

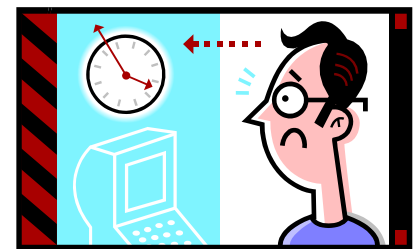


JSSPとは？

電子情報工学科
伊庭 斉志

どういう順で仕事をするか？

- Jobの割付の優先規則
 - コンフリクトの起こったときの解消方法
 - ディスパッチング・ルール
- レポートの作成順
 - SPT: Shortest processing time
 - EDD: Earliest due date
 - SLACK
 - FIFO
 - RANDOM
 - ... ect.



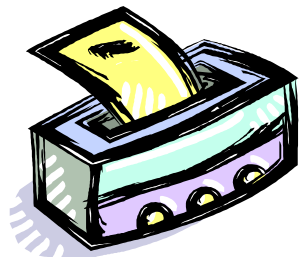


実用的なディスパッチングルール

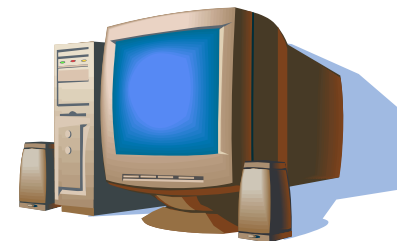
- **SPT (shortest processing time)**: 処理時間が最小の仕事を選ぶ。平均納期遅れを小さくするが、納期遅れの大きな仕事も出る。
- **EDD (earliest due date)**: 納期に最も近い仕事を選ぶ。納期遅れに関する評価尺度に対して有効。
- **SLACK (SLACK time)**: **スラック時間 (= 納期 - 現在時刻 - 総残り加工時間)** が最小の仕事を選ぶ。納期遅れに関する評価尺度に対して有効。
- **FIFO (first in first out)**: 中立的なルール。RANDOMと同様の性格だが、RANDOMよりも結果にばらつきがない。

スケジューリングの例

3つのタスクA,B,Cを2種のハードウェア M_1, M_2 で処理するのにどのような順で遂行すればよいか?



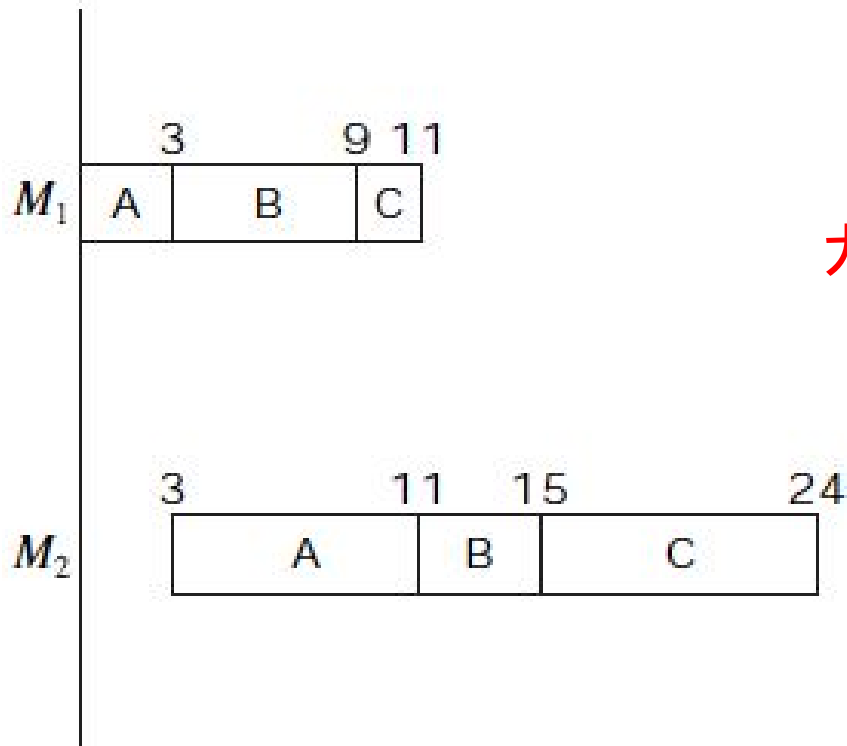
hard タスク	M_1	M_2
A	3	8
B	6	4
C	2	9



スケジューリングの例

	M_1	M_2
A	3	8
B	6	4
C	2	9

例: $A \rightarrow B \rightarrow C$



ガントチャート

ジョンソンの規準

- 所要時間の表の中の最小の値をマークする。
- これがもし M_1 側であれば、その処理を最初に持ってくる。
- もし M_2 側ならばラストにまわす。
- これをすべてのタスクが終わるまで繰り返す。

	M_1	M_2
A	3	8
B	6	4
C	2	9

$C \rightarrow A \rightarrow B$

ジョンソンの規準

タスクの数が一般に3種類以上になると、特殊な場合を除いては、ジョンソンのような見事な規準は見あたらない。



ジョブショップ・スケジューリング

- JSSP (Job Shop Scheduling Problem)

- 一般のJSSP

- NP完全問題

機械の順は任意

	1	2	3	4
J1	M1,3	M2,2	M3,4	M4,1
J2	M2,2	M4,2	M3,1	M1,3
J3	M1,2	M3,2	M2,1	M4,2

GYTPE:JOBの割付アルゴリズム

割付けの例

GTYPE: 1 2 2 2 2 3 ...

	1	2	3	4
J1	M1,3	M2,2	M3,4	M4,1
J2	M2,2	M4,2	M3,1	M1,3
J3	M1,2	M3,2	M2,1	M4,2

機械 割付スケジュール

1 2 3 4 5 6 7 8

M1: J1 J1 J1 J3 J3 J2 J2 J2

M2: J2 J2

M3: J2

M4: J2 J2

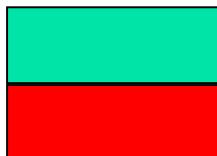
GYTPE: 交叉の工夫

親1の染色体:



親2の染色体:



の部分を特別に処理して
を構成する

子1の染色体:



子2の染色体:





GYTPE: 交叉の工夫

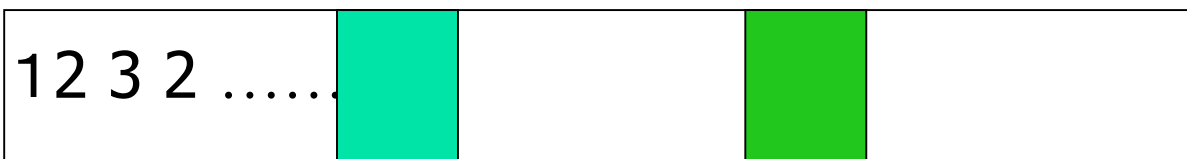
Step1: 同一遺伝子に印(*)を付ける

親1: 1 7 3 3 1 5 4

親2: 2 7 3 1 4 7 6

GYTPE: 突然変異操作

親の染色体:



遺伝子を交換する

子の染色体:

