納期内同時処理完了をめざす ジョブショップスケジューリング*

謝 長征[†]・田 雅杰[‡]・三宮 信夫[§]

Job Shop Scheduling Aiming at Completing All Jobs Simultaneously and within Due Date*

Changzheng XIE[†], Yajie TIAN[‡] and Nobuo SANNOMIYA[§]

1. はじめに

現実の生産プロセスは大規模、複雑であるとともに、種々の互いに競合する目的あるいは要求をもっており、できるだけそれらの目的が融合された妥当なスケジュールを作成しなければならない。また、それらの要求、目的やプロセスの設定条件は一定でなく、時間的に変化することも少なくなく、その変化に適応するために望ましいスケジュールを早急に求める必要がある。

本論文では、在庫コストを減らすために、同一期間内で処理することを要求されたジョブ群を、納期を満たしつつ、できるだけ同時に完了させるという要求を取り上げ、ジョブショップスケジューリング問題において、この問題の定式化を行い、解法として、以前に通常のジョブショップスケジューリング問題に提案した局所探索法 [1,2] を新たに改良して適用する。

2. 問題の記述

N 個のジョブ $J_i (i=1,2,\dots,N)$ が、 H 台の機械 $M_j (j=1,2,\dots,H)$ で処理されるジョブショップ工程を考える。各ジョブ J_i には H 個の作業 $Q_{ij} (j=1,2,\dots,H)$ があり、作業 Q_{ij} の処理時間を P_{ij} とする。各ジョブの作業は、添字 j の昇順に処理されるものとし、総作業数は $Q = NH$ である。

制約条件は、通常のジョブショップ工程におけるもの [3] と同様であるが、本問題ではこのほかに次の要求が課さ

れているものとする。

- 1) すべてのジョブは、納期内に処理を完了させること。
- 2) すべてのジョブは、できるだけ同時に処理を完了させること。

作業 Q_{ij} の処理開始時刻を S_{ij} とすれば、処理完了時刻は、 $C_{ij} = S_{ij} + P_{ij}$ となる。したがって、本問題は次のように表される。

$$\min Z = \max_{1 \leq i \leq N} C_{iH} - \min_{1 \leq i \leq N} C_{iH} \quad (1)$$

subject to

$$C_{max} \triangleq \max_{1 \leq i \leq N} C_{iH} \leq D \quad (2)$$

ここに、 D は納期を表す。

これより、メイクスパン C_{max} が D 以内で、処理最早ジョブと処理最遅ジョブの完了時刻差が最小となるような各機械上での作業処理順序および各作業の処理開始時刻を求める問題を考える。

3. 解法

3.1 解の表現

本研究で取り扱うジョブショップスケジューリング問題の解は、各ジョブ J_i が H 回反復される長さ $NH (=Q)$ のジョブ順列で与えられる。たとえば、 $(N,H) = (4,4)$ の場合、ジョブ順列を

$$J_1, J_2, J_3, J_4, J_3, J_2, \dots, J_3, J_1 \quad (3)$$

のように与えるとき、対応する作業順列 Π は次のように定まる。

$$\Pi : Q_{11}, Q_{21}, Q_{31}, Q_{41}, Q_{32}, Q_{22}, \dots, Q_{34}, Q_{14} \quad (4)$$

作業順列 Π の各作業を前から順番に一つずつ前詰めでガントチャート上に定めていくことにより、全体のスケジュールが作成できる。よって以下では、この作業順列 Π を本問題の解表現とする。

* 原稿受付 2003 年 3 月 28 日

[†] 京都工芸繊維大学 大学院 工学科学研究科 Graduate School of Science and Technology, Kyoto Institute of Technology; Matsugasaki, Sakyo ward, Kyoto city, Kyoto 606-8585, JAPAN

[‡] 国際電気通信基礎技術研究所 人間情報科学研究所 Human Information Science Laboratories, ATR; Seika, Kyoto 619-0288, JAPAN

[§] 京都工芸繊維大学 工学学部 Faculty of Engineering and Design, Kyoto Institute of Technology; Matsugasaki, Sakyo ward, Kyoto city, Kyoto 606-8585, JAPAN

Key Words: job shop scheduling, due date, simultaneous completion, local search, two-stage algorithm.

3.2 解法の概要

本研究では、短い計算時間で準最適解を得るために、局所探索法に基づいた二段階アルゴリズムを提案する。第1段階では、すべてのジョブが納期内に完了するスケジュールを見いだすために、目的関数をメイクスパンとして ($Z = C_{max}$)、これを最小化する過程で $C_{max} < D$ が成り立つ解を求める。第2段階では、この解の近傍で (2) を満たし、(1) を目的関数 Z とする問題の準最適解を探索する。

局所探索法では、何らかの方法で得られた暫定解に対して、その近傍の解を調べて、改善できればそれに置き換えるという手順を、改善が得られなくなるまで繰り返す。本アルゴリズムでは、第1段階の近傍の解は、 Q 個の作業のうち任意の一つの作業 (これをネゴシエータとよぶ) を選び、これとその近接の K 個の作業をそれぞれ順列の中で交換することによって、 K 個得られる。第2段階の近傍の解は、この交換操作と、ネゴシエータに近接する作業の直前または直後にネゴシエータを挿入する操作によって、 K 個得られる。また、暫定解において、同じネゴシエータを選んで、同じ近傍解を重複して探索しないように、一度選んだネゴシエータを記憶する集合として、チェックバッファ B_c を用意する。

3.3 第1段階のアルゴリズム

パラメータとして、近傍サイズを文献 [1] の推奨値 $K = 0.54Q$ に定めた。アルゴリズムは、次のようになる。

- Step 1** 初期解としてランダムに作業順列 Π_0 を作成し、目的関数値 Z_0 を求める。繰り返し計算回数を $t \leftarrow 1$ とする。また、チェックバッファを $B_c \leftarrow \{\phi\}$ とする。
- Step 2** 作業順列 Π_{t-1} より作業を一つランダムに選び、これをネゴシエータ Q_{nm} とする。
- Step 3** もし $Q_{nm} \notin B_c$ ならば Q_{nm} を集合 B_c に入れ、Step 4 に進む。もし $Q_{nm} \in B_c$ ならば Step 2 に戻る。
- Step 4** 交換を行って、新たに K 個の処理順列 $\Pi^k (k = 1, 2, \dots, K)$ を作成し、それぞれの目的関数値 Z^k を求める。
- Step 5** Step 4 で作成された解 Π^k および $Z^k (k = 1, 2, \dots, K)$ に対して、もし $\min Z^k \leq Z_{t-1}$ なら、 $Z_t = \min Z^k = Z^{best}$ および $\Pi_t = \Pi^{best}$, $B_c \leftarrow \{\phi\}$ とする。もし $\min Z^k > Z_{t-1}$ なら、 $Z_t = Z_{t-1}$ および $\Pi_t = \Pi_{t-1}$ とする。
- Step 6** もし $Z_t \leq D$ のとき、 $B_c \leftarrow \{\phi\}$, $t \leftarrow t+1$ とする。Step 5 で得られた作業順列を第2段階の初期解として、第2段階へ進む。
- Step 7** もし $|B_c| = Q$ となれば計算終了。否ならば $t \leftarrow t+1$ として、Step 2 に戻る。

注) Step 7 で計算終了となる場合は、問題が実行不可能なので、 D の設定値を変更して計算をやり直す。

3.4 第2段階のアルゴリズム

第2段階では、(1) で与えられる目的関数値を改善するため、近傍解を生成するとき近傍サイズを最大にとり、交換の近傍サイズは $Q - N$ 、挿入の近傍サイズは $Q - 1$ とし、全体の近傍サイズを $K = 2Q - N - 1$ とする。アルゴリズムは、次のようになる。

- Step 1** 第1段階で得られた作業順列 Π_{t-1} およびその目的関数値 Z_{t-1} を第2段階の初期解とする。
- Step 2** ネゴシエータを作業順列 Π_{t-1} より一つランダムに選び、 Q_{nm} とする。
- Step 3** もし $Q_{nm} \notin B_c$ ならば Q_{nm} を集合 B_c に入れ、Step 4 に進む。もし $Q_{nm} \in B_c$ ならば Step 2 に戻る。
- Step 4** 交換と挿入を行って、新たに K 個の処理順列 $\Pi^k (k = 1, 2, \dots, K)$ を作成し、それぞれの目的関数値 Z^k を求める。
- Step 5** Step 4 で作成された解 Π^k および $Z^k (k = 1, 2, \dots, K)$ に対して、もし $\min Z^k \leq Z_{t-1}$ ならば、 $Z_t = \min Z^k = Z^{best}$ および $\Pi_t = \Pi^{best}$, $B_c \leftarrow \{\phi\}$ とする。もし $\min Z^k > Z_{t-1}$ ならば、 $Z_t = Z_{t-1}$ および $\Pi_t = \Pi_{t-1}$ とする。
- Step 6** もし $|B_c| = Q$ となれば、Step 5 で得られた解を準最適解として計算終了。否ならば $t \leftarrow t+1$ として、Step 2 に戻る。

注) Step 4 で作成された作業順列のうち $C_{max} > D$ となる順列は除かれる。このとき、Step 5 の K の値は除かれた順列数だけ少なくなる。

4. 計算例

本問題に対して、 $(N, H) = (15, 10), (15, 15), (20, 15), (20, 20)$ の4種類の規模のジョブショップスケジューリングベンチマーク問題を取り上げ、各規模について5事例ずつ用い、計算を行った。各事例の納期の値は、これまでに得られているメイクスパンの最適解あるいは準最適解 [1] C_{max}^{opt} の α 倍、すなわち、 $D = \alpha C_{max}^{opt}$ で、 α の値は、1.1, 1.2, 1.5 とそれぞれに設定し、計算を行った。

Fig. 1 は、一例として、規模 $(N, H) = (15, 10)$ の事例 la22 で $\alpha = 1.2$ とした場合の探索過程を示す。同図より、第1段階でメイクスパンの値が急速に減少し、 $t = 129$ でメイクスパンの値が点線で示された納期より小さくなり、第2段階の探索に入る。第2段階では、メイクスパンの値は納期内で振動しながら、(1) で与えられる目的関数値 Z は急速に減少していることが分かる。

Fig. 2 は、各規模の問題において、メイクスパン最小化問題の準最適解 [1] と本問題に対する準最適解を、(1) で与えられる Z の値と C_{max} 値で比較したものである。ここに、 C_{max}^* はメイクスパン最小化問題の C_{max} 準最適値であり、 Z^* はそのときの (1) の値を表す。また、各規模の5事例につき乱数を変えてそれぞれ10回の計算

Fig. 1 Example of search process for Problem la22 in the case of $(N, H) = (15, 10)$

Fig. 2 Relationship between Z and C_{max} for various benchmark problems

を行っており、 Z/Z^* および C_{max}/C_{max}^* の値は、このように試行を繰り返した $5 \times 10 = 50$ 回の平均値を表す。

同図より、 $C_{max}/C_{max}^* > 1$ で $Z/Z^* < 1$ であることから、本解法により納期制約の範囲内でメイクスパンを悪化させて最早ジョブと最遅ジョブの完了時間差を小さくできることがわかる。また同図より、 α を大きくして納期制約を緩和すれば、 Z の値は減少する。すなわち、納期制約を大きく緩和すれば、それだけ完了時間差をより小さくできることがわかる。

5. おわりに

本論文では、納期内同時処理完了をめざすジョブショップスケジューリング問題を定義して、この問題を解く局所探索法による二段階のアルゴリズムを提案した。本解法の特徴は、第1段階で納期を満たすスケジュールを早急に見だし、第2段階で、納期を満たし、すべてのジョブができるだけ同時に処理されるスケジュールを求めることである。計算例により、本アルゴリズムで得られたスケジュールの(1)で表す目的関数値は、メイクスパン最小化の解より、大幅に減少することがわかった。これは、実際の生産現場において、同時に出荷しなければならないジョブ群のスケジューリングに有用であると思われる。

参考文献

- [1] 中野, 田, 三宮: 改善された局所探索法によるジョブショップスケジューリング問題の解法; 電気学会論文誌 C, Vol. 121C, No. 10, pp. 1627-1633 (2001)
- [2] Y. Tian, N. Sannomiya, C. Xie and T. Sawaragi: A combined algorithm for solving job shop scheduling problems; *Preprints of 15th Triennial World Congress of IFAC*, T-Tu-M196 (2002)
- [3] M. Pinedo: *Scheduling-Theory, Algorithms, and Systems, Second Edition*, Prentice-Hall (2002)