平成26年度　修士論文

探索点間の相互作用を考慮した
近傍探索に基づく多目的組合せ最適化手法

首都大学東京大学院　理工学研究科　電気電子工学専攻

高村 秋平

論文要旨

近年のシステムの大規模化・複雑化，システムの設計・運用・解析・制御や工業製品の性能に対する要求の高度化により，実用的な最適化に対する要求は急速に高まっている。その中でも，システム運用のスケジュール問題や，最短経路問題をはじめとする，多くの離散的構造を有する実問題は組合せ最適化問題として定式化できる。しかしながら，多くの組合せ最適化問題は，問題のサイズに対して計算時間が指数関数的に増大し，実用的な時間内で計算することが困難な，いわゆる $\mathcal{NP}$ 困難のクラスに属すことが知られている。一方で，実際の問題を解く際には，厳密な最適性の保証はなくとも，十分精度の高い近似解を実用的な時間内で求めることができれば十分である場合が多い。

さらに，実問題においては目的関数が複数あり，また目的関数間に優先順位をつけることが難しく，全て良くしたいと考えることがある。このような問題は多目的最適化問題と呼ばれる。通常，多目的最適化問題は複数の目的関数の間にトレードオフの関係があり，これを考慮した解の集合を効率よく求めることができる手法の必要性が増してきている。

以上より，本研究では，多目的組合せ最適化問題を対象とした，実用的な最適化手法の構築を目標とした。最適化手法として，メタヒューリスティクスに着目した。メタヒューリスティクスは，決定変数値情報と評価値情報のみで最適化ができる発見的近似手法の枠組みである。汎用性の高さや，探索時間に応じた最適性の高さを持つ解を得ることが可能である点が特徴として挙げられ，コンピュータパワーの飛躍的増大を背景として，実用的な最適化手法として注目が集まっている。

メタヒューリスティクスの探索の本質は，探索点の近傍の生成と，探索点の移動であると考えることができる。本研究では，探索点の近傍の生成について，Tabu Search をはじめとする近傍探索に基づく手法に着目した。近傍探索に基づく手法は，単一目的組合せ最

適化問題に対して高い性能を有することが知られており，多目的組合せ最適化問題に対しても有効な戦略であることが期待できる。

Tabu Search を多目的最適化問題に対して適用した例として，加重和最小化手法や制約変換法などにより，多目的最適化問題を単一目的最適化問題に変換して適用する方法が挙げられる。これに対し，本研究では，多目的組合せ最適化問題を変換せずに扱う，近傍探索に基づく多点探索型多目的組合せ最適化手法を提案した。本研究の要点は，以下の通りである。

(1) Tabu Search に立脚した多点探索型多目的組合せ最適化手法を提案した。提案手法において，探索点の近傍生成およびその際に用いる探索履歴情報は，代表的な近傍探索に基づく単一目的組合せ最適化手法である Tabu Search に基づくものとした。また，探索点の移動は，進化型多目的最適化の代表的手法である Nondominated Sorting Genetic Algorithm II（NSGA-II）を参考に，探索点間の相互作用を用いる移動方法を新たに提案した。典型的なベンチマーク問題を用いた数値実験により，提案手法の有用性を検証した。

(2) メタヒューリスティクスの探索における代表的な探索戦略として，多様化と集中化が挙げられる。あらゆるメタヒューリスティクスにおいて，探索性能の向上には探索過程における適切な多様化・集中化の実現が重要であることが広く知られている。また，多くの優れたメタヒューリスティクスにおいて，探索序盤は多様化を強く働かせ，探索終盤は集中化を強く働かせる戦略が見られる。(1) で提案した Tabu Search に立脚した多点探索型多目的組合せ最適化手法において，多様化・集中化のバランスを調節するパラメータを明らかにし，そのパラメータを探索段階に合わせてスケジューリングすることで，多様化と集中化の適切なバランスの実現を狙う新たな移動戦略を提案した。典型的なベンチマーク問題を用いた数値実験により，新たな移動戦略の有用性を検証した。

(3) 最適化問題の解構造に広く見られる良構造の偏りとして，近接最適性原理が知られている。近接最適性原理は，「良い解同士はなんらかの類似構造を有している」という漠然とした原理である。本研究では，良い解を評価値，類似構造を探索点間の距離で評価した上で，多目的最適化問題における近接最適性原理を新たに解釈した。これを基に，探索点の近傍生成および探索点の移動において，評価値情報と距離情報に基づく探索点間の相互作用を有する，新たな多点探索型多目的組合せ最適化手

法を提案した。数値実験を通じて，提案手法の有用性を検証した。

目次

1 序論

1.1 本論文の背景と目的

近年のシステムの大規模化・複雑化，システムの設計・運用・解析・制御や工業製品の性能に対する要求の高度化により，実用的な最適化に対する要求は急速に高まっている。その中でも，システム運用のスケジュール問題や，最短経路問題をはじめとする，多くの離散的構造を有する実問題は，組合せ最適化問題として定式化できる。しかしながら，多くの組合せ最適化問題は，問題のサイズに対して計算時間が指数関数的に増大し，実用的な時間内で計算困難な，いわゆる $\mathcal{NP}$ 困難のクラスに属すことが知られている[1]。一方で，実際の問題を解く際においては，厳密な最適性の保証はなくとも十分精度の高い近似解を実用的な時間内で求めることができれば十分である場合が多い[2]。これらに加えて，コンピュータパワーの飛躍的増大を背景として，近似手法の一つであるメタヒューリスティクスと呼ばれる枠組みに注目が集まっている。

さらに，実問題においては目的関数が複数あり，また，目的関数間に優先順位をつけることが難しく，それらを全て良くしたいと考えることがある。このような問題は多目的最適化問題と呼ばれる。通常，多目的最適化問題は複数の目的関数の間にトレードオフの関係があり，これを考慮した解の集合を効率よく求めることができる手法の必要性が増してきている。

近年，多目的最適化問題においては，代表的なメタヒューリスティクスのひとつである遺伝的アルゴリズム（Genetic Algorithm: GA）[3]をはじめとする進化的手法を用いたアプ

ローチである進化型多目的最適化（Evolutionary Multi-Objective Optimization: EMO）が注目を集めている[4]。この理由として，進化的手法は多点探索型手法であり，一度の探索で多様な解を効率的に求められることが挙げられる。

一方，本研究では，EMO のアプローチを意識しつつも，最適化手法として Tabu Search（TS）[5]をはじめとする近傍探索に基づく手法に着目する。近傍探索に基づく手法は，単一目的最適化問題において高い性能を有することが知られており，多目的最適化問題においても有用であることが期待できる。

本研究では，まず，近傍生成は TS に立脚し，探索点の移動において探索点間の新たな相互作用を用いた，多点探索型多目的組合せ最適化手法を提案する。代表的なベンチマーク問題を用いた数値実験を通して，提案した手法の有用性を検証する。

さらに，提案した TS に基づく多点探索型多目的最適化手法について，メタヒューリスティクスの探索における代表的な探索戦略である多様化と集中化[1]の観点から解析を行う。その結果を踏まえて，パラメータのスケジューリングを用いることで探索の多様化・集中化を制御し，探索序盤における解空間の多様な探索と，探索終盤における良い解周辺の集中的な探索を狙う新たな移動戦略を提案する。代表的なベンチマーク問題を用いた数値実験を通して，提案した移動戦略の有用性を検証する。

最後に，最適化問題の解構造に広く見られる良構造の偏りである近接最適性原理（Proximate Optimality Principle: POP）[1]について，多目的最適化問題における解釈を新たに行う。多目的最適化における POP を基に，探索点の近傍生成および探索点の選択・移動において，評価値情報と距離情報に基づく探索点間の相互作用を有する，新たな多点探索型多目的組合せ最適化手法を提案する。代表的なベンチマーク問題を用いた数値実験を通して，提案手法の有用性を検証する。

1.2　本論文の構成

本論文は全 7 章から構成されており，各章の概要は以下の通りである。

第 1 章の “序論” では，本研究の背景と目的について述べ，本研究の要点をまとめた。

第 2 章の “多目的最適化問題” では，多目的最適化問題の概要を述べる。

第 3 章の “近傍探索に基づく最適化手法” では，代表的な近傍探索に基づく最適化手法と，それらの多目的最適化への適用について述べる。

第 4 章の “Tabu Search に基づく多点探索型多目的最適化手法の提案” では，近傍生成は TS に基づき，選択・移動については，新たに提案した探索点間の相互作用を用いた，多点探索型多目的組合せ最適化手法を提案する。代表的なベンチマーク問題を用いた数値実験により，TS に基づく多点探索型多目的最適化手法の有用性を検証する。

第 5 章の “多様化と集中化を考慮した新たな移動戦略の提案” では，第 4 章で提案した多点探索型多目的最適化手法について，メタヒューリスティクスの探索における代表的な探索戦略である多様化と集中化の観点から解析を行う。さらに，解析に基づき，新たな移動戦略を提案する。代表的なベンチマーク問題を用いた数値実験により，提案した移動戦略の有用性を検証する。

第 6 章の “近接最適性原理に基づく多点探索型多目的最適化手法の提案” では，最適化問題の解構造に広く見られる良構造の偏りである近接最適性原理について述べ，多目的最適化問題における近接最適性原理を新たに解釈する。さらに，多目的最適化問題における近接最適性原理に基づく探索点間の相互作用を有する，多点探索型多目的組合せ最適化手法を提案する。代表的なベンチマーク問題を用いた数値実験により，提案した移動戦略の有用性を検証する。

第 7 章の “結論” では，本研究で得られた研究成果，および今後の研究課題をまとめる。

付録 A では，本論文の数値実験に用いた組合せ最適化問題について説明する。

2 多目的最適化問題

2.1 多目的最適化問題の概要

実問題においては，目的関数が複数あり，また，目的関数間に優先順位をつけることが難しく，それらをすべて良くしたいと考えることがしばしばある。このような，複数の目的関数を持つ最適化問題を，多目的最適化問題と呼ぶ。一般に，r 個の目的関数を持つ多目的最小化問題は以下のように定式化される。

$$\min \quad \boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ \vdots \\ f_r(\boldsymbol{x}) \end{bmatrix}$$

$$\text{subj. to} \quad \boldsymbol{x} \in \mathcal{F}$$

ここで，$\boldsymbol{x}$ は決定変数，$\boldsymbol{f}(\boldsymbol{x})$ はベクトル値目的関数，$f_i(\boldsymbol{x})$ は i 番目の目的関数，$\mathcal{F}$ は実行可能領域である。図 2.1 に，2 目的最小化問題のイメージを示す。

もし，すべての目的関数を最小にする解が存在するならば，その解は多目的最小化問題の完全最適解である。しかし，通常は複数の目的関数の間にトレードオフの関係が存在するために，完全最適解が存在することは一般的ではない[1]。そのため，多目的最適化においては，個々の解同士の優越関係に基づいて解を評価する。

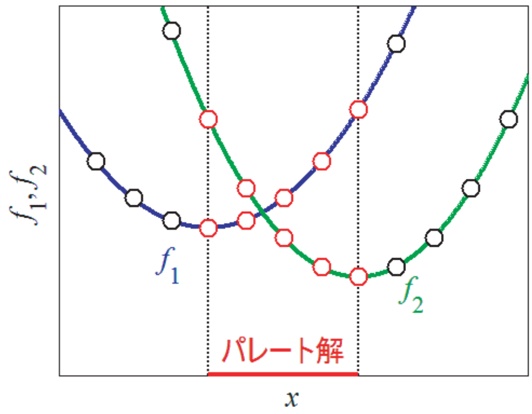


図 2.1： 多目的最適化問題

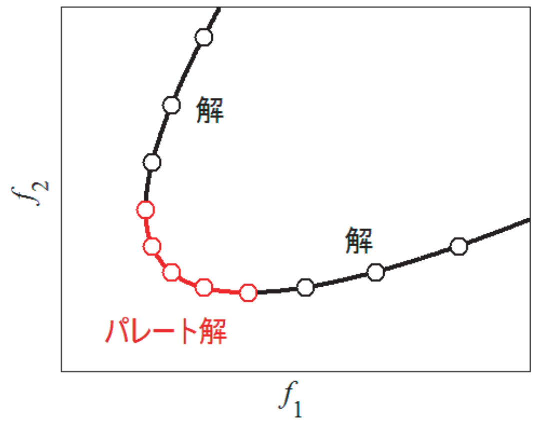


図 2.2： 多目的最適化問題の評価値空間

優越関係

$\boldsymbol{x}$ と $\boldsymbol{y}$ を多目的最小化問題の実行可能解とする。以下の条件が成り立つとき，解 $\boldsymbol{x}$ は解 $\boldsymbol{y}$ を優越するという。

$$^{\forall}i,\ f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y}) \quad \text{and} \quad ^{\exists}i,\ f_i(\boldsymbol{x}) < f_i(\boldsymbol{y})$$

多目的最適化では，ほかの解に優越されないという基準を用いて最適解が定義される。このような解はパレート解（Pareto Solution），あるいは非劣解などと呼ばれる [6]。パレート解同士はそれぞれ優越関係による客観的な比較が不可能な解であり，最終的な選好解の候補である。図 2.2 は，図 2.1 の 2 目的最小化問題を評価値空間で表したものである。

2.2　単一目的最適化問題への変換を用いた解法

多目的最適化問題に対して古くから用いられている方法として，多目的最適化問題を単一目的最適化問題に変換したうえで解く方法が挙げられる。この方法はスカラー化と呼ばれる。スカラー化に分類される方法としては，加重和最小化手法や，制約変換法（ε 制約法）などが挙げられる。

このとき，各手法について，意思決定者の選好情報が事前にある形で与えられる必要があると言える。例えば，加重和最小化手法の場合，各目的関数の重みが必要である。しかし，特に目的関数が多くなった場合，重みを調整することによって望み通りの解を得ることは一般には困難であることが知られている[6]。また，制約変換法の場合，各目的関数に関して許容できる最大値が必要となるが，このためには目的関数間に明確な優先順位をつけることができなければならない。

さらに，加重和最小化手法については，非凸なパレート解を得ることができないといった課題も知られている[1，6]。

2.3　進化型多目的最適化

近年注目されている多目的最適化問題に対するアプローチに，遺伝的アルゴリズム（Genetic Algorithm: GA）をはじめとした進化型計算を用いた方法である進化型多目的最適化（Evolutionary Multi-Objective Optimization: EMO）がある。EMO の特徴として，多目的最適化問題を他の問題に変換することなく扱い，アルゴリズムを一回実行することで多数のパレート解を同時に得ることを明確に意図していることが挙げられる。進化型計算は多点探索型手法であるため，複数の解を探す多目的最適化には都合が良いと言える[4]。

2.3.1　メタヒューリスティクス

GA をはじめとした進化的最適化手法や，Local Search [7] や Tabu Search をはじめとする近傍探索に基づく最適化手法，Particle Swarm Optimization [8] などのマルチエージェント型最適化手法は，メタヒューリスティクスと呼ばれる枠組みに分類される。メタヒューリスティクスは，最適化問題に対する発見的近似手法の枠組みであり，以下のような特徴を持つ。

- 物理現象，生物現象などにアナロジーを持つ手法が多い
- 計算時間に応じて最適性の高い解を得ることが可能
- 決定変数値情報と解情報のみで最適化が可能

- 探索点の近傍生成と，選択・移動を繰り返すことで最適化を行う

特に，決定変数値情報と解情報のみで最適化が可能であることから，シミュレータからの評価値情報を直接用いた最適化などが可能である。近年のコンピュータパワーの飛躍的増大や，シミュレーション技術の進歩を背景に，汎用性が高く，実用的な最適化手法として注目を集めている。

2.3.2　進化型多目的最適化の代表的手法

EMO に分類される最適化手法に，Deb らが提案した Nondominated Sorting Genetic Algorithm II（NSGA-II）[1, 9]がある。NSGA-II は, GA において特徴的な交叉・突然変異といった遺伝的操作を用いた近傍生成に加えて，解の選択において Crowding Tournament 選択と呼ばれる評価方法が導入されている手法であり，高い性能と汎用性から，代表的な EMO の最適化手法として知られている。特徴として，強力なエリート保存戦略が実現されていること，パラメータの数が少数であることなどが挙げられる。

NSGA-II において採用されている Crowding Tournament 選択は, Non-Dominated Sorting および Crowding Distance Sorting から構成される，解の評価方法である。以下に，それぞれの概要を述べる。

Non-Dominated Sorting

Non-Dominated Sorting は, 対象となるそれぞれの解に重要度を示すランク（Rank）を割り振り，ランクの値が良い順に並び替えるソートの方法である。本研究では，ランクを割り振ることをランキング（Ranking）と呼ぶ。ランキングは，Goldberg の示唆した方法[3, 6]を用いる。

ランキングの手順は以下の通りである。まず，対象とする解集合のうち，ほかの解に優越されていない解にランク 1 を割り振る。そして，ランクが割り振られていない解のみを考え，ほかの解に優越されていない解にランク 2 を割り振る。以下，同様にして全ての解にランクを割り振る。

図 2.3 に 2 目的最小化問題におけるランクの例を示す。

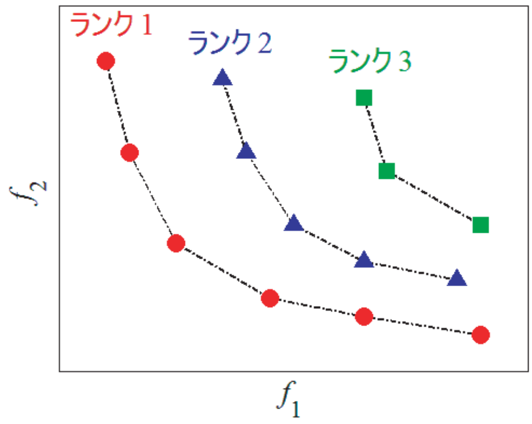


図 2.3： 解とランクの関係

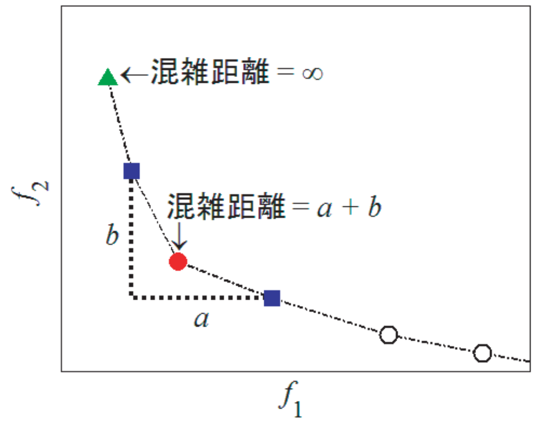


図 2.4： 混雑距離

Crowding Distance Sorting

Crowding Distance Sorting は，ランクが等しい解によって構成される集合について，混雑距離（Crowding Distance）が降順になるように並び替えるソートの方法である。ランクが等しい解の集合において，混雑距離は以下のように求める。

- **少なくともひとつの目的関数において最大値または最小値となる解**
  対象とする解の混雑距離は無限大とする。
- **その他の解**
  対象とする解について，個々の目的関数ごとに両隣に位置するふたつの解の距離を計算する。全ての目的関数に関する距離の総和を，対象とする解の混雑距離とする。

図 2.4 に 2 目的最適化問題における混雑距離の求め方を示す。

NSGA-II の選択において，ランクが高い解を上位とし，また，同一ランクの解の中でも混雑距離が大きい解を上位とする。上位の解から順に次世代の解として選択することで，よりパレート解に近い解の獲得ができる。さらに，混雑距離を用いることにより，同一ランクの解の中でも各目的関数について最適である解が上位となり，また他の解に似た評価値を持つ解は下位に位置づけられるため，より広がりのある解を得ることや，より広がりを持った近傍生成が可能となる。

本論文で用いた NSGA-II のアルゴリズムを以下に示す。

【Nondominated Sorting Genetic Algorithm II のアルゴリズム】

Step 0 :[準備]　探索点数 m，最大反復回数 $k_{\max}$，交叉率 P_{c}，突然変異率 P_{m} を与える。初期解 $\boldsymbol{x}_i^0 (i = 1, 2, \cdots, m)$ を与える。反復回数 $k := 0$ とする。

Step 1 :[親個体群の評価]　親個体群の集合 $\boldsymbol{Par}^k = \left\{ \boldsymbol{x}_i^k \mid i = 1, 2, \cdots, m \right\}$ とする。$\boldsymbol{Par}^k$ を対象に，Non-Dominated Sorting および Crowding Distance Sorting を行い，各解の優先順位を求める。

Step 2 :[子個体群の生成]　子個体群の集合 $\boldsymbol{Off}^k = \varnothing$ とする。Step 2-1 から Step 2-3 の操作を m 回繰り返す。

Step 2-1 :[親個体の選択]　$\boldsymbol{Par}^k$ からランダムに解を 2 個選び，優先順位の高い方の解を $\boldsymbol{u}_1$ とする。同様に，$\boldsymbol{u}_2$ を決定する。

Step 2-2 :[交叉・突然変異]　$\boldsymbol{u}_1$ と $\boldsymbol{u}_2$ に対して，確率 P_{c} に従って交叉の操作を行う。その後，確率 P_{m} に従って突然変異の操作を行う。操作後の個体をそれぞれ $\boldsymbol{u}_1'$，$\boldsymbol{u}_2'$ とする。

Step 2-3 :[子個体群への追加]　$\boldsymbol{Off}^k := \boldsymbol{Off}^k \cup \{\boldsymbol{u}_1', \boldsymbol{u}_2'\}$ とする。

Step 3 :[選択・移動]　$\boldsymbol{Par}^k \cup \boldsymbol{Off}^k$ の解集合を対象に，Non-Dominated Sorting および Crowding Distance Sorting を行い，各解の優先順位を求める。優先順位が上位となる m 個の解を，$\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_2^{k+1}, \cdots, \boldsymbol{x}_m^{k+1}$ とする。

Step 4 :[終了判定]　$k := k + 1$ とする。$k = k_{\max}$ ならば，探索を終了。さもなければ，Step 1 へ行く。

2.3.3　進化型多目的最適化手法の性能評価

EMO に分類される多目的最適化手法の性能を定量的に評価するために，様々な評価指標が提案されている。一般には，多目的最適化手法によって得られた解集合を用いて，解の正確性，解の広がり，解の分布，および計算時間から評価を行う [6]。主な評価指標に，Hypervolume，Hypervolume Ratio，Generational Distance，γ 評価指標などがある。以下に，それぞれの評価指標の概要を述べる。

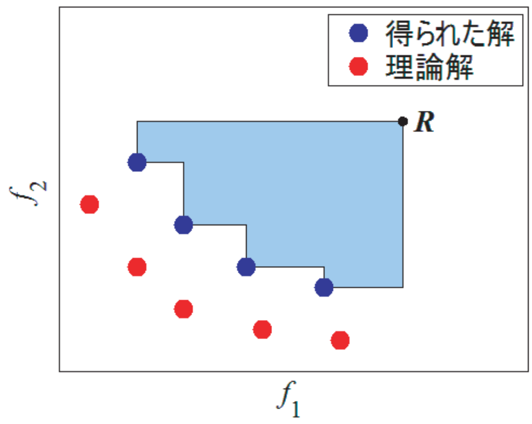


図 2.5： Hypervolume

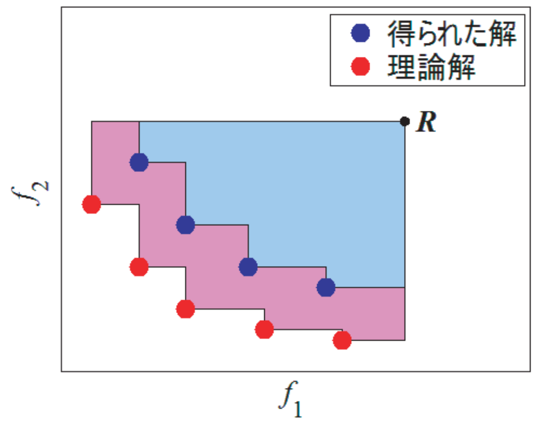


図 2.6： Hypervolume Ratio

Hypervolume

Hypervolume は，解の正確性と広がりを評価する指標の一つであり，得られたパレート解によって支配される目的関数空間の領域の超体積の大きさを指す。例えば，2 目的最適化問題の場合，Hypervolume は目的関数空間において解集合が支配する面積となる。図 2.5 に，2 目的最小化問題における Hypervolume を示す。Hypervolume は，解が正確であるほど大きい値となり，また解の広がりが十分であるほど大きい値となる。

評価指標としての Hypervolume の長所として，理論解集合が未知の場合でも得られた解を評価できることが挙げられる。一方，Hypervolume の問題点としては，算出するためには事前に基準点 $\boldsymbol{R} = (R_1, R_2, \cdots, R_r)$ を定める必要があること，基準点 $\boldsymbol{R}$ とパレート解が離れすぎているときに結果の差を数値として表しにくいこと，各目的のスケールを考慮していないことなどが挙げられる。

Hypervolume Ratio

Hypervolume Ratio は，得られた解集合の Hypervolume を，理論解集合の Hypervolume で正規化したものである。Hypervolume Ratio の値は 0 から 1 となり，解が正確である，または解の広がりが十分であるほど 1 に近づく。図 2.6 に，2 目的最小化問題における Hypervolume Ratio を示す。

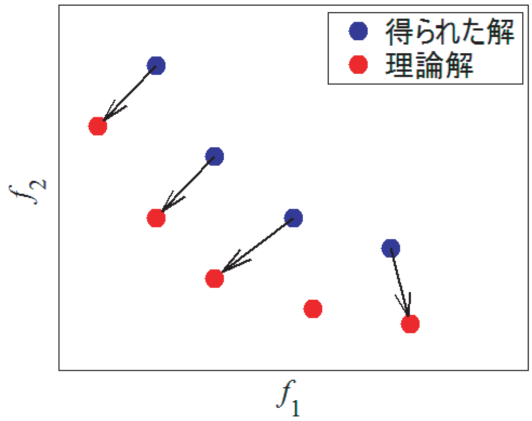


図 2.7：γ 評価指標

Generational Distance

Generational Distance は，得られた解集合から理論解集合までの距離に基づいた評価指標である。得られた解集合を $\mathcal{S}$，理論解集合を $\mathcal{P}$ とすると，Generational Distance（$GD(\mathcal{S}, \mathcal{P})$）は以下の式で与えられる。

$$GD(\mathcal{S}, \mathcal{P}) = \frac{\left(\sum_{i=1}^{|\mathcal{S}|} d_i^q\right)^{1/q}}{|\mathcal{S}|}$$

ここで，

$$d_i = \min_{\boldsymbol{p} \in \mathcal{P}} \sqrt{\sum_{k=1}^{n} \left(f_k(\boldsymbol{s}_i) - f_k(\boldsymbol{p})\right)^2}$$

である。また，$|\mathcal{S}|$ は得られた解集合 $\mathcal{S}$ に含まれる解の個数である。解が正確であるほど Generational Distance は小さい値となる。

γ 評価指標

$q = 1$ のときの Generational Distance は，特に γ 評価指標（Convergence Metric γ）と呼ばれる [6, 9]。解が正確であるほど γ 評価指標は小さい値となる。図 2.7 に，2 目的最小化問題における γ 評価指標のイメージを示す。

3 近傍探索に基づく最適化手法

3.1 近傍の定義

近傍 $\mathcal{N}$ は，次の写像

$$\mathcal{N} : \mathcal{F} \to 2^{\mathcal{F}}$$

によって定義される。この写像は，ある解 $\boldsymbol{x}$ に対して，解空間の任意の部分集合を充てることができる。

一般的に，メタヒューリスティクスにおいて解 $\boldsymbol{x}$ の近傍 $\mathcal{N}(\boldsymbol{x})$ は，解 $\boldsymbol{x}$ に対して何らかの意味で近い，似通った構造の解集合と位置づけられる。実際には，解 $\boldsymbol{x}$ になんらかの局所的な摂動を加えることにより得ることができる解の集合として，問題ごとに定義される。

以下に，主な組合せ最適化問題に対する，本研究で用いた解表現と近傍の定義を挙げる。

0-1 ナップサック問題

0-1 ナップサック問題（0-1 Knapsack Problem: 0-1KP）において，解 $\boldsymbol{x}$ を 0-1 表現によって表す。

解が二値ベクトルとして表現される問題に対する近傍生成方法として，λ bit 反転近傍[7, 10]が挙げられる。これは，二値ベクトル $\boldsymbol{x}$ に対して，λ 個の要素を反転させることによって得られるベクトルの集合である。本研究では，0-1KP の近傍として，1bit 反転近傍を採用する。

巡回セールスマン問題

巡回セールスマン問題（Traveling Salesman Problem: TSP）において，解 $\boldsymbol{x}$ を順序表現によって表す。

TSP に対する近傍生成方法として，一般的に λ-opt 近傍[7, 10]が用いられる。これは，巡回路 $\boldsymbol{x}$ の辺集合のうち，λ 本の枝を入れ替えることにより得られる巡回路の集合である。本研究では，TSP の近傍として，2-opt 近傍を採用する。

フローショップ・スケジューリング問題および二次割当問題

フローショップ・スケジューリング問題（Flow-Shop Scheduling Problem: FSP）と，二次割当問題（Quadratic Assignment Problem: QAP）において，解 $\boldsymbol{x}$ を順序表現によって表す。

解が順列として表現される問題に対する近傍生成方法として，交換近傍[7, 10]がある。これは，順列 $\boldsymbol{x}$ に対して，二つの要素の位置を交換することによって得られる順列の集合である。本研究では，FSP と QAP の近傍として，交換近傍を採用する。

3.2　代表的な近傍探索に基づく最適化手法

近傍探索に基づく手法としては，Local Search（LS，局所探索法）や Tabu Search（TS）などが挙げられる。

3.2.1　Local Search

LS は，現在の探索点 $\boldsymbol{x}$ から，何らかの変化を加えた解の集合である近傍 $\mathcal{N}(\boldsymbol{x})$ を生成し，近傍内のある改善解 $\boldsymbol{x}'$ への移動を繰り返すことによって，確実に局所解を求める手法である。このとき，改善解 $\boldsymbol{x}'$ の選択方法として，以下の戦略が挙げられる。

最良移動戦略　近傍内で最善となる解に移動する方法

即時移動戦略　近傍内の解をランダムに探索し最初に見つかった改善解に移動する方法

単一目的最小化問題における単点 LS のアルゴリズムを以下に示す。

【Local Search のアルゴリズム】

Step 0 :[**準備**]

初期解 $\boldsymbol{x}$ を与える。

Step 1 :[**終了判定**]

$\mathcal{N}(\boldsymbol{x})$ の解から，$f(\boldsymbol{x}') < f(\boldsymbol{x})$ となる解 $\boldsymbol{x}'$ を 1 個選ぶ。$\boldsymbol{x}'$ が存在しなければ，探索を終了。

Step 2 :[**解の移動**]

$\boldsymbol{x} := \boldsymbol{x}'$ とし，Step 1 へ行く。

3.2.2　Tabu Search

TS は，LS の枠組みを基に，近傍内に改善解が存在しない場合においても移動し続ける手法である。さらに，TS では，探索点が既に探索した解に戻るサイクリングを防ぐためにタブー（Tabu）と呼ばれる制約を利用する。これは，タブーリスト（Tabu List）に過去の移動または探索済みの解を逐次記憶し，一定の期間内に起こった移動に逆行するような移動を禁止するものである。以上より，TS は，近傍探索による集中的な探索に加えて，探索済みの解に探索点が戻らないよう探索履歴情報を用いて近傍生成に制限をかけることで解空間を多様に探索することが期待できる手法であると考えられる。

単一目的最小化問題における単点 TS のアルゴリズムを以下に示す。

【Tabu Search のアルゴリズム】

Step 0 :[**準備**]

タブーリスト長 L_{T}，最大反復回数 $k_{\max}$ を与える。初期解 $\boldsymbol{x}^0$ を与える。タブーリスト $\mathcal{T}$ を初期化し，反復回数 $k := 0$ とする。

Step 1 :[**解の移動**]

近傍 $\mathcal{N}(\boldsymbol{x}^k)$ に対し, タブーリスト $\mathcal{T}$ を参照し, タブーでない解による部分集合 $\tilde{\mathcal{N}}(\boldsymbol{x}^k)$ を生成する。$\tilde{\mathcal{N}}(\boldsymbol{x}^k)$ の解から，最良値となる解 $\boldsymbol{x}'$ を選び，$\boldsymbol{x}^{k+1} := \boldsymbol{x}'$ とする。

Step 2 :[タブーの記憶]

$\boldsymbol{x}^k$ から $\boldsymbol{x}^{k+1}$ への移動をタブーとしてタブーリスト T に追加する。タブーリストの大きさが L_{T} を超えた場合は，一番古いタブーを消去する。

Step 3 :[終了判定]

$k := k+1$ とする。$k = k_{\max}$ ならば，探索を終了。さもなければ，Step 1 へ行く。

3.3　近傍探索に基づく多目的最適化

近傍探索に基づく手法を多目的組合せ最適化問題に適用した例として，Hansen の研究[11]や Ishibuchi らの研究[12]が挙げられる。これらの研究は，いずれも EMO のアプローチを意識した，多点による探索である。一方で，いずれの研究においても，近傍探索に基づく手法を適用する際に加重和最小化手法を用いており，多目的最適化問題から単一目的最適化問題への変換を行っているものと解釈することができる。この変換の際に，探索点の評価値情報に基づく探索点間の相互作用を，加重和最小化手法の重みの決定に活用している。

これに対し，本研究では，多目的最適化問題から単一目的最適化問題への変換を行わないことを念頭に置く。その上で，探索点間の相互作用や探索履歴情報を活用した，多目的組合せ最適化問題に対する，近傍探索に基づく新たな多点探索型最適化手法を提案する。

4 Tabu Search に基づく多点探索型多目的最適化手法の提案

本章で提案する多点探索型多目的組合せ最適化手法を，近傍生成の観点および探索点の選択・移動の観点から説明する。

4.1　多点探索型多目的組合せ最適化手法における近傍生成

本章で提案する手法における近傍生成は，Tabu Search（TS）に基づくものとする。つまり，現在の探索点になんらかの局所的な摂動を加えることにより近傍を生成し，その中から解を選択し，新たな探索点とする。このとき，探索履歴情報を用いて探索に制限をかけることでサイクリングを防ぐ。ただし，提案する手法は多点による探索とし，過去の移動をタブーとする場合は，① 探索点それぞれに対して別々のタブーリストを設定し，② 新たな探索点が含まれる近傍を生成した探索点のタブーリストに，「新たな探索点が含まれる近傍を生成した探索点」から「新たな探索点」への移動をタブーとして追加するものとする。

図 4.1 に評価値空間上の探索点を，図 4.2 に評価値空間上の探索点と近傍解を，それぞれ示す。提案手法では，図 4.2 において，新たな探索点として選ばれた近傍解は，同色の探索点からの移動がタブーとなる。

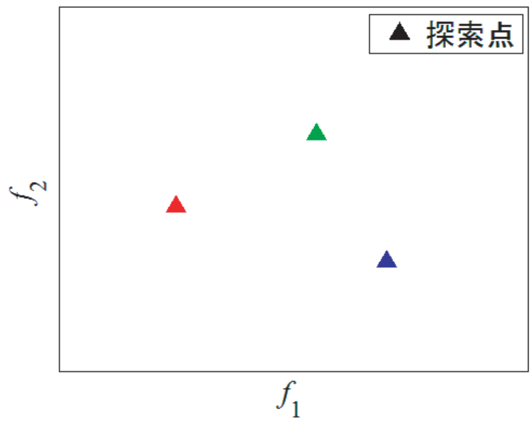


図 4.1：評価値空間上の探索点

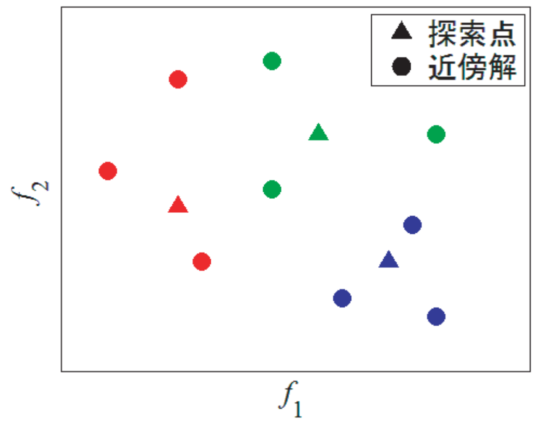


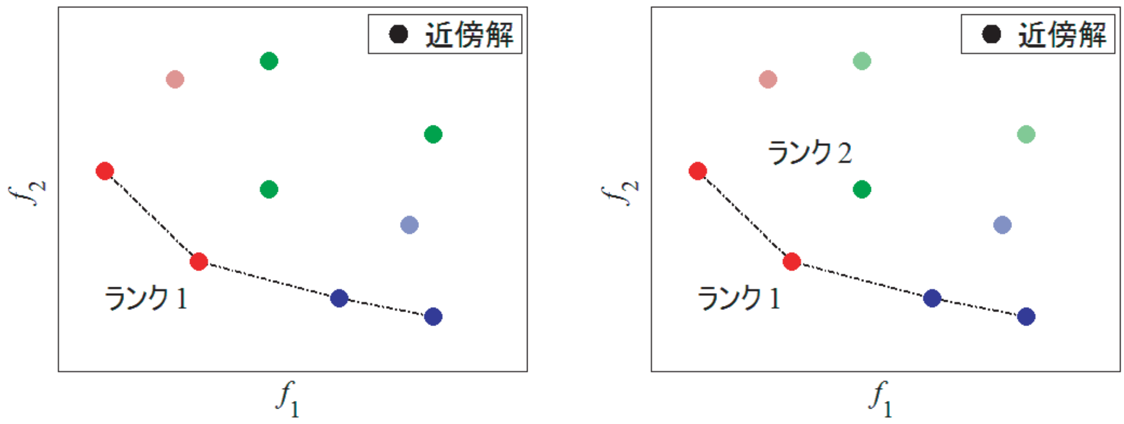
図 4.2：評価値空間上の探索点と近傍解

4.2　多点探索型多目的組合せ最適化手法における探索点の選択・移動

多目的最適化問題では優越関係に基づいて解を評価するため，近傍内に比較不可能である改善解が多数ある場合が考えられる。このようなとき，多目的最適化問題としての解の評価方法や，どの解を新たな探索点として選択するかといった探索戦略が重要となる。

本章で提案する手法では，NSGA-II において導入されている Crowding Tournament 選択を参考にし，近傍解に対してランクと混雑距離に基づいた優先順位を与え，解の評価を行う。探索点数を m，n を 1 から m までの整数とし，ある探索点 $\boldsymbol{x}_i$ の近傍 $\mathcal{N}(\boldsymbol{x}_i)$ から新たな探索点として選択される解の数は，最大で n 個までとなるように，優先順位の高い近傍解から順に新たな探索点を選ぶものとする。これによって，他の探索点と情報を共有することになるため，その相互作用により，単点による探索を超えた，より良い探索を期待することができる。

しかし，組合せ最適化問題に対して近傍探索に基づく多点探索型最適化手法を適用する場合，近傍解の数は膨大なものとなる。そのため，最適化の過程において，近傍解に対するランキングが大きな負担となる。また，ある探索点 $\boldsymbol{x}_i$ の近傍 $\mathcal{N}(\boldsymbol{x}_i)$ からは最大で n 個までしか新たな探索点として選択されないため，$\mathcal{N}(\boldsymbol{x}_i)$ に含まれるランクが下位の解については，優劣の評価が無駄となるばかりか，混雑距離の計算に影響を及ぼし，適切な探索



(a) ランク 1 の解の決定および解の削除　(b) ランク 2 の解の決定および解の削除

図 4.3： 提案したランキング（$n = 1$）

の妨げになる可能性が高いと考えられる。そのため，提案手法に適したランクの割り振りができるよう，対象となる解集合から解を適宜削除するランキングの方法を新たに提案する。以下に，提案手法で用いるランキングのアルゴリズムを示す。

【提案手法で用いるランキングのアルゴリズム】

Step 0 :[準備]　$\tilde{\mathcal{N}}' = \tilde{\mathcal{N}}'(\boldsymbol{x}_1) \cup \tilde{\mathcal{N}}'(\boldsymbol{x}_2) \cup \cdots \cup \tilde{\mathcal{N}}'(\boldsymbol{x}_m)$ に含まれる解を対象とする。パラメータ n を与える。$r := 1$ とする。

Step 1 :[ランクの割り振り]　対象とする解集合 $\tilde{\mathcal{N}}'$ のうち，ほかの解に優越されていない解にランク r を割り振る。

Step 2 :[解の削除]　$\tilde{\mathcal{N}}'$ から，ランクが割り振られた解を削除する。さらに，$\tilde{\mathcal{N}}'(\boldsymbol{x}_i)$ $(i = 1, 2, \cdots, m)$ に含まれる解のうち，n 個以上の解にランクが割り振られたならば，$\tilde{\mathcal{N}}'$ から $\tilde{\mathcal{N}}'(\boldsymbol{x}_i)$ を削除する。

Step 3 :[終了判定]　$r := r + 1$ とする。全ての解にランクが割り振られるか，または削除されたならば，ランキングを終了する。さもなければ，Step 1 へ行く。

図 4.3 に，$n = 1$ としたときの，提案手法で用いるランキングの方法を示す。図 4.3 (a) では，ランク 1 の解を求めた上で，既に n 個以上（今回は 1 個以上）の近傍解にランクが割り振られた赤と青の探索点について，近傍解を削除している。さらに，図 4.3 (b) では，

残っている解の中からランク 2 の解を求めた上で，n 個以上の近傍解にランクが割り振られた緑の探索個について，近傍解を削除している。

4.3 Tabu Search に基づく新たな多点探索型多目的最適化手法のアルゴリズム

本章で提案する，TS に基づく多点探索型多目的最適化手法のアルゴリズムを以下に示す。

【Tabu Search に基づく多点探索型多目的最適化手法のアルゴリズム】

Step 0 :[**準備**]　探索点数 m，各探索点の近傍から新たな探索点となる解の最大許可数 n，タブーリスト長 L_{T}，最大反復回数 $k_{\max}$ を与える。初期解 $\boldsymbol{x}_i^0 (i = 1, 2, \cdots, m)$ を与える。タブーリストを初期化し，$\mathcal{T}_i^0$ とする。反復回数 $k := 0$ とする。

Step 1 :[**解の評価**]　タブーリスト $\mathcal{T}_i^k$ を参照し，$\mathcal{N}(\boldsymbol{x}_i^k)$ からタブーでない部分集合 $\tilde{\mathcal{N}}(\boldsymbol{x}_i^k)$ を生成する。$\tilde{\mathcal{N}}^k = \tilde{\mathcal{N}}(\boldsymbol{x}_1^k) \cup \tilde{\mathcal{N}}(\boldsymbol{x}_2^k) \cup \cdots \cup \tilde{\mathcal{N}}(\boldsymbol{x}_m^k)$ に含まれる全ての解を対象に，ランクおよび混雑距離を求める。

Step 2 :[**解の移動**]　$\tilde{\mathcal{N}}^k$ に含まれる解のうち，上位 m 個の解をそれぞれ $\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_2^{k+1}, \cdots, \boldsymbol{x}_m^{k+1}$ とする。ただし，このとき，$\tilde{\mathcal{N}}(\boldsymbol{x}_i^k)$ から新たな探索点となる解は最大で n 個となるようにする。

Step 3 :[**タブーの記憶**]　$\boldsymbol{x}_j^k$ が，$\boldsymbol{x}_i^{k+1}$ を含む集合 $\tilde{\mathcal{N}}(\boldsymbol{x}_j^k)$ を生成したとする。$\mathcal{T}_i^{k+1} := \mathcal{T}_j^k$ とし，$\boldsymbol{x}_j^k$ から $\boldsymbol{x}_i^{k+1}$ への移動をタブーとしてタブーリスト $\mathcal{T}_i^{k+1}$ に追加する。タブーリストの大きさが L_{T} を超えた場合は，一番古いタブーを消去する。

Step 4 :[**終了判定**]　$k := k + 1$ とする。$k = k_{\max}$ ならば，探索を終了。さもなければ，Step 1 へ行く。

図 4.4 に，TS に基づく多点探索型多目的最適化手法における，探索点の選択と移動のイメージを示す。ここで，丸数字は，ランクと混雑距離を基にした近傍解の優先順位である。$n = m$ のとき，図 4.4 (a) に示す通り，ランクの値が良く，混雑距離が大きい近傍解が，優先的に新たな探索点として選択される。また，$n = 1$ のときは，図 4.4 (b) に示す通り，各探索点の近傍解から一つずつ，新たな探索点として選択される。

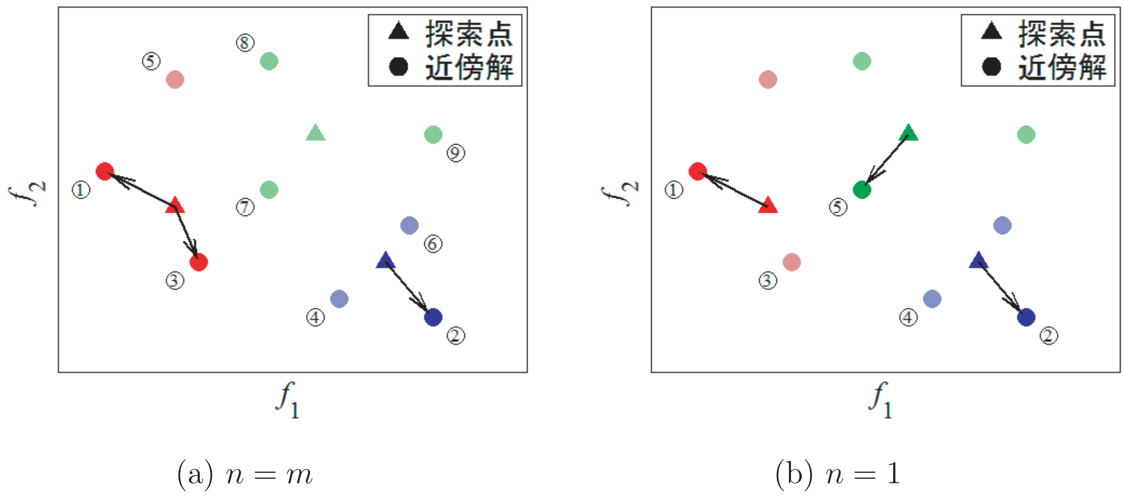


図 4.4： 提案手法の選択・移動

4.4　数値実験と考察

4.4.1　数値実験条件と結果

提案した TS に基づく多点探索型多目的最適化手法 (Proposed Method: PM) を，ベンチマーク問題に対して適用し，数値実験を行う。ベンチマーク問題として，2 目的の 0-1KP, TSP，FSP，QAP を用いる。0-1KP は，Zitzler らのベンチマーク問題[13，14]を用いる。TSP，FSP，QAP は，代表的な単一目的のベンチマーク問題のうち，解の次元が同一となる問題を組み合わせてベンチマーク問題を作成する。ここで，TSP は TSPLIB [15]，FSP は Taillard's instances [15]，QAP は QAPLIB [17] のベンチマーク問題を参照した。なお，0-1KP は最大化問題であり，TSP，FSP，QAP は最小化問題であることに注意されたい。表 4.1 に，ベンチマーク問題として用いた問題の，単一目的最適化問題としての情報を示す。

探索点数 $m = 20$ とし，各探索点の近傍から新たな探索点となる解の最大許可数 n については $n = m$ および $n = 1$ の二通りを用いて数値実験を行う。なお，タブーリストに追加する情報は，以下のものとする。

表 4.1： ベンチマーク問題

	Size		Name	$f(\boldsymbol{x}^*)$
TSP	48 Cities	$f_1(\boldsymbol{x})$	att48	33522
		$f_2(\boldsymbol{x})$	gr48	5046
FSP	20 jobs	$f_1(\boldsymbol{x})$	ta011	1582
	10 machines	$f_2(\boldsymbol{x})$	ta012	1659
QAP	30	$f_1(\boldsymbol{x})$	Tai30a	1818146
		$f_2(\boldsymbol{x})$	Nug30	6124

- 0-1KP は，変更された bit 番号を記憶し，記憶された bit 番号が変更される操作を禁止する。
- TSP は，取り除かれた枝を記憶し，記憶された枝が一つでも加えられる操作を禁止する。
- FSP は，変更されたジョブと処理番号の組を記憶し，記憶されたジョブがそれと組になる処理番号に変更される操作を禁止する。
- QAP は，変更された施設と地点の組を記憶し，記憶された施設がそれと組になる地点に変更される操作を禁止する。

各問題に対して，提案手法を 50 回試行し，その結果得られた解集合を Hypervolume Ratio を用いて評価する。ここで，Hypervolume の基準点 $\boldsymbol{R}$ は，全ての試行における初期解の評価値のうち，各目的関数について最悪値 $f_{i,\mathrm{worst}}$ を用いて，$\boldsymbol{R} = (f_1, f_2) = (f_{1,\mathrm{worst}}, f_{2,\mathrm{worst}})$ とする。また，理論解が判明している 0-1KP については，γ 評価指標を併せて評価に用いる。

提案手法に対する比較対象として，代表的な EMO の手法であり，高い性能を持つことで知られる NSGA-II を用い，提案手法と同一の初期解から探索を行う。NSGA-II について，0-1KP では，選択された親個体に対して確率 P_{c} で一点交叉を行い，さらに個体の各ビットに対して確率 P_{m} で bit 反転突然変異を行う。その他の問題に対しては，選択された親個体に対して確率 P_{c} で部分一致交叉を行い，さらに各個体に対して確率 P_{m} で交換突然

変異を行う。

表 4.2 に提案手法の $n = m$，$n = 1$，および NSGA-II のそれぞれについて，実験条件および 50 回試行したときの Hypervolume Ratio の平均値（Mean），最良値（Best），最悪値（Worst），標準偏差（S. D.）と，評価回数の平均値（Function Call）の結果を示す。また，表 4.3 に，提案手法の $n = m$，$n = 1$，および NSGA-II のそれぞれについて，0-1KP に対して 50 回試行したときの γ 評価指標の平均値（Mean），最良値（Best），最悪値（Worst），標準偏差（S. D.）の結果を示す。図 4.5 から図 4.8 に，0-1KP，TSP，FSP，QAP のそれぞれについて得られた解の例を示す。

表 4.2： 実験条件と結果（Hypervolume Ratio）

Problem	Size	Method	n	L_T	P_c	P_m	k_max	Mean	Best	Worst	S. D.	Function Call
0-1KP	500 Items	PM	m	50			300	0.741	0.802	0.683	0.0273	2776420
		PM	1	50			300	0.685	0.723	0.609	0.0210	2776420
		NSGA-II			0.8	1/500	138500	0.958	0.967	0.948	0.00415	2770020
		PM	m	50			1000	0.846	0.868	0.811	0.0124	9216420
		PM	1	50			1000	0.814	0.844	0.760	0.0173	9216420
		NSGA-II			1	1/500	460500	0.970	0.976	0.966	0.00252	9210020
TSP	48 Cities	PM	m	10			100	0.907	0.913	0.900	0.00291	2140373
		PM	1	10			100	0.899	0.904	0.894	0.00273	2138954
		NSGA-II			0.6	0.8	107500	0.835	0.861	0.801	0.0148	2150020
		PM	m	10			300	0.915	0.919	0.912	0.00146	6404853
		PM	1	10			300	0.912	0.915	0.909	0.00132	6397930
		NSGA-II			0.8	0.9	320500	0.851	0.876	0.815	0.0131	6410020
FSP	20 jobs 10 machines	PM	m	8			150	0.912	0.937	0.897	0.00811	534594
		PM	1	8			150	0.906	0.926	0.898	0.00618	534947
		NSGA-II			0.8	0.9	27000	0.869	0.909	0.801	0.0230	540020
		PM	m	8			500	0.925	0.939	0.912	0.00573	1778932
		PM	1	8			500	0.920	0.931	0.908	0.00514	1779945
		NSGA-II			1	1	89000	0.888	0.924	0.817	0.02208	1780020
QAP	30	PM	m	10			150	0.530	0.553	0.506	0.00906	1261094
		PM	1	10			150	0.508	0.523	0.497	0.00607	1261295
		NSGA-II			0.4	0.9	63500	0.496	0.522	0.463	0.0136	1270020
		PM	m	10			500	0.545	0.557	0.532	0.00744	4198185
		PM	1	10			500	0.529	0.542	0.519	0.00539	4198709
		NSGA-II			0.2	0.9	210000	0.503	0.531	0.479	0.0123	4200020

表 4.3：実験条件と結果（γ 評価指標）

Method	n	$k_{\max}$	Mean	Best	Worst	S. D.	Function Call
PM	m	300	1158.13	930.71	1369.61	98.7	2725520
PM	1	300	1424.65	1349.25	1488.87	53.8	2725520
NSGA-II		136500	231.61	186.53	284.88	21.3	2720020
PM	m	1000	722.53	617.40	814.56	54.0	9025520
PM	1	1000	936.01	884.81	986.96	31.1	9025520
NSGA-II		451500	190.28	156.00	226.38	16.2	9020020

4.4.2　数値実験に対する考察

0-1KP において，表 4.2 より，提案手法のうち，$n = m$ のときの方が $n = 1$ のときと比較して Hypervolume Ratio の値が良いことがわかる。また，表 4.3 より，提案手法のうち，$n = m$ のときの方が $n = 1$ のときと比較して γ 評価指標の値が良いことがわかる。以上から，提案手法において，$n = m$ のときは正確な解を得られており，Hypervolume Ratio が良くなっていることが分かる。これは，$n = m$ とすることで，より正確な近傍解を積極的に新たな探索点として選択し，良い解の付近を集中的に探索を行うことができるためであると考えることができる。

一方で，図 4.5 からもわかるように，提案手法において $n = 1$ としたときは，$n = m$ のときと比較して正確な解は得られていないものの，幅広い解を得られている。これは，Non-Dominated Sorting および Crowding Distance Sorting を用いた優先順位の割り振りが適切に行われているためであると考えられる。つまり，比較的正確ではない解についても，ある目的関数について最善となるような解や，他の解とは評価値が似ていない解を上位として考えるため，適切に多様な探索が行われており，その結果幅広い解を得ることができているものと考えられる。

以上のような傾向は，他の問題においても見られる。表 4.2 より，TSP，FSP，QAP のうち，多くの対象・探索時間において，提案手法について，$n = m$ としたときの方が $n = 1$ と比較して Hypervolume Ratio の値が良いことが確認できる。また，図 4.6，図 4.7，および図 4.8 からも，$n = m$ のときの方が $n = 1$ のときと比較して正確な解が得られていることがわかる。また，$n = 1$ のときは，$n = m$ のときと比較して正確性は欠けるものの，幅

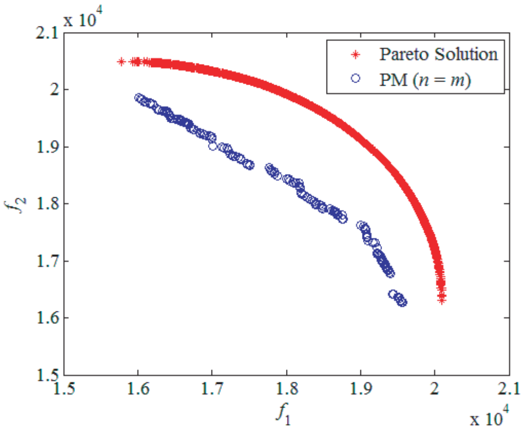
広い解が得られていることもわかる。

NSGA-II と比較すると，表 4.2 より，TSP，FSP，QAP において，提案手法の方が，Hypervolume Ratio が良い値となっている。このことから，提案手法の有用性が示すことができたと考えられる。これは，幅広い解を含む近傍を生成し，適切に選択・移動を繰り返すことができているためであると考えることができる。

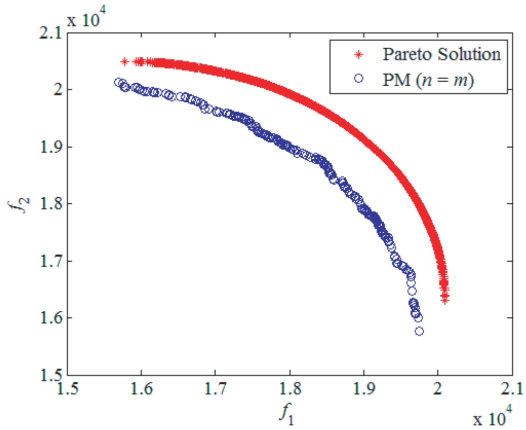
一方で，表 4.2 および表 4.3 より，0-1KP については，提案手法と比較して NSGA-II の方が，Hypervolume Ratio および γ 評価指標の両方について良い値となっていることがわかる。この理由として，

- 0-1KP の場合，近傍 $\mathcal{N}(\boldsymbol{x})$ 内の解は，$\boldsymbol{x}$ と比較して全ての目的関数において改良するか改悪するかの二通りに分別されるため，幅広い近傍生成を行うことができないこと
- 制約条件を満たさない解の扱いが不適切であること

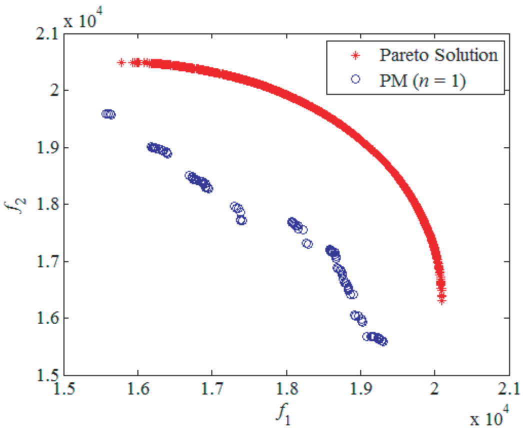
などが考えられる。



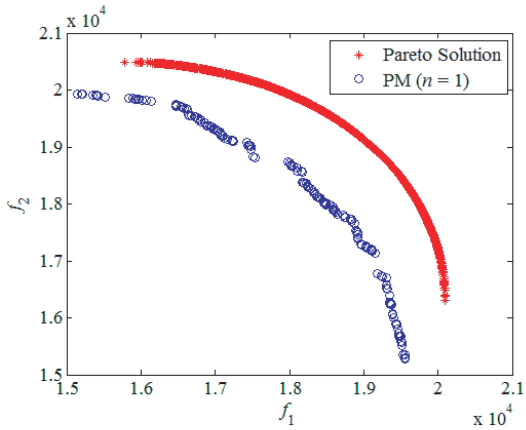
(a) 提案手法（$n = m$），$k_{\mathrm{max}} = 300$



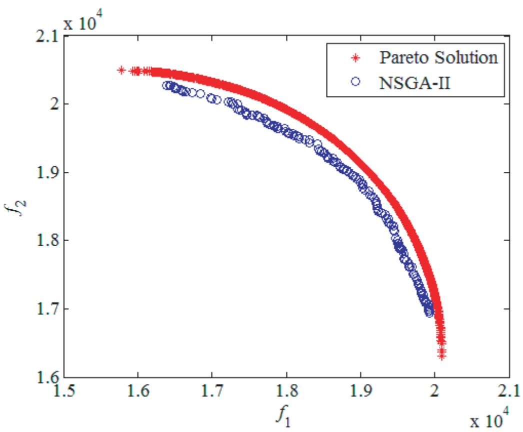
(b) 提案手法（$n = m$），$k_{\mathrm{max}} = 1000$



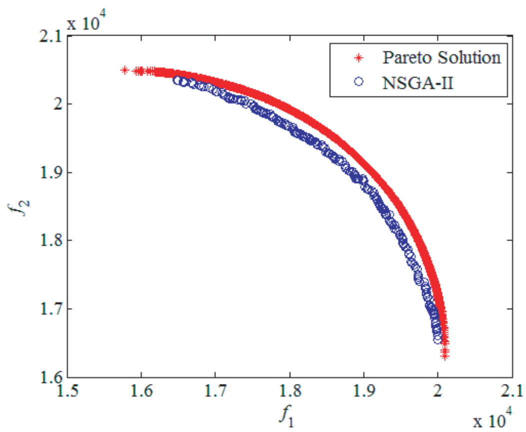
(c) 提案手法（$n = 1$），$k_{\mathrm{max}} = 300$



(d) 提案手法（$n = 1$），$k_{\mathrm{max}} = 300$

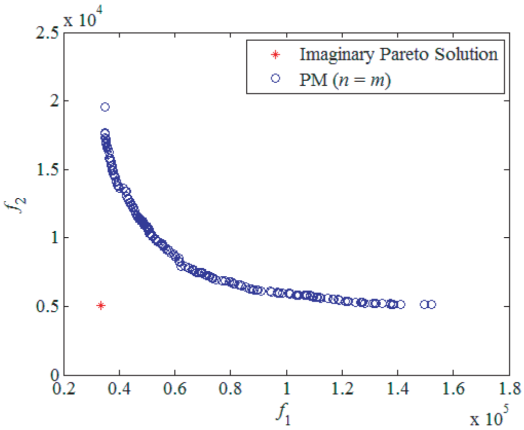


(e) NSGA-II，$k_{\mathrm{max}} = 138500$

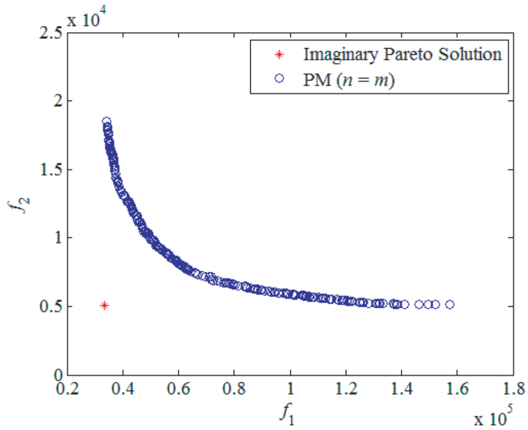


(f) NSGA-II，$k_{\mathrm{max}} = 460500$

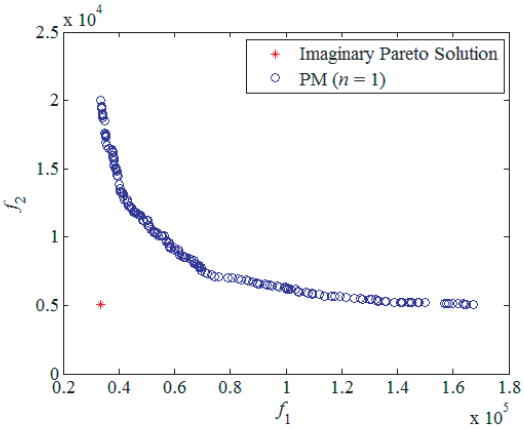
図 4.5： 得られた解の例（0-1KP）



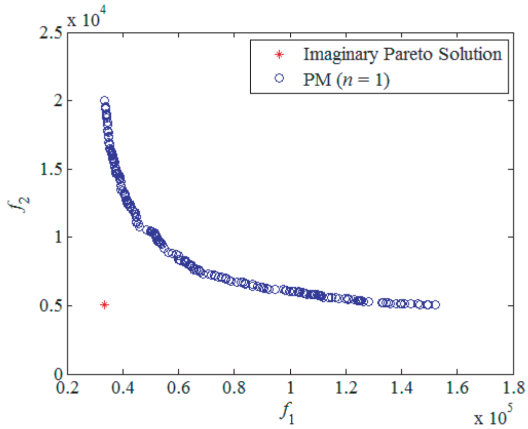
(a) 提案手法（$n = m$），$k_{\mathrm{max}} = 300$



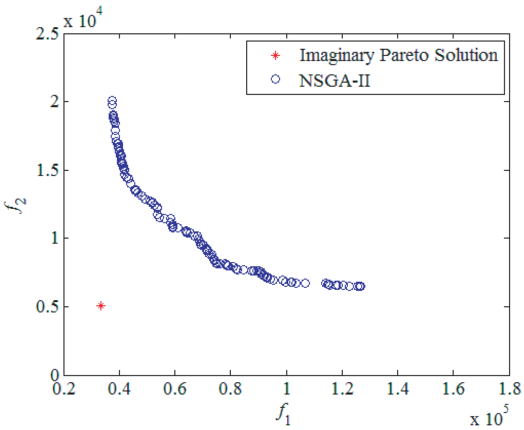
(b) 提案手法（$n = m$），$k_{\mathrm{max}} = 1000$



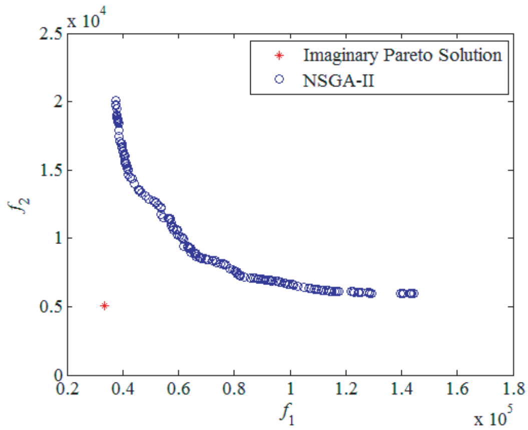
(c) 提案手法（$n = 1$），$k_{\mathrm{max}} = 300$



(d) 提案手法（$n = 1$），$k_{\mathrm{max}} = 300$

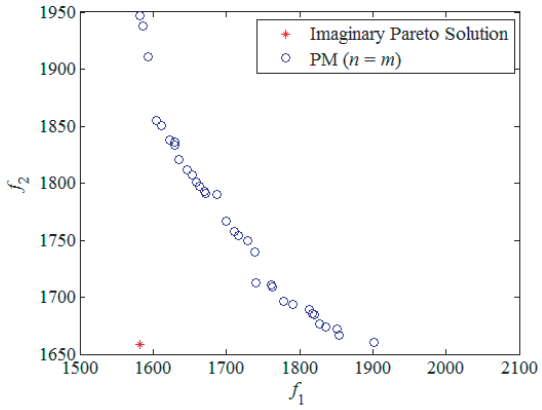


(e) NSGA-II，$k_{\mathrm{max}} = 107500$

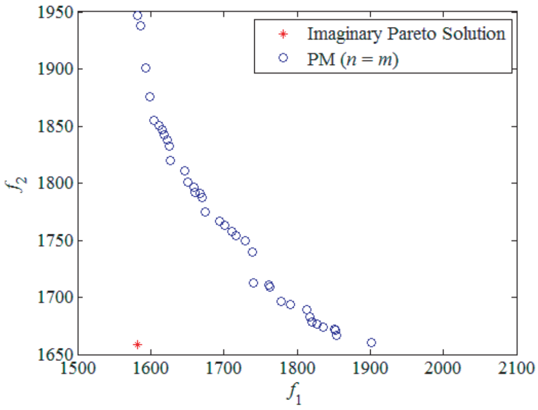


(f) NSGA-II，$k_{\mathrm{max}} = 320500$

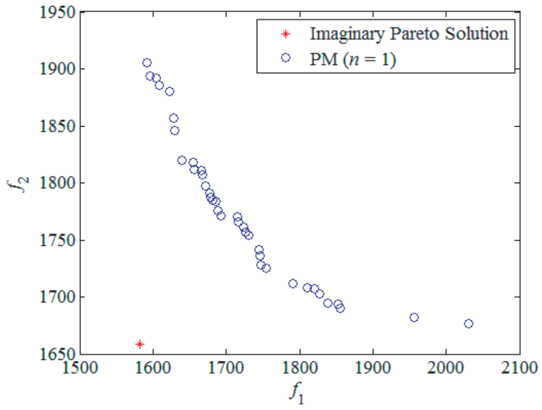
図 4.6：得られた解の例（TSP）



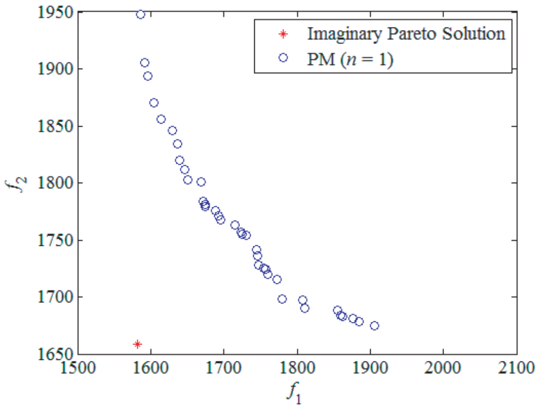
(a) 提案手法（$n = m$），$k_{\max} = 300$



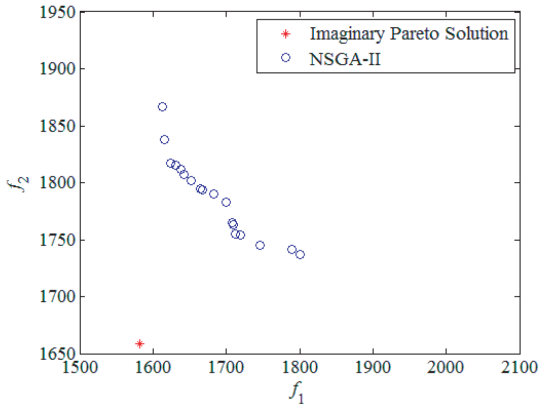
(b) 提案手法（$n = m$），$k_{\max} = 1000$



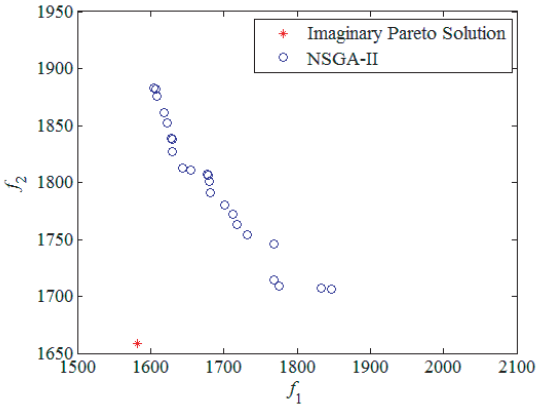
(c) 提案手法（$n = 1$），$k_{\max} = 300$



(d) 提案手法（$n = 1$），$k_{\max} = 300$

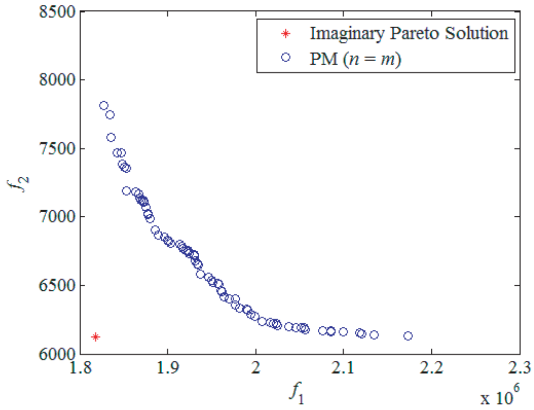


(e) NSGA-II，$k_{\max} = 27000$

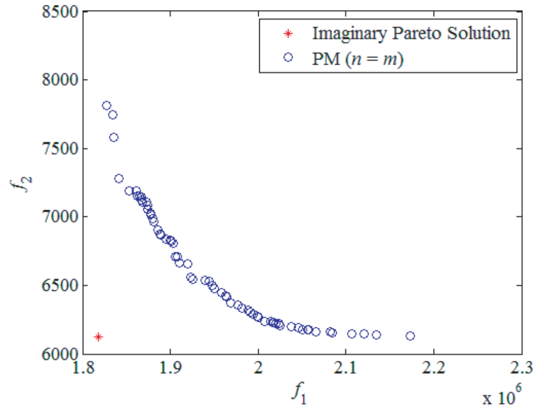


(f) NSGA-II，$k_{\max} = 89000$

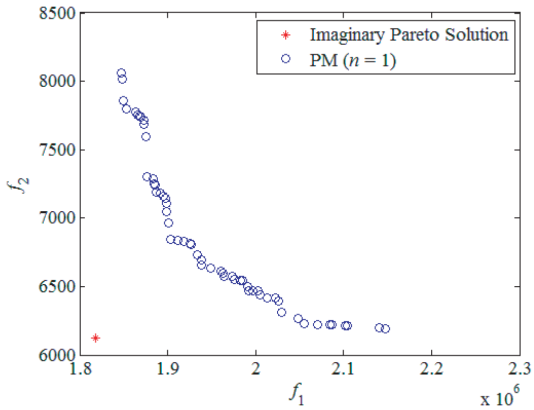
図 4.7：得られた解の例（FSP）



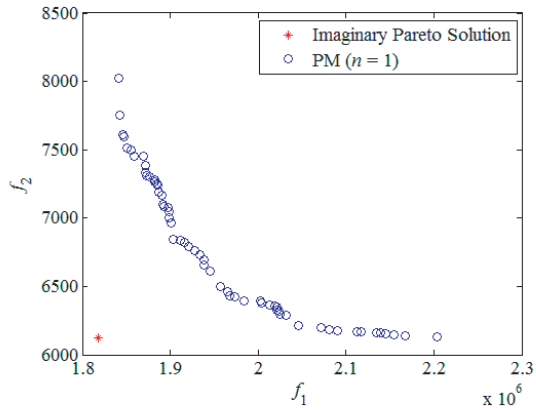
(a) 提案手法（$n = m$），$k_{\max} = 300$



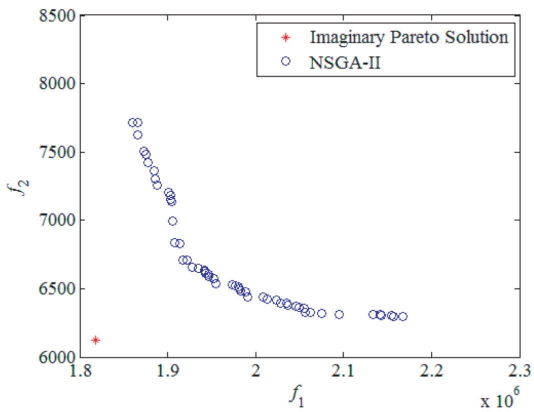
(b) 提案手法（$n = m$），$k_{\max} = 1000$



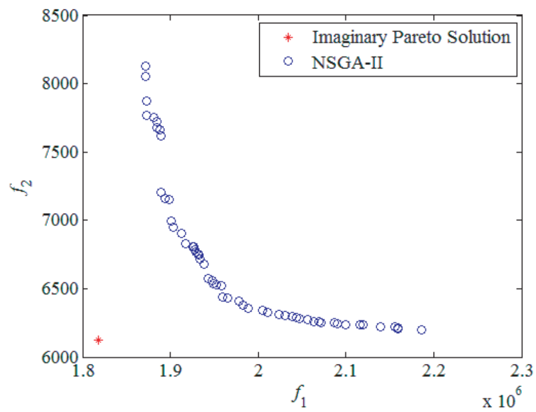
(c) 提案手法（$n = 1$），$k_{\max} = 300$



(d) 提案手法（$n = 1$），$k_{\max} = 300$



(e) NSGA-II，$k_{\max} = 63500$



(f) NSGA-II，$k_{\max} = 210000$

図 4.8： 得られた解の例（QAP）

5 多様化と集中化を考慮した新たな移動戦略の提案

4 章では，Tabu Search（TS）に基づく多点探索型多目的最適化手法を新たに提案し，数値実験を通じて有用性を検証した。本章では，提案した TS に基づく多点探索型多目的最適化手法について，メタヒューリスティクスの探索における代表的な探索戦略である多様化と集中化の観点から解析を行う。解析の結果を基に，TS に基づく多点探索型多目的最適化手法の新たな探索点の移動戦略を提案する。

5.1 メタヒューリスティクスにおける探索戦略

5.1.1 多様化と集中化

メタヒューリスティクスの探索における代表的な探索戦略に多様化と集中化がある。これらは，それぞれ以下のように解釈できる。

多様化 解空間を広く探索する戦略

集中化 良い解の周辺を集中的に探索する戦略

あらゆるメタヒューリスティクスにおいて，探索性能の向上には探索過程における適切な多様化・集中化の実現が重要であることが広く知られている[1]。

5.1.2　メタヒューリスティクスにおける多様化と集中化

多くの優れたメタヒューリスティクスにおいて，探索序盤は多様化を強く働かせ，探索終盤は集中化を強く働かせる戦略が見られる。このような戦略を適切に実現している手法に，単一目的連続型最適化手法の一つであるParticle Swarm Optimization（PSO）や，単点による近傍探索に基づく単一目的組合せ最適化手法であるSimulated Annealing（SA）[18]が挙げられる。

Particle Swarm Optimization

PSOは，1995年にKennedyらによって提案された，多点探索による単一目的連続型最適化手法の一つである。鳥の群れが餌を探す行動研究によって導かれた「情報を群れで共有している」という仮定に基づいた手法であり[1]，マルチエージェント型最適化手法に分類される。PSOの特徴として，

- 複数の探索点による多点探索
- 探索点ごとの最良解情報に加えて，探索全体における最良解情報を共有・利用した探索
- 一定の規則に則った探索

などが挙げられる。

PSOにおける探索点の更新式は，各個体が探索中に発見した最良解である $\boldsymbol{pbest}$ を用いた項，探索全体で発見した最良解である $\boldsymbol{gbest}$ を用いた項，および慣性項によって構成される。探索初期は，慣性項の影響を比較的強く受け，解空間を幅広く均等に探索する。探索中盤から終盤にかけては，次第に $\boldsymbol{pbest}$ および $\boldsymbol{gbest}$ の影響を受け，良い解が存在する領域を中心とした偏りを持つ探索を行う。これによって，探索序盤の多様化と，探索終盤の集中化を実現している。

Simulated Annealing

SAは，Kirkpatrickら，およびCernyによって独自に紹介された，単点による単一目的組合せ最適化手法の一つであり，焼きなましを模擬した手法である。焼きなましにおいては，高い温度のとき，物質の粒子はエネルギーが高い状態であり，ランダムに散らばっている。この状態から，温度を徐々に下げることで，粒子の並び

が整った形で整列された基底状態（最もエネルギーが低い状態）に到達する。SA では，最小化問題の評価値を焼きなましのエネルギーに見立てた手法である。

SA の特徴として，

- 単点による探索
- 近傍探索に基づく探索
- 評価値の他に，温度パラメータ T を用いた，確率的な移動

などが挙げられる。

SA の探索は，探索点からの近傍の生成と，近傍解からランダムに選択した一点に移動するかしないかの選択を繰り返すことによって構成されている。選択・移動の際，移動したときに評価値が改善する場合は必ず移動を行う。また，評価値が改悪するときは，評価値の差および温度パラメータ T に基づいた確率を用いて移動の可否を決定する。改悪する解に移動を許す確率は，T が大きいときは高くなり，T が小さいときは低くなる。また，探索序盤では T は大きく，探索終盤にかけて適切に T を小さくする。以上より，SA は，探索序盤には解空間をランダム探索に近い形で多様に探索し，温度パラメータ T が小さくなる探索終盤では良い解付近を集中的に探索することで，探索序盤の多様化と，探索終盤の集中化を適切に実現した手法である。

5.2　多様化と集中化を考慮した移動戦略

探索序盤での多様化と，探索終盤での集中化の実現を狙い，4 章で提案した TS に基づく多点探索型多目的最適化手法に対する新たな移動戦略を提案する。

4.4.2 の考察から，TS に基づく多点探索型多目的最適化手法のパラメータのうち，各探索点の近傍から新たな探索点となる解の最大許可数 n に着目する。$n = m$ のとき，正確な解の周辺に存在する解を新たな探索点として多く選択するため，解空間において集中的な探索が行われることになる。一方で，$n = 1$ のときは，新たな探索点として，各探索点の近傍から一つずつ解が選択されるため，解空間において比較的幅広く探索が行われている。つまり，$n = m$ のときは解空間において集中的な探索が，$n = 1$ のときは解空間において

多様な探索が行われると考えられる。

以上を踏まえ，各探索点の近傍から新たな探索点となる解の最大許可数 n を，探索の多様化・集中化を制御するパラメータと解釈できる。新たな移動戦略を提案するにあたって，このパラメータを，反復ごとに変化させるパラメータ n^k とした。新たな移動戦略では，探索序盤は n^k を 1 に近い値とすることで解空間を多様に探索し，探索終盤にかけて n^k を徐々に m に近づけることで，良い解周辺を集中的に探索することを狙う。

以下に，新たな移動戦略を用いた，TS に基づく多目的最適化手法のアルゴリズムを示す。

【多様化と集中化を考慮した TS に基づく多点探索型多目的最適化手法のアルゴリズム】

Step 0 :[**準備**]　探索点数 m，タブーリスト長 L_{T}，最大反復回数 $k_{\max}$ を与える。初期解 $\boldsymbol{x}_i^0 (i = 1, 2, \cdots, m)$ を与える。タブーリストを初期化し，$\mathcal{T}_i^0$ とする。反復回数 $k := 0$ とする。

Step 1 :[**解の評価**]　タブーリスト $\mathcal{T}_i^k$ を参照し，$\mathcal{N}(\boldsymbol{x}_i^k)$ からタブーでない部分集合 $\tilde{\mathcal{N}}(\boldsymbol{x}_i^k)$ を生成する。$\tilde{\mathcal{N}}^k = \tilde{\mathcal{N}}(\boldsymbol{x}_1^k) \cup \tilde{\mathcal{N}}(\boldsymbol{x}_2^k) \cup \cdots \cup \tilde{\mathcal{N}}(\boldsymbol{x}_m^k)$ に含まれる全ての解を対象に，ランクおよび混雑距離を求める。

Step 2 :[**解の移動**]　各探索点の近傍から新たな探索点となる解の最大許可数 n^k を求める。$\tilde{\mathcal{N}}^k$ に含まれる解のうち，上位 m 個の解をそれぞれ $\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_2^{k+1}, \cdots, \boldsymbol{x}_m^{k+1}$ とする。ただし，このとき $\tilde{\mathcal{N}}(\boldsymbol{x}_i^k)$ から新たな探索点となる解は最大で n 個となるようにする。

Step 3 :[**タブーの記憶**]　$\boldsymbol{x}_j^k$ が，$\boldsymbol{x}_i^{k+1}$ を含む集合 $\tilde{\mathcal{N}}(\boldsymbol{x}_j^k)$ を生成したとする。$\mathcal{T}_i^{k+1} := \mathcal{T}_j^k$ とし，$\boldsymbol{x}_j^k$ から $\boldsymbol{x}_i^{k+1}$ への移動をタブーとしてタブーリスト $\mathcal{T}_i^{k+1}$ に追加する。タブーリストの大きさが L_{T} を超えた場合は，一番古いタブーを消去する。

Step 4 :[**終了判定**]　$k := k + 1$ とする。$k = k_{\max}$ ならば，探索を終了。さもなければ，Step 1 へ行く。

5.3　数値実験と考察

5.3.1　数値実験条件と結果

提案した移動戦略の有用性を検証するため，ベンチマーク問題を用いて数値実験を行う。ベンチマーク問題としては，4.4 で用いた問題と同様のものを使用し，$n = m$，$n = 1$，$n = n^k$ のそれぞれについて，TS に基づく多点探索型多目的最適化手法 (Proposed Method: PM）を用いて探索を行う。探索点数 $m = 20$ とし，タブーリストに追加する情報は 4.4 と同様のものとする。ここで，提案手法の n^k については，簡単のため，

$$n^k = \begin{cases} 1 & k \leq k_{\max}/2 \\ \frac{m-1}{k_{\max}/2}k + 2 - m & \text{otherwise} \end{cases}$$

の式を用いてスケジューリングした。これは，n^k では反復回数が $k_{\max}/2$ に達するまでは $n = 1$ であり，それ以降は n を 1 から m まで線形に変化させることを意味する。ただし，n は整数しかとれないため，n^k は小数第一位で四捨五入した値を用いることとする。

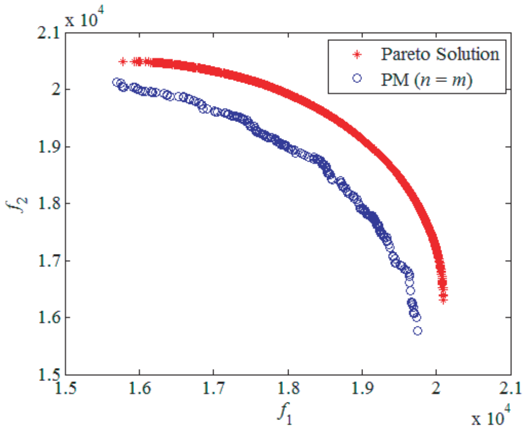
表 5.1 に，実験条件および 50 回試行したときの Hypervolume Ratio の平均値（Mean)，最良値 (Best)，最悪値（Worst)，標準偏差（S. D.）と，評価回数の平均値 (Function Call) の結果を示す。Hypervolume の基準点 $\boldsymbol{R}$ は，全ての試行における初期解の評価値のうち，各目的関数について最悪値 $f_{i,\mathrm{worst}}$ を用いて，$\boldsymbol{R} = (f_1, f_2) = (f_{1,\mathrm{worst}}, f_{2,\mathrm{worst}})$ とする。また，表 5.2 に，0-1KP に対して 50 回試行したときの γ 評価指標の平均値（Mean)，最良値 (Best)，最悪値（Worst)，標準偏差（S. D.）の結果を示す。図 6.2 から図 6.5 に，0-1KP，TSP，FSP，QAP のそれぞれについて得られた解の例を示す。

表 5.1： 実験条件と結果（Hypervolume Ratio）

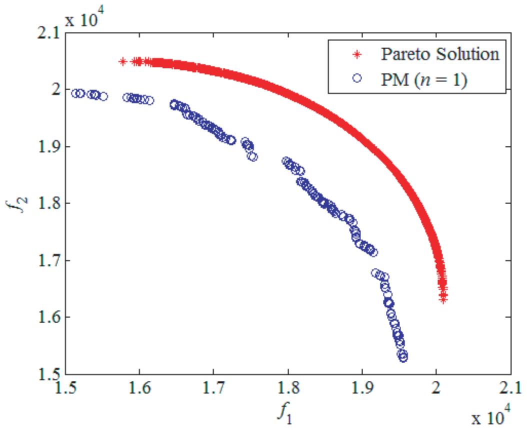
Problem	Size	n	L_{T}	$k_{\max}$	Mean	Best	Worst	S. D.	Function Call
		m			0.886	0.901	0.869	0.00787	9025520
0-1KP	500 Items	1	50	1000	0.854	0.864	0.843	0.00419	9025520
		n^k			0.878	0.888	0.868	0.00528	9025520
		m			0.915	0.919	0.912	0.00146	6404853
TSP	48 Cities	1	10	300	0.912	0.915	0.909	0.00132	6397930
		n^k			0.914	0.918	0.911	0.00165	6401660
	20 jobs	m			0.925	0.939	0.912	0.00573	1778932
FSP	10 machines	1	8	500	0.920	0.931	0.908	0.00514	1779945
		n^k			0.925	0.937	0.912	0.00604	1779543
		m			0.545	0.557	0.532	0.00744	4198185
QAP	30	1	10	500	0.529	0.542	0.519	0.00539	4198709
		n^k			0.544	0.557	0.528	0.00568	4198482

表 5.2： 実験条件と結果（γ 評価指標）

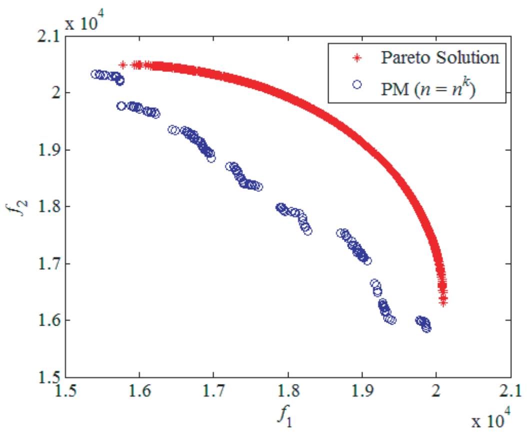
n	$k_{\max}$	Mean	Best	Worst	S. D.	Function Call
m		722.53	617.40	814.56	54.0	9025520
1	1000	936.01	884.81	986.96	31.1	9025520
n^k		1008.84	931.88	1098.54	53.6	9025520



(a) $n = m$

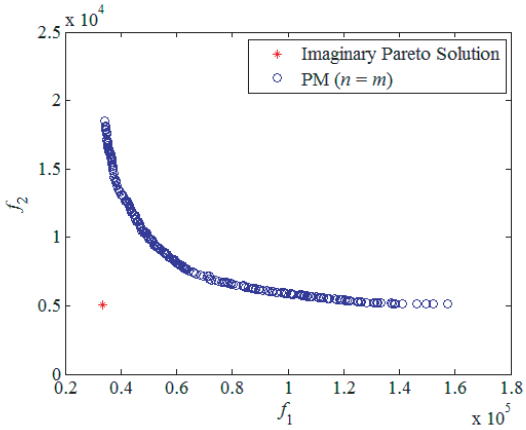


(b) $n = 1$

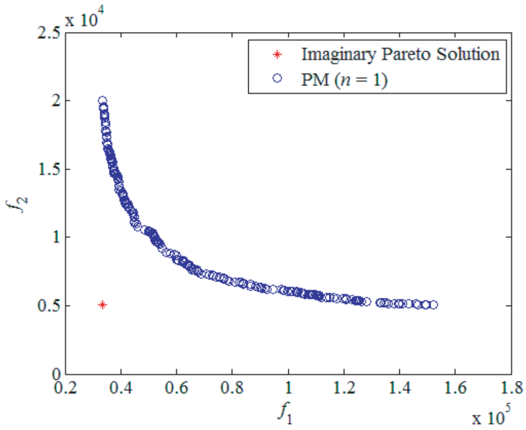


(c) $n = n^k$

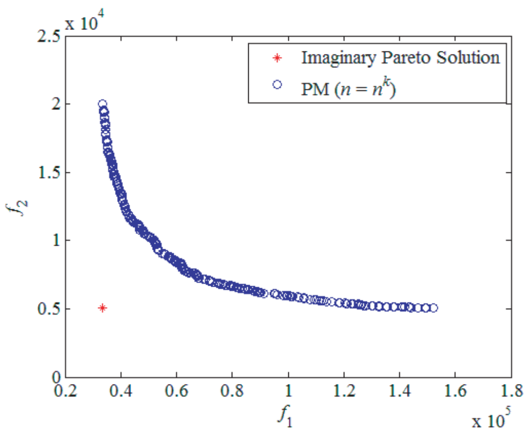
図 5.1： 得られた解の例（0-1KP）



(a) $n = m$

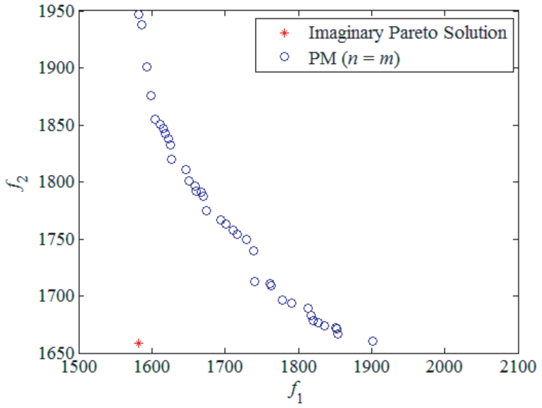


(b) $n = 1$

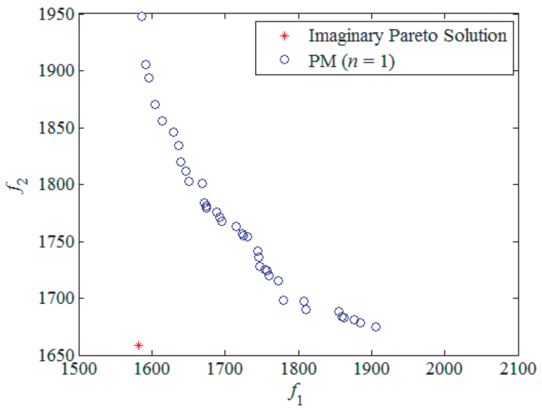


(c) $n = n^k$

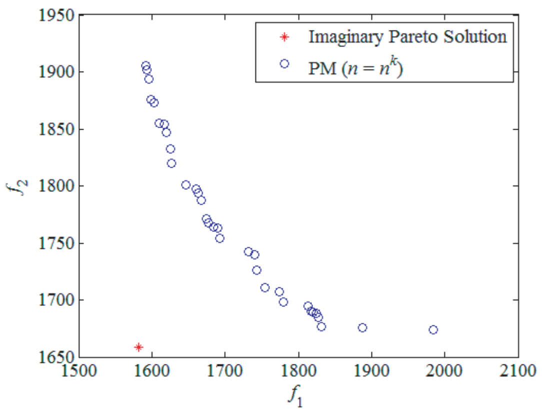
図 5.2： 得られた解の例（TSP）



(a) $n = m$

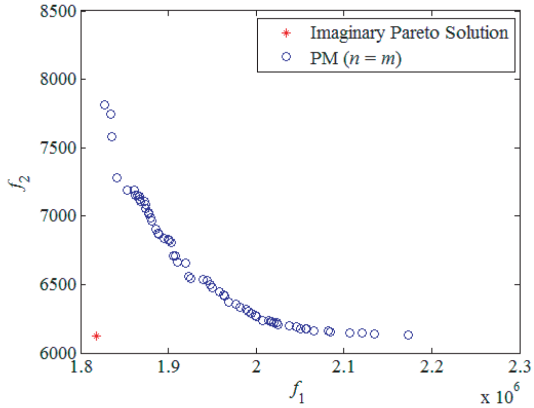


(b) $n = 1$

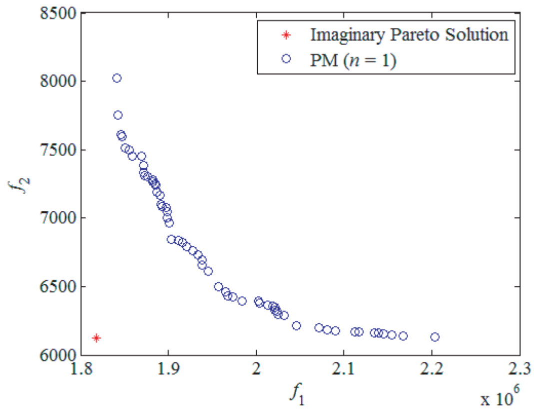


(c) $n = n^k$

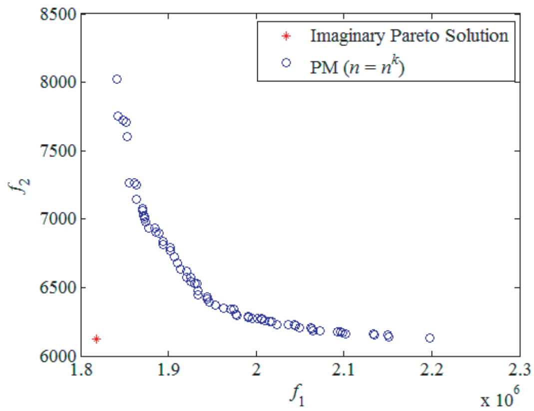
図 5.3： 得られた解の例（FSP）



(a) $n = m$



(b) $n = 1$



(c) $n = n^k$

図 5.4： 得られた解の例（QAP）

5.3.2　数値実験に対する考察

表 5.1 より，全ての問題について，$n = 1$ のときと比較して，$n = n^k$ のときの方が Hypervolume Ratio が良い。また，図からも，TSP，FSP，QAP において，$n = 1$ のときに幅広い解が得られるという特徴を活かしながら，より正確な解を得られていることが分かる。以上より，提案した戦略は一定の有用性があると言える。

一方で，0-1KP においては，表 5.1 より，$n = 1$ のときと比較して Hypervolume Ratio は改善しているものの，表 5.2 から解の正確性は $n = m$ のときおよび $n = 1$ のときの両方と比較して悪化していることが分かる。また，図からは，評価値空間で両端付近に存在する解は $n = 1$ のときよりも改善しているものの，それ以外の解の改善が見られていない。このような原因として，

- 0-1KP の場合，近傍 $\mathcal{N}(\boldsymbol{x})$ 内の解は，$\boldsymbol{x}$ と比較して全ての目的関数において改良するか改悪するかの二通りに分別されるため，幅広い近傍生成を行うことができないこと
- 制約条件を満たさない解の扱いが不適切であること

などが考えられる。

また，本章では多様化・集中化を制御するパラメータとして n を扱った。しかし，これは整数値しかとれないパラメータであり，また今回は単純なスケジューリングを用いているため特に探索中盤において多様化・集中化のバランスが適切にとれているかどうかに関しては検討の余地がある。適切な n^k の決定方法など，より適切に多様化・集中化を評価する指標・パラメータや，多様化・集中化を実現する移動戦略の開発に関しては，今後の課題であると言える。

6 近接最適性原理に基づく多点探索型多目的最適化手法の提案

4 章および 5 章では，Tabu Search に基づく多点探索型多目的最適化手法について扱った。一方で，最適化問題の解構造に広く見られる良構造の偏りとして，近接最適性原理が知られており，多くのメタヒューリスティクスは，POP を根幹に探索を行っていると考えられる。本章では，多目的最適化問題に対する近接最適性原理について検討を行う。そして，その検討を基にした近傍生成および探索点の選択・移動を行う，新たな多点探索型多目的組合せ最適化手法を提案する。

6.1　近接最適性原理

6.1.1　単一目的最適化問題における近接最適性原理

最適化問題の解構造に広く見られる良構造の偏りとして，近接最適性原理（Proximate Optimality Principle: POP）が知られている。POP は,「良い解同士はなんらかの類似構造を有している」という原理である。

多くのメタヒューリスティクスは，POP を根幹に探索を行っていると考えられる。また，神内らの研究[19]や，矢口らの研究[20]などに見られるように，POP を適切に用いることで，より高い探索性能を有する最適化手法の開発が可能である。

しかし，POP は,「良い解」「類似構造」などに漠然さを残している。これらの解釈について，単一目的最適化問題においては,「良い解」を評価値,「類似構造」を解同士の距離

を用いて評価することがある。以下に，各問題に対する距離の定義を示す。

0-1 ナップサック問題

0-1KP において，解は $n$bit の二値ベクトルとして表現される。距離関数は Hamming 距離[21]を用いて，次式で定義する。

$$d_{\mathrm{KP}}(\boldsymbol{x},\boldsymbol{y}) = d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{y}) =\mid \{i \mid x_i \neq y_i (i=1,2,\cdots,n)\} \mid$$

ここで，x_i は二値ベクトル $\boldsymbol{x}$ の i 番目の要素である。

巡回セールスマン問題

TSP において，解は n 頂点の Hamilton 閉路[22]として表現される。距離関数は Hamilton グラフの共通しない辺数として次式で定義する。

$$d_{\mathrm{TSP}}(\boldsymbol{x},\boldsymbol{y}) =\mid \boldsymbol{G}(\boldsymbol{x}) \setminus \boldsymbol{G}(\boldsymbol{y}) \mid$$

ここで，$\boldsymbol{G}(\boldsymbol{x})$ は Hamilton グラフ $\boldsymbol{x}$ の辺集合である。

フローショップ・スケジューリング問題および二次割当問題

FSP および QAP において，解は n-順列として表現できる。距離関数は Cayley 距離[21]を用いる。順列 $\boldsymbol{y}$ に対する順列 $\boldsymbol{x}$ の Cayley 距離は，任意の要素対を入れ替える操作により，$\boldsymbol{x}$ から $\boldsymbol{y}$ へ変換するために必要となる最小操作回数で定義される。解 $\boldsymbol{x}$ の端の要素から順に，解 $\boldsymbol{y}$ と一致するように要素対を入れ替えていき，操作回数を数えることで求めることができる。また，置換 $(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}$ の巡回置換 (cycle) の数 $\mid T\big((\boldsymbol{x}^{\mathrm{T}},\boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}\big) \mid$ を用いて以下の式で算出できることも知られており，特に，順列の要素数が多い場合には有効な手段となる。

$$d_{\mathrm{FSP}}(\boldsymbol{x},\boldsymbol{y}) = d_{\mathrm{C}}(\boldsymbol{x},\boldsymbol{y}) = n- \mid T\big((\boldsymbol{x}^{\mathrm{T}},\boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}\big) \mid$$

$$d_{\mathrm{QAP}}(\boldsymbol{x},\boldsymbol{y}) = d_{\mathrm{C}}(\boldsymbol{x},\boldsymbol{y}) = n- \mid T\big((\boldsymbol{x}^{\mathrm{T}},\boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}\big) \mid$$

6.1.2　多目的最適化問題における近接最適性原理

多目的最適化問題において POP を考える場合においても，「類似構造」は解同士の距離で評価することが可能である。一方で，「良い解」については，単純に評価値のみでは評価

できず，ランクを用いて評価することが妥当であると考えられる。しかし，多目的最適化問題において，ランクが良い解は多数あり，またそれらの構造は多様性を持っている。例えば，2 目的最適化問題において，同ランクでも，評価値空間上で両端にある解同士では構造は大きく異なるが，評価値空間において近くに存在する解は類似構造を持っていると考えられる。

以上を踏まえて，本研究では，多目的最適化問題における POP を，以下のように解釈する。

多目的最適化問題における POP

ランクの値が良く，評価値空間上の距離が小さい解同士は，なんらかの類似構造を有している

6.2　近接最適性原理を考慮した多点探索型多目的組合せ最適化手法

多目的最適化問題における POP を考慮した，新たな多点探索型多目的組合せ最適化手法を提案する。提案する手法の構想を，近傍生成の観点および探索点の選択・移動の観点から説明する。

6.2.1　近接最適性原理を考慮した多点探索型多目的組合せ最適化手法における近傍生成

評価回数の増加を防ぎ，効率のよい探索を行うために，タブーリストを用いることに加えて，多目的最適化問題における POP に基づき近傍生成に制限をかける。近傍生成の制限のかけ方は，ランクが 1 のときと，それ以外のときで異なるものとする。

以下に，近傍生成における基本的な方針を示す。

ランクが 1 の探索点

ランク 1 の探索点は，正確性を高めつつ，解同士が集中しないよう幅広く探索する

ことが必要である。そのため，ランク 1 の探索点については，評価値空間で最も近い探索点から遠ざかるように近傍生成に制限をかけるものとする。

ランクが 1 以外の探索点

ランクが 1 以外の探索点については，まず正確性を高くすることが重要である。そのため，ランク 1 の探索点に近づくように近傍生成に制限をかける。ここで，ランク 1 の探索点が複数ある場合は，評価値空間上で最も近い探索点を選択する。

ここで，評価値空間上での探索点間の距離は，探索点をそれぞれ$\boldsymbol{x}$，$\boldsymbol{y}$としたとき，

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{r} |f_i(\boldsymbol{x}) - f_i(\boldsymbol{y})|$$

によって求める。

以下に，近傍生成のアルゴリズムを示す。

【近接最適性原理を考慮した多点探索型多目的組合せ最適化手法における近傍生成】

Step 1 :[**探索点のランキング**]　現在の探索点 $\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, \cdots, \boldsymbol{x}_m^k$ を対象に，ランクを与える。その段階で求められているパレート解集合を $\mathcal{P}'$ とする。

Step 2 :[**近傍生成**]　$\boldsymbol{x}_j^k (j = 1, 2, \cdots, m)$ について，以下に従って近傍を生成する。

Step 2-1 :[**対象となる解の選択**]　$\boldsymbol{x}_j^k$ がランク 1 の探索点であり，かつ $|\mathcal{P}'| \neq 1$ ならば，$\left\{\mathcal{P}' \backslash \boldsymbol{x}_j^k\right\}$ の中から

$$d(\boldsymbol{x}_j^k, \boldsymbol{p}') = \sum_{i=1}^{r} \left|f_i(\boldsymbol{x}_j^k) - f_i(\boldsymbol{p}')\right|$$

が最小となる解 $\boldsymbol{p}'$ を選ぶ。さもなければ，集合 $\mathcal{P}'$ の中から

$$d(\boldsymbol{x}_j^k, \boldsymbol{p}') = \sum_{i=1}^{r} \left|f_i(\boldsymbol{x}_j^k) - f_i(\boldsymbol{p}')\right|$$

が最小となる解 $\boldsymbol{p}'$ を選ぶ。

Step 2-2 :[**解空間上での距離を用いた近傍生成**]　$\boldsymbol{x}_j^k$ がランク 1 の探索点のときは，解 $\boldsymbol{p}'$ から解空間上での距離が離れるように近傍生成を行う。それ以外のときは，解 $\boldsymbol{p}'$ に解空間上での距離が近づくように近傍生成を行う。

6.2.2 近接最適性原理を考慮した多点探索型多目的組合せ最適化手法における選択・移動

近傍解の評価はランキングによって行う。さらに，近傍生成に基づく最適化手法であることを最大限に活かすため，探索点ごとに一対一の移動を行うものとする。

しかし，6.2.1 に述べた方法で近傍生成を行った場合，通常の近傍探索に基づく組合せ最適化手法と比較して近傍生成に大きく制限を受け，近傍解は比較的集中したものとなる。よって，単純にランクと混雑距離を用いた選択・移動を行うと，適切な移動が行われない。

そのため，以下のアルゴリズムを用いて，探索点が集中しないように選択・移動を行う。

【近接最適性原理を考慮した多点探索型多目的組合せ最適化手法における選択・移動】

Step 0 :[**準備**]　整数 l を定める。現在の探索点を $\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, \cdots, \boldsymbol{x}_m^k$，探索点 $\boldsymbol{x}_i^k$ の近傍を $\tilde{\mathcal{N}}(\boldsymbol{x}_i^k)$ とし，$\tilde{\mathcal{N}}^k = \tilde{\mathcal{N}}(\boldsymbol{x}_1^k) \cup \tilde{\mathcal{N}}(\boldsymbol{x}_2^k) \cup \cdots \cup \tilde{\mathcal{N}}(\boldsymbol{x}_m^k)$ とする。

Step 1 :[**非劣解の計算**]　$\tilde{\mathcal{N}}^k$ に含まれる全ての解を対象としたときに卑劣解となる解の集合 $\tilde{\mathcal{N}}'^k$ を求める。新たな探索点の集合 $\mathcal{X}^{k+1} = \varnothing$ とする。

Step 2 :[**新しい探索点の選択**]　$\left|\tilde{\mathcal{N}}'^k\right| = 0$ となるまで，以下の操作を行う。

Step 2-1 :[**各目的関数についての最良解の選択**]　$r \leq l$ ならば，$\tilde{\mathcal{N}}'^k$ に含まれる解のうち，各目的関数 f_i について最良値を取る解 $\boldsymbol{x}'$ を新たな探索点とし，$\mathcal{X}^{k+1} := \mathcal{X}^{k+1} \cap \boldsymbol{x}'$ とする。このとき，新たな探索点として選ばれた解 $\boldsymbol{x}'$ が $\tilde{\mathcal{N}}(\boldsymbol{x}_j^k)$ から選ばれたとすると，$\tilde{\mathcal{N}}(\boldsymbol{x}_j^k)$ に含まれる解を $\tilde{\mathcal{N}}'$ から削除し，$\tilde{\mathcal{N}}'(\boldsymbol{x}_i)$ を $\tilde{\mathcal{N}}'^k$ から削除する。

Step 2-2 :[**評価値空間における疎密の判断**]　$\tilde{\mathcal{N}}'^k$ に含まれるすべての解について，以下の計算を行い，各解の d_{PM} を求める。

$$d_{\mathrm{PM}} = \sum_{i=1}^{r} d_i$$

$$d_i = \{f_i(\boldsymbol{x}) - f_i(\boldsymbol{x}_{i,1})\}^2 + \{f_i(\boldsymbol{x}) - f_i(\boldsymbol{x}_{i,2})\}^2$$

ただし，$\boldsymbol{x}$ は対象となる解であり，$\boldsymbol{x}_{i,1}$ および $\boldsymbol{x}_{i,2}$ は，解 $\boldsymbol{x}$ について，$\mathcal{X}^{k+1}$ に含まれる解のうち各目的関数 f_i ごとに両隣に位置する二つの解である。

Step 2-3 :[疎な領域の解の選択]　d_{PM} が最小となる解 $\boldsymbol{x}'$ を新たな探索点とし，$\mathcal{X}^{k+1} := \mathcal{X}^{k+1} \cap \boldsymbol{x}'$ とする。このとき，新たな探索点として選ばれた解 $\boldsymbol{x}'$ が $\tilde{\mathcal{N}}(\boldsymbol{x}_j^k)$ から選ばれたとすると，$\tilde{\mathcal{N}}(\boldsymbol{x}_j^k)$ に含まれる解を $\tilde{\mathcal{N}}'^k$ から削除し，$\tilde{\mathcal{N}}'(\boldsymbol{x}_i)$ を $\tilde{\mathcal{N}}'^k$ から削除する。Step 2-2 に行く。

Step 3 :[終了判定]　$r := r + 1$ とする。$\left|\mathcal{X}^{k+1}\right| = m$ ならば，終了。さもなければ，Step 1 に行く。

選択の際に，各目的関数について，評価値空間におけるパレート解との距離の 2 乗を用いて移動先を決定することで，各探索点について一対一に移動を行いながらも，評価値空間において幅広さを持った探索が期待できる。

6.3　近接最適性原理を考慮した多点探索型多目的組合せ最適化手法のアルゴリズム

以下に，POP を考慮した多点探索型多目的組合せ最適化手法のアルゴリズムを示す。

【近接最適性原理を考慮した多点探索型多目的組合せ最適化手法のアルゴリズム】

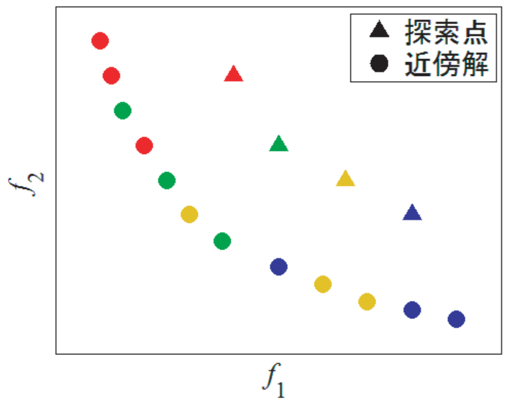
Step 0 :[準備]　探索点数 m，パラメータ l，タブーリスト長 L_{T}，最大反復回数 $k_{\max}$ を与える。初期解 $\boldsymbol{x}_i^0 (i = 1, 2, \cdots, m)$ を与える。タブーリストを初期化し，$\mathcal{T}_i^0$ とする。反復回数 $k := 0$ とする。

Step 1 :[解の評価]　6.2.1 の方法で，近傍 $\mathcal{N}(\boldsymbol{x}_i^k)$ を生成する。さらに，タブーリスト $\mathcal{T}_i^k$ を参照し，$\mathcal{N}(\boldsymbol{x}_i^k)$ からタブーでない部分集合 $\tilde{\mathcal{N}}(\boldsymbol{x}_i^k)$ を生成する。$\tilde{\mathcal{N}}^k = \tilde{\mathcal{N}}(\boldsymbol{x}_1^k) \cup \tilde{\mathcal{N}}(\boldsymbol{x}_2^k) \cup \cdots \cup \tilde{\mathcal{N}}(\boldsymbol{x}_m^k)$ とする。

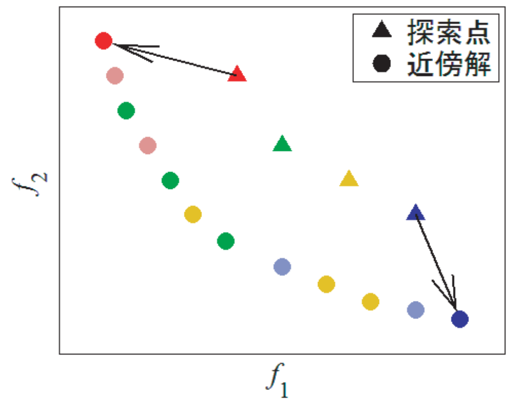
Step 2 :[解の移動]　6.2.2 の方法で，新たな探索点 $\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_2^{k+1}, \cdots, \boldsymbol{x}_m^{k+1}$ を決定する。

Step 3 :[タブーの記憶]　$\boldsymbol{x}_j^k$ が，$\boldsymbol{x}_i^{k+1}$ を含む集合 $\tilde{\mathcal{N}}(\boldsymbol{x}_j^k)$ を生成したとする。$\mathcal{T}_i^{k+1} := \mathcal{T}_j^k$ とし，$\boldsymbol{x}_j^k$ から $\boldsymbol{x}_i^{k+1}$ への移動をタブーとしてタブーリスト $\mathcal{T}_i^{k+1}$ に追加する。タブーリストの大きさが L_{T} を超えた場合は，一番古いタブーを消去する。

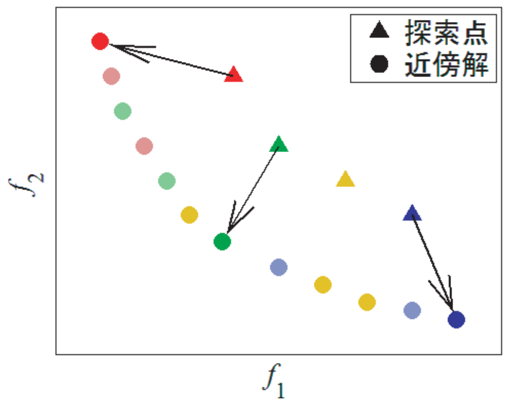
Step 4 :[終了判定]　$k := k + 1$ とする。$k = k_{\max}$ ならば，探索を終了。さもなければ，Step 1 へ行く。



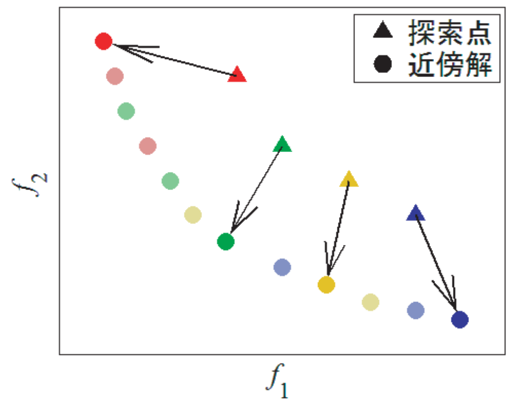
(a) 探索点および同ランクの近傍解



(b) 両端の解の選択



(c) 疎な領域の解の選択 (1)



(d) 疎な領域の解の選択 (2)

図 6.1： 提案手法における探索点の選択・移動

図 6.1 に，本章で提案した手法における探索点の選択・移動のイメージを示す。図 6.1 (a) は，探索点と，それらの近傍解のうちランクが同一のものを示している。これらを対象に新たな探索点を決定するとき，まず，図 6.1 (b) のように，各目的関数ごとに最良値となる近傍解が新たな探索点として選択される。さらに，図 6.1 (c) および (d) のように，新たな探索点とすることが決定している近傍解について，最も疎な領域にある近傍解が新たな探索点となるよう，逐次決定していくことを狙う。

6.4　数値実験と考察

6.4.1　数値実験条件と結果

本章で提案した手法の有用性を検証するため，ベンチマーク問題を用いて数値実験を行う。ベンチマーク問題としては，4.4 で用いた問題と同様のものを使用する。POP を考慮した多点探索型多目的組合せ最適化手法（Proposed Method: PM）について，探索点数 $m = 20$，$l = 2$ とし，タブーリストに追加する情報は 4.4 と同様のものとする。比較手法として，4 章で提案した手法を，$n = m$ および $n = 1$ として用いる。

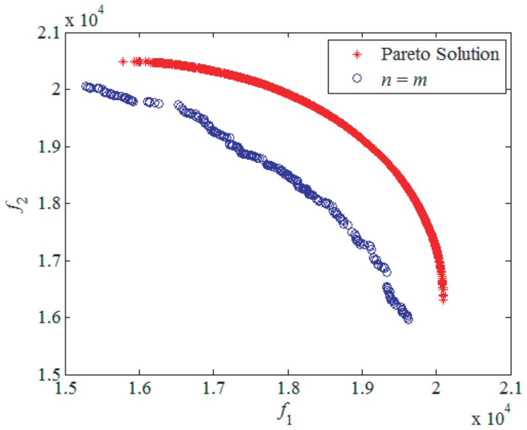
表 6.1 に，実験条件および 20 回試行したときの Hypervolume Ratio の平均値（Mean），最良値（Best），最悪値（Worst），標準偏差（S. D.）と，評価回数の平均値（Function Call）の結果を示す。Hypervolume の基準点 $\boldsymbol{R}$ は，全ての試行における初期解の評価値のうち，各目的関数について最悪値 $f_{i,\mathrm{worst}}$ を用いて，$\boldsymbol{R} = (f_1, f_2) = (f_{1,\mathrm{worst}}, f_{2,\mathrm{worst}})$ とする。また，表 6.2 に，0-1KP に対して 20 回試行したときの γ 評価指標の平均値（Mean），最良値（Best），最悪値（Worst），標準偏差（S. D.）の結果を示す。図 6.2 から図 6.5 に，0-1KP，TSP，FSP，QAP のそれぞれについて得られた解の例を示す。

表 6.1：実験条件と結果（Hypervolume Ratio）

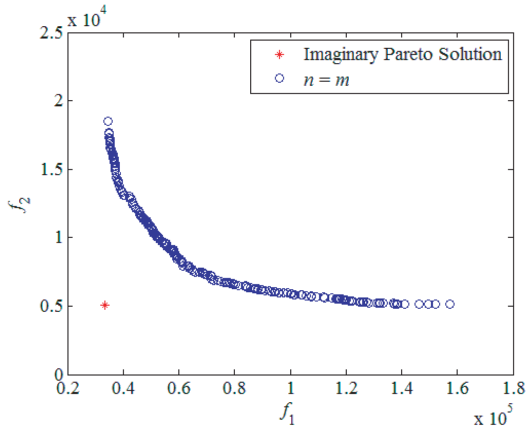
Problem	Size	Method	L_T	k_{max}	Mean	Best	Worst	S. D.	Function Call
		$n = m$		495	0.865	0.885	0.846	0.01072	4480520
0-1KP	500 Items	$n = 1$	50	495	0.827	0.840	0.813	0.00664	4480520
		PM		1000	0.684	0.786	0.529	0.06039	4488666
		$n = m$		159	0.912	0.917	0.907	0.00221	3398509
TSP	48 Cities	$n = 1$	10	159	0.907	0.912	0.901	0.00226	3395482
		PM		300	0.821	0.854	0.759	0.02212	3416424
	20 jobs	$n = m$		138	0.910	0.930	0.887	0.00796	491920
FSP	10 machines	$n = 1$	8	138	0.905	0.920	0.897	0.00567	492265
		PM		500	0.846	0.875	0.804	0.01403	494270
		$n = m$		151	0.530	0.553	0.506	0.00911	1269484
QAP	30	$n = 1$	10	151	0.509	0.523	0.497	0.00611	1269688
		PM		500	0.470	0.497	0.408	0.02417	1270547

表 6.2：実験条件と結果（γ 評価指標）

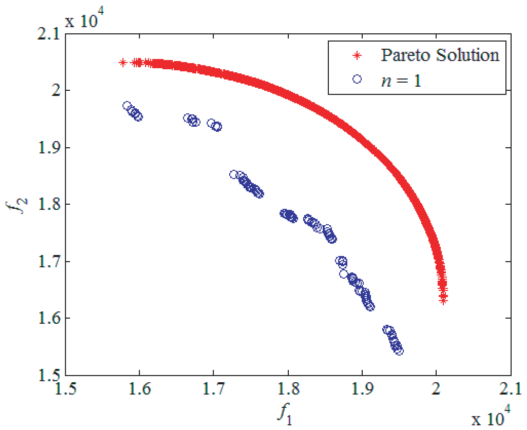
Method	k_{max}	Mean	Best	Worst	S. D.	Function Call
$n = m$	495	875.41	748.46	1003.21	63.1	4480520
$n = 1$	495	1106.22	990.63	1246.54	60.5	4480520
PM	1000	1760.34	1167.01	2513.59	344.0	4488666



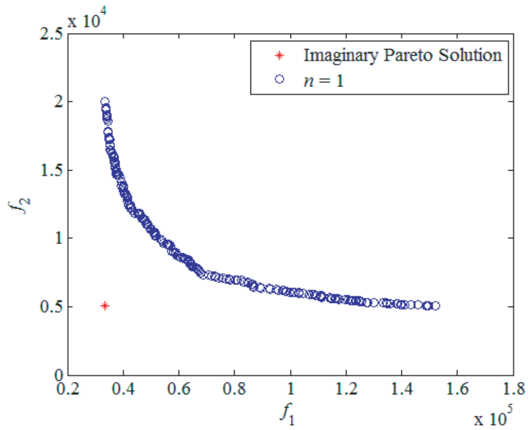
(a) $n = m$



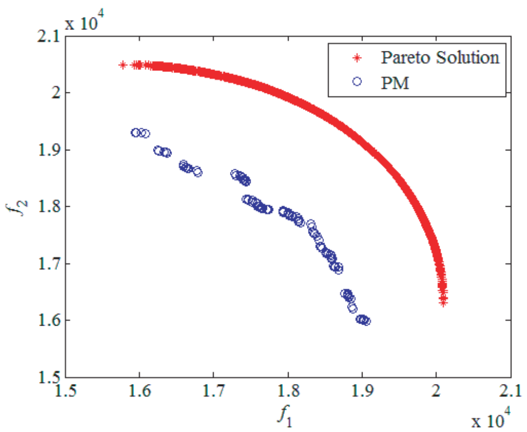
(a) $n = m$



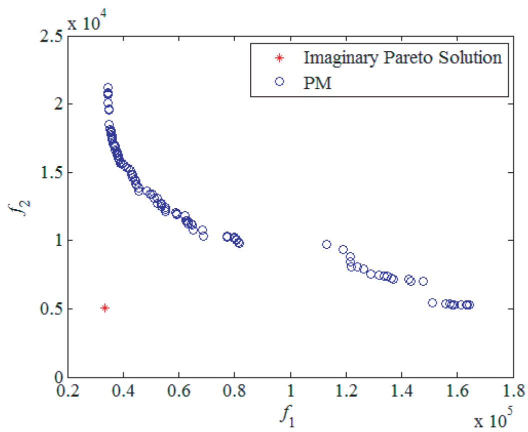
(b) $n = 1$



(b) $n = 1$



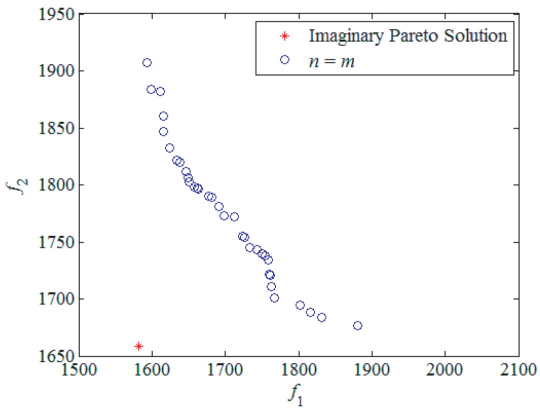
(c) PM



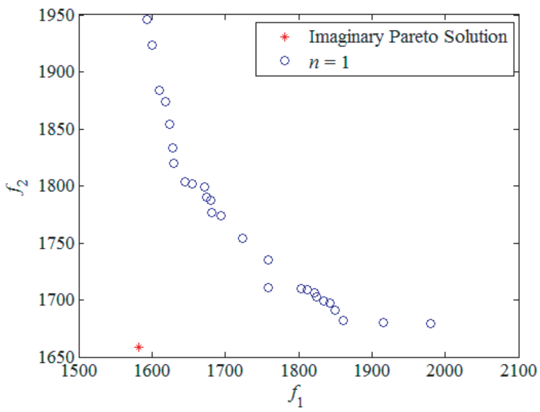
(c) PM

図 6.2： 得られた解の例（0-1KP）

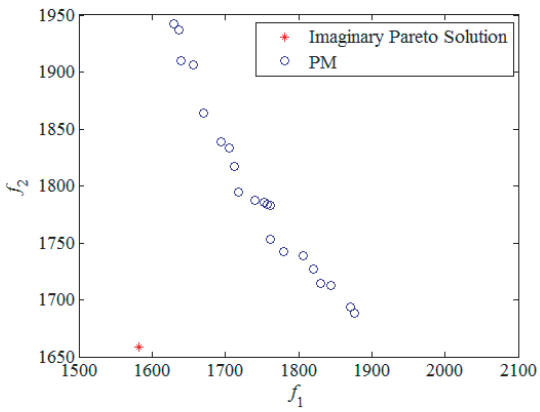
図 6.3： 得られた解の例（TSP）



(a) $n = m$

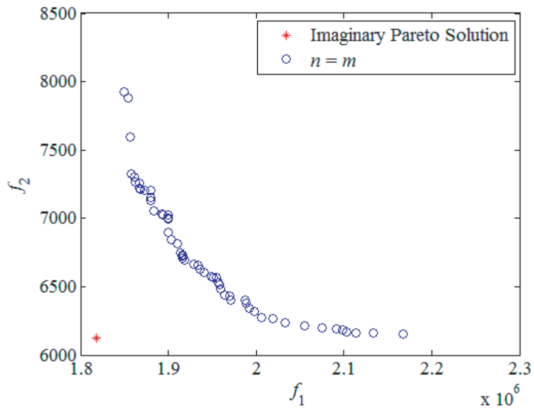


(b) $n = 1$

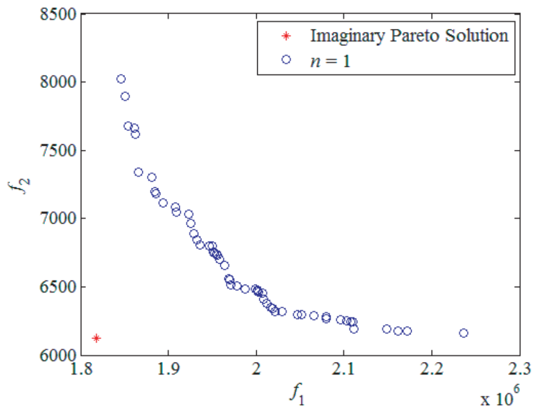


(c) PM

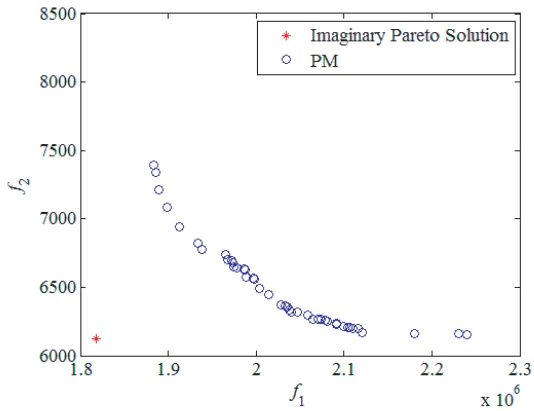
図 6.4： 得られた解の例（FSP）



(a) $n = m$



(b) $n = 1$



(c) PM

図 6.5： 得られた解の例（QAP）

6.4.2　数値実験に対する考察

表 6.1 より，全ての問題において，POP を考慮した多点探索型多目的組合せ最適化手法は，同じく一対一の移動を行う $n = 1$ のときと比較しても Hypervolume Ratio が低い値となっている。また，表 6.2，および図 6.2 から図 6.5 より，Hypervolume Ratio が低い原因として，$n = m$ のときおよび $n = 1$ のときと比較して，解の正確性が劣るからであると考えることができる。

この主な原因として，以下のものが考えられる。

- ランク 1 の探索点が，正確性を高めるための適切な移動を行えていない
  - POP を考慮した多点探索型多目的組合せ最適化手法では，評価値空間上で近くに位置する解から離れるような近傍生成を行い，さらにタブーを用いて探索のサイクリングを防ぐように方向づけを行ったが，適切な近傍生成になっていなかった
  - POP を考慮した多点探索型多目的組合せ最適化手法における，評価値空間上でランクが高く，疎な領域に存在する解を選択し，探索点ごとに一対一で移動する移動方法が，生成した近傍に対して適切なものではなかった
- ランク 2 以下の探索点が，広がりを持った解を得るための適切な移動を行えていない
  - POP を考慮した多点探索型多目的組合せ最適化手法では，得られたパレート解に評価値空間上で近づくような近傍生成を行ったが，このような近傍生成が，広がりを持つ解を得る妨げとなった
  - 各目的関数について最良値を取る解を新たな探索点とするような移動戦略が，うまく働かなかった

これらについての検証は，今後の課題である。

7 結論

7.1　まとめ

本論文では，多目的組合せ最適化問題を対象とした，実用的な最適化手法の構築を目標とし，近傍探索に基づく多点探索型多目的組合せ最適化手法を提案した。

そして，数値実験を通してその有用性を検証した。本論文で得られた成果を以下に述べる。

(1) 近傍生成は，近傍探索に基づく代表的な手法である TS に基づき，選択・移動については，EMO の代表的な手法である NSGA-II において導入されている Crowding Tournament 選択を参考に新たに提案した，TS に基づく多点探索型多目的最適化手法を提案した。代表的なベンチマーク問題を用いた数値実験を通して，NSGA-II と比較し，有用性を確認した。

(2) メタヒューリスティクスの探索における代表的な探索戦略である多様化と集中化の観点から，(1) で提案した TS に基づく多点探索型多目的最適化手法について解析を行った。解析を踏まえて，探索の多様化・集中化を制御するパラメータとして，各探索点の近傍から新たな探索点となる解の最大許可数 n に注目した。n を，反復ごとに変化させるパラメータ n^k と解釈し，探索段階によってスケジューリングすることで，探索序盤は解空間を多様に探索し，探索終盤にかけて良い解周辺を集中的に探索することを狙う新たな移動戦略を提案した。代表的なベンチマーク問題を用いた数値実験を通して，提案した移動戦略の有用性を確認した。

(3) 最適化問題の解構造に広く見られる良構造の偏りである POP について，多目的最適化問題において新たな解釈を行った。多目的最適化における POP を基に，探索点の近傍生成および探索点の選択・移動において，評価値情報と距離情報に基づく探索点間の相互作用を有する，新たな多点探索型多目的組合せ最適化手法を提案した。代表的なベンチマーク問題を用いた数値実験を通して，提案手法の有用性を検証した。

7.2　今後の課題

今後の課題としては，以下のようなことが挙げられる。

- 3 目的以上の最適化問題，特に多数目的最適化問題に対する提案手法の適用
- 多目的最適化問題に対する多様化と集中化の適切なバランスの検証，および適切なバランスを実現するための移動戦略の提案
- より適切に多様化・集中化を調節するための指標・パラメータや，それらを用いた適応的な調整方法の開発
- 多目的最適化問題における POP の成立についての検証
- 多目的最適化問題における POP をより適切に活かした最適化手法の開発
- 以上から得られた見地に立脚した，新たな多目的組合せ最適化手法の開発

謝辞

本論文を作成するにあたり，日頃からご指導頂いている首都大学東京大学院理工学研究科電気電子工学専攻教授　安田恵一郎先生をはじめ，研究室の方々には，多くの御指導，御助言を頂き誠にありがとうございました。

特に本研究テーマを与えて下さった安田恵一郎先生には多大なる御指導を頂きました。深く感謝しております。

参考文献

[1] 相吉英太郎・安田恵一郎 編著：「メタヒューリスティクスと応用」，電気学会，オーム社 (2007)

[2] 安田恵一郎：「メタヒューリスティクスの現在と未来」，計測と制御，Vol.47，No.6，pp.453–458 (2008)

[3] D. E. Goldberg: “Genetic Algorithms in Search, Optimizations and Machine Learning”, Addison-Wesley Publishing Company (1989)

[4] 社団法人電気学会 進化技術応用調査専門委員会 編：「進化技術ハンドブック —第 I 巻 基礎編—」，近代科学社 (2010)

[5] F. Glover and M. Laguna: “Tabu Search”, Kluwer Academic Publishers (2002)

[6] 中山弘隆・岡部達哉・荒川雅生・尹禮分：「多目的最適化と工学設計 —しなやかシステム工学アプローチ—」，現代図書，星雲社 (2007)

[7] E. Aarts and J. K. Lenstra (Editor): “Local Search in Combinatorial Optimization”, A Wiley-Interscience Publication, (1997)

[8] J. Kennedy, and R.C. Eberhart: “Particle Swarm Optimization”, Proc. of IEEE International Conference on Neural Networks, pp.1942-1948 (1995)

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan: “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”, IEEE Transactions on Evolutionary Computation, Vol.6, No.2, pp.182–197 (2002)

[10] 柳浦睦憲・茨木俊秀：「組合せ最適化—メタ戦略を中心として—」，朝倉書店 (2001)

[11] M. P. Hansen: “Tabu Search for Multiobjective Optimization: MOTS”, Proceedings of 13th International Conference on Multiple Criteria Decision Making (MCDM ’97) (1997)

[12] H. Ishibuchi, T. Yoshida, and T. Murata: “Balance Between Genetic Search and Local Search in Memetic Algorithms for Multiobjective Permutation Flowshop Scheduling”, IEEE Transactions on Evolutionary Computation, Vol.7, No.2, pp.204–223 (2003)

[13] E. Zitzler and L. Thiele: “Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach”, IEEE Transactions on Evolutionary Computation, Vol.3, No.4, pp257–271 (1999)

[14] ETH SYSTEM OPTIMIZATION Web サイト
http://www.tik.ee.ethz.ch/sop/download/supplementary/testProblemSuite/

[15] TSPLIB Web サイト
http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

[16] Éric Taillard’s page Web サイト
http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html

[17] QAPLIB Home Page Web サイト
http://www.opt.math.tu-graz.ac.at/qaplib/

[18] E. Aarts, and J. Korst: “Simulated Annealing and Boltzmann Machines”, John Wiley & Sons (1989)

[19] 安田恵一郎・神内宏幸・石亀篤司：「距離に基づく相互作用を用いた多点探索型組合せ最適化手法」，電気学会 電子・情報・システム部門誌，Vol.130，No.1，pp.6–13 (2010)

[20] 矢口航太・田村健一・安田恵一郎・石亀篤司：「近接最適性原理の定量的評価に基

づく組合せ最適化手法」，電気学会電子・情報・システム部門誌，Vol.133，No.6，pp.1218–1228 (2013)

[21] J. I. Marden: “Analyzing and Modeling Rank Data”, Chapman & Hall (1995)

[22] 藤重悟：「グラフ・ネットワーク・組合せ論」，共立出版（2002）

[23] 山本芳嗣・久保幹雄：「巡回セールスマン問題への招待」，朝倉書店（1997）

[24] 久保幹雄・松井知己：「組合せ最適化 [短編集]」，朝倉書店（1999）

[25] 黒田充・村松健児（編）：「生産スケジューリング」，朝倉書店（2002）

[26] E. Taillard: “Benchmarks for Basic Scheduling Problems”, ORWP89/21 (1989)

[27] 久保幹雄・J.P.ペドロソ：「メタヒューリスティクスの数理」，共立出版（2009）

A　数値実験に用いた組合せ最適化問題

本研究では，複数の単一目的最適化問題を同時に解くことで，多目的最適化問題として扱っている。付録 A では，本論文の数値実験に用いた組合せ最適化問題である，0-1 ナップサック問題，巡回セールスマン問題，フローショップスケジューリング問題，二次割当問題について，単一目的最適化問題として説明する [23，24，25，26，27]。

A.1　0-1 ナップサック問題

ナップサック問題（Knapsack Problem）とは，各荷物 i に対する，重量 a_i，価値 c_i，および重量制限 b が与えられたとき，与えられた n 個の荷物集合 $V = \{1, 2, \cdots, n\}$ からいくつかを選び，選ばれた荷物の重量の合計が重量制限を超えないという条件の下で，価値の合計を最大にする問題である。この問題の解は，x_i をナップサックに入れる荷物 $i \in V$ の個数としたときに $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ で与えられる (x_i は非負整数)。特に，$x_i \in \{0, 1\}$，$i \in V$ の場合を 0-1 ナップサック問題（0-1 Knapsack Problem）と呼ぶ。0-1 ナップサック問題は次のように定式化される。

$$\begin{aligned} \text{max} \qquad & f(\boldsymbol{x}) = \sum_{i=1}^{n} c_i x_i \\ \text{subj. to} \qquad & \sum_{i=1}^{n} a_i x_i \leq b \end{aligned}$$

A.2　巡回セールスマン問題

巡回セールスマン問題（Traveling Salesman Problem）とは，n 個の都市の集合 $V = \{1, \cdots, n\}$ と都市 i と都市 j の間の距離 $d(i,j)$，$i, j \in V$ が与えられたとき，すべての街をちょうど一度ずつ訪問したあと元に戻る巡回路のうち，距離が最小になるものを求める問題である。この問題の解は，$e_{(i,j)}$ を都市 i と都市 j を頂点としたときの都市 i, j 間の辺とすれば，n 個の点のハミルトングラフ $\boldsymbol{x} = (V, \{e_{(x_1,x_2)}, e_{(x_2,x_3)}, \cdots, e_{(x_{n-1},x_n)}\})$，$x_i \in V$ で与えられる。巡回セールスマン問題は次のように定式化される。

$$\min \qquad f(\boldsymbol{x}) = \sum_{k=1}^{n-1} d(x_k, x_{k+1}) + d(x_n, x_1)$$

A.3　フローショップ・スケジューリング問題

フローショップ・スケジューリング問題（Flow-Shop Scheduling Problem）とは，与えられた n 個の製品集合 $V = \{1, \cdots, n\}$ を S 台の機械により順に加工する加工順序をなんらかの評価基準を最大化あるいは最小化するように決める問題である。n 個の製品はすべての機械で同じ順序で加工されるとし，各機械は一度に 1 つの製品しか加工できず，一度ある製品の加工にとりかかると，それが完了するまで他の製品は加工できない。

機械 i における製品 j の処理時間は $t_p(i,j)$ として与えられるものとする。この問題の解は x_k を k 番目に加工される製品としたときに，n 個の製品の加工順序 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ で与えられる。加工順序 $\boldsymbol{x}$ で加工した場合での各機械における各仕事の加工終了時間 $t_c(i,j)$ は次のように計算される。

$$\begin{aligned} &t_c(i, x_k) = \max\{t_c(i-1, x_k),\ t_c(i, x_{k-1})\} + t_p(i, x_k) \\ &\ i = 1, 2, \cdots, S, \quad k = 1, 2, \cdots, n \end{aligned}$$

ただし，$t_c(0,j) = t_c(i, x_0) = 0$ とする。本研究では，評価基準をすべての製品の全機械における加工が完了する時間とし，その最小化問題を考える。この場合，フローショップ・

スケジューリング問題は次のように定式化される。

$$\min \qquad f(\boldsymbol{x}) = t_c(S, x_n)$$

A.4　二次割当問題

二次割当問題（Quadratic Assignment Problem）とは，配置すべき n 個の施設の集合 $V = \{1, \cdots, n\}$ と n 箇所の地点があり，施設 i から施設 j への物の移動量 $a(i, j),\ i, j \in V$，および，地点 k から地点 l までの距離 $b(k, l),\ k, l \in V$ が与えられたとき，物の総移動距離を最小とする施設の配置を求める問題である。この問題の解は，地点 x_j に施設 i を配置することを表す順列 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ で与えられる。二次割当問題は次のように定式化される。

$$\min \qquad f(\boldsymbol{x}) = \sum_{i=1}^{n} \sum_{j=1}^{n} a(i, j) b(x_i, x_j)$$