離散最適化基礎論 第 2 回
解の表現

岡本 吉央
okamotoy@uec.ac.jp

電気通信大学

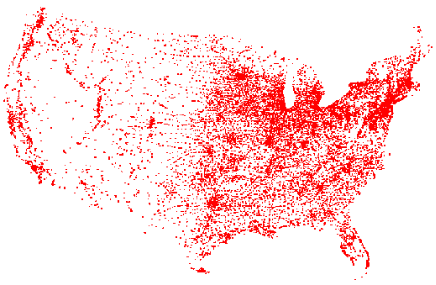
2013 年 10 月 18 日

最終更新：2013 年 10 月 18 日　16:40

目次

巡回セールスマン問題：例

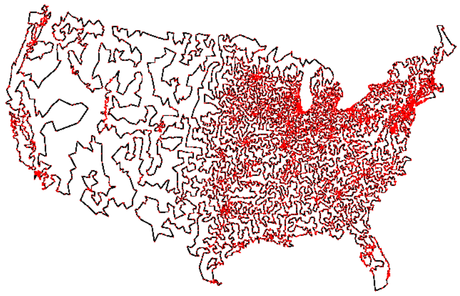
アメリカ合衆国の 13509 都市



http://www.math.uwaterloo.ca/tsp/gallery/idata/usa13509.html

巡回セールスマン問題：例 続き

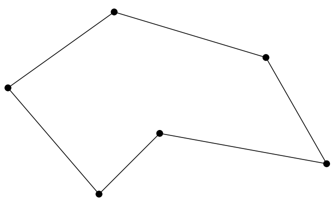
アメリカ合衆国の 13509 都市



http://www.math.uwaterloo.ca/tsp/gallery/itours/usa13509.html

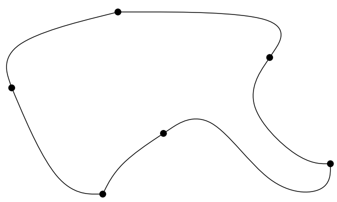
巡回セールスマン問題：別の例

ユークリッド平面上の巡回セールスマン問題



巡回セールスマン問題：別の例

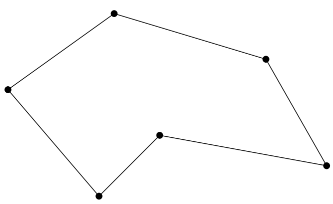
ユークリッド平面上の巡回セールスマン問題



**先ほどと同じ順に都市を訪問しているが，
最適解になりえない**

巡回セールスマン問題：別の例

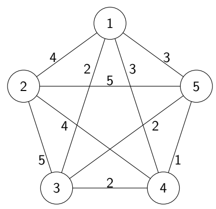
ユークリッド平面上の巡回セールスマン問題



最適解では必ず都市間を直線的に移動する

教訓 1
「解の候補」として，最適解となる可能性のないものは考えなくてよい

巡回セールスマン問題：許容集合



	1	2	3	4	5
1	0	4	2	3	3
2	4	0	5	4	5
3	2	5	0	2	2
4	3	4	2	0	1
5	3	5	2	1	0

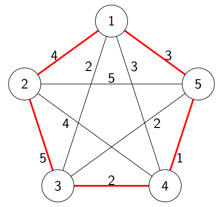
許容集合 $=\{(1,2,3,4,5),(1,2,3,5,4),(1,2,5,3,4),(1,5,2,3,4),$
$(5,1,2,3,4),(5,1,2,4,3),(1,5,2,4,3),(1,2,5,4,3),$
$(1,2,4,5,3),(1,2,4,3,5),(1,4,2,3,5),(1,4,2,5,3),$
$(1,4,5,2,3),(1,5,4,2,3),(5,1,4,2,3),(5,4,1,2,3),$
$(4,5,1,2,3),(4,1,5,2,3),(4,1,2,5,3),(4,1,2,3,5),\ldots\}$

例：巡回セールスマン問題 (許容集合をしっかりと書くと)

許容集合 $= \{$ (1, 2, 3, 4, 5), (1, 2, 3, 5, 4), (1, 2, 5, 3, 4), (1, 5, 2, 3, 4), (5, 1, 2, 3, 4), (5, 1, 2, 4, 3), (1, 5, 2, 4, 3), (1, 2, 5, 4, 3), (1, 2, 4, 5, 3), (1, 2, 4, 3, 5), (1, 4, 2, 3, 5), (1, 4, 2, 5, 3), (1, 4, 5, 2, 3), (1, 5, 4, 2, 3), (5, 1, 4, 2, 3), (5, 4, 1, 2, 3), (4, 5, 1, 2, 3), (4, 1, 5, 2, 3), (4, 1, 2, 5, 3), (4, 1, 2, 3, 5), (4, 1, 3, 2, 5), (4, 1, 3, 5, 2), (4, 1, 5, 3, 2), (4, 5, 1, 3, 2), (5, 4, 1, 3, 2), (5, 1, 4, 3, 2), (1, 5, 4, 3, 2), (1, 4, 5, 3, 2), (1, 4, 3, 5, 2), (1, 4, 3, 2, 5), (1, 3, 4, 2, 5), (1, 3, 4, 5, 2), (1, 3, 5, 4, 2), (1, 5, 3, 4, 2), (5, 1, 3, 4, 2), (5, 1, 3, 2, 4), (1, 5, 3, 2, 4), (1, 3, 5, 2, 4), (1, 3, 2, 5, 4), (1, 3, 2, 4, 5), (3, 1, 2, 4, 5), (3, 1, 2, 5, 4), (3, 1, 5, 2, 4), (3, 5, 1, 2, 4), (5, 3, 1, 2, 4), (5, 3, 1, 4, 2), (3, 5, 1, 4, 2), (3, 1, 5, 4, 2), (3, 1, 4, 5, 2), (3, 1, 4, 2, 5), (3, 4, 1, 2, 5), (3, 4, 1, 5, 2), (3, 4, 5, 1, 2), (3, 5, 4, 1, 2), (5, 3, 4, 1, 2), (5, 4, 3, 1, 2), (4, 5, 3, 1, 2), (4, 3, 5, 1, 2), (4, 3, 1, 5, 2), (4, 3, 1, 2, 5), (4, 3, 2, 1, 5), (4, 3, 2, 5, 1), (4, 3, 5, 2, 1), (4, 5, 3, 2, 1), (5, 4, 3, 2, 1), (5, 3, 4, 2, 1), (3, 5, 4, 2, 1), (3, 4, 5, 2, 1), (3, 4, 2, 5, 1), (3, 4, 2, 1, 5), (3, 2, 4, 1, 5), (3, 2, 4, 5, 1), (3, 2, 5, 4, 1), (3, 5, 2, 4, 1), (5, 3, 2, 4, 1), (5, 3, 2, 1, 4), (3, 5, 2, 1, 4), (3, 2, 5, 1, 4), (3, 2, 1, 5, 4), (3, 2, 1, 4, 5), (2, 3, 1, 4, 5), (2, 3, 1, 5, 4), (2, 3, 5, 1, 4), (2, 5, 3, 1, 4), (5, 2, 3, 1, 4), (5, 2, 3, 4, 1), (2, 5, 3, 4, 1), (2, 3, 5, 4, 1), (2, 3, 4, 5, 1), (2, 3, 4, 1, 5), (2, 4, 3, 1, 5), (2, 4, 3, 5, 1), (2, 4, 5, 3, 1), (2, 5, 4, 3, 1), (5, 2, 4, 3, 1), (5, 4, 2, 3, 1), (4, 5, 2, 3, 1), (4, 2, 5, 3, 1), (4, 2, 3, 5, 1), (4, 2, 3, 1, 5), (4, 2, 1, 3, 5), (4, 2, 1, 5, 3), (4, 2, 5, 1, 3), (4, 5, 2, 1, 3), (5, 4, 2, 1, 3), (5, 2, 4, 1, 3), (2, 5, 4, 1, 3), (2, 4, 5, 1, 3), (2, 4, 1, 5, 3), (2, 4, 1, 3, 5), (2, 1, 4, 3, 5), (2, 1, 4, 5, 3), (2, 1, 5, 4, 3), (2, 5, 1, 4, 3), (5, 2, 1, 4, 3), (5, 2, 1, 3, 4), (2, 5, 1, 3, 4), (2, 1, 5, 3, 4), (2, 1, 3, 5, 4), (2, 1, 3, 4, 5)$\}$

120 **個の巡回路**

巡回セールスマン問題：許容集合



	1	2	3	4	5
1	0	4	2	3	3
2	4	0	5	4	5
3	2	5	0	2	2
4	3	4	2	0	1
5	3	5	2	1	0

許容集合 $=\{$(1, 2, 3, 4, 5), (1, 2, 3, 5, 4), (1, 2, 5, 3, 4), (1, 5, 2, 3, 4), (5, 1, 2, 3, 4), (5, 1, 2, 4, 3), (1, 5, 2, 4, 3), (1, 2, 5, 4, 3), (1, 2, 4, 5, 3), (1, 2, 4, 3, 5), (1, 4, 2, 3, 5), (1, 4, 2, 5, 3), (1, 4, 5, 2, 3), (1, 5, 4, 2, 3), (5, 1, 4, 2, 3), (5, 4, 1, 2, 3), (4, 5, 1, 2, 3), (4, 1, 5, 2, 3), (4, 1, 2, 5, 3), (4, 1, 2, 3, 5), ...$\}$

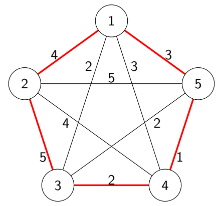
- **「同じ」巡回路を表している**
- 1 **で始まるものだけを集めれば，「同じ」巡回路が現れない**

例：巡回セールスマン問題 (1 で始まるものだけ書くと)

許容集合 $= \{$ (1, 2, 3, 4, 5), (1, 2, 3, 5, 4), (1, 2, 5, 3, 4), (1, 5, 2, 3, 4), (1, 5, 2, 4, 3), (1, 2, 5, 4, 3), (1, 2, 4, 5, 3), (1, 2, 4, 3, 5), (1, 4, 2, 3, 5), (1, 4, 2, 5, 3), (1, 4, 5, 2, 3), (1, 5, 4, 2, 3), (1, 5, 4, 3, 2), (1, 4, 5, 3, 2), (1, 4, 3, 5, 2), (1, 4, 3, 2, 5), (1, 3, 4, 2, 5), (1, 3, 4, 5, 2), (1, 3, 5, 4, 2), (1, 5, 3, 4, 2), (1, 5, 3, 2, 4), (1, 3, 5, 2, 4), (1, 3, 2, 5, 4), (1, 3, 2, 4, 5)$\}$

24 **個の巡回路**

巡回セールスマン問題：許容集合



	1	2	3	4	5
1	0	4	2	3	3
2	4	0	5	4	5
3	2	5	0	2	2
4	3	4	2	0	1
5	3	5	2	1	0

許容集合 $=\{$(1, 2, 3, 4, 5), (1, 2, 3, 5, 4), (1, 2, 5, 3, 4), (1, 5, 2, 3, 4), (1, 5, 2, 4, 3), (1, 2, 5, 4, 3), (1, 2, 4, 5, 3), (1, 2, 4, 3, 5), (1, 4, 2, 3, 5), (1, 4, 2, 5, 3), (1, 4, 5, 2, 3), (1, 5, 4, 2, 3), (1, 5, 4, 3, 2), (1, 4, 5, 3, 2), (1, 4, 3, 5, 2), (1, 4, 3, 2, 5), (1, 3, 4, 2, 5), (1, 3, 4, 5, 2), (1, 3, 5, 4, 2), (1, 5, 3, 4, 2), ...$\}$

- **「同じ」巡回路を表している**
- 2 **番目の都市の番号が最後の都市の番号より小さい巡回路だけ集めればよい**

例：巡回セールスマン問題 (1 で始まるものだけ書くと)

許容集合 $= \{$ (1, 2, 3, 4, 5), (1, 2, 3, 5, 4), (1, 2, 5, 3, 4), (1, 2, 5, 4, 3), (1, 2, 4, 5, 3), (1, 2, 4, 3, 5), (1, 4, 2, 3, 5), (1, 4, 3, 2, 5), (1, 3, 4, 2, 5), (1, 3, 5, 2, 4), (1, 3, 2, 5, 4), (1, 3, 2, 4, 5)$\}$

12 **個の巡回路**

教訓 2

「同じ解」が複数あるならば，その中の 1 つだけ残せばよい

解の表現：まとめ

教訓 1

「解の候補」として，最適解となる可能性のないものは考えなくてよい

教訓 2

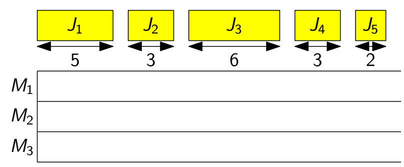
「同じ解」が複数あるならば，その中の 1 つだけ残せばよい

この 2 つの考えをもとにして，
考えなくてはならない許容解の数を大幅に減らすことができる

目次

最終完了時刻最小化スケジューリング問題

m **個の同一な機械** $M_1, \ldots, M_m$**，** n **個のジョブ** $J_1, \ldots, J_n$,
各ジョブ J_i **の処理時間** p_i



行うこと：すべてのジョブを機械で処理する

$$p_1 = 5,\ p_2 = 3,\ p_3 = 6,\ p_4 = 3,\ p_5 = 2$$

最終完了時刻最小化スケジューリング問題

決めること (スケジューリング，scheduling)

- 各ジョブをどの機械に割り当てるか？
- 各機械で，割り当てられたジョブをどの順に処理するか？

J_1	J_2	J_3	J_4	J_5
5	3	6	3	2

M_1	
M_2	
M_3	

ジョブの中断はないものとする (non-preemptive scheduling)

最終完了時刻最小化スケジューリング問題

目的：最終完了時刻の最小化

M_1	J_1	J_5	7
M_2	J_4	J_2	6
M_3	J_3		6

このスケジュールにおける最終完了時刻 $= 7$

最終完了時刻最小化スケジューリング問題

最終完了時刻最小化スケジューリング問題

入力
- m 個の同一な機械 $M_1, \ldots, M_m$
- n 個のジョブ $J_1, \ldots, J_n$
- 各ジョブ J_i の処理時間 p_i

許容解
- スケジュール (機械へのジョブの割当，処理順)

目的
- 最終完了時刻の最小化

M_1	J_1	J_5	7
M_2	J_4	J_2	6
M_3	J_3		6

ジョブの開始時刻

このようなスケジュールは考えるべきなのか？

M_1	J_5		J_1	8
M_2	J_4		J_2	9
M_3		J_3		7

考えなくてもよい

教訓 1

「解の候補」として，最適解となる可能性のないものは考えなくてよい

処理が終了したら，直ちに次のジョブを処理した方が
最終完了時刻は早くなる (より正確には，遅くならない)

スケジュールの表現

M_1	J_1	J_5	7
M_2	J_4	J_2	6
M_3	J_3		6

- 各ジョブをどの機械に割り当てるか？
  - M_1 に J_1, J_5 を割り当てる
  - M_2 に J_2, J_4 を割り当てる
  - M_3 に J_3 を割り当てる
- 各機械で，割り当てられたジョブをどの順に処理するか？
  - M_1 では J_1, J_5 の順に処理する
  - M_2 では J_4, J_2 の順に処理する
  - M_3 では J_3 の順に処理する

処理順を決めたら，遊休時間を持たせずにジョブを処理させる

最終完了時刻最小化スケジューリング問題の性質

注意

最終完了時刻最小化スケジューリング問題では
各機械での処理順が変わっても，最終完了時刻は変わらない

M_1	J_1	J_5	7
M_2	J_4	J_2	6
M_3	J_3		6

M_1	J_5	J_1	7
M_2	J_4	J_2	6
M_3	J_3		6

つまり

処理順に関する情報は，最適解を定めるために必要ない

教訓 2

「同じ解」が複数あるならば，その中の 1 つだけ残せばよい

最終完了時刻最小化スケジューリング問題：解の表現

解の表現にはジョブの割当だけで十分

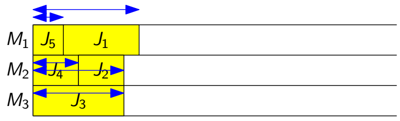
ジョブが 5 つ，機械が 2 台の場合

M_1: J_1, J_2, J_3, J_4, J_5 M_2:	M_1: J_1, J_2, J_4 M_2: J_3, J_5	M_1: J_2, J_4, J_5 M_2: J_1, J_3	M_1: J_2, J_4 M_2: J_1, J_3, J_5	M_1: J_3 M_2: J_1, J_2, J_4, J_5
M_1: J_1, J_2, J_3, J_4 M_2: J_5	M_1: J_1, J_2, J_5 M_2: J_3, J_4	M_1: J_3, J_4, J_5 M_2: J_1, J_2	M_1: J_2, J_5 M_2: J_1, J_3, J_4	M_1: J_4 M_2: J_1, J_2, J_3, J_5
M_1: J_1, J_2, J_3, J_5 M_2: J_4	M_1: J_1, J_3, J_4 M_2: J_2, J_5	M_1: J_1, J_2 M_2: J_3, J_4, J_5	M_1: J_3, J_4 M_2: J_1, J_2, J_5	M_1: J_5 M_2: J_1, J_2, J_3, J_4
M_1: J_1, J_2, J_4, J_5 M_2: J_3	M_1: J_1, J_3, J_5 M_2: J_2, J_4	M_1: J_1, J_3 M_2: J_2, J_4, J_5	M_1: J_4, J_5 M_2: J_1, J_2, J_3	M_1: M_2: J_1, J_2, J_3, J_4, J_5
M_1: J_1, J_3, J_4, J_5 M_2: J_2	M_1: J_1, J_4, J_5 M_2: J_2, J_3	M_1: J_1, J_4 M_2: J_2, J_3, J_5	M_1: J_3, J_5 M_2: J_1, J_2, J_4	
M_1: J_2, J_3, J_4, J_5 M_2: J_1	M_1: J_2, J_3, J_4 M_2: J_1, J_5	M_1: J_1, J_5 M_2: J_2, J_3, J_4	M_1: J_1 M_2: J_2, J_3, J_4, J_5	
M_1: J_1, J_2, J_3 M_2: J_4, J_5	M_1: J_2, J_3, J_5 M_2: J_1, J_4	M_1: J_2, J_3 M_2: J_1, J_4, J_5	M_1: J_2 M_2: J_1, J_3, J_4, J_5	

目次

完了時刻和最小化スケジューリング問題

各ジョブに着目して，ジョブの完了時刻の和を最小化



このスケジュールにおける完了時刻和 $= 2 + 7 + 3 + 6 + 6 = 24$

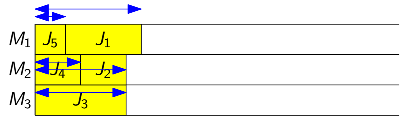
完了時刻和最小化スケジューリング問題

完了時刻和最小化スケジューリング問題

入力
- m 個の同一な機械 $M_1, \ldots, M_m$
- n 個のジョブ $J_1, \ldots, J_n$
- 各ジョブ J_i の処理時間 p_i

許容解
- スケジュール (機械へのジョブの割当，処理順)

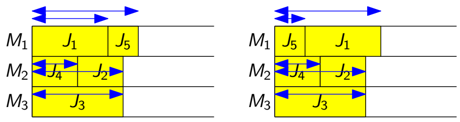
目的
- 完了時刻和の最小化



完了時刻和最小化スケジューリング問題の性質

注意

完了時刻和最小化スケジューリング問題では
各機械での処理順が変わると，完了時刻和は変わる (かもしれない)



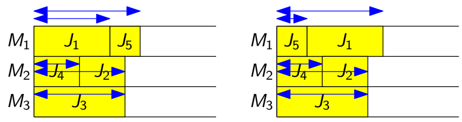
つまり

処理順に関する情報は，最適解を定めるために必要である！

完了時刻和最小化スケジューリング問題の性質：再考

注意

完了時刻和最小化スケジューリング問題では
各機械での処理順が変わると，完了時刻和は変わる (かもしれない)



つまり

処理順に関する情報は，最適解を定めるために必要である！ ???

割当を決めたとき，最適な処理順がすぐに分かればよい

教訓 1

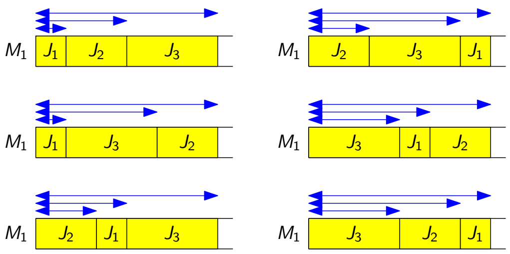
「解の候補」として，最適解となる可能性のないものは考えなくてよい

機械が 1 台のとき，最適な処理順は？

疑問

機械が 1 台のとき，最適な処理順はすぐに分かるだろうか？

ジョブが 3 つのときの例



処理時間が短い順に処理をすると最適になりそう？

機械が 1 台のとき，最適な処理順は？

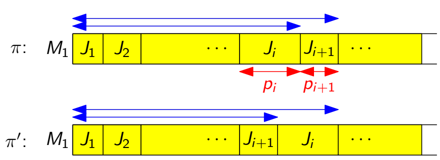
定理 (Smith '56)

機械が 1 台のとき，完了時刻和最小化スケジューリング問題において
処理時間の短いジョブから順に処理するスケジュールは最適

証明：背理法による証明
- 処理時間の短いジョブから順に処理するスケジュールが最適ではないと仮定
- π を最適スケジュールとする
- π では処理時間の短いジョブから順に処理しない
- ……
- これは ○○ に矛盾 □

定理の証明 (1)

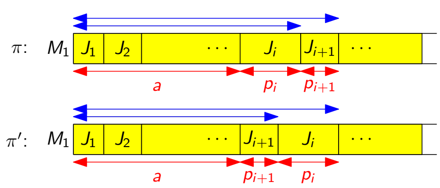
- そのスケジュール π ではジョブが $J_1, J_2, \ldots, J_n$ の順に処理されるとする
- π では処理時間の短いジョブから順に処理をしないので，ある i が存在して，$p_i > p_{i+1}$ となる (演習問題)
- ここで，J_i と J_{i+1} を入れ替えたスケジュール π' を考える



定理の証明 (2)

- π の完了時刻和と π' の完了時刻和の違いは，J_i と J_{i+1} の完了時刻和のみ
- π における J_i の完了時刻 $= a + p_i$ とすると

$$\begin{aligned} \pi \text{ における } J_{i+1} \text{ の完了時刻} &= a + p_i + p_{i+1}, \\ \pi' \text{ における } J_{i+1} \text{ の完了時刻} &= a + p_{i+1}, \\ \pi' \text{ における } J_i \text{ の完了時刻} &= a + p_{i+1} + p_i \end{aligned}$$

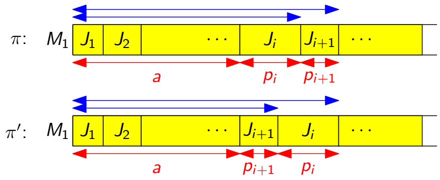


定理の証明 (3)

- よって，

$$\pi \text{ の完了時刻和} - \pi' \text{ の完了時刻和} = p_i - p_{i+1} > 0$$

- つまり，π' の完了時刻和 $< \pi$ の完了時刻和
- これは，π が最適スケジュールであったことに矛盾 □



定理の帰結

定理 (Smith '56)

機械が 1 台のとき，完了時刻和最小化スケジューリング問題において処理時間の短いジョブから順に処理するスケジュールは最適

補足

- この処理順を「処理時間順」(shortest processing time order) と呼ぶことがある
- 提唱者の名前から「Smith の規則」(Smith's rule) と呼ぶこともある
- 計算量は並べ替え (ソート) が支配し，$O(n \log n)$ 時間

帰結

- 解の表現として割当だけを考えればよい
- 割当から各機械における最適処理順は (定理から) 直ちに分かる

目次

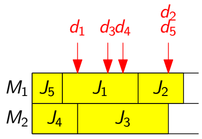
最大遅延最小化スケジューリング問題

最大遅延最小化スケジューリング問題

入力
- m 個の同一な機械 $M_1, \ldots, M_m$
- n 個のジョブ $J_1, \ldots, J_n$
- 各ジョブ J_i の処理時間 p_i
- 各ジョブ J_i の納期 d_i (deadline)

許容解
- スケジュール

目的
- 最大遅延の最小化



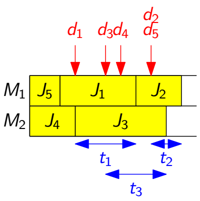
遅延の定義

ジョブ J_i の遅延

スケジュールにおけるジョブ J_i の遅延 (tardiness) とは

$$t_i = \max\{0, J_i \text{ の完了時刻} - d_i\}$$

最大遅延最小化スケジューリング問題は $\max\{t_1, t_2, \ldots, t_n\}$ を最小化

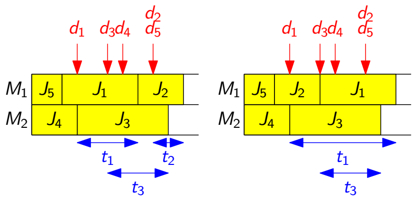


$t_1 = 4, t_2 = 2, t_3 = 4, t_4 = 0, t_5 = 0, \max\{t_1, t_2, t_3, t_4, t_5\} = 4$

最大遅延最小化スケジューリング問題の性質

注意

最大遅延最小化スケジューリング問題では
各機械での処理順が変わると，最大遅延は変わる (かもしれない)



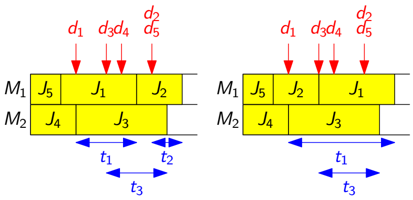
つまり

処理順に関する情報は，最適解を定めるために必要である！

最大遅延最小化スケジューリング問題の性質：再考

注意

最大遅延最小化スケジューリング問題では
各機械での処理順が変わると，最大遅延は変わる (かもしれない)



つまり

処理順に関する情報は，最適解を定めるために必要である！ ???

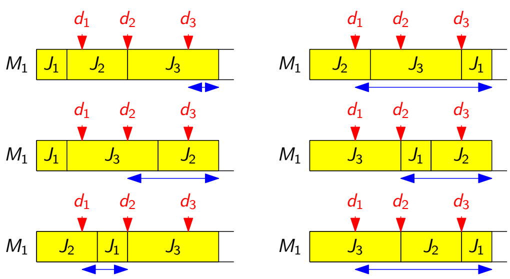
割当を決めたとき，最適な処理順がすぐに分かればよい

機械が 1 台のとき，最適な処理順は？

疑問

機械が 1 台のとき，最適な処理順はすぐに分かるだろうか？

ジョブが 3 つのときの例



納期の早い順に処理をすると最適になりそう？

機械が 1 台のとき，最適な処理順は？

定理

機械が 1 台のとき，最大遅延最小化スケジューリング問題において
納期の早いジョブから順に処理するスケジュールは最適

証明：演習問題

補足

- この処理順を「納期優先順」(earliest deadline first order) と呼ぶことがある
- 計算量は並べ替え (ソート) が支配し，$O(n \log n)$ 時間

帰結

- 解の表現として割当だけを考えればよい
- 割当から各機械における最適処理順は (定理から) 直ちに分かる

目次

今日のまとめ

解の表現

解の表現を工夫することで
考えなくてはならない許容解の数を大幅に減らすことができる

教訓 1

「解の候補」として，最適解となる可能性のないものは考えなくてよい

教訓 2

「同じ解」が複数あるならば，その中の 1 つだけ残せばよい

局所探索法との関係

- 局所探索法も，工夫された解の表現を使って考えればよい
- 探索の範囲が狭くなるので，効率が上がる (かもしれない)

残った時間の使い方

- 演習問題をやる
  - 相談推奨 (ひとりでやらない)
- 質問をする
  - 教員は巡回
- 退室時，小さな紙に感想など書いて提出する ← 重要
  - 内容は何でも OK
  - 匿名で OK