第4回 2026/5/1

2026年4月9日　13:30

greedy ← Jack Edmonds

貪欲アルゴリズム，貪欲戦略

- 区間スケジューリング

$J=\{1,2,\dots,n\}$　n件の仕事(jobs)

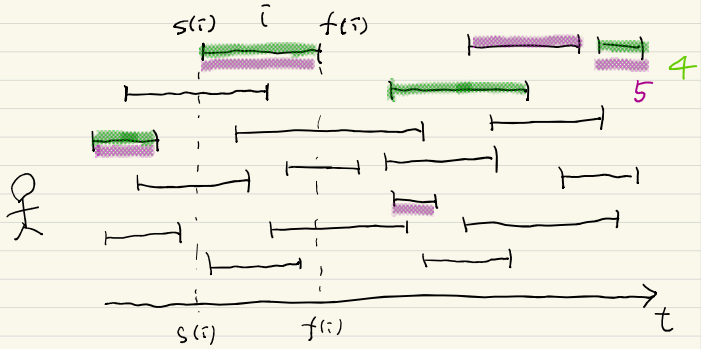
各ジョブ $j\in J$ に対し，開始時刻 $s(j)$

終了時刻 $f(j)$

が定められている．

ある人は ジョブにとりくむ，同時に1つのジョブしかできない

Problem 最大件数のジョブを実行するには？



アルゴリズム　実際に行うジョブの集合

$R:=J$, $A=\phi$　← 現時点で実行可能なジョブの集合

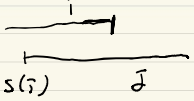
$R\neq\phi$ でないでかぎり (while $R\neq\phi$)

以下をくりかえす

- Choose $i\in R$ such that $f(i)$ is minimum

  $A\leftarrow A\cup\{i\}$　← iを実行すると できなくなるjob

- Remove from R all $j\in R$ with $s(j)<f(i)$



Prop このアルゴリズムは最大件数の実行可能なジョブを output (Aとよぶ)

☺ $A=\{i_1,i_2,\dots,i_k\}$ アルゴリズムの output

$A^*=\{j_1,j_2,\dots,j_r\}$ $(k\le r)$ 実行可能な最大件数のジョブ

並べかえて

$A=$ 

$A^*=$ 

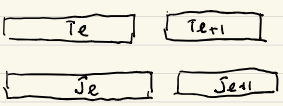
となっているとしてよい

Claim $f(i_\ell)\le f(j_\ell)$ $(\ell=1,2,\dots,k)$

☺(帰納法) アルゴリズムのやり方より

$f(i_1)\le f(j_1)$ (ベースケース)

$f(i_\ell)\le f(j_\ell)$ とする(帰納法の仮定)

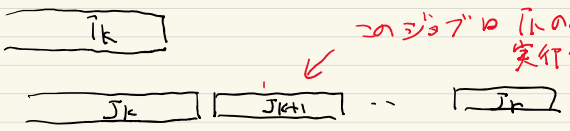


$j_{\ell+1}$ はアルゴリズムのはなびを $\ell+1$ ステップ時で実行可能

アルゴリズムは $f(j_{\ell+1})$ より おそくない終了時刻のジョブ $i_{\ell+1}$ が えらばれる

$\rightsquigarrow f(i_{\ell+1})\le f(j_{\ell+1})$

$\ell=k$ のときを考える．もしも $r>k$ なら



$f(i_k)\le f(j_k)$

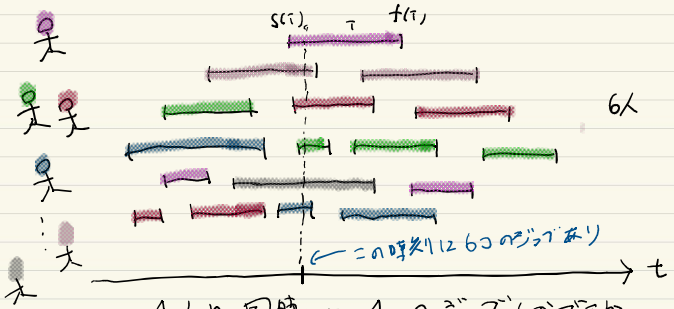
$\rightsquigarrow$ ムジュン (終了時 $\ell=k$ で実行可能なジョブはないはず)

□

区間分割

$J = \{1, 2, \cdots, n\}$ n件仕事

各 $j \in J$ に 開始時刻 $s(j)$, 終了時刻 $f(j)$

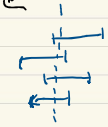


1人は 同時には 1つのジョブしかできない

Problem すべての仕事(ジョブ)を処理できる最小人数のやり方は？

d: 同時に処理するジョブ数の最大値

互いに 交わる区間の最大数



⇝ 少なくとも d人 必要

最小人数 $\geq d$

アルゴリズム

d人の作業者をよんできた

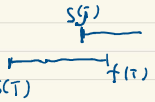
$L = \{1, 2, \cdots, d\}$

開始時刻でジョブを並べかえる

- $s(1) \leq s(2) \leq \cdots \leq s(n)$
- For $j = 1, 2, \cdots, n$; do
  - $U_j = L \setminus \{l_i \mid i < j, f(i) > s(j)\}$

l_iは従事している作業者$\in L$

すでに割り当てられている人で jを実行できない



- Choose any member l_j in U_j
  jの担当者を l_j

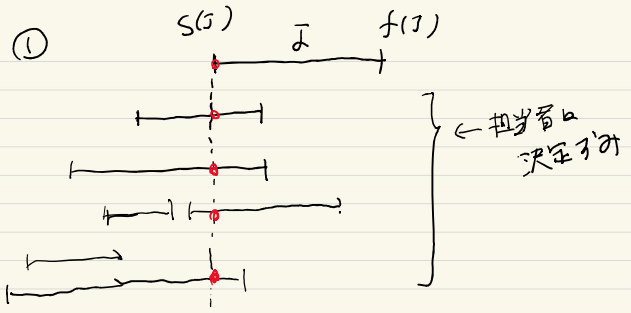
(?) $U_j = \phi$ はないのか？

1人が 同時に2つ以上 処理することはないのか？

Prop アルゴリズムは正しい

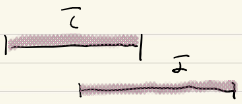
(::) ① U_j は空でない

② 担当者は 同時に2つ以上の仕事はしない

① 

- 高々 $d-1$ 個 $s(j)$と交わるジョブで $s(j)$の時点で仕事をしている人 高々 $d-1$人

⇝ U_jには 少なくとも フリーの人がいる

② 

となってしまったら ・・・

開始時刻をヒクク・・

jの担当者を 決めるてまえね

iの担当者は 決まっている

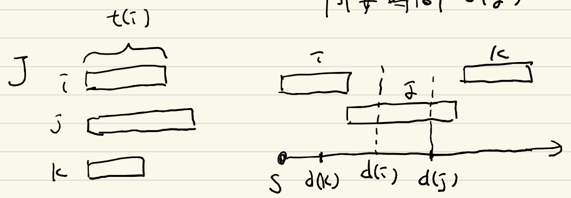
U_jの決め方から jは 担当者の候補からのぞかれているはずである

⇝ ムジュン ■

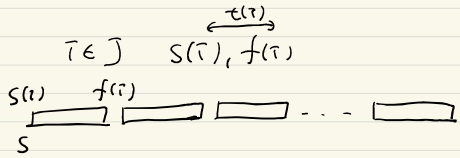
最大遅れ最小スケジューリング

$J=\{1,2,\dots,n\}$　　s：開始時刻

各 $j\in J$ に対して　deadline $d(j)$
納期
所要時間 $t(j)$



スケジュール　$i\in J$　$s(i), f(i)$



遅れ　$l(i) := \max\{0, f(i)-d(i)\}$
deadlineをどれだけ破ったか

最大遅れ　$L := \max_{i\in J} l(i)$

Problem　最大遅れを最小にするスケジュールを求めよ.

アルゴリズム

・deadlineの順にならべかえる

$d(1) \le d(2) \le \cdots \le d(n)$　　(<　<　<)

・$f := s$　　(現時刻をあらわす)

・For $i=1,2,\dots,n$ do

　$s(i) := f$

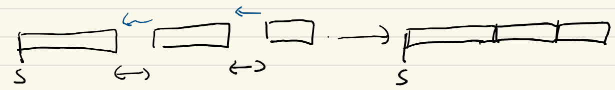
　$f \leftarrow f + t(i)$　　⟷ $i\in J$

　$f(i) := f$　　$[s(i), f(i)]$ で処理する.

Prop　アルゴリズムの定めるスケジュールが L を最小化する。　　← 反転がないスケジュールになってる

unknown
最適スケジュールを1つ考える

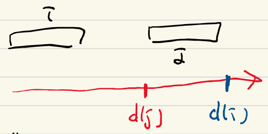
Obs　アイドル時間はないとしてよい



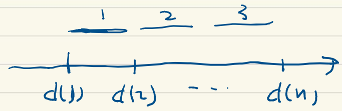
反転 ⟺ (i,j)

・i が先にスケジュール

・$d(i)>d(j)$：納期は j が先.



アルゴリズムで得られるスケジュールには反転なし



元のスケジュールでの

i の納期おくれ　$f(i)-d_i$
$= s'+t_i-d_i$

j の納期おくれ　$s'+t_i+t_j-d_j$

新スケジュール：i のおくれ
$= \bar{f}(i)-d_i = s'+t_i+t_j-d_i$

j のおくれ $= \bar{f}(j)-d_j = s'+t_j-d_j$

2024/5/18

Obs　もしもスケジュールに反転があれば
「となりあう」反転がある

(i,j) となりあってない反転
$d_j < d_i$

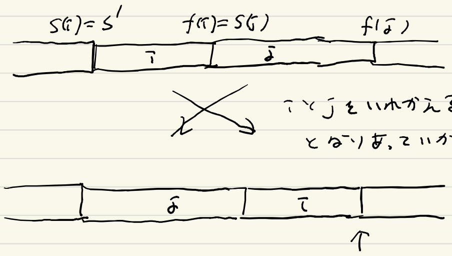
k の納期は？

もし $d_k < d_j$ → $d_k < d_j < d_i$
→ (i,k) も反転
→ j より前にある

もし $d_k > d_j$ → (k,j) がとなりあう反転

(i,k) に対し同じことをくり返すと となりあう反転がみつかる ∎

(i,j)：となり合う反転　$d_i > d_j$



$\bar{f}(i) - d_i = s' + t_i + t_j - d_i < s' + t_i + t_j - d_j \quad (d_j < d_i)$

$= f(j) - d_j$

$\bar{f}(j) - d_j = s' + t_j - d_j < s' + t_i + t_j - d_j = f(j) - d_j$

$\max\{\bar{f}(i) - d_i, \bar{f}(j) - d_j\} < f(j) - d_j$

→ i と j のいれかえで最大納期おくれを悪くならない

→ となりあう反転を解消していくと これを悪くならない

→ 反転がない optimal スケジュールが存在する

⟺ アルゴリズムで得られるスケジュール ■