

2004年度磐田南高校キャンパスゼミ

- 巧みなスケジュールの立て方
- 1つのプロジェクトを完成するには多くの作業が必要
- 多くの作業を段取り良くこなすためにスケジュールを立てよう。
- PERT, CPMを学ぶ
- 点と矢印で作業間の前後関係を表現しよう。
- 数学モデルをコンピュータで解こう

日程計画

仕事の段取りと進捗状況の管理

PERT Program Evaluation and Review Technique

大規模な仕事、プロジェクトに対して、
 1. 効率良い作業計画の決定
 2. 作業の進捗状況の管理
 => PERTの導入
 期間の短縮 コスト削減
 状況の容易な把握 リスク削減

プロジェクト完了日数をできるだけ早くなるように作業間の順序関係を守って各作業開始日程を決定

プロジェクトは複数の作業から構成。
 各作業には所要日数が所与。
 作業間には順序関係が存在。

プロジェクトの計画, 管理

企業内の情報システム構築でのトラブル

- 1: 納期遅れ (指定期日までにシステムが完成しない)
- 2: コスト超過 (予算オーバー)
- 3: 機能/品質不備

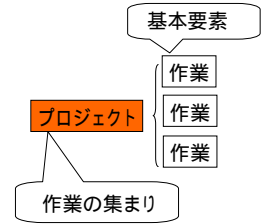
(日経システム構築 2003年8月)

プロジェクト全体を見通すことが大切
 プロジェクトの進捗管理 =>
 日程計画を立てること!
 柔軟に変更可能な日程計画を構築する管理技法
 スケジュールリング: PERT

例その1: 昼飯でカップラーメン準備

カップラーメン準備

- カップラーメン購入
- お湯を沸かす
- お湯を注ぎ、待つ



例: カップラーメン準備作業の順序

作業名: 作業内容

プロジェクト: カップラーメン準備

作業A: カップラーメン購入

作業B: お湯を沸かす

作業C: お湯を注ぎ、待つ

作業A: カップラーメン購入

作業B: お湯を沸かす

作業C: お湯を注ぎ、待つ



作業Aは作業Cの先行作業

作業Cは作業Aの後続作業

例: カップラーメン準備作業の時間など

プロジェクト: カップラーメン準備

作業A: カップラーメン購入
作業時間 5分

作業B: お湯を沸かす
作業時間 10分

作業C: お湯を注ぎ、待つ
作業時間 4分

各作業に関する情報: 作業名, 作業内容, 先行作業, 作業時間 + 費用, 必要人数など

例:カップラーメン準備作業一覧表

プロジェクト — カップラーメン準備

作業A:カップラーメン購入
作業時間 5分

作業B:お湯を沸かす
作業時間 10分

作業C:お湯を注ぎ、待つ
作業時間 4分

作業リスト

作業名	作業内容	作業時間	先行作業
A	購入	5	なし
B	お湯沸かす	10	なし
C	お湯注ぎ、待つ	4	A,B

例:カップラーメン準備の実施計画

今は12時

知りたい情報、把握したい情報

- ・いつ食べられる？
- ・「お湯注ぎ、待つ」作業はいつまでに終了？
- ・「お湯注ぎ、待つ」作業はいつから開始可能？
- ・各作業には「ゆとり」はある？

情報を手に入れるには？

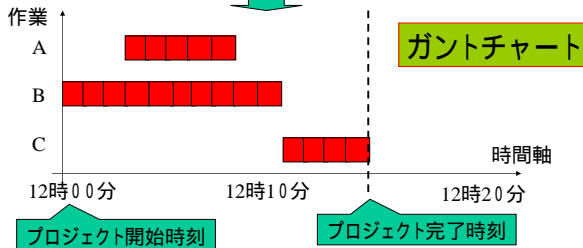
- ・「購入」作業はいつ開始？終了は？
- ・「お湯沸かす」作業はいつ開始？終了は？
- ・「お湯注ぎ、待つ」作業はいつ開始？終了は？

スケジュールの作成 適切な計画立案が大切

カップラーメン準備のスケジュール例

今は12時

作業名	作業内容	開始時刻	終了時刻
A	購入	12時3分	12時8分
B	お湯沸かす	12時00分	12時10分
C	お湯注ぎ、待つ	12時10分	12時14分



例: 計算機システムの導入

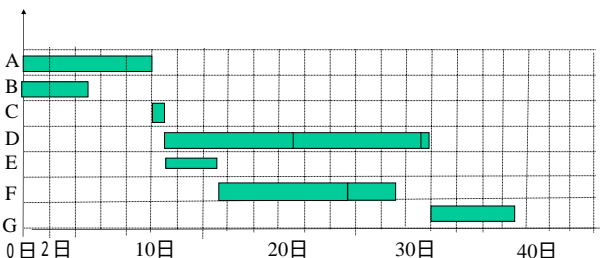
システム工学科では教育用計算機システムの導入することになりました。

表1 計算機システム導入に関する作業リスト

作業名	作業内容	作業日数	先行作業
A	仕様作成	10	なし
B	予算確保	5	なし
C	OS選択	1	A, B
D	アプリケーションソフト選択	21	C
E	機種選定	4	C
F	計算機据え付け	14	E
G	インストール	7	D, F

システム導入は最早で何日間で完了か？
各作業はいつ開始できるのか？

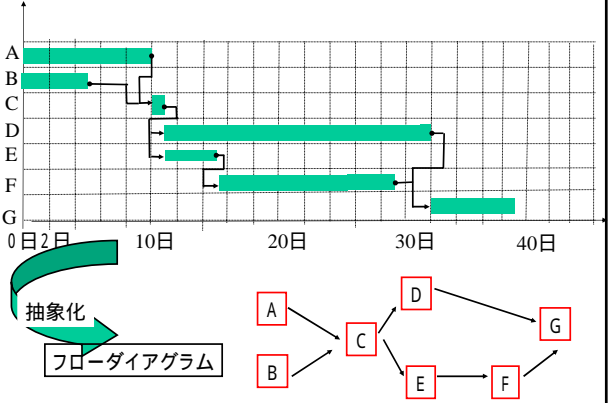
例2: スケジュール



ガントチャートの利点:
各作業の日程が見やすい。=> 日程の説明, 作業指示向き

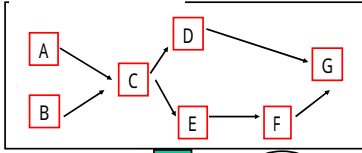
ガントチャートの欠点:
先行関係がわからない。作業のゆとりがわかりにくい
日程変更に対応しにくい。=> 計画立案には不向き

ガントチャートからフローダイアグラム

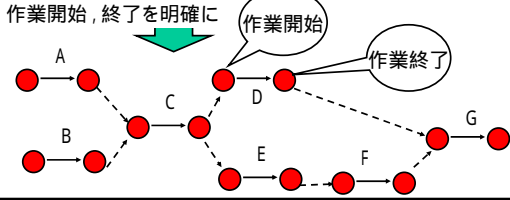


フローダイアグラムの欠点

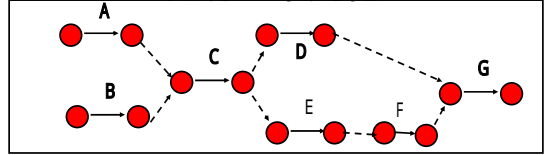
フローダイアグラム



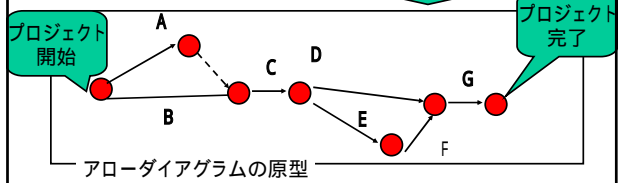
作業開始時点
作業終了時点
が表現されていない。



表現の単純化



先行作業の終了 = 後続作業の開始



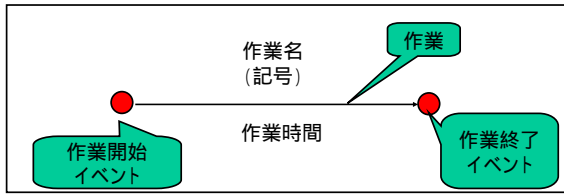
アローダイアグラム

- ・ 作業は矢線 ()
- ・ イベント(作業開始, 終了)は点 ()

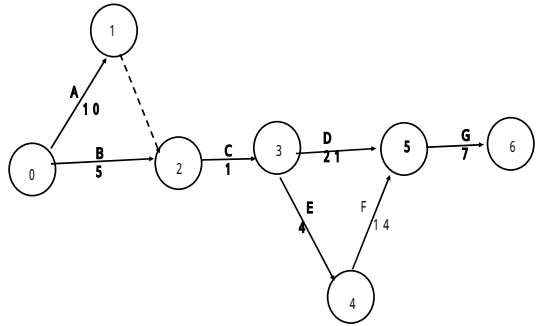
結合点

を用いてプロジェクト全体を表現したネットワーク

作業間の先行関係は矢線の前後関係

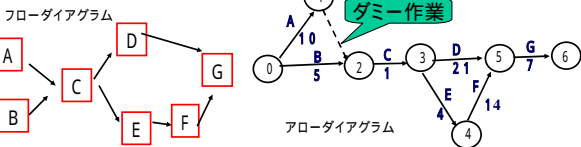
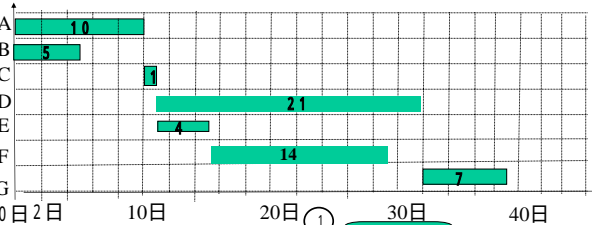


アローダイアグラムの例



ガントチャート, フローダイアグラムとアローダイアグラム

ガントチャート



スケジュール立案までのプロセス

作業リスト作成

アローダイアグラムの作成

最早結合点開始時刻
最遅結合点開始時刻の算出

線形計画問題へのモデル化

各作業の開始時刻, 終了時刻
プロジェクト完了時刻
各作業のゆとり

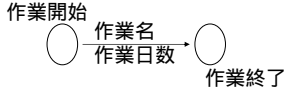
ガントチャートによる明示

アローダイアグラム

作業を示す矢線(矢印付き線, 枝)(→)
 イベント(作業開始, 終了)を表す点, ノード()

結合点とも呼ぶ

を用いて表現したネットワーク図.



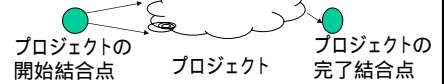
アローダイアグラムの約束

- プロジェクトは開始を示す1つノードと完了を示す1つノードを持つ.
- 作業の先行関係は守る. 平行な作業は禁止. 必要あればダミー作業 (作業時間は0)
- ノードの番号付けは, 矢印順に.

アローダイアグラム

アローダイアグラムの約束

- アローダイアグラムはプロジェクト全体の開始を示す1つ点(結合点)と完了を示す1つ点(結合点)を持つ.

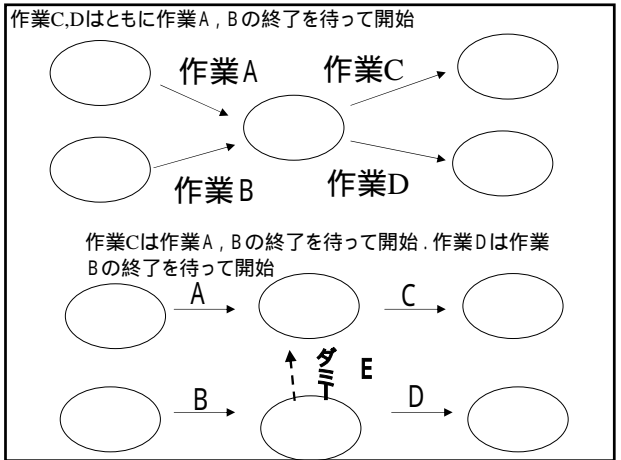
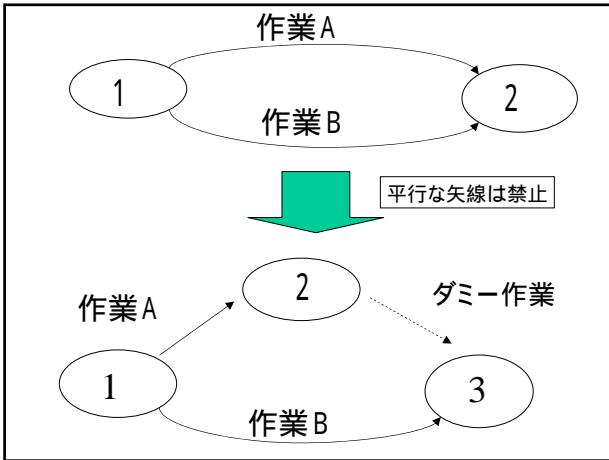


- 作業の先行関係は守る. 平行な矢線は禁止. 必要あればダミー作業



- 点(結合点)の番号が付く. 番号付けは, 矢印順に.

$$X1 < X2 < X3$$

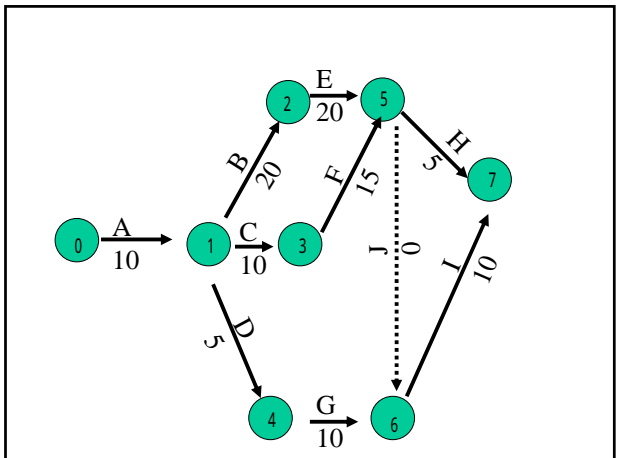


演習

作業リスト

作業	先行作業	作業時間
A	なし	10
B	A	20
C	A	10
D	A	5
E	B	20
F	C	15
G	D	10
H	E, F	5
I	E, F, G	10

アローダイアグラムを作成すること



演習：作業リストからアローダイアグラム

作業リスト1

作業	先行作業
A	なし
B	なし
C	A,B
D	B
E	C,D

作業リスト2

作業	先行作業
A	なし
B	A
C	B
D	B
E	C,D
F	D

作業リスト3

作業	先行作業
A	なし
B	なし
C	A,B
D	B
E	C
F	D
G	E,F

上記3つの作業リストに対応するアローダイアグラムを作成すること

アローダイアグラムから得られる情報

最早結合点開始時刻: その結合点(ノード)から出る全作業を最も早く開始できる時刻。

最遅結合点開始時刻: その結合点(ノード)から出る全作業の開始時刻をできるだけ遅くした時刻。

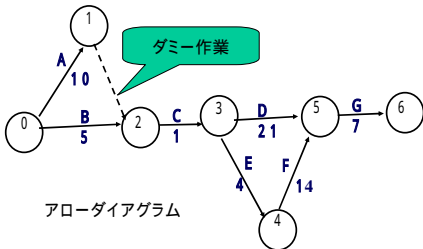
作業のゆとり: 全余裕(トータルフロート), 自由余裕(フリーフロート)

クリティカルパス: プロジェクト完了日数を達成する上で遅れてはならない作業。

各作業の開始時刻: 最早開始時刻とプロジェクト完了日数を遅らせないで最大限に遅く開始する時刻(最遅開始時刻)

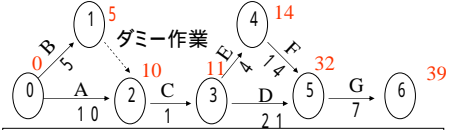
各作業の終了時刻: 最早終了時刻とプロジェクト完了日数を遅らせないで最大限に遅くした終了する時刻(最遅終了時刻)

アローダイアグラム



作業Cは作業A,Bが終了後、作業開始可能。

最早結合点開始時刻: その結合点(ノード)から出る全作業を最も早く開始できる時刻。ただし、結合点0の最早開始結合点時刻は0とする



結合点1から出るダミー作業が最も早く開始できる時刻(結合点1の**最早結合点開始時刻**)は5。

結合点2から出る作業Cが最も早く開始できる時刻(結合点2の**最早結合点開始時刻**)は10。

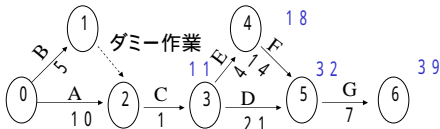
結合点3から出る作業D,Eが最も早く開始できる時刻(結合点3の**最早結合点開始時刻**)は11。

結合点4から出る作業Fが最も早く開始できる時刻(結合点4の**最早結合点開始時刻**)は15。

結合点5から出る作業Gが最も早く開始できる時刻(結合点5の**最早結合点開始時刻**)は32。

結合点6は辿り着いた作業Gの終了時刻 = $32 + 7 = 39$

最遅結合点開始時刻: プロジェクト完成時刻を遅らせることなく、その結合点から出る各作業の開始時刻をできるだけ遅くした時刻。結合点6のそれはプロジェクト完了時刻とする



結合点6の**最遅結合点開始時刻**は39

結合点5から出る作業Gが最も遅く開始できる時刻(結合点5の**最遅結合点開始時刻**)は32。

結合点4から出る作業Fが最も遅く開始できる時刻(結合点4の**最遅結合点開始時刻**)は18。

結合点3から出る作業D,Eが最も遅く開始できる時刻(結合点3の**最遅結合点開始時刻**)は11。

結合点2, 1, 0それぞれの**最遅結合点開始時刻**は?

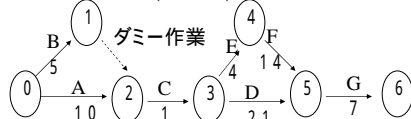
作業のゆとり: 全余裕 トータルフロート

その作業の終了を示す結合点の**最遅結合点開始時刻**

- その作業の**所要時間**

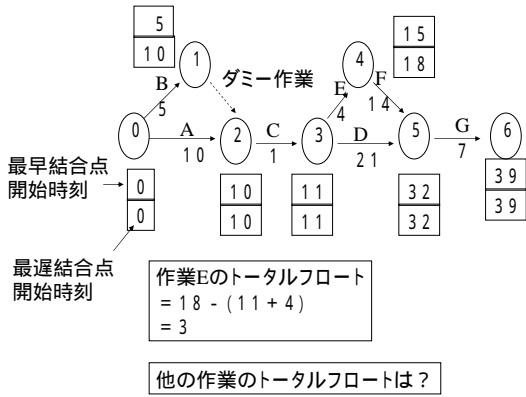
+ その作業の開始を示す結合点の**最早結合点開始時刻**

= トータルフロート(全余裕)

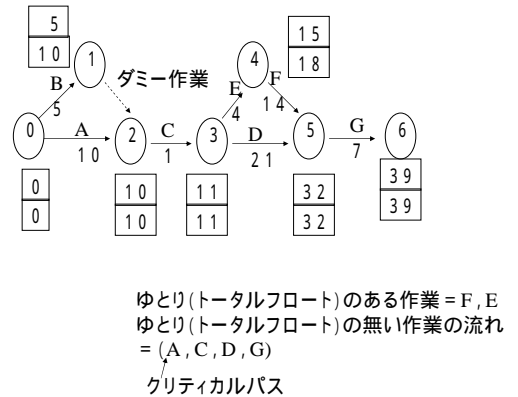


作業Eのトータルフロートとはプロジェクト完成時刻を遅らせることなく他の作業時刻を最大限調整して得られる作業Eのゆとり

各作業のゆとり(トータルフロート)を測定

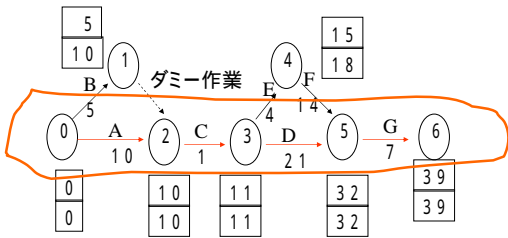


各作業のゆとり(トータルフロート)を測定



クリティカルパス:

プロジェクト完了日数を達成する上で遅れてはならない作業群.

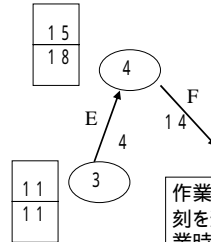


作業のゆとり:自由余裕(フリーフロート)

その作業の終了を示す結合点の最早結合点開始時刻

その作業の所要時間

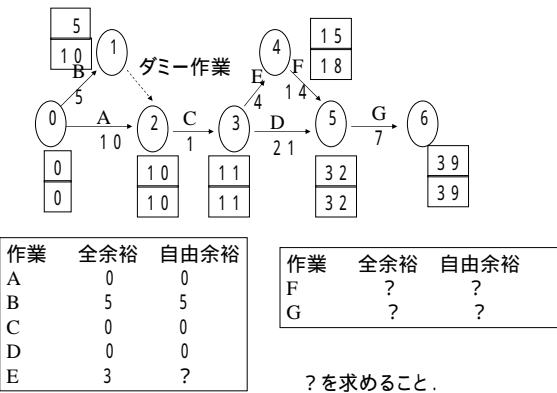
その作業の開始を示す結合点の最早結合点開始時刻



全余裕(トータルフロート) = $18 - (11 + 4) = 3$
自由余裕(フリーフロート) = $15 - (11 + 4) = 0$

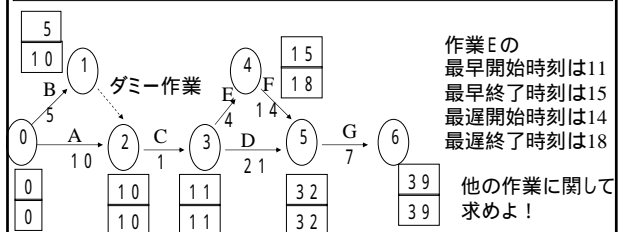
作業Eのフリーフロートとはプロジェクト完成時刻を遅らせることなく、さらに後続の最早開始作業時刻は遅らせないことで生じる作業Eのゆとり

演習: 各作業の2種類のゆとり



各作業の開始時刻, 終了時刻

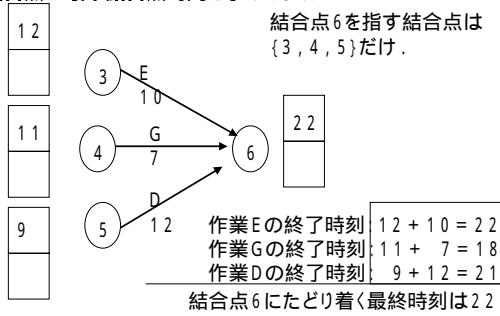
- 作業の最早開始時刻: その作業開始結合点の最早結合点開始時刻
- 作業の最早終了時刻: その作業の最早開始時刻 + 作業時間
- 作業の最遅終了時刻: その作業終了結合点の最遅結合点開始時刻
- 作業の最遅開始時刻: その作業の最遅終了時刻 - 作業時間



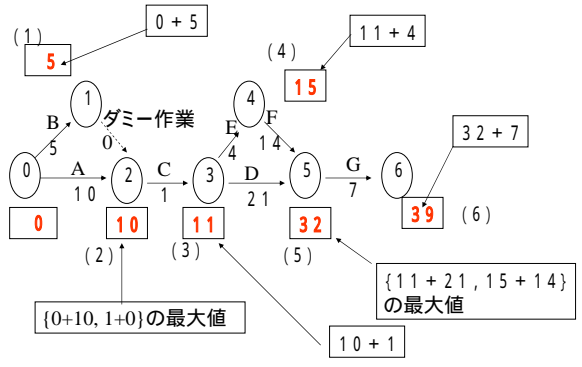
最早結合点開始時刻の求め方

- プロジェクト開始結合点の最早結合点開始時刻 = 0
- 結合点の番号順に計算していく。

例: 結合点の最早結合点時刻を求めるには?



最早結合点開始時刻の計算例

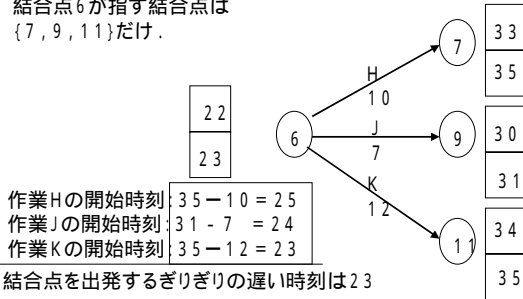


最遅結合点開始時刻の求め方

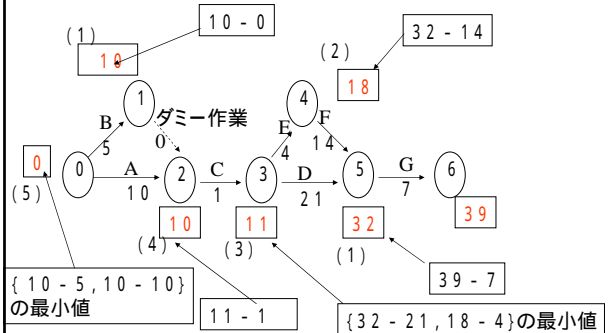
- プロジェクト完了結合点での最遅結合点完了時刻 = プロジェクト完了時刻
- 結合点番号の逆順に計算していく。

例: 結合点の最遅結合点時刻を求めるには?

結合点6が指す結合点は {7, 9, 11} だけ。



最遅結合点開始時刻の計算例

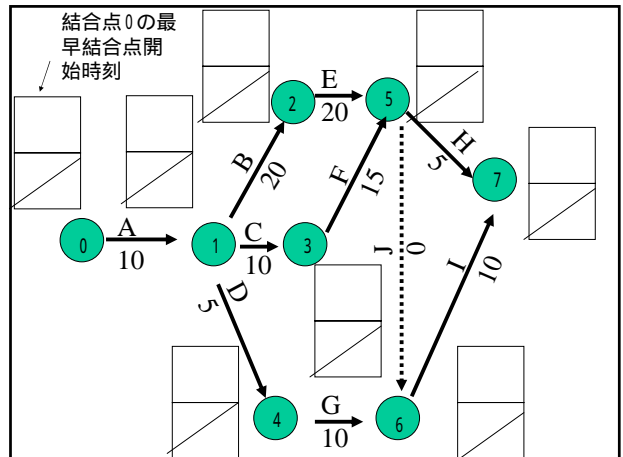


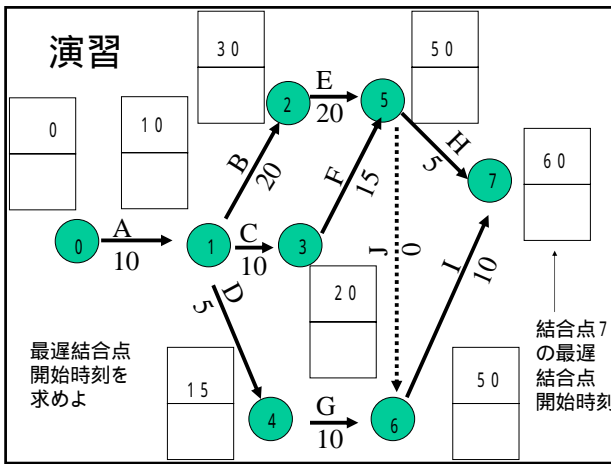
演習

作業リスト

作業	先行作業	作業時間
A	なし	10
B	A	20
C	A	10
D	A	5
E	B	20
F	C	15
G	D	10
H	E, F	5
I	E, F, G	10

各結合点の最早結合点開始時刻を求めよ。





各作業のゆとりの測定

プロジェクト完了時刻を遅らせない範囲で休める最大時間(ゆとり)

- 後続作業に影響を与えないで得るゆとり (自由余裕, フリーフロート)

$$\text{作業終了結合点の最早結合点開始時刻} - \left[\text{作業開始結合点の最早結合点時刻} + \text{作業時間} \right]$$

- 後続作業の開始時刻を遅らせ得るゆとり (全余裕, トータルフロート)

$$\text{作業終了結合点の最遅結合点開始時刻} - \left[\text{作業開始結合点の最早結合点時刻} + \text{作業時間} \right]$$

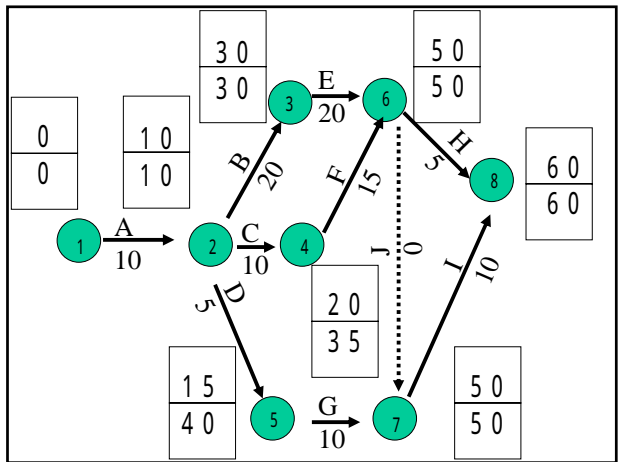
全余裕が0の作業群: クリティカルパス

演習

作業リスト

作業	先行作業	作業時間
A	なし	10
B	A	20
C	A	10
D	A	5
E	B	20
F	C	15
G	D	10
H	E, F	5
I	E, F, G	10

各作業のトータルフロート、フリーフロートを計算し、クリティカルパスを明示すること。



各作業の開始時刻, 終了時刻

- 作業の最早開始時刻: その作業開始結合点の最早結合点開始時刻
- 作業の最早終了時刻: その作業の最早開始時刻 + 作業時間
- 作業の最遅終了時刻: その作業終了結合点の最遅結合点開始時刻
- 作業の最遅開始時刻: その作業の最遅終了時刻 - 作業時間

作業Hの
 最早開始時刻 = 50
 最早終了時刻 = 55
 最遅開始時刻 = 55
 最遅終了時刻 = 60

作業	作業時間	最早開始	最早終了	最遅開始	最遅終了	全余裕	自由余裕	トータル余裕
A	10							
B	20							
C	10							
D	5							
E	20							
F	15							
G	10							
H	5							
I	10							
J (ダミー)	0							

その作業が始めることができる最も早い時刻
 その作業を終わらせることができる最も早い時刻

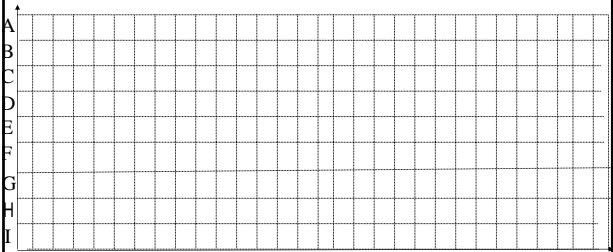
その作業がこの時刻までに始まっていないければ(終わっていないければ)プロジェクト完了時刻がずれ込んでしまう時刻

各作業の最早開始時刻, 最早終了時刻を求めよ.
 各作業の最遅開始時刻, 最遅終了時刻を求めよ.
 各作業の全余裕, 自由余裕を求めよ.
 クリティカルパスに属する作業を求めよ.

作業	作業時間	最早開始	最早終了	最遅開始	最遅終了	全余裕	自由余裕	クリティカルパス
A	10	0	10	0	10	0	0	*
B	20	10	30	10	30	0	0	*
C	10	10	20	10	35	15	0	
D	5	10	15	35	40	25	0	
E	20	30	50	30	50	0	0	*
F	15	20	35	35	50	15	15	
G	10	15	25	40	50	25	25	
H	5	50	55	55	60	5	5	
I	10	50	60	50	60	0	0	*
J(ダミー)	0	50	50	50	50	0	0	*

演習

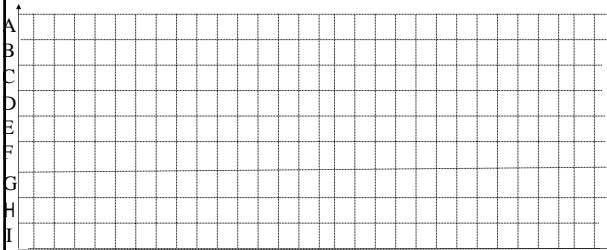
ガントチャート



クリティカルパスに含まれない作業はその作業の最早開始時刻で実施するとする.

演習

ガントチャート



クリティカルパスに含まれない作業はその作業の最遅開始時刻で実施するとする.

プロジェクトを効率良く短縮する: CPM

? : プロジェクト完了をある一定期間までに短縮するため, 費用ができるだけ掛からないスケジュールは?

そんな気軽に言うなよ!
 無駄金がかかったら文句言うくせに!!!!

プロジェクト完了に60日もかかる?
 多少費用がかかってもいいから,
 もう少し短縮できない?



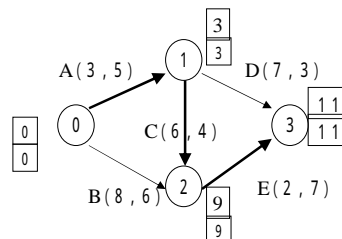
演習

作業	先行作業	標準作業日数 (最短作業日数)	短縮費用 (十万円)
A	なし	3 (2)	5
B	なし	8 (4)	6
C	A	6 (4)	4
D	A	7 (6)	3
E	C, B	2 (1)	7

1. アローダイアグラムを完成すること.
2. プロジェクト完了日数を求めること.
3. 2で求めたプロジェクト完了日数を1日毎徐々に短縮する. 1日短縮のために必要な最小追加費用を短縮作業を求めること.
4. 1日短縮で生じるクリティカルパスを求めよ.

アローダイアグラム

作業	先行作業	標準作業日数 (最短作業日数)	短縮費用 (十万円)
A	なし	3 (2)	5
B	なし	8 (4)	6
C	A	6 (4)	4
D	A	7 (6)	3
E	C, B	2 (1)	7



プロジェクト完了日数は11日

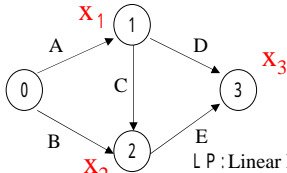
{A, C, E} がクリティカルパス

PERT, CPMを線形計画問題(LP)へ

作業	先行作業 (最短作業日数)	標準作業日数	短縮費用 (十万円)
A	なし	3 (2)	5
B	なし	8 (4)	6
C	A	6 (4)	4
D	A	7 (6)	3
E	C,B	2 (1)	7

PERT, CPM

ネットワーク上の最適化問題

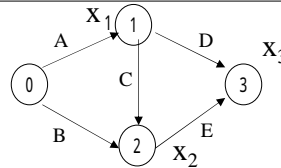


LP: Linear Programming
線形不等式の制約の下, 線形関数の最適化

変数 X_i の標準作業時間に関する制約

作業	先行作業	標準作業日数 (最短作業日数)	短縮費用 (十万円)
A	なし	3 (2)	5
B	なし	8 (4)	6
C	A	6 (4)	4
D	A	7 (6)	3
E	C,B	2 (1)	7

X_1	3
X_2	8
X_2	$X_1 + 6$
X_3	$X_1 + 7$
X_3	$X_2 + 2$

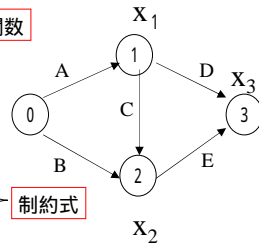


$X_i \geq 0 (i=1,2,3)$

非負制約

PERTのLPモデル

最小化 x_3 (目的関数)
 制約 (s.t.)
 $x_1 \leq 3$
 $x_2 \leq 8$
 $x_2 \leq x_1 + 6$
 $x_3 \leq x_1 + 7$
 $x_3 \leq x_2 + 2$
 $x_i \geq 0 (i=1,2,3)$ (非負制約)



PERTのLPモデルを解いて, わかること

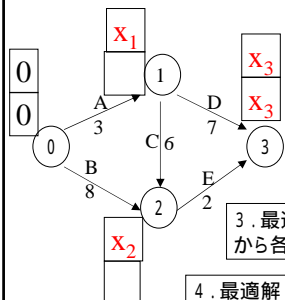
最小化 x_3
 制約 (s.t.)
 $x_1 \leq 3$
 $x_2 \leq 8$
 $x_2 \leq x_1 + 6$
 $x_3 \leq x_1 + 7$
 $x_3 \leq x_2 + 2$
 $x_i \geq 0 (i=1,2,3)$

制約式全てと非負条件 $x_i \geq 0 (i=1,2,3)$ を満たす変数 x_1, x_2, x_3 の組合せの中で, 最小の目的関数値を達成する変数 x_1, x_2, x_3 の組合せを**最適解**, 最小の目的関数値を**最適値**と呼ぶ

最適解 (x_1, x_2, x_3) は
結合点1の最早結合点開始時刻が x_1 , 結合点2の最早結合点開始時刻が x_2 , 結合点3の最早結合点開始時刻が x_3 であることを意味する.

最適値 x_3 は
プロジェクト完了時刻が x_3 であることを示す

PERT-LPモデルの最適解からPERT情報の算出



1. 最適解 (x_1, x_2, x_3) から各作業のトータルフロートを算出

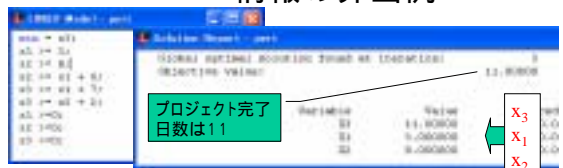
2. 最適解 (x_1, x_2, x_3) から最遅結合点開始時刻を算出

結合点2の最遅結合点開始時刻は $x_2 - 2$
結合点1の最遅結合点開始時刻は $\min(\text{結合点2の最遅結合点開始時刻} - 6, x_1 - 7)$

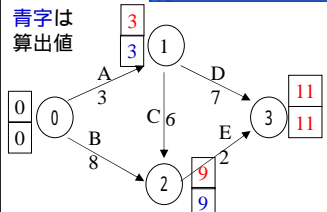
3. 最適解 (x_1, x_2, x_3) と最遅結合点開始時刻から各作業のフリーフロートを算出

4. 最適解 (x_1, x_2, x_3) と最遅結合点開始時刻から各作業の最早開始時刻, 最遅開始時刻を算出

PERTのLPモデルの最適解からPERT情報の算出例



青字は算出値



	トータルフロート	フリーフロート	最早開始	最遅開始
A	0	0	0	0
B	1	1	0	1
C	0	0	3	3
D	1	1	3	4
E	0	0	9	9

CPMのLPモデル: 作業短縮日数の変数の導入

作業	先行作業	標準作業日数 (最短作業日数)	短縮費用 (十万円)
A	なし	3(2)	5
B	なし	8(4)	6
C	A	6(4)	4
D	A	7(6)	3
E	C,B	2(1)	7

作業Aの短縮日数を
変数 X_A

⋮

作業Eの短縮日数を
変数 X_E

作業の短縮可能条件

作業	標準作業日数	短縮可能日数	変数	係数
A	3(2)	1	X_A	1
B	8(4)	4	X_B	4
C	6(4)	2	X_C	2
D	7(6)	1	X_D	1
E	2(1)	1	X_E	1

$$X_i = 0 \quad (i=A,B,C,D,E)$$

非負制約

CPMのLPモデル:完了日数の指定と費用最小化

短縮費用を
最小化

$$\text{Min. } 5X_A + 6X_B + 4X_C + 3X_D + 7X_E$$

$$\text{s.t. } \begin{array}{rcl} X_1 & 3 - X_A & X_A = 1 \\ X_2 & 8 - X_B & X_B = 4 \\ X_2 & X_1 + 6 - X_C & X_C = 2 \\ X_3 & X_1 + 7 - X_D & X_D = 1 \\ X_3 & X_2 + 2 - X_E & X_E = 1 \end{array}$$

完了日数を
10日に指定

$$X_3 = 10$$

$$X_i = 0 \quad (i=1,2,3) \quad X_i = 0 \quad (i=A,B,C,D,E)$$

CPMのLPモデル:完了日数の指定と費用最小化

完了日数を
9日に

$$\text{Min. } 5X_A + 6X_B + 4X_C + 3X_D + 7X_E$$

$$\text{s.t. } \begin{array}{rcl} X_1 & 3 - X_A & X_A = 1 \\ X_2 & 8 - X_B & X_B = 4 \\ X_2 & X_1 + 6 - X_C & X_C = 2 \\ X_3 & X_1 + 7 - X_D & X_D = 1 \\ X_3 & X_2 + 2 - X_E & X_E = 1 \end{array}$$

完了日数を
9日に指定

$$X_3 = 9$$

希望完了日数を指定

$$X_i = 0 \quad (i=1,2,3) \quad X_i = 0 \quad (i=A,B,C,D,E)$$

演習3

作業	先行作業	標準作業日数 (最短作業日数)	短縮費用 (十万円)
A	なし	3(2)	5
B	なし	8(4)	6
C	A	6(4)	4
D	A	7(6)	3
E	C,B	2(1)	7

1. プロジェクト完了日数を10日, 9日, 8日と短縮する場合, それぞれの完了日数で費用最小を達成するための短縮作業とその短縮日数を計算機を使って求めること.

2. 最短のプロジェクト完了日数を求めること.