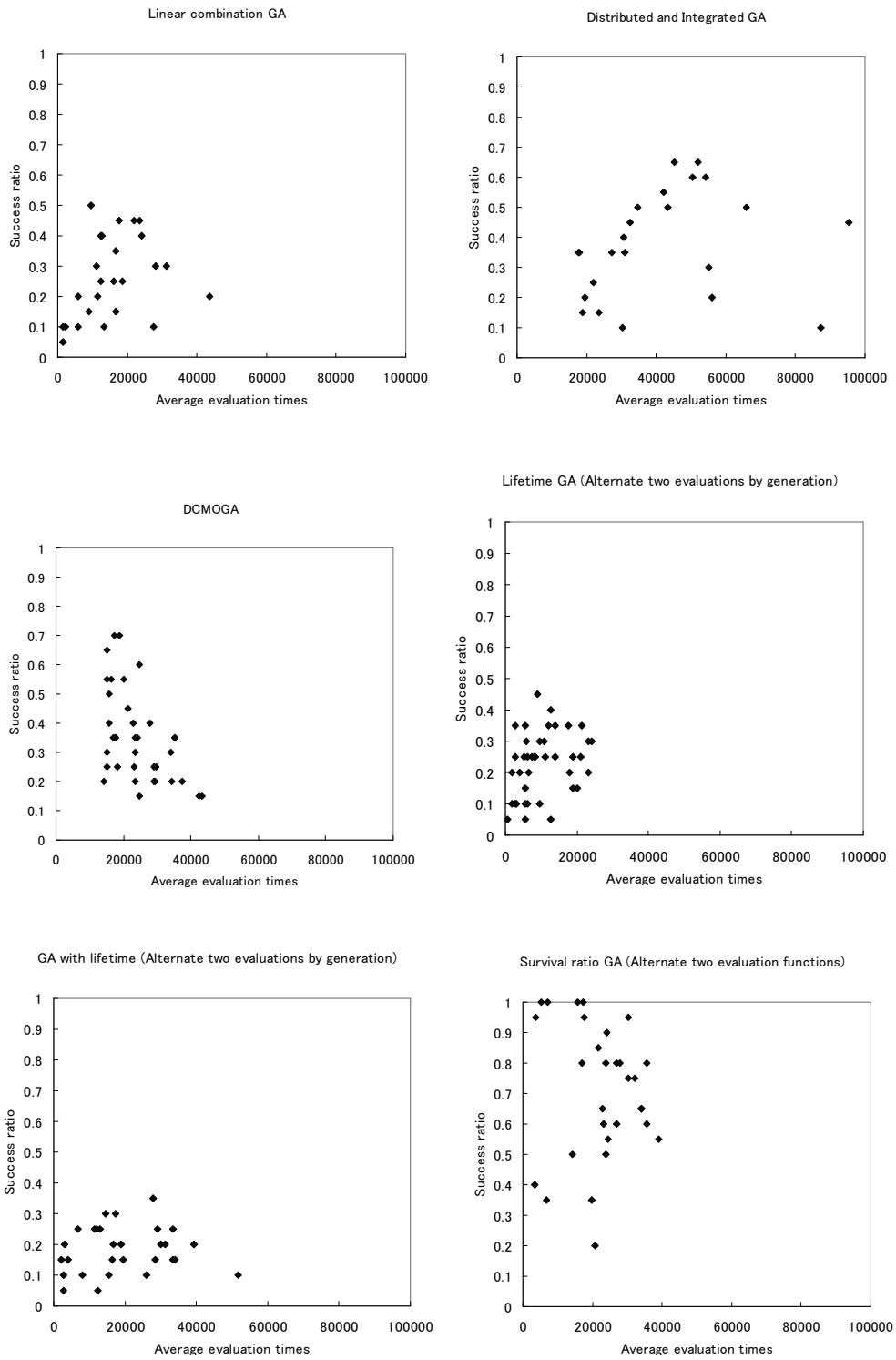


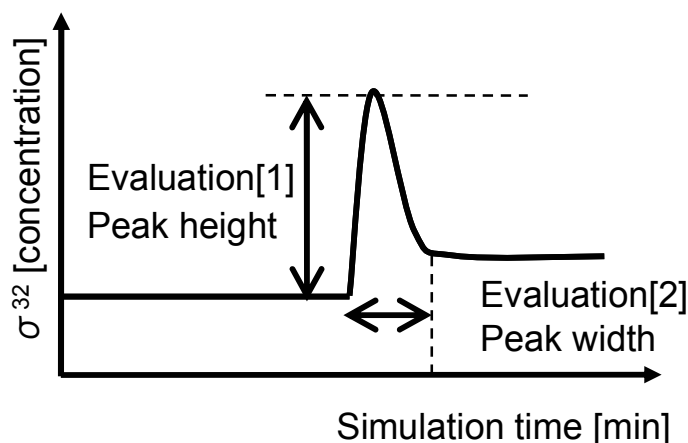
図 4-14. 各手法における最適化成功確率と平均計算回数との関係

4. 6. 単位の異なる 2 つの評価基準を持つ大腸菌熱ショック応答の最適化

ここまでの検証では、生存率 GA は他手法よりも良好な最適化性能を示した。そこで次に、対象とする評価基準が変わっても生存率 GA は良好な最適化性能を示すことができるのかを検証するために、同じ大腸菌の熱ショック応答シミュレーションを対象とするが、目的関数の単位が異なる評価基準を設定して、検証実験を進めた。一般的に、目的関数が変わると、GA の個体群が探索する空間の形状が変わるため、これまでの最適化手法が必ずしも同じような結果を示すとは限らない。また、2 つの目的関数の単位が異なる問題に重み関数を用いる場合には、これらの値のオーダーや変数強度などを考慮して適切な設定をしなければならないため、ユーザの負担になっている。生存率 GA を用いた手法では、重み関数は用いず、個体の優劣の決定は目的関数の値ではなく、目的関数で並び替えをされた際の順位(ランキング)であるため、目的関数のオーダーや強度を考慮する必要もない。

4. 6. 1. 単位の異なる 2 つの評価基準を持つ大腸菌熱ショック応答シミュレーション

評価基準は、 σ^{32} のピーク挙動における「ピークの高さ」と「ピークの幅」である(図 4-15)(表 4-4). この 2 つの評価基準の単位は、濃度と時間であるため、2 つの目的関数の依存強度やオーダが先程の最適化問題とは全く異なる探索空間となる。また、これら 2 つの目的関数は、共にピーク波形に関連する基準であるため、相関関係をもつ。



< Peak height >

$$Evaluation[1] = \frac{|\sigma^{32}[\text{simulation}] - \sigma^{32}[\text{target}]|}{|\sigma^{32}[\text{acceptable value}] - \sigma^{32}[\text{target}]|}$$

(0.0 ≤ Peak height ≤ 200.0)

(0.0 ≤ Evaluation[1] ≤ 1.0)

< Peak width >

$$Evaluation[2] = \frac{|\sigma^{32}[\text{simulation}] - \sigma^{32}[\text{target}]|}{|\sigma^{32}[\text{acceptable value}] - \sigma^{32}[\text{target}]|}$$

(0.0 ≤ Peak width ≤ 48.0)

(0.0 ≤ Evaluation[2] ≤ 1.0)

図 4-15. 大腸菌熱ショック応答シミュレーションにおける 2 つの目的関数 (評価 1:ピークの高さ, 評価 2:ピークの幅)

表 4-4. 大腸菌熱ショック応答最適化問題における評価値の許容範囲と終了条件(ピークの高さ・ピークの幅)

目的関数	目標値	許容範囲	終了条件
ピークの高さ [個/cell]	40.0	0.0~200.0	40.0 ± 2.0
ピークの幅 [min]	24.0	0.0~48.0	24.0 ± 0.48

4. 6. 2. 単位の異なる2つの目的関数最適化を各GA手法で解いた結果

図4-16は、生存率GAを用いて、熱ショック応答シミュレーションの最適化計算を行ったときの平均計算回数と最適化成功確率を示している。線形結合GAでは500個体で、重み係数の値を0.0~1.0までに変化させたときの結果。分散協力GAでは、各個体群の個体数が500個体で、重み係数の値を0.0~1.0までに変化させたときの結果。生存率GAでは、個体数が50, 100, 200個体の各場合において、生存率を0.0~0.9まで変化させたときの結果で示している。

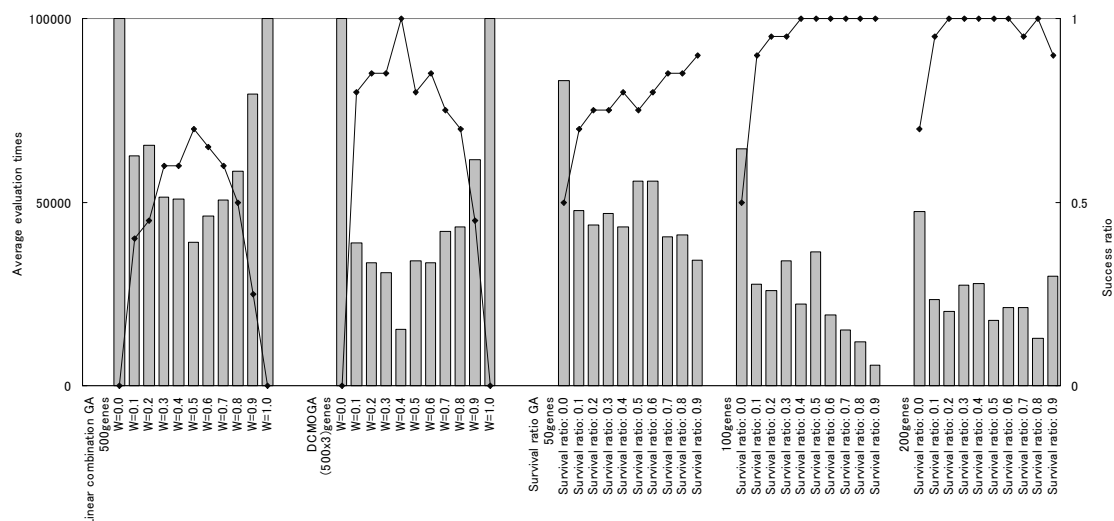


図4-16. 大腸菌熱ショック応答最適化問題における各手法の成功確率と平均計算回数（評価1:ピークの高さ，評価2:ピークの幅）

線形結合GAや分散協力GAでは、重み係数の設定が最適化成功確率に大きく影響する。また、個体群の個体数の設定が成功確率に大きく影響を与えた。生存率GAは、生存率値の設定により成功確率は影響を受けるが、平均して他手法よりも高い成功確率を示した。成功確率と平均計算回数はトレードオフとなることが多いが、生存率の値の設定が良好な場合では、平均計算回数は他手法と同程度だが、高い成功確率を示した。

また、各 GA の設定条件に関して、分散協力 GA などでは、最適化性能への依存の大きい重み係数に加えて、個体群の局所収束を司る移住間隔や移住率などの値も適切に設定する必要がある。これに対し、生存率 GA では生存率の設定だけで使用でき、図 4-16 からも生存率の設定条件が最適化性能に与える影響も他手法に比べても小さいため、ユーザは比較的フレキシブルな値の設定で、良好な最適化性能を得ることができる。

4. 7. 生存率 GA による熱ショック応答の最適化計算効率の向上

従来の最適化計算手法を用いた場合において、熱ショック応答シミュレーションの未知パラメータの探索範囲が広がると、最適化の成功確率が低下し、最適化に必要な反復計算回数も多く、効率的に目的関数を満たす性質を見つけ出すことは困難であった。そこで、2つの目的関数を定義し、この問題をより効率的に解くために生存率 GA を提案し、最適化計算を行った。

生存率 GA を 2 つの目的関数を持つ熱ショック応答最適化問題に使用したところ、従来手法と比べて最適化成功確率は約 2~3 倍、平均計算回数も半分以下で双方の目的関数を満たす性質を見つけ出すことができた。

生存率 GA でユーザが使用に際して設定しなければならないパラメータは個体数と生存率の 2 つのみであり、他手法に比べて少ない。GA では各最適化手法が十分な性能を発揮できるように、事前に各パラメータの値を適切に調節する必要がある。特に、2 つの目的関数を対象とする問題では、最適化性能に大きな影響を与える重み関数の重み係数の適切な設定が必要である。生存率 GA では重み関数を用いずに、設定が必要なパラメータ数も減らすことで、この負担を軽減している。また、2 つの目的関数の単位が異なる場合では、それぞれのオーダ関数間の強度なども考慮しなければならないため重み係数の設定はより複雑になる。4. 6. では、大腸菌の熱ショック応答シミュレーションにおいて、物質濃度と時間の 2 つを目的関数とする問題を設定して、各手法を用いた最適化計算を行い、その結果を比較した。重み関数を用いる手法では、各目的関数の値を用いて個体の優劣の決定を行っているが、生存率 GA では目的関数の値の順位(ランキング)を個体の優劣の判定基準としているため、目的関数のオーダ強度なども考慮せずに使用できる。また、生存率 GA では、1 世代おきに個体を評価する目的関数を交互に入れ替えるため、個体群は 2 つの探索軸を交互に往来しながら探索を進める。そのため、従来手法よりも広い範囲を探索することができ、高い最適化成功確率を示した。

5. 2つの評価基準を持つショウジョウバエ概日 リズムシミュレーションの最適化

前章まででは、一過性のピーク挙動を持つ生物シミュレーションを対象として、最適化実験を行ってきた。本章では、生物反応に特徴的なパターンである周期的な挙動の変化を示す概日リズムシミュレーションを対象とした検証も行った。概日リズムは、生物の体内時計の周期を司る反応としても有名であり、多くの生物において24時間の周期変化を示す。周期的な挙動を示す反応経路の最適化においては、実際の挙動と同じ周期や振幅を示す解を探索する以前に、恒常的な周期変化を維持できる解が存在する範囲は、探索範囲に比べて非常に狭く、これが最適解の発見を困難にしている原因の1つにもなっている。

5. 1. ショウジョウバエの概日リズムシミュレーション

概日リズムとは、約 24 時間の周期をもつ生物の自律的リズムである。この機構は多くの生物が有しており、夜と昼の生理学的挙動全般を制御している。概日リズムの周期変動は、複数の蛋白質(PER, TIM, dCLK, CYC)による多重のフィードバック制御により発現する(図 5-1)。また、最適化が必要な 47 個の動力学パラメータの値を変えるとさまざまな挙動を示す(図 5-2)。本反応では周期が 24 時間に近く、周期波形ができるだけ低い濃度で振幅を維持できるほど、実際の反応挙動と近いといえる。[15], [16]

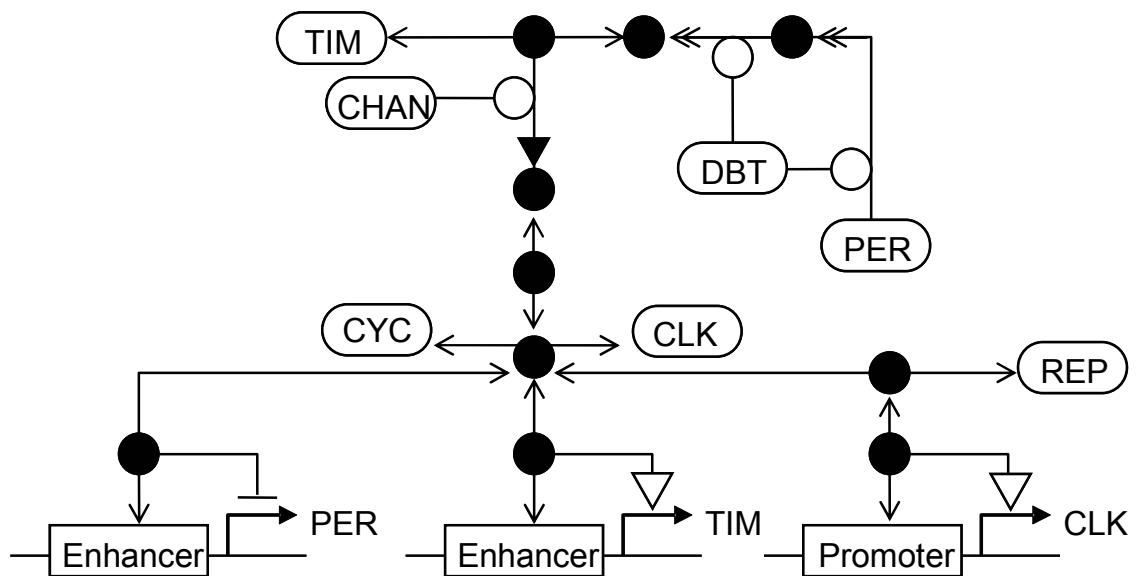


図 5-1. ショウジョウバエの概日リズムの反応経路

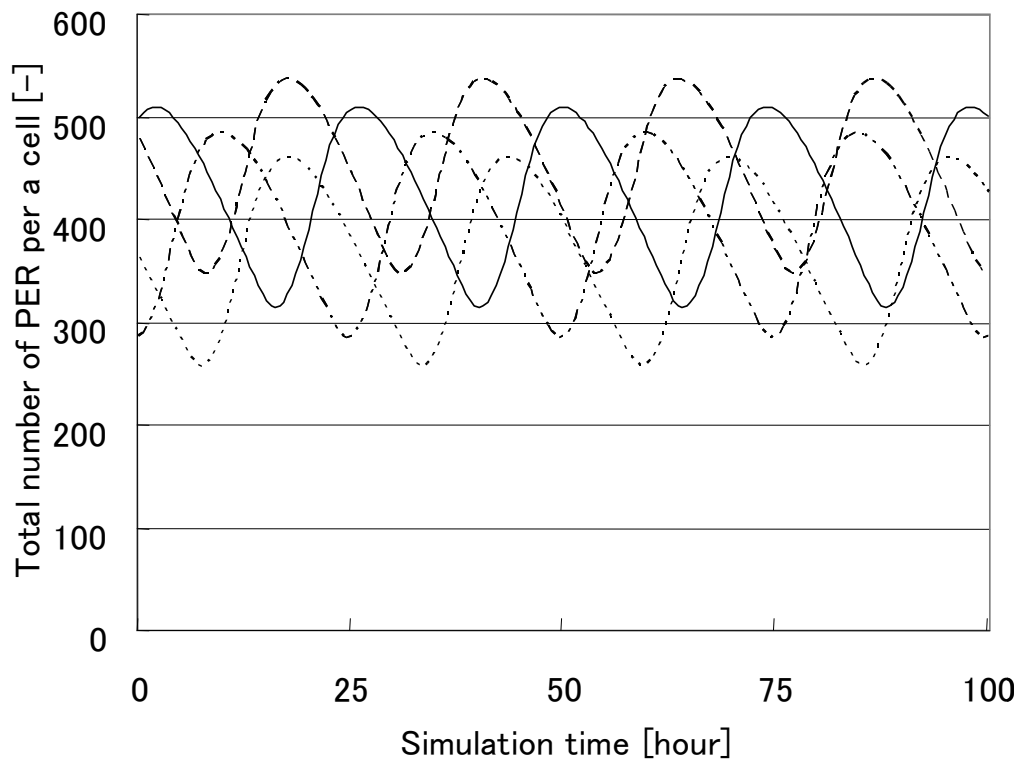


図 5-2. 概日リズムのシミュレーション結果. 最適化が必要な 47 個の動力学パラメータの値を変えるとさまざまな挙動を示す.

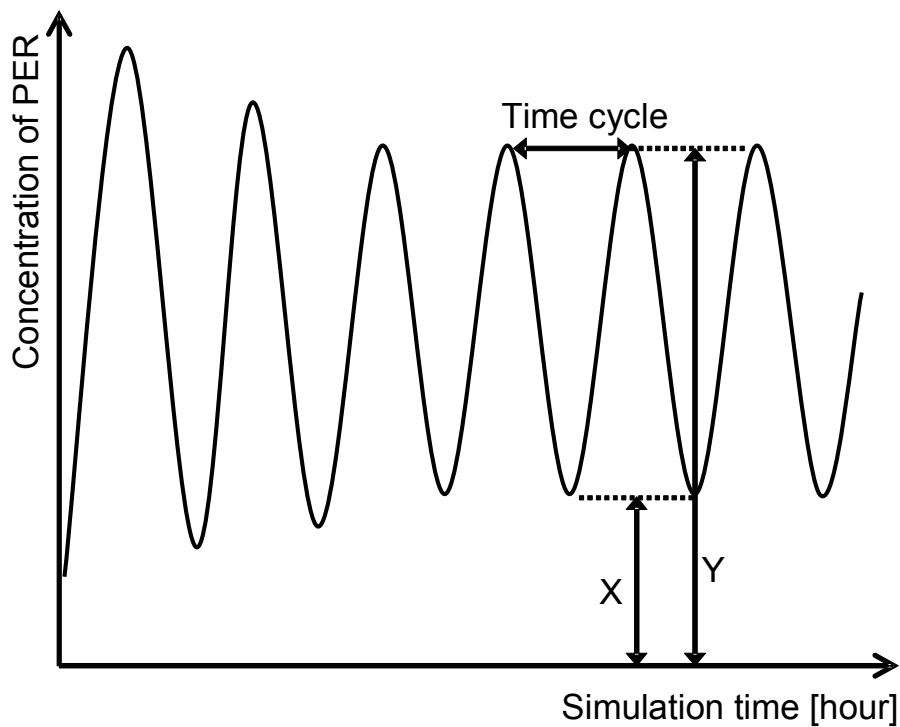
5. 1. 1. 概日リズムにおける2つの評価基準

本問題における2つの評価基準は、「周期が24時間との誤差」と「相対振幅」である(図5-3)。周期は24時間に近く、相対振幅は値が小さいほど、実際の挙動に近くなる。シミュレーションでは適当な物質初期濃度から計算を始めるため、開始直後は挙動が安定しない。そこで、シミュレーション時間を十分に長くして、各物質濃度が安定したところで、周期波形の判定を行う。本シミュレーションでは、恒常的な周期波形を示さず周期が収束、もしくは発散する場合も存在する。むしろシミュレーション開始直後からも周期的な挙動を示さない解の組み合わせの方が数多く存在する。そのため、恒常的な周期挙動を示さない場合は評価値を与えないものとした。

探索範囲に対して、恒常的な周期挙動を示す解が存在する範囲は非常に狭いため、通常のGAを用いた方法で最適解を発見することは困難である。そこで、GAの探索前にランダム探索を行い、恒常的な周期変化を示す解候補を見つけ、その後、その解候補を起点としてGAを用いた最適化計算を行い周期の誤差や相対振幅が実際の挙動に近い結果となるようにパラメータの調節を行なう(表5-1)

表5-1. 概日リズム最適化問題における評価値の許容範囲と終了条件
(相対振幅・周期の誤差)

目的関数	終了条件	
相対振幅[-]	終了条件 1:	0.50 以上
	終了条件 2:	0.60 以上
周期の誤差	終了条件 1:	1[min]以内
	終了条件 2:	1[sec]以内



< Relative Amplitude >

$$Evaluation[1] = 1 - \frac{|Y - X|}{Y + X}$$

$$(0.0 \leq Evaluation[1] \leq 1.0)$$

< Error of time cycle >

$$Evaluation[2] = \frac{|24[hour] - Time\ cycle[hour]|}{24[hour]}$$

$$(0.0 \leq Evaluation[2] \leq 1.0)$$

図 5-3. 概日リズムにおける 2 つの目的関数 (相対振幅と周期誤差)

5. 1. 2. 概日リズムの最適化問題と熱ショック応答との比較

前章での熱ショック応答シミュレーションでは最適化が必要な動力学パラメータの数が 9 個であったのに対し、概日リズムシミュレーションで最適化が必要なパラメータ数は 47 個である。つまり、熱ショック応答の 9 次元の空間探索に対し、概日リズムでは 47 次元の空間探索となる。大腸菌における一過性のピーク挙動は、与えられた刺激に対する反応であるため、ピーク挙動を示す解は比較的多く見つかる。しかし、概日リズムは平常時の恒常的に安定した挙動を維持しなければならぬため、このような結果を示す解は極めて狭い範囲にのみ存在する。概日リズムの最適化問題は、探索範囲が広く、解が存在する範囲も狭いため、熱ショック応答と比べると調整が非常に難しい問題であるといえる。

また、概日リズムのシミュレーションでは、挙動が安定するまでシミュレーション計算を続けて波形の判定を行うため、シミュレーション 1 回あたりの平均計算時間は約 30 秒である。熱ショック応答の計算時間が約 0.1 秒であるため、概日リズムの反復的な最適化計算には非常に多くの時間が要求され、研究者達の負担と研究進度のボトルネックにもなっている。

5. 2. 概日リズムにおける最適化実験の結果

5,000回(100個体で50世代分)の個体の評価計算を行う内に, 両方の目的関数の値が, ともに終了条件を満たせばその試行は最適化成功とする. 線形結合 GA, 分散協力 GA, 生存率 GA を用いて最適化計算を行った. 各 GA の設定は(表 5-2)の通りである.

表 5-2. 各GAの設定値 (ショウジョウバエ概日リズム)

Optimization problem	
Title: Parameters tuning of circadian oscillation system in <i>Drosophila</i>	
Number of parameters to be optimized: 47	
Evaluation functions:	
Evaluation 1: Error of time cycle between 24 hours	
Evaluation 2: Relative Amplitude	
GA setting	
Crossover: UNDX	
Max Evaluation times: 6,000	
Number of trials per setting in each GA: 20	
Survival ratio GA	Number of initial genes: 50, 100, 200 Survival ratio: 0.0~0.9 (every 0.1 interval)
Linear combination GA	Number of genes: 50, 100, 200, 500 Weight coefficient: 0.0~1.0 (every 0.1 interval)
DCMOGA	Genes in a population: 50, 100, 200, 500 Number of populations: 3 Migration ratio: 0.1 Migration interval: Every 10 generations Weight coefficient: 0.0~1.0 (every 0.1 interval)

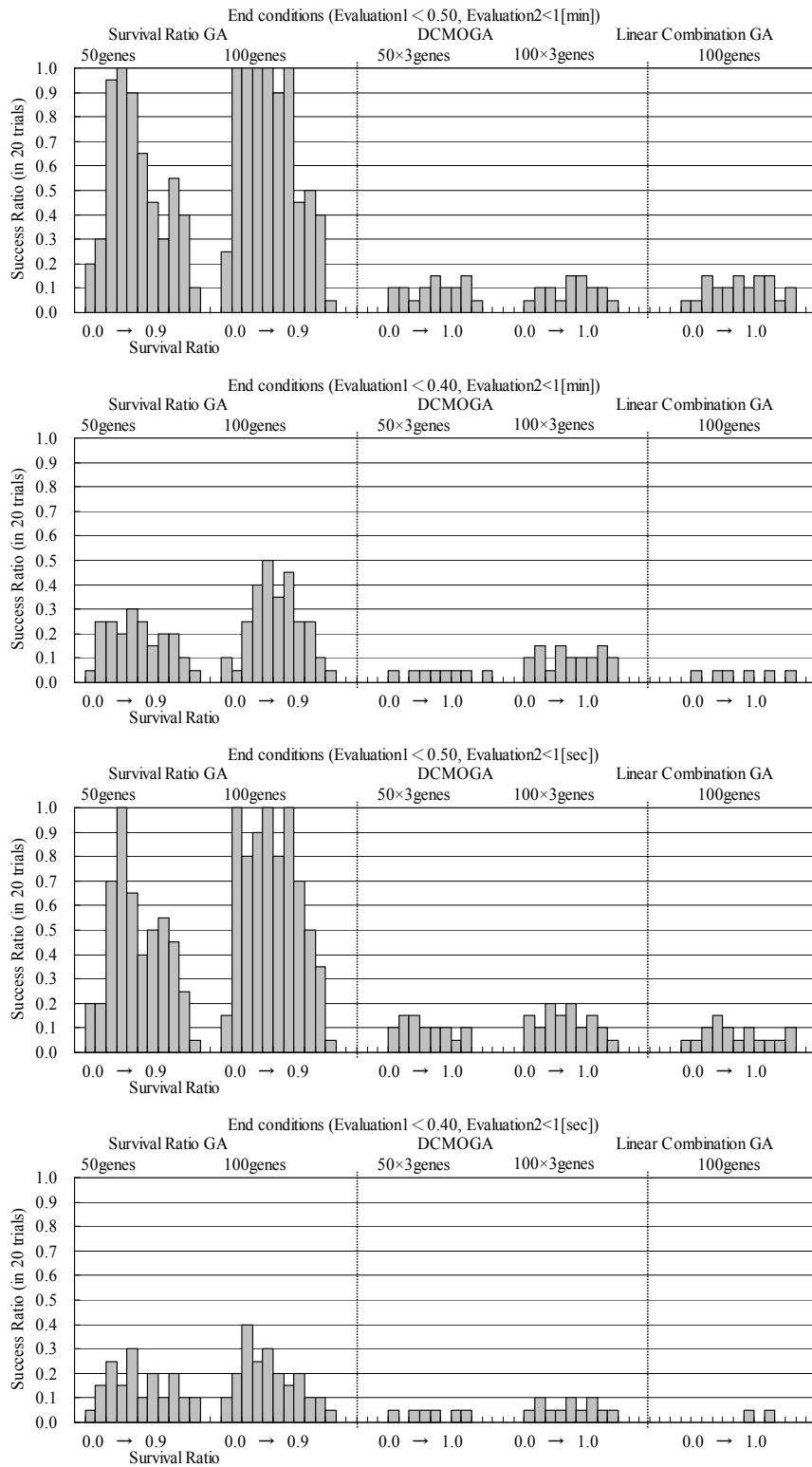


図 5-4. 各終了条件における最適化成功確率 (20 試行分の統計)

(図5-4)は、各終了条件での最適化成功確率を示している。線形結合 GA と分散協力 GA では、2つの目的関数のうち、片方は終了条件を満たす解を発見することは出来たが、双方を満たす解を見つけ出すことは困難であった。そのため、成功確率は低くなっている。一方、分散協力 GA では全個体数が多いため、世代の繰り返し数が少なくなってしまうが、成功確率は線形結合 GA と同程度であった。これは各個体群で独立した目的関数下で進化させたこと、これらの性質を混合する多様性の維持を行ったことによる効果である。生存率 GA は他の2つの手法に比べ高い成功確率を示した。

線形結合 GA と分散協力 GA では、最適化の終了条件が変わってもほぼ同程度の成功確率となっている。つまり、終了条件の厳しい解を見つけ出すことは出来るが、解のバリエーションは少なかった。一方、生存率 GA では、厳しい終了条件も緩い終了条件も満たす多様な解を、従来よりも高い最適化成功確率で見つけ出すことができています。生物の反応原理を追究しようとするシミュレーションの最適化問題においては、最適化計算速度を速く、成功確率を高くすることが求められるが、同時に条件を満たす複数の解候補を抽出することが要求されている。生存率 GA はこの両者の要求に応えている点から、本問題においては良好な最適化結果を示しているといえる。

5. 3. 概日リズムシミュレーションの広域探索

前節までの研究で、サーカディアン最適化計算は成功確率が向上し、解を得られるまでの時間も約 1/7 の 24 時間となった[17]. しかし、GA での最適化の前処理として、ランダム探索を行い、そこで見つかった僅かに振動波形を示す解からスタートして、GA での最適化を行っている. サルカディアン最適化問題においては、最適化が必要な変数の数も多く、探索範囲は非常に広大であるため、GA での探索の起点にもなる微振動性質をランダム探索で見つけるだけでも 10 日ほど必要である. GA での最適化にかかる時間を短縮できても、それ以上にランダム探索に時間がかかっているため、全体の計算時間はあまり短縮されていない(図 5-4). また、ランダム探索で見つかった 1 つの起点から GA の探索を始めるため、GA ではその起点の極近傍しか探索できず、大域的な最適解を見逃す可能性が高い. 特に生物系のシミュレーションにおいては、同様の挙動を示すが、各パラメータの異なる解が複数個存在すると言われている. 生物原理究明の研究において、最適化速度を向上させることも重要であるが、多彩な種類の最適解を発見することも重要な目的である.

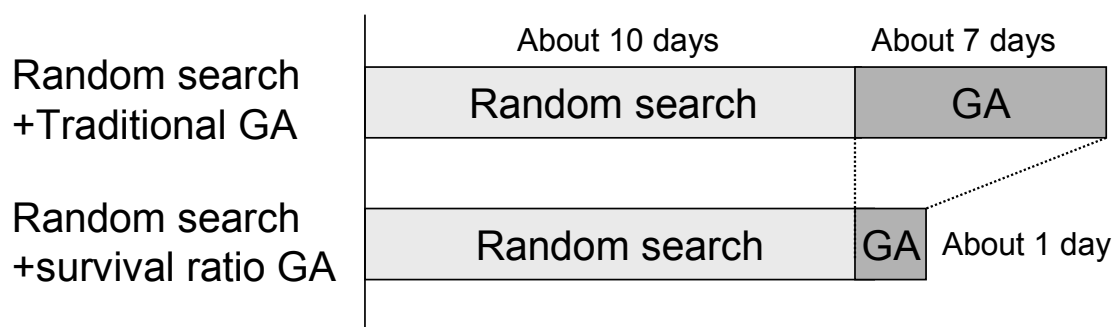


図5-5. 概日リズムの最適化に要する平均計算時間

(Pentium(R) 4 CPU 240GHz, 2. 39GHz, 512 MB RAM)

5. 3. 1. 1 段階の生存率 GA を用いた概日リズムシミュレーションの広域探索

これまではランダム探索と GA の 2 段階で行われていた、サーカディアン最適化計算を、GA だけの 1 段階の生存率 GA で行うことを試みる。生存率 GA では、評価に用いる目的関数を 1 世代おきに入れ替えるため、個体群が 2 つの探索軸を往来しながら空間探索を行うため、従来手法よりも広い範囲を調べる事ができる。また、過去の探索履歴を保持し、それらを参照した探索を行うことで個体性質の多様性も十分に確保することができるメリットに着目すると、1 段階の生存率 GA でも概日リズムシミュレーションの最適化が行えるのではないかと期待される。

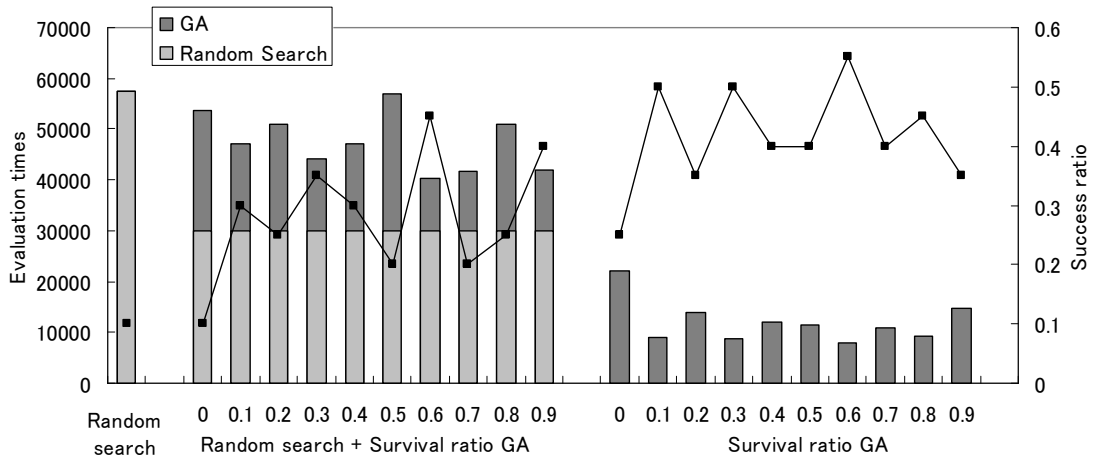
5. 3. 2. 概日リズムの広域探索における2段階最適化と1段階生存率GAでの結果

2つの評価基準を持つ概日リズムの広域探索において、2段階探索(ランダム探索+生存率GA)と1段階探索(生存率GAのみ)を用いて最適化実験を行った。2つの目的関数は前章に等しい。各手法の設定条件は(表5-3)のとおりである。最大計算回数は120000回とした。ランダム探索では、平均して約60000点の探索で、全探索範囲中から恒常的な周期挙動を示す1個の解候補を見つけ出すことができる。2段階探索の後半の生存率GAでは、この解候補を探索の起点として、初期個体を生成し、さらなる波形の調整を行っていく。1段階探索(生存率GAのみ)では、全探索範囲中のランダムな位置に初期個体を生成して探索を進める。

表5-3. 2段階探索(ランダム探索+生存率GA)と1段階探索(生存率GAのみ)の設定条件

	Setting
Random search	Max search points: 120000
Random search + Survival ratio GA	Random search Max search points: 60000 Survival ratio GA Genes: 100 Survival ratio: 0.0~0.9 Crossover: UNDX [7]
Survival ratio GA	Genes: 100 Survival ratio: 0.0~0.9 Crossover: UNDX

Termination condition 1: Relative Amplitude <0.45 , Error of time cycle <10 [min]



Termination condition 2: Relative Amplitude <0.45 , Error of time cycle <1 [min]

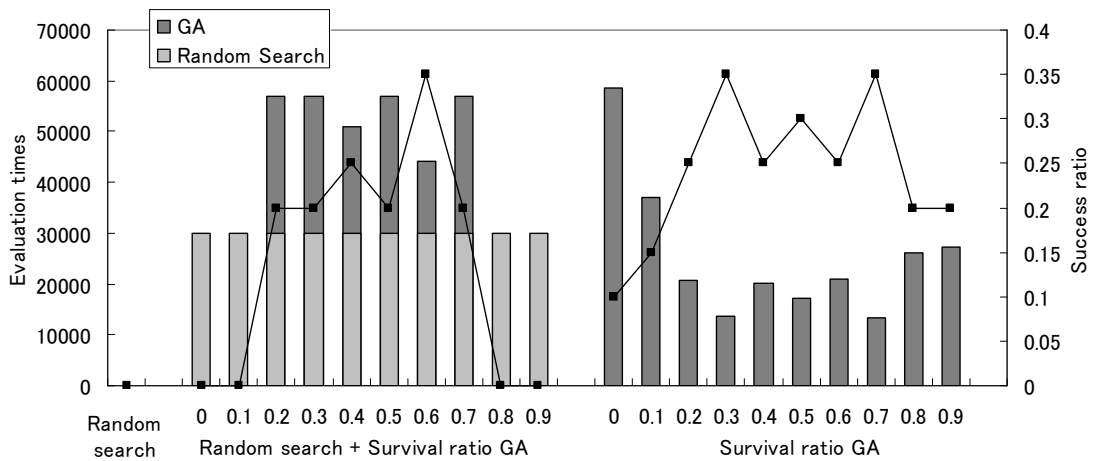
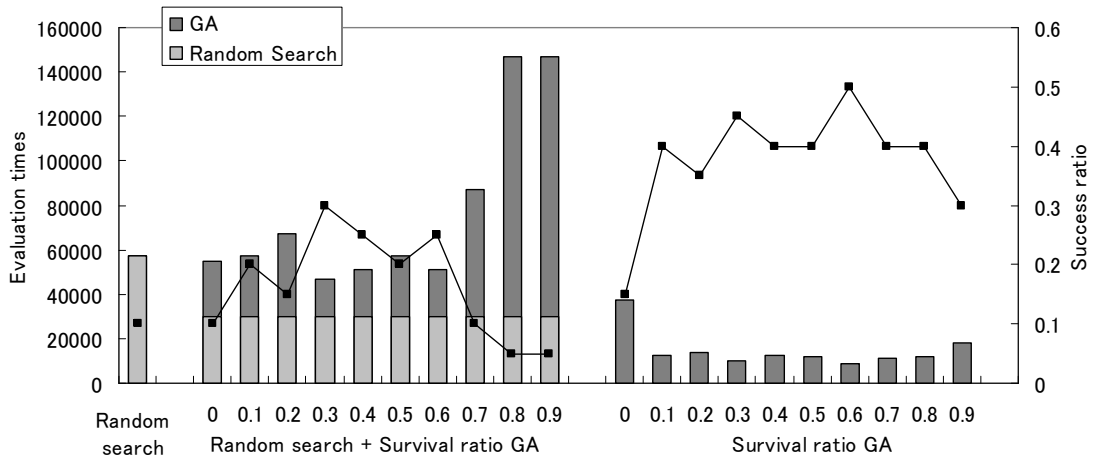


図5-6-A. 各終了条件における平均計算回数と最適化成功確率の関係

Termination condition 3: Relative Amplitude <0.40 , Error of time cycle <10 [min]



Termination condition 4: Relative Amplitude <0.40 , Error of time cycle <1 [min]

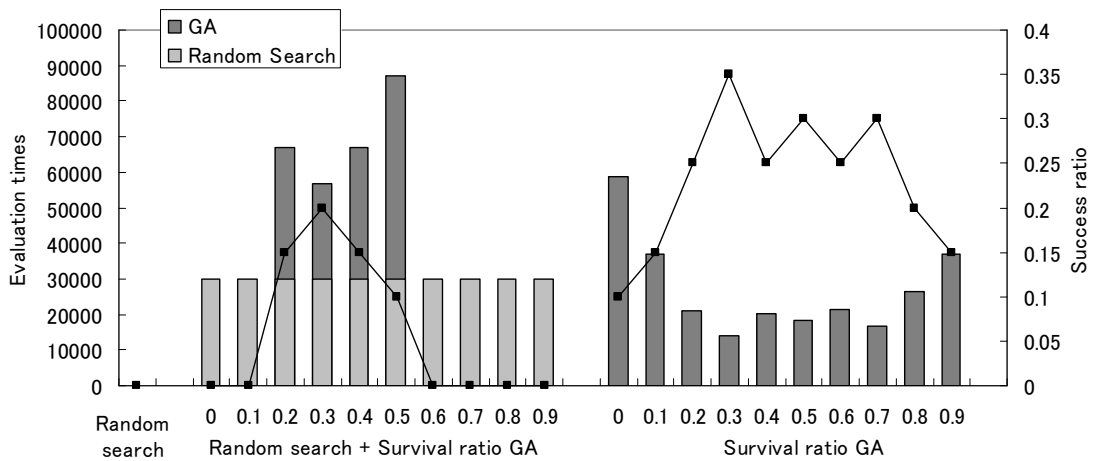


図5-6-B. 各終了条件における平均計算回数と最適化成功確率の関係

図 5-6 は各手法を用いた場合の最適化成功確率と平均計算回数のグラフである。さまざまな終了条件において、2つの目的関数が共に終了条件を満たすことができたかを比較している。

1段階探索(生存率 GA)は2段階探索(ランダム探索+生存率 GA)高い成功確率となり、平均計算回数も少ない。生存率が 0.0~0.2 では、個体群内に十分な過去探索履歴を蓄積することができないため、成功確率は2段階探索よりも低いが、生存率が 0.4~0.7 周辺では、成功確率は従来と同程度か高いくらいで、平均計算回数は少なくなった。生存率が 0.8 以上では、個体群内に過去の探索履歴を過剰に蓄積するため、優秀な性質を次世代の子個体が継承できる可能性が低くなる。そのため、GA の探索能力は低下し、成功確率も悪くなる。しかし、生存率 GA による1段階探索方は終了条件が厳しくなっても従来よりも高い成功確率での最適化が実現されている。

5. 3. 3. 最適化計算によって発見された局所解の個数について

図 5-7 は、従来の 2 段階探索と生存率 GA による 1 段階探索を用いて最適化計算を行った過程で発見された終了条件(Relative Amplitude <0.45 , Error of time cycle <10 [min])を共に満たす解を記録し、その類似性からグループ分類したときのグループ数を示している。つまり、このグループ数が最適化計算の過程で発見する事ができた局所最適解の個数であるといえる。グループの分類では、群平均法によりクラスタ分析を使用し、グループの判定基準となる最小類似度毎の結果で示している。ランダム探索だけでは 1 個の解が発見されている。ランダム探索の後に生存率 GA を行った 2 段階探索ではつまり、ランダム探索で見つかった 1 個に解候補から集中的な再探索をすることで 1~4 個の解が発見されている。しかし、2 段階探索では 1 段目のランダム探索で発見された解を起点として個体の分布を行うが、起点となった性質が個体群の性質に強い影響をあたえ、その地点に収束しやすい状態となってしまうため、結果として、ランダム探索で見つかった解候補の近傍以外は探索することが困難となるため、見つかった局所解の個数も 5 個程度となっている。

これに対して、生存率 GA のみの 1 段階探索では、GA 個体の初期分を制限する要素もなく、全探索空間中に満遍なく個体が配置され、2 つの評価基準が毎世代ごとに入れ替わる事で個体群の性質は大域的な振幅移動を繰り返しながら探索を進める。そのため、個体群は 2 つの探索軸を往来することで、2 段階探索よりも広い範囲を探索し、より多くの局所解グループを発見することができた。

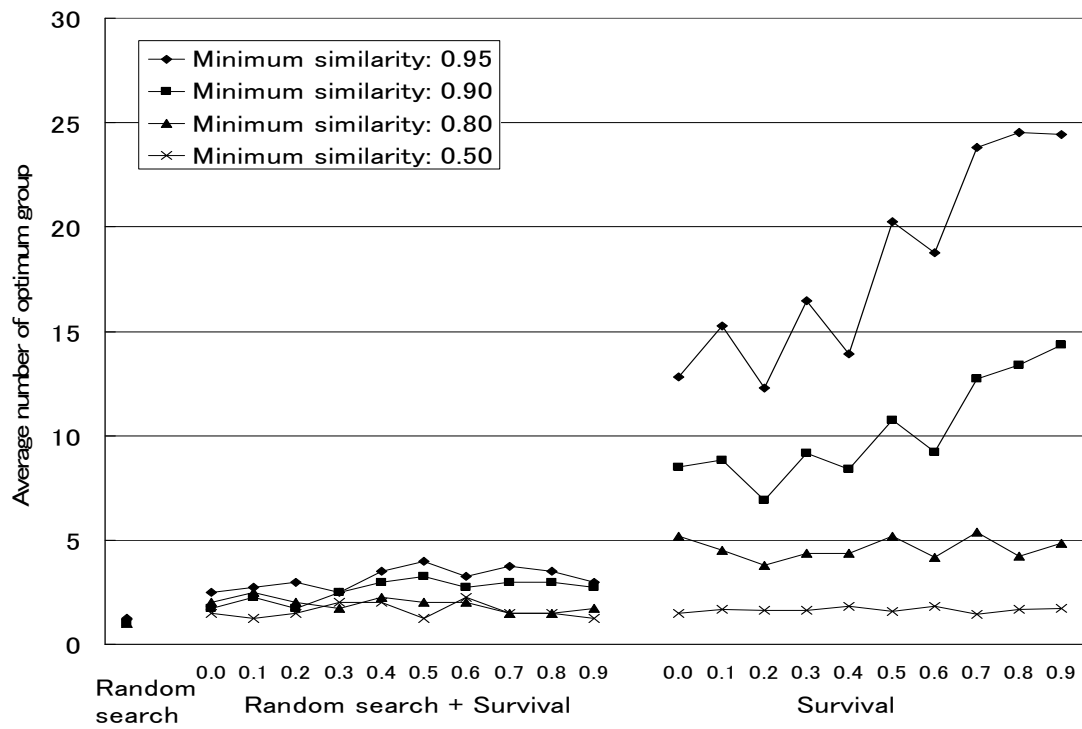
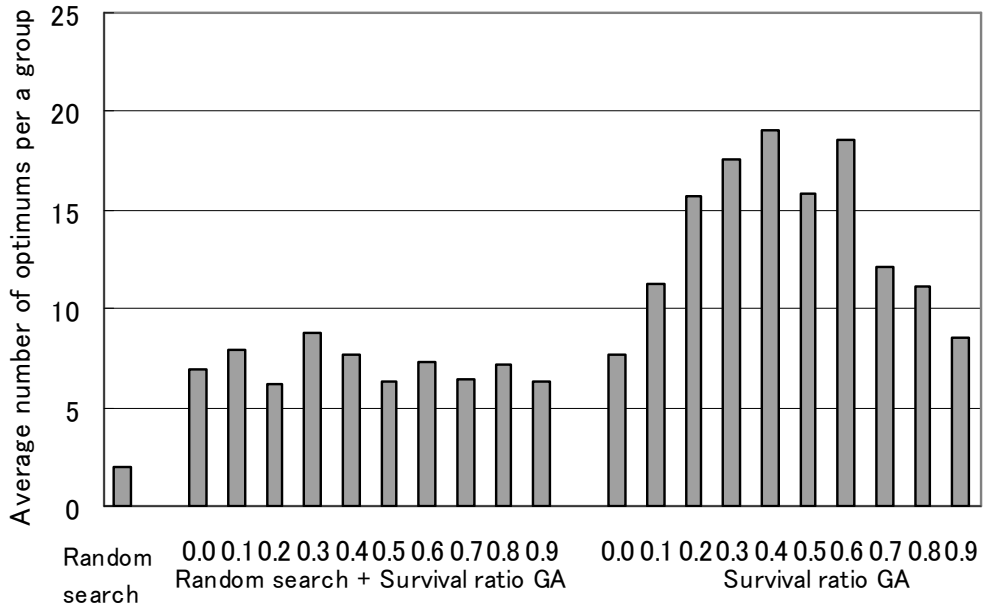


図5-7. 各条件のGAにおいて、最適化計算によって発見された解グループの個数
 (発見された全ての解を群平均法でクラスタリング、最小類似度ごとの結果)

Minimum similarity: 0.95



Minimum similarity: 0.90

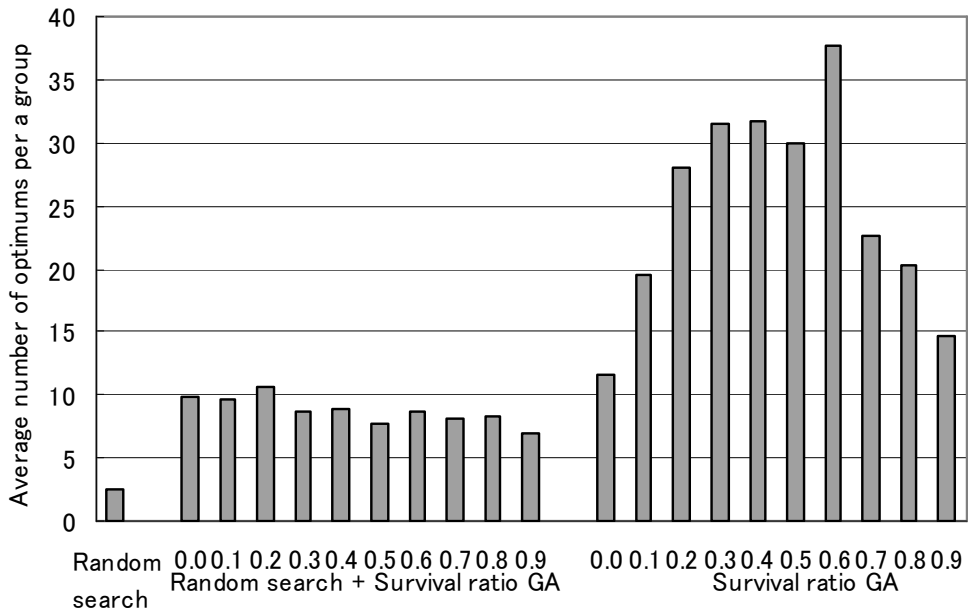
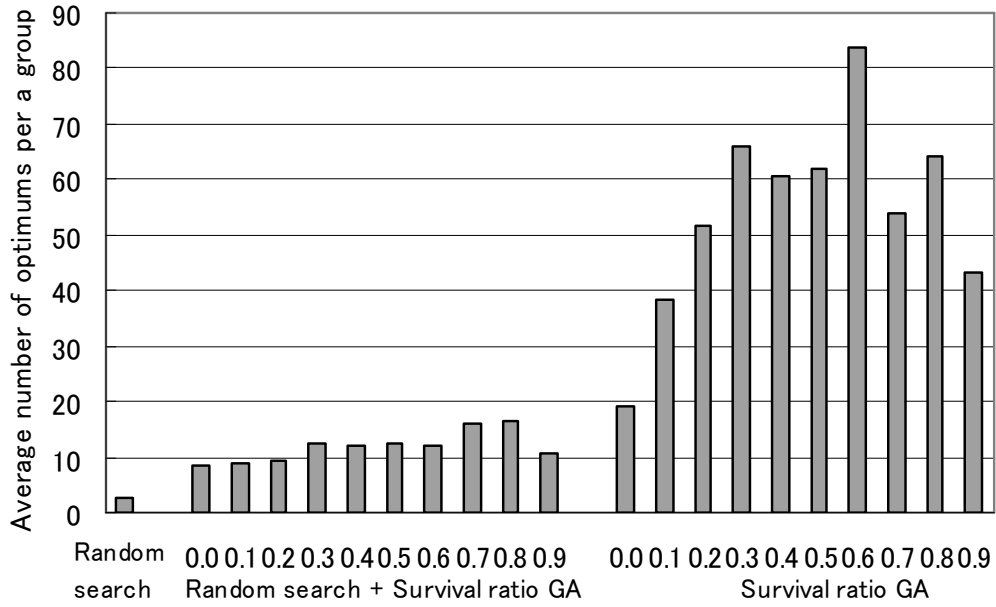


図5-8-A. 発見された各解グループに含まれる個体の平均数

Minimum similarity: 0.80



Minimum similarity: 0.50

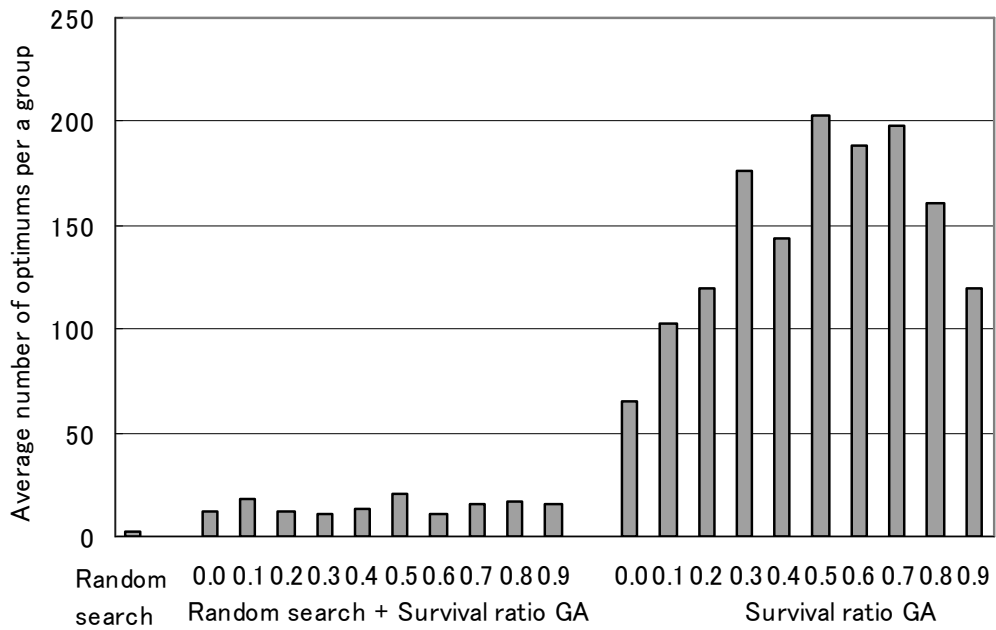


図5-8-B. 発見された各解グループに含まれる個体の平均数

また、図 5-8 は、クラスタ分析されたそれぞれの解グループの中に平均何個の解を発見できたかを示している。ランダム探索後に生存率 GA を用いた 2 段階探索手法では、1 つのグループに 10 個程度の解が存在している。一方、生存率 GA のみの 1 段階探索では、2 段階探索よりも多くの個数の解がグループ内に存在している。つまり、生存率 GA のみの 1 段階探索は、従来の手法よりも各解グループに対してもより集中的な探索がおこなえている。

2 段階探索では、探索空間を絞り込むためにランダム探索を行うのだが、見つかった解候補を起点として、GA での最適化を行うため、GA の個体の分布が起点となった個体性質に強く依存するため、結果として、その起点の近傍しか探索できずに個体群内の個体性質も起点となった性質に収束しやすく多様性を失いやすいため、各解グループ内に見つかる解の種類も少なくなる。

生存率 GA の 1 段階探索では、個体群が過去の探索履歴を保持することと、評価関数を毎世代入れ替えることで、子個体は多様な性質を継承することができ、従来手法よりもより多くの種類の解を発見することができた。

5.3.4.1 段階探索と2段階探索における探索経路

図 5-9 は、2 段階探索(ランダム探索)と1 段階探索(生存率 GA のみ)で広域探索を行ったときの探索経路を世代の順にプロットしたグラフである。縦軸と横軸にそれぞれ各目的関数の値をとっている。2 段階探索において、ランダム探索で見つかった解候補は○で示している。2 段階探索ではランダム探索で発見された解を起点として GA の探索を行うため、その起点から到達可能な狭い範囲しか探索されていない。一方、1 段階生存率 GA では、それぞれ片方の目的関数を満たす解を探索する事から始まり、1 世代おきに評価に用いる目的関数の入れ替えを行うことで、個体群は広い範囲を反復移動しながら探索を進めている。また、探索の過程で発見されたそれぞれの目的関数に対して優良な性質を個体群内に保持することで、子個体はそれらを組み合わせた性質を継承することができ、2 つの目的関数を満たす解を発見することができる。

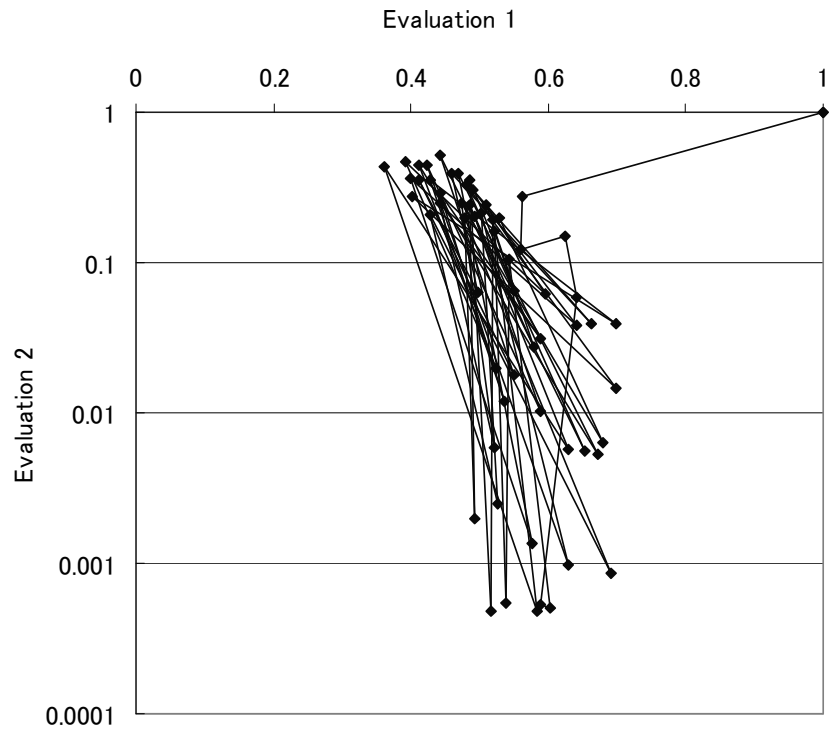
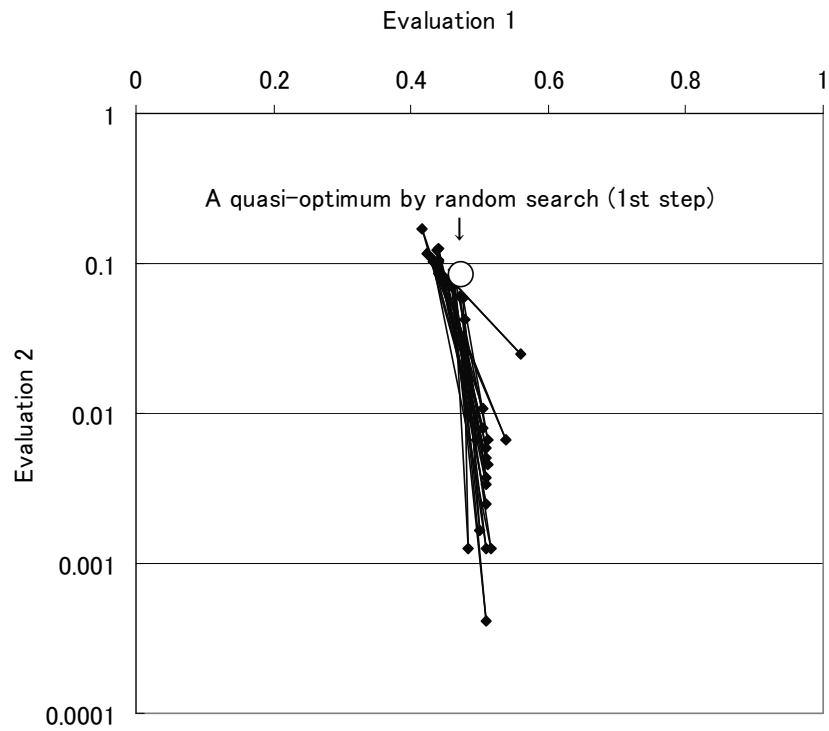


図5-9. 個体群中の最優良個体の目的関数値の世代推移.

ランダム探索 + 生存率GAの2段階探索(上図)

生存率GAのみの1段階探索(下図).

5. 4. 生存率 GA のみの 1 段階探索による概日リズムの最適化効率の向上

本章では、概日リズムの最適化計算をより効率的に行うために生存率 GA を用いた手法を用いた。従来の最適化は、広大な探索範囲を絞り込むためにランダム探索を行い、発見された解候補を起点として GA による再探索を行っていた。まず、この 2 段階探索において、GA の部分を従来手法から生存率 GA に置き換えて最適化計算を行うことで、平均計算回数の減少と成功確率の向上が実現できた。また、生存率 GA が各目的関数の性質をそのまま評価に取り込む点や個体群が多様な性質を保持する事で従来手法よりも広い範囲を探索できる点に着目して、生存率 GA のみの 1 段階探索を行った。従来手法の GA を用いた 1 段階探索では、十分な最適化成功確率が得られなかったため、慣例的にランダム探索と組み合わせた 2 段階探索が行われていたのだが、生存率のみの 1 段階探索手法は従来手法よりも少ない計算回数と高い成功確率で最適化を行うことができた。実際の最適化計算時間を比較しても、従来の約 1/3 に短縮され、研究の現場における最適化計算のボトルネックの軽減につながる(図 5-10)。また、生存率 GA のみの 1 段階探索手法では、従来よりも多くの解の種類を発見することができ、研究者に有用なデータをフィードバックすることができる。生存率 GA は周期的な挙動を探索する概日リズムの最適化問題においても非常に良好な結果を示した。

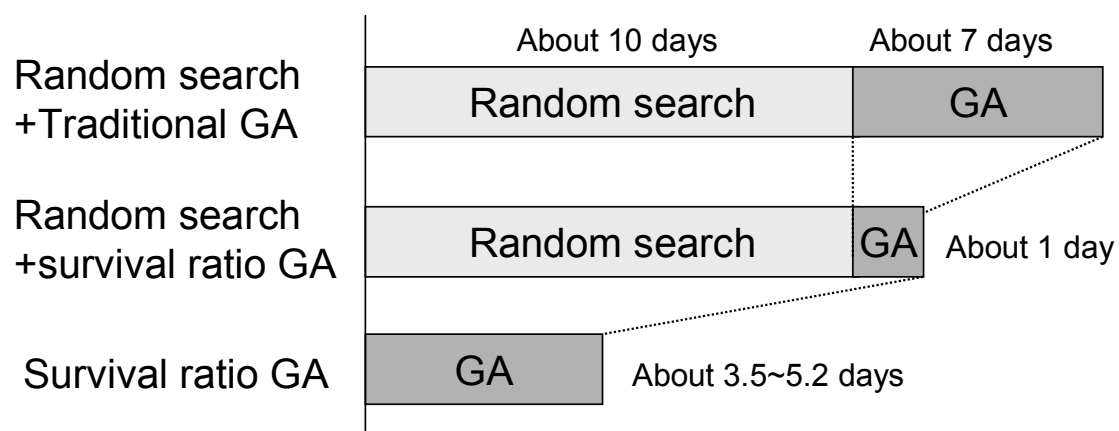


図5-10.生存率GAのみの1段階探索で短縮された概日リズムの最適化にかかる平均計算時間 (Pentium(R) 4 CPU 240GHz, 2. 39GHz, 512 MB RAM)

6. 大腸菌の窒素同化システムでの最適化計算

これまでの検証で、生存率 GA は一過性の挙動を示す反応系、周期的な挙動を示す反応系の最適化計算で良好な結果を示した。本章では、生存率 GA の高い最適化性能と 2 つの目的関数をオリジナルな性質のまま使用できることを利用して、モジュール分割された大腸菌の窒素同化システムの最適化計算に導入する。

6. 1. 大腸菌の窒素同化システムとモジュール分割

窒素同化とは、土壌中に含まれるアンモニウムイオンや硝酸イオンから必須アミノ酸であるグルタミンを生成する反応である。グルタミンの生成量は環境中のアンモニア濃度に依存してしまう。そこで、生物は細胞内の代謝物 α -ケトグルタル酸とグルタミン酸のバランスを一定に保つことで、環境の変化に対する直接的な影響を緩和している。大腸菌はアンモニアを細胞中に同化する酵素 GD, GDH, GOGAT を持つ。この中で中心的な役割を果たすのは GS(glutamine synthetase) である。窒素同化システムは、 α -ケトグルタル酸とグルタミン酸の濃度比にしたがって、GS の活性と発現量を調節する機構を持っている。この機構は GS 活性フィードバック制御と GS 合成フィードバック制御の 2 つに分けることができる(図 6-1)[18]。

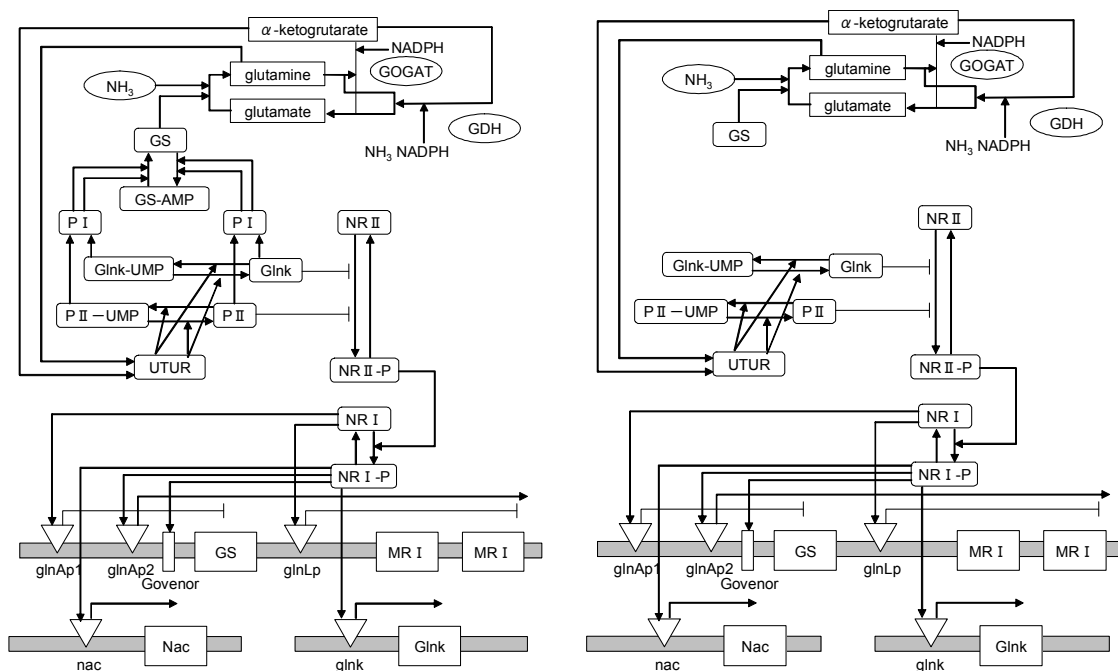
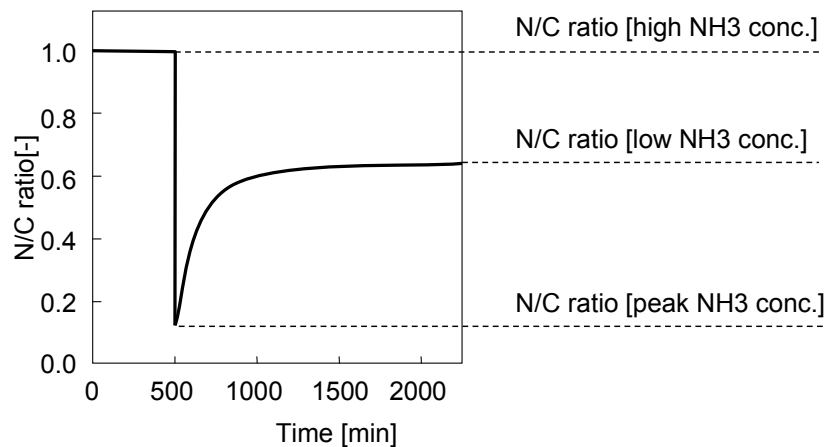


図 6-1. モジュール分割の図 (左:フルモデル, 右:合成モデル)



$$Evaluation = \frac{N/C \text{ ratio}[low NH_3 \text{ conc.}] - N/C \text{ ratio}[peak NH_3 \text{ conc.}]}{N/C \text{ ratio}[high NH_3 \text{ conc.}] - N/C \text{ ratio}[peak NH_3 \text{ conc.}]}$$

図 6-2. 遺伝的アルゴリズムの評価値の決定

窒素同化システムでは α -ケトグルタル酸とグルタミン酸のバランスを一定に保つ制御が行われている。 α -ケトグルタル酸は炭素源であり、グルタミン酸は窒素源である。この炭素源と窒素源の比のことを「N/C 比」と呼ぶ。窒素同化システムでは系に外乱が加えられたときでも、「N/C 比」を一定に維持する働きがみられる。図 6-2 は、シミュレーション時間 500min においてアンモニア濃度を $1.000000e+005[G]$ から $1/5$ まで低下させたときの N/C をシミュレーションした結果である。各設定のシミュレーションにおいて、外乱により減少した N/C がどこまで回復するかを評価基準とした。

6. 2. モジュール分割を利用した窒素同化システムの最適化

前章までの熱ショック、概日リズムの最適化問題では1つの系に対して、2つの目的関数を計算し、これら2つの目的関数を共に満たす性質の探索を行った。これに対し、窒素同化の最適化問題では2つの系を用いてシミュレーションを行なう。これまでは、フルモデルを始めから最適化することは困難であったため、まず合成モデルの最適化を行った後に、人手もしくはGAを用いてフルモデルが正しい挙動を示すように調整が行われていた。本論文では、フルモデルと合成モデルの最適化計算を同時に行い、合成モデルの探索結果を随時フルモデルの探索にフィードバックすることで、フルモデルのより効率的な最適化計算を期待する。

フルモデルでの未知変数は17個、合成モデルでは13個である。合成モデルの13個の変数はフルモデルと共通である。1世代おきにフルモデルと合成モデルでの個体評価法を入れ替え、生存率GAを用いて最適化を行った。個体群内には、フルモデルと合成モデルで評価された個体が蓄積され、新たに生成される子個体はそれら両方の性質を継承することになる。

前節までの対象問題では、1個体のシミュレーションを行い、その結果から2種類の目的関数を計算し2つの評価値を得ていた。しかし、本問題で2つの個体評価値を計算するためには、フルモデルのシミュレーションで1回、合成モデルのシミュレーションで1回、合計2回の計算が必要となる(図6-3)。重み関数を用いた手法では、2つの評価値を計算しなければならないため、個体の評価値計算に2倍の計算負荷が必要となる。シミュレーションを1回実行する時間が約60秒、1世代(100個体)の計算に約100分かかる。GAの1世代あたりの世代操作(ソート、選択、交叉、突然変異)に必要な時間が約0.1秒であるため、計算時間の大半は個体の評価計算である。事前の予備実験において、重み関数を用いる手法(線形結合GA、分散協力GAなど)では、最適化成功確率が低く、この計算回数の倍増も原因で良好な最適化結果が得られなかった。そこで、本問題では生存率GAを用いた手法のみを使用して最適化実験を行い、この手法を用いた場合の最適化結果についての検討を行う。

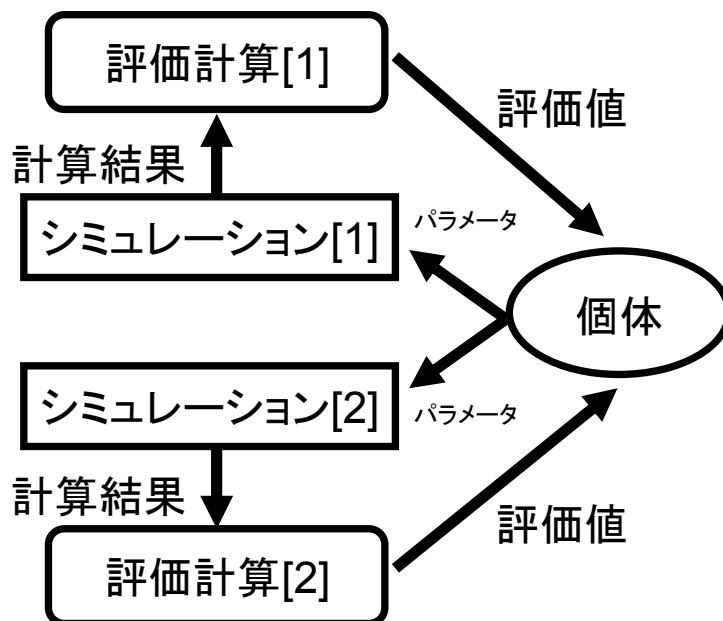
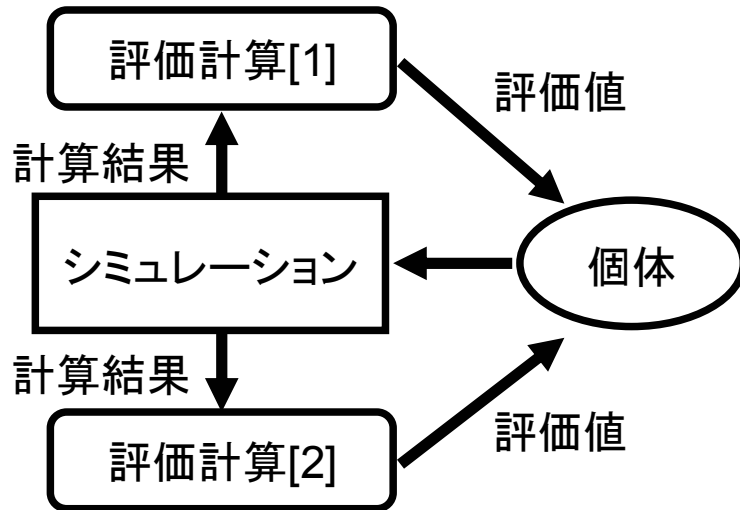


図 6-3. 1 個体の評価値を計算するまでの計算負荷

上図:前章までの最適化問題

下図:モジュール分割での最適化問題

6. 3. 生存率 GA を用いた窒素同化のモジュール分割法での最適化結果と性能の向上

図 6-4 は、生存率 GA を用いて、生存率を 0.0～0.9 まで推移させて窒素同化システムの最適化計算を行ったときの平均計算回数と最適化成功確率を示している。各試行の設定条件は表 6-1 の通りである。最適化計算の終了条件フルモデルの評価値が表 6-2 の値以上となることであり、グラフは各終了条件における結果を示している。各試行における最大計算回数は 2500 回とした。

2 つのモジュール分割に分割された窒素同化システムの最適化問題を生存率 GA を用いて最適化計算することで、フルモデルのみの最適化よりも成功確率が 2 倍以上に向上した。また、計算回数も少なくなり良好な最適化結果を示した。特に平均計算回数と終了条件の関係について、フルモデルでの最適化手法では、終了条件が厳しくなるほど多くの計算回数が必要であったが、モジュール分割の生存率 GA を用いた手法では、終了条件に対する計算回数の差が少なくなっている。つまり、個体に生存率という概念を与えて、個体群に過去探索個体を蓄積させ、子個体がそれらを参照した探索を繰り返す事で、終了条件が厳しい解をより効率的に見つけ出す事ができている。

表 6-1. 生存率 GA の条件設定

生存率 GA
個体数:100 個体
生存率:0.0～0.9 (0.1 刻み)
交叉:UNDX
試行回数:各 20 試行

表 6-2. 大腸菌窒素同化システムでの終了条件

フルモデルにおける評価値

終了条件 1: Evaluation \geq 0.95

終了条件 2: Evaluation \geq 0.90

終了条件 3: Evaluation \geq 0.85

終了条件 4: Evaluation \geq 0.80

終了条件 5: Evaluation \geq 0.75

終了条件 6: Evaluation \geq 0.70

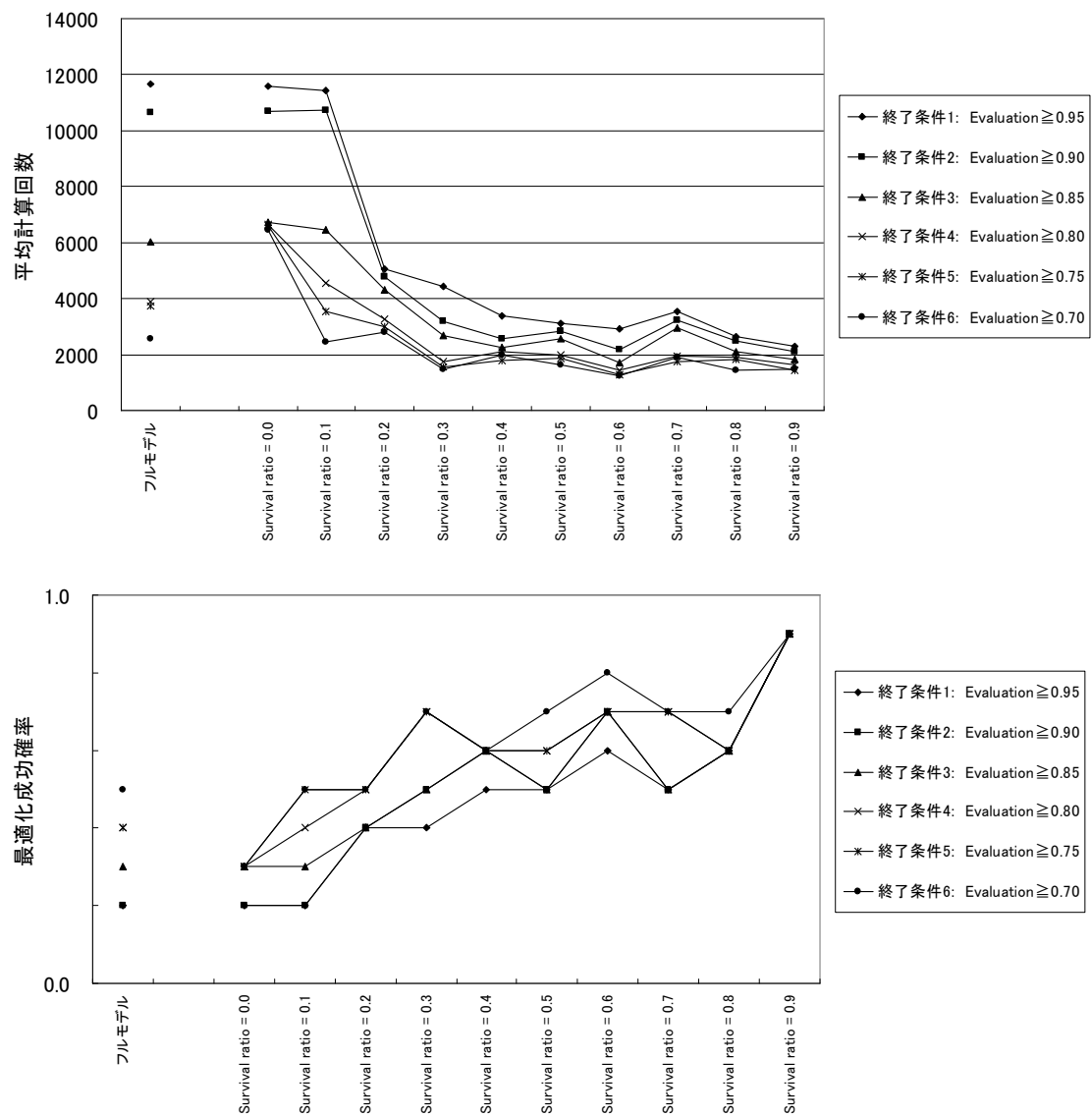


図 6-4. 各終了条件における成功確率と計算回数

7. 結論

本論文では生命分子ネットワークのダイナミックモデルのシミュレーションの最適化を対象問題として、シミュレーションの特徴を2つの評価基準(目的関数)で定義し、2つの目的関数を持つ最適化問題をより効果的に解くため手法についての提案・検討を行った。これまでは生物反応を計算機上に再現するために、1つの目的関数に対してGAを用いた最適化計算を行ってきた。しかし、未知変数の個数や探索範囲が広くなると、最適解を見つけ出すことは困難となってきた。そこで、そこで、2つの目的関数を定義し、これまで1つだった探索軸を2つにすることで最適化にたどり着ける可能性を向上できるのではないかと期待した。しかし、2つの目的関数を持つ最適化問題の最適化においては、目的関数間・変数間に強い相互依存関係があり、双方の評価基準を満たす性質を見つけ出さなければならないため、一般的な最適化計算に用いられる解析的な手法や従来型のGAでは十分な最適化成功確率を得ることは困難であった。そこで、GAの各個体に生存率という概念を導入した生存率GAという手法を提案し、数種類の最適化問題に適用することによって、その最適化性能を検証した。

生存率GAでは、個体群中の各個体が、次世代に生き残るかの判定を生存率により決定される。これにより、個体群の中では、これまでに生成された個体が数世代にわたり生き残り、年代層を形成する。この個体蓄積は、過去の探索履歴を保持するデータベースの役割を果たし、新しく生成される子個体は、個体群に生き残っている個体から親個体を選び、性質を継承するため、過去の探索履歴を参照した探索が可能となる。また、個体群内に過剰な個体が蓄積することは、子個体が優秀な親を選択できる可能性の低下につながるが、生存率により各年代の個体は次第に淘汰されていくため、個体群内の個体数は安定数を保つようになる。この生存率GAは、単一目的関数の最適化にも用いることができ、その効果が期待されるが、本論文では特に2つの目的関数を最適化問題をより効率的に解く手法として検証を進めた。

生存率 GA を 2 つの目的関数を持つ最適化問題に用いる際には、個体の性質とする目的関数を 1 世代おきに交互に入れ替える。これにより、個体群内には 2 種類の目的関数で評価を受けた個体が交互に蓄積されていく。新たに生成される子個体は、これら双方の性質を継承することができるため、多様な性質を得ることが出来る。また、子個体も順次、個体群内に蓄積されることでの次世代での探索のヒントとなる。

本論文では、一過性のピーク挙動を示す大腸菌の熱ショック応答と恒常的な周期挙動を示すショウジョウバエの概日リズム、反応系を二系統で表現できる大腸菌の窒素同化システムを対象問題として、2 つの評価基準を持つダイナミックモデルの最適化実験を行った。大腸菌熱ショック応答の最適化問題では、生存率 GA は従来手法よりも高い最適化成功確率を示し、平均計算回数も少なくなった。また、個体を評価する目的関数を 1 世代おきに入れ替えることで、個体群は 22 つの探索軸を往来しながら探索を行い、子個体は個体群内に蓄積された双方の多様な性質を継承することができるため、生存率 GA は従来手法よりも、広い範囲を探索する事ができ、より多い数の解候補も発見することができた。

ショウジョウバエの概日リズムの最適化問題においては、生存率 GA は高い最適化成功確率を示した。また、生存率 GA が多様な個体性質を保持し、従来よりも広範囲を探索することが出来る性質に着目し、これまでの 2 段階探索(ランダム探索+GA)を生存率 GA のみの 1 段階探索で行うことを試みた。一般的な生物シミュレーションの最適化においては、目標とする挙動を示す解は一意に決まるものではなく、複数個存在すると言われている。特に 2 つの目的関数を持つ最適化問題においては、双方の目的関数を満たす解候補は数種類存在する。生物反応の真理を研究するときは、できるたで多くの解候補を提示出来る最適化手法ほど、研究の助けとなる。これまでに用いられてきた 2 段階探索では、ランダム探索で発見された解候補を GA での探索起点として再探索を行っていたが、結果的にランダム探索で見つかった解候補の近傍しか探索出来ないため、十分な種類の解を得られないことが問題であった。しかし、生存率 GA のみの 1 段階探索では、探索初期から広範囲を探索し、個体群は 2 つの探索軸を往来しながら多様な個体性質を獲得することができるため、従来手法

よりも多くの解候補を見つけ出すことができた。しかも、最適化成功確率も従来手法よりも高い結果を示したため、研究のボトルネックとなっている最適化計算時間の長さ、多様な解候補が得られにくいという2つの問題を軽減することができた。

大腸菌の窒素同化システムの最適化問題においては、ある反応経路の最適化を支援するために、モジュール分割された部分的な反応系の最適化を同時に行うことで、従来手法よりも良好な最適化結果を得ることができた。

生存率 GA の利用に際し、ユーザが設定すべき値は個体数と生存率の2つである。生存率 GA における、1世代当たりの計算負荷は新たに生成された子個体の評価値計算のみであり、生存率に依存せず一定であり、従来の simple GA に等しい。また、ユーザが設定すべきパラメータ数が少なく、GA の最適化性能に与える影響が小さいので、設定依存性の問題を軽減している。特に、最適化性能が大きな影響を受ける重み関数を用いず、個体の優劣は評価値の値ではなく優劣の順位を基準としているため、重み係数の複雑な設定や、各目的関数のオーダ変数強度の差異などを考慮する必要もない。

最適化実験での検証結果から、生存率が高すぎるときは、過剰な過去探索結果が蓄積され、優秀な個体が親として選択される確率が低下してしまうため、成功確率が低下する傾向がみられた。逆に、生存率が低すぎるときは、過去の探索履歴の蓄積が少なくなるため、その効果は低下する生存率が中程度であれば、平均して高い成功確率を安定して得られる傾向がみられた。

本論文では、これまでの最適化計算に用いられてきた重み付き線形結合 GA や分散協力 GA との比較を行った。本対象問題における2つの目的関数の最適化実験の結果では、生存率 GA を用いた手法は、他手法より高い成功確率と計算回数の軽減を示し、非常に良好な最適化性能を示した。ただし、分散協力 GA などには2つ以上の目的関数を持つ多目的関数を対象とした手法であり、今回のような2つの目的関数に限定した実験での性能比較では、どちらが優れているかの判断をすることは適切でない。また、生存率 GA も2つ以上の目的関数を持つ最適化問題に適用可能だが、子個体が継承する各目的関数の性質の重みが軽くなるため、包括的な探索が

困難になることが推察される. 3 つ以上の目的関数を対象とする問題においては, 今後のさらなる検証が必要となる. しかし, 一般的な最適化問題においては, 複数の目的関数を定義することは, あまり行われたい. 理由は, 各目的関数の重みづけが低くなり, それぞれの性質を探索に活かしにくくなるからである. また, 評価基準が増えれば増えるほど局所解の個数も増えて, 最適解の発見が困難になる場合もある. そのため, 2 つの目的関数を持つ最適化問題において, 生存率 GA の使用は, その設定も比較的容易であるため, 最適化をより効率的に行う手段の 1 つとして利用していただきたい.

また, 生命分子ネットワークのダイナミックシミュレーションにおける他反応系の最適化問題では, 基本的なネットワーク特性は生命システム間で共通することも多い. また, 本論文では代表的な生物反応挙動として, 一過性の挙動, 恒常的な周期挙動, モジュール分割についての検証を行っているため, 他の生物反応を新たに取り扱う際には, 本論文の最適化実験の結果を参照することで, 効果的な最適化計算が行えることが期待される.

8. 謝辞

本研究を進めるにあたり，終始多大なる御指導御鞭撻を賜った九州工業大学情報工学部の大橋健助教授に深く感謝の意を表します．また，遺伝子発現ネットワーク関連でお世話になった九州工業大学情報工学部の倉田博之助教授にも深く感謝いたします．研究の初期の頃から御指導を賜った九州大学農学研究院生物機能科学部門生物機能制御学講座の岡本正宏教授にも心より感謝致します．最後に，公私にわたりお世話になりました九州工業大学大橋研究室の皆様にもお礼申し上げます．

9. 参考文献

- [1] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, (1989).
- [2] H. Kurata*, H. El Samad*, R. Iwasaki, H. Othake, J. C. Doyle, I. Grigorova, C. Gross, and M. Khammash, “Module-Based Analysis of Robustness Tradeoffs in the Heat Shock Response System”, PLoS Computational Biology, 2: e59, (2006).
- [3] H. El-Samad*, H. Kurata*, J. C. Doyle, C. A. Gross, M. Khammash, “Surviving heat shock: Control strategy for robustness and performance”, Proc Natl Acad Sci U S A. 102:2736-41, (2005).
- [4] H. Kurata, K. Masaki, Y. Sumida, R. Iwasaki, “CADLIVE Dynamic Simulator: Direct Link of Biochemical Networks to Dynamic Models, ” Genome Res. , 15, pp. 590-600, (2005).
- [5] Hiroyuki Kurata, Nana Matoba, Natsumi Shimizu, “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, (2003).
- [6] Michalewics, Z. , “An Experimental Comparison of Binary and Floating Point. ” Representations in Genetic Algorithms. 2 pp. 187-202, (1993).
- [7] Ono I. and Kobayashi S. , “A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover, ” Proc. 7th Int’ L Conf.

On Genetic Algorithms pp. 246–253, (1997).

[8] Shin Tanaka, Hiroyuki Kurata, Takeshi Ohashi, “Optimization of E. Coli Heat Shock Response parameter tuning using Distributed and Integrated Genetic Algorithms, ” IEEE International Conference on Systems, Man and Cybernetics, pp. 1243–1248, (2004).

[9] Shin Tanaka, Hiroyuki Kurata, Takeshi Ohashi, “Dominant Direction Priority Searching Method for Real-Coded Genetic Algorithm, ” International Conference on Parallel and Distributed Processing Techniques and Applications, pp. 1113–1119, (2003)

[10] W. M. Martin, j. Lienig and J. P. Cohoon, “Island (migration) models: Evolutionary algorithms based on punctuated equilibria”, Handbook of Evolutionary Computation, IOP and Oxford University Press, C. 6. 3, pp. 1–16, (1997).

[11] T. Okuda, T. Hiroyasu, M. Miki, S. Watanabe, “DCMOGA: Distributed Cooperation model of Multi-Objective Genetic Algorithm, ” The Seventh International Conference on Parallel Problem Solving from Nature, The PPSN VII Workshops, pp. 25–26, (2002).

[12] Shin Tanaka, Hiroyuki Kurata, Takeshi Ohashi, “Survival ratio GA for two-evaluation problem in parameter tuning of heat shock response in E. coli, ” IEEE International Conference on Systems, Man and Cybernetics, pp. 2078–2083, (2005).

[13] DeJong, K. A, “An Analysis of the Behavior of a Class of Genetic Adaptive Systems ”, University of Michigan, Ph. D. Dissertation, (1975).

- [14] Choi D. , Hasegawa J. , “The application of a GA with a chromosome limited life for the distribution system loss minimization re-configuration problem, ” Transactions of the Institute of Electrical Engineers of Japan, Part B Vol: 115-B Iss: 7 pp. 741-748, (1995).
- [15] Glossop N. R. , L. C. Lyons and P. E. Hardin, “Interlocked feedback loops within the *Drosophila* circadian oscillator, ” Science 286, pp. 766-768, (1999).
- [16] Ueda, H. R. , Hagiwara, M. and Kitano, H. , “Robust oscillations within the inter-locked feedback model of *Drosophila* circadian rhythm”, J Theor Biol, 210, 401-406, (2001).
- [17] Shin Tanaka, Hiroyuki Kurata, Takeshi Ohashi, “Reverse engineering for a dynamic model of the *Drosophila* circadian oscillator, ” International Conference on Computational Intelligence for Modeling Control and Automation, pp. 107-108, (2005).
- [18] Hiroyuki Kurata, Kouichi Masaki, Yoshiyuki Sumida, Rei Iwasaki, “CADLIVE Dynamic Simulator: Direct Link of Biochemical Networks to Dynamic Models”, Genome Res. , 15: 590-600, (2005).

10. 付録

単一目的関数最適化問題における生存率 GA

本論文では、2つの目的関数を持つ最適化問題を解くために、生存率 GA を用いた手法を提案して検証を行った。しかし、生存率 GA は単一目的関数の最適化問題にも適用可能である。単一目的関数を従来 GA で行う場合では、ある世代で個体が探索を行い、子個体を生成すると、親世代の個体は次世代には生き残られずに消滅してしまう。(ただし、エリート戦略を利用している GA では、最優良個体の数個体が次世代に残留する。) これに対し、生存率 GA では、過去の個体が一定の割合で個体群内に残留し、年代の層を形成する。子個体はこれらの個体を親個体として選択することで、過去の探索履歴を参照した性質継承が可能となる。

単一目的関数最適化問題における生存率 GA の性能を検証するために、大腸菌熱ショック応答シミュレーションを対象問題として実験を行った。

目的関数は本論文で用いた「ピークの高さ」と「ピーク後の定常値」を重み係数 ($w=0.5$) で線形結合した重み関数とした。生存率 GA の設定値、個体数と生存率の値を変えたとき(表 9-1)の最適化成功確率と平均計算回数を示している(図 9-1)。生存率が $w=0.0$ のときは、従来 GA に等しい。最大計算回数は 100000 回とした。

生存率 GA は従来手法($w=0.0$ の場合に等しい)よりも約 2 倍以上の高い成功確率を示した。また生存率という概念を GA に導入することで、個体群内にはこれまでの探索履歴が蓄積され、子個体はそれらを参照して性質継承を受けるため、従来手法よりも多様な性質を獲得する事ができたため、成功確率は高くなり、同時に平均計算回数も少なくなった。生存率 GA は大腸菌熱ショック応答の単一目的関数最適化問題においても良好な結果を示した。

表 9-1. 生存率 GA の条件設定

生存率 GA

個体数:50, 100, 200 個体

生存率:0.0~1.0 (0.1 刻み)

交叉:UNDX

試行回数:20 試行

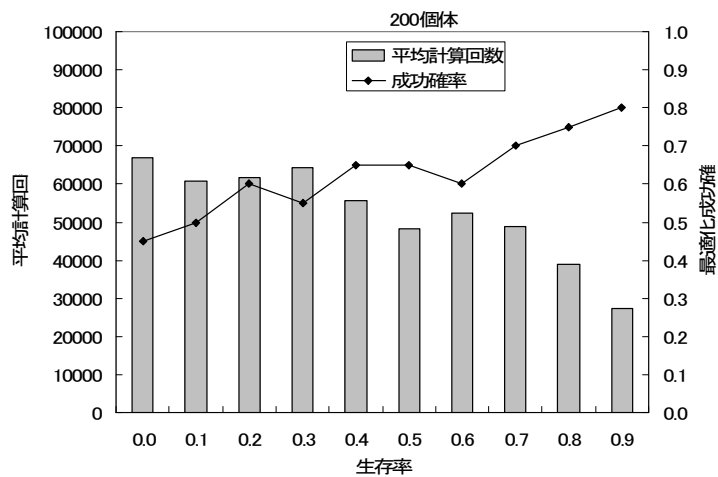
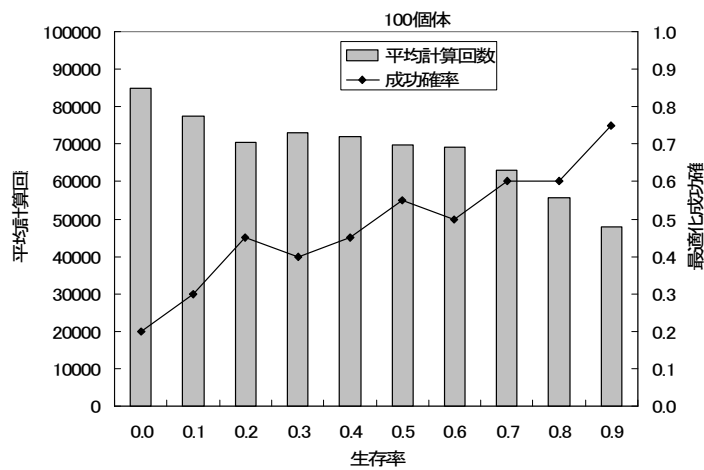
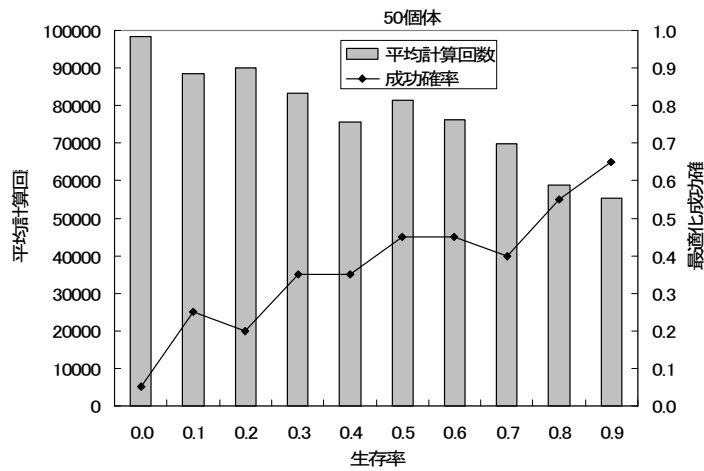


図 9-1. 単一目的関数(熱ショック応答)における生存率 GA の最適化成功率と平均計算回数.

探索開始時の個体数と生存率の設定を変えた場合の結果