発電プラントにおけるタービン負荷配分の 大域的最適化*

江本 源一[†]・福島 雅夫[‡]

Global Optimization of Turbine Load Allocation in a Power Plant*

Genichi EMOTO[†] and Masao FUKUSHIMA[‡]

Turbine load allocation is a difficult optimization problem due to its non-smooth characteristic of efficiency. We formulate the problem as a non-smooth, non-convex optimization problem. We propose a practical and efficient global optimization algorithm, which is based on Lipschitz optimization method. The effectiveness of the algorithm is examined through computational experiments with a real world problem.

1. はじめに

多大なエネルギーを消費する化学、石油精製、鉄鋼、紙パルプなどのプロセス産業において省エネルギーを追求していくことはコスト競争力の維持、CO₂削減の観点から最重要な課題の一つである。なかでも、蒸気・電力を工場に供給し、コンビナートの中核に位置する自家発電プラントにおける省エネルギーの意義は大きく、様々な取り組みがなされてきた[1–5]。自家発電プラントの省エネルギーへの取り組みは、大きく分けて、設備改造によるものと運転の改善によるものに大きく分けられる。設備改造は老朽設備を効率の良い最新の設備に更新する事例が最も代表的であり、確実な効果が得られる代わりにその投資が非常に大きい[1]。一方、運転による改善とは、従来の運転方法を見直してより効率の良い運転を実現することであり、その代表的な例が最適化制御システムの導入によるプラントの運転最適化であり、発電プラントの最適化を行う際に、最も効果的な項目の一つがタービンへの最適負荷配分である。通常、自家発電プラ

ントでは数台のタービンを有しており、効率の良いタービンになるべく負荷をかけることによってタービン群としての発電効率を最大化することができる。しかしながらタービンは一般に非凸で微分不可能な効率特性[3]をもつことから、その取扱いが難しい。従来はタービン特性を簡易化した線形なモデルで表現して線形計画法を適用する例[5,6]や、厳密で非線形なモデルを用いても逐次2次計画法などの従来の最適化手法を適用する例[2–4]がほとんどであり、これらのアプローチでは局所的な最適解が得られるにすぎない。また、自家発電プラントでは時々刻々と蒸気・電力デマンドが変化していくため、運転データをオンラインで収集し、リアルタイムで最適化計算を行い、その計算結果を下位の制御システムに制御目標値として設定するリアルタイム最適化システム[4]が必要とされる。したがって計算時間の短縮も重要な課題の一つである。

本論文では、タービン負荷配分最適化のための一般的なタービン発電モデルを示し、それを大域的最適化手法の一つであるリプシッツ最適化法に問題の特徴にもとづいて改良を加えた、有効で現実的なアルゴリズムを提案する。

2. タービンのモデリング

蒸気タービンにおいて蒸気が可逆的断熱（等エントロピー）膨張をすると考えたときの理論仕事から内部損失を差し引いた残りを内部仕事といい、内部仕事と理論仕事との比を内部効率という。内部仕事から機械の摩擦などによる外部損失を差し引いた仕事は、有効に外部に伝

* 原稿受付 2006年5月29日

[†] 三菱化学(株) 技術部 プロセス生産技術開発室 Optimization Technology Group, Production Technology Department, Mitsubishi Chemical Corp.; 3-10 Ushiodori, Kurashiki city, Okayama 712-8054, JAPAN

[‡] 京都大学 情報学研究科 Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University; Yoshida-Honmachi, Sakyo-ku, Kyoto city, Kyoto 606-8501, JAPAN

Key Words: power plant, global optimization, real-time optimization, nonlinear optimization.

えられる仕事で，自家発電プラントでは，この仕事が発電機に伝えられ電力に変換される．この有効仕事と理論仕事との比を有効効率 $\eta_e \in (0, 1)$ という．これは一般に負荷に応じて高くなるという性質をもち [7]，

$$\eta_e = \frac{H_{in} - H_{out}}{H_{in} - I_{out}} \quad (1)$$

で定義される．ここで， H_{in} , H_{out} はおのおのタービン入口蒸気，出口蒸気のエンタルピーであり， I_{out} はタービン入口から出口まで等エントロピー膨張をした場合のタービン出口のエンタルピーである．Fig. 1 に示される背圧タービンでは，通過蒸気流量（以降，負荷とよぶ）を $x[t/h]$ ，発電量を $y[MW]$ とした場合，発電量は

$$y = K h x \eta_e \eta_g \quad (2)$$

で計算される．ここで， $\eta_g \in (0, 1)$ は発電機効率であり，負荷 (x) に関わらず，ほぼ一定の値をとるものと見なせるので本論文では以降，定数として扱う． K は単位換算係数である． h はタービン入口から出口までの等エントロピー膨張した場合の理想熱落差 $[kJ/kg]$ ，すなわち $h = H_{in} - I_{out}$ である．以降では $\eta = \eta_e \eta_g$ とおき， η をタービン効率とよぶことにする．実際のタービンでは入口に設置された複数の絞り弁の開閉状態によって生じる差圧の影響を受けて，負荷 (x) と発電量 (y) は Fig. 2 のように多峰性の特性を示すことが知られている [3]．Fig. 2 において効率特性が変化する点 b_1, b_2, b_3 は弁点とよばれる．以下ではこの圧損の影響はタービン効率に含まれる，すなわち η が多峰性の特性を示すものとして取り扱う．タービン効率は計測された蒸気の温度・圧力から (1) 式より求められるが，その値は

$$\eta = F(x) \quad (3)$$

のように負荷 (x) の関数として表現できる．ここで関数 F は一般に x の多項式として運転データにもとづいた重回帰分析によって求められる．これを以下ではタービン効率モデルとよび，最適化計算に用いる．

Fig. 1 Extraction turbine

最適化計算を行う際，タービン入口蒸気の温度 (t_{in}) ，圧力 (p_{in}) ，および出口の圧力 (p_{out}) は一定と考える．

Fig. 2 Example of turbine efficiency

したがって負荷 (x) を与えると，Fig. 1 における発電量 (y) は以下の手順から求められる．まず，タービン入口のエントロピー (s_{in}) を

$$s_{in} = \theta_S(p_{in}, t_{in}) \quad (4)$$

で求め，タービン入口から出口までの理想熱落差 (h) を

$$h = \theta_H(p_{in}, t_{in}) - \theta_I(s_{in}, p_{out}) \quad (5)$$

で求める．ここで， θ_S は蒸気のエントロピー， θ_H は蒸気のエントルピーであり，それぞれ温度 (t) と圧力 (p) の関数として表される [8]．また， θ_I はタービン入口から出口まで等エントロピー膨張した場合の出口蒸気のエントルピーであり，入口蒸気のエントロピー (s_{in}) と出口の圧力 (p_{out}) の関数である [8]．最後に以上から得られた h , η を用いて発電量 (y) は

$$y = K h x \eta \quad (6)$$

として計算される．現実のタービンは多段の場合が多く，次段の計算のためにタービン出口の温度 (t_{out}) まで求めておく必要がある．そのためには，タービン効率の定義

$$\eta = \eta_g \frac{\theta_H(p_{in}, t_{in}) - \theta_H(p_{out}, t_{out})}{h} \quad (7)$$

にモデル (3) で求められたタービン効率 η を代入して (t_{out}) を求めればよい．実際の自家発電プラントは，工場内での蒸気デマンド，電力デマンドの量，蒸気と電力の比率，負荷変動の大きさなどからタービンの構成，すなわち抽気・背気・復水タービンの組合せは様々なパターンがとられている．ただし，タービンの台数は一つのプラントで十数台を超える例はまれであり，蒸気ヘッダーの圧力レベルも三つ（高圧，中圧，低圧）が最も一般的であることを考えると，タービン負荷配分を最適化問題としてとらえた場合，最適化の変数の次元はたかだか数十のオーダーである．

3. タービン負荷配分最適化の定式化

この節では一般的なタービン負荷配分最適化問題を定式化する．Fig. 3 に示すような N 台の抽気/背気タービ

ンがあり、各タービンの最大段数が M 段の自家発電プラントを考える。

Fig. 3 General power plant

タービン i ($i=1,2,\dots,N$) の第 j 段 ($j=1,2,\dots,M$) の抽気 (背気) 量を $x_{i,j}$ とする。ここでは簡単のため各タービンの段数はすべて等しいと仮定する。実際にはタービンごとに段数が異なるのがふつうであるが、以下の議論をそのような場合に拡張するのは容易である。

抽気 (背気) 量 $x_{i,j}$ ($i=1,2,\dots,N, j=1,2,\dots,M$) が与えられたとき、タービン効率および発電量は以下の (8) ~ (12) 式を各タービンについて $j=1$ から M まで、順次解いていくことによって得られる。ここで表記を簡単にするため $\mathcal{I}=\{1,2,\dots,N\}$, $\mathcal{J}=\{1,2,\dots,M\}$ と表している。

$$\eta_{i,j} = F_{i,j} \left(\sum_{k=j}^M x_{i,k} \right) \quad (i \in \mathcal{I}; j \in \mathcal{J}) \quad (8)$$

$$s_{i,j-1} = \theta_S(p_{i,j-1}, t_{i,j-1}) \quad (i \in \mathcal{I}; j \in \mathcal{J}) \quad (9)$$

$$h_{i,j} = \theta_H(p_{i,j-1}, t_{i,j-1}) - \theta_I(s_{i,j-1}, p_{i,j}) \quad (i \in \mathcal{I}; j \in \mathcal{J}) \quad (10)$$

$$y_{i,j} = K h_{i,j} \eta_{i,j} \sum_{k=j}^M x_{i,k} \quad (i \in \mathcal{I}; j \in \mathcal{J}) \quad (11)$$

具体的にいえば、入口蒸気の条件 ($p_{i,j-1}, t_{i,j-1}$) がわかっているとすれば、モデル (8) 式からタービン効率 ($\eta_{i,j}$) が求められる (第1段では入口蒸気の圧力 ($p_{i,0}$) と温度 ($t_{i,0}$) は所与である)。一方、(9) 式から入口蒸気のエントロピー ($s_{i,j-1}$) が求められ、 $s_{i,j-1}, p_{i,j-1}, t_{i,j-1}, p_{i,j}$ から理想熱落差 ($h_{i,j}$) が (10) 式より求められる。最後に以上から計算された $h_{i,j}, \eta_{i,j}$ を用いて (11) 式より発電量 ($y_{i,j}$) が求まる。なお、すべての圧力 ($p_{i,j} \ i \in \mathcal{I}; j \in \mathcal{J}$) と発電機効率 (η_g) は固定とする。

さらに、次段の計算をするにあたって $t_{i,j}$ を求めておく必要があるため、前に得られた $h_{i,j}, \eta_{i,j}$ を

$$\eta_{i,j} = \left(\frac{\eta_g}{h_{i,j}} \right) \theta_H(p_{i,j-1}, t_{i,j-1}) - \left(\frac{\eta_g}{h_{i,j}} \right) \theta_H(p_{i,j}, t_{i,j}) \quad (i \in \mathcal{I}; j \in \mathcal{J}) \quad (12)$$

に代入して $t_{i,j}$ を得ておく。

ここで、(8) 式に現れる $F_{i,j}$ は通過蒸気量の区分的多項式関数であることに注意する (Fig. 2 参照)。ボイラーから発生する蒸気量と蒸気デマンドはいずれも一定であると仮定する。そのとき発電量最大化を目的とする複数のタービンへの負荷配分最適化問題は以下のように記述される。

$$\max \sum_{i=1}^N \sum_{j=1}^M y_{i,j} \quad (13a)$$

$$\text{s.t. } a_{i,j} \leq x_{i,j} \leq b_{i,j} \quad (i \in \mathcal{I}; j \in \mathcal{J}) \quad (13b)$$

$$\sum_{i=1}^N x_{i,j} = D_j \quad (j \in \mathcal{J}) \quad (13c)$$

$$A_{i,j} \leq \sum_{k=j}^M x_{i,k} \leq B_{i,j} \quad (i \in \mathcal{I}; j \in \mathcal{J}) \quad (13d)$$

ここで、(13b) 式は各抽気・背気量の上下限制約、(13c) 式は蒸気ヘッダーごとの蒸気デマンドに関する制約、(13d) 式は各タービン内での各段蒸気通過量の上下限制約である。さらに、各段での発電量 $y_{i,j}$ は (8) 式と (11) 式より $\mathbf{x} = (x_{i,j})_{i \in \mathcal{I}, j \in \mathcal{J}}$ の微分不可能な非線形関数として表されることに注意する。

4. リプシッツ最適化

プロセス最適化では、一般にプロセス特性が非線形であることから、逐次2次計画法などの非線形計画法が用いられることが多いが、これらの方法で得られる解は一般に局所的最適解である。非線形性の著しいタービン負荷配分最適化には複数の局所的最適解が存在することが予想されることから、複数の局所的最適解の中から真の最適解 (大域的最適解) を得る手法として大域的最適化が近年注目されている [9–11]。一般に大域的最適解を得ることは困難であるが、問題の特徴を利用することによって効率的に大域的最適解もしくはその近似解を得ることは可能である。実時間で最適解を得る必要があることから本論文では分枝限定法の考え方に基づく大域的最適化手法であるリプシッツ最適化 [9, 12–14] を用いて近似最適解を求め、さらに得られた解を出発点とする逐次2次計画法 (SQP 法) [16] に切り替えるハイブリッドアルゴリズムを提案する。説明を簡単にするため、まずつぎの区間制約 (箱型制約) をもつ問題に対するリプシッツ最適化アルゴリズム [9] を述べる。

$$\begin{aligned} \max \quad & f(x) \\ \text{s.t.} \quad & a \leq x \leq b \end{aligned} \quad (14)$$

ここで、 $x \in \mathbb{R}^n$, $a = (a_1, a_2, \dots, a_n)^T$, $b = (b_1, b_2, \dots, b_n)^T \in \mathbb{R}^n$, $a < b$ であり、 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ はリプシッツ関数、すなわち、ある定数 $L > 0$ に対して $|f(x) - f(y)| \leq L \|x - y\|$, $\forall x, y \in \mathbb{R}^n$ を満たすものとする。とくに、このアルゴリズムは関数 f の微分可能性を必要としないため本論

文で扱う問題に適している。

【リプシッツ最適化アルゴリズム】

Step0：初期化

$$\begin{aligned} R &\leftarrow \{x: a \leq x \leq b\}; \\ x_R &\leftarrow (a+b)/2, Q \leftarrow \{a, b, x_R\}; \\ v &\leftarrow \operatorname{argmax}\{f(x): x \in Q\}, \gamma \leftarrow f(v); \\ \mu(R) &\leftarrow \min\{\min\{f(a), f(b)\} + L\|b-a\|, \\ &\quad f(x_R) + L\|b-a\|/2\}; \\ \mu &\leftarrow \mu(R); \mathcal{M} \leftarrow \{R\}. \end{aligned}$$

Step1：大域的最適解近傍への到達度判定

$|\gamma - \mu| < \epsilon$ ならば終了; そうでなければ Step2 へ。
ただし, ϵ は適当な定数とする。

Step2：領域の2分割

$$\begin{aligned} b_j - a_j &= \max\{b_i - a_i: i = 1, \dots, n\} \text{ となる } j \text{ をえらぶ;} \\ a^1 &\leftarrow a, \\ b^1 &\leftarrow (b_1, b_2, \dots, b_{j-1}, (b_j + a_j)/2, b_{j+1}, b_{j+2}, \dots, b_n)^T; \\ a^2 &\leftarrow (a_1, a_2, \dots, a_{j-1}, (b_j + a_j)/2, a_{j+1}, a_{j+2}, \dots, a_n)^T, \\ b^2 &\leftarrow b; \\ R_1 &\leftarrow \{x: a^1 \leq x \leq b^1\}, R_2 \leftarrow \{x: a^2 \leq x \leq b^2\}; \\ x_{R_1} &\leftarrow (a^1 + b^1)/2, x_{R_2} \leftarrow (a^2 + b^2)/2. \end{aligned}$$

Step3：上界値の更新

$$\begin{aligned} \mu(R_1) &\leftarrow \min\{\mu(R), \min\{f(a^1), f(b^1)\} + L\|b^1 - a^1\|, \\ &\quad f(x_{R_1}) + L\|b^1 - a^1\|/2\}; \\ \mu(R_2) &\leftarrow \min\{\mu(R), \min\{f(a^2), f(b^2)\} + L\|b^2 - a^2\|, \\ &\quad f(x_{R_2}) + L\|b^2 - a^2\|/2\}. \end{aligned}$$

Step4：下界値の更新

$$\begin{aligned} Q &\leftarrow \{v, b^1, x_{R_1}, a^2, x_{R_2}\}; \\ v &\leftarrow \operatorname{argmax}\{f(x): x \in Q\}, \gamma \leftarrow f(v). \end{aligned}$$

Step5：分枝限定操作

$$\begin{aligned} \mathcal{M} &\leftarrow (\mathcal{M} \setminus \{R\}) \cup \{R_1, R_2\}; \\ \mathcal{R} &\leftarrow \{R \in \mathcal{M}: \mu(R) > \gamma\}; \\ \mu &\leftarrow \max\{\mu(R): R \in \mathcal{R}\}; \\ \mu(R) &= \mu \text{ を満たす } R \in \mathcal{R} \text{ を選び, 集合 } R \text{ の上下限} \\ &\text{を新たに } a, b \text{ とおき, Step1 に戻る.} \end{aligned}$$

このアルゴリズムにおいて, μ と γ はそれぞれ問題 (14) 式の最大値の上界値と下界値を表す。さらに $\mathcal{M}$ は分割された直方体の集合, $\mathcal{R}$ はそれらのなかでまだ終端されていない直方体の集合, すなわち活性部分問題の集合を表す。

一般にリプシッツ定数 (L) は未知であるので, リプシッツ最適化では, いかにして L を推定するかが鍵になる。タービン負荷配分の最適化はリアルタイム性が要求

されるため, 簡易的に L を推定し, 短時間で計算を終了させる工夫が必要である。本論文では, 計算の各段階において, その時点で対象となっている直方体 R の中心から各頂点までの傾きの最大値にパラメータ $k \geq 1$ をかけたものを L の推定値とした。すなわち,

$$L = k \max_i \left(\frac{|f(c) - f(d_i)|}{\|c - d_i\|} \right) \quad (15)$$

ただし, $c = (a+b)/2$, d_i ($i = 1, 2, \dots, 2n$) は直方体 R の頂点の集合であり, $d_i \in \{a \cup b\}$ である。また, 分枝限定法において, 部分問題の実行可能領域が小さくなっていった時, L はそれに応じて小さくなっていくよう, 子問題に対する L の推定値が親問題のそれよりも小さくなった場合のみ, リプシッツ定数の推定値を更新する処理を付け加えて計算効率の向上を図った。

つぎに, 区間制約に加えて, 一般的な制約条件が加わった最適化問題

$$\begin{aligned} \max \quad & f(x) \\ \text{s.t.} \quad & a \leq x \leq b \\ & g_i(x) \leq 0 \quad (i = 1, \dots, m) \end{aligned} \quad (16)$$

を考える。この問題に対しては, 下の (17) 式のようにペナルティ関数を導入して区間制約の問題に変換することによってリプシッツ最適化法が適用可能となる。

$$\begin{aligned} \max \quad & f(x) - P \sum_{i=1}^m \max\{0, g_i(x)\} \\ \text{s.t.} \quad & a \leq x \leq b \end{aligned} \quad (17)$$

ただし, $P > 0$ はペナルティパラメータである。問題 (16) 式の最適解 x^* において制約想定とよばれる正則条件が成立するとき, ペナルティパラメータ P がある程度大きい (具体的には問題 (16) 式の Karush-Kuhn-Tucker 条件を満たすラグランジュ乗数より大きい) ならば, x^* は問題 (17) 式の最適解になっていることがいえる [15]。とくに, ペナルティパラメータを無限に大きくしなくても, もとの問題の最適解が得られることから, このペナルティ関数は正確な (exact) ペナルティ関数とよばれる。ペナルティパラメータを非常に大きく選ぶ必要がないことは, リプシッツ最適化法を適用するうえで大きな利点である。

最適化問題 (13) 式をリプシッツ最適化法が適用可能な形式 (17) 式に変換することを考える。リプシッツ最適化において計算時間を左右する要因として探索空間の次元の大きさがある。問題 (13) 式において線形等式制約 (13c) 式に着目すれば探索する空間の次元を小さくすることができる。(13c) 式を次式で表す。

$$Cx = d \quad (18)$$

ただし, x は $x_{i,j}$ を成分とする MN 次元のベクトルである。(18) 式はベクトル x を M 次元のベクトル x_B と $MN - M$ 次元のベクトル x_N に分割したとき, 対応す

る行列 C の分割 (C_B, C_N) において C_B が正則となるように表現することができる。

$$C_B x_B + C_N x_N = d \quad (19)$$

上式を用いて問題 (13) 式から変数 x_B を消去することにより問題の次元を小さくできる。一方、制約条件 (13d) 式および変数 x_B に関する上下限制約 (13b) 式に対するペナルティ関数を定義する。ここで x_B に属する $x_{i,j}$ の添字集合を $\mathcal{B}$ とする。

$$g_1(x) = \max_{(i,j) \in \mathcal{B}} \{0, x_{i,j} - b_{i,j}\} \quad (20a)$$

$$g_2(x) = \max_{(i,j) \in \mathcal{B}} \{0, a_{i,j} - x_{i,j}\} \quad (20b)$$

$$g_3(x) = \max_{i \in \mathcal{I}, j \in \mathcal{J}} \left\{ 0, \sum_{k=j}^M x_{i,k} - B_{i,j} \right\} \quad (20c)$$

$$g_4(x) = \max_{i \in \mathcal{I}, j \in \mathcal{J}} \left\{ 0, A_{i,j} - \sum_{k=j}^M x_{i,k} \right\} \quad (20d)$$

ここで a_N を x_N に対応する $a_{i,j}$ からなるベクトル、 b_N を x_N に対応する $b_{i,j}$ からなるベクトルとし、 x_N を変数とする関数 $\tilde{y}$ と $\tilde{g}$ を (11) 式と (20) 式から (19) 式を用いて以下のように定義する。

$$\tilde{y}_{i,j}(x_N) = y_{i,j}(C_B^{-1}(d - C_N x_N), x_N)$$

$$\tilde{g}_k(x_N) = g_k(C_B^{-1}(d - C_N x_N), x_N)$$

以上より (13) 式で示した最適化問題は以下のように書き換えることができる。

$$\begin{aligned} \max \quad & \sum_{i=1}^N \sum_{j=1}^M \tilde{y}_{i,j}(x_N) - P \sum_{k=1}^4 \max\{0, \tilde{g}_k(x_N)\} \quad (21) \\ \text{s.t.} \quad & a_N \leq x_N \leq b_N \end{aligned}$$

この問題に対してリブシッツ最適化アルゴリズムを適用して大域的最適解の近似解を求める。リブシッツ最適化アルゴリズムではパラメータ ϵ の大きさによって反復回数や近似解の精度が決定される。現実的にはある程度大きな ϵ を用いて大域的最適解の粗い近似解を求め、それを初期値として SQP 法を適用することにより、短い計算時間で良い解が得られる可能性がある。ここではリブシッツ最適化アルゴリズムを用いて大域的最適解の近似解を得た後で、それを初期値として SQP 法を適用するアルゴリズムを **アルゴリズム 1** とよぶことにする。ただし、タービン効率関数 $F_{i,j}$ が通過蒸気量の区分的多項式関数となることから、(9)、(10) 式より問題 (21) 式の目的関数には微分不可能な点が存在するので、そのままでは SQP 法を適用できない。

そこで、各タービン各段の弁点の個数を $v_{i,j}$ 、($i \in \mathcal{I}; j \in \mathcal{J}$)、各弁点における負荷の値を $V_{i,j,l}$ ($V_{i,j,1} < V_{i,j,2} < \dots < V_{i,j,v_{i,j}}$) とすると、リブシッツ最適化アルゴリズムによって得られた大域的最適解の近似解 $(x_{i,j}^*)$ の存在する範

囲は以下のように記述される。なお、 $\mathcal{L}_{i,j} = \{1, 2, \dots, v_{i,j}\}$ とする。

$$\tilde{a}_{i,j} \leq x_{i,j}^* \leq \tilde{b}_{i,j} \quad (22)$$

$$\tilde{a}_{i,j} \in \{a_{i,j}, V_{i,j,1}, V_{i,j,2}, \dots, V_{i,j,l}\} \quad (l \in \mathcal{L}_{i,j})$$

$$\tilde{b}_{i,j} \in \{b_{i,j}, V_{i,j,1}, V_{i,j,2}, \dots, V_{i,j,l}\} \quad (l \in \mathcal{L}_{i,j})$$

したがって、SQP 法は問題 (13) 式において制約条件 (13b) 式を (22) 式に置き換えた問題に適用する。

5. 例題への適用

Fig. 4 に示すボイラー 1 缶、タービン 2 台からなる自家発電プラントのタービン負荷配分最適化問題を考える。

Fig. 4 Power plant example

タービン T1, T2 とともに高圧段と低圧段の 2 段からなる抽気背圧タービンであり、電力と共に工場に HPS (高圧蒸気) と LPS (低圧蒸気) を供給する。蒸気デマンド一定の条件下で各タービンに負荷を最適配分することによって発電量の最大化が図れる。また、タービン入口蒸気の温度・圧力および抽気、背気の圧力は一定と仮定する。各タービン高圧段、低圧段とも効率特性に弁点が存在するものとする。

$$\text{T1 高圧段弁点} \quad \sum_{i=1}^2 x_{1,i} = 100 \text{ [t/h]}$$

$$\text{T1 低圧段弁点} \quad x_{1,2} = 80 \text{ [t/h]}$$

$$\text{T2 高圧段弁点} \quad \sum_{i=1}^2 x_{2,i} = 110 \text{ [t/h]}$$

$$\text{T2 低圧段弁点} \quad x_{2,2} = 70 \text{ [t/h]}$$

また、抽気・背気量の上下限制約は以下の (23a), (23b) 式で規定され、タービン効率 $\eta_{i,j}$ はタービン通過蒸気量の 2 次の多項式で表されるものとする。

$$10 \leq x_{1,1} < 50, 50 \leq x_{1,2} \leq 100 \quad (23a)$$

$$10 \leq x_{2,1} < 50, 50 \leq x_{2,2} \leq 100 \quad (23b)$$

Fig. 5 に T1, T2 の高圧段タービン効率を、Fig. 6 に T1, T2 の低圧段タービン効率を図示する。本問題は独立変数 (x_N) の次元が 2 であるから、目的関数 (発電量) は 3 次元空間において図示できる。HPS デマンドを (D_1) 、LPS デマンドを (D_2) とすれば、デマンドが以下の 2 パターン

Case1 $(D_1, D_2) = (70, 160)$

Case2 $(D_1, D_2) = (65, 165)$

に従うときの合計発電量の等高線はそれぞれ Fig.7 と Fig.8 のように表される．図中の星印は大域的最適点を示す．Case1, Case2における大域的最適点はおのおの, $(x_{1,1}, x_{1,2}) = (30, 90), (25, 95)$ である．デマンドの変化に伴って，大域的最適点が変わっており，リアルタイムで大域的最適化が必要になることがわかる．

Case1 と Case2 の条件に対してアルゴリズム 1 と逐次 2 次計画法 (SQP 法) のみを用いて得られる解の比較を行った．この問題には弁点が存在し，微分不可能なため，そのままでは SQP 法が適用できない．そこでシグモイド関数を用いて弁点の近傍で効率式を平滑化することにより，微分可能関数で近似した．アルゴリズム 1 では Case1, Case2 いずれの場合も大域的最適点が得られた．一方，SQP 法のみを適用した場合は，計算結果は初期値に依存するので初期値を中心値，各頂点の計 5 点として実行したところ，Case1 での正解率（大域的最適点が得られた率）は 1/5，Case2 では 3/5 であった． $(x_{1,1}, x_{1,2})$ の初期値と最適化計算結果および目的関数値を Table 1 に示す．

なお，本計算実験において SQP ソルバーは MATLAB¹ Optimization toolbox を用いた．

Solid Line : T1 Dashed Line : T2

Fig. 5 Turbine efficiency (HPS)

Solid Line : T1 Dashed Line : T2

Fig. 6 Turbine efficiency (LPS)

Fig. 7 Distribution of power generation (Case1)

Fig. 8 Distribution of power generation (Case2)

Table 1 Computational results of test problem (SQP)

Case	初期値	計算結果	目的関数値
1	(20, 60)	(20, 80.8)	31.175
	(50, 60)	(30, 90)	31.397
	(20, 100)	(20, 100)	31.149
	(50, 100)	(50, 100)	30.682
	(35, 80)	(50, 100)	30.682
2	(15, 65)	(25, 95)	31.790
	(50, 65)	(39.2, 80.8)	31.656
	(15, 100)	(25, 95)	31.790
	(50, 100)	(50, 100)	31.204
	(32.5, 82.5)	(25, 95)	31.790

6. 現実問題への適用

つぎに，現実のタービン負荷配分最適化問題に対してアルゴリズム 1 を適用して，その有効性を評価した．対象となる自家発電プラントは複数のボイラーおよび 3 台の抽気背気タービンからなり，工場に電力とともに，高圧・中圧・低圧蒸気を供給している．すなわち $N=3, M=3$ である．タービン 1 からは高圧蒸気と低圧蒸気を，タービン 2 からは中圧蒸気と低圧蒸気を，タービン 3 からは高圧蒸気，中圧蒸気，低圧蒸気を工場に供給する．また，

¹MATLAB は Mathworks 社の登録商標です．

弁点はタービン1の低圧段に二つ、タービン2の中圧段に一つ、タービン3の中圧段に一つ、タービン3の低圧段に一つある。

リアルタイムでの最適化を想定した場合、モデルから求めたタービン効率(3)が運転実績の温度、圧力から求めたタービン効率に一致しているとは限らない。そこで前処理としてタービン効率モデルと実績効率の差

$$\alpha = \eta_g \frac{\theta_H(p_{in}, t_{in}) - \theta_H(p_{out}, t_{out})}{h} - F(x) \quad (24)$$

を計算し、最適化計算では α をバイアスとして

$$\eta = F(x) + \alpha \quad (25)$$

によりタービン効率を計算した。計算実験では5ケースの運転実績データに対して汎用数値計画法ソフトウェアとして定評があり、最新のバージョンにおいて大域的最適化の機能を追加した数値システム社のNUOPT¹と結果および計算時間を比較した。本問題は前節の問題と異なり、あらかじめ大域的最適解を知ることができないので、どれくらい良い解が得られたか、すなわち初期値から目的関数値(発電量)がどれくらい増加したかによって解の良否を評価した。NUOPTにおいては非線形計画問題に対して内点法やSQP法などが適用可能であるが、問題を記述するプログラミング言語SIMPLE¹においてoptions.method = "global"と記述することによって大域的最適化が適用される。NUOPTにおける大域的最適化は計算グラフを用いた凸緩和と分枝限定法による手法[11]が用いられている。

デマンドを5通りに変化させたときアルゴリズム1によって得られた結果をTable 2に示す。Table 2では上段に目的関数値、下段に計算時間を示しており、目的関数(発電量)の初期値は100である。Table 2より、Case4では同じ解が得られているが、その他のCaseではすべてアルゴリズム1が良い解を得ていることがわかる。また、計算時間はCase1を除いてアルゴリズム1のほうが短い。平均で見るとNUOPTでは発電量が1%向上しているのに対して、アルゴリズム1では発電量が1.4%向上している。一方、計算時間は平均でNUOPTの約1/20となっている。

NUOPTは汎用の大域最適化ソフトウェアであるが、この結果は少なくともタービン負荷配分最適化の問題に対してアルゴリズム1は十分有効なアルゴリズムであることを示している。

7. おわりに

本論文では発電プラントにおけるタービン負荷配分問題を数値最適化問題として定式化するとともに、リブシッツ最適化法とSQP法を組み合わせた大域的最適化

Table 2 Computational results for actual plant

Case	NUOPT	アルゴリズム1
1	101.519	101.982
	22.2[s]	37.5[s]
2	101.284	101.457
	308.2[s]	13.0[s]
3	101.038	101.475
	408.2[s]	13.2[s]
4	101.271	101.271
	807.5[s]	13.2[s]
5	100.074	100.885
	282.5[s]	17.4[s]
平均	101.037	101.414
	365.7[s]	18.8[s]

アルゴリズムを開発し、その有効性を検証した。とくに本アルゴリズムは汎用の大域的最適化手法に比べ計算時間を大幅に短縮できるため、リアルタイム最適化システムに適したものといえる。ここで考察したモデルは復水タービンを含んでいないが、復水タービンも含む一般的なモデルに対しても、本論文で提案した手法は容易に適用可能である。

タービン負荷配分問題におけるリブシッツ定数は未知であるが、本論文で提案した推定方法は計算実験の結果、いずれの例題に対しても正しい結果を導いていることから、妥当であることが確認された。また、大域的最適化手法はリブシッツ最適化法以外にも様々な方法が提案されているが、その一つであるNUOPTという汎用ソフトウェアとの比較については計算実験の結果を報告したように、得られる近似解や計算時間に関して提案手法が勝っている。その他の様々な大域的最適化手法の適用可能性については今後の課題としたい。

最後に、タービン負荷配分最適化へのNUOPT適用に関してご助言を頂いた、数値システム(株)の田辺隆人博士に深く感謝します。

参考文献

- [1] 池田：製紙工場に見るスチームタービンによる自家発電導入事例；省エネルギー，Vol. 49, No. 10, pp. 44-48 (1997)
- [2] 喜多：自家発電の用途と課題；省エネルギー，Vol. 49, No. 10, pp. 21-24 (1997)
- [3] 喜多：自家発電設備の省エネルギー運用を支える最適化技術；東芝レビュー，Vol. 53, No. 4, pp. 59-62 (1998)
- [4] G. Emoto, A. Tsuda, T. Takeshita, M. T. Monical, S. Nakagawa and K. Fujita: Integrated large-scale multi-variable control and real-time optimization of a power plant; *Proc. of the 1998 IEEE International Confer-*

¹NUOPT および SIMPLE は数値システム社の登録商標です。

- ence on Control Applications, pp. 1368–1372 (1998)
- [5] 西澤：発電プラントへの高度制御適用事例紹介；ボイラー研究, No. 334, 12月号, pp. 15–22 (2005)
 - [6] 村井：線形計画問題に対するアフィン変換法の一構成法と用役プラント最適化問題への応用；システム制御情報学会論文誌, Vol. 10, No. 2, pp. 62–69 (1997)
 - [7] 谷松：蒸気工学, 裳華房 (1999)
 - [8] 日本機械学会：蒸気表 (1968)
 - [9] R. Horst, P. M. Pardalos and N. V. Thoai: *Introduction to Global Optimization*, Kluwer Academic Publishers (2000)
 - [10] C. A. Floudas: *Deterministic Global Optimization Theory, Methods and Applications*, Kluwer Academic Publishers (2000)
 - [11] 山下, 逸見：大域的最適化の実用化に向けて；第17回 RAMP シンポジウム論文集, pp. 229–238 (2005)
 - [12] R. H. Mladineo: An algorithm for finding the global maximum of a multimodal, multivariate function; *Mathematical Programming*, 34, pp. 188–200 (1986)
 - [13] P. Hansen, B. Jaumard and S. H. Lu: Global optimization of univariate Lipshitz functions: Survey and properties; *Mathematical Programming*, 55, pp. 251–272 (1992)
 - [14] P. Hansen, B. Jaumard and S. H. Lu: Global optimization of univariate Lipshitz functions: New algorithm and computational comparison; *Mathematical Programming*, 55, pp. 273–292 (1992)
 - [15] 福島：非線形最適化の基礎, 朝倉書店 (2000)
 - [16] 茨木, 福島：最適化の手法, 共立出版 (1993)
-