プロセス産業における時系列最適化のための 逐次2次計画分解法*

江本 源一[†]・福島 雅夫[‡]

A Sequential Quadratic Programming Decomposition Method for Real-Time Optimization in Process Industries*

Genichi EMOTO[†] and Masao FUKUSHIMA[‡]

In the process industry, the closed-loop real-time optimization technology has become widespread with several commercial packages. However, the current technology mainly focuses on snapshot optimization, and it may not be used effectively in a time-variant process such as a reactor with catalyst degradation and a boiler with fouling growth. Ideally it is required to consider multi-period optimization that can handle future constraints and the total run-time cost on the process. The main difficulty in treating a multi-period process optimization problem resides in heavy computation due to the size of the problem. This paper proposes an efficient SQP algorithm that decomposes a QP subproblem into individual single-period QP problems by exploiting the particular structure of the multi-period optimization problem. Our numerical experiments indicate that the proposed SQP decomposition algorithm may be advantageous for problems with long time horizon.

1. はじめに

近年、プロセスのリアルタイム最適化 (Real-Time Process Optimization: RTO) は様々なコマーシャルパッケージの普及と共に産業界に広く普及しつつある [1,2]. RTO は定常状態においてプロセスを物理化学モデル (一般に非線形な代数方程式群として表現される) によって定式化し, さらにコスト最小化を達成するような目標関数を定義して逐次2次計画法などの非線形最適化手法を用いて最適な運転条件を求めることにより, それを運転員を介さずに下位の制御システムの目標値として自動的に設定するものである. 現在の RTO の技術はあるクラスの問題を解くうえにおいてはほぼ成熟していると考えられるが, さらに機能を向上させることにより, その適用範囲が一層広がる可能性を秘めている. 本論文では時間と共に特性が変化するプロセスに対して, 全運転期間に

おけるコストの最小化をいかにして実現するかという点に着目する.

従来の RTO は対象を定常状態としてとらえ, その時刻における最適化を行うものであった. しかしながら現実のプロセスでは, 運転時間と共に触媒の活性が劣化する反応器や, 運転時間と共に汚れが進行し効率が劣化するボイラーなど, 時間と共に特性の変化するプロセスが少なくない. そのようなプロセスに対して, 瞬時ベースの最適化を行っても, 全運転期間を考えた場合, かえってそれが触媒の劣化を早めたり, 汚れの進行を早めたりする可能性も考えられる. したがって, このようなプロセスを対象とする場合には, 将来の状態を予測しながら全運転期間を対象とした最適化を実現する必要がある. 時系列のプロセス最適化を実現する際に大きな障害になると考えられるのは, 問題規模の巨大化と時変プロセスを適切にモデル化することの困難さである. 一般に RTO における変数の規模は $10^3 \sim 10^5$ に及ぶとされている [3]. 一方, プロセスの連続運転時間は数十日~1, 2年に及ぶ. たとえば連続運転時間を1年として1日単位での時系列最適化を考慮すると, 問題の規模は $(10^3 \sim 10^5) \times 365$ となり, 解くこと自体が困難になり得る. その対策として簡易モデルを用いることなども考えられるが, 本論文では大規模最適化問題に逐次2次計画法 (SQP法) [4,5] を直接適用することを試みる. とくに SQP 法の各反復で

* 原稿受付 2001年1月26日

[†] 三菱化学 (株) 科学技術研究センター Science and Technology Research Center, Mitsubishi Chemical Corp.; 3-10 Ushiodori, Kurashiki city, Okayama 712-8054, JAPAN

[‡] 京都大学 大学院 情報学研究科 Graduate School of Informatics, Kyoto University; Yoshida Honmachi, Sakyo ward, Kyoto city, Kyoto 606-8501, JAPAN

Key Words: real-time process optimization, sequential quadratic programming method, decomposition.

解かれる2次計画問題が時系列構造を内包する凸計画問題になることに着目して、その双対問題 [6] を考えることにより、各時刻単位の部分問題に分割する方法を提案する。さらにボイラーの最適運転問題に対する計算実験により、提案手法の有効性を検証する。

2. 時系列最適化問題の定式化

時系列プロセス最適化の例としてボイラーの最適運転問題を考える。石油精製工場や石油化学工場において、電力と蒸気を供給する自家発電プラントの運転は、刻々と変化するデマンドや、昼夜で異なる電力体系など考慮すべき事項が多く、高度な意思決定が必要とされる [7]。その中でも重要な意思決定の一つは複数のボイラーへの負荷配分問題である。さらにボイラーへの排ガスからもたらされるスケールの付着によるボイラー効率の低下がこの問題を一層複雑にしている。

一般的なボイラーの例として単管ボイラー (Sulzer boiler) の概略プロセスを Fig. 1 に示す。Fig. 1 のボイラープラントにおいて、給水ポンプから供給される水 (ボイラー給水) は節炭器でボイラー排ガスによって予熱され、蒸発器においてバーナーからの火焰により飽和蒸気となる。飽和蒸気は汽水分離器においてその一部がドレン水として取り除かれた後、過熱器を経て超高圧蒸気 (主蒸気) となりタービンへ供給される。主蒸気はスプレー水と呼ばれるボイラー給水の一部を用いて一定の温度に制御される。

Fig. 1 A Sulzer boiler

ボイラーを物理現象にもとづいて厳密に表現すれば、熱交換器モデル、火炉燃焼モデル、火炉放射伝熱モデルなどから構成される複雑で大規模なモデルとなる。ボイラーの性能は熱交換器モデルとして表現される節炭器と過熱器の性能に大きく左右される。さらに熱交換器の性能は総括伝熱係数の値によってその性能が左右されるが [8]、スケールの付着などにより、汚れが堆積してくる (汚れ係数が上昇する) と総括伝熱係数は低下し、ひいてはボイラーの効率を低下させ、運転末期には経済的な運転が困難になるばかりでなく、種々の機械的制約の維

持も困難になることがある。

Fig. 2 に実際のボイラープラントにおける汚れ係数の時系列的な推移を示す。データは1年に1回行われる定期修理直後から次の定期修理に入るまでのものであり、サンプル間隔は5日である。R1は節炭器の汚れ係数、R2は1次過熱器の汚れ係数、R3は2次過熱器と3次過熱器の平均汚れ係数を示す。ただし、汚れ係数の値は規準化されている。46サンプルあたりで汚れ係数が急落しているのは、ボイラーの緊急停止によるサーマルショックによってスケールがはがれ落ちたためと思われる。

Fig. 2 Transition of fouling factor

汚れの進行は排ガスから発生するスケールの付着によるものであり、その汚れ係数の進行速度は下式のようなボイラー負荷 (蒸気発生量) の関数として表される。

$$\frac{dR}{dt} = a v_{steam} \quad (1)$$

ここで a は適当な係数、 v_{steam} は蒸気発生量、 R はボイラーの汚れ係数である。さらに統計的な解析により、排ガス中の残酸素濃度などの運転因子を説明変数とすることにより、その予測精度は向上する [9]。

ここで Fig. 3 に示されるような一つの主蒸気ヘッダーに複数のボイラーがつながっている系における負荷配分最適化問題を考える。Fig. 3 において、B1~B3はボイラー、T1~T3はタービン、F1~F3は燃料を示す。このような系において各ボイラーが主蒸気圧力制御を行えば干渉が発生するので、主蒸気圧力制御を行うボイラーは1基に限られる。その他のボイラーでは、直接燃料の量を変化させると、蒸発管出口のしめり度を一定に保つ制御システムがボイラー給水量を調整し、その結果としてボイラーの負荷が変化する。すなわち、複数のボイラーが存在する系においてボイラー負荷配分を左右する独立変数は主蒸気圧力制御を行っていないボイラーへの燃料供給量であり、Fig. 3 のボイラー系列においては負荷配分を決定する独立変数は F2, F3 となる。また、ボイラー負荷配分には影響しないが各ボイラーの運転条件を左右する独立変数として蒸発管出口モイスチャー、排ガス中残酸素濃度、副生燃料の混入率などがある。

Fig. 3 Simplified process diagram of a power plant with multiple boilers

このようなボイラー系列の運転最適化問題を考える際に物理モデルを用いる場合もあるが、実際にはボイラープラント運転を左右する因子を独立変数とした非線形回帰式がしばしば用いられる。ボイラー系列のモデルは一般につきのような非線形代数方程式の集合体として表現できる。

$$g_j(\mathbf{z}, \mathbf{R}) = 0, \quad j = 1, \dots, M \quad (2)$$

ただし、燃料流量は1種類と仮定し、 $\mathbf{z} = \{z_{li}(t)\}$, $l = 1, \dots, L$, $i = 1, \dots, N$, $t = 1, \dots, T$ とする。ここで、 t は時刻、 l はボイラー番号、 i はプロセス変数番号、とくに $i = 1$ は燃料流量、 $i = 2$ は蒸気発生流量を示す。また、 $\mathbf{R} = (\mathbf{R}_1, \dots, \mathbf{R}_L)$, $\mathbf{R}_l = (R_l(1), \dots, R_l(T))$ であり、 $R_l(t)$ はボイラー l の時刻 t における汚れ係数を示す。実際には、汚れ係数はボイラー単体においても節炭器、過熱器など熱交換設備ごとに個別に計算されるが、ここでは簡単のために一つにまとめて表現する。

(2) 式にもとづく一般的なボイラープラントの時系列での運転最適化問題は、各時刻における制約および時系列に関する制約（プロセス特性の時間に伴う変化）を満たしたうえで、全運転期間での燃料消費量の積算値を最小化する問題であり、以下のように定式化される。ここではボイラーが L 缶、各ボイラーの等式制約（プロセスモデル）が M 個、燃料流量を含む各ボイラーの独立変数が N 個存在するものとし、考慮すべき全運転期間は T 時刻先までと仮定する。さらに、各時刻で満足すべき蒸気デマンドを $D(t)$ とする。

$$\min_{\mathbf{z}, \mathbf{R}} \sum_{t=1}^T \sum_{l=1}^L z_{l1}(t)$$

subject to

$$\sum_{l=1}^L z_{l2}(t) = D(t), \quad t = 1, \dots, T \quad (3)$$

$$g_j(\mathbf{z}_l(t), R_l(t)) = 0, \quad j = 1, \dots, M, l = 1, \dots, L, t = 1, \dots, T \quad (4)$$

$$P_l \mathbf{z}_l(t) \leq \mathbf{r}_l, \quad l = 1, \dots, L, t = 1, \dots, T \quad (5)$$

$$R_{l_min} \leq R_l(t) \leq R_{l_max}, \quad l = 1, \dots, L, t = 1, \dots, T \quad (6)$$

$$R_l(t) = \mathbf{a}_l^T \mathbf{z}_l(t) + b_l \sum_{\tau=1}^t z_{l2}(\tau) + c_l, \quad l = 1, \dots, L, t = 1, \dots, T \quad (7)$$

ここで $\mathbf{z}_l(t) = (z_{l1}(t), \dots, z_{lN}(t))$, $l = 1, \dots, L, t = 1, \dots, T$ であり、 P_l , $\mathbf{r}_l$, R_{l_min} , R_{l_max} , $\mathbf{a}_l$, b_l , c_l は適当な定数（行列、ベクトル）である。上記の問題において、蒸気デマンドに関する等式制約条件 (3)、プロセスモデル g_j に関する等式制約条件 (4)、変数に関する不等式制約 (5), (6) は各時刻 $t = 1, \dots, T$ において全く同じ構造をもって出現する。これらを以下では同時刻制約と呼ぶ。一方 $\mathbf{R}_l$ に関する制約条件 (7) は経時変化を表すモデルであり、これを以下では時系列制約と呼ぶ。この問題はより一般的には次のように表現できる。

$$\begin{aligned} & \text{(P)} \min_{\mathbf{x}} \sum_{t=1}^T f_t(\mathbf{x}(t)) \\ & \text{subject to} \\ & \sum_{t=1}^T \mathbf{c}_t(\mathbf{x}(t)) = 0 \quad : \text{時系列制約} \\ & \mathbf{x}(t) \in X(t), \quad t = 1, 2, \dots, T : \text{同時刻制約} \end{aligned}$$

ここで、

$$\begin{aligned} \mathbf{x}(t) &= (\mathbf{z}_1(t), \dots, \mathbf{z}_L(t), R_1(t), \dots, R_L(t)) \\ f_t(\mathbf{x}(t)) &= \sum_{l=1}^L z_{l1}(t) \\ \mathbf{c}_t(\mathbf{x}(t)) &= (c_{t11}(\mathbf{x}(t)), \dots, \\ & \quad c_{tL1}(\mathbf{x}(t)), \dots, c_{t1T}(\mathbf{x}(t)), \dots, c_{tLT}(\mathbf{x}(t))) \\ c_{t\ell\tau}(\mathbf{x}(t)) &= \begin{cases} 0, & l = 1, \dots, L, \tau = 1, \dots, t-1, \\ R_l(t) - \mathbf{a}_l^T \mathbf{z}_l(t) - b_l z_{l2}(t) - c_l, & l = 1, \dots, L, \tau = t \\ -b_l z_{l2}(t), & l = 1, \dots, L, \tau = t+1, \dots, T \end{cases} \end{aligned}$$

である。さらに (5), (6) 式をあわせた制約条件を $V_l \mathbf{x}_l(t) \leq \mathbf{w}_l$ と表し、

$$\begin{aligned} X(t) &= \{x(t) \mid \sum_{l=1}^L z_{l2}(t) - D(t) = 0, g_j(\mathbf{z}_l(t), R_l(t)) = 0, \\ & \quad j = 1, \dots, M, l = 1, \dots, L, \\ & \quad V_l \mathbf{x}_l(t) \leq \mathbf{w}_l, l = 1, \dots, L\} \end{aligned}$$

とする。以下、 $\mathbf{h}_t(\mathbf{x}(t)) = \left(\sum_{l=1}^L z_{l2}(t) - D(t), g_1(\mathbf{z}_1(t), R_1(t)), \dots, g_M(\mathbf{z}_L(t), R_L(t)) \right)$ と表す。

3. 最適化アルゴリズム

時系列最適化問題PをSQP法を用いて解くことを考える。そのために問題Pに対するペナルティ関数 $F_r(\mathbf{x})$ とその $(\mathbf{x})$ における1次近似 $\bar{F}_r(\mathbf{x}, \mathbf{d})$ を次式により定義する。

$$\begin{aligned} F_r(\mathbf{x}) &= \sum_{t=1}^T f_t(\mathbf{x}(t)) + r \left\| \sum_{t=1}^T \mathbf{c}_t(\mathbf{x}(t)) \right\|_1 \\ \bar{F}_r(\mathbf{x}, \mathbf{d}) &= \sum_{t=1}^T f_t(\mathbf{x}(t)) + \sum_{t=1}^T \nabla f_t(\mathbf{x}(t))^T \mathbf{d}(t) \\ &\quad + \frac{1}{2} \sum_{t=1}^T \mathbf{d}(t)^T \mathbf{B}_t \mathbf{d}(t) \\ &\quad + r \left\| \sum_{t=1}^T (\mathbf{c}_t(\mathbf{x}(t)) + \nabla \mathbf{c}_t(\mathbf{x}(t))^T \mathbf{d}(t)) \right\|_1 \end{aligned}$$

ここで $r > 0$ はペナルティパラメータ、 $\mathbf{B}_t$, $t=1, \dots, T$ は正定値対称行列であり、 $\mathbf{x} = (\mathbf{x}(1)^T, \dots, \mathbf{x}(T)^T)^T$, $\mathbf{d} = (\mathbf{d}(1)^T, \dots, \mathbf{d}(T)^T)^T$, $\|z\|_1 = \sum_{j=1}^L |z_j|$ である。

SQP法のアロリズムは一般的に以下のように記述される[4,5]。

[SQP アルゴリズム]

ステップ 1 初期値 $\mathbf{x}(t)^{(0)}$, $t=1, \dots, T$ を選び、 $\mathbf{B}_t^{(0)} = \mathbf{I}$ (単位行列), $t=1, \dots, T$ とおく。ペナルティパラメータ $r > 0$ を設定する。定数 $\rho > 0$, $\beta \in (0, 1)$, $\gamma \in (0, 1)$, $\varepsilon > 0$ を選ぶ。 $k := 0$ とする。

ステップ 2 QP部分問題を解いて最適解 $\mathbf{d}^{(k)} = (\mathbf{d}(1)^{(k)}, \dots, \mathbf{d}(T)^{(k)})$ とラグランジュ乗数 $\mathbf{u}^{(k+1)}$ を求める。

ステップ 3 ペナルティパラメータ r を以下の手順で更新する。

$$r := \begin{cases} \max\{|u_i^{(k+1)}| : i=1, \dots, M\} + \rho, \\ \text{if } r < \max\{|u_i^{(k+1)}| : i=1, \dots, M\} \\ r, \text{ else} \end{cases}$$

ステップ 4 以下の手続きにより直線探索を行いステップ幅 $\alpha^{(k)} > 0$ を求め、 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}$ とおく。

- 1) $\alpha := 1$ とおく。
- 2) $F_r(\mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)}) < F_r(\mathbf{x}^{(k)}) + \beta \alpha [\bar{F}_r(\mathbf{x}^{(k)}, \mathbf{d}^{(k)}) - F_r(\mathbf{x}^{(k)})]$ が成立していれば $\alpha^{(k)} := \alpha$ として終了。
- 3) $\alpha := \gamma \alpha$ として2)へ戻る。

ステップ 5 つぎの修正BFGS公式を用いて近似ヘッセ行列 $\mathbf{B}_t^{(k)}$, $t=1, \dots, T$ を更新する。

$$\mathbf{B}_t^{(k+1)} = \mathbf{B}_t^{(k)} + \frac{\tilde{\mathbf{y}}(t)^{(k)} (\tilde{\mathbf{y}}(t)^{(k)})^T}{(\tilde{\mathbf{y}}(t)^{(k)})^T \mathbf{s}(t)^{(k)}}$$

$$\begin{aligned} & - \frac{\mathbf{B}_t^{(k)} \mathbf{s}(t)^{(k)} (\mathbf{s}(t)^{(k)})^T \mathbf{B}_t^{(k)}}{(\mathbf{s}(t)^{(k)})^T \mathbf{B}_t^{(k)} \mathbf{s}(t)^{(k)}} \\ \mathbf{s}(t)^{(k)} &= \mathbf{x}(t)^{(k+1)} - \mathbf{x}(t)^{(k)} \\ \mathbf{y}(t)^{(k)} &= \nabla_x L_t(\mathbf{x}(t)^{(k+1)}, \mathbf{u}^{(k+1)}) \\ & \quad - \nabla_x L_t(\mathbf{x}(t)^{(k)}, \mathbf{u}^{(k+1)}) \\ \theta_t &= \begin{cases} 1, & \text{if } (\mathbf{s}(t)^{(k)})^T \mathbf{y}(t)^{(k)} \\ & \geq 0.2 (\mathbf{s}(t)^{(k)})^T \mathbf{B}_t^{(k)} \mathbf{s}(t)^{(k)} \\ & 0.8 (\mathbf{s}(t)^{(k)})^T \mathbf{B}_t^{(k)} \mathbf{s}(t)^{(k)} \\ & \left\{ (\mathbf{s}(t)^{(k)})^T \mathbf{B}_t^{(k)} \mathbf{s}(t)^{(k)} - (\mathbf{s}(t)^{(k)})^T \mathbf{y}(t)^{(k)} \right\} \end{cases}, \text{ else} \\ \tilde{\mathbf{y}}(t)^{(k)} &= \theta_t \mathbf{y}(t)^{(k)} + (1 - \theta_t) \mathbf{B}_t^{(k)} \mathbf{s}(t)^{(k)} \end{aligned}$$

ただし, $\nabla_x L_t(\mathbf{x}(t), \mathbf{u}) = \nabla f_t(\mathbf{x}(t)) - \sum_{i=1}^M u_i \nabla c_{it}(\mathbf{x}(t))$, $t=1, \dots, T$ である。

ステップ 6 収束判定条件

$$\|\mathbf{x}(t)^{(k+1)} - \mathbf{x}(t)^{(k)}\| < \varepsilon (1 + \|\mathbf{x}(t)^{(k)}\|), \\ t=1, \dots, T$$

が満たされるならば終了。そうでなければ $k := k+1$ としてステップ2へ戻る。

SQPアルゴリズムのステップ2においてQP部分問題を解く際に、時系列問題特有の構造を活かし効率よく解くことを考える。SQPアルゴリズムの k 回目の反復におけるQP部分問題は下記のように表される。

$$\begin{aligned} (\text{QP}^{(k)}) \min_{\mathbf{d}} \sum_{t=1}^T & \left\{ \nabla f_t(\mathbf{x}(t)^{(k)})^T \mathbf{d}(t) \right. \\ & \left. + \frac{1}{2} \mathbf{d}(t)^T \mathbf{B}_t^{(k)} \mathbf{d}(t) \right\} \end{aligned}$$

subject to

$$\begin{aligned} \sum_{t=1}^T & \left\{ \mathbf{c}_t(\mathbf{x}(t)^{(k)}) + \nabla \mathbf{c}_t(\mathbf{x}(t)^{(k)})^T \mathbf{d}(t) \right\} = 0 \\ \mathbf{d}(t) & \in \hat{X}(t)^{(k)}, \quad t=1, \dots, T \end{aligned}$$

ただし,

$$\begin{aligned} \mathbf{c}_t(\mathbf{x}(t)^{(k)}) &= [c_{1t}(\mathbf{x}(t)^{(k)}), \dots, c_{Lt}(\mathbf{x}(t)^{(k)})]^T \\ \nabla \mathbf{c}_t(\mathbf{x}(t)^{(k)}) &= [\nabla c_{1t}(\mathbf{x}(t)^{(k)}), \dots, \nabla c_{Lt}(\mathbf{x}(t)^{(k)})]^T \\ \hat{X}(t)^{(k)} &= \left\{ \mathbf{d}(t) \mid \mathbf{h}_t(\mathbf{x}(t)^{(k)}) + \nabla \mathbf{h}_t(\mathbf{x}(t)^{(k)})^T \mathbf{d}(t) = 0, \right. \\ & \quad \left. V_l \mathbf{x}_l(t)^{(k)} - \mathbf{w}_l + V_l \mathbf{d}(t) \leq 0, \quad l=1, \dots, L \right\} \end{aligned}$$

である。以下では、簡単のため $\mathbf{g}_t^{(k)} = \nabla f_t(\mathbf{x}(t)^{(k)})$, $\mathbf{c}_t^{(k)} = \mathbf{c}_t(\mathbf{x}(t)^{(k)})$, $\mathbf{A}_t^{(k)} = \nabla \mathbf{c}_t(\mathbf{x}(t)^{(k)})^T$ と表す。

SQPアルゴリズムのステップ5における行列 $\mathbf{B}^{(k)}$ の更新方法は、行列 $\mathbf{B}^{(k)}$ の正定値性が常に保たれるようにBFGS公式を修正したものである。したがって、元の問題Pが非凸であっても、QP部分問題は必ず凸にな

るので、双対定理が成立する [6]. このことに着目して、 $QP^{(k)}$ を解く代わりに、その双対問題を解くことを考える. L 次元のラグランジュ乗数ベクトル $\mathbf{u}$ を用いてラグランジュ関数を次式で定義する.

$$L^{(k)}(\mathbf{d}, \mathbf{u}) = \sum_{t=1}^T \left(\mathbf{g}_t^{(k)T} \mathbf{d}(t) + \frac{1}{2} \mathbf{d}(t)^T \mathbf{B}_t^{(k)} \mathbf{d}(t) - \mathbf{u}^T (\mathbf{c}_t^{(k)} + \mathbf{A}_t^{(k)} \mathbf{d}(t)) \right) \quad (8)$$

そのとき、 $QP^{(k)}$ のラグランジュ緩和問題 $QP^{(k)}(\mathbf{u})$ は次のように表される.

$$(QP^{(k)}(\mathbf{u})) \min_{\mathbf{d}} L^{(k)}(\mathbf{d}, \mathbf{u})$$

subject to $\mathbf{d}(t) \in \hat{X}^{(k)}, \quad t=1, \dots, T$

$\hat{X}^{(k)} = \hat{X}^{(1)(k)} \times \dots \times \hat{X}^{(T)(k)}$ と表し、ラグランジュ緩和問題 $QP^{(k)}(\mathbf{u})$ の最小値を用いて関数 $\phi^{(k)}(\mathbf{u})$ を

$$\phi^{(k)}(\mathbf{u}) = \min_{\mathbf{d} \in \hat{X}^{(k)}} L^{(k)}(\mathbf{d}, \mathbf{u})$$

によって定義する. さらに $QDP^{(k)}$ の双対問題を次式によって定義する.

$$(QDP^{(k)}) \max_{\mathbf{u}} \phi(\mathbf{u})$$

以下では便宜上、双対問題 $QDP^{(k)}$ を等価な最小化問題に置き換え、表記を簡単にするため、添字 (k) を省略する.

$$(QDP) \min_{\mathbf{u}} \psi(\mathbf{u})$$

ただし $\psi(\mathbf{u}) = -\phi^{(k)}(\mathbf{u})$ である. ここで不等式制約条件も含むすべての制約条件をラグランジュ関数に取り入れて双対問題を構成する方法も考えられるが、期間数が大きくなるにつれて不等式制約の数も増えるため、双対問題の次元が著しく増大すること、および双対問題が変数の非負条件を含む制約つき問題となることを考慮し、本論文ではその方法は採用しなかった.

上記のラグランジュ緩和問題 $QP^{(k)}(\mathbf{u})$ と双対問題 QDP は以下の性質を持つ [6].

性質(1): $\mathbf{u}$ を固定したとき、 $L(\mathbf{d}, \mathbf{u})$ は $\mathbf{d}$ に関して狭義凸2次関数であるから $\mathbf{d}(\mathbf{u}) = \arg \min_{\mathbf{d} \in \hat{X}} L(\mathbf{d}, \mathbf{u})$ は常に唯一存在する.

性質(2): $\psi(\mathbf{u})$ は連続的微分可能な凸関数であり、その勾配は次式で与えられる.

$$\nabla \psi(\mathbf{u}) = -\nabla_{\mathbf{u}} L(\mathbf{d}(\mathbf{u}), \mathbf{u}) \quad (9)$$

性質(1), (2) より双対問題 QDP は連続的微分可能な凸関数 $\psi(\mathbf{u})$ の制約なし最小化問題となるので準ニュートン法を適用することができる. 準ニュートン法において $\mathbf{u}$ を更新する際には、関数 $\psi(\mathbf{u})$ の値と勾配 $\nabla \psi(\mathbf{u})$ を

求める必要があるが、ラグランジュ関数の定義 (8) 式と (9) 式より、これらはラグランジュ緩和問題 $QP(\mathbf{u})$ を解き、得られた $\mathbf{d}(\mathbf{u})$ を用いて次のように計算できる.

$$\psi(\mathbf{u}) = -L(\mathbf{d}(\mathbf{u}), \mathbf{u}) \quad (10)$$

$$\nabla \psi(\mathbf{u}) = \sum_{t=1}^T (\mathbf{c}_t + \mathbf{A}_t \mathbf{d}_t(\mathbf{u})) \quad (11)$$

とくに、双対問題 QDP の最適解 $\mathbf{u}^*$ が得られたとき、その $\mathbf{u}^*$ に対する $\mathbf{d}(\mathbf{u}^*)$ が部分問題 QP の最適解となる. また、ラグランジュ緩和問題 $QP(\mathbf{u}^{(j)})$ は各時刻 t に対応する T 個の独立な部分問題

$$\min_{\mathbf{d}} \mathbf{g}_t^T \mathbf{d}(t) + \frac{1}{2} \mathbf{d}(t)^T \mathbf{B}_t \mathbf{d}(t) - \mathbf{u}^{(j)T} (\mathbf{c}_t + \mathbf{A}_t \mathbf{d}(t)) \quad (12)$$

$$\text{subject to } \mathbf{d}(t) \in \hat{X}(t)$$

に分割して並列に解くことが可能である.

以上をまとめると、部分問題 QP を解くための準ニュートン法のアルゴリズムは以下のように記述できる. なお、アルゴリズム中に現れる $\mathbf{H}^{(j)}$ はヘッセ行列 $\nabla^2 \psi(\mathbf{u}^{(j)})$ の逆行列に対する近似行列である [4,5].

Procedure QPSOLVE

ステップ 1 適当な初期値 $\mathbf{u}^{(0)}$ と正定値対称行列 $\mathbf{H}^{(0)}$ を選ぶ. 十分小さい正数 $\varepsilon_1, \varepsilon_2$ を選ぶ. $j := 0$ とおく.

ステップ 2 T 個の2次計画問題 (12) を解いて、ラグランジュ緩和問題 $QP(\mathbf{u}^{(j)})$ の最適解 $\mathbf{d}(\mathbf{u}^{(j)})$ を求める. 探索方向ベクトル $\mathbf{p}^{(j)}$ を次式によって定める.

$$\begin{aligned} \mathbf{p}^{(j)} &= -\mathbf{H}^{(j)} \nabla \psi(\mathbf{u}^{(j)}) \\ &= -\mathbf{H}^{(j)} \sum_{t=1}^T (\mathbf{c}_t + \mathbf{A}_t \mathbf{d}_t(\mathbf{u}^{(j)})) \end{aligned} \quad (13)$$

ステップ 3 適当な直線探索アルゴリズムを用いてステップサイズ $\alpha^{(j)} > 0$ を定め、 $\mathbf{u}^{(j+1)} = \mathbf{u}^{(j)} + \alpha^{(j)} \mathbf{p}^{(j)}$ とおく.

ステップ 4 収束判定条件

$$\|\nabla \psi(\mathbf{u}^{(j)})\| < \varepsilon_1,$$

$$\psi(\mathbf{u}^{(j)}) - \psi(\mathbf{u}^{(j+1)}) < \varepsilon_2 (1 + |\psi(\mathbf{u}^{(j)})|)$$

がともに成立すれば計算を終了する. そうでない場合は BFGS 公式を用いて $\mathbf{H}^{(j)}$ を更新し、 $j := j+1$ においてステップ 2 へ戻る.

4. 計算実験

同時刻制約に現れるプロセスモデルを簡略表現したボイラーの時系列運転最適化問題に対して、本論文で提

$$\begin{bmatrix}
0 & d_A - 10 & 0 & 0 & 0 & -(d_A + e_A) & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & d_B - 1 & 0 & 0 & 0 & -(d_B + e_B) & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & d_A & -10 & 0 & 0 & -(d_A + e_A) & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_B & -1 & 0 & 0 & 0 & -(d_B + e_B) & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_A - 10 & 0 & 0 & 0 & -(d_A + e_A) & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_B & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -(d_B + e_B) & 1
\end{bmatrix}
\begin{bmatrix}
x(1) \\
x(2) \\
x(3) \\
x(4)
\end{bmatrix}
=
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\quad (23)$$

案したアルゴリズムを適用した。ここでは、同じ燃料を用い、同じ構造を有するが効率のやや異なる2基のボイラーに対して、全運転期間にわたる使用燃料の総量を最小化する問題を考える。

$$\min_{\mathbf{z}} \sum_{t=1}^T (z_{A1}(t) + z_{B1}(t))$$

subject to

$$\frac{z_{A1}(t)}{z_{A2}(t)} = a_A z_{A2}(t) + b_A R_A(t) + c_A, \quad t=1, \dots, T \quad (14)$$

$$\frac{z_{B1}(t)}{z_{B2}(t)} = a_B z_{B2}(t) + b_B R_B(t) + c_B, \quad t=1, \dots, T \quad (15)$$

$$R_A(t) - R_A(t-1) = d_A (z_{A2}(t) - z_{A2}(t-1)) + e_A z_{A2}(t), \quad t=2, \dots, T \quad (16)$$

$$R_B(t) - R_B(t-1) = d_B (z_{B2}(t) - z_{B2}(t-1)) + e_B z_{B2}(t), \quad t=2, \dots, T \quad (17)$$

$$z_{A2}(t) + z_{B2}(t) = D(t), \quad t=1, \dots, T \quad (18)$$

$$LOW_A \leq z_{A2}(t) \leq HIGH_A, \quad t=1, \dots, T \quad (19)$$

$$LOW_B \leq z_{B2}(t) \leq HIGH_B, \quad t=1, \dots, T \quad (20)$$

$$R_A(1) - R_{A0} = 0 \quad (21)$$

$$R_B(1) - R_{B0} = 0 \quad (22)$$

ここで $t=1, \dots, T$ はサンプル時刻であり、 z_{A1} はボイラー A における所要燃料カロリー、 z_{A2} はボイラー A における発生蒸気量、 R_A はボイラー A の汚れ係数、 z_{B1} はボイラー B における所要燃料カロリー、 z_{B2} はボイラー B における発生蒸気量、 R_B はボイラー B の汚れ係数、 LOW_A はボイラー A 発生蒸気量の下限值、 LOW_B はボイラー B 発生蒸気量の下限值、 $HIGH_A$ はボイラー A 発生蒸気量の上限值、 $HIGH_B$ はボイラー B 発生蒸気量の上限值、 R_{A0} はボイラー A 汚れ係数の初期値、 R_{B0} はボイラー B 汚れ係数の初期値、 $D(t)$ は時刻 t における蒸気デマンド、(14)、(15) 式は過去の運転データから得られたボイラーの効率を表現する式、(16)、(17) 式は過去の運転データから得られた汚れ係数の進行を表す経験式、 a, b, c, d, e は t に依存しない定数、(18) 式は各時刻における蒸気デマンドに関する制約、(19)、(20) 式は各ボイラー発生蒸気量に関する制約、(21)、(22) 式

はボイラー汚れ係数の初期条件を表す。

上記の問題においては制約条件 (16) 式と (17) 式が時系列制約、それ以外の制約条件が同時刻制約になっている。たとえば $T=4$ の場合、制約条件 (16) 式と (17) 式は (23) 式のように表現される。

左辺の係数行列は各制約条件 ((16) 式、(17) 式) にたかだか二つの時刻に関する変数しか含まれないので時系列制約 T が増加するにしたがってますます疎になる。このような零要素の構造を利用することにより、ラグランジュ緩和問題 $QP(\mathbf{u}^{(j)})$ を (12) 式のように分解して解く際に、計算の効率化を図ることができる。

上記のボイラー運転最適化問題に対して、本論文で提案した時系列単位の小規模部分問題に分解して QP 部分問題を解く SQP 法を適用した場合 (Case1) と QP 部分問題を分解せず直接解く通常の SQP 法を適用した場合 (Case2) の2ケースについて計算を行い、計算時間の差異を比較した。その結果を Table 1 に示す。Table 1 における “Ideal Comp. Time” とは Procedure QPSOLVE において完全な並列計算が実現できると想定した場合の推定計算時間を示す。ここでいう完全な並列計算とはプロセッサ間のデータ通信や同期のオーバーヘッドがなく、各時刻単位に分割される部分問題数に等しいプロセッサが使えることを想定している。Table 1 の結果より Time Horizon が長くなるにともない、ここで提案した分解法が有利になることがわかる。なお、計算実験では MATLAB を用いてプログラムを作成し、とくに QP 部分問題に対しては MATLAB Optimization Toolbox の QP サブルーチン QUADPROG を用いた。

Table 1 Comparison of computation time

Time Horizon		5	10	15	20
Case1	Comp. Time [s]	38.28	156.54	286.11	620.71
	Ideal Comp. Time* [s]	16.76	70.45	91.84	214.79
	No. of Iterations	12	25	29	34
Case2	Comp. Time [s]	23.23	123.53	305.72	1074.4
	No. of Iterations	11	16	18	35

さらに時系列問題特有の構造、性質を利用して下記の工夫を施すことによって計算時間の効率化が図れる。たとえば $T=15$ の場合は (1) の工夫を適用することによって計算時間が 35% 削減された。(2) の効果はこの例題では顕著な効果は出なかったが、問題の規模、構造によ

ては有効であると考えられる。

- (1) SQP アルゴリズムの各反復において, Procedure QPSOLVE を用いて新しい QP 部分問題を解く際の初期行列 $\mathbf{H}^{(0)}$ として, SQP アルゴリズムの前の反復における Procedure QPSOLVE によって最後に得られた $\mathbf{H}^{(j)}$ を用いる。
- (2) Procedure QPSOLVE のステップ 2 において QP 部分問題 (12) を解く際に, 制約条件の行列が (23) 式のような疎構造を有することを考慮して計算量を軽減する。

5. おわりに

時系列プロセス最適化問題に対して時系列問題構造の特徴を有効に利用した SQP 法のアルゴリズムを提案した。ここで提案した方法を適用することにより, とくに運転期間が長期にわたる問題に対して計算時間の効率化が可能となることが示された。現在, 全運転期間が 30 日から 40 日, サンプル間隔が 6~12 時間程度の触媒劣化を伴う反応器に対して時系列最適化を実現しようとしているが, 本論文で提案した最適化手法を用いれば計算時間が 30 分以内に収まり十分実時間制御に耐える。さらに並列計算機を用いて計算を行えば計算時間の大幅な低減が期待される。また, 今回は考慮しなかったが, 時系列プロセスの最適化を運用していくうえでモデルと実プロセス間の誤差が問題となる場合がありうる。そのような場合には, モデルに実プロセスとのずれを補償するバイアスを持たせ, 最適化実行の周期ごとにモデルを更新し, 最適化の精度を上げることが必要と考えられる。これは今後の重要な研究課題である。

参考文献

- [1] C. R. Cutler and R. T. Perry: Real-time optimization with multivariable control is required to maximize profits; *Comp. & Chem. Eng.*, Vol. 7, pp. 663-667 (1983)
- [2] G. Emoto, A. Tsuda, T. Takeshita, M. T. Monical, S. Nakagawa and K. Fujita: Integrated large-scale multivariable control and real-time optimization of a power plant; *Proceedings of the 1998 IEEE International Conference on Control Applications*, pp. 1368-1372 (1998)
- [3] J. D. Perkins: Plant-wide optimization: opportunities and challenges; *Proceedings of FOCAPO 98*, pp. 15-26 (1998)
- [4] 茨木, 福島: 最適化の手法, 共立出版 (1993)
- [5] 茨木, 福島: FORTRAN77 最適化プログラミング, 岩波書店 (1991)
- [6] 福島: 非線形最適化の基礎, 朝倉書店 (2001)
- [7] A. Bell: Plant optimization of utility complex; *Hydrocarbon Processing*, Vol. 78, No. 12, pp. 61-65 (1999)
- [8] 国井: 熱的単位操作 (上), 丸善 (1976)
- [9] G. Emoto, S. Nakagawa, S. Ebara and K. Fujita: Industrial approaches to multi-period process optimization; *Proceedings of PSE Asia 2000*, pp. 119-124 (2000)