2001年度日本オペレーションズ・リサーチ学会
春季研究発表会

凸型時間ずれコストをもつ資源制約スケジューリング問題

01405524 京都大学大学院情報学研究科 *野々部 宏司 NONOBE Koji
01001374 京都大学大学院情報学研究科 茨木 俊秀 IBARAKI Toshihide

1 はじめに

資源制約スケジューリング問題 (Resource Constrained Project Scheduling Problem, RCPSP) は, 限られた資源の下, 複数の作業をどのように処理するか決定する問題であり, 幅広い分野において現れるスケジューリング問題を定式化することができる [2]. そのため, 高性能な RCPSP アルゴリズムは, 様々なスケジューリング問題を扱うことのできる**汎用スケジューラ**として用いることができる. この考えに基づき, 我々はこれまでに RCPSP に対する近似解法の開発を行ってきた [3]. しかし最近の SCM(サプライチェーンマネジメント) の議論の中で, 納期ずれ時間和など, いわゆる非正規 (nonregular) 評価基準を持つ問題の重要性が指摘されているにもかかわらず, これらに対しては, しばしば効果的な適用を行うことができなかった. そこで本研究では, この問題を解消するため, 新たに**凸型時間ずれコスト**を評価基準として導入し, これを最小化する問題を考える. そして, この問題に対して, 局所探索に基づく近似解法を適用する.

2 問題定義

資源集合を $\mathcal{R}$, 作業集合を $\mathcal{J}$ とする. 時間軸は離散化され, 各資源 $r \in \mathcal{R}$ は, 毎単位時間, ある定められた量 K_r (≥ 0) ずつ供給されるものとする. 各作業 $i \in \mathcal{J}$ の処理時間は非負整数 p_i で表され, その処理中, 毎単位時間, 各資源 r を k_{ir} (≥ 0) ずつ消費するものとする. また, 作業間には先行関係があり, 半順序 $\prec$ で与えられる. すなわち, $i \prec j$ のとき, 作業 i の処理が完了するまで作業 j を開始することはできない. さらに本研究では, 順序付けされた作業の組 $(i,j) \in \mathcal{T}$ $(\subseteq \mathcal{J} \times \mathcal{J})$ に対し, それらの開始時刻 s_i, s_j の差 Δ_{ij} に応じたコスト $c_{ij}(\Delta_{ij})$ を与えることができる. 以下, これを時間ずれコストとよび, $c_{ij}(\Delta)$ は凸関数であるとする. 本研究では, 資源制約と先行制約の下, 時間ずれコスト和を最小化するよう, 各作業の開始時刻 $s = (s_i \mid i \in \mathcal{J})$ を決定することを目的とする. すなわち,

$$
\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in\mathcal{T}} c_{ij}(\Delta_{ij}) \\
\text{subject to} \quad & \sum_{i\in\mathcal{J}_{rt}} k_{ir} \leq K_r, \quad r \in \mathcal{R}, t \in [1,T], \\
& s_j - s_i \geq p_i, \quad i \prec j, \\
& \Delta_{ij} = s_j - s_i, \quad (i,j) \in \mathcal{T}, \\
& s_i \in \{0,1,2,\ldots,T\}, \quad i \in \mathcal{J}.
\end{aligned}
$$

ここで, $\mathcal{J}_{rt} = \{i \mid 0 \leq (t-1) - s_i \leq p_i - 1\}$, すなわち, 期間 $[t-1,t)$ において処理中である作業集合, T は開始時刻 s_iの上限をそれぞれ表す.

以下に, 時間ずれコストを用いた例を 2 つ挙げる.

時間制約

時間制約 (temporal constraint) は先行制約の一般化であり, 定数 d_{ij} を用いて,

$$s_j - s_i \geq d_{ij},$$

と表せる [2]. d_{ij} は負の値でもよく, その場合, 作業 i の作業 j に対する**最大遅延時間** (maximal time lag) が$-d_{ij}$ であることを示す. 時間制約は, 十分大きな定数 M を用い, 時間ずれコストを

$$c_{ij}(\Delta_{ij}) = \begin{cases} M\Delta_{ij}, & \Delta_{ij} \leq d_{ij}, \\ 0, & \text{その他}, \end{cases}$$

とすることで記述できる.

納期ずれ時間最小化

作業 i の納期を $\overline{d}_i$ としたとき, 作業 i の納期ずれ時間は $|s_i + p_i - \overline{d}_i|$ で与えられる. これは, すべての作業に先行し, 時刻 0 に完了する仮想作業 0 (ソース, source) を用いて,

$$c_{0i}(\Delta_{0i}) = |\Delta_{0i} + p_i - \overline{d}_i|$$

と表現できる. より一般的に, 開始時刻 s_i に対して定まる凸型コスト $\overline{c}_i(s_i)$ は, $c_{0i}(\Delta_{0i}) = \overline{c}_i(s_i)$ とすることで記述できる.

3 アルゴリズム

RCPSP は一般に NP 困難であるため, 本研究では, 局所探索法に基づく近似解法 [4] の構築を行う.

3.1 概要

まず, 資源制約がない場合, 凸型時間ずれコスト和を最小化する問題は**凸費用流問題**に帰着することができ,

$O(nm\log n\log(nT))$ 時間で解けることに着目する [1]. ここで, $n = |\mathcal{J}|, m =$ (先行制約の数) $+ |T|$ である. そして, 以下の 2 つの条件を満たす作業集合 $\mathcal{J}$ 上の半順序 $\rightarrow$ を導入する:

1. $i \prec j \Rightarrow j \not\rightarrow i$,
2. すべての $i, j \in \mathcal{J}'$ $(i \neq j)$ に対し, $i \not\rightarrow j$ を満たす任意の部分集合 $\mathcal{J}' \subseteq \mathcal{J}$ (すなわち, 任意の antichain $\mathcal{J}'$) に対し,

$$\sum_{i \in \mathcal{J}'} k_{ir} \leq K_r, \quad r \in \mathcal{R}.$$

すなわちこの半順序関係 $\rightarrow$ は, 資源を共有する作業に対し, その競合を避けるよう処理順序を定めるものである. よって, 元々与えられている先行関係 $\prec$ に加え, $\rightarrow$ を作業間の先行関係として与えれば, 資源制約は冗長となり取り除くことができる. 資源制約がなければ, 凸型時間ずれコスト和最小化問題を (多項式時間で) 解くことで $\rightarrow$ の制約下での最適スケジュールを求めることができる. 結局, コストを最小にするような半順序 $\rightarrow^*$ をいかに求めるかが問題となる. 本研究では, この目的のために局所探索法 (およびそれに基づくメタ解法) [4] を用いる. 以下では, 局所探索法の実装において重要な, 解表現と近傍について述べる.

3.2 解表現

局所探索において, 半順序をそのまま解表現として用いることは効率的でないと考えられる. そこで本研究では, 作業集合 $\mathcal{J}$ の複製を 2 個, $\mathcal{J}^- = \{i^- \mid i \in \mathcal{J}\}$ と $\mathcal{J}^+ = \{i^+ \mid i \in \mathcal{J}\}$ を用意し, それらに含まれる $2n$ 要素の順列 $\sigma : \{1, 2, \ldots, 2n\} \mapsto \mathcal{J}^- \cup \mathcal{J}^+$ を解表現として用いる. ただし,

$$\sigma^{-1}(i^-) < \sigma^{-1}(i^+), \quad i \in \mathcal{J},$$

であるとする. そして解 σ に対し, 作業集合 $\mathcal{J}$ 上の半順序関係 $\rightarrow_\sigma$ を

$$i \rightarrow_\sigma j \Leftrightarrow \begin{array}{l} \sigma^{-1}(i^+) < \sigma^{-1}(j^-), \\ \exists r \in \mathcal{R}, k_{ir} > 0 \text{ and } k_{jr} > 0, \end{array}$$

と定義し, 資源制約の代わりに, 先行制約 $\rightarrow_\sigma$ の下での凸型時間ずれコスト和最小化問題を考える. すなわち i^-, i^+ は, それぞれ作業 i の開始時刻, 完了時刻に対応しており, 順列 σ はそれらの順序を定めていると考える. (ただし, 資源を共有しない 2 作業 i, j に対しては, それらの処理順序を制限することはしない.) そして, 得られた最適スケジュール s^σ の評価値を解 σ の評価値とする.

なお, 順列 σ に対応する半順序 $\rightarrow_\sigma$ が, 必ずしも前節で述べた 2 条件を満たすとは限らないため,

$$i \prec j \Rightarrow \sigma^{-1}(i^+) < \sigma^{-1}(j^-),$$

といった条件を加えるなどの必要がある.

3.3 近傍

現在の解 σ に対応するスケジュール s^σ において,

$$i \not\prec j,\ i \rightarrow_\sigma j \Rightarrow s_i + p_i < s_j, \tag{1}$$

が成り立つならば, s^σ は最適スケジュールである. 逆に, (1) が成り立たない (すなわち, $s_i + p_i = s_j$ である) 場合, 先行関係 $i \rightarrow_\sigma j$ を取り除くことで, コストの減少が期待できる. そこで順列 σ に対し, i^+ を j^- の直後に (あるいは j^- を i^+ の直前に) 移動させた後, 3.1節で述べた 2 条件を満たすよう修正するという操作を行う. このようにして得られる順列 (解) σ' の集合を, 解 σ の近傍 $N(\sigma)$ とする.

なお, s^σ を求めるために凸費用流問題を解くことになるが, 凸費用流問題において, 各作業 $i \in \mathcal{J}$ はネットワークの節点に, 各先行関係 $i \rightarrow_\sigma j$ はネットワークの枝 (i, j) に対応づけられる. もし枝 (i, j) の流量が 0 であれば, たとえ $s_i + p_i = s_j$ であっても, 先行関係 $i \rightarrow_\sigma j$ が取り除かれることでコストが減少することはない. よって, 枝 (i, j) の流量が正であるときに限り (このとき $s_i + p_i = s_j$ が成り立つ), $i \rightarrow_\sigma j$ を取り除くことを考えればよい.

4 最後に

本研究では, より広範なスケジューリング問題に対応できるよう, RCPSP に凸型時間ずれコストを導入し, これを最小化する問題を考えた. そして, この問題に対する近似解法の適用を行った.

なお, 詳しい計算結果は当日発表させて頂く予定である.

参考文献

[1] R.K. Ahuja, D.S. Hochbaum, J.B. Orlin, "Solving the convex cost integer dual network flow problem," in: G. Cornuejols, R.E. Burkard and G.J. Woeginger (eds.), *Lecture Notes in Computer Science* 1610 (IPCO'99), pp. 31–44, 1999.

[2] P. Brucker, A. Drexl, R. Möhring, K. Neumann and E. Pesch, "Resource-constrained project scheduling: Notation, classification, models, and methods," *European Journal of Operational Research* 112, pp. 3–41, 1999.

[3] K. Nonobe and T. Ibaraki, "Formulation and tabu search algorithm for the resource constrained project scheduling problem," in: C. Ribeiro and P. Hansen (eds.), *Essays and Surveys in Metaheuristics*, Kluwer Academic Publishers, 2001 (to appear).

[4] 柳浦睦憲, 茨木俊秀, 組合せ最適化 —メタ戦略を中心として—, 朝倉書店, 2001.