多目的最適化

原田 圭　廣安 知之　日和 悟

2016年11月30日

IS Report No. 2017022306

IS Report

Medical Information
System Laboratory

Abstract

本稿では，多目的最適化の概要について述べる．まずは，多目的最適化問題の定義について述べたあと，その解法である多目的最適化手法について解説する．多目的最適化手法では，主に多目的遺伝的アルゴリズムについて述べる．また，多目的最適化手法によって得られる解の評価方法についても解説する．最後に，代表的なベンチマーク問題である多目的ナップサック問題について述べる．

目次

第 1 章　最適化問題

1.1　単目的最適化問題

単目的最適化問題とは，ある目的関数を最大化もしくは最小化する問題として定義され，m 個の制約条件を伴う単目的最適化問題は式 (1.1)～式 (1.3) のように定式化される．

$$\max(\min)\ f_1(\boldsymbol{x}) \tag{1.1}$$

$$\text{subject to } x_k \in \{0, 1\} \tag{1.2}$$

$$\text{s.t. } g_i \leq 0\ (i = 1, 2, ..., m) \tag{1.3}$$

ここで，式 (1.1) を目的関数といい，この目的関数を最大化もしくは最小化する設計変数を求めることになる．設計変数は式 (1.2) のように表すことができ，0 と 1 からのみ構成される離散値の場合の例を示している．設計変数を考える場合，連続値をとるものと離散値をとるものの大きく二種類に分別される．特に設計変数を離散値として扱い，設計変数の値に大小関係を持たない問題を組み合わせ最適化問題と呼ぶ．本稿では，この組み合わせ最適化問題を中心として，多目的最適化問題や多目的最適化手法，ベンチマーク問題の解説を行う．また，式 (1.3) は制約条件であり，制約条件下において目的関数を最大化させる設計変数の組み合わせを求める問題が，単目的最適化問題である．

1.2　多目的最適化問題

1.2.1　定式化

1.1 節では，単目的最適化問題について解説した．単目的最適化問題に対して，複数の目的関数を同時に最大化もしくは最小化する問題を多目的最適化問題と呼び，n 個の目的関数を持つ多目的最適化問題は式 (1.4)～式 (1.6) のように定式化される．

$$\max(\min)\ f_1(\boldsymbol{x}), \max(\min)\ f_2(\boldsymbol{x}), ..., \max(\min)\ f_j(\boldsymbol{x})\ (j = 1, 2, ..., n) \tag{1.4}$$

$$\text{subject to } x_k \in \{0, 1\} \tag{1.5}$$

$$\text{s.t. } g_i \leq 0 (i = 1, 2, ..., m) \tag{1.6}$$

多目的最適化問題とは，制約条件を満足し，複数の目的関数を同時に最適化するような設計変数を求める問題と定義される．単目的最適化問題との相違点は，式 (1.4) で示されるように取り扱う目的関数が複数になる点である．

このように，多目的最適化問題では，複数の目的関数を同時最適化しなければいけないため，以下の 2 点の特徴が挙げられる．一点目は，取り扱う目的関数によって単位が異なる場合がある点である．異なる単位を持つ目的関数は，単純な比較ができないという問題点をもつ．二点目は，複数の目的関数は，互いに競合する場合が存在するため，それらを同時に最適化することが困難である場合が多い

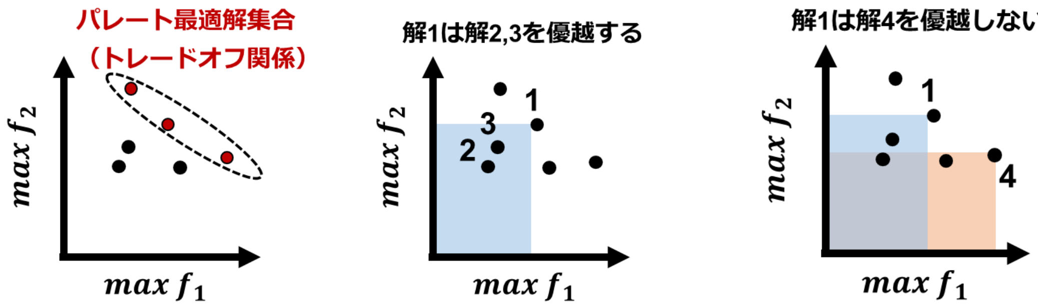
点である．これは，ある目的関数の改善を図ると，他方の目的関数が悪化し，両目的関数を同時に改善することができないことを意味し，これを一般的にトレードオフ関係と呼ぶ．そこで，次項ではこの問題点を解決するための概念である，パレート最適解について述べる．

1.2.2 パレート最適解

本項では，多目的最適化問題におけるパレート最適解の概念について述べる．単目的最適化問題では，解空間の中から最も優れた設計変数をもつ解が探索され，その解は最適解と呼ばれる．しかしながら，多目的最適化問題では，1.2.1 項でも述べた通り，複数の目的関数を同時に最適化することができず，目的関数間に競合が起きる．そのため，単目的最適化問題のように最適解が一意に決定するとは限らない．このため，多目的最適化問題では，他の解よりも劣っていない解集合を求めることが一般的で，これをパレート最適解，もしくは非優越解と呼ばれる．パレート最適解は，解同士の目的関数間の優越関係によって定義されている．ここで，優越とパレート最適解は次のように定義されている．

- 個体である $x_1, x_2 \in X$ に対して，個体が持つ目的関数 $f_i(x_1) \leq f_i(x_2)$ が成り立ち，さらに目的関数 $f_i(x_1) < f_i(x_2)$ の個体が存在する場合に，x_1 は x_2 に優越するという．
- 個体 x_0 に対して優越する $x_1 \in X$ が存在しない時，x_0 をパレート最適解といい，パレート最適解 x_0 の集合のことをパレート最適解集合という．

多目的最適化問題におけるパレート最適解に関して，トレードオフ関係を Fig. 1.1(a) に，解の優越を Fig. 1.1(b) に，非優越解を Fig. 1.1(c) に示した．多目的最適化問題においては，上記で定義されるパレート最適解集合を把握することが求められており，その手法の一つに多目的最適化手法が存在する．多目的最適化手法は，多目的最適化問題に存在するパレート最適解集合を一度に探索できる利点を持つ．次章では，多目的最適化手法の一つである多目的遺伝的アルゴリズムを中心とした解説を行う．



(a) トレードオフ関係 (b) 解の優越 (c) 非優越解

Fig. 1.1 多目的最適化問題におけるパレート最適解の概念 (自作)

第 2 章　多目的最適化手法

多目的最適化問題において，パレート最適解集合を求めるための一つに，多目的最適化手法が存在する．多目的最適化手法には，荷重和法，制約法などが存在するが，本章では多目的遺伝的アルゴリズムに焦点をあてる．

2.1　多目的遺伝的アルゴリズム

2.1.1　概要

遺伝的アルゴリズム [1] は，幅広い問題に対して効果的なメタヒューリスティックスなアルゴリズムで，生物の進化過程を模擬しているのが特徴である．生物は，親の遺伝子を複製し，その遺伝子に対して交叉や突然変異といった操作によって新たな遺伝子が生成される．その後，生成した遺伝子に対して，どの遺伝子を残し，次世代に引き継ぐかという生存選択が行われる．遺伝的アルゴリズムでは，このような生物の進化に基づいた手法をアルゴリズムに取り入れている．

遺伝的アルゴリズムは多目的最適化問題への新たな戦略の一つとして注目を集めており，多目的遺伝的アルゴリズム [2] と呼ばれている．多目的最適化問題では，競合する複数の目的関数に対して同時に最適化を行う必要があるため，探索空間が複雑となり，パレート最適解集合を求めることが困難である．この問題に対して，遺伝的アルゴリズム特徴である多点的探索が効果的であり，この理由から多目的遺伝的アルゴリズムの研究が進められた．

2.1.2　多目的遺伝的アルゴリズムの定義

多目的遺伝的アルゴリズムは 1985 年に Schaffer によって提案されたベクトル評価型遺伝的アルゴリズム (vector Evaluated Genetic Algorithm:VEGA)[3] が始まりである．その後，Goldberg によって提唱されたランキング法 [4] の導入などによって研究が進められてきた．多目的遺伝的アルゴリズムは，以下のような要求に基づいて定義されている．

- パレート最適解に対して評価と選択を適切に行い，次世代に残すこと．
- パレート最適解に対して多様性の維持を図る選択の戦略を行うこと．
- 交叉や突然変異などの遺伝子操作に関わる工程において，パレート最適解を効果的に生成される仕組みが導入されていること．

上記の要求は，多目的最適化問題が単目的最適化問題と違い，パレート最適解という概念が存在するためである．パレート最適解集合を求める際には，その複雑な問題構造を効率的に探索し，解空間に対して幅広い探索を行うことが必要である．そのために，上記のような要求に従ったアルゴリズムとして構成されることが重要となる．

2.1.3　多目的遺伝的アルゴリズムのプロセス

本項では，多目的遺伝的アルゴリズムのプロセスに述べ，その手順を以下に示す．

Step 0 初期パラメータの設定

遺伝子数，母集団数，交叉率，突然変異率，世代数や評価数といった終了条件の設定を行う．

Step 1 初期個体の生成

Step 0 にて設定したパラメータに基づいて，決定した遺伝子数からなる個体を母集団数分だけ生成する．

Step 2 初期個体の評価

生成した初期個体に対して，取り扱う各目的関数による評価を行う．

Step 3 パレート最適解の生成

各目的関数によって評価された値を基にパレート最適解を生成する．

Step 4 遺伝的操作による子個体の生成

母集団からランダムで 2 個体を選択する．選択された 2 個体に対して，染色体の情報の一部を交換する交叉という操作を行う．この交叉の行うかどうかは Step 0 で設定した交叉率に依存する．交叉によって新たに生成された個体は，個体の染色体の一部をランダムで変更する突然変異が行われ，子個体が生成される．突然変異を行うかどうかは，Step 0 で設定した突然変異率に依存する．生成した子個体は，目的関数に従って評価される．

Step 5 パレート最適解の更新

母集団をもとに生成されたパレート最適解に対して，生成した子個体の集団を基にパレート最適解の更新が行われる．

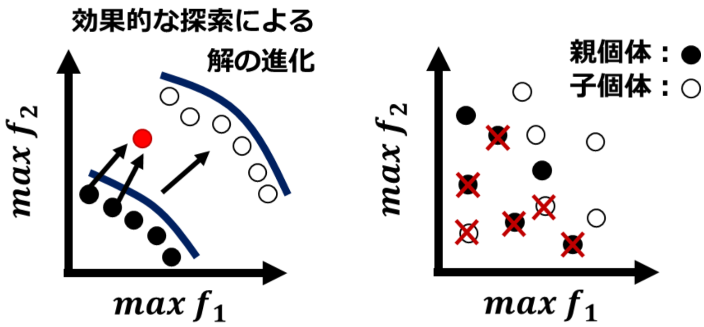
Step 6 次世代の母集団となる個体の選択

母集団と生成した子個体の集団から，次世代の母集団となる個体の選択を行う．この時，前項で述べた多目的遺伝的アルゴリズムの要求に従った，多様性を維持した手法が導入される．

Step 7 終了条件の確認

Step 0 にて設定した終了条件を満たしているかどうかを判断する．満たしていなければ Step 4 へと戻り，そうでなければ探索を終了する．

一般的な多目的遺伝的アルゴリズムは上記のような流れで構成される．ここで，Step 4 における 2 個体の選択方法，選択した 2 個体の交叉方法，突然変異方法は多数提案されており，上記の操作が探索に与える影響は大きい．そのため，対象問題によって取り扱う手法を適宜変更する必要があると考えられる．交叉などによる効果的な探索による解の進化を Fig. 2.1(a) に示した．また，Step 6 における次世代の母集団の選択方法の探索に大きな影響を与える．多目的最適化問題では，より広範囲のパレート最適解集合の把握が求められるため，多様性を維持した探索が必要となる．そのため，母集団に基づいて各世代で探索が行われる多目的遺伝的アルゴリズムでは，次世代の母集団の決定方法に注意を払う必要がある．次世代の母集団の決定よる淘汰を示した図を Fig. 2.1(b) に示した．



(a) 解の進化　　(b) 淘汰

Fig. 2.1　多目的遺伝的アルゴリズムにおける要件 (自作)

2.1.4　提案されてきた多目的遺伝的アルゴリズム

多目的遺伝的アルゴリズムの研究は 1985 年頃から盛んになり始め，それ以降様々なアルゴリズムが提案されてきた．提案されてきたアルゴリズムは大きく三つに分類することができる．それは，パレートランキングに基づいたアルゴリズム，多様性の維持を考慮したパレートランキングに基づいたアルゴリズム，各目的関数の重み付加に基づいたアルゴリズムである．以下に各戦略の代表的なアルゴリズムを述べる．

第一世代 パレートランキングに基づいたアルゴリズム

1985 年に Schaffer によって提案された Vector Evaluated Genetic Algorithm(VEGA)[3)]

第二世代 多様性の維持を考慮したパレートランキングに基づいたアルゴリズム

1993 年に Fonsera らによって提案された Multiobjective Genetic Algorithm(MOGA)[5)]

1995 年に Deb によって提案された Non-dominated Sorting Genetic Algorithm(NSGA)[6)]

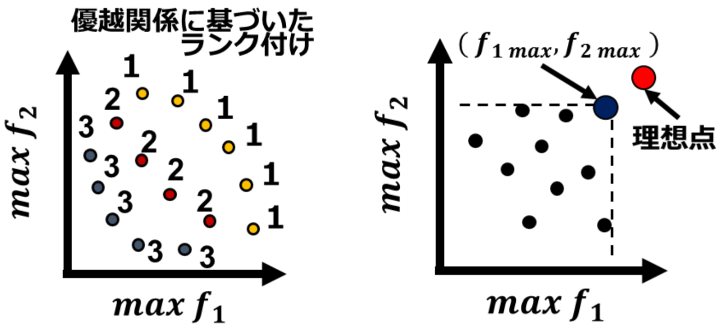
第三世代 多目的関数の重み付加に基づいたアルゴリズム

1998 年に Ishibuchi らによって提案された Random weight Genetic Algorithm(RWGA)[7)]

2002 年に Zitzle らによって提案された Strength Pareto Evolutionary Algorithm II(SPEAII)[8)]

2002 年に Deb によって提案された Non-dominated Sorting Genetic Algorithm II(NSGAII)[9)]

上記のようなアルゴリズムでは，パレートランキングによるアプローチ，もしくは多目的関数の重み付加に基づいたアプローチのいずれかが使用されている．パレートランキングによるアプローチを Fig. 2.2(a) に，多目的関数の重み付加に基づいたアプローチを Fig. 2.2(b) に示した．パレートランキングによるアプローチでは，解の優越関係に基づいて定められたランクによる選択が行われる．Goldberg によるランキング法[4)] が代表的なアプローチの一つである．多目的関数の重み付加に基づいたアプローチでは，多目的関数にそれぞれ重みを加えることで単一目的への変換が行わる．そして，各目的関数の最大値から理想点を割り出している．この理想点を基準として，理想点に対する重みを加えることで，単一目的への変換を行う．



(a) パレートランキング　(b) 多目的関数の重み付加

Fig. 2.2　多目的遺伝的アルゴリズムにおけるアプローチの違い (自作)

2.2　パレート最適解の評価方法

1 章では多目的最適化問題について定義し，2 章では多目的最適化手法について述べてきた．しかし，多目的最適化手法によって得られたパレート最適解集合に対して，単目的最適化問題のように最大値や最小値や中央値を算出して性能を比較するだけでは物足りない．そのため，パレート最適解集合を評価する以下の手法が存在し，各手法の概念図を Fig. 2.3 に示す．

解の優越支配に基づいた性能比較

概念図を Fig. 2.3(a) に示す．まず，比較を行う各アルゴリズムで得られた解を基に，パレート最適解集合が求められる．その後，求めたパレート最適解集合と各アルゴリズムで得たパレート最適解集合を比較し，精度やばらつきを確認することで性能を比較し，各アルゴリズムの評価を行う．また，求めたパレート最適解集合の個体数に対して，各アルゴリズムで求めたパレート最適解集合が選択されている割合を指標として性能評価を行う方法もある．

真のパレート最適解集合との誤差に基づいた性能比較

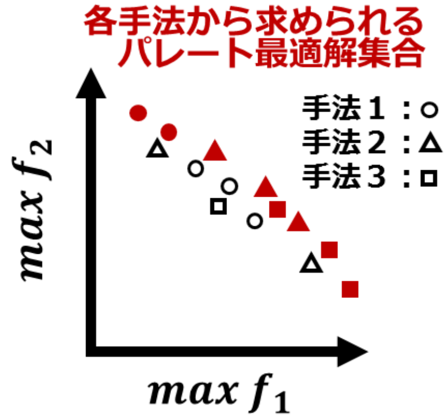
概念図を Fig. 2.3(b) に示す．多目的最適化問題において，真のパレート最適解集合が既知の場合にこの手法が有効的である．真のパレート最適解集合が既知である場合，各アルゴリズムによって得られるパレート最適解集合とのユークリッド距離の平均は誤差と考えることができる．この誤差が小さいほど，得られた解がより良い解集合であるとわかる．

被覆率に基づいた性能比較

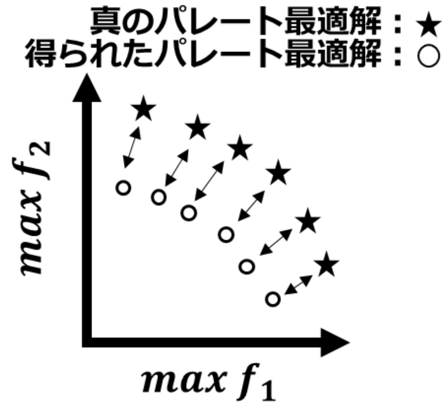
概念図を Fig. 2.3(c) に示す．パレート最適解集合の探索においては，パレート最適解集合に属する解の幅広さが重要な指標となる．そこで，解の幅広さ，ばらつきを示す指標として被覆率が挙げられる．この手法では，各目的関数における最大値と最小値を算出しておき，その間を任意に決定した分割数に基づいて分割する．分割した全領域数に対して，各アルゴリズムで得られたパレート最適解集合が存在する領域数から割合を求める．これを被覆率という．被覆率が 1 となれば，分割した全領域に解が存在することを意味するため，解が幅広く探索されていることがわかる．

得られたパレート最適解集合の幅広さに基づいた性能比較

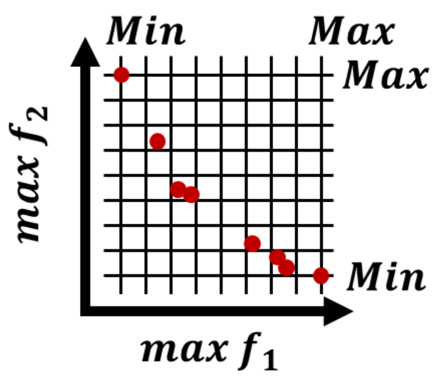
概念図を Fig. 2.3(d) に示す．各アルゴリズムで得られたパレート最適解集合において，各目的関数の最大値と最小値を求める．これによって，パレート最適解集合における幅広さの評価を行うことができる．



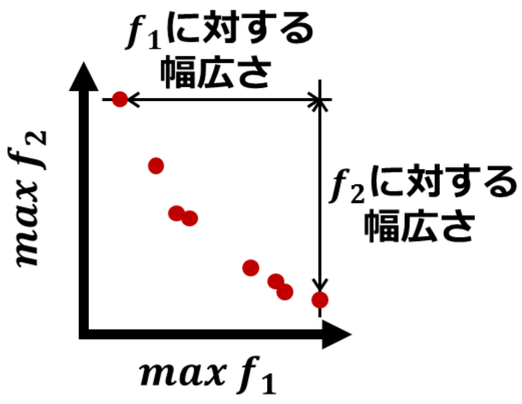
(a) 解の優越支配



(b) 真のパレート最適解集合との誤差



(c) 被覆率



(d) 得られたパレート最適解集合の幅広さ

Fig. 2.3　パレート最適解の評価方法 (自作)

第 3 章　ベンチマーク問題

多目的最適化手法に限らず，アルゴリズムの研究を進めるうえでベンチマーク問題による動作確認は重要である．新たなアルゴリズムを提案する場合などは，一般的に公開されているベンチマーク問題の結果をもとに性能を評価していることがほとんどである．そのため，アルゴリズムの性能を評価するために，それぞれの異なる特徴を持ったベンチマーク問題が今までに数多く提案されてきている．本章では，多目的最適化問題の代表的なベンチマーク問題である，多目的ナップサック問題について解説する．

3.1　ナップサック問題

多目的ナップサック問題は，目的関数が単一の場合のナップサック問題を拡張した問題であるため，まずは目的関数が単一の場合のナップサック問題について述べる．ナップサック問題 [10] は，組み合わせ最適化問題の一種であり，最適化手法のベンチマーク問題として知られる．ナップサック問題では，存在する複数のアイテムに対して，それぞれに価値と重量が与えられており，ナップサックに入る定められた総重量内で最も価値が高くなるようなアイテムの組み合わせを探索する．ナップサック問題は式 (3.1)～式 (3.3) のように定式化される．また，ナップサック問題の概念図を Fig. 3.1 に示す．

$$\max \quad \sum_{i=1}^{n} p_i x_i \tag{3.1}$$

$$\text{subject to} \quad \sum_{i=1}^{n} r_i x_i \leq b \tag{3.2}$$

$$\mathrm{x_i} = \{0, 1\} \quad (i = 1, ..., n) \tag{3.3}$$

式 (3.1)～式 (3.3) において，p_i はアイテム i の価値，r_i はアイテム i の重量を示している．また，x_i ではアイテム i の選択有無を表現しており，選択する場合は 1 を，選択しない場合は 0 を取る変数である．b はナップサックに格納できるアイテムの総重量を示している．式 (3.1) では，ナップサックに格納したアイテムの総価値を示しており，目的関数にあたる．式 (3.2) では，ナップサック問題に格納できるアイテムの総重量数を示しており，制約条件にあたる．式 (3.3) では，アイテム i に対して 1 から順番に番号が割り振られており，ナップサック問題における設計変数を示している．ナップサック問題では，式 (3.2) の条件を満たす時に，式 (3.1) で得られる総価値を最大化するアイテムの組み合わせ探索する問題である．

3.2　多目的ナップサック問題

多目的ナップサック問題では，複数のナップサックが用意されており，各ナップサックに格納するアイテムが同じ数だけ存在する．このため，アイテムに割り振られた番号を選択することで，各ナップサックに格納するためのアイテムの組み合わせが決定する仕組みとなっている．また，各ナップ

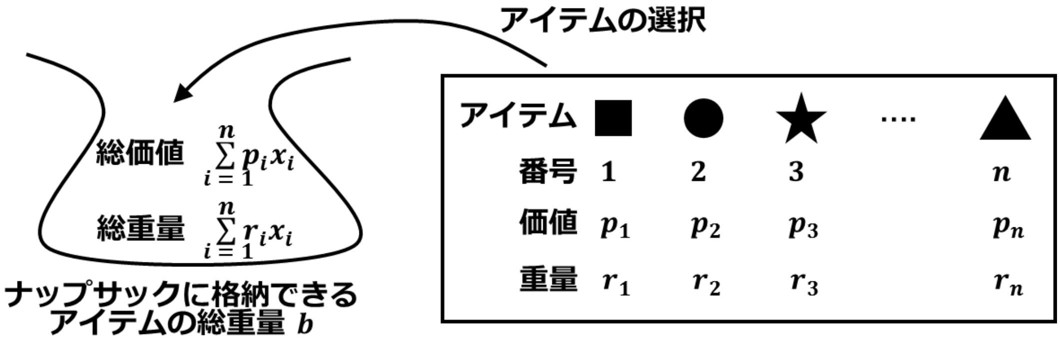


Fig. 3.1 ナップサック問題の概念図 (自作)

サックに対して制約条件となる，格納するアイテムの総重量は異なる．多目的ナップサック問題は，制約条件を満たした状態で，各ナップサックに格納するアイテムの総価値を最大化するアイテム番号の組み合わせを探索する問題である．m 個の目的関数を持つ多目的ナップサック問題は式 (3.4)～式 (3.6) のように定式化される．

$$\max\ f_k(x) = \sum_{i=1}^{n} p_i x_i\ (k = 1, 2, ...m) \tag{3.4}$$

$$\text{subject to}\quad g_k(x) = \sum_{i=1}^{n} r_i x_i \leq b_k\ (k = 1, 2, ..., m) \tag{3.5}$$

$$\mathrm{x_i} = \{0, 1\} \quad (i = 1, ..., n) \tag{3.6}$$

現在，多目的ナップサック問題は多くの種類が公開されており，例えばアイテム数が 100，250，500，750，ナップサック数が 2，3，4 といったものが公開されている．これらの問題に関する各アルゴリズムの性能比較は，多くの論文でも議論されており，代表的なベンチマーク問題であることが伺える．

また，多目的最適化問題のベンチマークには，多目的ナップサック問題のほかにもいくつか存在し，ZDT benchmark，DTLZ benchmark，WFG benchmark，LZ09 benchmark といった問題が挙げられる．

第 4 章　まとめ

本稿では，多目的最適化問題を中心に多目的最適化手法及びそのベンチマーク問題による解説を行った．多目的最適化手法では，その代表例である多目的遺伝的アルゴリズムについて述べ，定義や一般的な処理手順，そして提案されてきたアルゴリズムについて解説した．また，ベンチマーク問題として多目的ナップサック問題を取り上げた．これらの分野における研究はここ 30 年ほどで目まぐるしく成長しており，今後もより一層盛んになることが予測される．

参考文献

1) L. Davis, "Handbook of genetic algorithms," 1991.

2) K. Deb, "Multi-objective genetic algorithms: Problem difficulties and construction of test problems," *Evolutionary computation*, Vol.7, No.3, pp.205–230, 1999.

3) J.D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the 1st international Conference on Genetic Algorithms*L. Erlbaum Associates Inc., pp.93–100 1985.

4) J. Holland and D. Goldberg, "Genetic algorithms in search, optimization and machine learning," 1989.

5) C.M. Fonseca, P.J. Fleming, et al., "Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization.," in *Icga*, Vol.93Citeseer, pp.416–423 1993.

6) N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary computation*, Vol.2, No.3, pp.221–248, 1994.

7) H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol.28, No.3, pp.392–403, 1998.

8) E. Zitzler, M. Laumanns, L. Thiele, et al., "Spea2: Improving the strength pareto evolutionary algorithm," 2001.

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) S. Martello and P. Toth, Knapsack problems: algorithms and computer implementations, John Wiley & Sons, Inc., 1990.