

改良MOPSOによる工場エネルギー供給計画の多目的最適化

正員 北村 聖一* 正員 森 一之*

非会員 進藤 静一* 正員 泉井 良夫*

Modified Multi-objective Particle Swarm Optimization Method and its Application to Energy Management System for Factories

Syoichi Kitamura*, Member, Kazuyuki Mori*, Member, Seiichi Shindo*, Non-member,
Yoshio Izui*, Member

Conventional energy management systems (EMS) for factories have been developed to minimize energy costs. However we have to reduce energy consumptions and CO_2 emissions to preserve our earth environment. This paper proposes a multi-objective optimization method based on particle swarm optimization (PSO) to minimize energy costs and CO_2 emissions, and presents its effectiveness through simulation results.

キーワード：多目的最適化，メタヒューリスティクス，Particle Swarm Optimization，エネルギー管理システム

Keywords: Multi-objective Optimization, Metaheuristics, Particle Swarm Optimization, Energy Management System

1. はじめに

従来、工場向けエネルギー管理システム (energy management system: EMS) は、自家発電設備から構成されるエネルギー供給システムをエネルギーコストの最小化を目的とする非線形最適化問題として定式化し、数理計画法 (非線形混合整数計画法) により最適なエネルギー供給計画を立案していた⁽¹⁾⁽²⁾。しかし、今後の工場運営においては地球環境保全の観点から従来の目的に加え、一次エネルギー使用量と CO_2 などの温室効果ガスの排出量を削減することが必須となる。そのため、複数の目的を同時に考慮することが可能な多目的最適化に対応した EMS へのニーズが高まっている。

これに対応するため、多目的最適化問題を何らかのパラメータを用いて単目的の数理計画問題に変換する方法が考えられており、その代表的なものとして、各目的の加重和を全体の目的関数とする方法 (重み付け法)⁽³⁾、一つの目的以外を制約条件に埋め込む手法 (制約変換法)⁽⁴⁾ がある。しかし、エネルギー供給計画におけるエネルギーコスト、一次エネルギー使用量、 CO_2 排出量など目的関数の単位や次元が違う場合や、ある目的が他の目的にどのように影響しているのか明確ではない場合は、異なる目的関数に重み付けをすることは難しい。また、重み付けをすることで最適

解が一意的に定まってしまい、例えば目的関数 f_1 と f_2 がトレードオフの関係にある場合に、 f_1 をわずかに改悪するだけで f_2 が大きく改善されることがあっても、その見極めができないという問題が生じる。

これらの問題に対し、パレート解集合を一括して求めることができる多目的最適化手法として遺伝的アルゴリズム (genetic algorithm: GA) による手法が提案されている⁽⁵⁾。また、これをエネルギー供給計画問題へ適用した事例が報告されている⁽⁶⁾。一方、連続型の非線形最適化問題を効率良く解く手法として PSO (particle swarm optimization)⁽⁷⁾ が提案されており、コジェネプラントのエネルギー供給計画問題への有効性が確認されている⁽⁸⁾。また、PSO を多目的最適化手法として拡張した MOPSO (multi-objective particle swarm optimization)⁽⁹⁾ が提案されているが設計変数が多くなると最適解が得られにくいという PSO と同様の問題がある⁽¹⁰⁾。

本論文では、この問題を解決するため、エネルギー供給計画における多目的最適化問題に対し、原問題を部分問題に分割し、各部分問題のパレート最適解集合を求め、部分問題のパレート最適解を組み合わせることにより原問題のパレート最適解集合を求める多目的最適化手法 (改良 MOPSO) を新たに提案する。

本論文の構成は以下の通りである。第 2 章では、提案手法の基本となる MOPSO の概要を説明するとともに提案手法の詳細アルゴリズムについて説明する。第 3 章では、工場エネルギー供給計画問題を多目的最適化問題として定式化し、さらに重み付け法と MOPSO の比較を通して提案手

* 三菱電機（株）先端技術総合研究所

〒 661-8661 兵庫県尼崎市塚口本町 8-1-1

Advanced Technology R&D Center, Mitsubishi Electric Corp.
8-1-1, Tsukaguchi-Honmachi, Amagasaki, Hyogo 661-8661

法の有効性について議論する。

2. 多目的最適化手法の提案

PSO は非常に簡潔なアルゴリズムで構成されているが、連続型の非線形最適化問題を効率良く解くことができる手法で、単一の目的関数に対する最適化手法として近年注目されている。その PSO のアルゴリズムを多目的最適化問題を扱えるように改良した多目的最適化手法 MOPSO が提案されている。MOPSO は効率良く多目的最適解集合つまりパレート解集合を求めることができる。しかし PSO と同じ難点として、設計変数が多くなると指標関数的に計算時間が増大し、最適化が困難になることが経験的にわかっている。そこで本章では、次の 3 つのステップからなる手法、すなわち最適化問題を部分問題に分割し、MOPSO を用いて部分問題のパレート最適解集合を求め（第 1 ステップ）、これらを組み合わせて再構成することにより原問題のパレート最適解集合を求める（第 2 ステップ、第 3 ステップ）方法を提案する。

2・1 MOPSO MOPSO の概略アルゴリズムを図 1 に示す。また、図 1 の Step1-6 の詳細を図 2 に示す。MOPSO は、群状に分布し m 次元の探索空間を移動する探索点 $x \in R^m$ が、自身の持つ最良解の位置情報 $pbest \in R^m$ と群れで共有するパレート解の位置情報 $gbest \in R^m$ を使って移動ベクトル $v \in R^m$ を生成して解を探索し、最終的に残った $gbest$ の集合をパレート最適解集合とする手法である。MOPSO の探索方式を以下に示す。

Step1-1 初期設定 初めに探索点個数 N_I 、反復計算回数 N_K 、保存点個数最大値 N_R^{max} を決定し、 $x(i)$ 、 $pbest(i)$ 、 $gbest(r)$ 、 $v(i)$ の初期設定を行う。ただし、 i は探索点番号、 r は保存点番号を表す。 $x(i)$ ($1 \leq i \leq N_I$) は制限

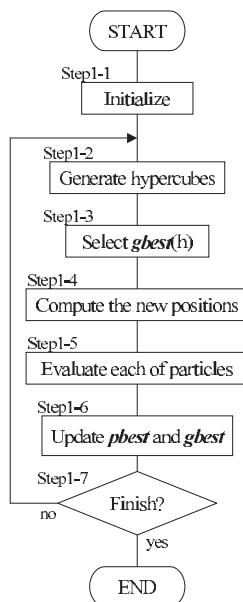


図 1 MOPSO のアルゴリズムの概略

Fig. 1. Algorithm of MOPSO.

値内で無作為に決定し、 $pbest(i) = x(i)$ ($1 \leq i \leq N_I$)、 $gbest(r) = x(i)$ ($r = i, 1 \leq r \leq N_I$)、 $v(i) = 0$ ($1 \leq i \leq N_I$) とおく。 $gbest(r)$ ($N_I + 1 \leq r \leq N_R^{max}$) については初期値を持たず、保存点個数を N_R とすると初期設定では $N_R = N_I$ である。

Step1-2 ハイパークリューブ生成 n 目的最適化問題 ($n >$

1) を扱う場合、探索点 $x(i)$ は n 個の目的関数値を持つので、それらの値によって n 次元の目的関数空間内の位

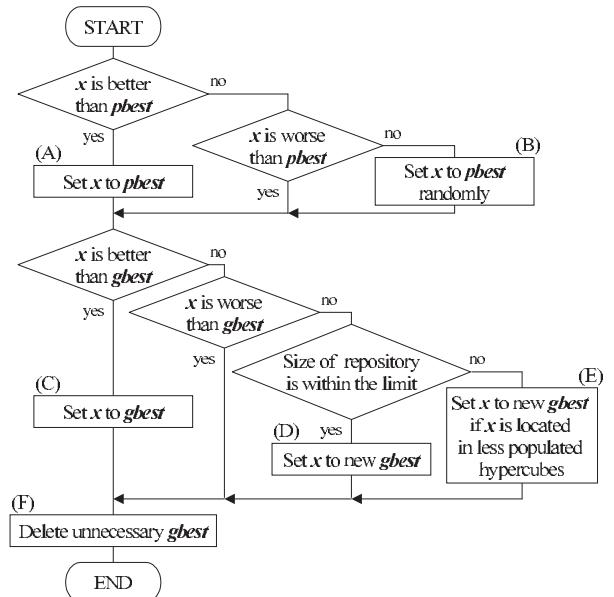


図 2 MOPSO における Step1-6 の詳細アルゴリズム

Fig. 2. Detailed algorithm of Step1-6 in MOPSO.

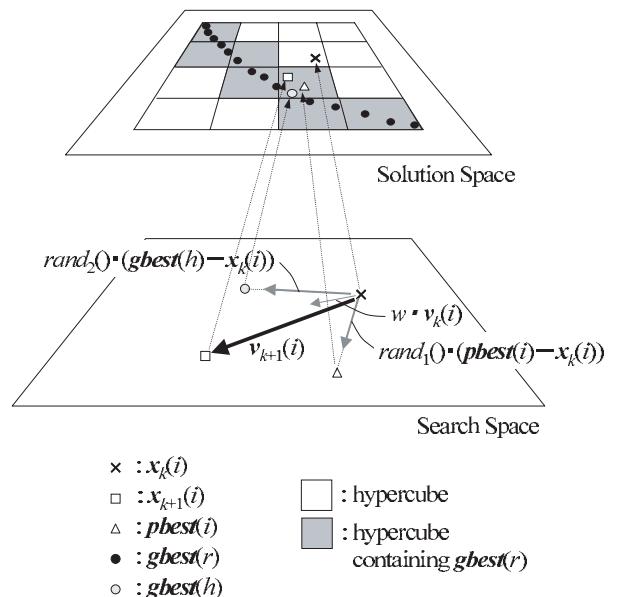


図 3 探索空間内の探索点および目的関数空間内の探索点とハイパークリューブの概念図

Fig. 3. Concepts of searching point in the search space and searching point and hypercubes in the solution space.

置が決まり、各探索点を評価することができる。これらの探索点自身が持つ最良解の位置情報 $pbest(i)$ と群れで共有するパレート解の位置情報 $gbest(r)$ も同様に n 個の目的関数値を持ち、 n 次元の目的関数空間内に存在することになる。MOPSO では、全ての $gbest(r)$ が存在する n 次元の目的関数空間を任意の数だけ分割するようにハイパーキューブ (n 次元立方体) を生成する。図 3 の上半分に目的関数空間内におけるハイパーキューブ生成の様子を示す。

Step1-3 $gbest(h)$ の選択 Step1-4 で移動ベクトル $v(i)$ を生成する際に必要となる $gbest(h)$ を選択する手順は以下の通りである。目的関数空間において、少なくとも 1 つの $gbest(r)$ を含む全てのハイパーキューブに着目し、各ハイパーキューブに属する $gbest(r)$ の個数を c とする。 $rand()$ を 0 から 1 までの一様乱数として、 $rand() / c$ が最大となるハイパーキューブを 1 つ指定し、ハイパーキューブ h とする。ハイパーキューブ h の中から $gbest(r)$ を 1 つ無作為に選択し、選択された $gbest(r)$ を $gbest(h)$ とする。このように $gbest(h)$ を選択することで、探索点を $gbest(r)$ の密度の低い領域へと引き寄せ、広範囲の探索を行うことができるという効果がある。

Step1-4 新たな探索点生成 $k+1$ 回目の探索において、探索空間内では i 番目の探索点 $x_{k+1}(i)$ が、式(1)で記述される移動ベクトル $v_{k+1}(i)$ に従って式(1)で示される新たな位置へと移動する。

$$\begin{aligned} v_{k+1}(i) = & w \cdot v_k(i) \\ & + rand_1() \cdot (pbest(i) - x_k(i)) \\ & + rand_2() \cdot (gbest(h) - x_k(i)) \end{aligned} \quad \dots \quad (1)$$

$$x_{k+1}(i) = x_k(i) + v_{k+1}(i) \quad \dots \quad (2)$$

式(1)に関して、 w は慣性値、 $rand_1()$, $rand_2()$ は 0 から 1 までの一様乱数を表す。右辺第 1 項は前回移動した方向への慣性を表すベクトルである。右辺第 2 項は探索点を自身の持つ最良解の位置へ引き寄せるベクトル、右辺第 3 項は探索点を $gbest(h)$ の位置へ引き寄せるベクトルである。なお、乱数 $rand_1()$, $rand_2()$ により多様な探索が実現可能となる。新たな探索点生成のイメージを図 3 の下半分に示す。

Step1-5 探索点評価 探索点 $x_{k+1}(i)$ の位置から目的関数値を求める。

Step1-6 $pbest(i)$, $gbest(r)$ の更新と保存 本ステップは以下の(A)~(F)のステップから成る。

Step1-6(A) $x_{k+1}(i)$ が全ての目的関数値に対して $pbest(i)$ よりも優れている場合は $pbest(i)$ を $x_{k+1}(i)$ に更新する。

Step1-6(B) $x_{k+1}(i)$ がある目的関数値に対して $pbest(i)$ よりも優れているが他の目的関数値に対して $pbest(i)$ よりも劣っている場合は $pbest(i)$ を $x_{k+1}(i)$ に更新するかどうかを無作為に決める。

Step1-6(C) 全ての目的関数値に対して $x_{k+1}(i)$ よりも劣るような $gbest(r)$ が存在する場合はそのうち 1 つを $x_{k+1}(i)$ に更新する。それ以外は後の処理(Step1-6(F))で削除される。

Step1-6(D) $x_{k+1}(i)$ が全ての $gbest(r)$ に対して少なくとも 1 つの目的関数値において優れている(つまり、パレート解である)場合、 $N_R < N_R^{max}$ であるならば、新たな $gbest(r)$ ($r = N_R + 1$) として $x_{k+1}(i)$ を保存する。またこのとき保存点個数が 1 つ増えるので保存点個数を $N_R + 1$ としておく。

Step1-6(E) Step1-6(D) の条件で $N_R \geq N_R^{max}$ であるならば、 $gbest(r)$ の個数がある値以下であるハイパーキューブに属する $x_{k+1}(i)$ のみ新たな $gbest(r)$ ($r = N_R + 1$) として保存する。またこのとき保存点個数が 1 つ増えるので保存点個数を $N_R + 1$ としておく。

Step1-6(F) 保存と更新の際に非パレート解になつた $gbest(r)$ を削除する。このときの保存点個数を新たに N_R とする。

なお、本文中の Step1-6(A)~(F) は図 2 の (A)~(F) に対応する。

Step1-7 探索終了 指定された反復計算回数 N_K に到達するまでは Step1-2~Step1-6 の手順を繰り返す。計算回数が N_K に到達すると探索を終了し、最終的に残った $gbest(r)$ の集合をパレート最適解集合とする。

2・2 提案手法 提案手法では、多目的最適化問題 \mathcal{P} を s 個の部分問題 $\{\mathcal{P}_1, \dots, \mathcal{P}_s\}$ に分割して解く。部分問題とは、原問題を構成する設計変数と目的関数の一部を含む問題とする。ただし、各部分問題の制約条件が他の部分問題に大きく依存する場合は、原問題を部分問題に分割することができない。ここでは各部分問題の制約条件が他の部分問題に依存しない場合を想定する。

第 1 ステップでは、各部分問題 $\mathcal{P}_a (a = 1, \dots, s)$ を 2・1 節で述べた MOPSO を用いて解く。求まった $gbest(r)$ の

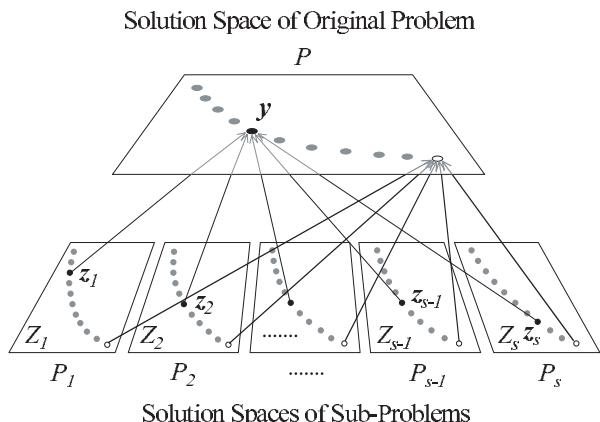


図 4 提案手法における部分解と全体解の概念図

Fig. 4. Concepts of sub-problems solutions and original problem solution in proposed method.

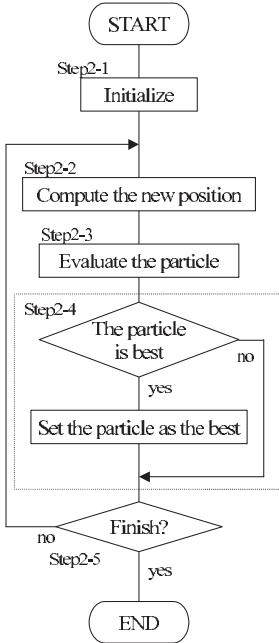


図 5 提案手法のアルゴリズム(第 2 ステップ)

Fig. 5. Algorithm of the second step in proposed method.

集合すなわちパレート最適解集合を改めて $\mathcal{Z}_a (a = 1, \dots, s)$ とおく。ただし、 \mathcal{Z}_a は n 次元の目的関数空間内の解集合であり、 $\mathcal{Z}_a \in R^n$ とする。 \mathcal{Z}_a の概念図を図 4 の下半分に示す。

第 2, 第 3 ステップでは、第 1 ステップで求まったパレート最適解集合 \mathcal{Z}_a を部分解集合とし、各部分解集合から部分解 $z_a \in \mathcal{Z}_a$ を 1 つずつ選択し組み合わせて解を再構成する。さらに組み合わせを改善することにより原問題すなわち全体問題の多目的最適化を行う。再構成された解(第 2, 第 3 ステップでは探索点として扱う)を y とし、概念図を図 4 の上半分に示す。ただし、 y も n 次元の目的関数空間内の点であり、 $y \in R^n$ とする。

2・2・1 第 2 ステップ 第 2 ステップでは、前述したようにまず第 1 ステップで求めた各部分問題 $\mathcal{P}_a (a = 1, \dots, s)$ の解 z_a を組み合わせて解を再構成することにより探索点 y を生成する。さらに探索点 y を構成する部分解 z_a の一部を入れ替えることで自身の持つ最良解よりも優れた解を探索する。第 2 ステップのアルゴリズムを図 5 に示す。

Step2-1 初期設定 全体最適化で組み合わせる s 個の各部分解集合 $\mathcal{Z}_a (a = 1, \dots, s)$ において、それぞれ部分解を同数(N_J 個ずつ)用意して(足りない場合は解を複製して)改めて \mathcal{Z}_a とおく。部分解を組み合わせるときにより広範囲で探索点の初期値を与えるために、各部分解集合 \mathcal{Z}_a において部分解を 1 つの目的関数値に従って並べ替え順番を付ける。並べ替えた後、各部分解集合 \mathcal{Z}_a の j 番目の部分解を $z_a(j) (1 \leq j \leq N_J) \in \mathcal{Z}_a$ とすると、全体最適化における j 番目の探索点の初期値 $y_0(j) (1 \leq j \leq N_J)$ は式(3)で表される。また、初期設定で反復計算回数 N_L

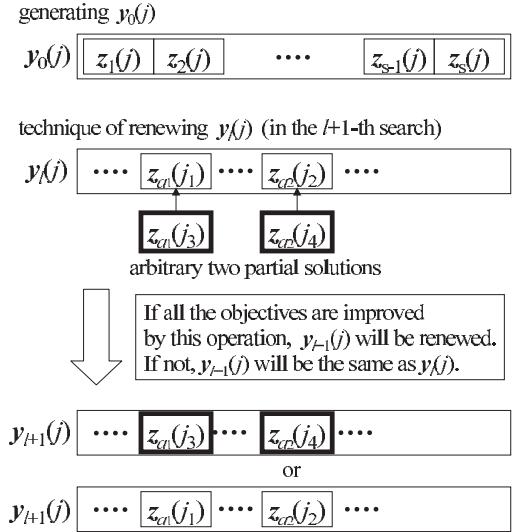


図 6 提案手法第 2 ステップにおける新たな探索点生成の概念図

Fig. 6. Concept of generating a new searching point in the second step.

を決定しておく。

$$y_0(j) = \sum_{a=1}^s z_a(j) \dots \dots \dots \quad (3)$$

Step2-2 新たな探索点生成 $l + 1$ 回目の探索において、式(4)に示すように、 l 回目の探索点 $y_l(j)$ を構成する部分解を無作為に 2 つ選択し、選択された部分解 $z_{a1}(j_1)$, $z_{a2}(j_2)$ を引き、その代わりに部分解集合 \mathcal{Z}_{a1} , \mathcal{Z}_{a2} からそれぞれ無作為に選択した部分解 $z_{a1}(j_3)$, $z_{a2}(j_4)$ を加えることで探索点 $y_{l+1}(j)$ を生成する。その様子を図 6 に示す。

$$y_{l+1}(j) = y_l(j) - (z_{a1}(j_1) + z_{a2}(j_2)) \\ + (z_{a1}(j_3) + z_{a2}(j_4)) \dots \dots \dots \quad (4)$$

この操作は、パレート解同士を入れ替えると、ある目的関数値は良くなるが他の目的関数値は悪くなるというパレート解の性質を生かしたものである。つまり定性的な例えとして 2 目的を扱う場合、 $z_{a1}(j_1)$ と $z_{a1}(j_3)$ を入れ替えることで目的関数 f_1 が大きく改善され、目的関数 f_2 が少し改悪され、一方 $z_{a2}(j_2)$ と $z_{a2}(j_4)$ を入れ替えることで f_1 が少し改悪され、 f_2 が大きく改善されるとすると、全体として 2 目的とも改善されるという特性を利用したものである。

Step2-3 探索点評価 探索点 $y_{l+1}(j)$ の各目的関数値を評価する。

Step2-4 $y_{l+1}(j)$ の更新方法 $y_{l+1}(j)$ が全ての目的関数値に対して $y_l(j)$ よりも優れていれば、 $l + 1$ 回目の探索点 $y_{l+1}(j)$ を自身の持つ最良解として更新する。それ以外は、 $l + 1$ 回目の探索結果として、 l 回目の探索点 $y_l(j)$ を自身の持つ最良解であるものとしてそのまま保持する。

Step2-5 探索終了 指定された反復計算回数 N_L に到達

するまでは Step2-2～Step2-4 の手順を繰り返す。計算回数が N_L に到達すれば探索を終了する。

2・2・2 第3ステップ 第3ステップでは、第2ステップで求めた最良解を改めて探索点 y として、パレート解の隙間を埋めるような解を探索する。群れで共有するパレート解として $rbest \in R^n$ を用いる。第3ステップのアルゴリズムを図7に示す。

Step3-1 初期設定 初めに反復計算回数 N_M 、保存点個数最大値 N_U^{max} を決定し、 $y(j)$ 、 $rbest(u)$ の初期設定を行う。ただし、 j は探索点番号、 u は保存点番号を表す。 $y(j)$ ($1 \leq j \leq N_J$) は第2ステップで最終的に求まった解とし、 $rbest(u) = y(j)$ ($u = j$, $1 \leq u \leq N_J$) とおく。 $rbest(u)$ ($N_J + 1 \leq u \leq N_U^{max}$) については初期値を持たず、探索点個数を N_U とすると初期設定では $N_U = N_J$ である。

Step3-2 ハイパーキューブ生成 2・1節のMOPSOと同様に全ての $rbest(u)$ が存在する範囲で任意の数だけハイパーキューブを生成する。

Step3-3 新たな探索点生成 式(5)に示すように $y(j)$ を構成する部分解のうち1つを無作為に選択し、選択された部分解 $z_{a_3}(j_5)$ を同じ部分解集合 \mathcal{Z}_{a_3} 内で無作為に選択した部分解 $z_{a_3}(j_6)$ に入れ替えることで探索点 $y'(j)$ を生成する。

$$y'(j) = y(j) - z_{a_3}(j_5) + z_{a_3}(j_6) \dots \dots \dots (5)$$

Step3-4 探索点評価 探索点 $y'(j)$ の各目的関数値を評価する。

Step3-5 $rbest(u)$ の更新、保存方法 生成した解が全体としてパレート解である場合、 $N_U < N_U^{max}$ であるな

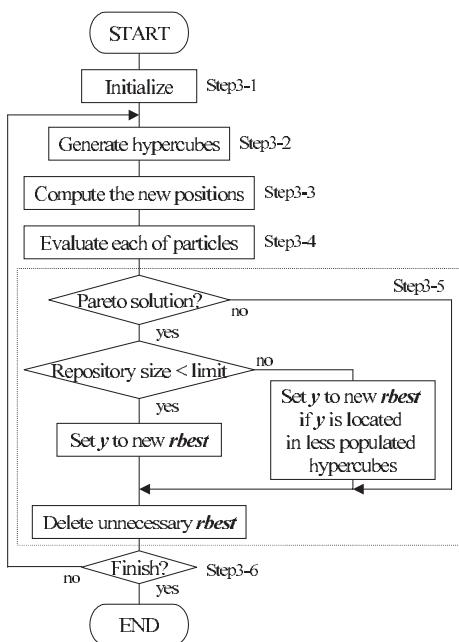


図7 提案手法のアルゴリズム（第3ステップ）

Fig. 7. Algorithm of the third step in proposed method.

らば新たな $rbest(u)$ ($u = N_U + 1$) として $y'(j)$ を保存し、 $N_U \geq N_U^{max}$ であるならば $rbest(u)$ の個数がある値以下であるハイパーキューブに属する $y'(j)$ のみ新たな $rbest(u)$ ($u = N_U + 1$) として保存する。このとき保存点個数が1つ増えるので保存点個数を $N_U + 1$ としておく。一方、生成した解が全体としてパレート解ではない場合、 $y'(j)$ は保存されない。また、保存、更新の際にパレート解でなくなってしまった $rbest(u)$ を削除し、このときの保存点個数を新たに N_U とする。

Step3-6 探索終了 指定された反復計算回数 N_M に到達するまでは Step3-2～Step3-5 の手順を繰り返す。計算回数が N_M に到達すると探索を終了し、最終的に残った $rbest(u)$ の集合をパレート最適解集合とする。

以上、第2ステップ、第3ステップを段階的に記述したが、これらを交互に何度か行う、または統合して行うことも可能である。

3. 工場エネルギー供給システムへの適用

本章では、エネルギーコストと CO_2 排出量を同時に最小化することを目的として、エネルギー供給システムの24時間における2目的運用計画問題を解く。全体最適化では24時間分の問題を一度に解くことになるが、設計変数が多くなるため最適解が得られにくい。一方、問題を1時間毎に分割し、提案手法を用いて各時刻の2目的運用計画問題を解いて24時間分の運用計画を立案することができる。その例を以下に記述する。ただし、時刻間の制約はないものとする。

3・1 定式化 典型的な工場のエネルギー供給システムを図8に示す。用役系は、1台のボイラと2台の蒸気タービン発電機（背圧タービン、抽気復水タービン）で構成される。

このモデルを定式化し、目的関数を式(6)、(7)、電力と蒸気のバランス制約式を式(8)～(18)とする。

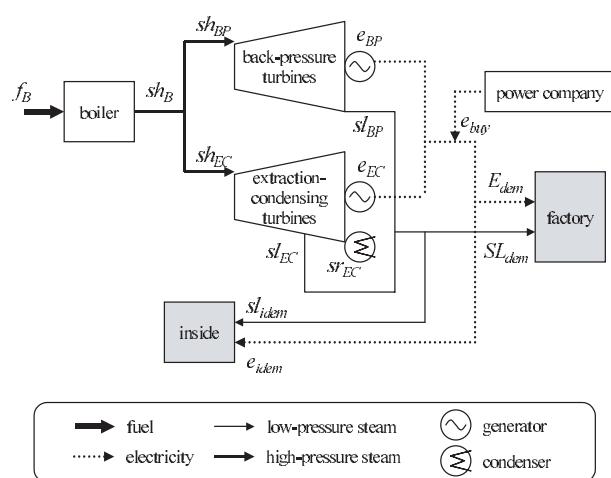


図8 エネルギー供給システム

Fig. 8. Model of cogeneration system.

$$\begin{aligned} \min . \quad o_{cost}(t) &= F_{uni} \times f_B(t) \\ &\quad + E_{uni}(t) \times e_{buy}(t) \end{aligned} \quad \dots \dots \dots (6)$$

$$\begin{aligned} \min . \quad o_{CO_2}(t) &= F_{CO_2} \times f_B(t) \\ &\quad + E_{CO_2} \times e_{buy}(t) \end{aligned} \quad \dots \dots \dots (7)$$

$$s.t. \quad sh_B(t) = p_B(f_B(t)) \quad \dots \dots \dots (8)$$

$$e_{idem}(t) = p_i(sh_B(t)) \quad \dots \dots \dots (9)$$

$$sl_{idem}(t) = q_i(sh_B(t)) \quad \dots \dots \dots (10)$$

$$sh_{BP}(t) = (1 - d(t)) \times sh_B(t) \quad \dots \dots \dots (11)$$

$$e_{BP}(t) = p_{BP}(sh_{BP}(t)) \quad \dots \dots \dots (12)$$

$$sl_{BP}(t) = q_{BP}(sh_{BP}(t)) \quad \dots \dots \dots (13)$$

$$sh_{EC}(t) = d(t) \times sh_B(t) \quad \dots \dots \dots (14)$$

$$e_{EC}(t) = p_{EC}(sh_{EC}(t), sl_{EC}(t)) \quad \dots \dots \dots (15)$$

$$\begin{aligned} sl_{EC}(t) &= SL_{dem}(t) \\ &\quad + sl_{idem}(t) - sl_{BP}(t) \end{aligned} \quad \dots \dots \dots (16)$$

$$sr_{EC}(t) = q_{EC}(sh_{EC}(t), sl_{EC}(t)) \quad \dots \dots \dots (17)$$

$$\begin{aligned} e_{buy}(t) &= E_{dem}(t) + e_{idem}(t) \\ &\quad - e_{EC}(t) - e_{BP}(t) \end{aligned} \quad \dots \dots \dots (18)$$

ここで、 $o_{cost}(t)$:エネルギーコスト、 $o_{CO_2}(t)$: CO_2 排出量、 F_{uni} : 燃料単価、 $E_{uni}(t)$: 購入電力単価、 F_{CO_2} : 燃料 CO_2 排出原単位、 E_{CO_2} : 購入電力 CO_2 排出原単位、 $f_B(t)$: ボイラ入力燃料量、 $e_{buy}(t)$: 電力購入量、 $E_{dem}(t)$: 工場電力需要、 $SL_{dem}(t)$: 工場低圧蒸気需要、 $sh_B(t)$: ボイラ出力蒸気量、 $e_{idem}(t)$: 用役系内電力需要、 $sl_{idem}(t)$: 用役系内低圧蒸気需要、 $d(t)$: $sh_B(t)$ の各タービンへの振り分け配分、 $sh_{BP}(t)$: 背圧タービン入力蒸気量、 $e_{BP}(t)$: 背圧タービン発電量、 $sl_{BP}(t)$: 背圧タービン排気量、 $sh_{EC}(t)$: 抽気復水タービン入力蒸気量、 $e_{EC}(t)$: 抽気復水タービン発電量、 $sl_{EC}(t)$: 抽気復水タービン抽気量、 $sr_{EC}(t)$: 抽気復水タービン復水量、をそれぞれ表す（小文字は変数、大文字は定数を表す）。また、 p , q は用役系内需要特性、あるいは各機器の入出力特性を表し p_B は 2 次関数、 p_i , q_i , p_{BP} , q_{BP} , p_{EC} , q_{EC} は 1 次関数である。なお、 t は時刻を表す。さらに、上下限値のある変数に関する制約式を式 (19)~(29) とする。

$$\underline{F}_B \leq f_B(t) \leq \overline{F}_B \quad \dots \dots \dots (19)$$

$$\underline{SH}_B \leq sh_B(t) \leq \overline{SH}_B \quad \dots \dots \dots (20)$$

$$0 \leq d(t) \leq 1 \quad \dots \dots \dots (21)$$

$$\underline{SH}_{BP} \leq sh_{BP}(t) \leq \overline{SH}_{BP} \quad \dots \dots \dots (22)$$

$$\underline{E}_{BP} \leq e_{BP}(t) \leq \overline{E}_{BP} \quad \dots \dots \dots (23)$$

$$\underline{SL}_{BP} \leq sl_{BP}(t) \leq \overline{SL}_{BP} \quad \dots \dots \dots (24)$$

$$\underline{SH}_{EC} \leq sh_{EC}(t) \leq \overline{SH}_{EC} \quad \dots \dots \dots (25)$$

$$\underline{E}_{EC} \leq e_{EC}(t) \leq \overline{E}_{EC} \quad \dots \dots \dots (26)$$

$$\underline{SL}_{EC} \leq sl_{EC}(t) \leq \overline{SL}_{EC} \quad \dots \dots \dots (27)$$

$$\underline{SR}_{EC} \leq sr_{EC}(t) \leq \overline{SR}_{EC} \quad \dots \dots \dots (28)$$

$$\underline{E}_{buy} \leq e_{buy}(t) \leq \overline{E}_{buy} \quad \dots \dots \dots (29)$$

ただし，“ $\underline{\quad}$ ”，“ $\overline{\quad}$ ”は各変数の上限値、下限値をそれぞれ表す。

3・2 シミュレーション実験 提案手法と比較するために、複数の目的に対して重み付けをして足し合わせ、単一の目的として解く重み付け法を用いた実験と、MOPSO を用いて全体最適解を一度に求める実験を行う。

3・2・1 重み付け法 3・1 節で定式化した 2 つの目的関数にそれぞれ重み付けをして、式 (30) に示す単一の目的関数を新たに作成する。ただし、 $0 \leq \alpha \leq 1$ とする。

$$\begin{aligned} \min . \quad o_{total}(t) &= \alpha \times o_{cost}(t) + (1 - \alpha) \times o_{CO_2}(t) \\ &\dots \dots \dots (30) \end{aligned}$$

$\alpha = 0.005 \times n (n = 0, 1, \dots, 200)$ として、全ての n について PSO を用いて各時刻の $o_{total}(t)$ を求める。各 n において、各時刻の $o_{total}(t)$ を構成する $o_{cost}(t)$, $o_{CO_2}(t)$ をそれぞれ 24 時間分足し合わせて o_{cost} , o_{CO_2} とする。各 n の (o_{cost} , o_{CO_2}) をプロットしパレート解集合を求める。なお PSO の探索点個数を 100, 反復計算回数を 100,000 とする。その結果、反復計算回数 200 の段階で、内点法 (interior point method) と逐次 2 次計画法 (sequential quadratic programming) の 2 種類の計画法で求めた解と同等の解が、反復計算回数 500 の段階でそれよりも最適な解が得られた。さらに 100,000 回反復計算を行ってもそれ以上は改善されなかった。200 回、500 回、100,000 回の反復計算に要する時間はそれぞれ約 35 分、約 80 分、約 15,000 分であった (CPU 900MHz のノート PC 使用、以下同様)。

3・2・2 MOPSO 2 つの目的関数 o_{cost} , o_{CO_2} を式 (31), (32) のように作成し、問題を部分化することなく 2・1 節で述べた MOPSO を用いて一度に全体最適化を行い、パレート解集合を求める。

$$\min . \quad o_{cost} = \sum_t o_{cost}(t) \quad \dots \dots \dots (31)$$

$$\min . \quad o_{CO_2} = \sum_t o_{CO_2}(t) \quad \dots \dots \dots (32)$$

なお、MOPSO の探索点個数を 40, 反復計算回数を 100,000, 保存点個数最大値を 200, 移動ベクトルの慣性値を 0.4, ハイパーキューブの個数を 49 とする。各パラメータについては、Coello が推奨する値⁽⁹⁾を参考にしている。計算時間は約 210 分であった。

3・2・3 提案手法 まず、各時刻における 2 つの目的関数 $o_{cost}(t)$, $o_{CO_2}(t)$ を 2・1 節で述べた MOPSO を用いて最適化し、各時刻のパレート解集合を求める。なお、MOPSO の探索点個数を 40, 反復計算回数を 3,000, 保存点個数最大値を 100, 移動ベクトルの慣性値を 0.4, ハイパーキューブの個数を 49 とする。各パラメータについては、Coello が推奨する値⁽⁹⁾を参考にしている。提案手法第 1 ステップ終了時における部分解集合を図 9 に示す。図 9 のグラフの値は、0 時の各目的関数値の最大値を 1 と換

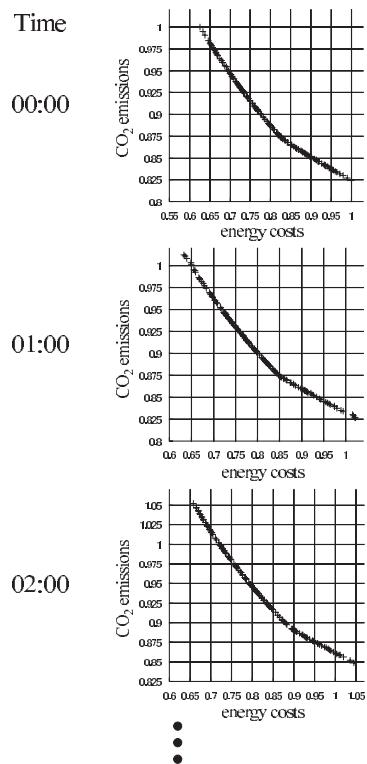


図 9 各時刻におけるパレート解集合

Fig. 9. Sets of pareto solutions at each time.

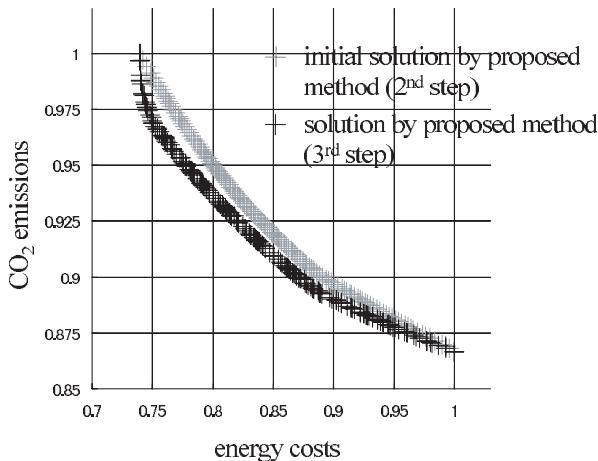


図 10 提案手法によるパレート解集合

Fig. 10. Set of pareto solutions by proposed method.

算したものである。図 9 から、第 1 ステップ終了時における部分解集合はパレート解集合になっていることがわかる。

次に 2・2 節の手法を用いて、上記の各時刻におけるパレート解集合を部分解集合として全体最適化を行う。なお、探索点個数を 121 個、反復計算回数を 100,000 回、保存点個数最大値を 200、ハイパーキューブの個数を 49 とする。第 1 ステップ開始から第 3 ステップ終了までの計算時間は約 105 分であった。得られた 1 日におけるパレート解集合を図 10 に示す。全時刻のパレート解を並べ替えて足し合せた 1 日における探索点の初期値（提案手法第 2 ステップ

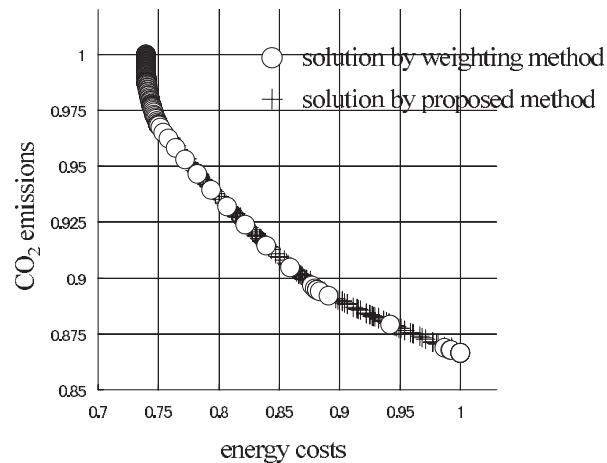


図 11 提案手法による解と重み付け法による解の比較

Fig. 11. Comparison between solutions by proposed method and weighting method.

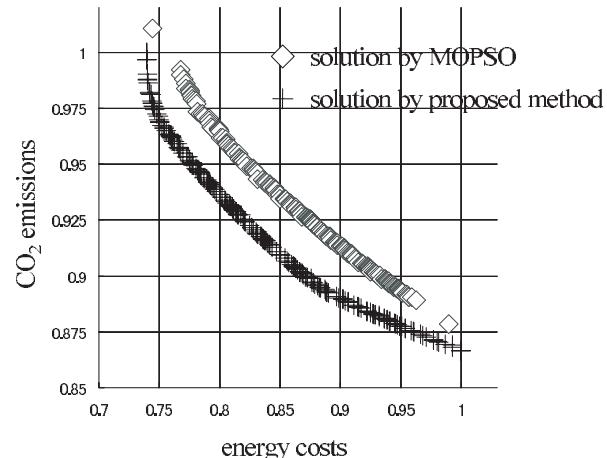


図 12 提案手法による解と MOPSO による解の比較

Fig. 12. Comparison between solutions by proposed method and MOPSO.

の初期設定における解）と、改善した最終的なパレート解（提案手法第 3 ステップ終了時の解）をそれぞれプロットする。図 10 のグラフの値は、提案手法で求めた各目的関数値の最大値を 1 と換算したものである。第 1 ステップ開始から第 3 ステップ終了までの計算時間は約 105 分であった。

3・2・4 計算結果の比較 提案手法と重み付け法を比較するため、提案手法によって得られたパレート解集合と重み付け法によって得られたパレート解集合を図 11 に示す。図 11 のグラフの値は、提案手法で求めた各目的関数値の最大値を 1 と換算したものである。図 11 から、重み付け法によって得られた解にはばらつきがあることがわかる。被覆率⁽³⁾⁽⁶⁾を比較すると、提案手法が 97.0 % であるのに対して重み付け法は 43.0 % と低い。被覆率からも提案手法がパレート解集合を効率良く得ることが可能であることがわかる。

次に提案手法によって得られたパレート解集合とMOPSOを用いて一度に全体最適化を行って求めたパレート解集合を図12に示す。図12のグラフの値は、提案手法で求めた各目的関数値の最大値を1と換算したものである。図12から、MOPSOでは問題の変数が多くなると全体最適化が困難となり、解の精度が提案手法よりも大きく劣ることがわかる。

4. おわりに

本論文では、複数の目的関数を扱う多目的最適化問題に対する手法として、改良MOPSOを提案し、重み付けをすることなくパレート最適解集合を求めることが可能であることを示した。さらに、MOPSOの問題点である変数を多く含む問題の最適化に対しても、原問題を部分問題に分割して部分問題のパレート最適解を求め、それらを再構成することにより、原問題のパレート最適解集合を求めることができるということを示した。改良MOPSOは全ての非線形最適化問題に対して適用可能ではないが、本論文で取り上げた工場エネルギー供給計画問題のように時間単位で分割できるような問題に対して有用であることを確認した。

今後の課題としては、部分問題間の制約が厳しい場合への対応と部分問題のパレート最適解の組み合わせ方法の効率化があげられる。

(平成16年4月27日受付、平成16年9月17日再受付)

文 献

- (1) 上田隆美・鈴木直彦：「エネルギー管理システム」，三菱電機技報，Vol.75, No.9, pp.29-32, (2001)
- (2) N. Suzuki, T. Ueda, and K. Sasakawa: "The Automatic Formulating Method of the Optimal Operating Planning Problem For Energy Supply Systems", IEEJ Transactions on Industry Applications, Vol.124, No.4, pp.366-372 (2004-4) (in Japanese)
鈴木直彦・上田隆美・笠川耕一：「エネルギー供給システム最適運転計画問題の自動定式化手法」，電学論誌D, 124, 4, pp.366-372 (2004-4)
- (3) T. Hiroyasu, M. Miki, and S. Watanabe: "Evalutionary Algorithms for Multiobjective Optimization", 13th SICE Symposium on Decentralized Autonomous Systems, pp.295-300 (2001) (in Japanese)
廣安知之・三木光範・渡邊真也：「進化的手法による多目的最適化」，計測自動制御学会第13回自律分散システム・シンポジウム, pp.295-300 (2001)
- (4) 中山弘隆・谷野哲三：多目的計画法の理論と応用，計測自動制御学会 (1994)
- (5) 三宮信夫・喜多一・玉置久・岩本貴司：遺伝アルゴリズムと最適化，朝倉書店 (1998)
- (6) T. Harada: "Diversified Evaluation for Energy Systems by Evolutionary Computation", SICE Conference on Systems and Information, pp.393-396 (2001-11) (in Japanese)
原田拓：「進化計算によるエネルギーシステムの多面的評価」，計測自動制御学会システム・情報部門学術講演会 2001, pp.393-396 (2001-11)
- (7) A. Ide, K. Yasuda, and N. Iwasaki: "Analysis of The Dynamics of Particle Swarm Optimization and Its Application to Global Optimization", Technical Meeting on System, SICE, pp.1-6 (2003-3) (in Japanese)
井出東・安田恵一郎・岩崎信弘：「Particle Swarm Optimization のダイナミクスの解析と大域的最適化への応用」，計測自動制御学会第28回システム工学部会研究会, pp.1-6 (2003-3)

- (8) S. Kitagawa, Y. Fukuyama, and Y. Todaka: "Particle Swarm Optimization for Optimal Operational Planning of a Cogen-eration Plant", 2004 National Convention Record IEE Japan, Vol.4, pp.S19(25)-(28) (2004-3) (in Japanese)
北川慎治・福山良和・戸高雄二：「Particle Swarm Optimization のコーディネート最適運用への適用」，平16電気学会全大, Vol.4, pp.S19(25)-(28) (2004-3)
- (9) Carlos A. Coello Coello : "MOPSO : A Proposal for Multiple Objective Particle Swarm Optimization", in Congress on Evolutionary Computation (CEC'2002), Vol.2, pp.1051-1056, IEEE Service Center, Piscataway, New Jersey, (2002-5)
- (10) N. Iwasaki, K. Yasuda, and A. Ide: "Application of Particle Swarm Optimization to Large-scale optimization problems", Technical Meeting of Industrial Instrumentation and Control in IEE Japan, pp.87-92 (2004-3) (in Japanese)
岩崎信弘・安田恵一郎・井出東：「Particle Swarm Optimization の高次元問題への適用に関する検討」，電気学会産業計測制御研資, pp.87-92 (2004-3)

北村聖一（正員）1977年4月30日生。2002年3月大阪大学大学院基礎工学研究科博士前期課程修了。同年4月三菱電機（株）入社。同社先端技術総合研究所勤務。工場エネルギー管理システム、生産シミュレーション、最適化手法の研究に従事。



森一之（正員）1962年4月2日生。1987年3月東京理科大学大学院理工学研究科修士課程修了。同年4月三菱電機（株）入社。同社中央研究所、産業システム研究所を経て現在先端技術総合研究所勤務。博士（工学）。離散事象システム、生産スケジューリング、工場計画の研究に従事。計測自動制御学会、IEEE SMC, EMSの会員。



進藤 静一（非会員）1957年9月22日生。1983年3月京都大学大学院工学研究科博士前期課程修了。同年4月三菱電機（株）入社。同社中央研究所、産業システム研究所を経て現在先端技術総合研究所勤務。上下水道の情報システム、ビルエネルギー管理システム、工場エネルギー管理システム、知的システムの研究に従事。情報処理学会、日本オペレーションズ・リサーチ学会各会員。



泉井良夫（正員）1959年2月2日生。1986年3月東京大学大学院工学系研究科博士課程修了。同年4月三菱電機（株）入社。同社中央研究所、産業システム研究所を経て現在先端技術総合研究所勤務。工学博士。主に、電力システムにおける計画・運用ならびに最適化に関する研究開発に従事。計測自動制御学会、IEEE等各会員。

