多重 SA 法を用いた化学プロセスの スケジューリング*

村上 佳広**・内山 寛信**・長谷部伸治***・橋本 伊織***

Application of Repetitive SA Method to Scheduling Problems of Chemical Processes*

Yoshihiro MURAKAMI**, Hironobu UCHIYAMA**, Shinji HASEBE*** and Iori HASHIMOTO***

One of the predominant characteristics of chemical batch processes is that the material leaving a batch unit is fluid. Therefore, the starting times of jobs at each unit must be determined by taking into account the availability of storage between two units. In this paper, for such chemical processes, a scheduling algorithm using simulated annealing (SA) method is proposed.

In the proposed method called repetitive SA method, in order to reduce the probability of being trapped in a bad local optimum, the scheduling using SA method is executed repeatedly, and a best schedule is selected at the final stage of repetition. The problem is how to determine the number of repetitions of scheduling and the number of schedules searched in a round of scheduling. In order to find out the best combination of those two numbers, first the probability distributions of the performances of the schedules are calculated for various cases where the number of schedules searched in a round of scheduling is different from one another. Then, the best combination of those numbers is selected using the derived probability distributions.

The results of applying the repetitive SA method to scheduling problems of chemical processes suggest that the proposed method is effective typically in reducing the deviation of the performances of the schedules.

1. はじめに

シミュレーティッド・アニーリング法 (SA 法) は, 常に実行可能解が得られる点や, 許される計算時間に応じて最適に近い解を求めることができる点などから, スケジューリング問題の解法として広く利用されるようになってきている. SA 法を用いた解法では, 無限回の探索を行うという仮定のもとで, 確率的に最適解に到達することが保証されている. しかし, 実際に SA 法をスケジューリング問題に適用する際には, 限られた計算時間内に評価の良い解を得ることが要求される. 種々の制約条件を考慮する必要がある現実のスケジューリング問題では, 各

装置でのジョブの処理順が与えられたとしても, 評価を最適とする各ジョブの各装置での開始時刻を求めるには多大な計算時間を要する. このような問題では, 解の探索回数を十分多くとることができないため, 一度局所最適解に陥るとそこから抜け出すことは難しく, 限られた時間内の探索で常に良い評価の解が得られるとは限らない.

遺伝的アルゴリズム (GA) では, 複数のスケジュールを発生させ, それらの点から探索を進めることにより, 局所最適解に陥る可能性を減らしている. 本研究では, SA 法を複数回行い, その中で一番良い評価の解を採用することにより, 局所最適解に陥る確率を減らしたスケジューリング手法を提案する. そして, SA 法で得られる解の評価値の分布が得られたとき, 1 回の SA 法での探索回数と, SA 法の実行回数の好ましい組み合わせを, 統計的に求める方法を示すとともに, 得られた手法をフローショップ問題に適用し, その有効性を検証する.

* 原稿受付 1996 年 5 月 28 日

** 関西大学 工学部 Faculty of Engineering, Kansai University; 3-3-35 Yamatecho, Suita city, Osaka 564, JAPAN

*** 京都大学 大学院 工学研究科 Graduate School of Engineering, Kyoto University; Yoshida-Honmachi, Sakyo ward, Kyoto city, Kyoto 606-01, JAPAN

Key Words: simulated annealing, scheduling system, chemical process, simulation, flow shop problem.

2. SA法によるスケジューリング

2.1 化学プロセスのスケジューリング問題

Fig. 1 に示すような、二つの装置と中間タンクからなる簡単な化学プロセスのスケジューリング問題を例に考える。装置間には、中間タンクが1器設置されており、一つのジョブを保持できるものとする。各装置では、一つのジョブを処理するごとに装置の洗浄操作が必要であり、洗浄に要するコスト（ジョブの切り換えコスト）は、連続して処理するジョブの種類に依存するものとする。また、各ジョブは異なる納期を有し、製品の完成が納期に間に合わない場合、納期遅れ時間に比例したペナルティが課せられるものとする。このようなプロセスに対し、切り換えコストと納期遅れペナルティの重み付き和を最小にするスケジュールを求める問題について考える。

Fig. 1 Chemical process consisting of two batch units

この問題は中間在庫制約を有するフローショップ問題とみなすことができる。化学プロセスのスケジューリング問題が、機械系の問題と大きく異なる点は、流体を扱うため、装置間に物質を保持するためには容器（中間タンク）が必要である点である。一つの装置で処理が終了したジョブの流出操作は、装置間に中間タンクが存在しなければ、必ず次の装置の流入操作と時間的に同期をとらなければならない。また、装置間でジョブの処理順を変更するためには、追い抜かれるジョブは必ず中間タンクに保持されなければならない。このような特徴を有する化学プロセスのスケジューリング問題について、以下に検討する。

2.2 スケジューリングアルゴリズム

2.1 で説明した問題に対して、SA法を用いたスケジューリングアルゴリズムを考えるには、まずスケジュールの表現法とその変更法を明確にする必要がある。現実のスケジューリング問題では、夜間や週末における装置の使用に関する制約など、種々の制約を考慮してジョブの開始時刻を定めなければならない。よって、各装置でのジョブの処理順が与えられたとしても、その処理順のもとで与えられた評価を最適とするジョブの処理開始時刻を求めるアルゴリズムは複雑なものとなる。このような点を考慮して、本研究では、各装置でのジョブの処理順を決定・変更するアルゴリズムと、ジョブの処理順が与えられた条件下での開始時刻を計算するアルゴリズム

を分離して考えた。このようにすることによって、それぞれのアルゴリズムを独立に開発でき、また、種々の処理順決定アルゴリズムの評価を、開始時刻計算アルゴリズムと独立して行うことができる。

たとえば、Fig. 1 に示したプロセスに対して、9個のジョブの処理順は、Fig. 2 のように表される。本研究では、納期遅れに対するペナルティを評価に加えていることから、ジョブは各装置において可能な限り早く処理することが望ましい。よって、処理順が与えられたとき、「前詰め」で処理を行うように、各装置でのジョブの処理開始時刻を求めた。ジョブの処理開始時刻を計算するアルゴリズムについては、2.3 で説明する。

Fig. 2 Processing order of 9 jobs on two batch units

SA法を用いて逐次スケジュールを変更していくアルゴリズムでは、どのような近傍を考えるかが問題となる。本研究では、ジョブをランダムに一つ選び、各装置ごとにそのジョブをランダムに選んだ別のジョブの前または後ろに挿入して得られるスケジュールの全体を、ある一つのスケジュールの近傍とした。このような手法に基づき作成したスケジューリングアルゴリズムは、以下のステップからなる。

Step 1 SA法で用いるパラメータ (T_0 : 初期温度, T_f : 最終温度, N_s : 同一温度で生成するスケジュールの数, δ : アニール比) を読み込む。

Step 2 装置1, 2共に同一となるようなジョブの処理順をランダムに生成する。得られた処理順に対して、各ジョブの処理開始時刻を求め、現時点でのスケジュール (R と表す) とする。また、 R を現時点での最適スケジュール (R_{opt}) とおく。温度パラメータ T を T_0 とおく。

Step 3 R から新たなスケジュール (R_{new}) を、以下の手順で生成する。

- (i) ジョブを一つランダムに選ぶ。
- (ii)(i) で選んだジョブの装置1での処理順を、ランダムに変更する。
- (iii)(i) で選んだジョブの装置2での処理順を、中間タンク制約を考慮して実行可能なスケジュールが得られる範囲でランダムに変更する。化学プロセスでは、流体を扱うため、装置間にタンクがなければジョブの追い抜きが許されない。このような制約から、装置2でのジョブの処理順を勝手に決めることはで

きない。制約を満たす挿入箇所の決め方は、付録に示す。

(iv)得られた処理順に対して、各ジョブの処理開始時刻を計算する。

Step 4 スケジュール R に対する評価関数値を、 $g(R)$ と表すことにする。 $g(R_{\text{new}})$ が $g(R)$ より小さければ（良ければ）、 $R \leftarrow R_{\text{new}}$ とし、そのスケジュールを無条件で受け入れる。そうでない場合は、次式を満たすか否かを計算し、満たす場合は $R \leftarrow R_{\text{new}}$ とする。

$$p < \exp \left\{ -\frac{g(R_{\text{new}}) - g(R)}{T} \right\} \quad (1)$$

ここで、 p は $(0,1]$ の一様乱数である。

Step 5 $g(R_{\text{new}}) < g(R_{\text{opt}})$ であれば、 $R_{\text{opt}} \leftarrow R_{\text{new}}$ とする。

Step 6 Step 3~5 を N_s 回実行する。

Step 7 $T \leftarrow \delta \cdot T$ とし、 T が T_f より大きければ、Step 3に戻る。

Step 8 得られたスケジュールにおいて、同一の製品を生産する複数のジョブが納期順になっていなければ、それらを納期順に並び換える。

2.3 処理開始時刻計算法

ジョブの処理順が与えられた条件下での、開始時刻計算法¹⁾を示す。ここでは、二つの装置からなるプロセスを対象に示すが、3装置以上の場合も同様に計算できる。装置1で処理を終えたジョブは、装置2が待ち状態にあり、かつそのジョブ以前に処理すべきジョブの処理が終了していれば、直ちに装置2に送られる。そうでない場合で、装置間に設置されたタンクが利用可能であれば、そのジョブは一時的にタンクに保持される。

装置 k で j 番目に処理されるジョブの番号を $J(k,j)$ 、ジョブ i の装置 k での処理開始時刻を t_i^k 、ジョブ i の装置 k での処理時間を p_i^k 、装置 k と装置 $k+1$ の間に設置された中間タンク数を T^k とする。ここでは、説明を簡単にするため、装置への流入操作、装置からの流出操作に要する時間は無視できるとし、装置への流入操作終了後は直ちに処理を開始すると仮定する。また、流出終了後の洗浄操作に要する時間も無視できると仮定する。このように考えると、各ジョブ（ジョブ数を N とする）の開始時刻は、以下のようにして求めることができる。

Step 1 すべてのジョブの両装置での処理開始時刻を、可能な最早開始時刻とする。また、0番目のジョブの処理時間を0とし、 $t_{J(1,0)}^1, t_{J(2,0)}^2$ も最早開始時刻とする。

Step 2 i を1から N まで変え、以下のStep 3~6を繰り返す。

Step 3 $t_{J(1,i)}^1 < t_{J(1,i-1)}^1 + p_{J(1,i-1)}^1$ ならば、 $t_{J(1,i)}^1 \leftarrow t_{J(1,i-1)}^1 + p_{J(1,i-1)}^1$ とする。

Step 4 $t_{J(2,i)}^2 < \max\{t_{J(2,i-1)}^2 + p_{J(2,i-1)}^2, t_{J(2,i)}^1 + p_{J(2,i)}^1\}$ ならば、 $t_{J(2,i)}^2 \leftarrow \max\{t_{J(2,i-1)}^2 + p_{J(2,i-1)}^2, t_{J(2,i)}^1 + p_{J(2,i)}^1\}$ とする。

Step 5 装置1で $J(2,i)$ より前であつ装置2で $J(2,i)$ より後ろにあるジョブの数 S_i を求める。

Step 6 装置1と装置2でともに $J(2,i)$ より前あるジョブの集合を G とする。装置2の G に属するジョブの中で、 $J(2,i)$ より $(T^1 - S_i)$ 個前のジョブの処理順を j' とすると、 $t_{J(2,i)}^1 + p_{J(2,i)}^1 < t_{J(2,j')}^2$ ならば、 $t_{J(2,i)}^1 \leftarrow t_{J(2,j')}^2 - p_{J(2,i)}^1$ とする。もし、そのようなジョブが存在しなければ、Step 7へ。

Step 7 $t_{J(2,N)}^2$ が、あらかじめ定めた最大値より大きければ、与えた処理順での実行可能な解は無いとして、計算を終了する。

Step 8 Step 4~6 で開始時刻が変更されたジョブが一つでも存在すれば、Step 2に戻る。変更されたジョブがなければ、実行可能解として、出力する。

たとえば、Fig. 1のプロセスでジョブ5~9がFig. 2に示した処理順で処理される場合、そのスケジュールはFig. 3になる。上記の計算法で得られたジョブの開始時刻は、処理順の制約を満たす最早開始時刻になる。開始時刻間の制約は、線形の不等式で表すことができることから、処理順の制約を満たす最早開始時刻は、LP問題を解くことによっても得ることができる。本研究では、計算時間の点や、週末や夜間の処理禁止制約などの制約が追加された際の拡張のしやすさを考え、上記のようなアルゴリズムを用いた。

Fig. 3 Gantt chart for the schedule shown in Fig. 2

2.4 パラメータ決定法

SA法のパラメータは、探索の効率に影響する重要な変数である。本研究では、初期温度と最終温度を決定するために、以下に述べるウォームアップ法^{2),3)}を用いた。

まず、初期スケジュールをランダムに作成し、そのスケジュールの近傍から10,000個の新たなスケジュールをランダムに選ぶ。得られたスケジュールの中で、元のスケジュールより評価の悪いスケジュールの集合を Q とおく。温度パラメータ T が与えられたとき、 Q に属するス

スケジュールがSA法で受け入れられる平均確率を $\mu_A(T)$ とし、(2)式で定義する。

$$\mu_A(T) = \frac{1}{N_b} \sum_{R_i \in Q} \exp \left\{ -\frac{g(R_i) - g(R_0)}{T} \right\} \quad (2)$$

ここで、 R_0 は初期スケジュール、 N_b は Q に属するスケジュールの数である。

(2)式を用いて、初期温度 T_0 、最終温度 T_f を、 $\mu_A(T_0) = 0.5$ 、 $\mu_A(T_f) = 0.002$ となるように決定した。

2.5 スケジューリング結果

これまで提案した方法で、10個のジョブのスケジューリング問題を解いた。温度更新回数 N_a は30回で一定とし、同一温度での探索回数 N_s の種々の値に対して、2.2で示したアルゴリズムを用いて解を求めた。得られた結果を、Fig. 4に示す。Fig. 4は、 N_s の各値に対して、同一の問題を10,000回解いて得られた解の評価値の平均値、最小値、最大値、分散をプロットしたものである。この図より、平均値と分散は N_s を増やすとともに若干減少していくが、その傾向は、徐々に緩やかになっていくことがわかる。また、10,000回の最適化計算で得られた解の中での最適解の評価値は、探索回数の多少にあまり影響されていない。これは、評価関数が多峰性であることから、少ない探索回数で良い解に収束することもあるが、一度局所最適解に収束してしまうとそこから抜け出すことは、探索回数を多少多くしても (N_s を大きくしても) 難しいことを示している。

Fig. 4 Scheduling results for various N_s values

規模の大きな問題で総探索回数が制限されている場合、このような状況は頻繁に生じると考えられる。本研究では、このような状況から抜け出す方法として、SA法を複数回実行し、その中から最適な解を選ぶ、という方法を提案する。

3. Repetitive SA 法

3.1 Repetitive SA 法による解の分布の導出

SA法による最適化を複数回繰り返す、その中で最良の解を選択する方法を、Repetitive SA法 (以後、rSA法と略す) と呼ぶ。従来、rSA法を経験的に有効だと考え、SA法を複数回繰り返すその中で最良の解を選択する方法を適用した例^{4)~6)}は見られるが、SA法を何回繰り返せばよいかを理論的に導出した文献は見られない。スケジューリングに利用できる計算時間が一定という条件下では、SA法を複数回行う分だけ、同一温度での探索回数 (N_s)、あるいは温度更新回数 (N_a) を少なくしなければならない。SA法による探索を1回行って得られる解の評価は、 N_s や N_a の値を小さくすれば悪くなる。よって、SA法の実行回数 (N_r) をどのような基準で定めるかが問題となる。

同一の問題に対してSA法を複数回実行した場合、その解の評価値はある分布を持つ。この分布がわかれば、rSA法で得ることのできる解の分布は、以下のようにして導くことができる。

一回のスケジューリングでの解の総探索回数 (N_s と N_a の積) を、以後一回の探索回数と呼び、 N_m で表す。SA法で得た解の評価関数値 x の確率密度関数を $f(x, N_m, 1)$ 、累積分布関数を $F(x, N_m, 1)$ とする。SA法による最適化を N_r 回繰り返した中で最良の解の評価が、区間 $[x, x + \Delta x]$ に属する確率 P は、 N_r 回の繰り返しの中で、少なくとも1回、区間 $[x, x + \Delta x]$ に属する評価の解が得られればよいことから、次式のように表すことができる。

$$P = \sum_{i=1}^{N_r} \{1 - F(x + \Delta x, N_m, 1)\}^{i-1} \cdot \{1 - F(x, N_m, 1)\}^{N_r-i} \cdot \int_x^{x+\Delta x} f(x, N_m, 1) dx \quad (3)$$

(3)式より、SA法を N_r 回繰り返すrSA法で得られる解の評価関数値の確率密度関数 $f(x, N_m, N_r)$ は、以下のように表すことができる。

$$f(x, N_m, N_r) = N_r \{1 - F(x, N_m, 1)\}^{N_r-1} f(x, N_m, 1) \quad (4)$$

3.2 繰り返し回数 N_r の決定法

1回のスケジューリングに利用可能な時間内で行うことのできる、解の総探索回数を N_t 、その時間内でのSA法の繰り返し回数を N_r とした場合、得られる解の期待値、およびその分散は、(5)、(6)式で表される。

〈 N_r 回SA法を実行した時の解の評価値の期待値〉

$$\bar{x}(N_t, N_r) \equiv \int_0^\infty x f\left(x, \frac{N_t}{N_r}, N_r\right) dx \quad (5)$$

< N_r 回 SA 法を実行した時の解の評価値の分散 >

$$\sigma(N_t, N_r) \equiv \int_0^\infty \{x - \bar{x}(N_t, N_r)\}^2 f\left(x, \frac{N_t}{N_r}, N_r\right) dx \quad (6)$$

これらの式中の $f(x, N_t/N_r, N_r)$ は、探索回数 N_m を N_t , $N_t/2$, $N_t/3$, ..., とした際の確率密度関数 $f(x, N_m, 1)$ (あるいは累積分布関数 $F(x, N_m, 1)$) をスケジューリング結果から求めることにより, (4) 式で計算できる. よって, (5) 式あるいは (6) 式の積分を N_r を変えて実行することにより, 解の期待値, あるいはその分散を最小にする SA 法の繰り返し回数 N_r を求めることができる.

2.1 で説明した問題に対して, 解の探索回数を 9,000 回とした SA 法を, 10,000 回実行して得た解の累積度数分布と, その分布をワイブル分布で近似した際の累積分布関数を Fig. 5 に示す.

Fig. 5 Approximation of scheduling result's distribution by Weibull distribution function

SA 法を複数回実行して得られた解の評価関数値の分布が, ワイブル分布に従うと仮定できれば^{7),8)}, (5) 式, あるいは (6) 式を最小にする N_r は, (5), (6) 式を数値積分せずに, 以下のようにして求めることができる.

ワイブル分布の確率密度関数 $w(x)$ と累積分布関数 $W(x)$ は, 以下のように表される.

$$w(x) = \begin{cases} \left(\frac{\gamma}{\beta}\right) \left(\frac{x-\alpha}{\beta}\right)^{\gamma-1} \exp\left\{-\left(\frac{x-\alpha}{\beta}\right)^{\gamma}\right\} & : x > \alpha \\ 0 & : x \leq \alpha \end{cases} \quad (7)$$

$$W(x) = \begin{cases} 1 - \exp\left\{-\left(\frac{x-\alpha}{\beta}\right)^{\gamma}\right\} & : x > \alpha \\ 0 & : x \leq \alpha \end{cases} \quad (8)$$

ここで, $\alpha \geq 0$, $\beta > 0$, $\gamma > 0$ である. パラメータ α は, 評価関数の大域的最小値の推定値, β は, 分布の広がり具合を, そして, γ は分布の形状を表す.

SA 法で得られる解の評価関数値が, 上記のワイブル分布で表されると, この試行を N_r 回行って得られた解の最小値の確率密度関数 $w(x, N_r)$ は, (4) 式より, 以下のように表される.

$$w(x, N_r) = \begin{cases} \left(\frac{\gamma}{\beta'}\right) \left(\frac{x-\alpha}{\beta'}\right)^{\gamma-1} \exp\left\{-\left(\frac{x-\alpha}{\beta'}\right)^{\gamma}\right\} & : x > \alpha \\ 0 & : x \leq \alpha \end{cases} \quad (9)$$

ここで,

$$\beta' = \frac{\beta}{\sqrt[N_r]{N_r}} \quad (10)$$

である.

(7) 式のワイブル分布の平均 μ_1 と分散 μ_2 は, ガンマ関数 Γ を用いて, 以下のように表すことができる.

$$\mu_1 = \alpha + \beta \Gamma\left(1 + \frac{1}{\gamma}\right) \quad (11)$$

$$\mu_2 = \beta^2 \left\{ \Gamma\left(1 + \frac{2}{\gamma}\right) - \Gamma^2\left(1 + \frac{1}{\gamma}\right) \right\} \quad (12)$$

ここで,

$$\Gamma(a) = \int_0^\infty x^{a-1} \exp(-x) dx \quad (13)$$

とする.

したがって, 1 回の探索回数 (N_m) を変えて SA 法を行った, 種々のデータに対するワイブル分布近似関数が求められていれば, そのパラメータより, (10)~(12) 式を用いて, 平均, あるいは分散を最小にする N_r を求めることができる.

3.3 2工程フローショップ問題に対する適用結果

2.1 でとりあげた問題に対して, 探索回数を変えた九つのケースに対して, SA 法で得られた結果をワイブル分布で近似した際のパラメータ, 解の平均および分散値を, Table 1 に示す.

いま, 9,000 回の解の探索を行えるだけの計算時間が許されていたとする. Table 1 の結果を用いることにより, SA 法の繰り返し回数 (N_r) の種々の値に対して, (10)~(12) 式を用いて, rSA 法を用いた際の解の平均と分散値, β' の値を求めることができる. 得られた結果を Fig. 6 に示す.

Fig. 6 より, 総探索回数が 9,000 回に限定された場合, 1 回の探索回数 (N_m) を 300 回とした SA 法によるスケジューリングを 30 回繰り返し, その中で最適解を選ぶ方法が, 得られる解の平均, 分散という観点から最適であることがわかる. $N_m=9,000$, $N_r=1$ とした場合, お

Table 1 Parameters in Weibull distribution functions

N_m	α	β	γ	average	variance
9,000	164	8.6	1.91	171	17
4,500	164	10.8	1.81	174	30
3,000	164	12.2	1.84	175	37
1,500	164	14.8	1.97	177	48
900	164	17.4	2.05	179	62
600	164	20.8	1.55	183	151
300	164	34.4	1.09	197	945
150	164	62.5	1.27	222	2105
30	164	162.1	2.41	308	4047

Fig. 6 Estimated results of rSA method

よび $N_m=300$, $N_r=30$ とした場合の、ワイブル分布で近似した解の確率密度関数を、Fig. 7 に示す。この図からも、rSA 法を用いることにより、得られる解が評価の小さい方へ集中していることがわかる。また、Table 1 の例のように、 N_m を変えたスケジューリング結果をワイブル分布で近似した際の α , γ の値がほぼ等しければ、解の平均や分散を求めなくても、 β' の値で最適な N_r の値を推定できることが、(10)~(12) 式からわかる。

Fig. 7 Estimated distributions of scheduling results

Table 1 のデータが得られれば、 N_t の異なる種々のケースについて、最適な N_r と N_m の組み合わせを求めることができる。また、解のばらつきの分散をある値以

下に抑えたいとき、最低限必要な総探索回数を (10), (12) 式を用いて推定することもできる。

3.4 現実規模の問題への適用

問題のサイズが大きくなるにつれ、探索空間（実行可能な解の数）が大きくなり、一つの局所最適解から、別の局所最適解に移るのに必要なスケジュールの変更回数も多くなる。したがって、問題のサイズが大きいほど、局所最適解から抜け出すことは難しく、SA 法によって得られる解のばらつきは大きくなると考えられる。このような点をふまえ、提案した rSA 法をより大きなサイズの問題に適用した。対象としたプロセスは、三つの装置からなるフローショップタイプの化学プロセスで、Fig. 8 に示すように、装置 1 と 2 の間に三つの中間タンク、装置 2 と 3 の間に二つの中間タンクが存在するものとする。このようなプロセスに対して、40 個のジョブのスケジュールを求める問題を対象とした。また、スケジュール作成に利用できる時間内に、450,000 個のスケジュールを探索できるものとした。

Fig. 8 Chemical process consisting of three batch units

Fig. 9 は、温度更新回数 N_a を 300 で一定とし、一つの温度での探索回数 N_s を変えて同一の問題を解いたときの解の評価値を示している。一つの N_s に対して、10 回ずつ SA 法を実行して得た解の評価値をプロットした。この図より、この問題に対しても、 $N_s \geq 150$ において、 N_s を増加させても評価値のばらつきは減少しにくいことがわかる。この問題では、1 回の SA 法に比較的最長の計算時間が必要であるため、各 N_s の値に対して、解の確率分布が得られるまでの試行を行うことは困難である。したがって、各 N_s の値に対して 10 回実行した結果を用いて、rSA 法の適用の可否を検討した。

Fig. 9 Scheduling results for three-unit problem

Fig. 9 よりこの問題では、 $N_s \geq 150$ で各 N_s に対して得られる最良解がほぼ一定の値になっている。よって、ある程度良い局所最適解を得るためには、一つの温度で150回以上の探索が必要と判断し、 $N_s \geq 150$ となる N_s と N_r の組み合わせに対して、rSA 法を適用した。具体的には、総探索回数が450,000回、温度更新回数が300回であることから、SA法の繰り返し回数 N_r を1,3,5,10 ($N_s = 1500, 500, 300, 150$ に対応) としてrSA法の効果を調べた。

各 N_r に対して、rSA法を10回実行して得られた解の平均値と分散の値を、Table 2 に示す。この例では、 N_r を大きくしても解の平均値はあまり変化しないが、分散は小さくなり、分散最小という観点から見れば、 $N_r = 10$ が最適であるといえる。

Table 2 Results of rSA method

N_r	average	variance	min	max
1	756	2659	708	865
3	751	1880	708	845
5	755	2167	683	834
10	756	1490	708	825

4. おわりに

化学プロセスを対象とした、SA法を繰り返し用いるスケジューリング手法を提案した。現実のスケジューリング問題は、広い解空間を持ち、かつスケジュール作成に利用可能な時間が制限される場合が多い。このような問題にSA法を適用する場合、パラメータ設定を適切に行ったとしても、一度局所最適解に捕捉されると、その解から大きく構造の異なる別の局所最適解に推移することは難しい。本研究では、探索回数を多くしても、得られる解のばらつきを小さくすることが困難である問題が存在することを、例題を用いて示した。そして、このような問題に対しては、1回のSA法の探索回数を増やすよりも、探索回数の少ないSA法を繰り返し行い、その中で最適解を選ぶ方が、解の評価の平均、分散を小さくできるケースがあることを示した。また、総探索回数が与えられたとき、SA法の繰り返し回数と1回のSA法での解の探索回数の適切な組み合わせを、統計的手法を用いて求めた。

提案した手法は、複数回のSA法を独立に行っている。よって、スケジュール作成に利用可能な時間が限られている場合、安価な計算機を多数利用、あるいは並列計算機の利用に適した手法ということができる。

参考文献

- 1) S. Hasebe, Y. Murakami and I. Hashimoto: A flexible simulation system for scheduling with various constraints; *J. Chem. Eng. Japan*, Vol. 29, No. 2, pp. 323-329 (1996)
- 2) B. E. Rosen, 中野: シミュレーテッドアニーリング基礎と最新技術; 人工知能学会誌, Vol. 9, No. 3, pp. 365-372 (1994)
- 3) 山田, B. E. Rosen, 中野: クリティカルブロックシミュレーテッドアニーリング法によるジョブショップスケジューリング問題の解法; 電学論C, Vol. 114, No. 4, pp. 476-482 (1994)
- 4) D. S. Johnson, C. R. Aragon, L. A. McGeoch and C. Schevon: Optimization by simulated annealing: an experimental evaluation (part I, graph partitioning); *Opns. Res.*, Vol. 37, No. 6, pp. 865-892 (1989)
- 5) M. Malek, M. Guruswamy, M. Pandya and H. Owens: Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem; *Annals of Opns. Res.*, Vol. 21, pp. 59-84 (1989)
- 6) E. Arts and J. Korst: *Simulated Annealing and Boltzmann machines*, Chap. 6, John Wiley & Sons (1989)
- 7) G. Kudva, A. Elkamel, J. F. Pekny and G. V. Reklaitis: Heuristic algorithm for scheduling batch and semi-continuous plants with production deadlines, intermediate storage limitations and equipment changeover costs; *Computers chem. Engng*, Vol. 18, No. 9, pp. 859-875 (1994)
- 8) B. L. Golden and F. B. Alt: Interval estimation of a global optimum for large combinatorial problems; *Naval Res. Logist. Q.*, Vol. 26, No. 1, pp. 69-77 (1979)

付 録

中間タンク制約を考慮したジョブの処理可能順序

装置1でジョブ i より前に処理され、装置2でジョブ i より後で処理されるジョブの数が、装置2でジョブ i を処理している際に装置間に貯留されているジョブの数となる。この考え方をを用いることにより、選ばれたジョブ(ジョブ k とする)の装置1での処理順が定まった際、装置2でのジョブ k の実行可能な挿入位置は、以下のようにして求めることができる。

- (1) 装置1でジョブ k より前に処理されるジョブの集合を S とする。
- (2) 装置2で処理されるジョブを後ろから順にチェックし、 S に属するジョブで(装置間に設置されたタンク数+1)番目に現れるジョブを求める。得られたジョブの後ろの位置が、中間タンク制約を満たすジョブ k の一番早い処理順となる。そのようなジョブが存在しなければ、装置2で最初に処理されるジョブの前が、ジョブ k の一番早い処理順となる。
- (3) ジョブ k を除いたすべてのジョブに対して、装置2でそのジョブを処理している際に装置間に貯留されているジョブの数を計算する。

- (4) 装置 1 でジョブ k より後で処理されているジョブを順にチェックし, (3) で求めた貯留ジョブ数が, 装置間に設置されたタンク数に等しいジョブを求める. 得られたジョブの前の位置が, 中間タンク制約を満たすジョブ k の一番遅い処理順となる. そのようなジョブが存在しなければ, 装置 2 での最終ジョブの後ろがジョブ k の一番遅い処理順となる.

【例】

装置 1 でのジョブの処理順が, (1, 2, 3, 4, k , 5, 6, 7, 8, 9), 装置 2 でのジョブの処理順が, (1, 3, 2, 4, 5, 6, 7, 9, 8) と与えられたとする. 装置間に貯留可能なジョブ数が 1 の場合, 装置 2 でのジョブ k の挿入可能範囲は, ジョブ 2 より後ろで, ジョブ 9 より前となる.