

大規模問題に対する遺伝的アルゴリズム
の収束性能の向上：電力システム
における最適化問題への適用

所 健一

目次

第1章 序論	4
1.1 研究の背景	4
1.2 メタヒューリスティクス	5
1.2.1 タブサーチ	6
1.2.2 遺伝的アルゴリズム	6
1.2.3 シミュレーテッド・アニーリング	7
1.3 メタヒューリスティクスを用いた電力システムの最適化	8
1.3.1 配電・送電系統計画	8
1.3.2 FACTS機器の設備計画	8
1.3.3 最大電力予測	9
1.3.4 発電機の起動停止計画	10
1.3.5 電源補修／作業停止計画	10
1.3.6 事故時復旧操作	10
1.3.7 電圧無効電力制御	11
1.4 電力システム最適化アルゴリズムの開発	11
1.4.1 実用化のために解決すべき課題	11
1.4.2 遺伝的アルゴリズムの選択理由	12
1.5 対象とする最適化問題	13
1.5.1 割当て問題	13
1.5.2 混合整数計画問題	14
1.5.3 確率計画問題	14
1.6 論文の構成	15
第2章 遺伝的アルゴリズム	17
2.1 遺伝的アルゴリズムの概要	17
2.1.1 遺伝的アルゴリズムによる解の探索手順	17
2.1.2 染色体表現	20
2.1.3 適合度	20
2.1.4 スケーリング	21

2.1.5	個体選択	22
2.1.6	交叉	24
2.1.7	突然変異	25
2.2	遺伝的アルゴリズムの最適性	25
2.2.1	スキーマ定理	25
2.2.2	NFL 定理	27
2.3	遺伝的アルゴリズムの特徴	31
2.4	本論文での遺伝的アルゴリズムの定義	31
第 3 章	割当て問題のための染色体表現方法	33
3.1	電力システムにおける割当て問題	33
3.2	周波数割当て問題の定式化	34
3.3	開発アルゴリズム	36
3.3.1	発見的手法：再帰的割当て	37
3.3.2	発見的手法：AAF ヒューリスティック	38
3.3.3	提案アルゴリズム	39
3.4	適用結果	40
3.4.1	ベンチマーク問題への適用	41
3.4.2	九州電力における周波数割当て問題への適用	43
第 4 章	混合整数計画問題のための個体評価方法	46
4.1	電力システムにおける混合整数計画問題	46
4.2	発電機起動停止計画問題の定式化	49
4.2.1	記号一覧	49
4.3	アルゴリズム	51
4.3.1	初期集団の生成	52
4.3.2	個体評価	53
4.3.3	個体の選択・再生	58
4.3.4	終了判定	59
4.3.5	開発アルゴリズムの特徴	60
4.4	数値実験	60
4.4.1	ベンチマーク問題	60
4.4.2	遺伝的アルゴリズムのパラメータ設定	62
4.4.3	提案計画手法の性能評価	62
4.4.4	潮流制約の追加	66

第5章	確率計画問題のための個体選択方法	71
5.1	電力システムにおける確率計画問題	71
5.2	確率計画問題への遺伝的アルゴリズムの適用	73
5.3	アルゴリズム	74
5.3.1	個体評価	74
5.3.2	$z'(S)$ の分布	75
5.3.3	統計的選択方法	76
5.3.4	開発アルゴリズムのフロー	77
5.3.5	開発アルゴリズムの特徴	79
5.4	数値実験	79
5.4.1	確率的施設配置問題の定式化	79
5.4.2	染色体表現	81
5.4.3	シナリオ	82
5.4.4	遺伝的操作とパラメータ設定	82
5.4.5	結果	82
第6章	おわりに	90

第1章 序論

1.1 研究の背景

電力自由化の進展と、社会の電力依存度の上昇を背景に、電力会社ではコスト削減と、供給信頼度の維持・向上とが、重要な経営課題となっている。自由化の導入以前は、地域毎に電力会社が独占して電力供給を行っていたが、平成12年3月に電力市場が自由化され、契約電力が2,000kW以上の需要家については、電力会社以外の事業者からも電力を購入できるようになった。その後、電力自由化の適用範囲は順次拡大され、現在では契約電力が50kW以上の需要家までが自由化の対象となっており、さらなる適用範囲の拡大も検討されている。こうした状況の下、電力会社では、より低廉な電力の供給へ向けた、電力供給コスト削減のための各種取組みが行われている。

一方で、電力会社には供給信頼度の維持・向上が望まれている。電力依存度の上昇にともない、電力の供給停止は社会により大きな影響を与えるようになってきている。多くの企業では、日常業務にコンピュータをはじめとするIT機器を利用しており、電力供給が停止すると円滑な業務が行えなくなる。また、IHクッキングヒータやヒートポンプ式給湯機の普及により、空調、調理、給湯のすべてを電力で賄うオール電化ハウス/マンションが増加しているが、こうしたオール電化ハウス/マンションに暮らす家庭では、電力の供給停止により日常生活に重大な支障が生じる。

電力会社において、要求される電力の供給信頼度レベルを維持しつつ、供給コストを削減していくには、電力供給に密接に関連した電力システムの最適化が特に重要となる。要求信頼度とコストのバランスを考えて、電力システムの設計・運用・制御を最適に行っていく必要がある。こうした電力システムの最適化を行う際に利用可能なツールの一つに、最適化手法がある。

最適化手法とは、対象とする問題を数学的な問題としてモデル化し、この問題を解くことで最適解を求める手法であり、その代表に線形計画法

(LP: Linear Programming) をはじめとする数理最適化手法がある。しかし、電力システムに関する最適化問題には以下の特徴があり [1]、数理最適化手法が適用可能な問題は限定される。

- 時間をかけて大域的最適解を得るより、限られた時間内にできる限り良質な解を得ることが求められる問題が多い。
- 現場の運用ルールなど、数式として表現できない制約条件が多数存在し、これを考慮できなければ実用化が難しい問題が多い。
- 電力の潮流計算など、問題向けのパッケージが既に開発されており、例えば目的関数値の計算などにこうしたパッケージを利用することで、大幅なシステム開発コストの削減や、開発期間の短縮が行える問題が多い。
- 目的関数や制約条件が単純な線形関数や微分可能な関数では記述できず、線形化などのモデル化を行って求めた厳密解では、実用的な解とならない問題が多い。

一方、上記の数理最適化手法の課題を解決する新たな最適化手法として、メタヒューリスティクスと呼ばれる最適化手法が大きな注目を集めている。

1.2 メタヒューリスティクス

メタヒューリスティクスとは、単純なルールや発見的手法を反復的に用いることで、効率的に解を求める手法の総称であり、F.Glover がタブサーチを提案する論文の中で最初に使ったとされている。メタヒューリスティクスに関しては様々な研究が行われ、これに計算機性能の飛躍的な向上が合わさることで、これまで最適化手法の適用が困難であった大規模・複雑な問題についても、メタヒューリスティックを用いて実用的な計算時間で解くことが可能となっている。代表的なメタヒューリスティクスに、タブサーチ、遺伝的アルゴリズム、シミュレーテッド・アニーリングの三つの手法がある。以下、それぞれの手法の概要について説明する。

1.2.1 タブサーチ

タブサーチ (Tabu Search) は, 1989年に F. Glover によって提案されたアルゴリズムである [2, 3]。タブリストと呼ぶリストに既に通過した解を記録しておき, このリストに記録された解には戻らないようにすることから, タブサーチと名付けられている。タブサーチによる最適解 (目的関数を最大とする解) の探索手順は以下のようなになる。

ステップ1 解空間の中から, 初期解 x_0 を選び, $x^* = x_0$ とする。 $k = 0$ とし, タブリスト $T = \phi$ とする。

ステップ2 x_k の近傍 $N(x_k)$ に属し, かつタブリスト T に含まれない解の中で, 最良の解を選び x_{k+1} とする。

ステップ3 タブリストを更新する (T に x_k を加える)。

ステップ4 $x_{k+1} \geq x^*$ ならば $x^* = x_{k+1}$ とする。

ステップ5 終了条件を満たしていたならアルゴリズムを終了し, x^* を最適解として出力する。もし終了条件を満たしていなければ, $k = k+1$ としてステップ2からの操作を繰り返す。なお, 終了条件としては, 繰り返し回数 k の値があらかじめ決めた一定数に達したらアルゴリズムを終了する, などの条件が採用されるのが一般的である。

1.2.2 遺伝的アルゴリズム

遺伝的アルゴリズム (GA: Genetic Algorithm) は, 1970年代に J. H. Holland によって行われた, 人工進化システムの研究に端を発している [4]。その後, 組合せ最適化問題などでの近似解の探索手法としての研究が進められ, 様々な最適化問題への適用が検討された [5, 6, 7, 8]。遺伝的アルゴリズムによる最適解探索の基本的手順を以下に示す [9]。

ステップ1 染色体に解をコード化した個体をランダムに N 個生成する。ここで N は事前に決めた個体数を表している。このランダムに生成した N 個の個体を初期集団と呼ぶ。

ステップ2 染色体にコード化された解に対する目的関数値を基に, 各個体の適合度を計算する。

ステップ3 終了条件を満たしていたならアルゴリズムを終了し、それまでに生成された個体のうち、適合度が最大となる個体の染色体にコード化されている解を最適解として出力する。終了条件としては、世代数があらかじめ決めた一定数に達したらアルゴリズムを終了するなどの条件が一般的である。

ステップ4 各個体の適合度に比例した確率で、次世代へと生き残らせる個体を選択する。

ステップ5 生き残らせた個体に交叉と突然変異の遺伝的操作を加えて新しい個体を生成し、ステップ2からの操作を繰り返す。交叉とは二つの個体の染色体を同じ位置で切断し、お互いに染色体の一部を交換する操作である。突然変異とは、確率的に選択した遺伝子の値を、他の選択可能な染色体（対立遺伝子）の値へと変化させる操作である。

1.2.3 シミュレーテッド・アニーリング

シミュレーテッド・アニーリングとは、1983年に Kirkpatrick らによって提案された、鉄などの焼きなまし（アニーリング）の物理現象をモデル化して作られた探索アルゴリズムである [10]。鉄などの焼きなましでは、加熱によって物体の運動エネルギーを高めた後、ゆっくりと冷却することで物体の内部エネルギーが最小となる状態を作り出す。この物理現象をモデル化したシミュレーテッド・アニーリングでは、目的関数を内部エネルギーとしてとらえ、ランダムな状態遷移が支配的な状態から、目的関数の最小化が支配的な状態へと、温度に相当するパラメータを徐々に減少させながら探索を行う。シミュレーテッド・アニーリングによる最適解（目的関数を最大とする解）の探索手順は以下ようになる。

ステップ1 解空間の中から、初期解 x_k ($k = 0$) をランダム選び、初期温度 T_k を設定する。

ステップ2 x_k の近傍 $N(x_k)$ から、 y をランダムに選ぶ。

ステップ3 $\delta = f(y) - f(x) \geq 0$ なら、 $x_{k+1} = y$ とする。 $\delta < 0$ ならば、確率 $\exp(\delta/T_k)$ で $x_{k+1} = y$ とし、確率 $1 - \exp(\delta/T_k)$ でステップ2に戻る。

ステップ4 ステップ2からステップ3の繰り返し回数が、あらかじめ定めた反復数を超過していれば、あらかじめ定めた冷却スケジュールに従い温度 T_k の値を減少し、ステップ5へ進む。そうでなければ $k = k + 1$ として、ステップ2に戻る。

ステップ5 終了条件を満たしていたならアルゴリズムを終了する。終了条件を満たしていないなら、 $k = k + 1$ としてステップ2へ戻る。

1.3 メタヒューリスティクスを用いた電力システムの最適化

電力システムにおける最適化問題に対するメタヒューリスティクスの適用に関しては、例えば以下のような研究が行われている [11]。

1.3.1 配電・送電系統計画

配電線を流れる電力の電圧を規定値以下に抑えつつ、配電損失（配電の過程で失われる電力）が最小となる、最適な配電系統構成を求める遺伝的アルゴリズムが提案されている [12]。配電系統の構成は開閉器の状態により変化する。そこで、この遺伝的アルゴリズムでは、全開閉器の状態を染色体にコード化して、遺伝的アルゴリズムのメカニズムを用いて、最適な系統構成（開閉器の状態）を探索する。

また、与えられた電力負荷の増加に対して、系統コスト（フィーダーや変電所の増強コストと、配電損失コストの合計）が最小となる、最適な配電系統の拡張計画を求めるタブサーチに基づく計画手法 [13] や、分散型電源の最適な配置場所と、配電系統の設備計画を求める遺伝的アルゴリズム [14] が提案されている。

上記の研究は配電系統の最適化を扱ったものであるが、送電系統の最適化に関しては、送電線の容量を制約に、線路の設置コストが最小となる計画をタブサーチにより求める研究が行われている [15]。

1.3.2 FACTS 機器の設備計画

FACTS (Flexible AC Transmission System) とは、送電線を流れる電力の量をコントロールする機器の名称である。基本的に送電線の各枝線

を流れる電力の量はコントロールすることができないが、FACTS機器を適切な場所に設置することで、各枝線を流れる電力量のコントロールが可能となる。松尾らは、メッシュ系統の一部に線路容量の超過が発生した場合に、どの枝線に何台のFACTS機器を設置したら良いかをタブサーチにより探索する手法を提案している [16]。

また、森らは電圧の変動制御を目的に、配電系統用のFACTS機器であるD-FACTSの最適配置を行う2層構造のタブサーチを提案している [17]。このタブサーチでは、1層目のタブサーチでD-FACTSの設置場所を探索し、2層目のタブサーチでD-FACTSの出力を決定する。

さらに、電圧の上下制限約と、線路を流れる電力潮流の上下制限約の下、設置費用の最小化と、電圧安定性維持、電圧逸脱の回避を目的に、ステップ式自動電圧調整器 (SVR: Step Voltage Regulator) と静止型無効電力保障装置 (SVC: Static Var Compensator) の協調を考えた最適な設置場所を求める、タブサーチに基づく手法が提案されている [18]

1.3.3 最大電力予測

需給運用計画の基礎となる、翌日の最大電力を予測するための数多くの研究が行われており、ニューラル・ネットワークやファジー推論モデルを用いた予測手法が提案されている。こうした予測手法では、ニューラルネットワークやファジー推論モデルの最適な構造/パラメータを決定するために、メタヒューリスティクスを利用している。ニューラルネットワークでは学習を始める際に、中間ユニット数や層数などの構造をあらかじめ決める必要がある。こうした構造の決定は組合せ最適化問題として定式化され、最適解を得ることが難しい。そこで、遺伝的アルゴリズムやタブサーチを用いて、最大電力を予測するニューラルネットワークの最適構造を求める研究が行われている [19, 20, 21, 22]。

また、ファジー推論では、メンバーシップ関数の数と位置を最適化する必要があり、これらの最適化のためにタブサーチが利用されている [23]。

さらに雪田らは、構造的な表現を扱えるよう拡張した構造的遺伝的アルゴリズムと、多変量解析手法の一つであるGMDH (Group Method of Data Handling) を組合せ、翌日最大電力の予測に用いる変数選択と予測式の生成を行う手法を提案している [24]。

1.3.4 発電機の起動停止計画

発電機の起動停止計画は、需給バランス、供給予備力や発電機出力の上下限值などの種々の制約を満たし、発電コストが最小となる、各発電機の起動停止状態と発電出力とを同時決定する問題である。電力会社では、日々数多くの発電機を稼働させており、起動停止計画はコスト削減を行ううえで、非常に重要な計画の一つとなっている。こうした起動停止計画問題に対しては、発電機の起動停止状態をタブサーチで探索し、発電機の出力は等ラムダ法を用いて決定する手法が提案されている [25]。また、 N 台の発電機の各時間帯での起動停止状態を $N \times T$ ビットの遺伝子で表し、最適な起動停止状態を探索する遺伝的アルゴリズムが提案されている [26]。

1.3.5 電源補修／作業停止計画

定期点検などによる設備停止要求に対して、作業の緊急度、需給状況、系統状況などを考慮して、最適な作業実施日をタブサーチを基に計画する手法が提案されている [27]。提案されている手法では、潮流（電力の流れ）に関する制約と、同時実行が不可能な作業に関する制約を条件に、最適な作業開始日をタブサーチを用いて探索する。

また、法律で義務化されたボイラー／タービンの点検期間や、電力需要、出水率（水力発電所のある川の水量）、電源設備の計画外停止などを考慮したうえで、所有する発電機の最適な補修スケジュールを求める、タブサーチを用いた手法 [28] や、シミュレーテッド・アニーリングを用いた計画手法 [29]、またシミュレーテッド・アニーリングと遺伝的アルゴリズムを融合した計画手法 [30] が提案されている。

1.3.6 事故時復旧操作

事故により停電が発生した場合には、変圧器容量の制約内で、停電区間が最小となるように開閉器を操作する。この開閉器の操作を事故時復旧操作と呼ぶ。事故時復旧操作の目的とする、最適な復旧操作後の系統構成（復旧系統）を決定する問題は、開閉器の開／閉状態を変数とする組合せ最適化問題として定式化される。事故時に迅速に復旧系統を求めるため、この問題へのメタヒューリスティクスの適用が検討されており、タブサーチを用いて最適な復旧系統を求める手法が提案されている [31]。

また、停電区間の負荷量の最小化と、復旧に要する時間の最小化の2目的に対するパレート最適解を求める、遺伝的アルゴリズムとシミュレーテッドアニーリングの組合せアルゴリズム [32] や、並列タブサーチを用いて、効率的に最適な開閉器操作の手順を求める手法が提案されている [33]。

1.3.7 電圧無効電力制御

太陽光発電など、出力変動の大きい分散型電源の普及拡大に伴って、無効電力設備等によって制約条件を満足しながら電力損失を軽減し、より多くの高品質な電力を供給する、電圧無効電力制御問題が重要となっている。こうした無効電力制御問題に対して、遺伝的アルゴリズムとタブサーチの組合せにより、最適な調整用設備機器の切り替え手順を求めるアルゴリズムが提案されている [34]。このアルゴリズムでは、遺伝的アルゴリズムの個体生成過程にタブサーチを加え、交叉とタブサーチにより新たな個体を生成する。

1.4 電力システム最適化アルゴリズムの開発

上述のように、様々な電力システムに関する最適化問題へと、メタヒューリスティクスを適用する研究が行われている。しかし、その多くは研究レベルの段階にあり、電力システムの最適化問題におけるメタヒューリスティクスの実用化を進めるには、以下に述べる収束性能の向上など、解決しなければならない課題が残されている。そこで、他の手法との融合が容易な遺伝的アルゴリズムの特徴を活かすことで、実用化への課題を解決し、電力システム最適化の実問題へと適用可能なアルゴリズムを開発することを検討した。

1.4.1 実用化のために解決すべき課題

電力システムに関する最適化問題における、メタヒューリスティクスの実用化を進めていくには、以下の二つの課題解決が望まれる。

- 大規模問題での収束性能の向上
- 最悪ケースにおける解の精度保障

電力会社の供給エリアは広大であり、このエリアへ電力を供給するのに利用される電力システムは大規模なシステムとなる。このため電力システムの最適化問題は、一般に大規模問題として定式化される。こうした大規模問題にメタヒューリスティクスを適用した場合、実用化のためには収束性能の向上（計算効率の向上と得られる解の精度向上）が求められるケースが多い。メタヒューリスティクスを用いて効率良く、精度の高い解を得るには、大域的探索能力と局所探索能力のバランスが重要となるが、遺伝的アルゴリズムの場合には、大域的探索能力は高い反面、局所探索能力が低く、収束性能向上のためには局所探索能力の向上が必要となる。

また、常に大域的最適解が得られる保証が無い、メタヒューリスティクスを電力システム最適化の実問題へと適用しようとした場合、最悪ケースでの精度保障（大域的最適解と比べた目的関数値の悪化度合いが、どの程度に抑えられるか）を求められるケースが多い。しかし、メタヒューリスティクスの適用において、こうした精度保障を行うことは困難である。遺伝的アルゴリズムの最適性を保障する定理としてスキーマ定理（2.2参照）などがあるが、こうした定理を用いて精度保障を行うことはできない。

1.4.2 遺伝的アルゴリズムの選択理由

本論文では、他手法との融合が容易であるという遺伝的アルゴリズムの特徴に着目し、この特徴を活かすことで、上述の実用化のための課題を解決する遺伝的アルゴリズムを開発した。

遺伝的アルゴリズムと他手法とを融合することで、最悪ケースでの精度が間接的に保障されたアルゴリズムを開発することが可能である。例えば、これまで担当者が何らかの経験則を基に最適化を行っていた問題であれば、この経験則を遺伝的アルゴリズムに組み入れることで、最悪でも経験則で求まる解が得られることが保障されたアルゴリズムを開発できる。

さらに、遺伝的アルゴリズムと他手法を融合することで、収束性能を向上させることが可能である。探索する解空間の集中化が行われるように、他の手法を遺伝的アルゴリズムのメカニズムの中で活用すれば、交叉・突然変異のみによる探索を行った場合と比較して、局所探索能力や解探索の効率を向上することができる。

表 1.1: 本論文で検討する 3 タイプの最適化問題

問題のタイプ	電力システムにおける最適化問題の例
割当て問題	電力用移動無線の周波数割当て問題 発電機の部品のローテーション計画問題など
混合整数計画問題	発電機の起動停止計画問題 系統の切替え計画問題など
確率計画問題	長期の電源計画問題 電力設備の施設配置問題など

具体的には、3章で説明する割当て問題を解くアルゴリズムでは、グリーディアルゴリズムとの組合せにより、遺伝的アルゴリズムの局所探索能力を向上し、さらにグリーディアルゴリズム以上の解が得られることを保障している。また、4章で説明する混合整数計画問題を解くアルゴリズムでは、連続緩和法との融合により、局所探索能力を向上させるとともに、連続緩和法より優れた解が得られることを保障している。さらに5章で説明する確率計画問題を解くアルゴリズムでは、統計的検定手法との組合せにより、遺伝的アルゴリズムの世代進行のメカニズムを活用した計算効率の向上を実現している。

1.5 対象とする最適化問題

電力システムにおける最適化問題を解くアルゴリズムを開発するにあたり、全ての問題に対して有効な万能アルゴリズムを開発できれば理想的である。しかし、NFL (No Free Lunch) 定理 (2.2.2 参照) が証明するように、全てのタイプの問題を効率的に解くアルゴリズムを開発することは困難である。そこで、本論文では電力システムにおける最適化問題を代表する、表 1.1 の 3 タイプの最適化問題に対して、それぞれの問題に適したアルゴリズムを開発した。

1.5.1 割当て問題

式 (1.1) から式 (1.4) により定式化される、2部グラフの完全マッチング問題を割当て問題と呼ぶ。

$$\text{Minimize } f(x_{ij}) \quad (1.1)$$

$$\text{subject to } \sum_{i=1}^n x_{ij} = 1 \quad (1.2)$$

$$\sum_{j=1}^m x_{ij} = 1 \quad (1.3)$$

$$x_{ij} \in \{0,1\} \quad (1.4)$$

割当て問題に分類される電力システムの最適化問題の例としては、電力用移動無線システムの周波数割当て問題や、発電機部品のローテーション計画問題などがある。電力用移動無線の周波数割当て問題とは、災害時の復旧作業などに用いられる電力用移動無線システムの基地局に対して、干渉する基地局では同一の周波数チャンネルを利用しないという条件の下で、周波数を有効利用するよう（基地局全体で使用する周波数チャンネル数が最小となるように）、各基地局へ周波数を割り当てる問題である。また、発電機部品のローテーション計画問題とは、同一の部品が利用可能な発電機の間で、法定点検のスケジュールに従い、部品寿命が無駄にならない、最適な部品ローテーションの計画を求める問題である。

1.5.2 混合整数計画問題

決定変数のうちの一部に整数変数を含む最適化問題が混合整数計画問題である。例えば、発電機の起動停止計画問題や、系統の切り替え計画問題などが、電力システムに関する最適化問題のうち、混合整数計画問題に分類される問題の代表である。発電コストが最小となる、発電機の各時間帯での起動・停止状態と、出力を同時決定する起動停止計画問題では、発電機の状態は整数変数として表現されるため、混合整数計画問題として定式化される。また、系統切り替え計画問題では、開閉器の状態が整数変数として表現されるため、混合整数計画問題として定式化される。

1.5.3 確率計画問題

一般的な最適化手法は確定環境下における最適化を対象としており、与えられた目的関数や制約に関する一組の入力パラメータに対して最適な

解を求める。しかし、現実の計画問題などでは、パラメータの実現値が得られる以前の段階で、意思決定をしなければならないケースも多い。こうした不確実環境下における最適化を行うことを目的とした最適化問題が、確率計画問題である。確率計画問題には何タイプかの問題があるが、例えば、起こり得る目的関数や制約のパラメータの組合せが複数のシナリオとして与えられ、このシナリオに対して平均的に高い最適性を発揮する解を求める問題が、確率計画問題の代表的な問題である。電力システムでも、将来の需要データなど、不確実なパラメータを基に最適化を行わなければならない計画問題が多数存在する。長期の需要変動を考えた電源計画などは、確率計画問題として定式化される問題の代表である。また、将来の需要を考慮した設備の最適配置計画なども、確率計画問題として定式化される。

1.6 論文の構成

本論文の構成は以下の通りである。次章において、本論文での議論の前提となる遺伝的アルゴリズムの概要と、その最適性の根拠となるスキーマ定理について説明する。また、全ての最適化問題に対して最高性能を発揮する万能アルゴリズムは存在しないことを示した、NFL定理についても説明する。

3章では、グリーディアルゴリズムとの組合せにより、大規模割当て問題を解く遺伝的アルゴリズムについて述べる。このアルゴリズムでは、染色体にコード化した割当て優先度と、グリーディアルゴリズムにより計算される割当て優先度の組合せにより割当て順序を決定する。これによりグリーディアルゴリズム以上の解が得られることが保障される。また、グリーディアルゴリズムを組み合わせることで局所探索能力が向上し、移動無線の周波数割当てのベンチマーク問題に適用したところ、既存手法と比べ効率良く、使用周波数の帯域幅が最小となる周波数割当てを求めることができた。

4章では、大規模な非線形混合整数計画問題を解く、個体評価に連続緩和法を組み入れたことを特長とする遺伝的アルゴリズムを説明する。このアルゴリズムでは、一部の整数変数の値を固定し、残りを未固定とした部分解を個体の染色体にコード化する。そして、個体評価の際には、未固定となっている整数変数と連続変数の値を、連続緩和問題の最適解を基に決定する。このアルゴリズムでは、連続緩和法を遺伝的アルゴリズム

ムと組み合わせることで、連続緩和法より優れた解が得られることを保障するとともに、遺伝的アルゴリズムの局所探索能力を向上させている。このアルゴリズムを大規模な非線形混合整数計画問題の代表である、発電機の起動停止計画のベンチマーク問題に適用したところ、既存の計画手法では求めることのできなかつた、経済的な起動停止計画の策定が可能となった。

5章では、多数のシナリオが与えられる確率計画問題の解を効率的に求める、統計的検定を用いて個体選択を行う遺伝的アルゴリズムについて述べる。このアルゴリズムでは、ランダムに選択したシナリオを用いて個体を評価することで、個体評価に要する計算時間を削減する。また、統計的検定により有意に最良個体とはなり得ないと判断された個体のみを淘汰する個体選択を行うことで、サンプリング誤差により適合度の高い個体が淘汰されてしまう危険率を一定値以下に抑える。提案アルゴリズムを確率計画問題に分類される、電力設備の配置計画問題へ適用したところ、多数シナリオに対する期待利益が最大となる配置案を短時間で求めることができた。

第2章 遺伝的アルゴリズム

2.1 遺伝的アルゴリズムの概要

遺伝的アルゴリズム (Genetic Algorithm: GA) は、自然界の生物システムの遺伝と自然選択による適応過程に着想を得て、ミシガン大学の J.H. Holland により提案されたアルゴリズムである [4]。Holland らの研究では、自然界の生物システムの適応プロセス、および自然の重要なメカニズムを有する人工的システムの設計を目的としていたが、1970年代に入って DeJong により関数最適化問題へ遺伝的アルゴリズムを適用する計算機実験が試みられた。さらに 1989年には Goldberg によってアルゴリズムの枠組みが整理され [35]、最適化問題の解法としての利用に注目が集まるようになった。

自然界の生物の進化過程においては、ある世代を構成する個体集団の中で、環境への適合度の高い個体が高い確率で生き残り、さらに個体を特徴付ける、染色体（複数の遺伝子の集まり）に交叉や突然変異が加わることで、次世代の新たな個体集団が形成されていく。遺伝的アルゴリズムでは、この生物の進化過程を模擬して、最適化問題の解探索などを行う。以下に一般的な遺伝的アルゴリズムを用いた解の探索手順を示す。

2.1.1 遺伝的アルゴリズムによる解の探索手順

遺伝的アルゴリズムでは、図 2.1 に示したフローにより、環境への適合度の高い個体を探索する。図 2.1 のフローにおける処理の概要は以下のとおりである。

- (1) **初期集団の生成:** 染色体の各遺伝子に解に関する情報をコード化した個体を、あらかじめ決められた集団サイズの数だけ生成する。
- (2) **選択・再生:** 各個体の適合度を計算し、その適合度の高さに比例した確率で、個体を次世代へと選択・複製する。

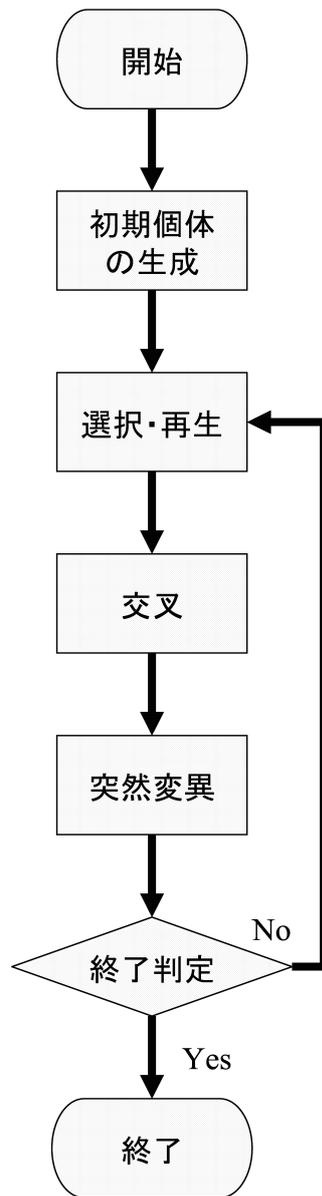


図 2.1: GA の基本的フロー

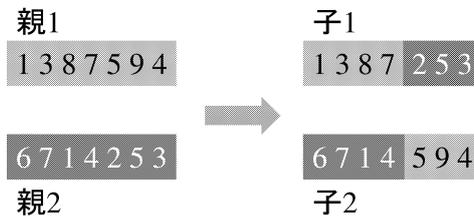


図 2.2: 交叉の例

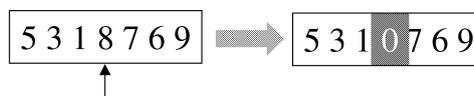


図 2.3: 突然変異の例

- (3) **交叉:** (2) で選択した個体からランダムにペアを作り，このペアに対し一定の確率にしたがって交叉の遺伝的操作を適用する。例えば交叉の代表的な遺伝的操作である一点交叉では，図 2.2 のように 2 つの染色体をランダムに選んだ同じ位置で切断し，これを連結して新しい子の個体ペアを生成する。
- (4) **突然変異:** 各個体の遺伝子に対して一定の突然変異確率にしたがって，突然変異の遺伝的操作を適用する。例えば突然変異の代表的な操作である一様変異では，図 2.3 に示したように，染色体の遺伝子をランダムに選び，この遺伝子の値を他の対立遺伝子の値へと置き換える。
- (5) **終了判定:** 終了条件が満たされたなら，それまでに得られた最良の個体を解とする。そうでなければ (2) へ戻る。

なお，(5) での終了条件としては，

- 得られた最良個体の適合度が，あらかじめ設定した下限値を超える
- 集団の平均適合度が，あらかじめ設定された値を超える
- 集団の世代数が，あらかじめ設定した数に達する

のいずれかが採用されるのが一般的である。

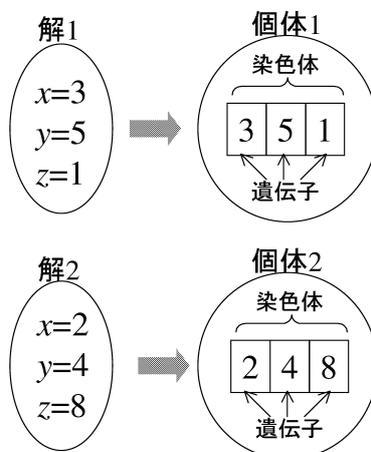


図 2.4: 染色体へのコード化

2.1.2 染色体表現

遺伝的アルゴリズムでは、例えば図 2.4 のように、一つの解候補を一つの個体として捉え、記号列として表現した解に関する情報を、個体の染色体にコード化する。染色体上で 1 個の遺伝子が占める位置を遺伝子座、同じ遺伝子座を占め得る何種類かの遺伝子に対立遺伝子と呼ぶ。

n 個の記号の並び $x = (x_1, x_2, \dots, x_n)$ を n 個の遺伝子座からなる染色体とすると、 x_i は第 i 遺伝子座における遺伝子であり、 x_i の取り得る値が対立遺伝子である。対立遺伝子としては、ある整数の組、ある範囲の実数値、単なる記号など、様々なものを考えることができる。最も基本的なのは 0 と 1 の 2 値を考える場合である。このように表される染色体は個体の遺伝子型と呼ばれる。また、染色体に対応して定まる変数 x の値は表現型と呼ばれる。問題によっては冗長性を持たせたほうが有利な場合もあるが、通常、遺伝子型と表現型は 1 対 1 に対応させる。

2.1.3 適合度

遺伝的アルゴリズムでは、染色体にコード化された情報に基づき計算される、適合度により個体を評価する。一般に適合度は最適化問題の目的関数を基に計算するが、対象とする最適化問題が最小化問題の場合は、

個体 i の適合度 f_i は例えば式 (2.1) で計算される。

$$f_i = \begin{cases} C_{\max} - z_i & z_i \leq C_{\max} \\ 0 & z_i > C_{\max} \end{cases} \quad (2.1)$$

ここで z_i は個体 i の染色体にコード化された解に対応した目的関数値であり、 C_{\max} は遺伝的アルゴリズムの探索過程において、それまでに得られた z_i の最大値を表している。

また、対象とする最適化問題が最大化問題の場合は、個体の適合度 f_i は例えば式 (2.2) で計算される。

$$f_i = \begin{cases} z_i + C_{\min} & z_i + C_{\min} \geq 0 \\ 0 & z_i + C_{\min} < 0 \end{cases} \quad (2.2)$$

ここで C_{\min} は遺伝的アルゴリズムの探索過程において、それまでに得られた z_i の最小値を表す。

2.1.4 スケーリング

次世代へと生き残らせる個体を選択する際に、適合度の値をそのまま個体の選択確率に反映させるのではなく、何らかの関数を用いて、適合度の差異を拡大／縮小させたほうが、より効果的な選択が行われる場合がある。このような適合度の差異を拡大／縮小する目的で行われるのが、適合度のスケーリングである。

適合度のスケーリングとしては、

- 線形スケーリング
- シグマスケーリング
- べき乗スケーリング

などの方法が一般的に用いられる。それぞれのスケーリング方法の概要を以下に示す。

線形スケーリング

個体 i の適合度 f_i を式 (2.3) の線形写像により変換する方法である。

$$f'_i = a \cdot f_i + b \quad (2.3)$$

例えば後述するルーレット選択を行う際には、集団の適合度の平均 \bar{f} と同じ適合度の個体が、確率的に最低1個は再生されるよう、 $\bar{f} = \bar{f}'$ となるように変換するのが一般的である。なお、線形スケーリングで $f'_i < 0$ となる場合は、 $f'_i = 0$ とする。

シグマスケーリング

個体集合の中で、多くの個体が高い適合度を持ち、残りの個体が非常に小さな適合度を持つ場合、線形スケーリングを用いると $f'_i < 0$ となる個体が多くなる。こうしたケースでは、シグマスケーリングが有効であるとされている。シグマスケーリングは式 (2.4) によりスケーリング後の適合度 f'_i を計算する方法である。

$$f'_i = f_i - (\bar{f} - c \cdot \sigma) \quad (2.4)$$

ここで、 \bar{f} は個体集団の平均適合度を、 σ は分散をそれぞれ表している。 c は1から3の値を取る定数で、 $f'_i < 0$ であれば $f'_i = 0$ と変換される。

べき乗スケーリング

個体 i の適合度 f_i を式 (2.5) より計算するスケーリング方法であり、 k は世代の進行に応じて変化するパラメータである。

$$f'_i = (f_i)^k \quad (2.5)$$

2.1.5 個体選択

適合度の高い個体が次世代により多くの子孫を残すという自然淘汰の考えは、遺伝的アルゴリズムでは選択と呼ばれ、適合度の高い個体は選択され次世代へと生き残り、適合度の低い個体は淘汰される。自然淘汰の考えに基づく個体選択の規則として、

- ルーレット選択
- 期待値選択
- ランキング選択

- トーナメント選択
- エリート保存選択

などがある。以下、それぞれの選択規則の概要を述べる。

ルーレット選択

ルーレット選択は、集団の中の各個体のスケーリング済みの適合度 f'_i とその総計を求めて、適合度の総計に対する各個体の割合を、その個体の選択確率とする選択規則である。つまりルーレット選択において、個体 i が選択される確率 p_i は式 (2.6) で計算される。

$$p_i = \frac{f'_i}{\sum_{j=1}^n f'_j} \quad (2.6)$$

期待値選択

ルーレット選択に代表される確率的な選択方法では、集団の個体数が十分に大きくない場合、乱数の揺らぎによって適合度が正確に反映されない選択が行われる危険性がある。このような問題に対応するために、期待値選択が提案されている。期待値選択では、集団内の各個体の適合度とその総計を計算し、適合度の総計に対する各個体の割合を基に個体数を調整する。

ランキング選択

ルーレット選択や期待値選択のように、適合度の値を基に個体を選択した場合、適合度の非常に高い個体が存在すると、集団の個体が同じものばかりになってしまう。また、集団の個体の適合度にほとんど差が無い場合には、良い個体がなかなか増殖しないという問題が発生する。こうした問題に対して、バランス良く再生を行うことを目的にランキング選択が提案されている。ランキング選択では、各個体の適合度に基づいて個体をランキング付け（順位付け）する。そして、各順位毎にあらかじめ決めておいた数の個体を選択する。

トーナメント選択

個体集団の中から、あらかじめ決められた数の個体をランダムに選び出し、その中で適合度のもっとも高い個体を次世代に残すという操作を、次世代に残す数の個体を選択されるまで繰り返す選択方法である。

エリート保存選択

確率的に個体を選択し、これに後述する交叉や突然変異の遺伝的操作を加えると、遺伝的アルゴリズムによる探索の過程で非常に良い個体が生成されたにもかかわらず、この個体の染色体が破壊されてしまい、効率的な最良個体の探索が阻害される危険がある。そこで、個体集団の中で最も適合度の高い個体は、無条件でそのまま残すというエリート保存選択が提案されている。エリート保存選択を採用すれば、最良の個体は交叉や突然変異により壊されることなく、次世代へと残されることが保障される。ただし、エリート保存選択を採用した場合には、エリート個体が集団に急速に広がり、集団の個体の多様性が失われ、大域的な解の探索が阻害される危険性が高くなる。このためエリート保存選択は通常、他の選択方法と組み合わせて適用される。

2.1.6 交叉

交叉は選択された個体間で染色体を組み換えることで、新しい個体を生成する遺伝的操作である。代表的な交叉方法として、

- 一点交叉
- 多点交叉
- 一様交叉

などが提案されている。

一点交叉

二つの親の染色体上で交叉点をランダムに一箇所選び、交叉点の右側の染色体を交換することで、二つの子個体を生成する交叉方法である。

多点交叉

一点交叉の拡張である多点交叉では、二つの親の染色体上で、交叉点をランダムに複数個選び、交叉点で交互に親の染色体を交換することで、二つの子の染色体を生成する。多点交叉のうち、交叉点が二点のものを二点交叉、三点のものを三点交叉と呼ぶ。

一様交叉

一様交叉では、染色体の遺伝子の数 (n 個) だけランダムに 0,1 を発生させて、 n ビットのマスクパターンを作成する。そして、このマスクパターンで 1 (0) をとる遺伝子には親個体 1 (親個体 2) の遺伝子を受け継ぐ子個体 1 と、逆にマスクパターンで 1 (0) をとる遺伝子には親個体 2 (親個体 1) の遺伝子を受け継ぐ子個体 2 とを生成する交叉方法である。

2.1.7 突然変異

突然変異は、染色体上のある遺伝子座の値を、他の対立遺伝子に置き換える遺伝的操作である。突然変異により交叉だけでは生成されない子の生成が可能となる。

2.2 遺伝的アルゴリズムの最適性

他のメタヒューリスティクスと同様に、遺伝的アルゴリズムは確率的な手法であり、常に大域的最適解が得られる保証は無い。こうした遺伝的アルゴリズムの最適化メカニズムの最適性を理論的に証明しようとする研究が行われており、数々の定理が発表されている。その代表にスキーマ定理がある。

2.2.1 スキーマ定理

スキーマとは、個体評価など行うための数学的な道具である。スキーマは 0,1,* からなる文字列で定義され、* はその遺伝子座の値が 1 でも 0 でも良いことを表す。例えば $H=*11*0$ というスキーマは、01100, 01110

11100, 11110 という 4 つの染色体を表している。この時、この 4 つの染色体はスキーマ H に属する呼ぶ。

スキーマに関しては、オーダーと定義長という 2 つの尺度が定義される。 H 中の*以外の記号の数をスキーマ H のオーダー $o(H)$ と定義する。例えば $o(10*1*) = 3$ となる。また、スキーマを左から見て最初の*以外の文字と、最後の*以外の文字の間の距離を、スキーマ H の定義長 $\delta(H)$ と呼ぶ。例えば $\delta(10*1*) = 3$ となる。

スキーマ定理を導き出すにあたり、以下の単純化した遺伝的アルゴリズムを考える。

選択： 世代 t の個体手段内の各個体 A_i の適合度 $f(A_i)$ を計算し、ルーレット選択により次世代へと生き残させる個体を選択する。

交叉： 個体集団内の個体をランダムにペアリングし、あらかじめ定めた交叉確率 p_c で一点交叉を加え、新たな個体を生成する。

突然変異： 個体の各遺伝子座について、あらかじめ定めた交叉確率 p_m で、遺伝子の値を他の対立遺伝子へと置き換える。

t 世代の個体集団の中で、スキーマ H に属する染色体を持つ個体の数を $m(H, t)$ で表すと、 $m(H, t+1)$ は以下のように計算される。まず選択において $t+1$ 世代の個体集団に含まれる、スキーマ H に属する染色体を持った個体の数を考えると、ルーレット選択により次世代に生き残るスキーマ H に属する染色体を持つ個体 A_i の数は、式 (2.7) で計算される。

$$m(H, t+1) = m(H, t) \frac{f(H, t)}{\bar{F}(t)} \quad (2.7)$$

ここで $f(H, t)$ は t 世代において、スキーマ H に含まれる染色体を持つ個体の平均適合度を表し、 $\bar{F}(t)$ は t 世代の個体集団の平均適合度を表している。

つぎに交叉による個体数の変化を考える。交叉によるスキーマの破壊は、左から見て最初の*以外の記号と、最後の*以外の記号の間に交叉点選ばれた時に起こり得る。そして、このような交叉点選ばれる確率は $p_c \delta(H) / (l-1)$ であり、交叉後に世代 $t+1$ において生き残るスキーマ H の数は、式 (2.8) で計算される。

$$m(H, t+1) \geq m(H, t) \frac{f(H, t)}{\bar{F}(t)} \left(1 - \frac{p_c \delta(H)}{l-1}\right) \quad (2.8)$$

なお、同じスキーマ H に属する染色体を持った個体同士が交叉し、スキーマが壊されないことがあること、また交叉により新たにスキーマ H に属する染色体を持った個体が生成される場合があることから、式 (2.8) は不等号となっている。

最後に突然変異によるスキーマ数の変化を考える。スキーマが突然変異により壊されるのは、スキーマの 0 または 1 の値に突然変異が加えられた場合である。したがって、ある個体が突然変異後も同じスキーマにとどまる確率は $(1 - p_m)^{o(H)}$ となる。一般に遺伝的アルゴリズムの突然変異率は小さいので、この確率は近似的に $1 - p_m \cdot o(H)$ と計算することができ、突然変異後のスキーマ H に属する染色体を持った個体の数は、式 (2.9) で計算される。

$$m(H, t+1) \geq m(H, t) \frac{f(H, t)}{F(t)} [1 - p_c \frac{\delta(H)}{l-1} - o(H)p_m] \quad (2.9)$$

なお、交叉と同様、突然変異により、新たにスキーマ H に属する染色体を持った個体が生成される可能性があるので、式 (2.9) は不等号となる。

式 (2.9) はスキーマ定理と呼ばれ、遺伝的アルゴリズムにおける基本原理となっている。スキーマ定理は、

- 定義長が短い
- オーダーが小さい
- 適合度が平均以上

のスキーマの数が、飛躍的に増大することを表している。このようなスキーマはビルディング・ブロックと呼ばれ、遺伝的アルゴリズムではこのビルディング・ブロックをうまく組み合わせて、最適な探索を実行していると考えられる。これをビルディング・ブロック仮説と呼ぶ。ただし、仮説であり、ある種の問題（だまし問題）では、この仮説がうまく機能しないことが知られている。

2.2.2 NFL 定理

ここでは組合せ最適化問題を対象に、NFL 定理を証明する。 X を問題の実行可能領域、 Y をこの実行可能領域に対応した目的関数値の取り得る値の集合とする (f で目的関数を表す)。 X は非常に広いが有限である

と仮定する（ Y も同様に有限となる）。なお，ここでは組合せ最適化問題を対象にNFL定理を示すが，この結果はデジタルコンピュータ上の最適化問題へと一般化することができる。デジタルコンピュータ上では，実数も32ビットや64ビットの2進数で表現されるので，上記の仮定が成立する。

あるアルゴリズム a が探索した m 個の解を探索順に並べた列を，

$$d_m = \{(d_m^x(1), d_m^y(1)), \dots, (d_m^x(m), d_m^y(m))\} \quad (2.10)$$

で表す。なお，遺伝的アルゴリズムなど過去に評価した解を記憶しないアルゴリズムでは，実行過程において，過去に探索した解を再び評価することがある。しかし，ここでは単純化のため，過去に探索した解はフィルタリングにより d_m から除外する。 $d_m^x(i)$ はアルゴリズムが i 番目に探索した X の値， $d_m^y(i)$ はそれに対応した目的関数値を表している。また $d_m^y = \{d_m^y(1), \dots, d_m^y(m)\}$ で，アルゴリズムが探索した点に対応した目的関数値の数列を表す。

ここでは目的関数 f に対して，アルゴリズム a の操作を m 回繰り返した時の性能を， $P(d_m^y | f, m, a)$ で評価する。 $P(d_m^y | f, m, a)$ は目的関数 f に対してアルゴリズム a の操作を m 回繰り返したときに， d_m^y という数列が得られる条件付確率を表している。この $P(d_m^y | f, m, a)$ の値を基に，他の性能評価も簡単に行える。たとえばアルゴリズム a のパフォーマンスを $\Phi(d_m^y) = \min_i \{d_m^y(i) : i = 1, \dots, m\}$ で評価すれば，目的関数 f の最小値を見つけ出す性能でアルゴリズムを評価することができる。

NFL 定理

任意のアルゴリズムのペア a_1, a_2 に対して，

$$\sum_f P(d_m^y | f, m, a_1) = \sum_f P(d_m^y | f, m, a_2) \quad (2.11)$$

が成り立つ。つまり実現可能な関数全体で評価したアルゴリズムの性能は，アルゴリズムによらず一定となる。

証明

数学的帰納法により， $P(d_m^y | f, m, a)$ が a に依存せず一定であることを証明する。

$m = 1$ の場合を考える。アルゴリズムが探索した点を $d_1 = \{d_1^x, d_1^y\} = \{d_1^x, f(d_1^x)\}$ とすると，

$$\sum_f P(d_1^y | f, m = 1, a) = \sum_f \delta(d_1^y, f(d_1^x)) \quad (2.12)$$

が成り立つ。ここで $\delta(x, y)$ は Kronecker delta 関数であり,

$$\delta(x, y) = \begin{cases} 0 & x \neq y \\ 1 & x = y \end{cases} \quad (2.13)$$

の値をとる。 $f(d_1^x) = d_1^y$ となる関数の総数は $|Y|^{|X|-1}$ 個となるので,

$$\sum_f P(d_1^y | f, m=1, a) = |Y|^{|X|-1} \quad (2.14)$$

となり, $\sum_f P(d_1^y | f, m=1, a)$ は a に独立であることが示せる。

つぎに $\sum_f P(d_m^y | f, m, a)$ が a とは独立であると仮定した場合には, $\sum_f P(d_{m+1}^y | f, m+1, a)$ も a とは独立となることを示す。このためまずは $\sum_f P(d_{m+1}^y | f, m+1, a)$ を, 以下のように変形する。

$$\begin{aligned} & P(d_{m+1}^y | f, m+1, a) \\ &= P(\{d_{m+1}^y(1), \dots, d_{m+1}^y(m)\}, d_{m+1}^y(m+1) | f, m+1, a) \\ &= P(d_m^y, d_{m+1}^y(m+1) | f, m+1, a) \\ &= P(d_{m+1}^y(m+1) | d_m^y, f, m+1, a) P(d_m^y | f, m+1, a) \end{aligned} \quad (2.15)$$

したがって

$$\begin{aligned} & \sum_f P(d_{m+1}^y | f, m+1, a) \\ &= \sum_f P(d_{m+1}^y(m+1) | d_m^y, f, m+1, a) P(d_m^y | f, m+1, a) \end{aligned} \quad (2.16)$$

と表すことができる。ここで $d_{m+1}^y(m+1)$ の値は, x の値と f にのみ依存して決まるので, 上式を x に関して展開すると, 以下の式が得られる。

$$\begin{aligned} & \sum_f P(d_{m+1}^y | f, m+1, a) \\ &= \sum_{f,x} P(d_{m+1}^y(m+1) | f, x) \cdot P(x | d_m^y, f, m+1, a) \cdot P(d_m^y | f, m+1, a) \\ &= \sum_{f,x} \delta(y_{m+1}^y(m+1), f(x)) \cdot P(x | d_m^y, f, m+1, a) \cdot P(d_m^y | f, m+1, a) \end{aligned} \quad (2.17)$$

ここで x の値は a, d_m^x, d_m^y で決まり, f には直接依存していない。そこで $P(x | d_m^y, f, m+1, a)$ の中の f の依存性を排除するように d_m^x を展開す

ると

$$\begin{aligned}
& \sum_f P(d_{m+1}^y | f, m+1, a) \\
&= \sum_{f, x, d_m^x} \delta(d_{m+1}^y(m+1), f(x)) \cdot P(x | d_m, a) \\
&\quad \cdot P(d_m^x | d_m^y, f, m+1, a) \cdot P(d_m^y | f, m+1, a) \quad (2.18)
\end{aligned}$$

を得る。ここで

$$P(x | d_m, a) = \delta(x, a(d_m)) \quad (2.19)$$

$$P(d_m | f, m+1, a) = P(d_m | f, m, a) \quad (2.20)$$

であるので、

$$\begin{aligned}
& \sum_f P(d_{m+1}^y | f, m+1, a) \\
&= \sum_{f, d_m^x} \delta(d_{m+1}^y(m+1), f(a(d_m))) \cdot P(d_m | f, m, a) \quad (2.21)
\end{aligned}$$

となる。左辺の $P(d_m | f, m, a)$ は $x \in d_m^x$ において f に依存するが、一方で $\delta(d_{m+1}^y(m+1), f(a(d_m)))$ は $x \notin d_m^x$ においてのみ f に依存する。したがって

$$\begin{aligned}
& \sum_f P(d_{m+1}^y | f, m+1, a) \\
&= \sum_{d_m^x} \sum_{f(x \in d_m^x)} p(d_m | f, m, a) \sum_{f(x \in d_m^x)} \delta(d_{m+1}^y(m+1), f(a(d_m))) \quad (2.22)
\end{aligned}$$

を得る。 $\sum_{f(x \in d_m^x)} \delta(d_{m+1}^y(m+1), f(a(d_m)))$ の値は、 d_m^x に含まれる点で定義されず、 $(d_{m+1}^x, f(a(d_m)))$ を通過する関数の数と等しく、 $|Y|^{|X|-m-1}$ となる。そこで、

$$\begin{aligned}
\sum_f P(d_{m+1}^y | f, m+1, a) &= |Y|^{|X|-m-1} \sum_{f(x \in d_m), d_m^x} P(d_m | f, m, a) \\
&= 1/|Y| \sum_{f, d_m^x} P(d_m | f, m, a) \\
&= 1/|Y| \sum_f P(d_m^y | f, m, a) \quad (2.23)
\end{aligned}$$

となる。仮定より $P(d_m^y | f, m, a)$ は a とは独立であるので、式 (2.23) より $P(d_{m+1}^y | f, m+1, a)$ も a とは独立となり、NFL 定理が証明される。

2.3 遺伝的アルゴリズムの特徴

遺伝的アルゴリズムの特徴の一つが、柔軟性の高さである。遺伝的アルゴリズムは、どのような目的関数や制約条件を持つ最適化問題にも適用が可能である。極端なケースとして、目的関数値（個体の適合度）が厳密に計算できない問題であっても、個体の優劣さえ付けられれば、ランキング選択を用いることで遺伝的アルゴリズムの適用が可能である。こうした遺伝的アルゴリズムの柔軟性の高さは、電力システム最適化での実用化に非常に適している。最適化手法の実用化を図る場合、開発の途中段階において、新たな制約条件の追加や、モデル変更などの要求が出されることも多い。柔軟性の高い遺伝的アルゴリズムであれば、こうした要求に比較的容易に対応することができる。

また、他の手法と容易に融合が行えることも遺伝的アルゴリズムの特徴である。遺伝的アルゴリズムの探索メカニズムは単純であり、個体の選択・再生方法、個体の選択方法など、遺伝的アルゴリズムの探索メカニズムの中に、他の手法を組み合わせることが容易に行える。

2.4 本論文での遺伝的アルゴリズムの定義

本論文では以降、広義の意味で遺伝的アルゴリズムという用語を用いる。以下に示すメカニズムにより最適解を探索するアルゴリズムを、本論文では遺伝的アルゴリズムと呼ぶ。

- 解に関する情報を個体の染色体にコード化する。
- 染色体にコード化された解の情報を基に、個体の適合度を計算する。
- 適合度を基に次世代へと生き残らせる個体を選択する。
- 交叉と突然変異の遺伝的操作により、新たな個体（探索点）を生成する。

例えば次章で説明する割当て問題を解く遺伝的アルゴリズムでは、染色体にコード化された優先度と、グリーディアルゴリズムにより計算される割当て優先度の組合せから、割当て順序を決定する。このため一般的な遺伝的アルゴリズムとは異なり、遺伝子型と表現型の対応は1対1の関係ではなく、多対1の関係になる。つまり、染色体にコード化された遺

伝子型は異なっても、表現型は同一となる場合がある。本論文ではこうしたアルゴリズムも遺伝的アルゴリズムと呼ぶ。

また、4章で述べる混合整数計画問題を解く遺伝的アルゴリズムでは、解の一部の情報のみを個体の染色体にコード化し、連続緩和法を組み合わせることで、全ての解の値を決定している。このため割当て問題を解く遺伝的アルゴリズムと同様、遺伝子型と表現型の対応は多対1の関係になる。また、染色体のみの情報で全ての解の値が決定するわけではない。本論文では、こうしたアルゴリズムも、遺伝的アルゴリズムと呼ぶ。

さらに、一般的な遺伝的アルゴリズムでは、現世代の個体の適合度の高さに比例して、次世代へと残す個体を選択する。言い換えれば、適合度の低い個体を確率的に淘汰しているが、5章で説明する、確率計画問題を解く遺伝的アルゴリズムでは、単純に現世代の適合度だけでなく、統計的検定手法を組み合わせることで淘汰する個体を決定している。このため現世代の適合度だけで見れば、適合度の低い個体が生き残り、適合度の高い個体が淘汰されることもある。本論文では、こうした適合度の高さに比例して個体選択を行わないアルゴリズムも、遺伝的アルゴリズムと呼ぶ。

第3章

割当て問題のための染色体表現方法：電力用移動無線の周波数割当て問題への適用

3.1 電力システムにおける割当て問題

電力システムの割当て問題の代表に、電力用移動無線システムの周波数割当て問題がある。電力用移動無線システムとは、電力会社が独自に構築・運用している移動無線システムであり、電力の安定供給のために利用されている。たとえば台風災害により電力設備に被害が生じた場合、複数の作業チームで感電事故を防ぎつつ、迅速に復旧作業を行ううえで、電力用移動無線が重要な役割を果たす。

現在、この電力用移動無線システムのデジタル化が電力各社で検討／実施されており、基地局への周波数割当てが大きな課題となっている。近年の移動無線の急速な普及により周波数が不足する状況の下、電力用移動無線システムのデジタル化にあたっては、周波数の効率利用が強く求められている。このためには干渉しない基地局で同一の周波数を再利用し、周波数の使用帯域幅が最小となるように基地局へ周波数を割り当てる必要がある。しかし、こうした周波数割当てを行うことは容易ではない。

こうした基地局に対する周波数割当ての組合せの中から、使用する周波数帯域幅が最小となる割当てを求める問題は、周波数割当て問題（FAP: Frequency Assignment Problem）と呼ばれ、これまでに（1）ニューラルネットワークを用いた手法 [36, 37, 38, 39, 40]、（2）遺伝的アルゴリズム（GA）を用いた手法 [39, 40, 41, 42, 43, 44, 45, 46]、（3）分枝限定法な

どの数理計画法による手法 [47, 48], (4) 制約充足問題の解法を用いた手法 [49, 50] など, 数多くの割当て手法が提案されている。

提案されている割当て手法の中では, 染色体表現として順序表現を用いた遺伝的アルゴリズムが, 特にフィラデルフィア問題と呼ばれるベンチマーク問題で高い性能を発揮している [51, 52]。しかし, 現実の電力用移動無線システムの周波数割当て問題は, フィラデルフィア問題と比べ大規模・複雑な問題となる。周波数を割り当てる基地局数, またお互いが干渉する基地局数が多く, 順序表現を用いた遺伝的アルゴリズムでは, 周波数の帯域幅が最小となる周波数割当てを求めることは難しい。現実の周波数割当て問題で遺伝的アルゴリズムを実用化するには, 大規模問題における解の精度向上が必要となる。

そこで, 大規模な周波数割当て問題を効率的に解く, 染色体の表現方法に特徴を持つ遺伝的アルゴリズムを開発した。開発した遺伝的アルゴリズムでは, 基地局へ周波数を割り当てる優先度を染色体として表現し, この優先度と基地局に割り当てるチャンネル数と, その基地局に割当可能なチャンネル数から計算される優先度の二つの値を基に, 基地局へ周波数を割り当てていく。開発した遺伝的アルゴリズムをフィラデルフィア問題に適用したところ, これまでに提案された手法では求めることができなかった, より帯域幅の狭い割当てが求められた。また, 現実の電力会社の周波数割当て問題に適用したところ, 従来手法と比べ, より使用帯域幅の少ない周波数割当てが求められ, 開発したアルゴリズムによる精度向上の効果が確認された。

3.2 周波数割当て問題の定式化

周波数割当て問題には, チャンネル割当て方式の違いなどにより, いくつかのタイプの問題があるが, 電力用移動無線システムの周波数割当て問題は, このうちの固定チャンネル割当 (FCA) 方式を用いた移動無線システムの最小スパン割当て問題に相当する。FCA 方式とは, 各基地局に対してあらかじめチャンネルを割り当てておき, その基地局のエリア内で生じた通話要求には, その基地局に割り当てられたチャンネルの中から一つを選んで割り当てる方式である。また, 最小スパン割当て問題とは, チャンネル間隔条件などの制約を満たす周波数割当ての中で, 使用する周波数の帯域幅がもっとも小さくなる割当てを求める問題である。本論文では以降, 最小スパン割当て問題を単に周波数割当て問題と呼ぶ。

使用する周波数のうち、チャンネル番号最大のもとの、最小のものとの差を帯域幅と定義すると、最小スパン割当て問題は、以下のように定式化できる。

$$\min \max_{i \in X, l \in d_i} f_{i,l} \quad (3.1)$$

$$\text{Subject to } |f_{i,k} - f_{j,l}| \geq c_{i,j}, i, j \in X, k \in d_i, l \in d_j \quad (3.2)$$

ここで、

X : 基地局の集合

n : 基地局の総数

d_i : 基地局 i で必要となるチャンネル数

F : 利用可能なチャンネルの集合

$f_{i,k}$: 基地局 i に割り当てられる k 番目の周波数。 $f_{i,k} \in F, k \leq d_i$

$c_{i,j}$: 基地局 i と j の間のチャンネル間隔条件

である。なお、チャンネル間隔条件としては、以下の三つの条件が一般的である。

(1) 同一チャンネル割当て条件： 信号対干渉電力比（CIR）が許容値を下回る（干渉する）基地局の組には、同一のチャンネルを割り当てない。つまり、局 i と局 j の組に対して、同一のチャンネルを利用しても CIR の条件を満足するなら $c_{i,j} = 0$ 、そうでなければ $c_{i,j} \neq 0$ とする。

(2) 近傍チャンネル割当て条件： 局 i と局 j で CIR の条件を満たすために必要となるチャンネル間隔条件。局 i と局 j で CIR の条件を満たすためには、 μ 以上のチャンネル間隔が必要なら $c_{i,j} = \mu$ とする。

(3) 同一局でのチャンネル間隔条件： 同一局で利用するチャンネルの間隔条件。前述の近傍チャンネル割当て条件と同じ意味を持つが、それよりも大きな間隔が設定されることが多い。局 i に割り当てるチャンネルには μ' 以上の間隔が必要なら $c_{i,i} = \mu'$ とする。

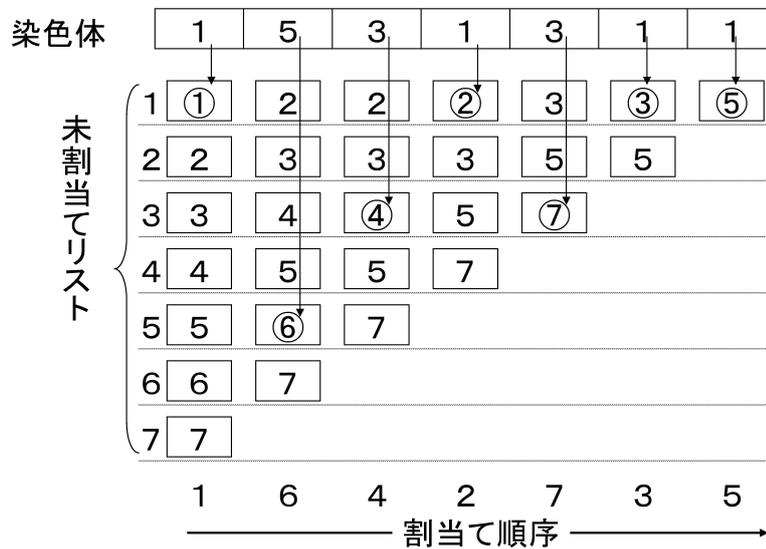


図 3.1: 染色体と基地局への割当て順序

3.3 開発アルゴリズム

これまでに行われた、周波数割当て問題に遺伝的アルゴリズムを適用する研究では、染色体表現として割当て問題で一般的に用いられている、順序表現が採用されることが多い。順序表現での i 番目の遺伝子は、 $i-1$ 番目までの遺伝子の情報に従った割当てが行われた時点において、次に周波数を割り当てる基地局が、未割当てリストの何番目の要素であるかを表している。未割当てリストとは、まだ周波数が割り当てられていない基地局の番号を列挙したものである。例としてチャンネルを割り当てる基地局の総数が 7 の場合における、染色体 $[1, 5, 3, 1, 3, 1, 1]$ に対応する基地局の割当て順序を図 3.1 に示した。なお、図 3.1 の縦に並んだ数値列は、それぞれの割当て時点における未割当てリストを表している。

しかし、現実の電力用移動無線システムの周波数割当て問題では、周波数割当ての対象となる基地局数が非常に多く、染色体に順序表現を用いた遺伝的アルゴリズムでは、帯域幅が最小となる周波数割当てを求めることが難しい。そこで本論文では、発見的手法と組み合わせることで、

効率的に精度の高い割当てを求める遺伝的アルゴリズムを開発した。開発した遺伝的アルゴリズムは、染色体の表現方法に特徴があり、発見的手法により求めた割当て順序の擾乱項として遺伝子の値を利用する。以降、開発アルゴリズムが用いる二つの発見的手法について説明し、次に開発した遺伝的アルゴリズムについて説明する。

3.3.1 発見的手法：再帰的割当て

効率的に使用帯域幅が小さい基地局への周波数割当てを求める発見的手法として、より単純な問題における最適な周波数の割当て順序を求め、この順序に従い再帰的に周波数割当てを繰り返すことで、元の問題の割当て順序を決める、以下の発見的手法を遺伝的アルゴリズムと組み合わせる。

- (1) **初期化** 基地局の中で必要となるチャネル数が最小の基地局を m とする。
- (2) **必要チャネル数の標準化** $d'_i = \lceil d_i/d_m \rceil, i \in X$ とする。ここで $\lceil x \rceil$ は x 以上の最小の整数を表すものとする。
- (3) **割当て順序の探索** 各基地局の必要チャネル数を d'_i とした問題において、使用周波数の帯域幅を最小とする基地局への周波数の割当て順序 P を求める。
- (4) **周波数の割当て** P の割当て順序に従い、各基地局に周波数を割り当てることを d_m 回繰り返す

例として、全ての基地局の必要チャネル数が、基地局 i によらず一定の 3 チャネルである場合を考える。この場合、再帰的割当てのヒューリスティックでは、すべての基地局の必要チャネル数を 1 とした、より単純な問題で最適な割当て順序を求め、この順序に従った割当てを再帰的に 3 回繰り返すことで、元の問題での割当てを決定する (図 3.2)。

もちろん、単純な問題における最適な割当て順序を再帰的に繰り返すことで、元の問題において使用帯域幅が最小となる割当てが求まる保証はない。しかし、多くの問題においては、この再帰的な割当て順序により、使用帯域幅が最小となる割当てが求まると期待される。また、各基地局の必要チャネル数を少なくすることで、単純な問題での解空間はより小さなものとなり、解の探索効率が向上する。

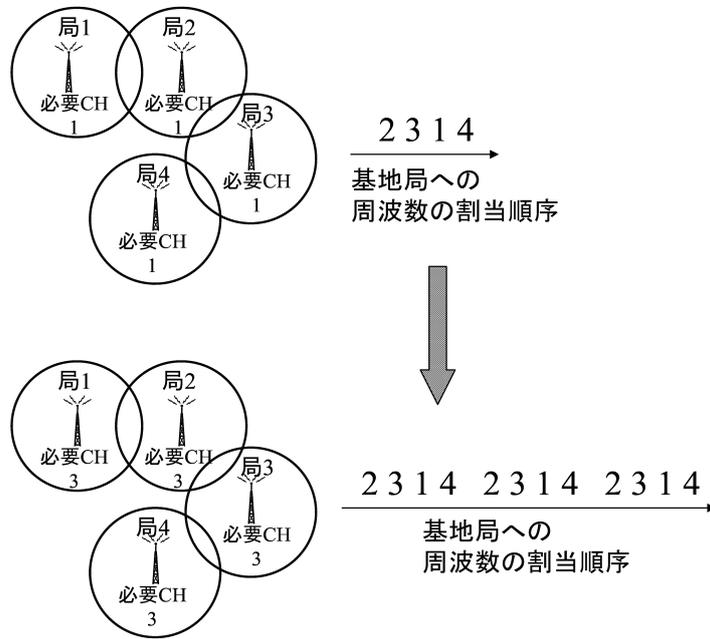


図 3.2: 再帰的割当て

3.3.2 発見的手法：AAFヒューリスティック

使用する周波数の帯域幅が小さい，基地局への周波数割当て順序を求めるヒューリスティックとして，横尾らはAAF（Average Available Frequency）を提案している [50]。これは基地局への周波数割当てのある時点において，基地局 i に既に割り当てられたチャンネル数を n_i^a ，基地局 i に割当て可能なチャンネル数を n_i^f とした時， $AAF_i = n_i^f / (d_i - n_i^a)$ の値がもっとも小さい基地局 i から，周波数を割り当てていく方法である。開発アルゴリズムでも， AAF_i の値を基地局へ周波数を割り当てていく順序を決定するのに利用する。

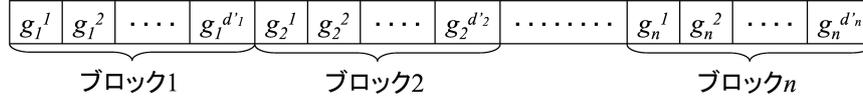


図 3.3: 染色体表現

3.3.3 提案アルゴリズム

上述の二つの発見的手法を用いることで、ある程度までは使用周波数の帯域幅が小さい割当てが求まるものの、実用上十分な精度を持った割当てを得ることはできない。そこで、上述の発見的手法と遺伝的アルゴリズムの組合せにより、最適な周波数割当てを効率的に求めるアルゴリズムを開発した。開発アルゴリズムでは、使用帯域幅を最小とする割当て順序の中には、再帰的なものが存在することを仮定し、再帰的な割当て順序の中で使用周波数の帯域幅が最小となる割当て順序を探索する。また、割当て順序の決定においては、AAFにより計算される優先度を考慮する。

開発した遺伝的アルゴリズムでは、基地局の割当て順序の優先度を染色体として表現する。染色体は n ブロックから構成され、ブロック i は d_i 個の遺伝子により構成される。ここで i 番目のブロックの k 番目の遺伝子を g_i^k とすると、これは基地局 i に割り当てる k 番目のチャンネルの優先度を表すものとする (図 3.3)。各遺伝子は一定範囲 $[w_l, w_h]$ の実数を、その値としてとる。

提案アルゴリズムによる割当て手順を示すと以下のようになる。

(1) 初期化：以下の手順により初期集団を生成する。

- (1.1) 必要チャンネル数が最小となる基地局 m を見つけ、 $d_i' = \lceil d_i/d_m \rceil, i \in X$ を計算する。
- (1.2) 基地局 i の必要チャンネル数を d_i' としたうえで、図 3.3 の染色体を持つ個体をランダムに n_p 個作り初期集団とする。

(2) 適合度の計算：現在の集団の各個体に対して以下の操作を行う。

- (2.1) $AAF_i \times g_i^k$ の値を計算し、この値がもっとも小さい基地局 i に利用可能な最小のチャンネル番号の周波数を割り当てる。

- (2.2) (2.1) の割当て結果に基づき、各基地局で利用可能な周波数のチャンネル情報と AAF_i の計算に用いる n_i^f, n_i^g の値を更新する。
- (2.3) すべての基地局に d_i' 個の周波数が割り当てられていれば (2.4) へ。そうでなければ (2.1) を繰り返す。
- (2.4) (2.3) までの操作により得られた割当て順序を A とすると、この A に従った割当て順序に従い、各基地局に周波数を割り当てることを d_m 回繰り返す。
- (2.5) (2.4) で得られた割当ての使用帯域幅を計算し、これより個体の適合度を計算する。
- (3) 個体の選択・再生：以下の手順により適合度の高い個体を選択し、これに遺伝的操作を加えることで新たな個体を生成する。
- (3.1) 現在の集団の中から、適合度の高い個体をルーレット選択により n_g 個選び出す。
- (3.2) 選び出した個体からランダムにペアを作り、それぞれのペアに対し、確率 r_c で交叉の遺伝的操作を加える。
- (3.3) 選び出した各個体の遺伝子に対して、確率 r_m で突然変異の遺伝的操作を加える。
- (4) 新世代の生成：(3) で新たに作成した個体の集団と、元の集団を合わせ、その中で適合度の高い個体 n_p 個を選び、これを次世代の集団とする。
- (5) 終了判定：終了条件が満たされたなら、それまでに得られた最良の個体を解として採用する。そうでなければ、(2) に戻る。

3.4 適用結果

開発した遺伝的アルゴリズムを、周波数割当て問題のベンチマーク問題と、実際の九州電力における周波数割当て問題に適用した結果を示す。

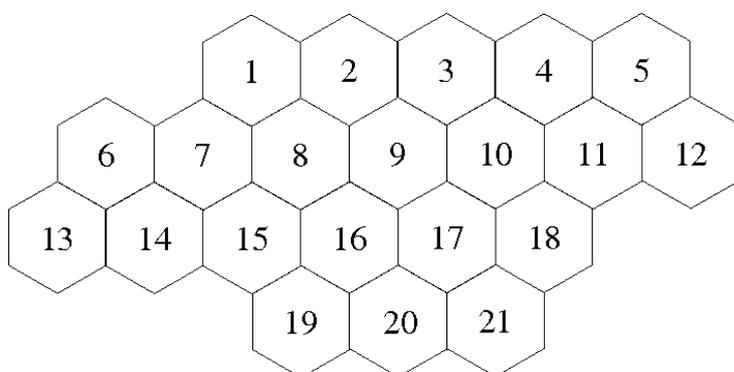


図 3.4: フィラデルフィア問題

3.4.1 ベンチマーク問題への適用

数多くの研究で参照されている、周波数割当て問題の代表的なベンチマーク問題にフィラデルフィア問題がある。フィラデルフィア問題とは、フィラデルフィア近郊における実際の周波数割当て問題から派生して作られた問題群であり、図 3.4 に示した配置の 21 のセルに対し周波数を割り当てる問題である。フィラデルフィア問題には、要求チャンネル数や、チャンネル間隔条件が異なる、さまざまなバリエーションの問題が存在する。

Funabiki らが検討したフィラデルフィア問題 [37] に対しては、文献 [51] の染色体に順序表現を用いた遺伝的アルゴリズムにより、常に最適解を求めることができる。しかし、この順序表現を用いたアルゴリズムでは、最適解への収束性が低い問題のバリエーションが存在する [53]。そこで、ここではこれらの問題に対して開発したアルゴリズムを適用し、その有効性を検証した。

ここで検討した五つのフィラデルフィア問題 (P1~P5) のパラメータは表 3.1 のとおりである。表 3.1 で d_k は k チャンネルの間隔を空けた周波数が利用できるようになる最小のセル間の距離を表している。ここでセルの距離とは、隣接するセルの中心間の距離を 1 とした場合の距離を表している。また $M_i = \{m_1^i, m_2^i, \dots, m_n^i\}$ の要素 m_j^i はセル j に割り当てる要求チャンネル数を表す。 N は全基地局の要求チャンネル数の合計を表している。

フィラデルフィア問題に開発アルゴリズムを適用するにあたり、各個

表 3.1: フィラデルフィア問題のパラメータ

問題	d_0	d_1	d_2	d_3	d_4	d_5	必要チャンネル	N
P1	4	2	1	1	1	0	M_1	470
P2	3	2	1	1	1	0	M_1	470
P3	4	2	1	1	1	0	M_2	420
P4	3	2	1	1	1	0	M_2	420
P5	4	2	1	1	1	0	M_3	481

$$M_1 = \{5, 5, 5, 8, 12, 25, 30, 25, 30, 40, 40, 45, 20, 30, 25, 15, 15, 30, 20, 20, 25\}$$

$$M_2 = \{20, 20\}$$

$$M_3 = \{8, 25, 8, 8, 8, 15, 18, 52, 77, 28, 13, 15, 31, 15, 36, 57, 28, 8, 10, 13, 8\}$$

体の適応度 E を以下のように計算した。

$$E = w \cdot (\max_{i \in X} f_{ik} - L) + n_u \quad (3.3)$$

ここで L は各フィラデルフィア問題の必要帯域幅の下界値を表し, n_u は

$$n_u = \sum_{i \in X, f_{ik} > L} (f_{ik} - L) \quad (3.4)$$

と計算される。 w はあらかじめ決めた定数である。

また、以下の遺伝的アルゴリズムのパラメータ設定を行った。各基地局の割当て優先度を $[1.0, 10.0]$ の実数として染色体にコード化した。また、一世代の個体数を 400 ($n_p=400$) とし、ここからルーレット選択により 320 個の個体を選び出し ($n_g=320$)、これに交叉率 0.5 ($r_c=0.5$) の一様交叉と、突然変異率 0.02 ($r_m=0.02$) の一様変異の遺伝的操作を加えた。適合度の重み w は 100 とした。終了条件としては、適合度が 0 (下界値と等しい帯域幅) の割当てが見つかるか、世代数が 1000 となった時点でアルゴリズムを終了した。

開発した遺伝的アルゴリズムをフィラデルフィア問題に適用したところ、表 3.2 に示す結果が得られた。表 3.2 にはそれぞれのフィラデルフィア問題の使用帯域幅の下界値と、二つの発見的手法のみを適用した場合の帯域幅、それに文献 [53] の順序表現を用いた遺伝的アルゴリズムと、本論文で開発した遺伝的アルゴリズムを適用した結果を示してある。なお、遺伝的アルゴリズムの適用結果としては、それぞれのアルゴリズムを 50 回実行した時、得られた最小の帯域幅を「最小」の列に、また下界値と等しい帯域幅が得られる割合を「収束率」の列に記した。

表 3.2: フィラデルフィア問題への適用結果

問題	下界値	発見的手法	文献 [53] の手法		開発手法	
			最小	収束率	最小	収束率
P1	257	286	257	14	257	70
P2	252	274	252	68	252	20
P3	289	309	289	32	289	100
P4	178	207	194	0	179	0
P5	524	568	524	64	524	70

表 3.2 の結果を見ると問題 P2 を除き、開発アルゴリズムを用いることで、最適解への収束確率が向上している。また、問題 P4 については、開発アルゴリズムでも下界値を達成できなかったものの、使用する帯域幅を大幅に削減する割当てを求めることができた。なお、この問題 P4 に関しては、文献 [53] 以外の手法を用いても、これまでに 194 以下の帯域幅の割当ては見つかっていない。特に問題 P4 に関して、開発アルゴリズムが高い性能を発揮したのは、問題 P4 はすべての基地局の必要チャネル数が 20 と一定の問題であることから、再帰的割当ての発見的手法が有効に機能したものと考えられる。

3.4.2 九州電力における周波数割当て問題への適用

次に開発アルゴリズムを九州電力における周波数割当て問題へと適用し、文献 [52] で提案したアルゴリズムとの比較を行った。なお、本節では九州電力での周波数割当て問題の概略は説明するが、問題の詳細については文献 [52] を参照されたい。

九州電力における周波数割当て問題は、二つの使用帯域幅の最小化を目的とすることと、一般的な問題のチャネル間隔条件に加え、同一の伝送装置に割り当てる周波数のチャネル間隔を一定値以下とする条件を考慮するという点で、一般的な周波数割当て問題とは異なる。一般的な周波数割当て問題では、全基地局で使用する周波数の帯域幅の最小化のみを目的とするが、九州電力における周波数割当て問題では、これに加えて他電力の基地局との干渉地域で利用する周波数を最小化することが目的とされる。つまり九州電力での周波数割当て問題では、以下の二つの異なる帯域幅をともに最小とする周波数割当てを求めることが目的となる。

- (1) 利用周波数帯域幅の最小化：全ての基地局で利用する周波数の帯域幅を最小化する。つまり $\max_{i \in X} f_{i,k}$ が最小となるように周波数を割り当てる。
- (2) 干渉地域利用周波数帯域幅の最小化：他電力会社との干渉地域で利用する周波数の帯域幅を最小化する。 $C \subset X$ を干渉地域に含まれる基地局の集合とした時、 $\max_{i \in C} f_{i,k}$ が最小となるように周波数を割り当てる。

また、一般的な周波数割当て問題で考慮されるチャンネル間隔条件は、チャンネルを割り当てる間隔を一定値以上とするものであるが、九州電力における周波数割当て問題ではこれに加えて、同一の伝送装置に割り当てられるチャンネル間隔が一定値 (μ') 以下となるように周波数を割り当てる。つまり f_{ik} と f_{il} を局 i の同一の伝送装置上に割り当てられる周波数とすると、 $|f_{ik} - f_{il}| \leq \mu'$ となるように周波数を割り当てる。

九州電力における割当て問題では、全基地局での利用帯域幅を最小とすることに加えて、他電力との干渉地域で利用する周波数の帯域幅を最小化することが求められる。そこで、ここでは個体の適合度 E を

$$E = w \cdot (f_s - \max_{i \in X} f_{i,k}) + w \cdot (f_c - \max_{i \in C} f_{i,k}) + n_u \quad (3.5)$$

で計算する。ここで f_s は基地局全体で必要となる周波数の帯域幅の下界値、 f_c は干渉地域に位置する基地局で必要となる帯域幅の下界値を表すものとする。また、ここでは n_u を

$$n_u = \sum_{i \in X, f_{i,k} > f_s} (f_{i,k} - f_s) + \sum_{i \in C, f_{i,k} > f_c} (f_{i,k} - f_c) \quad (3.6)$$

で計算する。なお、下界値の値は、対象とする周波数割当て問題をグラフの彩色問題として定式化し、このグラフの最大クリークを求めることで求められる。

九州電力の周波数割当て問題へ開発アルゴリズムを適用するにあたり、以下のパラメータ設定を行った。各基地局の割当て優先度は $[1.0, 50.0]$ の実数とした。また、一世代の個体数を 75 ($n_p=75$) とし、ここからルーレット選択により 45 個の個体を選び ($n_g=45$)、これに交叉率 1.0 ($r_c=1.0$) の一様交叉と、突然変異率 0.03 ($r_m=0.03$) の一様変異の遺伝的操作を加えた。適合度の重み w は 1 とした。終了条件としては、適合度が 0 (下界値と等しい帯域幅) の割当てが見つかるか、世代数が 700 となった時点でアルゴリズムを終了した。

表 3.3: 九州電力の周波数割当て問題への適用結果

手法	収束率 (%)	平均計算時間 (秒)
文献 [52] の手法	48	116
提案手法	90	15

開発アルゴリズムと文献 [52] のアルゴリズムを、九州電力の周波数割当て問題に適用した結果を表 3.3 に示す。なお、表 3.3 で収束率とは、全基地局と干渉地域にある基地局の両方の使用帯域幅が、ともに下界値と一致する割当てが求められる確率を表している。また、平均計算時間とは、全基地局と干渉地域にある基地局の両方の使用帯域幅がともに下界値と一致する、厳密な最適割当てを得るまでに必要とされる平均の CPU 時間を表している。なお、この CPU 時間は Alpha 21264 (700MH) での CPU 時間である。

表 3.3 から明らかなように、開発アルゴリズムでは最適な割当てへの収束率が向上している。しかも、文献 [52] の手法では、安定した最適割当ての探索には個体数が 200 必要であったものが、開発アルゴリズムでは発見的手法を組み合わせることで収束性能が向上し、75 の個体数で安定した解の探索が可能となっている。この結果、発見的手法の計算に余分な時間が必要となるものの、最適解を求めるのに必要となる平均計算時間を大幅に短縮することができた。

第4章

混合整数計画問題のための の個体評価方法：発電機起 動停止計画問題への適用

4.1 電力システムにおける混合整数計画問題

電力システムにおける混合整数計画問題の代表に、発電機の起動停止計画問題がある。発電機の起動停止計画問題とは、比較的短い(1日~1週間)期間を対象に、与えられた電力需要の予測値に対して、需給バランス、運転予備力、発電機出力の上下限、最小運転/停止時間などの制約条件を満たし、発電コストが最小となる発電機の起動停止状態と出力とを同時決定する問題である。日々の発電コストのわずかな違いも長期的には膨大な額となるため、供給コスト削減のためには経済的な起動停止計画の立案が重要となる。

発電機の起動停止計画問題は、混合整数計画問題のうち、目的や制約条件が非線形関数で記述される、非線形混合整数計画問題に分類される。非線形混合整数計画問題は解くことが非常に難しく、実用的な時間内に厳密にコスト最小となる起動停止計画を求めることは困難である。このため、より発電コストの少ない起動停止計画を、実用的な計算時間で求める計画手法が数多く提案されている。こうした計画手法の代表に、(1) ラグランジュ緩和法を用いた計画手法 [54, 55, 56, 57, 58, 59], (2) 進化型計算手法を用いた計画手法 [60, 61, 62, 63, 64, 65, 66, 67, 68], (3) 複数の手法の組合せによる計画手法 [69, 57, 70] がある。

このうちラグランジュ緩和法を用いる手法については、得られる起動停止計画の経済性の問題が指摘されている [69]。ラグランジュ緩和法では、需給バランス制約や運転予備力制約などの、系統全体に関わる制約条件を緩和したうえで、問題を発電機毎の部分問題に分割して最適な計画を

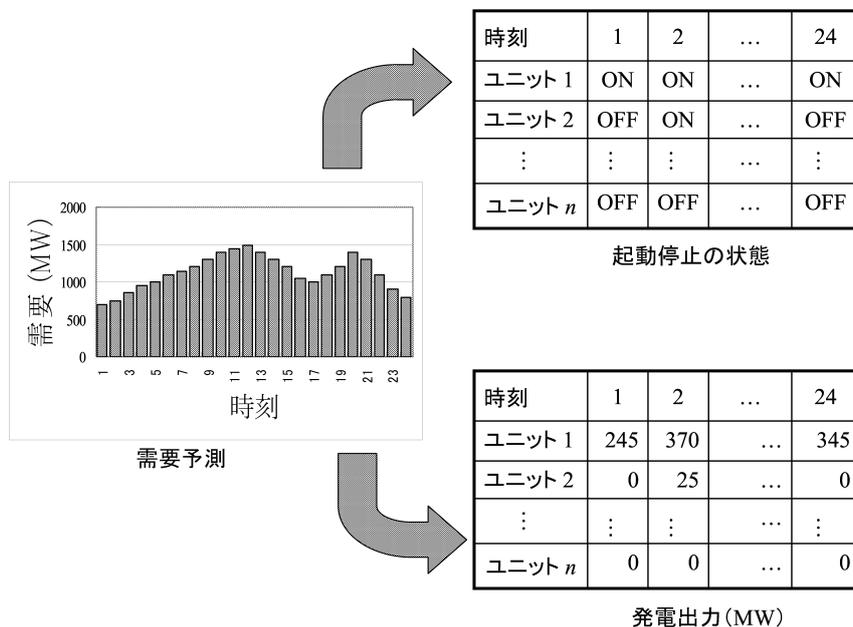


図 4.1: 発電機の起動停止計画問題

求める。このため燃料費特性が同一の発電機が複数存在した場合、これらの発電機のすべてが同時に起動/停止する計画しか求めることができず、経済性の劣る起動停止計画となる可能性が高い [57]。特に現実の発電所では、一つの発電所内に燃料費特性が同一の発電機が複数設置される場合が多く、燃料費特性が同一の発電機が同時に起動/停止する計画しか求められないラグランジュ緩和法の特徴は、実用上の問題となる。また、問題を発電機毎の部分問題に分割して解くため、系統の潮流制約や、同一の発電所内の同時起動・停止台数の制約などを考慮した計画を行うことができないことも実上の問題となる [71]。

遺伝的アルゴリズムや進化型プログラミングなどの、進化型計算手法を用いる計画手法については、様々な制約条件を考慮した起動停止計画が比較的容易に行える柔軟性の高さがメリットとして挙げられる。しかし、局所探索能力の弱さが問題点として指摘されており [72]、計画対象となる発電機数が多くなった場合には、ラグランジュ緩和法などと比べ、経済性が劣る起動停止計画しか得られなくなる傾向がある。

複数の手法の組合せによる計画手法としては、ラグランジュ緩和法と遺伝的アルゴリズムの組合せによる手法 [69]、優先リスト法と経験則の

組合せによる手法 [70], ラグランジュ緩和法と経験則の組合せによる手法 [57] が提案されており, 特に単純なベンチマーク問題において高い性能を発揮している。

このうちラグランジュ緩和法と遺伝的アルゴリズムの組合せによる手法は, 遺伝的アルゴリズムを用いてラグランジュ緩和法の探索パラメータの最適化を行いつつ, 効率的に経済的な起動停止計画を求める手法である。優先リスト法と経験則の組合せによる手法は, あらかじめ発電機の性能を基に決めたリストの順番に発電機を起動させていく優先リスト法により決定した計画に, 経験則による修正を加えていく手法である。ラグランジュ緩和法と経験則の組合せによる手法は, ラグランジュ緩和法により求めた計画に, 経験則による修正を行っていく手法である。

これらの手法の問題点としては, 計画手法の柔軟性の低さが挙げられる。ラグランジュ緩和法を基本とする手法 [69] では, ラグランジュ緩和法で扱うことのできない, 潮流制約などの制約を考慮することはできない。また, ベンチマーク問題に合わせて作られた経験則を用いる手法 [57, 70] では, ベンチマーク問題で考慮されていない制約が加わった問題の最適な起動停止計画を, 効率的に求めることが難しい。

そこで, 燃料費特性が同一の発電機を含め, 数多くの発電機が存在する問題においても経済的な起動停止計画が求められ, また制約条件の追加などにも柔軟に対処できる実用的な計画手法を目指して, 非線形計画法と遺伝的アルゴリズムとを組み合わせ, 新たなアルゴリズムを開発した。

近年の非線形最適化パッケージの進展により, 整数変数が含まれない問題であれば, 比較的規模が大きな非線形最適化問題も, 実用的な計算時間で解くことが可能となっている [73, 74, 75]。そこで, 遺伝的アルゴリズムによる探索過程において, 非線形最適化問題の解を活用することで, 遺伝的アルゴリズムの弱点とされている局所探索能力を補い, 大規模な起動停止計画問題を効率的に解くアルゴリズムを開発した。

開発した遺伝的アルゴリズムでは, 計画対象となる発電機の起動停止スケジュールのうち, 一部の発電機の起動/停止を固定し, 残りの発電機の起動/停止は未固定とした, 部分スケジュールを個体の染色体に記録する。そして, 未固定の発電機の起動停止については, 整数変数を連続と仮定した非線形最適化問題 (連続緩和問題) の解を基に決定する。したがって最悪でも, 全ての整数変数を連続と仮定した問題を解き, 得られた最適解の値の大きいものから順に, 制約の範囲内で発電機を起動さ

せていく，連続緩和法を用いた場合と同程度の発電コストの計画が求まることが保障される。

基本性能を検証するため，多くの論文 [69, 61, 62, 57, 70] が性能評価に用いている，起動停止計画問題のベンチマーク問題に開発した遺伝的アルゴリズムを適用した。その結果，これまでの遺伝的アルゴリズムをベースとする計画手法では，大規模問題での収束性能が問題点として指摘されていたが，開発手法では発電機数が多いベンチマーク問題において，これまでに提案された手法の中で，もっとも経済的な起動停止計画を求めることができた。

また，現実の起動停止計画を行ううえで重要となる，潮流制約を考慮した，より複雑な起動停止計画問題に開発した遺伝的アルゴリズムを適用したところ，潮流制約を考慮した問題においても，経済的な起動停止計画を効率良く求めることができ，開発した遺伝的アルゴリズムの有効性が確認できた。

4.2 発電機起動停止計画問題の定式化

4.2.1 記号一覧

発電機の起動停止計画の定式化を行うに当たり，以下の記号を定義する。

i : 発電ユニットの添え字, $i = 1, 2, \dots, n$

t : 時刻の添え字, $t = 1, 2, \dots, m$

u_i^t : ユニット i の時刻 t における運転状態 (0: 停止, 1: 稼動)

p_i^t : ユニット i の時刻 t での発電出力 (MW)

D^t : 時刻 t における電力需要 (MW)

R^t : 時刻 t で必要な運転予備力 (MW)

p_i^{\max} : ユニット i の最大出力 (MW)

p_i^{\min} : ユニット i の最小出力 (MW)

T_i^{on} : ユニット i の最小起動時間

T_i^{off} : ユニット i の最小停止時間

考慮する制約条件が異なる，さまざまなタイプの起動停止計画問題が研究されているが，多くの論文で性能評価に用いられているベンチマーク問題の発電機の起動停止問題は，以下のように定式化される。

$$\min \sum_{t=1}^m \sum_{i=1}^n \{b_i(p_i^t) \cdot u_i^t + S_i(x_i^{\text{off}}(t)) \cdot u_i^t(1 - u_i^{t-1})\} \quad (4.1)$$

$$\text{subject to} \quad \sum_{i=1}^n p_i^t - D^t = 0, t = 1, 2, \dots, m \quad (4.2)$$

$$\sum_{i=1}^n u_i^t \cdot p_i^{\max} - D^t - R^t \geq 0, t = 1, 2, \dots, m \quad (4.3)$$

$$u_i^t \cdot p_i^{\min} \leq p_i^t \leq u_i^t \cdot p_i^{\max}, \\ i = 1, 2, \dots, n, t = 1, 2, \dots, m \quad (4.4)$$

$$(x_i^{\text{on}}(t-1) - T_i^{\text{on}})(u_i^{t-1} - u_i^t) \geq 0, \\ i = 1, 2, \dots, n, t = 1, 2, \dots, m \quad (4.5)$$

$$(x_i^{\text{off}}(t-1) - T_i^{\text{off}})(u_i^t - u_i^{t-1}) \geq 0, \\ i = 1, 2, \dots, n, t = 1, 2, \dots, m \quad (4.6)$$

$$u_i^t \in \{0, 1\}, i = 1, 2, \dots, n, t = 1, 2, \dots, m \quad (4.7)$$

$$p_i^t \geq 0, i = 1, 2, \dots, n, t = 1, 2, \dots, m \quad (4.8)$$

$$x_i^{\text{on}}(t), x_i^{\text{off}}(t) \in \{1, 2, \dots\}, t = 0, 1, \dots, m \quad (4.9)$$

ここで $b_i(p_i^t)$ はユニット i の時刻 t での燃料コストを表しており，発電出力 p_i^t の非線形関数として計算される。また $S_i(x_i^{\text{off}}(t))$ はユニット i の起動コストを表す。発電機は停止状態から起動する場合にコストが発生する。ボイラーの温度が下がる程，起動にはより多くのエネルギーが必要となることから，起動コストはユニット i が時刻 t までに連続して停止していた時間を表す $x_i^{\text{off}}(t)$ の単調増加関数として計算される。また，同様にユニット i が時刻 t までに連続して稼動していた時間を $x_i^{\text{on}}(t)$ で表す。このように，その発電機の現時点の状態だけでなく，過去の時点の状態に依存して決まる発電機の起動コストの存在により，発電機の起動停止計画問題はより複雑な問題となっている。

式 (4.2) は需給バランス制約と呼ばれ，時刻 t における電力需要と発電力がバランスしていることを保証する。式 (4.3) は運転予備力制約を表している。天候の急変などによる需要変動に対応できるよう，稼動している発電機のすべてを最大出力で稼動した場合，電力需要に供給予備力を加えた電力量が発電可能なことを保証する。式 (4.4) は発電機出力が決

められた範囲内となることを保証する制約であり、発電機出力の上下制限制約と呼ばれる。また、急激な温度変化は発電機のボイラーの寿命を短くするため、一旦発電機を起動（停止）した場合、ある決められた期間、発電機を停止（起動）することはしない。この制約が式 (4.5) (式 (4.6)) に対応しており、最小運転（停止）時間制約と呼ばれる。制約条件のうち最小起動（停止）時間に関する制約も、発電機の起動コストと同様に、発電機の現時点の状態だけでなく、過去の状態に依存する制約となっており、こうした動的な制約条件の存在が、起動停止計画問題をより複雑なものとしている。

また、この問題の決定変数の数は、発電機の起動停止状態を表す 0-1 変数が $n \times m$ 個、発電機の出力を表す連続変数の数が $n \times m$ 個になる。例えば日本の大きな電力会社では 70 機以上の発電機を有しており、こうした電力会社の翌日 24 時間の起動計画を求める問題は、 $70 \times 24 = 1,680$ 以上の 0-1 変数と連続変数を持つ、非常に大規模な問題となる。

4.3 アルゴリズム

本節では、開発した起動停止計画問題を解くアルゴリズムについて説明する。詳細は後述するが、開発したアルゴリズムは一般的な遺伝的アルゴリズムと同様に、以下の手順で、よりコストの少ない発電機の起動停止計画を探索する。

ステップ 1（初期集団の生成） 発電機の起動停止スケジュールを染色体に記録した個体を、ランダムにあらかじめ決めた数だけ生成する。

ステップ 2（個体評価） 染色体にコード化された起動スケジュールに従った場合の発電コストを基に、個体の適合度を計算する。

ステップ 3（個体の選択・再生） 個体の適合度に比例した確率で、集団から m 個の個体ペアを選び出す。次にこの個体ペアに交叉・突然変異の遺伝的操作を加え、新たな $2m$ 個の個体を生成する。そして、元の個体を含め、適合度の低い $2m$ 個の個体を淘汰する。

ステップ 4（終了判定） 世代の進行により個体集団の適合度の改善が一定値以下となる、あるいは世代数があらかじめ決めた数に達するなど、終了条件を満たしたらアルゴリズムを終了する。終了条件が満たされていないならば、ステップ 2 からの操作を繰り返す。

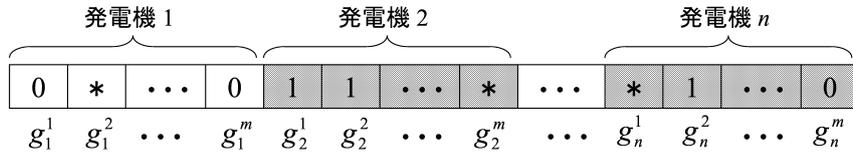


図 4.2: 染色体表現

以下、アルゴリズムの各ステップの詳細を説明する。

4.3.1 初期集団の生成

発電機の起動停止に関する情報を、図 4.2 に示す形式で記録した染色体を持つ個体を、あらかじめ決めた数だけ生成する。染色体の遺伝子 g_i^t は発電機 i の時刻 t における起動停止の状態を表しており、 g_i^t は 0, 1, * のいずれかの値をとる。 $g_i^t = 1$ は発電機 i が時刻 t で起動していることを、 $g_i^t = 0$ は発電機 i が時刻 t で停止していることを表す。 $g_i^t = *$ は発電機 i の時刻 t での起動/停止について、未固定であることを表す。

つまり個体の染色体には、計画対象となっている発電機の起動停止スケジュールのうち、一部の発電機の起動/停止が固定され、残りは未固定のまま残された起動スケジュール（以降、部分スケジュールと呼ぶ）が記録されている。開発したアルゴリズムでは、遺伝的アルゴリズムの進化メカニズムを用いて、この部分スケジュールをより経済的なものへと進化させていく。

なお、発電機の起動停止計画問題は制約条件が多く、アルゴリズムの初期集団を生成するのに、一般的な遺伝的アルゴリズムと同様の手順で、ランダムに遺伝子に 0, 1, * の値を割り当てたのでは、制約条件のすべてを満たす個体が生成される確率は非常に小さい。そして、制約条件を満たさない初期個体集団から進化を開始した場合には、探索効率が非常に悪化する。そこで、ここでは

- 式 (4.7) の起動・停止を表す変数 u_i^t の整数条件を緩和 (u_i^t は 0 以上 1 以下の実数)
- 起動コストを停止時間によらず一定と仮定 (最小停止時間での起動コストを仮定)

- 式 (4.5) , (4.23) の最小運転/停止時間制約を省略

という単純化を行った発電機の起動停止問題を, 非線形最適化手法を用いて解き, この問題の最適解の情報を基に初期個体を生成する。

初期個体生成のために解く単純化した問題は以下のようなになる。

$$\min \sum_{t=1}^m \sum_{i=1}^n \{b_i(p_i^t) \cdot u_i^t + S_i(T_i^{\text{off}}) \cdot u_i^t(1 - u_i^{t-1})\} \quad (4.10)$$

$$\text{subject to} \quad \sum_{i=1}^n p_i^t - D^t = 0, t = 1, 2, \dots, m \quad (4.11)$$

$$\sum_{i=1}^n u_i^t \cdot p_i^{\max} - D^t - R^t \geq 0, t = 1, 2, \dots, m \quad (4.12)$$

$$u_i^t \cdot p_i^{\min} \leq p_i^t \leq u_i^t \cdot p_i^{\max}, \\ i = 1, 2, \dots, n, t = 1, 2, \dots, m \quad (4.13)$$

$$0 \leq u_i^t \leq 1, i = 1, 2, \dots, n, t = 1, 2, \dots, m \quad (4.14)$$

$$p_i^t \geq 0, i = 1, 2, \dots, n, t = 1, 2, \dots, m \quad (4.15)$$

そして, この単純化した問題の最適解 \underline{u}_i^t と, $[0,1]$ の乱数 r の値から g_i^t の初期値を以下の式で決定する。

$$g_i^t = \begin{cases} 1 & \text{if } \underline{u}_i^t \geq 0.9 \text{ and } r < 1/2 \\ * & \text{if } \underline{u}_i^t \geq 0.9 \text{ and } r \geq 1/2 \\ 0 & \text{if } \underline{u}_i^t \leq 0.1 \text{ and } r < 1/2 \\ * & \text{if } \underline{u}_i^t \leq 0.1 \text{ and } r \geq 1/2 \\ 1 & \text{if } 0.1 < \underline{u}_i^t < 0.9 \text{ and } r < 1/3 \\ 0 & \text{if } 0.1 < \underline{u}_i^t < 0.9 \text{ and } 1/3 \leq r < 2/3 \\ * & \text{if } 0.1 < \underline{u}_i^t < 0.9 \text{ and } 2/3 \leq r \end{cases} \quad (4.16)$$

式 (4.16) は, 単純化した問題の最適解 \underline{u}_i^t が 1 (0) に近ければ, 式 (4.1) – 式 (4.8) の元の問題の最適解が 0 (1) の値をとる可能性は低いと考えられるというアイデアに基づいている。後述する数値実験で確認を行ったところ, この方法で初期個体を生成することで, ランダムに個体を生成した場合と比べ, 制約条件を満たす個体を生成することができた。

4.3.2 個体評価

ステップ 2 では, 染色体に記録されている情報を基に, 以下の手順で個体を評価する。

ステップ 2.1：染色体情報の修正 染色体に記録された部分スケジュールで固定されている発電機の起動／停止のうち，最小運転／停止時間制約を満たさない部分があれば，これを修正する。また，前後の時間帯の起動停止の状況から，起動／停止が確定する未固定部分があれば，この起動停止を固定する。

ステップ 2.2：全発電機の起動停止の決定 染色体の部分スケジュールでは起動停止が固定されていない発電機の起動／停止を，変数の整数条件を緩和した非線形最適化問題の解を基に決定する。

ステップ 2.3：最適発電出力の決定 決定した全発電機の起動停止スケジュールの下での発電機の最適出力を，非線形最適化問題を解いて求め，得られた燃料コストと起動コストから発電コストを計算する。

ステップ 2.4：適合度の計算 計算された発電コストを基に，各個体の適合度を計算する。

以下，それぞれの手順の詳細を説明する。

染色体情報の修正

染色体に記録された部分スケジュールを調べ，式 (4.5)，式 (4.6) の最小運転／停止時間の制約を満たさない部分があれば，制約を満たすように遺伝子の値を変更する。また，対応する遺伝子が*となっている変数のうち，最小運転／停止時間制約から一意に0,1の値が決定する変数があれば，これを固定する。

以下では最小運転／停止時間が5時間の発電機*i*を例に，この染色体情報の修正方法を説明する。染色体の修正は部分スケジュールにおける各発電機の起動停止を時間の早い順に調べ，以下の4パターンのいずれかが生じていた場合に対応した修正を行う。

パターン1 染色体が図 4.3 (a) のようになっていた場合，*となる遺伝子が0の値をとったとしても最小停止時間制約を満たさないので，図 4.3 (a) でグレーに塗られた遺伝子は1の値をとることはできない。よって図 4.3 (b) のように修正される。同様に最小運転時間制約を満たさない0の値をとる遺伝子があった場合には，この遺伝子の値を1へと修正する。

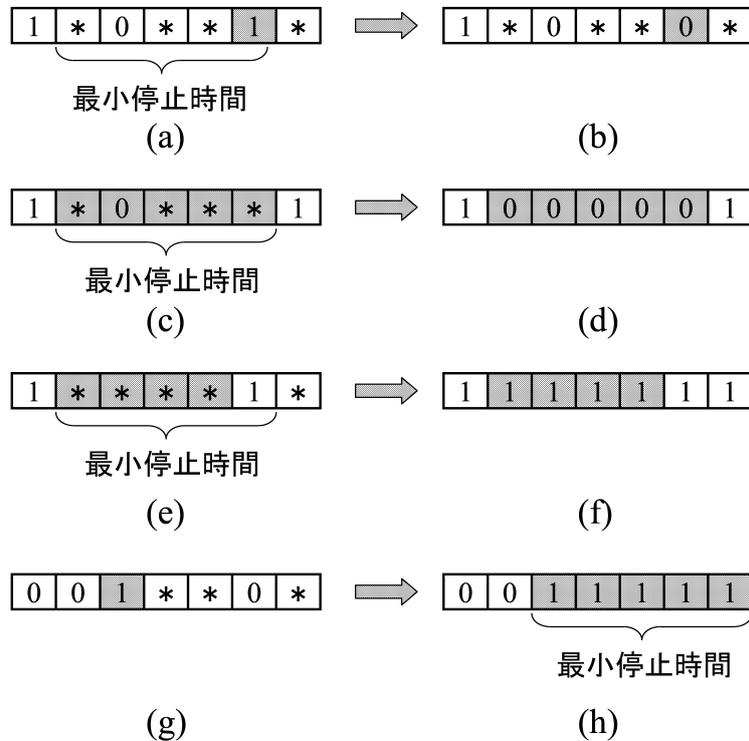


図 4.3: 染色体情報の修正

パターン 2 発電機 i に関する染色体において、図 4.3 (c) のように、1 の値をとる二つの遺伝子の間に、最小停止時間に等しい数の 0 または * の値を取る遺伝子が挟まれ、この中に一つ以上の 0 の値をとる遺伝子が含まれていた場合 (図 4.3 (c) のグレーに塗られた遺伝子)、最小運転時間制約を満たすためには、1 に挟まれた遺伝子はすべて 0 でなければならない。よって遺伝子は図 4.3 (d) のように修正される。同様に 0 の値をとる二つの遺伝子の間に、最小運転時間に等しい数の 1 または * の値をとる遺伝子が挟まれ、この中に一つ以上の 1 の値をとる遺伝子が含まれていた場合には、0 に挟まれた遺伝子のすべてを 1 に確定させる。

パターン 3 図 4.3 (e) のように、発電機 i に関する染色体において、1 の値をとる遺伝子に挟まれる * の数が最小停止時間 (5 時間) より小さい場合、最小停止時間制約より、この * をとる遺伝子が 0 の値をとること (発電機を停止すること) はできない。よって染色体は図 4.3 (f) のよう

に修正される。同様に0の値をとる遺伝子に挟まれる*の数が最小運転時間より小さい場合、0に挟まれた*の遺伝子の値を0に確定する。

パターン4 たとえば図4.3 (g) のように、0をとる遺伝子の直後に1の遺伝子がある場合、最小運転時間制約を満たすためには、この1をとる遺伝子の後には、少なくとも最小運転時間に等しい数の1の値をとる遺伝子が必要ではない。したがって、遺伝子を図4.3 (h) のように修正する。同様に1をとる遺伝子の直後に0の遺伝子が現れた場合には、最小停止時間の数だけ0が続くように値を確定する。

全発電機の起動停止状態の決定

以下の変数の整数制約と最小運転/停止時間制約を緩和した問題を、非線形最適化法を用いて解く。なお、起動コストとしては、染色体の情報から確定する部分については確定した値を、そうでない部分は取り得る起動コストの最小値を採用する。

$$\min \sum_{t=1}^m \sum_{i=1}^n \{b_i(p_i^t) \cdot u_i^t + S_i(x_i^{\text{off}}(t)) \cdot u_i^t(1 - u_i^{t-1})\} \quad (4.17)$$

$$\text{subject to} \quad \sum_{i=1}^n p_i^t - D^t = 0, t = 1, 2, \dots, m \quad (4.18)$$

$$\sum_{i=1}^n u_i^t \cdot p_i^{\max} - D^t - R^t \geq 0, t = 1, 2, \dots, m \quad (4.19)$$

$$u_i^t \cdot p_i^{\min} \leq p_i^t \leq u_i^t \cdot p_i^{\max}, \\ i = 1, 2, \dots, n, t = 1, 2, \dots, m \quad (4.20)$$

$$u_i^t = 0, (i, t) \in \{(i, t) \mid g_i^t = 0\} \quad (4.21)$$

$$u_i^t = 1, (i, t) \in \{(i, t) \mid g_i^t = 1\} \quad (4.22)$$

$$0 \leq u_i^t \leq 1, (i, t) \in \{(i, t) \mid g_i^t = *\} \quad (4.23)$$

$$p_i^t \geq 0, i = 1, 2, \dots, n, t = 1, 2, \dots, m \quad (4.24)$$

ここで $x_i^{\text{off}}(t)$ は、すべての自由変数が1を取った場合の $x_i^{\text{off}}(t)$ の値($x_i^{\text{off}}(t)$ の最小値)を表す。

上記の問題の最適解(u_i^t)を基に最小運転/停止時間制約を満たす、発電機の起動停止計画(\tilde{u}_i^t)を作成する。 \tilde{u}_i^t を作成する手順の概略は以下のようになる。

- 時刻 $t-1$ で起動しており，時刻 t における連続運転時間が最小運転時間に満たない発電機はすべて起動させる。
- u_i^t の値が 1 に近く，最小停止時間制約を満たす発電機 i は起動させる。
- 起動させた発電機の最大出力 p_i^{\max} の合計が，式 (4.3) の運転予備力制約を満たしていなければ，運転予備力制約を満たすまで，運転予備力制約への影響の大きい ($u_i^{*t} \cdot p_i^{\max}$ の値が大きい) 発電機 i から起動させていく。
- 起動させる発電機 i が時刻 $t-1$ で停止しており，最小停止時間制約を満たしていなければ，時刻 $t-1$ 以前の連続停止していた期間のすべてで発電機 i を起動させ，最小停止時間制約を満たすようにする。また，発電機 i を起動させた $t-1$ までの時刻で，運転予備力制約を満たす範囲で停止できる発電機があれば，最大出力時の発電単価の高い発電機から停止させる。

以上の手順を式で表すと以下ようになる。

Step 2.2.1: $t = 1$, $P^{\text{tot}} = 0$, $\tilde{u}_i^t = 0$, $i = 1, 2, \dots, n$, $t = 1, 2, \dots, m$ とする。

Step 2.2.2: 時刻 $t-1$ で稼働しているユニット i の連続稼働時間が最小運転時間に達していなければ， $\tilde{u}_i^t = 1$, $P^{\text{tot}} = P^{\text{tot}} + p_i^{\max}$ とする。

Step 2.2.3: $u_i^t \geq 0.99$ の条件を満たすユニット i が最小停止時間制約を満たしているならば， $\tilde{u}_i^t = 1$, $P^{\text{tot}} = P^{\text{tot}} + p_i^{\max}$ とする。

Step 2.2.4: $P^{\text{tot}} \geq D^t + R^t$ なら Step 2.2.9 へ。

Step 2.2.5: $\tilde{u}_{i^*}^t = 1$, $P^{\text{tot}} = P^{\text{tot}} + p_{i^*}^{\max}$ とする。ここで

$$i^* = \arg \max_{i \in \{j | \tilde{u}_j^t = 0\}} \{u_i^t \cdot p_i^{\max}\} \quad (4.25)$$

である。

Step 2.2.6: ユニット i^* が時刻 $t-1$ で停止しており，最小停止時間制約を満たしていなければ，ユニット i を時刻 $\max(0, t - x_{i^*}^{\text{off}}(t-1))$ から $t-1$ の期間稼働させる。つまり $\tilde{u}_{i^*}^{t-j} = 1$ for $j = 1, 2, \dots, x_{i^*}^{\text{off}}(t-1)$ とする。

Step 2.2.7: 時刻 $t - x_{i^*}^{\text{off}}(t-1)$ から $t-1$ の期間で, Step2.2.5 で稼働させた発電機の代わりに停止できる発電機があれば停止する。

Step 2.2.8: $P^{\text{tot}} \geq D^t + R^t$ なら Step 2.2.9 へ, そうでなければ Step 2.2.5 からの操作を繰り返す。

Step 2.2.9: $t < m$ なら $t = t+1$, $P^{\text{tot}} = 0$ として Step 2.2.2 へ。そうでなければ終了。

最適な発電出力の決定

確定した発電機の起動停止計画 \tilde{u}_i^t に対する, 各ユニットの最適な発電出力 p_i^t を, 以下の非線形計画問題を解いて求める。

$$\min \sum_{t=1}^m \sum_{i=1}^n b_i(p_i^t) \quad (4.26)$$

$$\text{subject to } \sum_{i=1}^n p_i^t - D^t = 0, t = 1, 2, \dots, m \quad (4.27)$$

$$\tilde{u}_i^t \cdot p_i^{\min} \leq p_i^t \leq \tilde{u}_i^t \cdot p_i^{\max}, \\ i = 1, 2, \dots, n, t = 1, 2, \dots, m \quad (4.28)$$

$$p_i^t \geq 0, i = 1, 2, \dots, n, t = 1, 2, \dots, m \quad (4.29)$$

上記の問題で最低運転出力制約を満たす実行可能解が存在しない場合, 優先度の低いものから順に発電機を停止する。同様に需給バランス制約や供給予備力制約を満たす実行可能解が存在しなければ, 優先度の順に発電機を起動する。

そして, 上記の問題を解いて決定した発電機出力における燃料コストと, 起動停止スケジュール \tilde{u}_i^t における起動コストの合計として計算される発電コストを基に, 個体の適合度を計算する。

4.3.3 個体の選択・再生

個体集団の中から適合度の高さに比例した確率で, 一定数の個体を選び出す。選び出した個体の中でランダムにペアを作り, これにあらかじめ決めた一定の確率に従って, 交叉の遺伝的操作を加える。交叉とはペアとなった個体 (親) の染色体の一部を交換し, 新しい個体 (子) を生

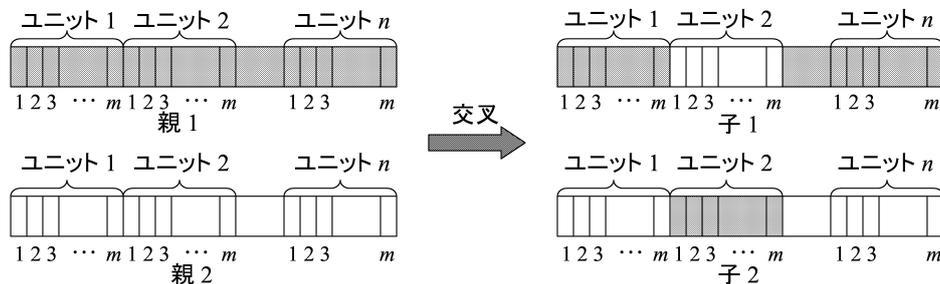


図 4.4: 交叉の遺伝的操作



図 4.5: 突然変異の遺伝的操作

成する遺伝的操作である。遺伝的操作としては図4.4に示すように、ペアとなった個体の染色体に記録された起動停止の情報を、発電機単位でランダムに入れ替える操作を加えた。

次に、選び出した個体集合に突然変異の遺伝的操作を加える。ただし、一般的な突然変異の操作により、1つの遺伝子の値を変化させただけでは、最小運転/停止時間制約のため、発電機の起動停止に変化が生じない可能性が高い。そこで後述する数値実験では、ランダムに選んだ1つの遺伝子の値を変化させるとともに、その遺伝子の前後 $\max\{T_i^{\text{on}}, T_i^{\text{off}}\}$ 個の遺伝子の値を*に変化させる突然変異の操作（図4.5）を加え、突然変異により起動停止に変化が生じ易くした。

4.3.4 終了判定

終了条件を満たしているかをチェックする。もし、終了条件を満たしていなければ、個体評価からの操作を繰り返す。なお、後述する数値実験では終了判定の基準として、あらかじめ決めた世代数に達しているかをチェックし、あらかじめ決めた世代数に達した時点でアルゴリズムを終了

し、個体集団の中でもっとも適合度の高い個体の染色体に記録された起動停止計画を、最良な起動停止計画とした。

4.3.5 開発アルゴリズムの特徴

上述のように、開発したアルゴリズムでは連続緩和法を遺伝的アルゴリズムと組み合わせるが、4.4節で示す数値実験のように、発電機の燃料コストは一般に2次項の係数が小さい2次関数として計算される。このため連続緩和問題の最適解では、0または1に近い値を取る変数が多くなり、上述のアルゴリズムにより効率的に最適な発電機起動停止計画の探索が行えると期待される。

また、発電機の起動停止計画は、需要に合わせて一定期間毎（例えば1時間毎）に見直しが行われるが、実用化の面からは、過去に求めた計画と比べ、発電機の起動・停止のタイミングが大きく異なる計画が求まるのが理想である。こうした点でも、連続緩和法の最適解を基に初期集団を生成する開発アルゴリズムでは、単純に遺伝的アルゴリズムを適用した場合と比べ、発電機の起動・停止のタイミングが大きく異なる計画が求まる確率が小さく抑えられる。

4.4 数値実験

4.4.1 ベンチマーク問題

まずは数多くの文献 [69, 61, 62, 57, 70] が性能評価に用いている、発電機の起動停止計画問題のベンチマーク問題に開発したアルゴリズムを適用し、その基本性能を検証した。

ベンチマーク問題は、翌日 24 時間の各時刻の発電機の起動停止を決定する問題である。ベンチマーク問題では、ベースとなる 10 機の発電機の性能に関するデータが表 4.1 のように与えられている。

表 4.1 で、 p_i^{\max} と p_i^{\min} はそれぞれ、発電機 i の最大出力と最低出力を表している。 a_{i0} 、 a_{i1} 、 a_{i2} はそれぞれ、発電機 i の燃料コスト関数 $b_i(\cdot)$ の係数であり、発電出力が p の場合の燃料コストは、

$$b_i(p) = a_{i2} \cdot p^2 + a_{i1} \cdot p + a_{i0} \quad (4.30)$$

と計算される。

表 4.1: 発電機のデータ

発電機	1	2	3	4	5
p_i^{\max} (MW)	455	455	130	130	162
p_i^{\min} (MW)	150	150	20	20	25
a_{i0} (\$/h)	1000	970	700	680	450
a_{i1} (\$/MWh)	16.19	17.26	16.60	16.50	19.70
a_{i2} (\$/MW ² -h)	0.00048	0.00031	0.002	0.00211	0.00398
T_i^{on} (h)	8	8	5	5	6
T_i^{off} (h)	8	8	5	5	6
SC_i^{h} (\$)	4500	5000	550	560	900
SC_i^{c} (\$)	9000	10000	1100	1120	1800
H_i^{cs} (h)	5	5	4	4	4
τ_i (h)	8	8	-5	-5	-6

発電機	6	7	8	9	10
p_i^{\max}	80	85	55	55	55
p_i^{\min} (MW)	20	25	10	10	10
a_{i0} (\$/h)	370	480	660	665	670
a_{i1} (\$/MWh)	22.26	27.74	25.92	27.27	27.79
a_{i2} (\$/MW ² -h)	0.00712	0.00079	0.00413	0.00222	0.00173
T_i^{on} (h)	3	3	1	1	1
T_i^{off} (h)	3	3	1	1	1
SC_i^{h} (\$)	170	260	30	30	30
SC_i^{c} (\$)	340	520	60	60	60
H_i^{cs} (h)	2	2	0	0	0
τ_i (h)	-3	-3	-1	-1	-1

T_i^{on} , T_i^{off} は発電機 i の最小運転時間と、最小停止時間を表す。 SC_i^{h} , SC_i^{c} は起動コストに関するパラメータであり、ベンチマーク問題では、起動コスト $S_i(\cdot)$ は発電機の停止時間 x により 2 段階に変化する、式 (4.31) で計算する。

$$S_i(x) = \begin{cases} SC_i^{\text{h}} & x \leq T_i^{\text{on}} + H_i^{\text{cs}} \\ SC_i^{\text{c}} & x > T_i^{\text{on}} + H_i^{\text{cs}} \end{cases} \quad (4.31)$$

τ_i は発電機 i の初期状態を表しており、この値が正ならば初期状態で発電機 i は起動しており、連続運転時間が τ_i であることを表している。この値が負ならば初期状態で発電機 i は停止しており、連続停止時間が $-\tau_i$

表 4.2: 電力需要

時間	D^t (MW)						
1	700	7	1,150	13	1,400	19	1,200
2	750	8	1,200	14	1,300	20	1,400
3	850	9	1,300	15	1,200	21	1,300
4	950	10	1,400	16	1,050	22	1,100
5	1,000	11	1,450	17	1,000	23	900
6	1,100	12	1,500	18	1,100	24	800

であることを表す。またベースとなる 24 時間の電力需要が表 4.2 のように与えられる。

ベンチマーク問題では、このベースとなる 10 機の発電機と電力需要を基に、発電機数が 20 機, 40 機, 60 機, 80 機, 100 機の起動停止計画問題を作る。たとえば発電機数が 40 機のベンチマーク問題では、ベースとなる 10 機の発電機が 4 セットあり、電力需要が 4 倍であるものとして、発電機の起動停止を計画する。つまり 40 機のベンチマーク問題では、同一性能を持った 4 台の発電機が 10 組存在することとなる。

4.4.2 遺伝的アルゴリズムのパラメータ設定

ベンチマーク問題へ適用するにあたり、開発した遺伝的アルゴリズムのパラメータを以下のように設定した。初期個体数を 20 とし、ここから適合度の高さに比例した確率で 18 個体を選び出し、これに確率 0.8 で交叉、確率 $0.03 / (\text{発電機数} \times 24)$ で突然変異の遺伝的操作を加えた。終了条件としては、個体集団の進化が 100 世代に達した時点でアルゴリズムを終了し、個体集団の中でもっとも適合度の高い個体の染色体に記録された起動停止計画を最良な起動停止計画として採用した。

4.4.3 提案計画手法の性能評価

発電機数を 10, 20, 40, 60, 80, 100 とした 6 ケースのベンチマーク問題に対して、提案する計画手法を適用し、これまでに発表されている計画手法と比較した。比較対象としては、ベンチマーク問題で高い性能を発揮している、優先リスト法と経験則の組合せによる手法 (PL) [70], ラグランジュ緩和法 (LR) [57], 進化型計算手法 (EP) [61], ラグランジュ緩

表 4.3: 発電コストの比較

発電機数	PL	LR	EP	LRGA
			最小	平均
10	563,977	565,508	564,551	564,800
20	1,124,369	1,126,720	1,125,494	1,122,622
40	2,246,508	2,249,790	2,249,093	2,242,178
60	3,366,210	3,371,188	3,371,611	3,371,079
80	4,489,322	4,494,487	4,498,479	4,501,844
100	5,608,440	5,615,893	5,623,885	5,613,127
	ELR	開発アルゴリズム		
		最小	平均	最大
10	563,977	563,977	563,977	563,977
20	1,123,297	1,123,297	1,123,548	1,123,834
40	2,244,237	2,243,380	2,244,062	2,245,267
60	3,363,491	3,361,251	3,363,210	3,364,673
80	4,485,633	4,482,301	4,484,090	4,485,363
100	5,605,678	5,602,567	5,604,239	5,606,757

表中の下線はもっとも経済的な起動停止計画を表す。

和法と遺伝的アルゴリズムの組合せによる手法 (LRGA) [69], ラグランジュ緩和法と経験則の組合せによる手法 (ELR) [57] の5つを選択した。各文献に記載されている, それぞれの手法で求められた起動停止計画の発電コストと, 開発した計画手法により得られた起動停止計画の発電コストを比較すると表 4.3 のようになる。

なお, 開発したアルゴリズムは一般的な遺伝的アルゴリズムと同様に, 確率的な探索方法であり, 常に同じ起動停止計画が得られる保証は無い。そこで, 表 4.3 には, 開発した遺伝的アルゴリズムを 20 回実行して得られた, 起動停止計画の発電コストの最小値, 平均値, 最大値を記載した。また, 開発した計画手法と同じく, 確率的な探索方法である EP については, 文献に記載されている, アルゴリズムを 20 回実行して得られた起動停止計画の発電コストの最小値を, 同じく確率的な探索方法である LRGA については, 20 回実行して得られた起動停止計画の発電コストの平均値を記載した。

表 4.3 を見ると, 開発した遺伝的アルゴリズムにより, 6 ケース中 4 ケースで, もっとも経済的な起動停止計画が求められた。得られた起動停止計画の発電コストの平均で比較しても, 6 ケース中 4 ケースで開発した

表 4.4: 計算時間の比較 (秒)

	PL	LR	EP
計算機	Pentium4 1.5GHz	Pentium4 1.6GHz	HP C160
10	0.72	3.2	100
20	2.97	12.0	340
40	11.9	34	1176
60	23.0	67	2267
80	44.4	111	3584
100	64.5	167	6120
	LRGA	ELR	開発手法
計算機	PC486DX2-66	Pentium4 1.6GHz	AMD Opteron 2.6GHz
10	518	4.0	16
20	1147	16	46
40	2165	52	140
60	2414	113	286
80	3383	209	429
100	4045	345	811

アルゴリズムにより求めた起動停止計画がもっとも経済的となっている。また、特に発電機数が多い問題ほど、開発したアルゴリズムにより求めた起動停止計画のコスト削減額が大きく、発電機数が多い起動停止計画問題での、開発したアルゴリズムの有効性が確認できる。

また、使用している計算機が異なるため、直接計算時間の比較は行えないが、開発した遺伝的アルゴリズムにより起動停止計画を求めるのに必要となる計算時間を、文献より得られる各手法の計算時間と比較すると表 4.4 のようになる。なお、表 4.4 の計算時間のうち、PL は Pentium4 1.5GHz の計算機を用いた結果、LR と ELR は Pentium4 1.6GHz の計算機を用いた結果、EP は HP C160 ワークステーションを用いて 20 回実行した時の平均時間、LRGA は PC486DX2-66 の計算機を用いた結果である。開発した計画手法の計算時間については、AMD Opteron 2.6GHz の計算機を用いて 20 回実行した際の計算時間の平均を示した。開発したアルゴリズムの計算時間は、ラグランジュ緩和法を基にした LR、ELR や、優先リスト法を用いた PL より長くなっている。しかし、他の進化型手法 (EP, LRGA) と比べると計算時間は短い。また、発電機の起動停止計画は、需要の変化に合わせて 1 時間程度毎に計画の見直しが行われるが、発

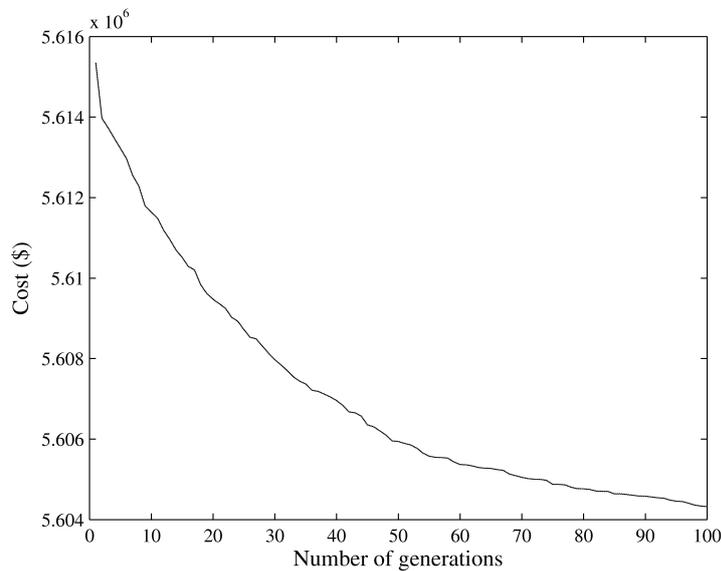


図 4.6: 最良個体の平均適合度の変化

電機数 100 の問題でも開発アルゴリズムの計算時間は 15 分以内となっており、所有する発電機の数も最大でも 80 機程度の日本の電力会社の起動停止計画に適用するには、実用上問題の無い計算時間となっている。

発電機数 100 のベンチマーク問題を開発アルゴリズムを用いて 20 回解いた際の、各世代の最良個体の平均発電コストの変化を図示すると、図 4.6 のようになる。図 4.6 からは、開発アルゴリズムが初期段階で、発電コストの低い個体の生成に成功していることが分かる。開発アルゴリズムでは初期個体の段階で既に、進化型計算手法の中でもっとも性能の高い EP で 50 個体を 5,000 世帯まで進化させた場合よりも、発電コストの低い個体が得られている。これは開発アルゴリズムでは、緩和問題の最適解を基に初期個体を生成することで、初期の段階から適合度の高い個体が得られるためだと考えられる。

また、発電機数 10 機と 20 機のベンチマーク問題における、開発アルゴリズムにより得られた起動停止計画を図示すると、それぞれ図 4.7、図 4.8 のようになる。図 4.7 と図 4.8 を比較すると、単純に 10 機のベンチマーク問題の起動停止計画を複製しただけでは、20 機の問題の最適な起動停止計画が得られないことが分かる。また、発電機数 80 のベンチマーク問題における、同一性能の発電機の起動停止計画を図示すると図 4.9 のよ

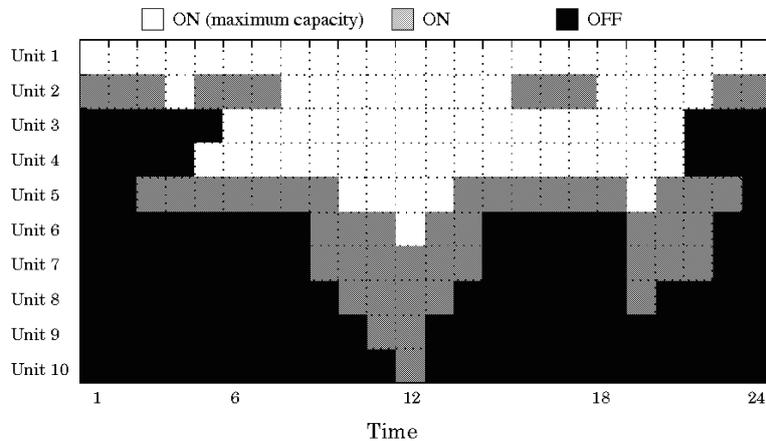


図 4.7: 開発アルゴリズムにより求めた最適な起動停止計画（発電機数 10）

うになる。図 4.9 からは、最適な起動停止計画において、同一性能の発電機が同時に起動停止していないことが分かる。

4.4.4 潮流制約の追加

前述のベンチマーク問題への適用結果から、開発したアルゴリズムの基本性能の高さが確認できた。そこで、現実の発電機の起動停止計画を行う際に考慮する必要があるが、LRGA や ELR などの手法では考慮することが難しい、潮流制約を追加した、より複雑な起動停止問題に開発したアルゴリズムを適用した。

発電機の性能などはベンチマーク問題と同様とし、発電機の設置場所を東地域と西地域の 2 つに分け、発電機 1～発電機 5 が東地域、発電機 6～発電機 10 が西地域に属するものとした。なお、この設定では、東地域に発電コストの安い発電機が多く存在することになる。そして、東地域と西地域の潮流制約として、東地域に属している発電機の発電量と、西地域に属している発電機の発電量の差を一定以下に抑える、

$$\left| \sum_{i=1}^5 p_i^t - \sum_{i=6}^{10} p_i^t \right| \leq 1000 \quad (4.32)$$

の制約を追加した。

開発したアルゴリズムを用いて、この潮流制約を考慮した起動停止計画問題を 5 回解いたところ、得られる起動停止計画のコストの平均は 573,524

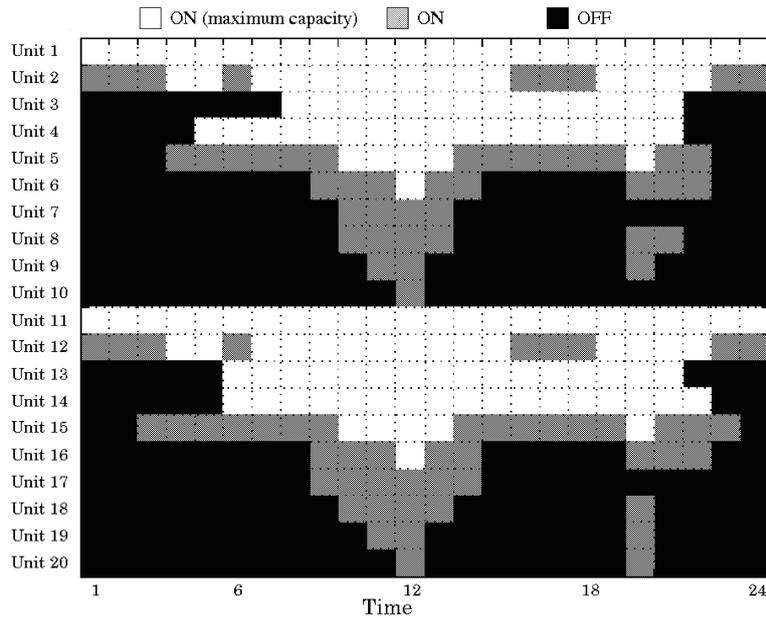


図 4.8: 開発アルゴリズムにより求めた最適な起動停止計画（発電機数 20）

であり，もっとも経済的な起動停止計画のコストは573,084となった。また，計画を得るのに必要な計算時間は，制約を考慮しない場合と同程度の計算時間であった。

開発したアルゴリズムにより求めた，潮流制約を考慮した場合の10機の発電機の各時刻の発電量は表 4.5 のようになる。表 4.6 の潮流制約を考慮しなかった場合の発電量と比べ，潮流制約を考慮した場合には，ピーク時に発電コストの高い西地域の発電機が起動していることが分かる。このように潮流制約を考慮した問題においても，開発したアルゴリズムを用いて経済的な起動停止計画を求めることができ，他の手法と比べて，制約条件の追加にも柔軟に対処できる手法であることが確認できた。

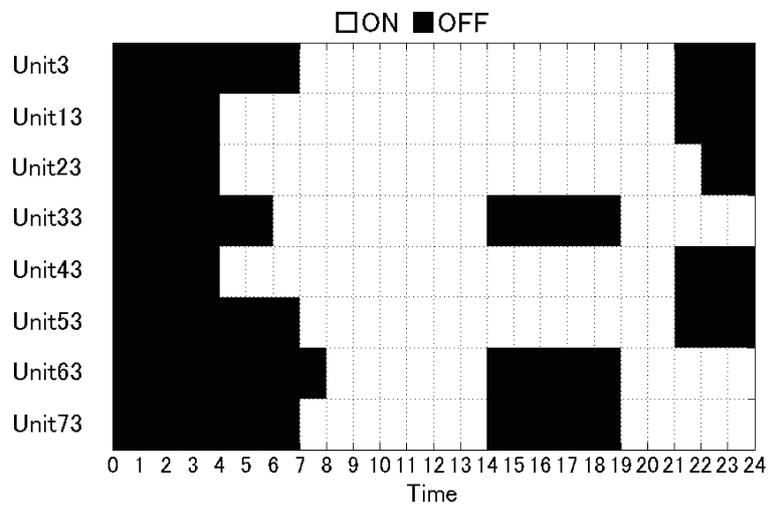


図 4.9: 発電機数 80 のベンチマーク問題における同一性能の発電機の起動停止計画

表 4.5: 連係線制約を考慮した場合の発電量 (MW)

時刻	1	2	3	4	5	6	7	8	9	10	11	12
発電機 1	455	455	455	455	455	455	455	455	455	455	455	455
発電機 2	245	295	370	455	390	440	455	455	410	455	455	455
発電機 3	0	0	0	0	0	0	0	0	130	130	130	130
発電機 4	0	0	0	0	130	130	130	130	130	130	130	130
発電機 5	0	0	25	40	25	25	35	60	25	30	55	80
発電機 6	0	0	0	0	0	50	75	75	80	80	80	80
発電機 7	0	0	0	0	0	0	0	25	70	65	35	50
発電機 8	0	0	0	0	0	0	0	0	0	55	55	55
発電機 9	0	0	0	0	0	0	0	0	0	0	55	55
発電機 10	0	0	0	0	0	0	0	0	0	0	0	10
東地域計	700	750	850	950	1000	1050	1075	1100	1150	1200	1225	1250
西地域計	0	0	0	0	0	50	75	100	150	200	225	250

時刻	13	14	15	16	17	18	19	20	21	22	23	24
発電機 1	455	455	455	455	455	455	455	455	455	455	455	455
発電機 2	455	410	455	415	370	440	455	455	410	440	315	215
発電機 3	130	130	0	0	0	0	0	130	130	130	130	130
発電機 4	130	130	130	130	130	130	130	130	130	0	0	0
発電機 5	30	25	60	25	25	25	60	30	25	25	0	0
発電機 6	80	80	75	25	20	50	75	80	80	0	0	0
発電機 7	65	70	25	0	0	0	25	65	70	50	0	0
発電機 8	55	0	0	0	0	0	0	55	0	0	0	0
発電機 9	0	0	0	0	0	0	0	0	0	0	0	0
発電機 10	0	0	0	0	0	0	0	0	0	0	0	0
東地域計	1200	1150	1100	1025	980	1050	1100	1200	1150	1050	900	800
西地域計	200	150	100	25	20	50	100	200	150	50	0	0

表 4.6: 連係線制約を考慮しない場合の発電量 (MW)

時刻	1	2	3	4	5	6	7	8	9	10	11	12
発電機 1	455	455	455	455	455	455	455	455	455	455	455	455
発電機 2	245	295	370	455	390	360	410	455	455	455	455	455
発電機 3	0	0	0	0	0	130	130	130	130	130	130	130
発電機 4	0	0	0	0	130	130	130	130	130	130	130	130
発電機 5	0	0	25	40	25	25	25	30	85	162	162	162
発電機 6	0	0	0	0	0	0	0	0	20	33	73	80
発電機 7	0	0	0	0	0	0	0	0	25	25	25	25
発電機 8	0	0	0	0	0	0	0	0	0	10	10	43
発電機 9	0	0	0	0	0	0	0	0	0	0	10	10
発電機 10	0	0	0	0	0	0	0	0	0	0	0	10
東地域計	700	750	850	950	1000	1100	1150	1200	1255	1332	1332	1332
西地域計	0	0	0	0	0	0	0	0	45	68	118	168

時刻	13	14	15	16	17	18	19	20	21	22	23	24
発電機 1	455	455	455	455	455	455	455	455	455	455	455	455
発電機 2	455	455	455	310	260	360	455	455	455	455	420	345
発電機 3	130	130	130	130	130	130	130	130	130	0	0	0
発電機 4	130	130	130	130	130	130	130	130	130	0	0	0
発電機 5	162	85	30	25	25	25	30	162	85	145	25	0
発電機 6	33	20	0	0	0	0	0	33	20	20	0	0
発電機 7	25	25	0	0	0	0	0	25	25	25	0	0
発電機 8	10	0	0	0	0	0	0	10	0	0	0	0
発電機 9	0	0	0	0	0	0	0	0	0	0	0	0
発電機 10	0	0	0	0	0	0	0	0	0	0	0	0
東地域計	1332	1255	1200	1050	1000	1100	1200	1332	1255	1055	900	800
西地域計	68	45	0	0	0	0	0	68	45	45	0	0

第5章

確率計画問題のための個 体選択方法：確率的電力 設備配置問題への適用

5.1 電力システムにおける確率計画問題

不確実性を考慮した最適化が行える確率計画法は、現実の意思決定に非常に有用である。線形計画法など、確定環境下における最適化を対象とした一般的な最適化手法では、与えられた目的関数や制約に関する一組の入力パラメータに対して最適な解を求めている。しかし現実の意思決定においては、パラメータの実現値が得られる以前の段階で意思決定をしなければならないケースも多い。電力システムでも、将来の需要データなど、不確実なパラメータを基に最適化を行わなければならない計画問題が多数存在する。こうした不確実環境下における最適化を行う手法として、与えられる入力パラメータの実現可能な複数の組合せ（シナリオ）に対して、平均的に高い最適性を発揮する解を求める、確率計画法の研究が行われている。

確率計画法については、これまでもさまざまな研究が行われてきたが、現実の意思決定をより一層有効に支援するためには、より複雑な問題を効率的に解く手法の開発が望まれる。確率計画法の伝統的な手法の多くは、一般的な線形、あるいは非線形計画法が適用可能な問題を対象としている。しかし、電力システムに関する最適化問題の多くは、非線形、不連続、離散的な問題として定式化され、より複雑である。また多くの研究では、3から5程度のシナリオが与えられる問題を対象としているが、現実の電力システムの最適化に適用するには、より多くのシナリオを考慮した最適化を行うことが必要となる。

こうした複雑な確率計画問題を、遺伝的アルゴリズムを用いて効率的

に解く研究が行われている。[76, 77, 78, 79, 80]。これらの研究では、効率的に解を得るための方法としてサンプリングアプローチが採用されており、モンテカルロサンプリングなどにより選び出したシナリオの一部を用いて個体を評価することで、個体評価に要する計算時間を削減している。しかし、これまでに行われた研究では、選び出した一部のシナリオで個体を評価・選択することの理論的妥当性は示されていない。また、サンプル数を少なくすることで計算効率は高まるものの、個体評価の誤差が増し、得られる解の精度が悪化する。

そこで、サンプリングアプローチにより計算の効率化を図りつつ、統計的検定を用いた選択を行うことで、得られる解の精度を維持する遺伝的アルゴリズムを開発した。開発アルゴリズムでは、個体評価の度に新たなシナリオをサンプルする。そして、現在までの各世代における個体の適合度の分布を基に統計的な検定（Welch 検定）を行い、最良個体とはなりえないと判断された個体のみを淘汰の対象とする。淘汰の対象とならなかった個体については次世代へと生き残らせ、新たにサンプルしたシナリオでの評価を付け加えていく。これにより選び出した一部のシナリオで個体を評価することによる誤差（サンプル誤差）が原因で、遺伝的アルゴリズムの探索過程で生成された最良個体が誤って淘汰されてしまう確率は一定値以下に抑えられる。また、より多くの世代を生き残る個体ほど、多くのシナリオで評価されることになり、その個体の評価値はサンプル誤差の少ない、より正確な値となっていく。

開発アルゴリズムの有効性を確認するため、代表的な確率計画問題の一つである、確率的施設配置問題へ適用する数値実験を行った。27個のシナリオが与えられる有限シナリオのケースと、シナリオが確率分布として与えられる無限シナリオのケースの、二つのケースについて数値実験を行ったところ、いずれのケースにおいても開発アルゴリズムの有効性が確認できた。有限シナリオのケースにおいては開発アルゴリズムにより、すべてのシナリオを用いて個体を評価した場合と期待利益がほぼ等しい解を、約1/3の計算時間で求めることができた。また、無限シナリオのケースにおいては、100個のシナリオを用いて個体を評価した場合と同程度の期待利益の解が、約1/10の計算時間で求められた。

5.2 確率計画問題への遺伝的アルゴリズムの適用

本論文では式 (5.1) で定式化される、不確実なパラメータの実現可能な組合せがシナリオとして与えられる、リコース問題と呼ばれるタイプの確率計画問題を対象に検討を行う。

$$\max_x z = f(x) - \sum_{k \in K} \pi_k \cdot Q(x, \xi_k) \quad (5.1)$$

ここで、 $x = (x_1, x_2, \dots, x_n)$ は決定変数を表し、 $f(x)$ は目的関数のシナリオに依存しない部分を表している。 K はシナリオ集合を、 ξ_k はシナリオ k が発生した場合のパラメータの実現値を、 π_k はシナリオ k の実現確率を表している。 $Q(x, \xi_k)$ はシナリオ確定前に行った意思決定 x に対して、シナリオ k が実現した場合のリコースを表しており、

$$Q(x, \xi_k) = \min\{g(y, \xi_k) \mid h_j(y, \xi_k) \geq T_j(x, \xi_k), j = 1, 2, \dots, m\} \quad (5.2)$$

で計算される。ここでリコースとは、シナリオ確定後にシナリオの実現値と意思決定との差異を埋めるために必要となるコストを意味する。

この確率計画問題へ遺伝的アルゴリズムを適用する単純な方法は、決定変数 x を個体の染色体にコード化し、式 (5.1) の z で個体を評価することである。しかし、このためにはすべてのシナリオに対して $Q(x, \xi_k)$ を計算しなければならず、与えられたシナリオが少数の場合を除き、個体評価に多くの計算時間が必要となる。また無限シナリオのケースでは、シナリオの分布が特別な確率分布に従う場合を除き、全シナリオにおける平均適合度 (z) を計算することはできない。

このため、これまでに行われた確率計画問題へ遺伝的アルゴリズムを適用した多くの研究では、モンテカルロサンプリングなどにより選び出したシナリオの一部を用いて個体を評価する、サンプリングアプローチが採用されている [76, 77, 79, 78, 80]。しかし、これまでの研究では、サンプルしたシナリオを用いて個体を評価・選択することの理論的妥当性は示されていない。また、サンプル数を少なくすることで計算効率は高まるものの、個体評価の誤差が増し、得られる解の精度が悪化する。

サンプルした 100 から 200 のシナリオを個体評価に用いている文献 [76, 77, 79] の研究に対して、Loughlin らは各個体を評価する度に 10 個のシナリオをサンプルし、これを用いて個体を評価することで効率的に解を求める遺伝的アルゴリズムを提案している [78]。このアルゴリズムにより、個体評価に要する計算時間は大幅に削減される。しかし少数のサンプル

しか用いないことで個体評価の誤差が増し、文献 [78] の数値実験では、厳密には問題の制約条件を満たさない個体が最良個体として評価されている。Loughlin らの研究では、ある一定確率以内であれば制約条件を満たさないシナリオの存在を許容する、機会制約条件問題と呼ばれる確率計画問題を対象としているが、得られた解において制約を満たさないシナリオの割合は、必ずしも許容値を満たしていない。このため Loughlin らは、制約を満たさないシナリオの許容割合を 20% と厳しくすることで、制約を満たさないシナリオの割合が 25% 以下となる解を求めている。しかし、本論文で扱うリコース問題においては、このような方法で解の精度を改善することはできず、個体評価の誤差はそのまま目的関数値の悪化につながる。また、吉富らは各世代毎に一つのシナリオを選び出し、このシナリオに対する適合度で個体を評価し、最終的に全世代を通してもっとも出現頻度の高かった個体を最良個体とする遺伝的アルゴリズムを提案しているが、この方法でもシナリオ数が 2 の単純な問題においても最適解が得られないケースがあるなど、得られる解の精度に問題がある [80]。

5.3 アルゴリズム

複雑な確率計画問題を効率的に解くアルゴリズムとして、少数のサンプルしたシナリオにおける適合度で個体を評価し、統計的検定を用いた選択を行うことで解の精度を維持する、以下に示す遺伝的アルゴリズムを開発した。

5.3.1 個体評価

シナリオの発生確率に比例した確率で、モンテカルロサンプリングなどの方法によりシナリオを選び出し、式 (5.3) で計算される、サンプリングされたシナリオにおける平均適合度で個体を評価する。

$$z'(S) = f(x) - 1/|S| \sum_{k \in S} Q(x, \xi_k) \quad (5.3)$$

ここで $S \subset K$ は、与えられたシナリオ集合 K から、シナリオの発生確率 π_k に比例した確率で選び出したシナリオの集合を表す。また、 $|S|$ は集合 S の要素数である。

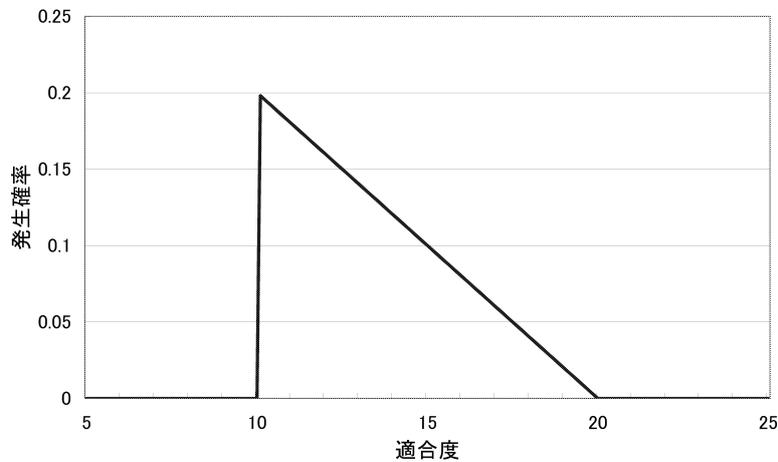


図 5.1: 個々のシナリオにおける適合度 ($z'(k)$) の分布 (1)

5.3.2 $z'(S)$ の分布

$z'(S)$ の値は選び出したシナリオ S に依存する。ただし、 $z'(S)$ の値は個々のシナリオ $k \in S$ における、ある個体の適合度の平均であるので、サンプルするシナリオ数 ($|S|$) が十分に大きければ、中心極限定理により、 $z'(S)$ の分布は正規分布に従うと仮定することができる。さらに大数の法則により、シナリオをサンプルし、 $z'(S)$ を計算することを繰り返すと、得られる $z'(S)$ の分布の平均は、全シナリオにおける個体の平均適合度 z に収束すると仮定できる。なお、ここでは上述のように、中心極限定理により $z'(S)$ の分布に正規分布を仮定するため、シナリオを選び出す確率にのみシナリオの発生確率を反映し、 $z'(S)$ の計算にはシナリオの発生確率を用いない。

たとえば与えられたシナリオに対して、ある個体の適合度が図 5.1、図 5.2 のように分布しているとする。この時、シナリオの発生確率に比例した確率で、それぞれ 5, 10, 20 個のシナリオを選び出し、この個体の平均適合度 ($z'(S)$) を計算することを 100 回繰り返すと、得られる $z'(S)$ の分布は表 5.1、表 5.2 のようになる。ここで表 5.1、表 5.2 の各列の括弧内の数値は、 $z'(S)$ が平均 z の正規分布に従った場合の理論値を表している。このように無限シナリオで、しかもシナリオによる個体の適合度の偏りが大きい図 5.1 や図 5.2 のケースにおいても、 $z'(S)$ の分布はほぼ正規分布と一致する。

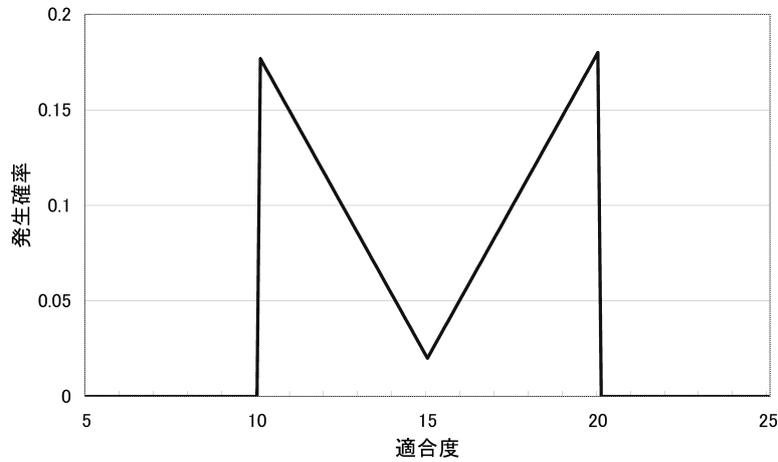


図 5.2: 個々のシナリオにおける適合度 ($z'(k)$) の分布 (2)

表 5.1: 図 5.1 のシナリオにおける $z'(S)$ の分布

$ S $	平均		分散		歪度		尖度	
5	13.31	(13.33)	1.08	(1.11)	0.15	(0.00)	-0.55	(3.0)
10	13.40	(13.33)	0.51	(0.56)	0.12	(0.00)	0.10	(3.0)
20	13.32	(13.33)	0.33	(0.28)	0.001	(0.00)	-0.74	(3.0)

5.3.3 統計的選択方法

シナリオをサンプルし、各個体の $z'(S)$ を計算することを繰り返すことで得られる $z'(S)$ の分布が、平均 z の正規分布に従うとすると、以下に示す Welch 検定を用いることで、二つの異なる個体の z の差の有意性を、いくつかのサンプルシナリオ S に対する $z'(S)$ の値を基に推定することができる。

Welch 検定

$x_i, i = 1, 2, \dots, m$ と $y_j, j = 1, 2, \dots, n$ をそれぞれ $N(\mu_1, \sigma_1^2)$, $N(\mu_2, \sigma_2^2)$ から選び出した m , n 個のサンプルとする。この時、式 (5.4) で計算される t は、自由度 ν の t 分布に従う。

$$t = -\frac{X - Y}{\sqrt{s_x^2/m + s_y^2/n}} \quad (5.4)$$

表 5.2: 図 5.2 のシナリオにおける $z'(S)$ の分布

$ S $	平均		分散		歪度		尖度	
5	14.95	(15.00)	2.17	(2.33)	0.02	(0.00)	-0.38	(3.0)
10	15.18	(15.00)	1.06	(1.17)	0.03	(0.00)	0.32	(3.0)
20	15.00	(15.00)	0.68	(0.58)	-0.16	(0.00)	-0.51	(3.0)

ここで、 X, Y はそれぞれ x_i, y_j の平均を表し、 s_x^2, s_y^2 はそれぞれ x_i, y_j の分散を表している。また、 ν は式 (5.5) で計算される ν' にもっとも近い整数である。

$$\nu' = \frac{(s_x^2/m + s_y^2/n)^2}{\frac{(s_x^2/m)^2}{m-1} + \frac{(s_y^2/n)^2}{n-1}} \quad (5.5)$$

開発アルゴリズムでは、この Welch 検定を用いて、現世代の最良個体と比較して、有意に適合度が低いと判断された個体のみを淘汰の対象とし、この中から各世代毎にある一定数を上限に適合度の低い個体を淘汰する。

5.3.4 開発アルゴリズムのフロー

開発アルゴリズムでは以下に示す手順により、サンプルしたシナリオにおける適合度から個体を評価・選択し、最適な解を探索する。

(1) 初期化

- (1-1) ランダムに N 個の個体からなる初期集団を生成する。
- (1-2) 個体 i の履歴データを $g_i = Z_i = \sigma_i^2 = 0$ とする。ここで g_i は個体 i が生き延びた世代数を表し、 Z_i と σ_i^2 はそれぞれ、個体 i が生き延びた過去の世代における適合度 $z'(S)$ の平均と分散を表す。

(2) 個体の評価

- (2-1) 各シナリオの発生確率に比例した確率で、与えられたシナリオから m 個のシナリオをランダムに選び出す。
- (2-2) 選び出した m 個のシナリオの集合を S とし、シナリオ集合 S に対する個体 i の平均適合度 $z'_i(S)$ を計算する。

(2-3) 個体 i の履歴データを式 (5.6) - 式 (5.8) により更新する。

$$g_i = g_i + 1 \quad (5.6)$$

$$Z_i = Z_i + (z'_i(S) - Z_i)/g_i \quad (5.7)$$

$$\sigma_i^2 = \frac{(g_i - 2)(z'_i(S) - Z_i)^2}{g_i} + \frac{(g_i - 2)\sigma_i^2}{g_i - 1} \quad (5.8)$$

(3) 個体の選択

(3-1) 現在の世代の中で Z_i が最大となる個体を選び、これを i^* とする。

(3-2) 個体の集合 $D = \{j \mid t_j > t_\alpha(\mu)\}$ を選ぶ。ここで

$$t_j = \frac{Z_{i^*} - Z_j}{\sqrt{\sigma_{i^*}^2/g_{i^*} + \sigma_j^2/g_j}} \quad (5.9)$$

である。また、 $t_\alpha(\mu)$ は自由度 μ 、信頼区間 $\alpha\%$ の t 値であり、 μ は

$$\mu' = \frac{(\sigma_{i^*}^2/g_{i^*} + \sigma_j^2/g_j)^2}{\frac{(\sigma_{i^*}^2/g_{i^*})^2}{g_{i^*}-1} + \frac{(\sigma_j^2/g_j)^2}{g_j-1}} \quad (5.10)$$

にもっとも近い整数を表す。

(3-3) 集合 D に含まれる個体数が n_w 以上なら、適合度の低い順に n_w 個の個体を淘汰する。そうでなければ集合 D に含まれるすべての個体を淘汰する。

(4) 再生

(4-1) 現世代の集団から $\min\{n_w, |D|\}$ 個の個体を選び出す。

(4-2) 選び出した個体からランダムに個体ペアを作り、交叉の遺伝的操作を加えて新しい個体を生成する。

(4-3) 新しい個体に突然変異の遺伝的操作を加える。

(4-4) 新しい個体の履歴データを初期化する。

(5) 終了判定

終了条件を満たしたならアルゴリズムを終了する。 g_i の値が一定数以上で、 Z_i の値が最大となる個体を最良個体として選択する。そうでなければ (2) からの操作を繰り返す。

5.3.5 開発アルゴリズムの特徴

この開発アルゴリズムでは，現世代で平均適合度 (Z_i) が最大となる個体と比べ，Welch 検定により有意に平均適合度が低いと判断された個体のみを淘汰の対象とし，他の個体については次の世代へと生き残らせる。これにより遺伝的アルゴリズムの探索過程で生成された最良の個体が，サンプル誤差により誤って淘汰されてしまう確率は，あらかじめ定めた一定値以下に抑えられる。理論的には各世代の選択で最良の個体が淘汰されてしまう確率は $\alpha\%$ 以下となる。また生き残った個体については，次の世代で新たにサンプルされたシナリオでの評価が加えられることで，その平均適合度 (Z_i) はより正確な (z に近い) 値となり， $z'(S)$ のサンプルが増えることから，より正確な差の検定が行えるようになる。

このように開発アルゴリズムでは遺伝的アルゴリズムの世代が進行していく中で，より多くの世代を生き残る個体ほど，その平均適合度はより正確な値となっていく。一方，平均適合度の低い個体については，ある程度の精度の適合度が得られた段階で淘汰されるので，適合度の低い個体の評価に要する計算時間は削減される。

5.4 数値実験

開発アルゴリズムの有効性を検証するため，代表的な確率計画問題の一つである，確率的施設配置問題 [81, 82] を対象とする数値実験を行った。施設配置問題とは最適な施設の規模と，与えられた候補地の中から最適な施設の設置場所を決める問題である。ここでは需要，製造コスト，輸送コスト，製品の価格を不確定パラメータとし，これらのパラメータが有限シナリオとして与えられるケースと，確率分布として与えられる無限シナリオのケースの，二つのケースについて数値実験を行った。

5.4.1 確率的施設配置問題の定式化

確率的施設配置問題を定式化するにあたり，以下の記号を定義する。

I : 需要地の集合

J : 施設が設置可能な候補地の集合

K : 与えられたシナリオの集合

x_j : 候補地 j に施設が設置されるなら 1, その他の場合は 0 の値をとる変数

$y_{i,j}^k$: シナリオ k が生じた場合, 需要地 i の需要のうち, 候補地 j に設置された施設から供給される製品の割合

f_j : 候補地 j に施設を設置するために必要となる固定費

v_j : 候補地 j に設置する施設の生産能力を表す変数

g_j : 候補地 j に施設を設置するために必要となる, 工場の生産能力に比例する変動費

d_i^k : シナリオ k が生じた場合の需要地 i の需要

p_i^k : シナリオ k が生じた場合の需要地 i での製品一単位当たりの価格

$c_{i,j}^k$: シナリオ k が生じた場合の, 候補地 j の施設で製品を製造し, これを需要地 i へと供給するために必要となる, 製造コストと輸送コストの合計

π_k : シナリオ k の発生確率

上記の記号を用いると, 確率的施設配置問題は以下のように定式化される。

確率的施設配置問題

$$\max_{x_j, y_{i,j}^k, v_j} Z_p = - \sum_{j \in J} g_j \cdot v_j - \sum_{j \in J} f_j \cdot x_j \quad (5.11)$$

$$+ \sum_{i \in I, j \in J, k \in K} \pi_k \cdot d_i^k (p_i^k - c_{i,j}^k) y_{i,j}^k \quad (5.12)$$

$$\text{s.t.} \quad \sum_{j \in J} y_{i,j}^k \leq 1, i \in I \quad (5.13)$$

$$\sum_{i \in I} d_i^k \cdot y_{i,j}^k - v_j \leq 0, j \in J, k \in K \quad (5.14)$$

$$y_{i,j}^k - x_j \leq 0, i \in I, j \in J, k \in K \quad (5.15)$$

$$x_i \in \{0, 1\}, i \in I \quad (5.16)$$

$$y_{i,j}^k \geq 0, i \in I, j \in J, k \in K \quad (5.17)$$

$$v_j \geq 0, j \in J \quad (5.18)$$

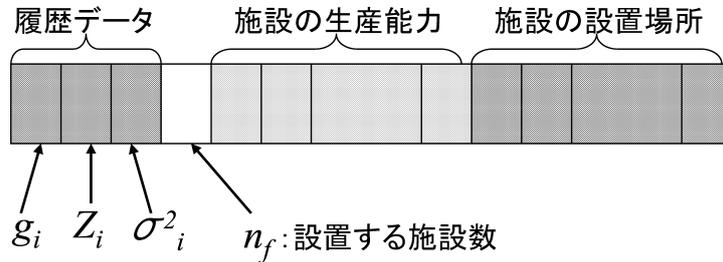


図 5.3: 開発アルゴリズムでの染色体表現

問題のパラメータについては、文献 [81] などで用いられている代表的なベンチマーク問題 (Kuehn-Hamburger 問題 [83]) を参考に、以下のように設定した。需要地数を 50, 施設配置の候補地数を 16 とし, 施設の固定コスト f_j を設置場所 j に関わらず 750, 設備の変動費 g_j を j に関わらず 150 とした。設備の生産能力については文献 [81] では連続値としているが, ここではより現実的な条件として, 選択可能な設備の生産能力を 150, 300, 450 の三種類とした。

5.4.2 染色体表現

上述の確率的施設配置問題は, x_j と v_j を一段階目の決定変数, $y_{i,j}^k$ を二段階目の決定変数とする二段階問題となる。そして一段階目の決定変数 x_j, v_j が確定すれば, これに対して最適な二段階目の決定変数 $y_{i,j}^k$ の値は, 単純な輸送問題を解くことで求められる。そこで, ここでは一段階目の決定変数のみを染色体としてコード化し, 個体評価の際には輸送問題を解くことで, 二段階目の変数 $y_{i,j}^k$ の最適な値を求めた。

一段階目の決定変数の染色体へのコード化については, 図 5.3 に示す表現方法を採用した。染色体は大きく四つのブロックにより構成される。このうち最初のブロックは, Welch 検定に利用する個体の履歴情報を保持する三つの遺伝子により構成される。最初の遺伝子はその個体は何世代生き残っているかを表している。二番目と三番目の遺伝子にはそれぞれ, その個体が過去の世代において評価を受けた $z'(S)$ の平均と分散が記録される。二つ目のブロックは一つの遺伝子から構成され, 設置する施設数 n_f を表す。第三, 第四のブロックは, とともに候補地の数と等しい数の遺

表 5.3: 開発アルゴリズムによる解の期待利益

危険率	期待利益		
	平均	最良	最悪
1%	6309.8	6321.4	6297.6
5%	6309.1	6321.4	6264.5
10%	6305.4	6321.4	6274.5

伝子から構成される。それぞれ最初の n_f 個の遺伝子だけが意味を持ち、施設を設置する候補地と設置する施設の規模を表している。

5.4.3 シナリオ

無限シナリオのケースについては、需要 d_i^k 、製品価格 p_i^k 、製品の製造・輸送コスト $c_{i,j}^k$ が、それぞれ $30.0 \pm 10\%$ 、 $20.0 \pm 15\%$ 、 $15.0 \pm 25\%$ の一様分布に従うものとした。また有限シナリオのケースについては、それぞれの不確実パラメータの分布が上記の一様分布に従うとしたうえで、27 個のシナリオをモンテカルロサンプリングにより選び出した。

5.4.4 遺伝的操作とパラメータ設定

ルーレット選択を用いて、適合度 (Z_i) の高さに比例した確率で二個体を選び出し、これらの個体に対して確率 0.8 で二点交叉を加え、新たな二個体を作り出す。また、新たに作られた個体に対して、確率 0.1 の一様変異を加える。数値実験では初期集団の個体数を 50 とし、この集団を 100 世代に達するまで進化させ、集団の中の 20 世代以上を生き残った個体の中から、もっとも Z_i が大きい個体を最良個体として採用した。

5.4.5 結果

最良個体が淘汰される危険率 (Welch 検定を行う際の有意水準) を 1%、5%、10% に設定した 3 ケースについて、開発アルゴリズムをそれぞれ 10 回実行したところ、一回当たり約 900 秒の計算時間で、表 5.3 に示す期待利益 z の配置案が求められた。次に開発アルゴリズムとの比較のため、与

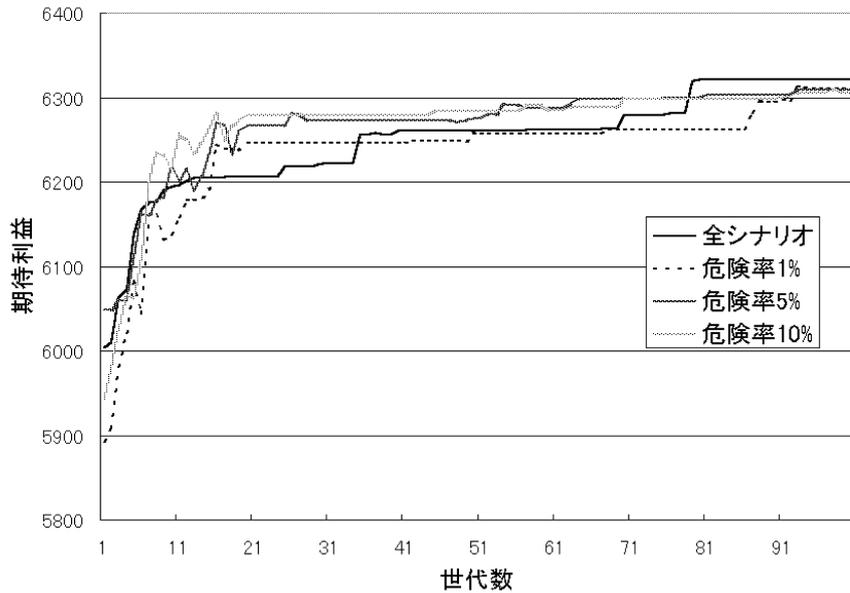


図 5.4: 世代数に対する最良個体の変化

えられた 27 シナリオ全てで個体を評価する遺伝的アルゴリズム（以下，全シナリオ GA と呼ぶ）により解を求めたところ，10 回の実行の全てにおいて期待利益 z が 6321.4 となる解が，約 4,000 秒の計算時間で求められた。なお，全シナリオ GA の遺伝的操作とパラメータについては，5.4.4 節で示した提案アルゴリズムと同様とした。

この数値実験では開発アルゴリズムにより，全シナリオ GA とほぼ同程度の解が 1/4 以下の計算時間で得られている。例えば危険率を 5% とした場合，全シナリオ GA と比較した期待利益の減少は平均で 0.19% であり，最悪ケースでも 1% 以下である。また，10 回の実行のうちの 6 回，全シナリオ GA による解と同一の解を得ることができた。アルゴリズムの実行過程における性能で見ても，二つのアルゴリズムの性能はほぼ同程度となっている（図 5.4）。図 5.4 には，それぞれのアルゴリズムを実行した際に得られた，各世代の最良個体の全シナリオで評価した適合度（ z ）の 10 回の実行における平均が図示してある。なお，開発アルゴリズムについては，世代数が 20 以上の場合は 20 世代以上を生き残った個体の中で， Z_j が最大となる個体を最良個体とした。

図 5.4 では，世代の進行に対する最良個体の進化を比較したが，全シナリオ GA と開発アルゴリズムとでは，個体評価に用いるシナリオ数に違

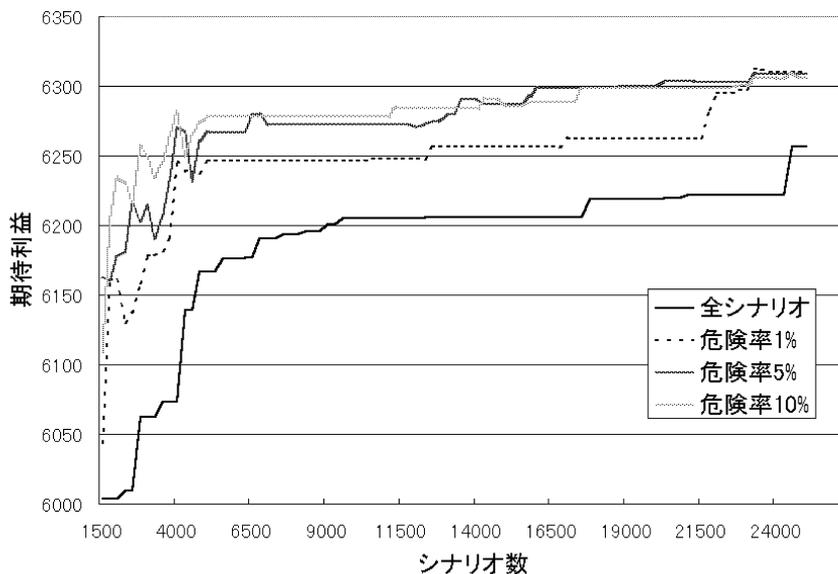


図 5.5: シナリオ数に対する最良個体の変化

いがある。そして、一つの個体を評価するための計算時間は、評価に用いるシナリオ数に比例して増加する。そこで、二つのアルゴリズムの計算効率を比較するため、アルゴリズムの実行段階からの個体評価に用いたシナリオの累計と、最良個体の進化との関係を調べたところ、図 5.5 のようになった。図 5.5 を見ると、開発アルゴリズムの有効性がより一層明らかとなる。たとえば開発アルゴリズムでは約 4,000 シナリオを評価した段階で、期待利益 6250 (誤差約 1%) の解が得られているが、全シナリオ GA で同程度の解を得るには、6 倍以上の約 24,000 シナリオの評価が必要となっている。なお、図 5.4, 図 5.5 を見ると、特にアルゴリズム実行の初期段階で、危険率の設定が低いほど適合度の高い個体が得られない傾向が見受けられる。これは危険率を低く設定すると、特にすべての個体が少しの世代しか経過していない (個体の評価値の精度が高くない) アルゴリズムの初期段階においては、Welch 検定により淘汰の対象と判断される個体数が少なくなり、この結果新たな個体が生成されず、個体進化が進まなくなることが原因と考えられる。

上記のように、全シナリオ GA とほぼ同程度の解を探索するうえで、開発アルゴリズムは非常に効率的である。ただし、表 5.3 に示したように、開発アルゴリズムにより、10 回の実行のすべてで全シナリオ GA と等しい

表 5.4: 期待利益 6321.4 の解を得るまでに個体評価に要した平均シナリオ数

	平均シナリオ数
全シナリオ	31,860
危険率 1%	32,850
危険率 5%	27,875
危険率 10%	29,000

解が得られているわけではない。そこで、アルゴリズムの終了条件である 100 世代に達しても解が得られなかった場合を含め、期待利益が 6321.4 となる個体を得るまでに、個体評価に要したシナリオ数の累計の平均を比べたところ表 5.4 のようになった。表 5.4 の結果を見ると、期待利益最大の解を得るという観点からも、全シナリオ GA より、危険率を 5%, 10% に設定した開発アルゴリズムのほうが効率的となっている。

次に以下に示す二つの個体の評価・選択方法を採用した遺伝的後リズムと、開発アルゴリズムとを比較した。

方法 1: アルゴリズムの実行に先立ち、モンテカルロサンプリングにより 100 個のシナリオを選び出し、常にこのシナリオにおける平均適合度で個体を評価する

方法 2: 個体評価の度に新たに 10 個のシナリオを選び出し、このシナリオにおける適合度の平均で個体を評価する

文献 [78, 76, 77] の研究は機会制約条件問題を対象としたものであるが、これらの研究で採用されている個体の評価・選択方法を、そのまま本論文で扱うリコース問題に適用したとすると、方法 1 は文献 [76, 77, 79] で採用されている評価・選択方法に、方法 2 は文献 [78] で採用されている評価・選択方法にそれぞれ相当する。また、開発アルゴリズムについては、最良個体が淘汰される危険率（Welch 検定を行う際の有意水準）と、サンプルするシナリオ数を以下のように設定した。危険率の設定としては 10%, 15%, 20% の 3 種類、サンプルするシナリオ数については 5, 10 の 2 種類の、合計 6 種類のパラメータ設定の組合せについて数値実験を行った。

表 5.5: 数値実験の結果 (有限シナリオケース)

	方法 1	方法 2	開発アルゴリズム	
	27	10	5	5
危険率	-	-	10%	15%
平均	6321.4	6304.4	6305.4	6314.6
分散	0.0	675.1	366.3	241.9
最大	6321.4(10)	6321.4(6)	6321.4(5)	6321.4
最小	6321.4	6252.4	6274.5	6274.5
計算時間 (秒)	1,200	780	450	450

	開発アルゴリズム			
	5	10	10	10
危険率	20%	10%	15%	20%
平均	6312.2	6316.9	6319.3	6319.0
分散	262.8	89.9	44.3	56.8
最大	6321.4(7)	6321.4(8)	6321.4(9)	6321.4(9)
最小	6274.5	6297.6	6300.4	6297.6
計算時間 (秒)	450	850	850	850

有限シナリオのケース

与えられた 27 のシナリオに対する最適な施設配置を、開発アルゴリズムと方法 1, 2 の評価・選択方法を採用した遺伝的アルゴリズムを用いて、それぞれ 10 回ずつ求めたところ、表 5.5 の結果となった。表 5.5 には 10 回の実行における期待利益の平均と分散、得られた期待利益の最大値と最小値、それに 1 回の実行に要した計算時間を示した。また、期待利益の最大値の行の括弧内の数字は、それぞれの方法を 10 回実行した時に、期待利益が 6321.4 となる解が得られた回数を表している。なお、与えられたシナリオの数が 100 以下となるこのケースでは、方法 1 の適用にあたっては、すべてのシナリオにおける適合度の平均で個体を評価した。

表 5.5 の結果を見ると、開発アルゴリズムを用いることで、よりサンプル誤差の影響の少ない個体選択が可能となっている。単純にサンプルした 10 個のシナリオで個体を評価する方法 2 を用いた場合には、すべてのシナリオで個体を評価する方法 1 を用いた場合と比べ、期待利益は平均で 0.27%、最悪ケースでは 1.09% 減少している。これに対して同じサンプル

数でも、開発アルゴリズムを用いた場合の期待利益の減少はより小さく、たとえば危険率を15%とした場合の期待利益の減少は、平均で0.03%、最悪ケースでも0.33%に抑えられている。また10回の実行で方法2とサンプル数を10とした開発アルゴリズムにより得られた期待利益の差を統計的に検定すると、危険率の設定が10%の場合には有意水準20%で、危険率の設定が15%と20%の場合には有意水準15%で、その差は有意なものとなった。なお、各手法により得られた期待利益の分布が偏っており、母集団に正規性を仮定することができないことから、ここではノンパラメトリック検定の一種であるWilcoxon検定により期待利益の差を検定した。Wilcoxon検定とは得られたデータを順位に置き換えることで、二つの母集団の差を検定する手法である。

また、この数値実験では危険率を10%とした場合を除き、サンプル数を5としても得られる解の精度はそれほど悪化しておらず、サンプル数を5とした開発アルゴリズムの計算効率が非常に高くなっている。サンプル数を10とした場合と比べ、サンプル数を5とすることで期待利益の平均は減少し、分散は増加する傾向が見られる。しかし、この差は大きなものではなく、Wilcoxon検定を適用した結果では、危険率を10%に設定した場合を除き、サンプル数を5とした場合とサンプル数を10とした場合とで、得られた期待利益の間に統計的に有意な差は認められない。一方でサンプル数を5とすることで、解を得るまでに要する計算時間は大幅に削減されている。また、サンプル数を5とした場合でも、平均的には開発アルゴリズムを2回実行することで、期待利益が6321.4となる配置案が求まっている。サンプル数を5とした場合には、方法1の約1/3の計算時間で解を求めることができることを考えると、期待利益が6321.4となる配置案を得る平均的な性能で評価しても、サンプル数を5とした開発アルゴリズムの方が方法1より効率的となっている。

なお、この数値実験では危険率を10%、サンプル数を5に設定した場合、他の設定に比べて開発アルゴリズムにより得られる解の精度が悪化している。これはサンプル数を5とした場合には $z'(S)$ の分散が大きく、危険率の設定を10%と厳しくした場合には、Welch検定により淘汰の対象と判断される個体が非常に少なくなっていることから、新たに生成される個体が少なくなり、集団の進化が阻害されることが原因になっていると考えられる。

表 5.6: 数値実験の結果 (無限シナリオケース)

	方法 1	方法 2	開発アルゴリズム	
	100	10	5	5
サンプル数	100	10	5	5
危険率	-	-	10%	15%
平均	6253.2	5868.8	6255.1	6248.8
分散	220.5	127395.9	202.4	596.8
最大	6274.2	6237.8	6286.2	6296.8
最小	6228.2	5109.25	6236.2	6210.2
計算時間 (秒)	4,430	780	450	450

	開発アルゴリズム			
	5	10	10	10
サンプル数	5	10	10	10
危険率	20%	10%	15%	20%
平均	6242.7	6243.4	6245.3	6239.2
分散	287.0	411.6	270.3	401.3
最大	6274.8	6274.0	6269.6	6262.1
最小	6219.6	6212.8	6213.5	6209.0
計算時間 (秒)	450	850	850	850

無限シナリオのケース

不確実パラメータが確率分布として与えられた無限シナリオのケースに対して、開発アルゴリズムと方法 1, 2 の個体の評価・選択を採用した遺伝的アルゴリズムを、それぞれ 10 回ずつ適用し、最適な配置案を求めた。それぞれのアルゴリズムで得られた配置案の期待利益を、モンテカルロサンプリングにより選び出した 1,000 個のシナリオを用いて評価したところ、表 5.6 に示す結果となった。

この無限シナリオのケースでは、開発アルゴリズムの有効性がより一層明らかとなっている。単純に 10 個のシナリオで個体を評価する方法 2 を用いた場合には、得られる解の期待利益のばらつきは非常に大きくなっている。これに対して開発アルゴリズムを用いた場合には、サンプル数を 5 とした場合でも、100 個のシナリオで個体を評価する方法 1 を用いた場合と同程度の期待利益の解を約 1/10 の計算時間で得ることができている。この場合についても Wilcoxon 検定を適用したところ、個体評価に方法 2 を用いた場合と開発アルゴリズムにより得られた期待利益の差は、

いずれのサンプル数と危険率の設定においても有意水準1%で有意となった。また、この無限シナリオのケースでは、開発アルゴリズムのサンプル数の違いによる期待利益の差は、いずれのケースにおいても統計的に有意とはならなかった。

第6章 おわりに

メタヒューリスティクスの発展と、計算機性能の向上により、これまで最適化問題の適用が難しかった問題も、実用的な計算時間で解くことが可能となっている。しかし、電力システムに関する最適化問題でメタヒューリスティクスの実用化を進めるには、大規模問題での収束性能の向上と、最悪ケースでの精度保障の二つの課題解決が望まれる。そこで本論文では、上記の課題を解決し、電力システムの実問題の最適化に適用可能なアルゴリズムを開発した。

本論文では、電力システムにおける代表的な3タイプの最適化問題（割当て問題，混合整数計画問題，確率計画問題）に対し，大規模問題となる実問題に対しても，実用上必要となる精度の解を効率的に求める遺伝的アルゴリズムを開発した。開発したアルゴリズムでは，他の手法との融合が容易であるという遺伝的アルゴリズムの特徴を活かし，問題の特徴に合わせて選択した他の手法を組み合わせることで，間接的に最悪ケースでの精度保障を行うとともに，大規模問題での収束性能を向上させている。

電力用移動無線の周波数割当て問題に代表される大規模割当て問題に対しては，グリーディアルゴリズムとの組合せにより，大規模問題を効率良く解く遺伝的アルゴリズムを開発した。染色体にコード化した割当て優先度と，グリーディアルゴリズムにより計算される割当て優先度との組合せにより，割当て順序を決定する開発アルゴリズムでは，最悪ケースでもグリーディアルゴリズムで求まる解が得られることが保障される。また，基地局数が500を超える大規模な問題に開発アルゴリズムを適用したところ，これまでの順序表現を用いた遺伝的アルゴリズムでは求めることができなかつた，必要周波数が少ない割当てを求めることができ，探索能力の向上が確認できた。このグリーディアルゴリズムと遺伝的アルゴリズムの組合せアルゴリズムについては，周波数割当て問題に限らず，割当て問題として定式化される，ガスタービン発電機部品の最適なローテーション計画においても，その有効性が確認されている [84]。

発電機の起動停止計画問題に代表される，大規模な非線形混合整数計画問題に対しては，個体評価に連続緩和法を組み入れたことを特長とする遺伝的アルゴリズムを開発した。開発アルゴリズムでは，一部の整数変数の値を固定し，残りを未固定とした部分解を個体の染色体にコード化する。そして，個体評価の際には，未固定となっている整数変数と連続変数の値を，連続緩和問題の最適解を基に決定する。このため最悪ケースでも，連続緩和法により求まる解以上の解が求まることが保障される。また，開発アルゴリズムを大規模な発電機の起動停止計画のベンチマーク問題に適用したところ，連続緩和法を組み合わせることにより遺伝的アルゴリズムの探索能力が向上し，遺伝的アルゴリズム以外の手法を含め，既存の計画手法では求めることのできなかつた，経済的な起動停止計画の策定が可能となった。

数多くのシナリオが与えられる，複雑な確率計画問題に対しては，統計的検定を用いて個体選択を行う遺伝的アルゴリズムを開発した。開発アルゴリズムでは，ランダムに選択したシナリオを用いて個体を評価することで，個体評価に要する計算時間を削減する。ただし単純にサンプリングしたシナリオで個体評価を行った場合，全シナリオで評価した適合度が最大となる最良個体が，サンプリング誤差により淘汰されてしまう危険性がある。そこで，開発アルゴリズムでは，統計的検定により有意に最良個体とはなり得ないと判断された個体のみを淘汰する個体選択を行うことで，サンプリング誤差により適合度の高い個体が淘汰されてしまう危険率を一定値以下に抑える。開発アルゴリズムを施設配置計画問題へ適用したところ，多数シナリオに対する期待利益が最大となる配置案を短時間で求めることができ，提案遺伝的アルゴリズムの有効性が確認できた。開発アルゴリズムについては，複数の需要シナリオに対して，処理時間とシステムの信頼を制約に，保守コストを含めたシステムコストが最小となる，最適な計算機機器の選択／配置とネットワーク容量を決定する，電力用情報通信システムの最適設計問題へと適用し，大規模・複雑な問題が開発アルゴリズムを用いることで，実用的な計算時間で解けることを確認した [85]。

参考文献

- [1] 福山良和：「メタヒューリスティック手法の電力・エネルギー分野への適用例」，電学論 B, 124, 5, 679–682 (2004)
- [2] F. Glover: “Tabu Search Part I”, ORSA Journal of Computing, Vol. 1, No. 3, pp190–206 (1989)
- [3] F. Glover: “Tabu Search PartII”, ORSA Journal of Computing, Vlo. 2, No. 1, pp4–32 (1990)
- [4] J.H. Holland: “Adaptation in Natural and Artificial Systems,” University of Michigan Press (1975), MIT Press (1992).
- [5] L. Chambers (ed.): *Practical Handbook of Genetic Algorithms: Applications Volume I*, CRC Press (1995).
- [6] L. Chambers (ed.): *Practical Handbook of Genetic Algorithms: New Frontiers Volume II*, CRC Press (1995).
- [7] D.B. Fogel: *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press (1999).
- [8] Z. Michalewicz and D.B. Fogel: *How to Solve It: Modern Heuristics*, Springer (1999).
- [9] 北野宏明（編）：「遺伝的アルゴリズム」，産業図書，(1993).
- [10] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi: “Optimization by simulated annealing”, Science, 220, 671–680 (1983)
- [11] 電力系統へのメタヒューリスティクス応用調査専門委員会：「電力系統へのメタヒューリスティクス応用技術」，電気学会技術報告，第 923 号，(2003)

- [12] 西村, 他: 「配電系統計画支援システムの開発 -GA の配電損失最小化問題, 負荷融通問題への応用-」, 平成 10 年電気学会全国大会, 138-139 (1998)
- [13] 森啓之, 飯村吉典: 「2 層近傍タブサーチによる分散電源を考慮した配電系統拡張計画」, 電学論 B, Vol. 123, No.10, 1142-1148 (2003)
- [14] 小枝一彦, 造賀芳文, 上山哲平, 餘利野直人, 佐々木博司: 分散型電源の導入を考慮した配電系統の設備計画手法の検討」, 電学論 B, Vol.126 No.8 (2006)
- [15] 森, 他: 「並列タブサーチを用いた送電網拡張計画法」, 平成 11 年電気学会電力技術・電力系統技術合同研究会, PE-99-13, PSE-99-10 (1999)
- [16] 松尾, 他: 「送電網の過負荷解消のための経済性を考慮した FACTS 機器設置の最適化」, 電学論 B, 120, 1061-1070 (2000)
- [17] 森啓之, 谷英伸: 「2 層近傍タブサーチに基づく配電系統における D-FACTS の最適配置解法」, 電学論 B, Vol.124, No.3 (2004)
- [18] 杉本淳司郎, 横山隆一, 藤田吾郎, 福山良和: 分散型電源導入時における電圧変動を考慮した SVR および SVC の協調配置手法」, 電学論 B, Vol.126, No.12 (2006)
- [19] 森啓之, 小笠原寿志: 「短期負荷予測のための GA による最適ニューラルネットワーク」, 電気学会研究会資料, PE-93-98, pp49-55 (1993)
- [20] 加藤裕之, 他: 「GA を利用したコンカレントニューラルネットワークによる翌日最大電力予測」, 電学論 B, 115, 8, pp875-882 (1995)
- [21] 森啓之, 曾根康之: 「タブサーチを用いたニューラルネットワークによる翌日最大負荷予測」, 平成 10 年度電気学会全国大会, 1376, 6-177 (1998)
- [22] 板垣忠大, 森啓之: 「短期電力負荷予測におけるクラスタ再構成前処理手法」, 電学論 B, Vol.125, No.3 (2005)
- [23] 森啓之, 曾根康之: 「タブサーチを用いた最適化ファジイモデリングによる翌日最大負荷予測」, 平成 9 年度電気学会全国大会, 1379, 6-19 (1997)

- [24] 雪田和人, 加藤慎也, 後藤泰之, 一柳勝宏, 川島靖弘: 「構造的 GA による GMDH を用いた翌日最大電力需要予測」, 電学論 B, Vol.124, No.3 (2004)
- [25] 森啓之, 松崎理: 「発電機起動停止計画問題への並列タブサーチの適用」, 平成 11 年電気学会全国大会, 1393, 6-139 (1999)
- [26] 永田武, 他: 「遺伝的アルゴリズムによる大規模な発電機起動停止計画問題の解法」, 電学論 B, 118, 4, pp.413-419 (1998)
- [27] 吉川元庸: 「タブサーチを用いた自動作業停電調整支援システムの開発」, 平成 8 年度電気学会電力・エネルギー部門大会, 243, pp.85-86 (1996)
- [28] 奈良宏一, 他: 「火力発電機定期補修計画問題へのタブ探索の適用」, 電気学会研究会資料, PE-98-137, PSE-98-127, pp.7-11 (1998)
- [29] 佐藤泰司, 他: 「Simulated Annealing 法による火力発電機の定期補修計画」, 電気学会研究会資料, PE-90-21, pp.129-138 (1990)
- [30] 金賢哲, 他: 「SA と GA の融合による火力発電機定期補修計画問題の解法」, 電気学会研究会資料, PE-93-102, pp.81-90 (1993)
- [31] 森啓之, 荻田能弘: 「PTS の事故復旧操作のための配電系統再構成最適化への応用」, 電気学会研究会資料, PSE-01-5, pp.25-30 (2001)
- [32] 稲垣潤, 中島純, 長谷川美紀, 北島秀夫: 「配電系統事故復旧問題の遺伝的アルゴリズムを用いた多目的最適化に基づく解法」, 電学論 B, Vol.124, No.3 (2004)
- [33] 古田篤広, 森啓之: 「配電系統復旧制御への並列タブサーチを用いた階層的最適化手法の適用」, 電学論 B, Vol.126, No.1 (2006)
- [34] 山本顕正, 青木秀憲, 直井健二, 水谷芳史: 「制御手順を考慮した GA と TS の融合による電圧無効電力制御」, 電学論 B, Vol. 124, No.12 (2004)
- [35] D. E. Goldberg: “Genetic Algorithms in Search, Optimization, and machine Learning,” Addison-Wesley (1989).

- [36] D. Kuntz: "Channel assignment for cellular radio using neural networks," *IEEE Trans. Veh. Technol.*, vol.40, no.1, pp.188-193, Feb. (1991).
- [37] N. Funabiki and Y. Takefuji: "A neural network parallel algorithm for channel assignment problems in cellular radio networks," *IEEE Trans. Veh. Technol.*, vol.41, no.4, pp.430-437, Nov. (1992).
- [38] M. O. Berger: "Neural channel assignment - the fast way," *Proc. of IEEE Int. Conf. Neural Netw.*, vol.1, pp.1557-1560, Nov. (1995).
- [39] J.-S. Kim., S. H. Park, P. W. Dowd, and N. M. Nasrabadi: "Comparison of two optimization techniques for channel assignment in cellular radio network," *Proc. of IEEE Int. Conf. Commun.*, vol.3, pp.850-1854, Sep. (1995).
- [40] S. Hurley and D. H. Smith: "Fixed spectrum frequency assignment using natural algorithms," *Proc. of Genetic Algorithms in Engineering Systems: Innovations and Applications*, pp.373-378, IEE, Sep. (1995).
- [41] M. Cuppini: "A genetic algorithm for channel assignment problems," *Eur. Trans. Telecommun.*, vol.5, no.2, pp.285-294, Mar. (1994).
- [42] 安部智, 渡辺郁, 田村裕, 仙石正和, 篠田庄司: 「遺伝的アルゴリズムを用いたセルラ移動通信系におけるダイナミックチャンネル割当に関する一考察」, *信学技報 CAS94-52*, pp.13-20, Sep. (1994).
- [43] C. Y. Ngo and V. O. K. Li: "Fixed channel assignment in cellular radio networks using a modified genetic algorithm," *IEEE Trans. Veh. Technol.*, vol.47, no.1, pp.163-172, Feb. (1998).
- [44] 村上誉, 小川恭孝, 大鐘武雄: 「遺伝的アルゴリズムを用いた移動通信における固定チャンネル割当法」, *信学会論 (B)*, vol. J83 - B, no.6, pp.769-779, June (2000).
- [45] W. Crompton, S. Hurley, and N.M. Stephens: "Applying genetic algorithms to frequency assignment problems, *Proc. SPIE Conf.*,

- Neural and Stochastic Methods in Image and Signal Processing, vol.2304, pp.76-84, Jul. (1994).
- [46] K. W. Lai and G. G. Coghill: "Channel assignment through evolutionary optimization," *IEEE Trans. Veh. Technol.*, vol.45, no.1, pp.91-96, Feb. (1996).
- [47] K. I. Aardal, H. Hipolito, C. van Hoesel, B.Jansen, C. Ross and T. Terlaky: "Implementation and Testing of Polyhedral Techniques and Interior Point Methods", EUCLID CALMA Radio Link Frequency Assignment Project Report 2.2.1, 1995, [ftp://ftp.win.tue.nl/pub/techreports /CALMA/221.ps.gz](ftp://ftp.win.tue.nl/pub/techreports/CALMA/221.ps.gz).
- [48] K. I. Aardal, H. Hipolito, C. van Hoesel, B.Jansen, C. Ross and T. Terlaky: "A Branch-and Cut Algorithm for the Frequency Assignment Problem," EUCLID CALMA Radio Link Frequency Assignment Project Technical Annex T-2.2.1 A: (1995), <ftp://ftp.win.tue.nl/pub/techreports /CALMA/T221A.ps.gz>.
- [49] A. Kolen and S. van Hoesel: "A Constraint Satisfaction Approach to the Radio Link Frequency Assignment Problem", EUCLID CALMA Radio Link Frequency Assignment Project Report 2.2.2, (1995), <ftp://ftp.win.tue.nl/pub/techreports/CALMA/222.ps>.
- [50] 横尾真, 平山勝敏: 「制約充足テクニックを用いた移動体通信の周波数割当問題の解法」, *情報処理学会論文誌*, vol.41, no.4, pp.1234 – 1243, April (2000).
- [51] 松井正一, 所健一: 「割当て順序表現を用いる遺伝的アルゴリズムによる移動無線における固定チャンネル割当ての高速解法」, *信学会論 (B)*, vol.J83-B, no.5, pp.645 – 653, May (2000).
- [52] 所健一, 松井正一, 西野能啓, 橋口英夫: 「GAによる移動無線基地局への周波数割当ての最適化」, *電気学会通信研究会資料 CMN-00-06*, pp.29 – 34, January (2000).
- [53] S. Matsui and K. Tokoro: "A new genetic algorithm for minimum span frequency assignment using permutation and clique," in *Proc. Of GECCO-2000*, pp.682 – 689, July (2000).

- [54] A. Cohen, and S. Wan: A method for solving the fuel constrained unit commitment problem, *IEEE Transactions on Power Systems*, **2**, 608–614 (1987).
- [55] A. Merlin, and P. Sandrin: A new method for unit commitment at electricite de france, *IEEE Transactions on Power Application Systems*, **PAS-102**, 1218–1225 (1983).
- [56] F. Lee: A fuel constrained unit commitment method, *IEEE Transactions on Power Systems*, **4**, 691–698 (1989).
- [57] W. Ongsakul, and N. Petcharaks: Unit commitment by enhanced adaptive lagrangian relaxation, *IEEE Transactions on Power Systems*, **19-1**, 620–628 (2004).
- [58] S. Virmani, C. Adrian, K. Imhof, and S. Mukherjee: Implementation of a lagrangian relaxation based unit commitment problem, *IEEE Transactions on Power Systems*, **4**, 1373–1380 (1989).
- [59] F. Zhuang, and F. Galiana: Toward a more rigorous and practical unit commitment by lagrangian relaxation, *IEEE Transactions on Power Systems*, **3**, 763–770 (1988).
- [60] I. Damousis, A. Bakirtzis, and P. Dokopoulos: A solution to the unit-commitment problem using integer-coded genetic algorithm, *IEEE Transactions on Power Systems*, **19-2**, 1165–1172 (2004).
- [61] K. Juste, H. Kita, E. Tanaka, and J. Hasegawa: An evolutionary programming solution to the unit commitment problem, *IEEE Transactions on Power Systems*, **14-4**, 1452–1459 (1999).
- [62] S. Kazarlis, A. Bakirtzis, and V. Petridis: A genetic algorithm solution to the unit commitment problem, *IEEE Transactions on Power Systems*, **11-1**, 83–90 (1996).
- [63] Z. Ouyang, and S. Shahidehpour: A multi-stage intelligence system for unit commitment, *IEEE Transactions on Power Systems*, **7**, 639–646 (1992).

- [64] H. Sasaki, M. Watanabe, and R. Yokoyama: A solution method of unit commitment by artificial neural networks, *IEEE Transactions on Power Systems*, **7**, 974–981 (1992).
- [65] K. Swarup, and S. Yamashiro: Unit commitment solution methodology using genetic algorithm, *IEEE Transactions on Power Systems*, **17-1**, 87–91 (2002).
- [66] C. Wang, and S. Shahidehpour: A decomposition approach to nonlinear multi-area generation scheduling with tie-line constraints using expert systems, *IEEE Transactions on Power Systems*, **7**, 1409–1418 (1992).
- [67] W. Xing, and F. Wu: Genetic algorithm based unit commitment with energy contracts, *Electrical Power and Energy Systems*, **24**, 329–336 (2002).
- [68] F. Zhuang, and F. Galiana: Unit commitment by simulated annealing, *IEEE Transactions on Power Systems*, **5**, 311–318 (1990).
- [69] C. Cheng, C. Liu, and C. Liu: Unit commitment by lagrangian relaxation and genetic algorithms, *IEEE Transactions on Power Systems*, **15-2**, 707–714 (2000).
- [70] T. Senjyu, K. Shimabukuro, K. Uezato, and T. Funabashi: A fast technique for unit commitment problem by extended priority list, *IEEE Transactions on Power Systems*, **18-2**, 882–888 (2003).
- [71] 高田亨, 高橋順一, 横井宏臣, 中野宏, 青柳真理, 加藤政一, 島田和恵, 荒井純一: 「遺伝的アルゴリズムと数理計画法を融合した発電機起動停止計画問題の解法」, *電学論*, Vol. 119-B, No. 3, pp. 333–343 (1999).
- [72] H. Mashhadi, H. Shanechi and C. Lucas: A New Genetic Algorithm With Lamarckian Individual Learning for Generation Schedule, *IEEE Trans. Power Systems*, Vol. 18, No. 3, pp. 1181–1186 (2003).

- [73] H. Yamashita: A globally convergent primal-dual interior point method for constrained optimization, *Optimization Methods and Software*, **10**, 443–469 (1998).
- [74] H. Yamashita, and H. Yabe: Superlinear and quadratic convergence of some primal-dual interior point methods for constrained optimization, *Mathematical Programming*, **75**, 377–397 (1996).
- [75] H. Yamashita, H. Yabe, and T. Tanabe: A globally and superlinearly convergent primal-dual interior point trust region method for large scale constrained optimization, *Mathematical Programming Ser.B*, **102–1**, 111–115 (2005).
- [76] B.J. Ritzel, J.W. Eheart and S. Ranjithan: Using genetic algorithms to solve a multiple objective groundwater pollution containment problem, *Water Resources Research*, Vol. 30, No. 5, pp. 1589–1603 (1994).
- [77] S.E. Cieniawski, J.W. Eheart and S. Ranjithan: Using genetic algorithms to solve multiobjective groundwater monitoring problem, *Water Resources Research*, Vol. 31, No. 2, pp. 399–409 (1995).
- [78] D.H. Loughlin and S. Ranjithan: Chance-Constrained Genetic Algorithms, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, Vol. 1, Morgan Kaufmann, Orlando, Florida, pp. 369–376 (1999).
- [79] L. Painton and J. Chambell: Genetic Algorithms in Optimization of System Reliability, *IEEE Transactions on Reliability*, Vol. 44, No. 2, pp. 172–178 (1995).
- [80] Y. Yoshitomi, H. Ikenoue, T. Takeba and S. Tomita: Genetic Algorithm in Uncertain Environments for Solving Stochastic Programming Problem, *Journal of the Operations Research Society of Japan*, Vol. 43, No. 2, pp. 266–290 (2000).
- [81] F.V. Louveaux and D. Peeters: A dual-based Procedure for stochastic facility location, *Operations Research*, Vol. 40, No. 3, pp. 564–573 (1992).

- [82] G. Laporte, F.V. Louveaux and L.V. Hamme: Exact Solution to a Location Problem with Stochastic Demand, *Transportation Science*, Vol. 28, No. 2, pp. 95–103 (1994).
- [83] A.A. Kuehn and M.J. Hamburger: A Heuristic Program for Locating Warehouses, *Management Science*, Vol. 9, pp. 643–666 (1963).
- [84] 所, 藤井, 藤岡, 森永, 高橋: 「ガスタービン高温部品管理支援システムの開発—発見的手法と GA の組合せによる最適部品ローテーション計画手法—」, 電力中央研究所研究報告, R03012, (2004)
- [85] 所健一, 渡邊勇: 「ユーザサイドからの信頼性評価に基づいた情報システムの最適設計手法」, 電力中央研究所研究報告, R00021, (2001)

謝辞

本研究の過程において、終始懇切な御指導と御鞭撻を賜り、本論文をまとめるにあたり親身な御指導と励ましを頂いた、慶應義塾大学 西野寿一名誉教授、慶應義塾大学理工学部管理工学科 増田靖教授に心より感謝を申し上げます。

本論文の審査過程において、数々の御助言と御指導を賜りました、慶應義塾大学理工学部管理工学科 櫻井彰人教授、首都大学東京大学院理工学研究科 安田恵一郎教授に深謝申し上げます。

また、これまでの研究過程において、数々の有益なご助言を頂いた、財団法人電力中央研究所 システム技術研究所 松井正一情報システム領域リーダー、篠原靖志システム最適化グループリーダー、渡邊勇主任研究員をはじめとする、電力中央研究所の皆様には感謝いたします。