重み付き総遅延最小化スケジューリング問題の計算複雑性
Computational complexity of scheduling problems to minimize total weighted tardiness

数学専攻 浅井大樹
Hiroki ASAI

1 はじめに

スケジューリング理論は様々な分野に応用されており, その代表的な例として交通機関における運行スケジューリングが挙げられる. これらのスケジューリングに際しては, 運行に必要な各 job に対する遅延を最小化する総遅延最小化スケジューリング問題や, 各 job の重要度を表す重み付けを行なった上で遅延を最小化する重み付き総遅延最小化スケジューリング問題を考えることが多い. しかし, これらの問題の多くは NP-hard であることが既に知られており (Lenstra-Kan-Brucker [5]), 実社会のスケジューリング問題を解く際には, 制約条件をどの程度付加すると計算複雑性の観点から現実的なスケジューリング問題として解くことが可能かをあらかじめ知っておくことが重要となる. また, job を処理するための装置である machine が大量に存在する場合の計算複雑性を求めるには, machine を 1 台に限定した 1 機械スケジューリング問題に対する計算複雑性を求めることが必要となる.

1 機械総遅延最小化スケジューリング問題および 1 機械重み付き総遅延最小化スケジューリング問題の計算複雑性に関する研究では, J. K. Lenstra, A. H. G. Rinnooy Kan, P. Brucker が 制約条件のない重み付き総遅延最小化スケジューリング問題が NP-hard であることを示した (Lenstra-Kan-Brucker [5]). さらに, J. Du と Y. T. Leung は任意の job に対して, その重みを 1 としたとき, すなわち制約条件のない総遅延最小化スケジューリング問題は NP-hard であることを示した (Du-Leung [3]). またこれらと並行して, 重みや締切時刻に対してより強い制約を付した重み付き総遅延最小化スケジューリング問題についての研究も行われており, 特に, E. M. Arkin と R. O. Roundy は任意の job に対して, その重みがその job の処理時間に等しいような重み付き総遅延最小化スケジューリング問題は NP-hard であることを示した (Arkin-Roundy [1]). さらに, 各 job の実行可能時刻に対して制約を加えた重み付き総遅延最小化スケジューリング問題の計算複雑性に関しては, J. K. Lenstra, A. H. G. Rinnooy Kan, P. Brucker が 任意の job の実行可能時刻が 0 以上となるような重み付き総遅延最小化スケジューリング問題が NP-hard であることを示した (Lenstra-Kan-Brucker [5]).

本論文では, 任意の job に対してその重みがその job の処理時間に等しくかつ, 各 job の実行可能時刻に対して制限を加えた 1 機械重み付き総遅延最小化スケジューリング問題が NP-hard であることの証明を与える.

2 スケジューリングの定義と記法

本節では, 本論文で用いる定義と記法について述べる. スケジューリング理論の定義に関しては Pinedo [7] を, 計算複雑性理論の定義に関しては Garey-Johnson [4] を参照する.

スケジューリング (scheduling) とは, いくつかの **機械** (machine) によっていくつかの **仕事** (job) を処理するとき, machine の環境 α に対して, 制約条件 β を満たすような job の処理順序を定めることをいい, このときの処理順序を **スケジュール** (schedule) という. 特に, 目的関数 γ を最小化するスケジュール S を **最適スケジュール** (optimal schedule) という. また, 最適スケジュール S を求める問題を **スケジューリング問題** (scheduling problem) といい,

$\alpha \,|\, \beta \,|\, \gamma$ で表す．さらに, スケジュール上で使用される $m+1$ 台の machine の集合を $M = \{M_0, M_1, \ldots, M_m\}$, それらの machine で処理する $n+1$ 個の job の集合を $J = \{J_0, J_1, \ldots, J_n\}$ とする．また, スケジュールを表す際に使用する図を **ガントチャート** (Gantt chart) という (図 1)．本論文では machine が 1 台の場合のみを扱うため, 以後 $m = 0$, $\alpha = 1$ とする．さらに, あるスケジュールにおいて J_i が J_k よりも前に処理されることを, $J_i < J_k$ で表し, J_i が J_k よりも後に処理されることを, $J_i > J_k$ で表す.

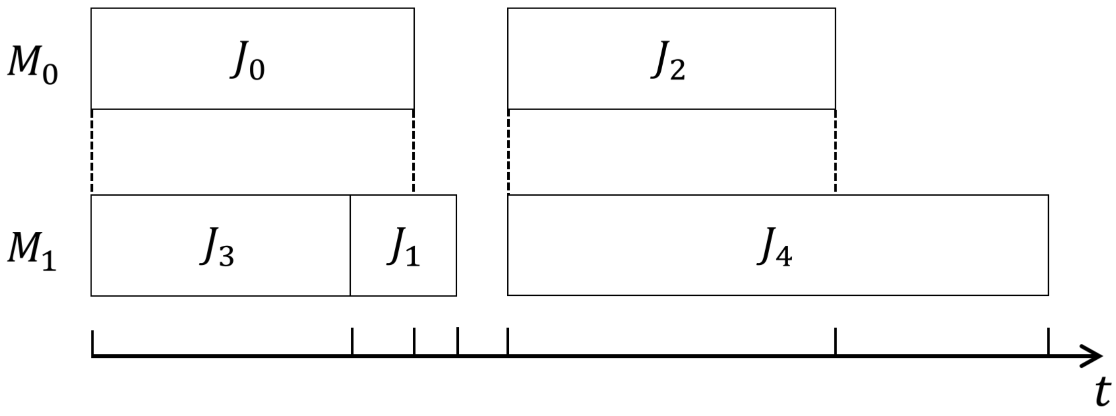

図 1 ： $m = 1$, $n = 4$ のときのスケジュールを表したガントチャートの例

J_i の **処理時間** (processing time) とは, J_i を machine で処理するために要する時間のことをいい, p_i で表す．また, J_i の **実行可能時刻** (release date) とは, machine が J_i の処理を開始することが可能となる時刻をいい, r_i で表す．さらに, J_i の **完了時刻** (completion time) とは, machine が J_i の処理を完了する時刻をいい, C_i で表す.

J_i の **締切時刻** (due date) とは, machine が J_i の処理を完了しなければならない時刻をいい, d_i で表す．また, J_i の **重み** (weight) とは, スケジュール全体におけるその job の重要度を表す数値をいい, w_i で表す．本論文では, 任意の $J_i \in J$ に対する weight を正の整数とするが, このようにしても weight 間の大小関係は変わらないことから, スケジューリング問題の一般性は失われない.

任意の $J_i \in J$ に対して, $max(C_i - d_i, 0)$ を J_i の **遅延** (tardiness) といい, T_i で表す (図 2)．また, 本論文では $max(C_i - d_i, 0)$ を $(C_i - d_i)^+$ と表すこととする．さらに, 各 $J_i \in J$ に対して, その weight と tardiness の積, すなわち $w_i T_i$ を **重み付き遅延** (weighted tardiness) といい, 全ての $J_i \in J$ に対する weighted tardiness の総和, すなわち $\sum_{i=0}^{n} w_i T_i$ を **重み付き総遅延** (total weighted tardiness) という.

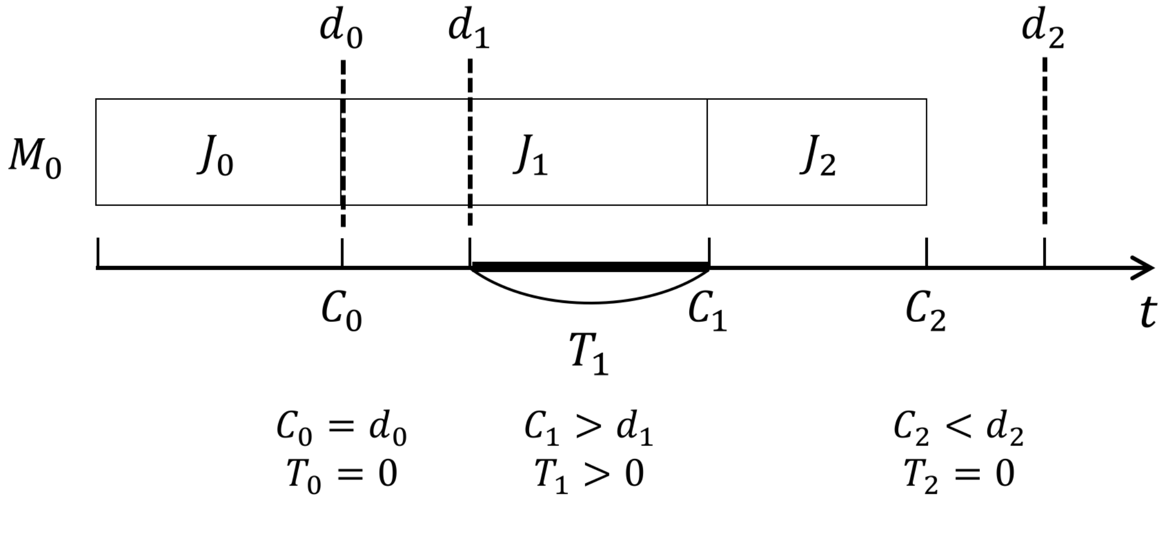

図 2 ： $n = 2$ のときのスケジュールを表したガントチャートに対する各 job の遅延の例.

スケジューリングの条件が **preemption** であるとは, ある job を処理するとき, その処理を任意の時点で中断させ, 他の job の処理を行うようなスケジュールをスケジューリングすることを認めるときをいう. machine が 1 台の場合, preemption の有無によって, スケジューリングされた重み付き総遅延最小化スケジュール S に対する重み付き総遅延の値は変化しないことが知られている (McNaughton [6]). よって本論文のスケジューリング問題では, preemption を考えないこととする.

ある machine が job の処理を開始するまで待機している状態を **idle** という. また, あるスケジュールにおいて, job の処理を開始することが可能であるのに idle しているような machine が存在しないとき, そのようなスケジュールを **non-delay** であるという. machine が 1 台でかつ preemption を考えないような重み付き総遅延最小化スケジューリング問題の場合, スケジューリングされた non-delay でないスケジュールは, idle している時間を削除し job の処理開始時刻を早めることで, total weighted tardiness が等しいまたはより小さい non-delay なスケジュールにスケジューリングし直すことが可能であるから, non-delay なスケジュールが常に最適スケジュールとなる. よって, 本論文では non-delay なスケジュールのみを考えることとする.

ある problem Q が **決定問題** (decision problem) であるとは, Q に対する解が Yes または No である問題をいう. また Q, Q' を problem とするとき, Q' が Q へ **還元可能** (reducible) であるとは, Q' の任意の instance に対して Q の instance を解くと Q' の instance も解けるように Q' の instance から Q の instance を polynomial time で construct することができるときをいい, $Q' \propto Q$ で表す.

また, problem Q が **NP 完全** (NP-complete) であるとは, $Q \in \mathrm{NP}$ であり, 任意の $Q' \in \mathrm{NP}$ に対して, $Q' \propto Q$ を満たすときをいう. さらに, problem Q が **NP 困難** (NP-hard) であるとは, 任意の $Q' \in \mathrm{NP}$ に対して, $Q' \propto Q$ を満たすときをいう.

3 1 機械重み付き総遅延最小化スケジューリング問題

本節では, 1 機械重み付き総遅延最小化スケジューリング問題の概要について述べる.

$\alpha = 1$, $\gamma = \sum_{i=0}^{n} w_i T_i$ を満たすスケジューリング問題, 即ち $1 \,||\, \sum_{i=0}^{n} w_i T_i$ を, **1 機械重み付き総遅延最小化スケジューリング問題** (one machine scheduling problem to minimize total weighted tardiness) という.

1 機械重み付き総遅延最小化スケジューリング問題に対しては, 次のことが知られている.

定理 3.1. (Lenstra-Kan-Brucker [5]) スケジューリング問題 $1 \,||\, \sum_{i=0}^{n} w_i T_i$ は NP-hard である.

4 release date を制限した 1 機械重み付き総遅延最小化スケジューリング問題

本論文では, weight 及び release date を制限した 1 機械重み付き総遅延最小化スケジューリング問題, 特に $1 \,|\, w_i = p_i,\, r_i = 0\,(0 \le i \le a),\, r_i = b\,(a+1 \le i \le n) \,|\, \sum_{i=0}^{n} w_i T_i$ が NP-hard であることを, 以下の Subset Sum Problem が この問題の決定問題へ reducible であることを利用して示した.

Subset Sum Problem (Karp, 1972)
INSTANCE : 正の整数を要素に持つ有限集合 A, 正の整数 B.
QUESTION : 要素の総和が B となるような A の部分集合 A' は存在するか.

次の定理が知られている.

定理 4.1. *Q, Q' を problem とする. Q' が NP-complete であり, $Q \in \mathrm{NP}$, $Q' \propto Q$ ならば, Q は NP-complete.*

さらに, weight を制限した 1 機械重み付き総遅延最小化スケジューリング問題について, 次のことが知られている.

定理 4.2. (Arkin-Roundy [1]) スケジューリング問題 $1 \mid w_i = p_i \mid \sum_{i=0}^{n} w_i T_i$ は NP-hard である.

E. M. Arkin と R. O. Roundy が定理を示す際に用いた instance を応用すると, 次を示すことができる.

定理 4.3. $0 \leq a, b \leq n$ を満たす任意の a, b に対して,
スケジューリング問題 $1 \mid w_i = p_i,\, r_i = 0\,(0 \leq i \leq a),\, r_i = b\,(a+1 \leq i \leq n) \mid \sum_{i=0}^{n} w_i T_i$ は NP-hard である.

5 due date を制限した 1 機械重み付き総遅延最小化スケジューリング問題

weight 及び due date に制限を加えた 1 機械重み付き総遅延最小化スケジューリング問題について, 次のことが知られている.

定理 5.1. (Cheng-Ng-Yuan-Liu [2]) $0 < \alpha < 1$ に対して, スケジューリング問題 $1 \mid d_i = \alpha p_i + q,\, w_i = k p_i \mid \sum_{i=0}^{n} w_i T_i$ は NP-hard である.

6 終わりに

任意の job に対して, その weight がその job の processing time に等しいという制約条件に release date の制限を加えた 1 機械重み付き総遅延最小化スケジューリング問題が NP-hard であることを本論文で示した. 今後は, これに due date の制限を加えた 1 機械重み付き総遅延最小化スケジューリング問題に対する計算複雑性の証明に取り組んでいきたい. スケジューリング問題に対して様々な制限を加える程, 計算複雑性を求めるための instance やその計算は複雑になることから, 本論文で計算複雑性を示したスケジューリング問題に due date の制限を加えた 1 機械重み付き総遅延最小化スケジューリング問題に対する計算複雑性の証明に際しては, 新たな instance の構成法やその計算手法について考察することが必要となるであろう.

参考文献

[1] E. M. Arkin, and R. O. Roundy, Weighted-tardiness scheduling on parallel machines with proportional weights, Operations Research Vol.39 No.1 (1991), 64-81.

[2] T. C. E. Cheng, C. T. Ng, J. J. Yuan, and Z. H. Liu, Single machine scheduling to minimize total weighted tardiness, European Journal of Operational Research 165 (2005), 423-443.

[3] J. Du, J. Y. T. Leung, Minimizing total tardiness on one machine is NP-hard, Mathematics of Operations Research Vol.15 No.3 (1990), 483-495.

[4] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W H Freeman & Co, 1979.

[5] J. K. Lenstra, A. H. G. Rinnooy Kan, and P. Brucker, Complexity of machine scheduling problems, Annals of Discrete Mathematics Vol.1 (1977), 343-362.

[6] R. McNaughton, Scheduling with deadlines and loss functions, Management Science Vol.6 No.1 (1959), 1-12.

[7] M. L. Pinedo, Scheduling Theory, Algorithms, and Systems, Springer, 2016.