

A-009

# 選択グラフの性質とジョブショップスケジューリング問題

## Properties of a Disjunctive Graph and Job Shop Scheduling

榊原 静†  
Shizu Sakakibara

中森眞理雄‡  
Mario Nakamori

### 1. はじめに

**選択グラフ** (離接グラフ, disjunctive graph) は, すでに順番が決まっている点間の一連の流れを示す**固定辺** (conjunctive arc) と呼ばれる有向辺と, 先行関係がなく排他関係だけがある点間を結ぶ**選択辺** (disjunctive arc) とよばれる両方向の有向辺で構成されるグラフで, スケジューリング問題などの順序づけ問題の記述に利用されている。

ジョブショップスケジューリング問題は, Balas による分枝限定法での解法[1]が基本的である。ジョブショップスケジューリング問題を表現した選択グラフは, いくつかの特徴的な性質を持ち, その性質を利用することにより問題を効果的に解決できるのではないかと考え, 選択グラフの性質を調べ, その性質を数種類のジョブショップスケジューリング問題の解法に応用することを試みた。

### 2 スケジューリング問題における選択グラフ

#### 2.1 選択グラフ

Balas[1]のジョブショップスケジューリング問題では,  $M$  台の相異なる機械で,  $I$  種類のアイテムに作業を行なうシステムを取り扱う。一台の機械では一度に一つの作業しかできず, 一つのアイテムは一度に, 一台の機械でのみ作業され, 各機械での作業順序は任意である。各作業の作業時間が決まっており, システム全体での作業開始から, すべての作業が終了するまでの最大完了時間を最小化する問題である。

このシステムを表す選択グラフは, 作業の数  $M \times I$  個の点に加えて, システム全体の作業開始とすべての作業の完了を示すダミーの点  $0, n$  を持つ。各機械  $m$  で行なわれる  $I$  個の作業の点間を選択辺で結ぶ。機械  $M$  での作業の点とその選択辺からなる部分グラフを  $G_m$  とする。点  $0$  からグラフ  $G_1$  の各点,  $G_m$  の各点から点  $n$  へ固定辺を設け, 各グラフ  $G_m$  と  $G_{m+1}$  の同じアイテムの点間に, アイテムの流れを作業の実行順に固定辺を設けたグラフ  $D$  を選択グラフと呼ぶ (図 1)。全ての選択辺をどちらか一方の向きに決めると (グラフ  $D^*$ ), ひとつの実行可能な処理順序が得られる。

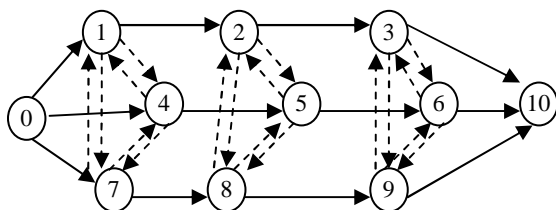


図 1 機械数 3, アイテム数 3 の選択グラフの例

†東京農工大学大学院工学研究科情報コミュニケーション工学専攻

‡東京農工大学工学部情報コミュニケーション工学科

#### 2.2 パラシュによる解法

まず実行可能な初期解  $F_0$  を何らかの方法で作出し, すべての選択辺を自由とする。  $F_0$  を根として解の木を作る。木の作り方は次のとおりである。

一般に解  $F_r$  が得られているとして,  $F_r$  における自由な選択辺が,  $(i_1, j_1), (i_2, j_2), \dots, (i_l, j_l)$  とするとき,

1)  $(j_1, i_1)$  を固定,  $(i_2, j_2), \dots, (i_l, j_l)$  を自由。

2)  $(i_1, j_1), (j_2, i_2)$  を固定,  $(i_3, j_3), \dots, (i_l, j_l)$

.....

l)  $(i_1, j_1), \dots, (i_{l-1}, j_{l-1})$  を固定,  $(j_l, i_l)$  を固定。

という  $l$  個の子を作り, 各子から再帰的に子を作り出す。

すべての組み合わせの数は極めて大きいので, 各種のテストを用いて, 一部の組み合わせだけの生成にとどめる。

たとえばパラシュは次の方法を提案している。

(i)  $F_r$  において固定された選択辺のみを用いたグラフから子孫の解の下界値  $v(F_r)$  を求める。  $v(F_r)$  がすでに得られている最良の解以上なら上記 1),  $\dots$ , l) は不要。

(ii)  $F_r$  におけるクリティカルパスに含まれない自由な選択辺は逆転しない。

$F_0$  においてはすべての選択辺は自由である。暫定解  $F^*$  の目的関数値を  $v^*$  とする。(初期状態では無限大)

以上のアルゴリズムは以下の関数  $b\_and\_b$  を再帰的に呼び出すことによって実行できる。

```

b_and_b ( F_r ) {
  下界値 v(F_r) を計算;
  if ( v(F_r) < v* ) {
    クリティカルパス d_{0,n}, 各評価値, 分枝の候補を計算;
    if ( d_{0,n} < v* ) 暫定解 F* と, v* を更新;
    F_n = F_r;
    while (分枝の候補がある) {
      (i,j) = 最も評価値の低い分枝の候補;
      F_c = F_n; F_c に (j,i) を追加; F_n に (i,j) を追加;
      b_and_b ( F_c );
    }
  }
}

```

### 3. 特殊なスケジューリング問題と選択グラフの性質

本論文で扱うスケジューリング問題(1),(2),(3)を示す。

(1) Balas[1]の問題で, 同じ機械で行なう作業に関して, 作業間に空けるべき時間が与えられている。この時間を直前の作業時間に加えて考えても良いが, こうすると, 作業の順番によって実行時間が変化することになる。

(2) Balas[1]の問題で, すべてのアイテムで, 機械 1, 2,  $\dots$ ,  $m$  の順で作業が行なわれる場合 (フローショップスケジューリング問題)。

(3) 上記(2)の問題で, 作業ごとに実行できない時間帯が定められている場合。

選択グラフの性質(a)から(c)を挙げる.

- (a) Balas の解法[1]は選択グラフの 1 つの点から出るすべての辺の長さが同じである ( または 1 つの点に入るすべての辺の長さが同じである ) 場合にしか使えない. 問題(1)を解くために, 選択グラフの各辺を作業間ごとに作業時間に作業間の時間を加えた長さにすると, グラフ  $D^*$  のクリティカルパスがシステムの最大完了時間と異なる場合がある. ( 作業  $a, b_1, b_2, \dots, b_l$  の順に実行されるものが作業  $a, b_l$  間の時間が極端に長いと, クリティカルパスが  $a, b_l$  を経由する ) よって, Balas [1] のアルゴリズムが適用できず, 選択グラフに余計な辺や点を追加する必要がある.
- (b) 問題(2)では, グラフ  $D$  で, アイテム  $i$  での作業を表す点の集合を  $O_i$  とする. グラフ  $D^*$  のクリティカルパスには, 各アイテム  $i$  で  $O_i$  のうち少なくともひとつは含まれる.
- (c) 選択グラフは, 以下に述べる, 各機械での作業の順番をアイテムの番号によって記述した, 順番保持表を用いて表現できる.

### 3.1 順番保持表

性質(a)の問題解決のために, 各機械での作業の順番をアイテムの番号によって記述する. 各機械  $m$  で作業を実行する順番で作業のアイテム番号を並べ, この配列を順番表と呼ぶ. また, アイテム  $i$  の作業が, 機械  $m$  で何番目に実行されるかを調べる探索を避けるため, 順番表に対して, 各アイテムが各機械で何番目に実行される作業であることを記述する. この配列を番号表と呼ぶ. 対になった順番表と番号表で, 作業の実行順序を保持することによって, その都度, 選択グラフからグラフを再構成してクリティカルパスを求めるのではなく, 高速に解を求める. この順番表と番号表の対を順番保持表と呼ぶ ( 図 2 ).

フローショップ問題 ( 問題(2), (3) ) では, 各アイテムでの作業は必ず機械 1, 2,  $\dots$ ,  $M$  の順で実行されるため, グラフ  $D^*$  でのクリティカルパスは, 機械 1, 2,  $\dots$ ,  $M$  の順で各機械での実行順に, 各作業の実行する時間をひとつずつ求めることで計算できる. よって各機械での実行順序がわかっているならば, クリティカルパスは対応する順番保持表で, 機械 1 の 1 番目の作業から 2 番目, 3 番目と作業の実行時間を計算し, 機械 2, 機械 3 と計算することで求まる.

	機械 1	機械 2	機械 3		機械 1	機械 2	機械 3
1 番目の作業	1	2	3	アイテム 1	1	2	2
2 番目の作業	2	1	1	アイテム 2	2	1	3
3 番目の作業	3	3	2	アイテム 3	3	3	1
	順番表				番号表		

図 2 機械数 3, アイテム数 3 の順番保持表の例

## 4. 選択グラフの性質を用いた各問題の解法

### 4.1 性質(a)を利用した下界値の改善

問題(2)で, 下界値にあたる経路に作業が含まれていないアイテムがあるとき, そのアイテムの作業のうち一番小さい値を加算することができる. Balas[1]による下界値を, 性質(a)を用いて改善し, 比較実験を行なった. 実験に用いたデータは[3]にあるものを利用した.

結果を表 1, 表 2 に示す. 表 1 は, 分枝数による比較である.

平均 2.66 の改善比で減少した. 表 2 は, 実行時間による比較である. ただし, 計測値が 1 秒以下になったものは, 0.5 秒として計算した. 平均 2.43 の改善比で減少した.

表 1 各下界値による分枝数

	本手法	Balas[1]	改善比
平均	8303.02	20652.68	2.66
最大	112877	273773	21.41
最小	16	19	1

表 2 各下界値による実行時間 ( 秒 )

	本手法	Balas[1]	改善比
平均	8.98	21.78	2.43
最大	137	311	2.27
最小	0.5	0.5	1

### 4.2 性質(a), (c)を用いた特殊なスケジューリング問題の解法

問題(1)は, 順番保持表を用いることで解決できる.

また, 順番保持表は問題(3)のようにフローショップで, 作業のできない時間を避けるため, ある作業の開始または終了の時間によっては作業時間が変化する問題にも有効である. グラフで, 固定点からの単純な最長距離計算は, 通過する辺数ごとに求まるため, 固定点から特定の点までの最長距離は最後まで計算をしないとわからない. このことから, 計算した結果, 所定の条件を満たないと判定された作業を条件に合うよう変更すると, 再計算が必要となる. これを避けるには, あらかじめダミーの点を加えて常に条件を満たすよう調整しなければならない. 順番保持表を用いると, 機械  $m+1$  による作業の各終了時間を計算する時には, 機械  $m$  までの作業の終了時間は求まっているため, 条件を満たすよう調整しながら計算できる.

## 5. おわりに

スケジューリング問題における選択グラフの性質から, フローショップスケジューリング問題の下界値の改善と, 特殊なスケジューリング問題の解法を考えた.

本手法の下界値を用いることで, Balas[1]による下界値より平均 2.66 の改善比で分枝数が減少した.

各機械でのアイテムの作業順序による順番保持表を提案し, 条件を付加した特殊な問題の解法を考えた.

### 参考文献

- [1] E.Balas: Machine Sequencing via disjunctive Graphs :An Implicit Enumeration Algorithm, J. Operations Research Society of America, 17, 941-957 (1969).
- [2] 中川: 鉄鋼業における離散事象システムの最適化, 室田(編): 離散構造とアルゴリズム III, 近代科学社, 1994.
- [3] E.Demirkjol, S.Mehta, R.Uzsoy: Benchmarks for shop scheduling problem, European J. Operational Research, 109, 137-141(1998).
- [4] 伊理: 線形計画法, 共立出版, 1986.
- [5] 田村, 村松: 最適化法, 共立出版, 2002.
- [6] 日本オペレーションズ・リサーチ学会 (編): OR 辞典 2000, (社)日本オペレーションズ・リサーチ学会, 2000.