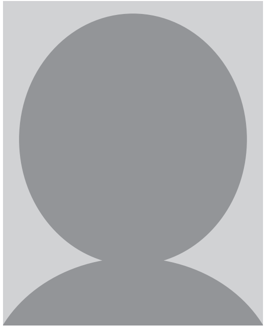
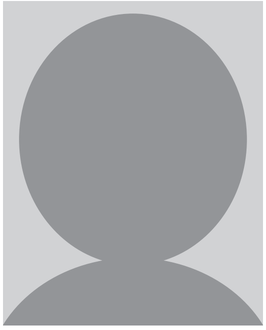
問題解決の数理（’１７）

- 収録本番とは多少異なっていることがあります.
- 内容の間違いのご指摘は歓迎します.
- 「完全に無保証」です.



- 今回の講義では，スケジューリングについてお話します．
- スケジュールとは作業やイベントの時刻を定めた計画のことで，その計画を立てることをスケジューリングといいます．
- スケジューリングには，様々なタイプのものがありますが，今回はプロジェクトのスケジューリングについてお話します．



- ここでのプロジェクトとは，特定の目的を達成するための作業の集まりのことを指します.
- そして，プロジェクトを構成する各作業時刻を定めた計画を立てることをプロジェクトのスケジューリングといいます.
- 製品開発など多くの作業からなるプロジェクトでは，目標とする日数でプロジェクトを完了できるようにスケジューリングを行います.

PERT

PERT (Program Evaluation and Review Technique)

- 大規模プロジェクトを期間内に完了するために，各作業の開始時期，投入資源などを分析する手法
- PERT/CPM CPM: (Critical Path Method)

- 今回は，スケジューリングの手法として，PERT と CPM について，お話します.
- PERT は，プロジェクトを一定期間内に完了させるには，どの作業をいつから開始するか，重点となる作業は何かなどを分析する手法です.
- 1950 年代後半に，潜水艦発射の弾道ミサイル開発プロジェクトの中で，開発された手法です.
- ほぼ同時期に，化学プラント建設に際して、コスト最適化を目的に開発された CPM と呼ばれる手法もあります.
- PERT と CPM は独立に開発されたのですが，非常に類似した方法で，これらを合わせて PERT/CPM とまとめて呼ぶこともあります.
-

PERT

PERT の手順

1. プロジェクトを構成する作業に分解
2. 作業間の先行関係を記述
3. 作業間の先行関係に基づき，プロジェクトを構成する作業を結合したネットワーク（アロー・ダイアグラム）を構成
4. 各作業に所要時間を割り当てる
5. 作業の時間的な指標を算出し分析

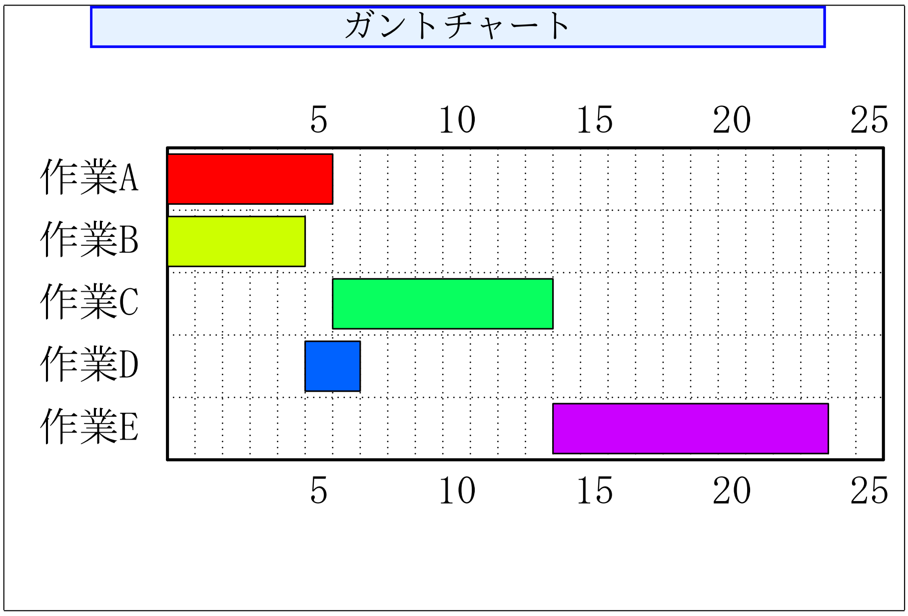
- PERT の一般的な手順は次の通りです．
- まず，プロジェクトを明確化し，プロジェクトを構成する作業に分解します．
- ある作業を開始するには，別の作業が終了している必要があるということがあります．
- これを作業間の先行関係と言いますが，PERT の次のステップでは，作業間の先行関係を記述します．
- 次に，作業間の先行関係に基づき，プロジェクトを構成する作業を結合したアロー・ダイアグラムと呼ばれるネットワークを構成します．
- その次に，各作業に所要時間を割り当てます．
- これら準備の後，作業の時間的な指標を算出し，各作業をいつ開始すべきか，プロジェクトの完了を遅らせないために，どの作業に資本を投入するかなどを分析します．
- また，プロジェクトが予定時間通りに進行しない場合，以降の作業計画の再検討なども行います．
- PERT の説明の前に，プロジェクトの具体例を示します．

PERT

ケーキの飾りつけプロジェクトの作業リスト

記号	作業内容	所要時間 (分)	先行作業
A	スポンジの整形	6	-
B	ホイップクリームを作る	5	-
C	スポンジにホイップクリームを塗る	8	A, B
D	絞り袋にクリームを詰める	2	B
E	クリームを絞り飾り付ける	10	C, D

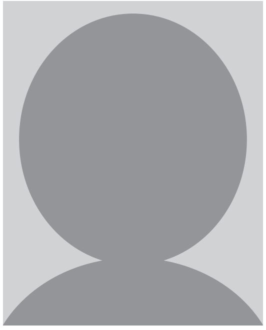
- ここでは，プロジェクトと呼ぶにはいささか大げさではありますが，ケーキの飾り付けを考えてみます．
- 表はケーキの飾りつけの作業リストです．
- 作業には時間的順序関係があります．
- 例えば，作業Ｃのスポンジにホイップクリームを塗る作業を開始するには，作業Ａのスポンジの整形と作業Ｂのホイップクリーム作りが終了している必要があります．
- ある作業の直前の作業を「先行作業」，ある作業の直後の作業を「後続作業」と呼びます．
- この作業リストは次のように読みます．
- 作業Ａのスポンジの整形は先行作業がないので，いつでも開始することができ，作業には６分かかります．
- 作業Ｂのホイップクリーム作りも先行作業がないので，いつでも開始することができ，作業には５分かかります．
- 作業Ｃのスポンジにホイップクリームを塗る作業を開始するには，先行作業である作業Ａと作業Ｂが終了している必要があり，作業Ｃには８分かかります．
- 作業Ｄの「絞り袋にホイップクリームを詰める作業」を開始するには，先行作業である作業Ｂが終了している必要があり，作業Ｄには２分かかります．
- 作業Ｅの「クリームを絞り飾りつける作業」を開始するには，先行作業である作業Ｃと作業Ｄが終了している必要があり，作業Ｅには１０分かかります．
-



- プロジェクトの可視化の方法の一つにガントチャートがあります．
- ガントチャートは，横軸を時間，縦軸を作業とした棒グラフによりプロジェクトを表現します．
- ガントチャートは，各作業がいつ始まり，いつ終わるかを分かりやすく表現します．
- この図では，例えば，作業 A を最初に始めて，作業 C は 6 分目から始め，作業 E は 24 分目に終了し，プロジェクトは完了するといったことが見て取れます．
- この図では縦軸を作業としていますが，縦軸をその作業の担当者とすれば，プロジェクトのメンバーには，自分がいつ何をすればよいかが一目瞭然で，スケジュールの把握が容易です．
- ただし，作業間の相互の依存関係が分かりづらく，特定の作業の作業時間の短縮や遅延が，全体にどのような影響を及ぼすかについて分析することが困難ですので，プロジェクトの遂行中の計画変更や状況変化に対して柔軟に対応できないという欠点があります．

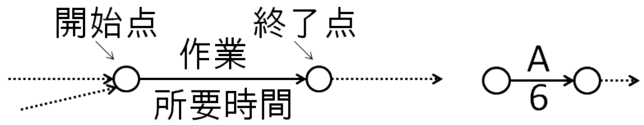
- そのため，ガントチャートは，既に出来上がったスケジュールを表示するには有効ですが，スケジューリングにはあまり適していません．
- スケジューリングには，ガントチャートより，次に説明するアロー・ダイアグラムのほうが有効です．

アロー・ダイアグラム



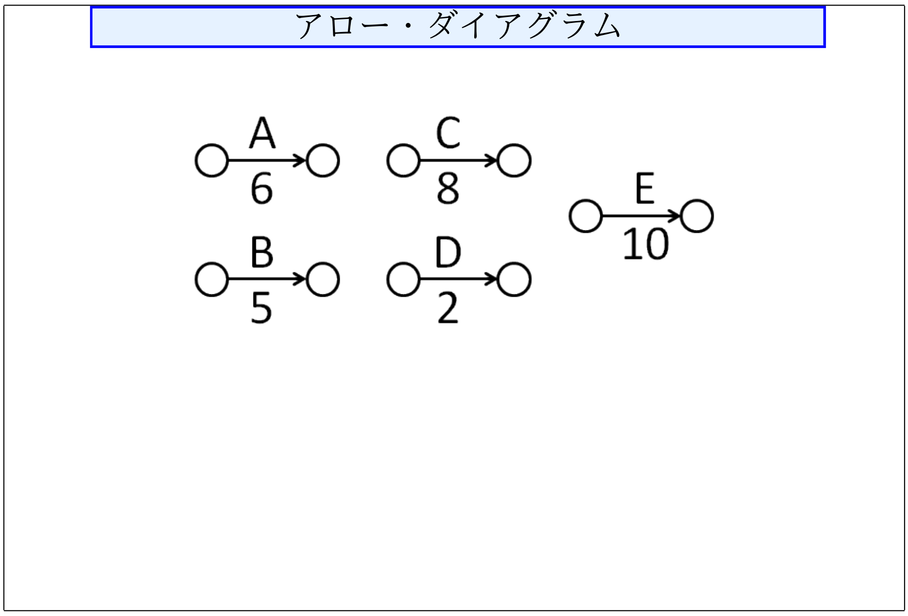
- ここでは，プロジェクトにおける作業の依存関係を表現するアロー・ダイアグラムについて説明します．

アロー・ダイアグラム



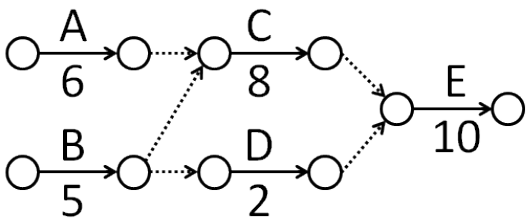
(a) (b)

- アロー･ダイアグラムではプロジェクトの作業を枝，ここでは矢印で表します．
- 作業の開始時刻を表す開始点，終了時刻を表す終了点を…点，ここでは○で表します．
- 図 (a) に示すように，作業の開始点と終了点を矢印で結び，その作業を示す記号と作業所要時間を書き込みます．
- ケーキの飾り付けにおける作業 A は 6 分かかるので，図 (b) のように表します．
- ケーキの飾りつけの各作業の開始点と終了点を矢印で結び，作業を示す記号と所要時間を記入すると次のようになります．

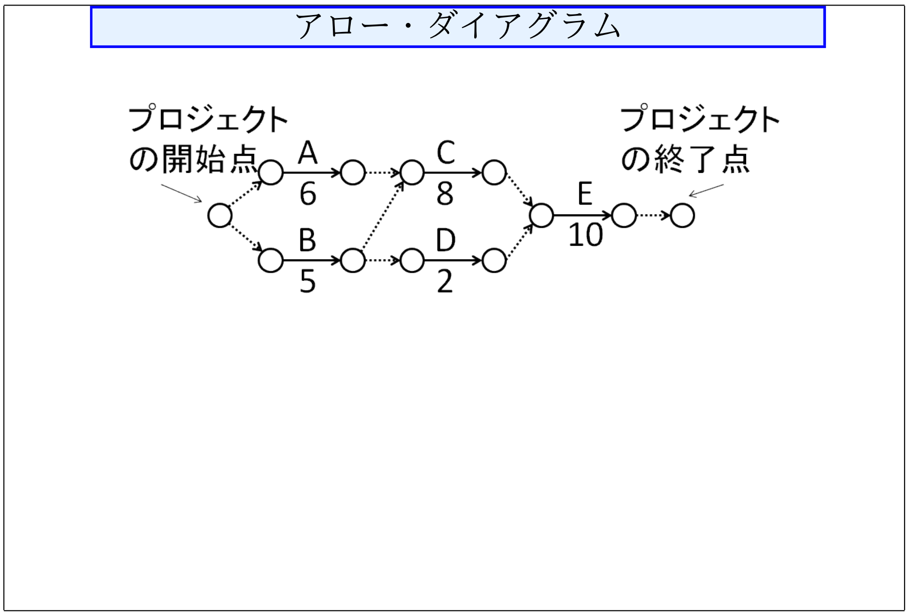


- これでは，作業間の先行・後続関係が分かりません．
- 作業の先行・後続関係を明示するために，先行作業の終了点と後続作業の開始点を「ダミー作業」と呼ばれる作業時間０の作業で結びます．
- ダミー作業は「ダミー枝」と呼ばれる破線の矢印で表します．
- ケーキの飾り付けのプロジェクトの作業間をダミー枝で結ぶとこのようになります．

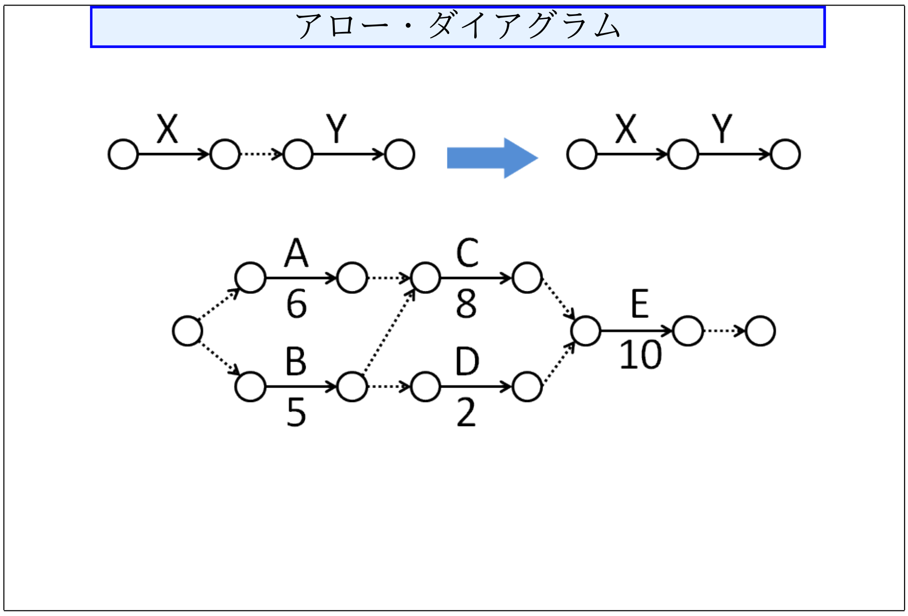
アロー・ダイアグラム



- 作業Ｃの先行作業はＡとＢです.
- 作業Ｄの先行作業はＢです.
- 作業Ｅの先行作業はＣとＤです.
- これで作業間の先行・後続関係が表現できました.
- さらにプロジェクトの開始点と終了点を加えます.

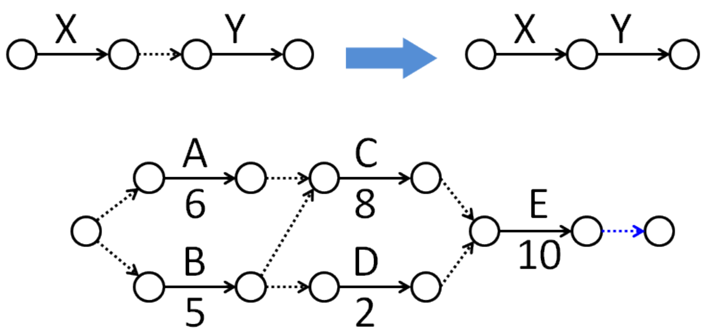


- プロジェクトの開始点と，先行作業のない作業の開始点をダミー枝で結びます.
- また，後続作業のない作業の終了点とプロジェクトの終了点をダミー枝で結びます.
- これで，プロジェクトのアロー・ダイアグラムは一応できあがりました.



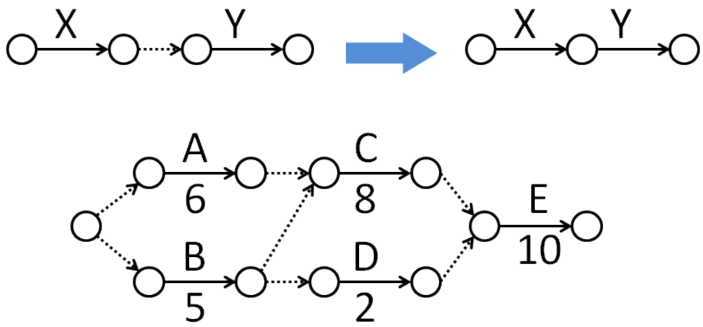
- ここまでで，アロー・ダイアグラムは一応でき上がったのですが，作業の数が少ない割に，点や枝の数が多く，見やすくありません．
- そこで，ダミー作業を省略してコンパクトで見やすく変換していきます．
- まず，ダミー作業省略の最初のルールについて説明します．
- 上の図をご覧ください．
- 先行作業Ｘと後続作業Ｙが直列に結ばれていますが，このような場合，先行作業の終了点と後続作業の開始点を１つにまとめ，その間のダミー枝は省略することができます．
- このルールを適用していきます．

アロー・ダイアグラム

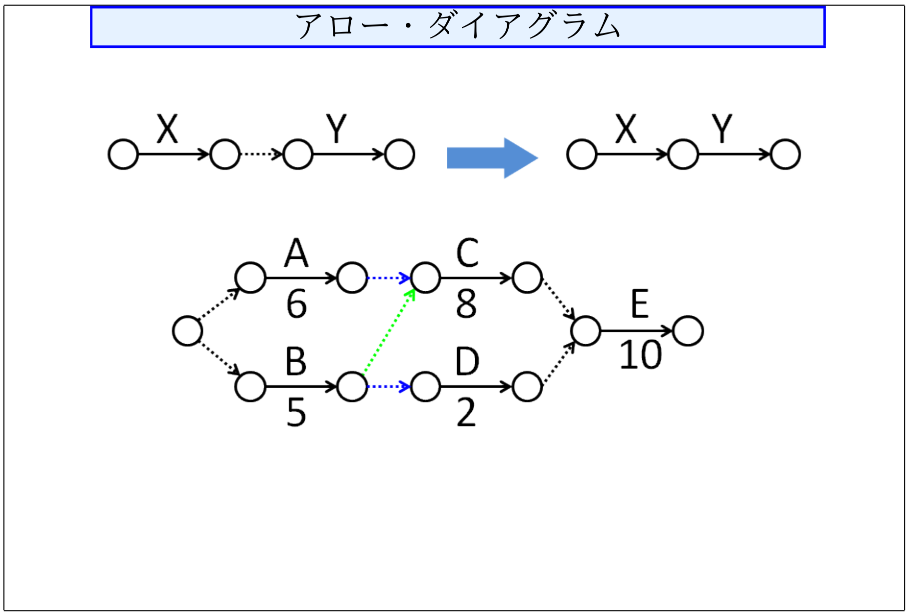


- まず，プロジェクトの終了点に繋がっている，一番後ろのダミー作業に注目します．
- プロジェクトの開始点や終了点では，枝の出入りの一方がありませんが，同様に扱います．
- この部分に，上の図のダミー作業省略ルールを適用すると，このようになります．

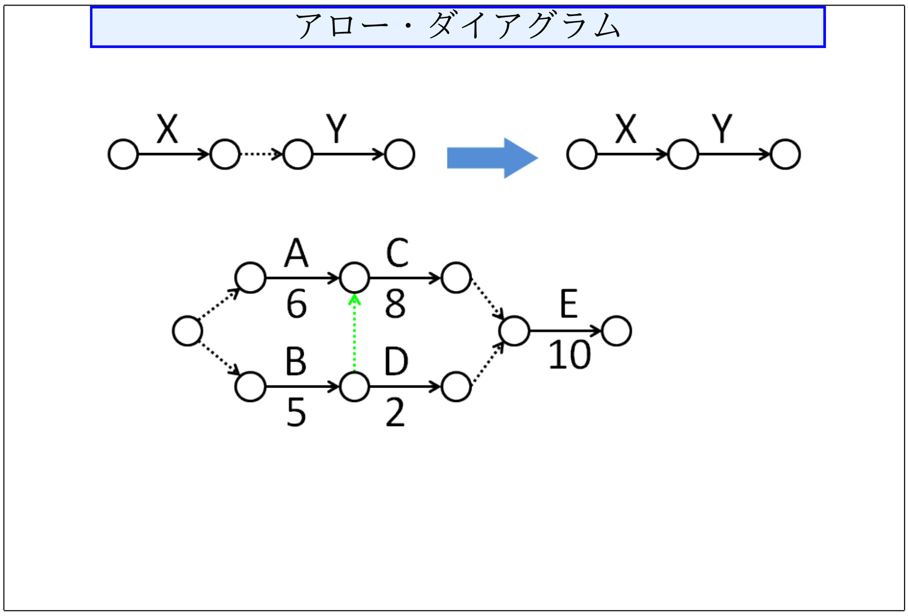
アロー・ダイアグラム



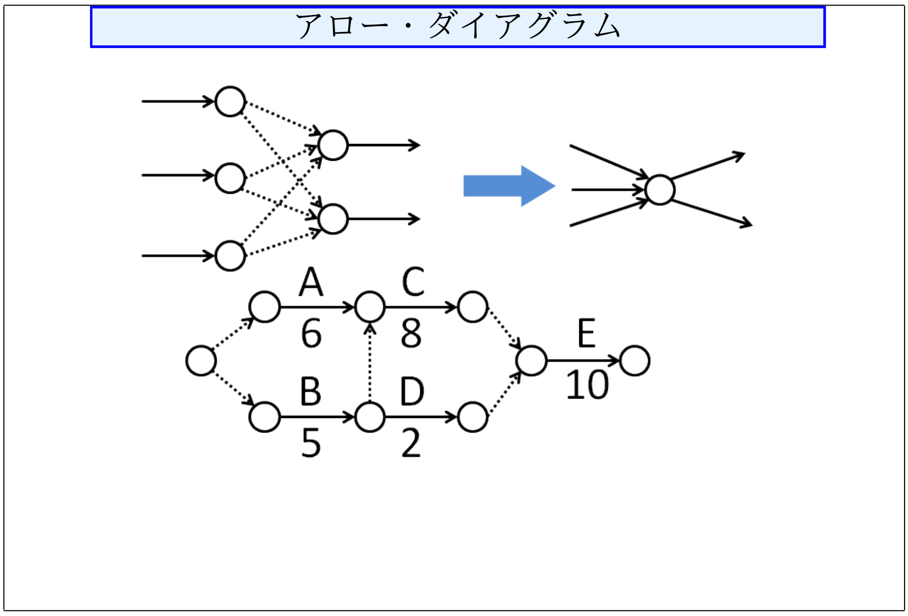
- ちょっとコンパクトになりました．



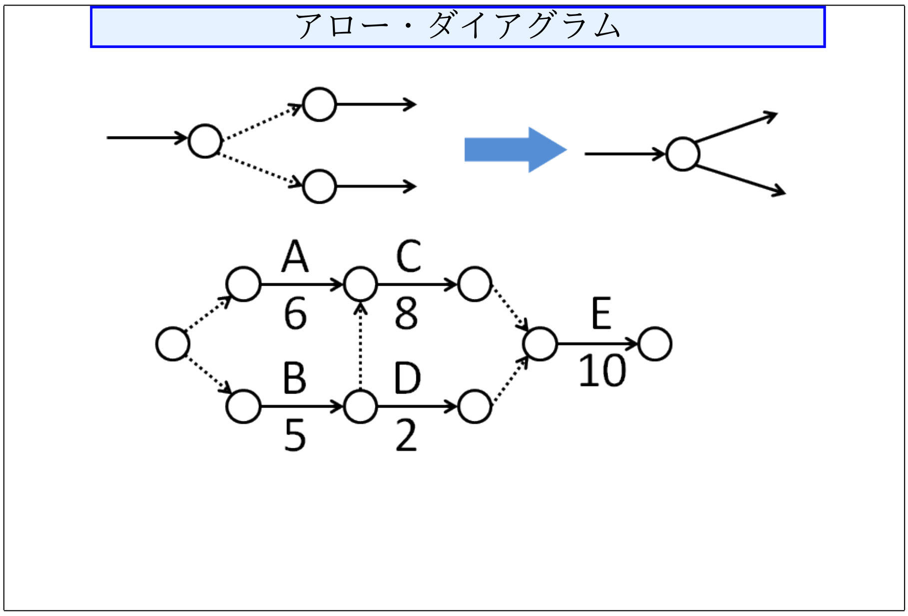
- 次に，中央部分のダミー作業に，ダミー作業省略ルールを適用します．
- 作業 A の終了点と作業 C の開始点を 1 つにまとめ，間のダミー枝を省略します．
- 同様に，作業 B の終了点と作業 D の開始点を 1 つにまとめ，間のダミー枝を省略します．
- 作業 B の終了点と作業 C の開始点を結んでいるダミー枝は，引き続き作業 B の終了点と作業 C の開始点を結びます．



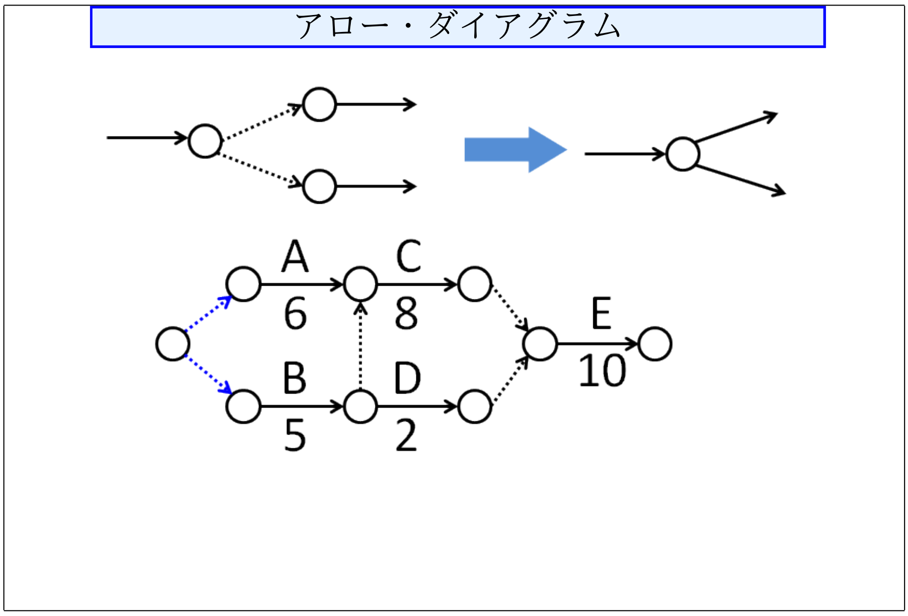
- ダミー作業を省略した結果，このようにさらにコンパクトになります．



- 今度は，上の図のようなダミー作業の省略ルールを用います．
- 図のように，同じ先行作業群を持つ，複数の作業開始点が存在する場合，ダミー作業を省略することができます．
- ちょっとややこしいので，この例に合うように，単純化したものを示します．

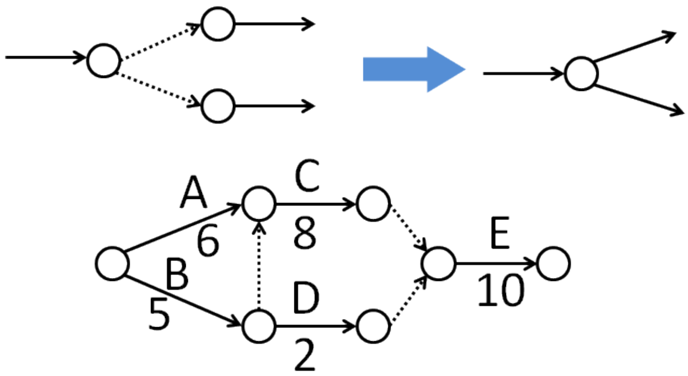


- このルールは先のルールの特殊形になっています.
- このルールを下のアロー・ダイアグラムに適用します.

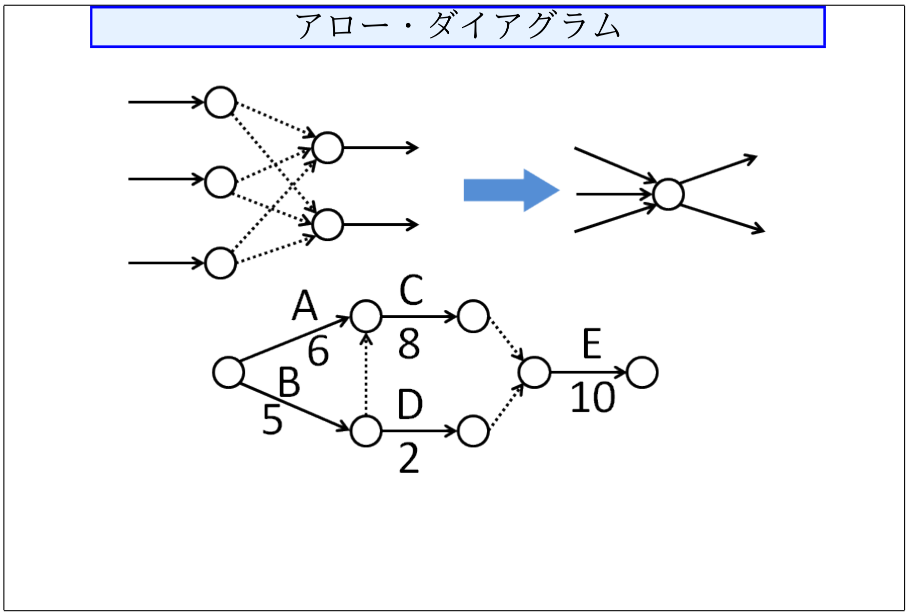


- このルールをプロジェクトの開始点に適用します．
- 開始点の先行作業がありませんが，開始点に入る枝があるのと同様に考えます．

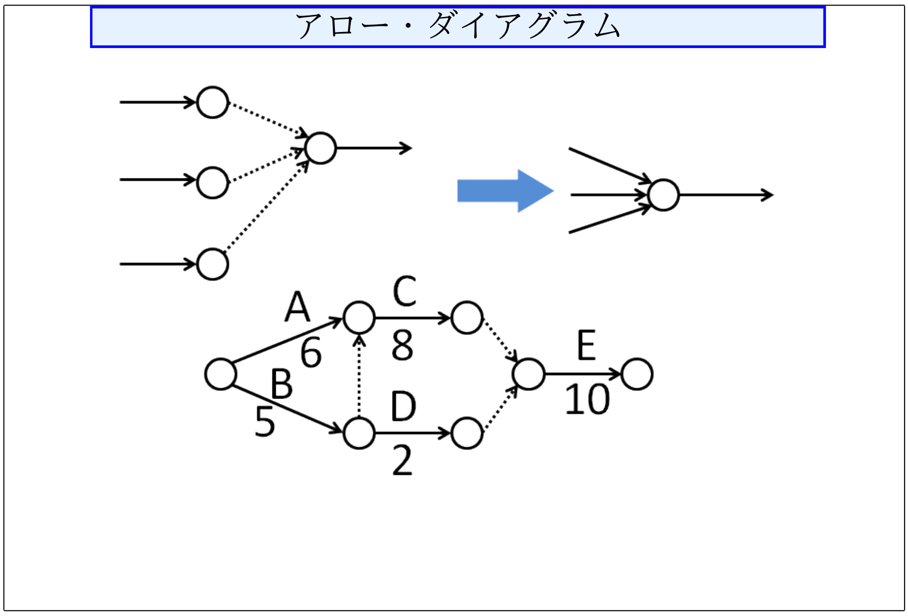
アロー・ダイアグラム



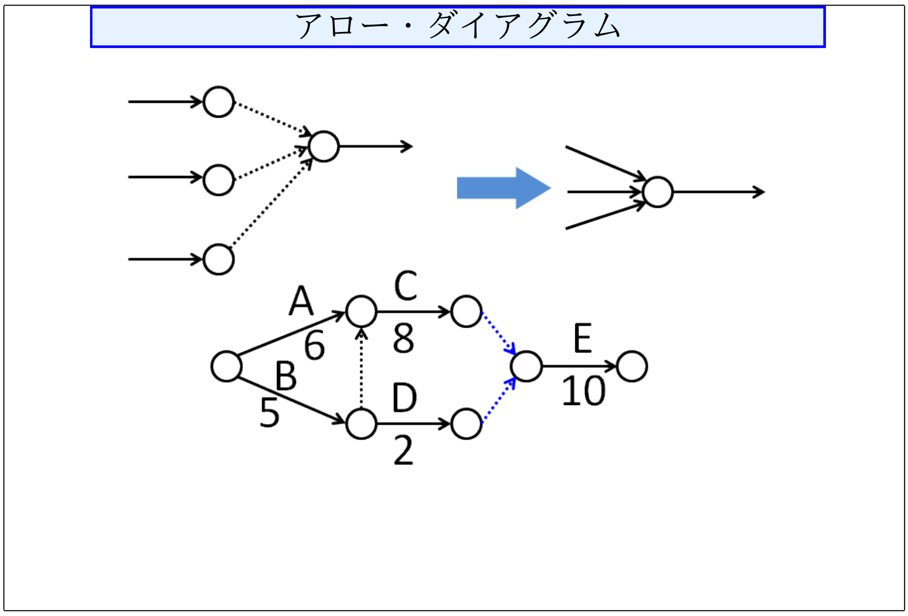
- 開始点にダミー作業省略ルールを適用した結果，このようにだいぶすっきりしてきました．



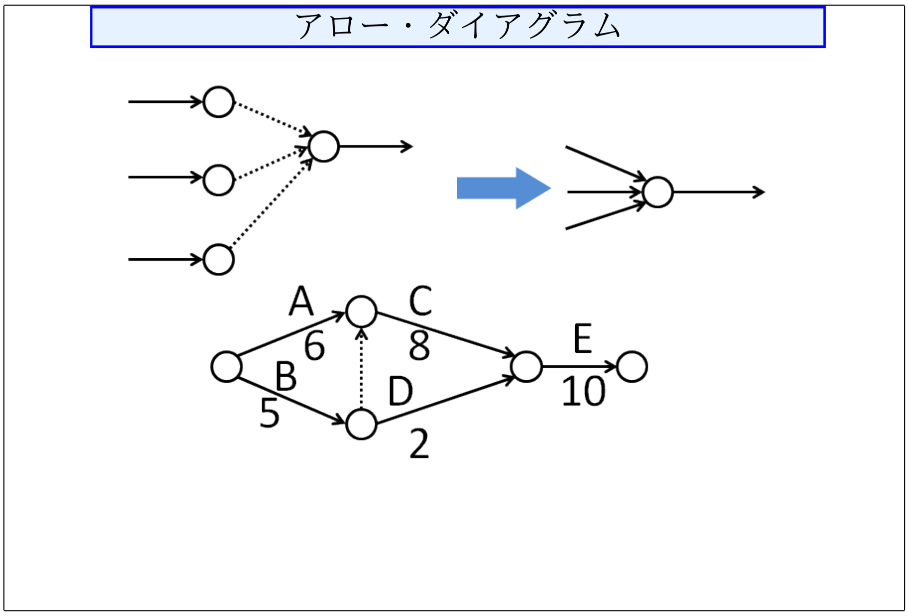
- 次は，上の図のルールの別の特殊形を考えます.



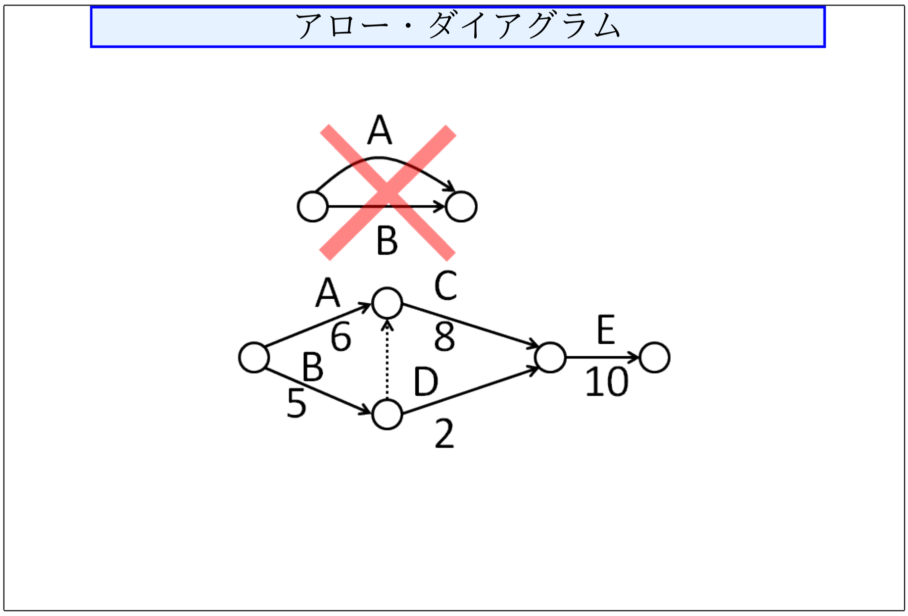
- このようなものです．
- これもまた，先のルールの特殊形になっています．
- このルールを下のアロー・ダイアグラムに適用します．



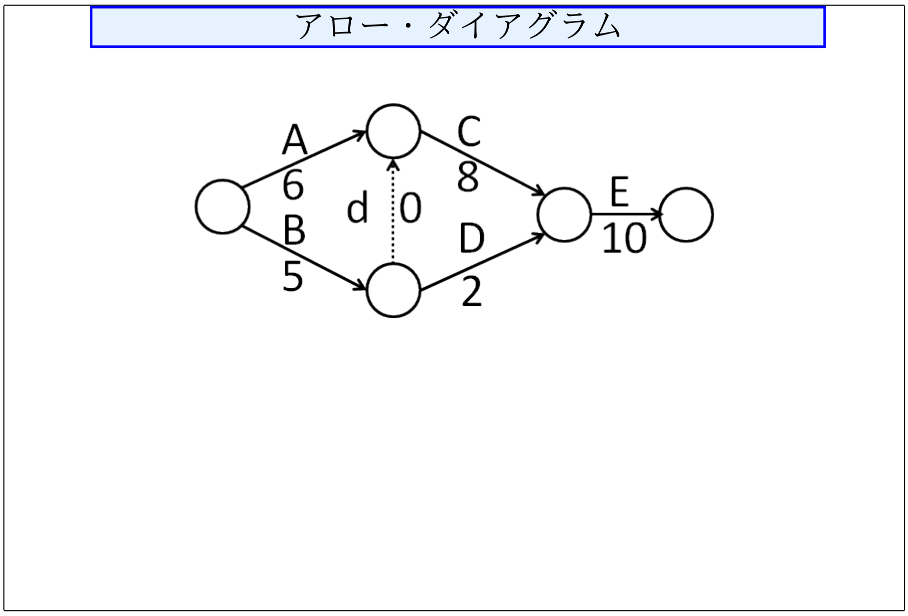
- この部分に注目します.
- この部分に上のルールを適用します.



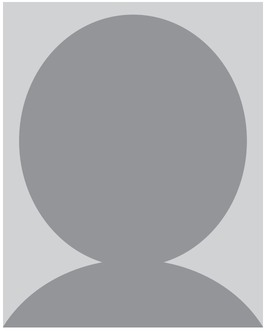
- すると，このようになります．



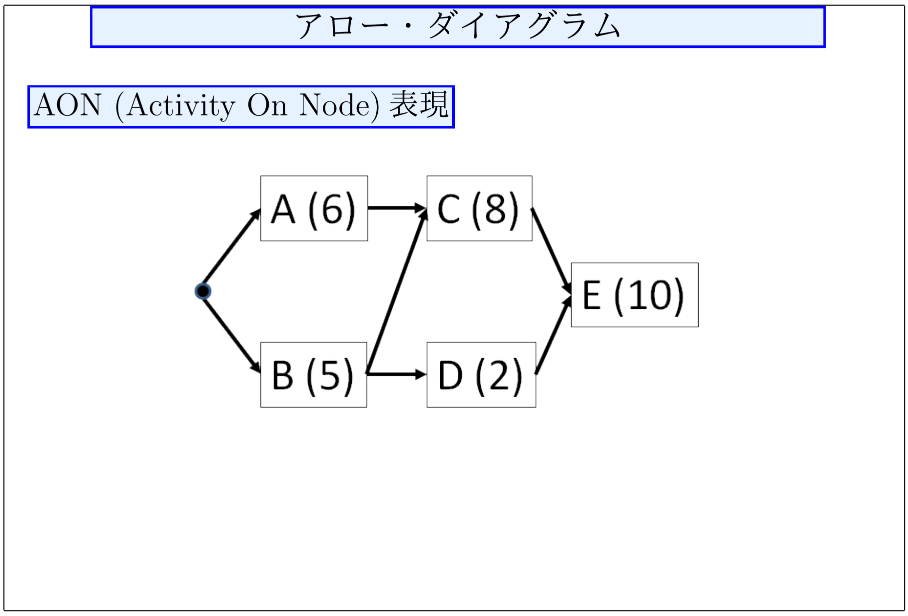
- まだ，ダミー作業が 1 つ残っています．
- 作業 B の終了点と作業 C の開始点を 1 つにまとめて，ダミー作業を省略すると，作業 A と作業 B は同一の開始点と終了点を持つことになります．
- しかし，アロー・ダイアグラムは，2 点間を直接結ぶ枝が複数になることを許していませんので，このダミー作業を省略することはできません．
- 形状がいびつになったので，整形すると，アロー・ダイアグラムは次のようになります．



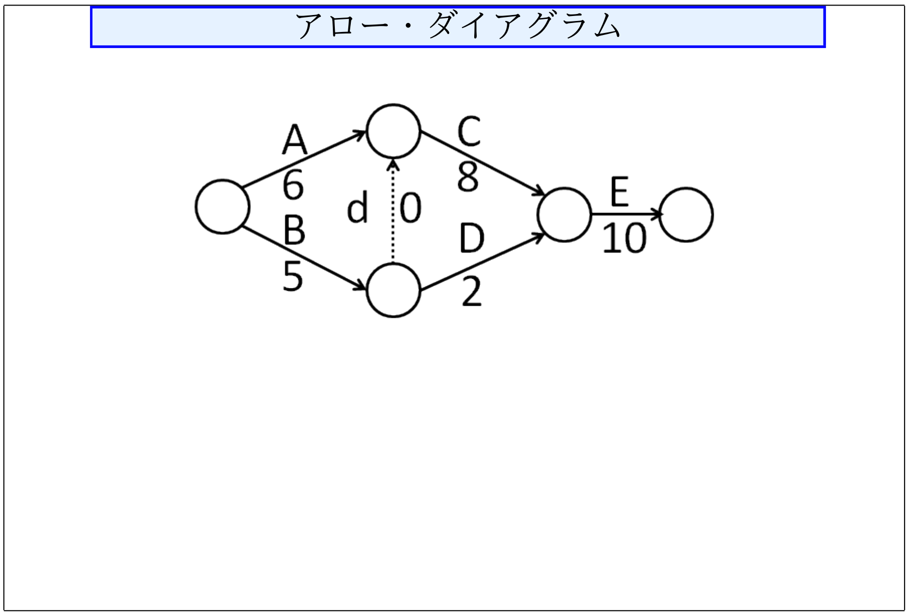
- ダミー作業の省略により，元のアロー・ダイアグラムに比べると，かなりすっきりしました． 次，顔出し



- ダミー作業の省略が難しいと感じる方もいるかと思います.
- ダミー作業は残しておいても，スケジュール自体に影響を与えることはありませんので，何が何でもすべてのダミー作業を省略しなければいけないというわけではありません.
- また，実際のプロジェクトのスケジューリングにおいては，ネットワークの描画や計算を行うプロジェクト管理ソフトウエアを利用するのが一般的です.

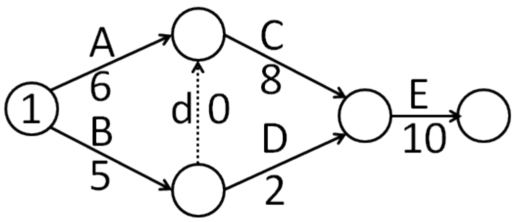


- 実は，アロー・ダイアグラムは計算機でも扱いにくく，プロジェクト管理ソフトウエアの多くは，これまでに示したアロー・ダイアグラムではなく，図に示すような AON 表現のダイアグラムを用います．
- AON は activity on node のことで，枝ではなく，点に作業の記号や所要時間を書き込む表現方法です．
- AON 表現ではダミー作業は使われません．
- 一方，教科書や資格試験では，アロー・ダイアグラムが使われることが多く，また，コンパクトに情報を表現できますので，この講義ではアロー・ダイアグラムを用いていますが，ダミー作業の省略に関しては，あまり神経質になる必要はなく，アロー・ダイアグラムは読めれば十分かと思います．



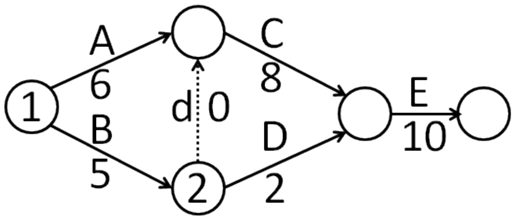
- 次は，アロー・ダイアグラムの点に番号をつけていきます．
- 番号付けは次のルールに基づき行われます．
- まず，先行作業がない点，すなわちプロジェクトの開始点に 1 をつけます．
- 以降，番号付けされた点から出ている枝以外に先行作業のない点から順番に番号をつけていきます．
- 途中で番号付けされた点から出ている枝以外に，先行作業がない点が複数現れたときは，それらの間ではどの順に番号をつけても構いません．
- このような順序を「トポロジカル順」といいます．
- この方法でこのアロー・ダイアグラムの点に番号をつけて行きます．

アロー・ダイアグラム



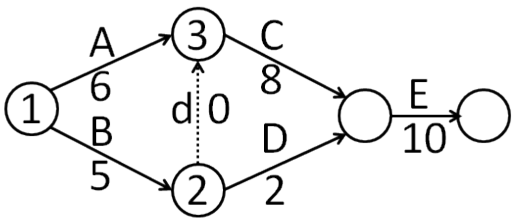
- まず，プロジェクトの開始点に 1 をつけます．
- 以降，番号付けされた点から出ている枝以外に先行作業のない点から順番に番号をつけていきます．
- 点 1 からは作業 A に相当する枝と作業 B に担当する枝が出ています．
- これらのうち，作業 A の終了点には作業 A の枝以外に，まだ番号付けされていない点からダミー作業スモール d の枝が入っています．
- 一方，作業 B の終了点は，点 1 から作業 B の枝のみが入っています．
- すなわち，作業 B の終了点は，番号付けされた点 1 から出ている作業 B 以外に先行作業がありませんので，作業 B の終了点に 2 をつけます．
-

アロー・ダイアグラム



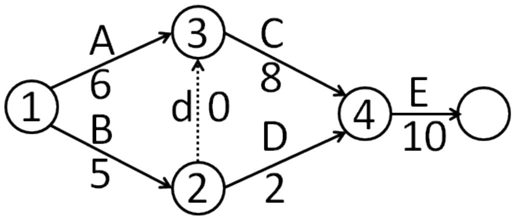
- 点 2 からは作業 D に相当する枝とダミー作業スモール d に担当する枝が出ています．
- ダミー作業スモール d の終了点は，作業 A の終了点でもあります．
- ダミー作業スモール d の開始点は点 2，作業 A の開始点は点 1 と，いずれも番号付けされていますので，ダミー作業スモール d および作業 A の終了点に 3 をつけます．
-

アロー・ダイアグラム



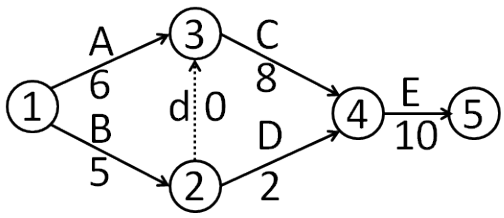
- 点 3 からは作業 C に相当する枝が出ています.
- 作業 C の終了点は作業 D の終了点でもあります.
- 作業 C の開始点は 3，作業 D の開始点は 2 と番号付けされておりますので，作業 C および作業 D の終了点に 4 をつけます.
-

アロー・ダイアグラム



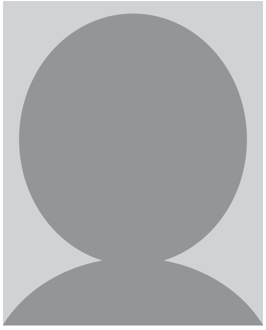
- 残る点は 1 つだけで，プロジェクトの終了点に 5 をつけます.

アロー・ダイアグラム



- 以上で，すべての点にトポロジカル順に番号をつけることができました．
- これで，ようやく分析の準備ができました．

特性値の算出



顔出し

- アロー・ダイアグラムが完成しましたので，プロジェクトのスケジュールの分析に移ります.
- そのために，スケジュールを特徴づける指標となる特性値を算出します.

特性値の算出

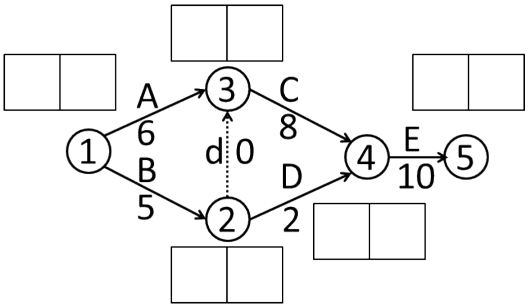
最早節点時刻 各点において最も早く作業を開始できる時刻

最遅節点時刻 プロジェクトを最早で完了させるために，各点から開始する作業の少なくとも一つを開始すべきもっとも遅い時刻

- まず，各点において，最も早く作業を開始できる時刻である,「最早節点時刻」を求めます.
- 次に，プロジェクトを最も早くで完了させるために，各点から開始する作業の少なくとも一つを開始すべき最も遅い時刻である,「最遅節点時刻」を求めます.

特性値の算出

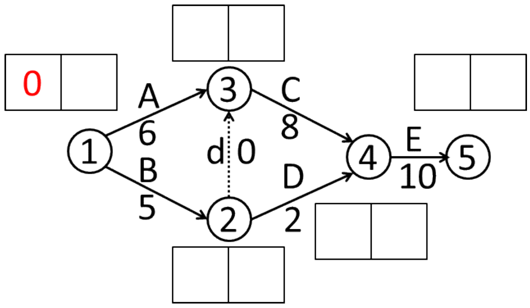
アロー・ダイアグラム



- アロー・ダイアグラムの点の傍らにマスを設けます.
- 左のマスに最早節点時刻，右のマスに最遅節点時刻を書き込んでいきます.

特性値の算出

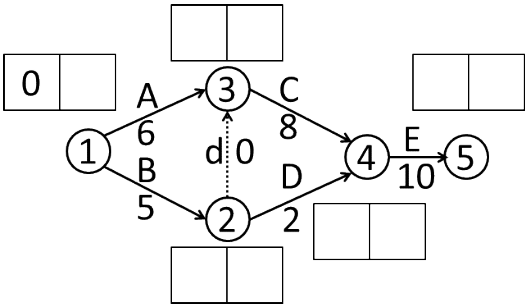
最早節点時刻



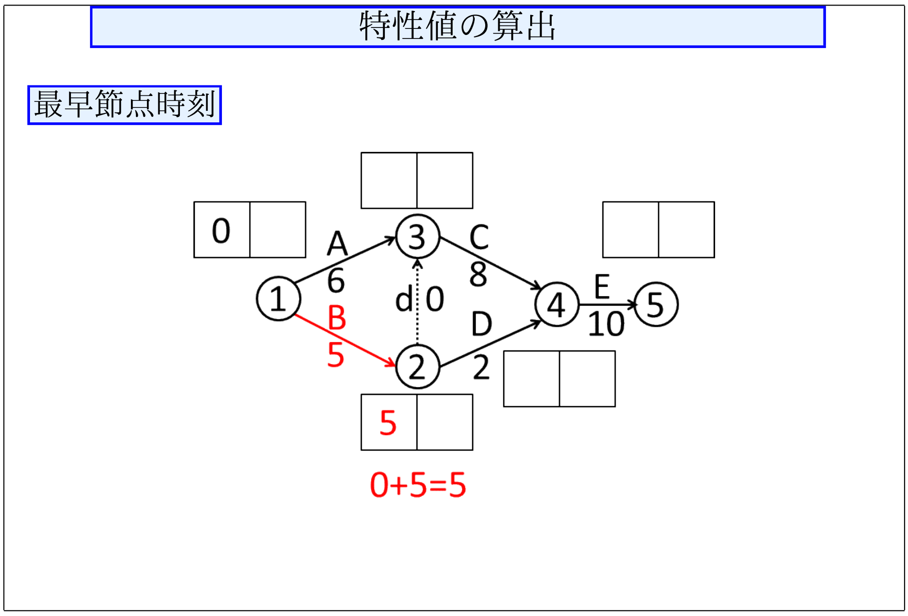
- まず，各点において，作業を最も早く開始できる時刻である最早節点時刻を算出します．
- 最初に，プロジェクトの開始点の最早節点時刻を 0 とします．
- 通常，PERT の分析対象になるプロジェクトは，完了まで何日，あるいは何か月，場合によっては何年もかかるものですので，プロジェクトの開始点の最早節点時刻は第 0 日ということになります．
- ケーキの飾りつけは分単位のプロジェクトですので，時刻 0 分ということになります．

特性値の算出

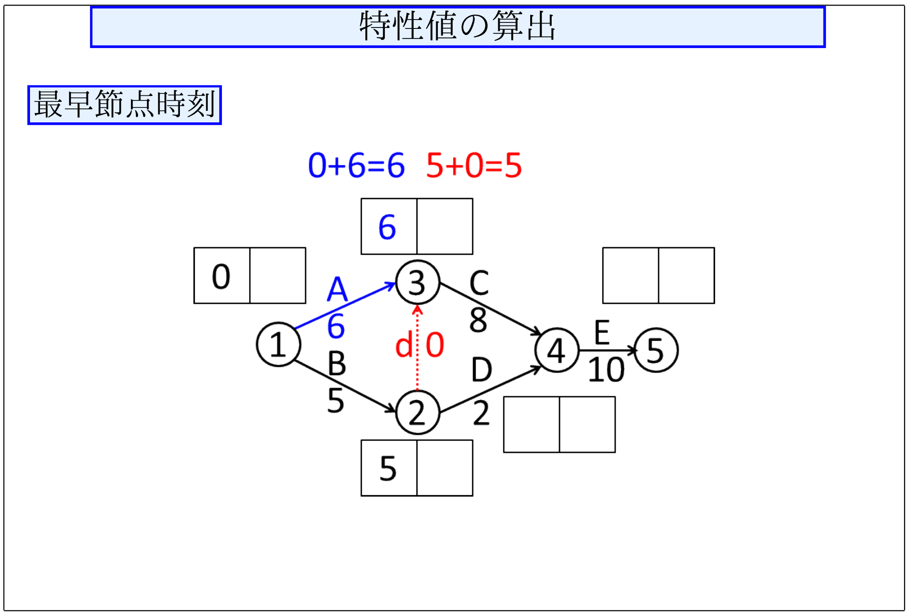
最早節点時刻



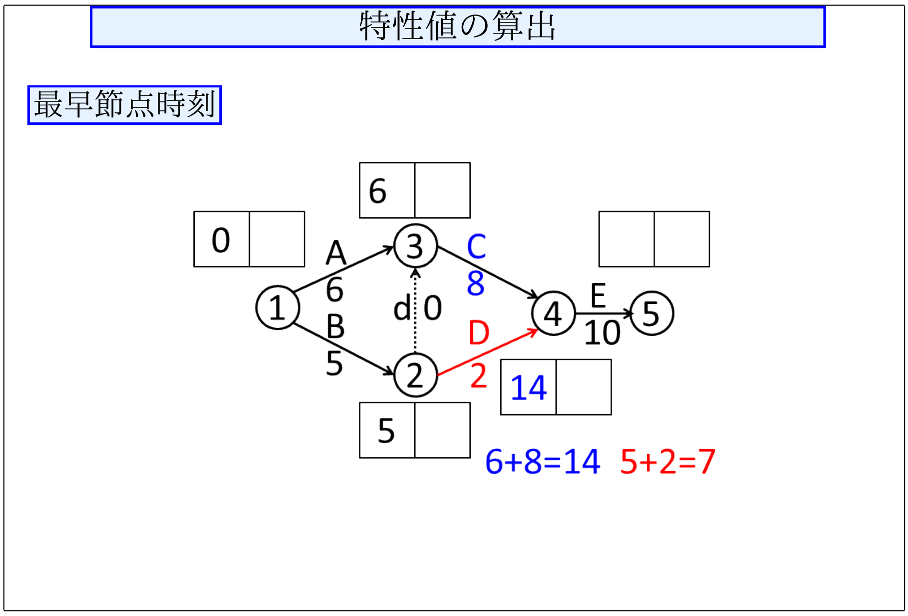
- 各点の最早節点時刻をトポロジカル順に計算していきます．
- 点はトポロジカル順に番号付けしましたので，点の番号順に計算していきます．
-



- 各点の最早節点時刻は，先行作業の開始点の最早節点時刻に，先行作業の所要時間を加えたものになります．
- このプロジェクトでは，点 2 は点 1 から作業 B の枝でつながっています．
- 点 1 の最早節点時刻は 0 分で，作業 B が 5 分かかることから，点 2 の最早節点時刻は，$0 + 5 = 5$ (分) となります．
-



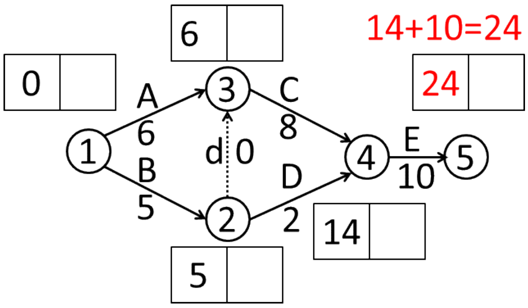
- 次に点 3 の最早節点時刻を算出します.
- 点 3 は点 1 から作業 A，点 2 からダミー作業スモール d でつながっています.
- 点 3 の最早節点時刻はこれらの作業がすべて終了した時刻になります.
- 点 1 の最早節点時刻は 0 分で，作業 A が 6 分かかることから，作業 A の終了時刻は $0 + 6 = 6$ (分) となります.
- また，点 2 の最早節点時刻は 5 分で，ダミー作業スモール d は所要時間 0 であることから，ダミー作業スモール d の終了時刻は，$5 + 0 = 5$ (分) となります.
- 点 3 の最早節点時刻はこれらの作業がすべて終了した時刻になりますから，6 分となります.
-



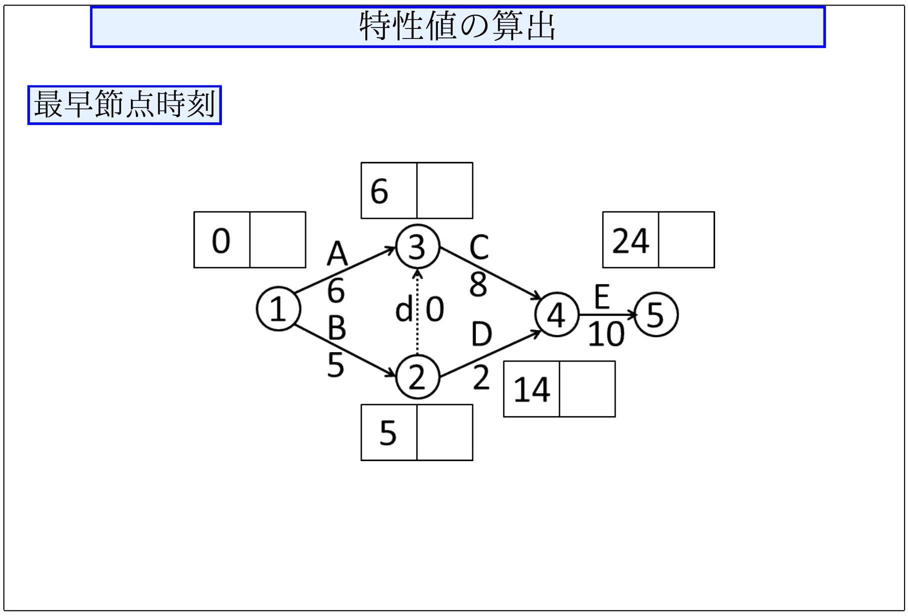
- 次に点 4 の最早節点時刻を算出します．
- 点 4 は点 3 から作業 C，点 2 から作業 D でつながっています．
- 点 4 の最早節点時刻はこれらの作業がすべて終了した時刻になります．
- 点 3 の節点開始時刻は 6 分で，作業 C が 8 分かかることから，作業 C の終了時刻は $6+8=14$(分) となります．
- また，点 2 の最早節点時刻は 5 分で，作業 D が 2 分かかることから，作業 D の終了時刻は，$5+2=7$(分) となります．
- 点 4 の最早節点時刻はこれらの作業がすべて終了した時刻になりますから，14 分となります．
-

特性値の算出

最早節点時刻



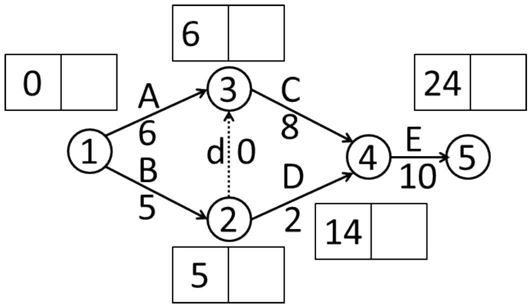
- 最後に点 5 の最早節点時刻を算出します.
- 点 5 は点 4 から作業 E の枝が繋がっています.
- 点 4 の最早節点時刻は 14 分で，作業 E が 10 分かかることから，点 5 の最早節点時刻は，$14 + 10 = 24$ (分) となります.



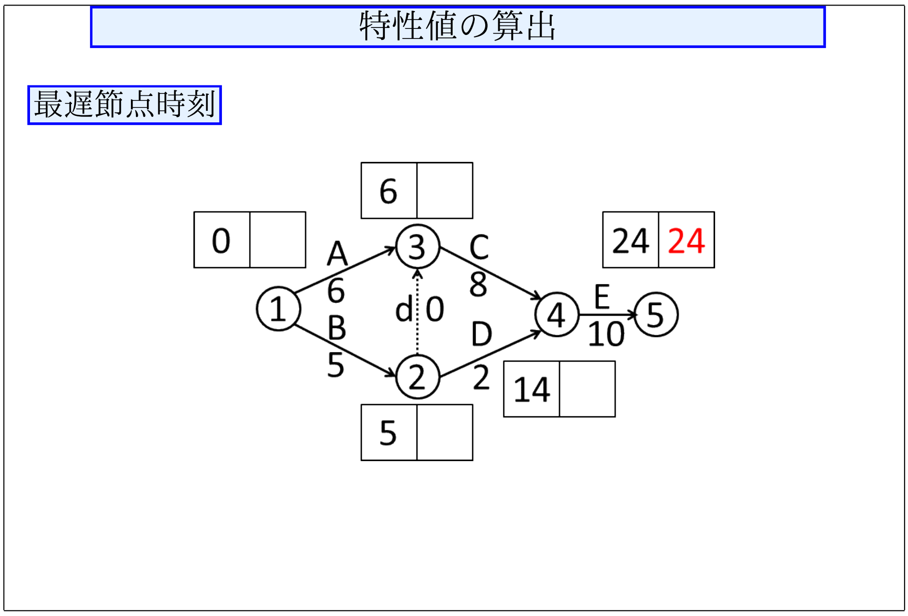
- 以上で，すべての点の最早節点時刻を算出しました．
- プロジェクトの終了点である点 5 の開始時刻，24 分は，プロジェクトが最早で終了する場合の，終了時刻を示しています．
-

特性値の算出

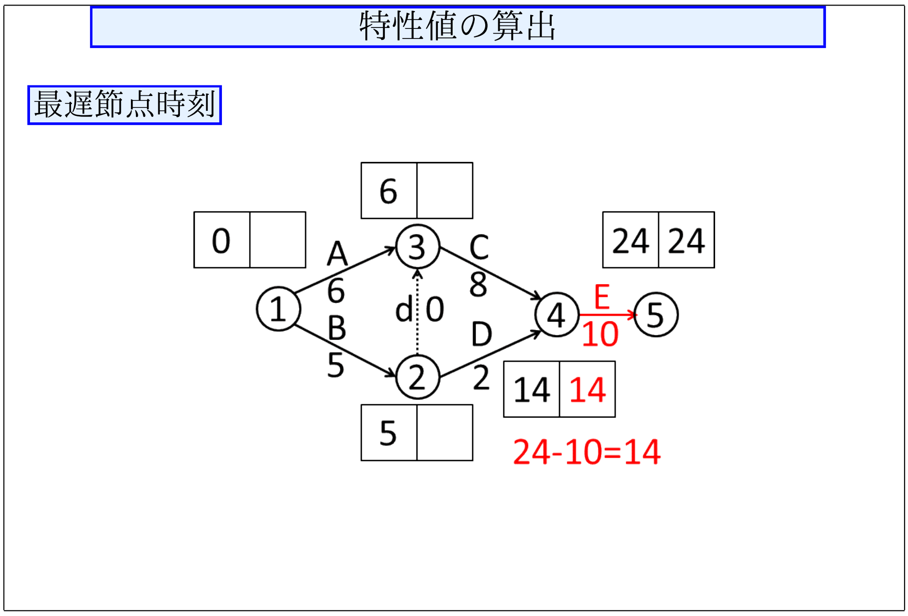
最遅節点時刻



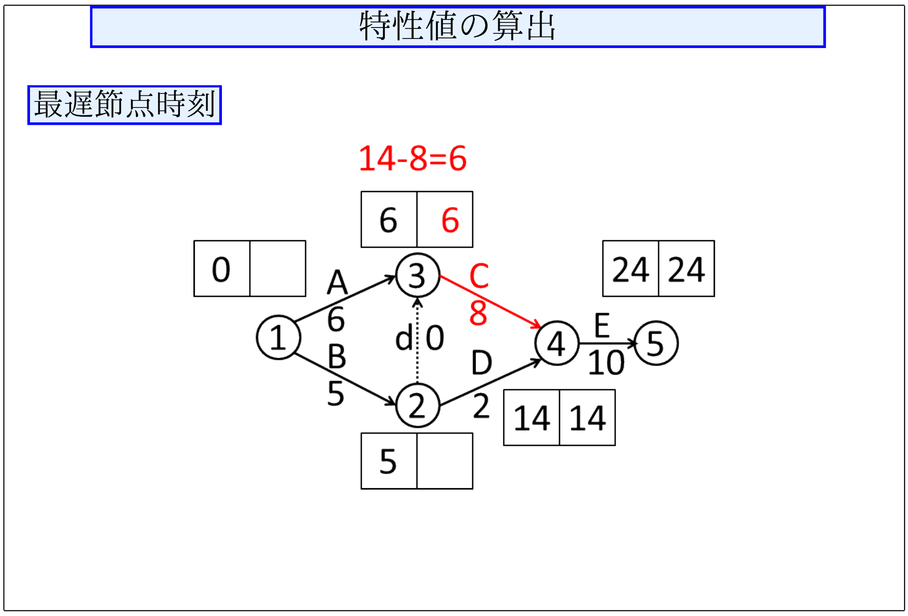
- 今度は，プロジェクトを最も早くで完了させるために，各点から開始する作業の少なくとも一つを開始すべき最も遅い時刻である,「最遅節点時刻」を求めます．



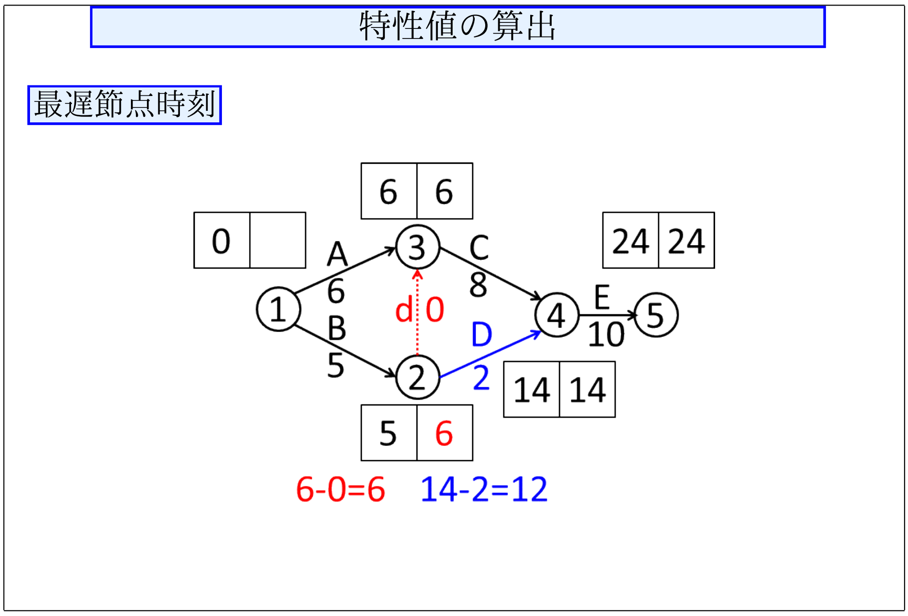
- 最遅節点時刻の算出はプロジェクトの終了点から始めます．
- 終了点の最遅節点時刻は最早節点時刻と同じです．
- なぜなら，終了点の最早節点時刻はプロジェクトの最早終了時刻ですから，終了点の最遅節点時刻が終了点の最早節点時刻より遅くなれば，プロジェクトの終了は遅れるからです．
- そこで，点 5 の最遅節点時刻を 24 分とします．以降は，トポロジカル順の逆に計算していきます．



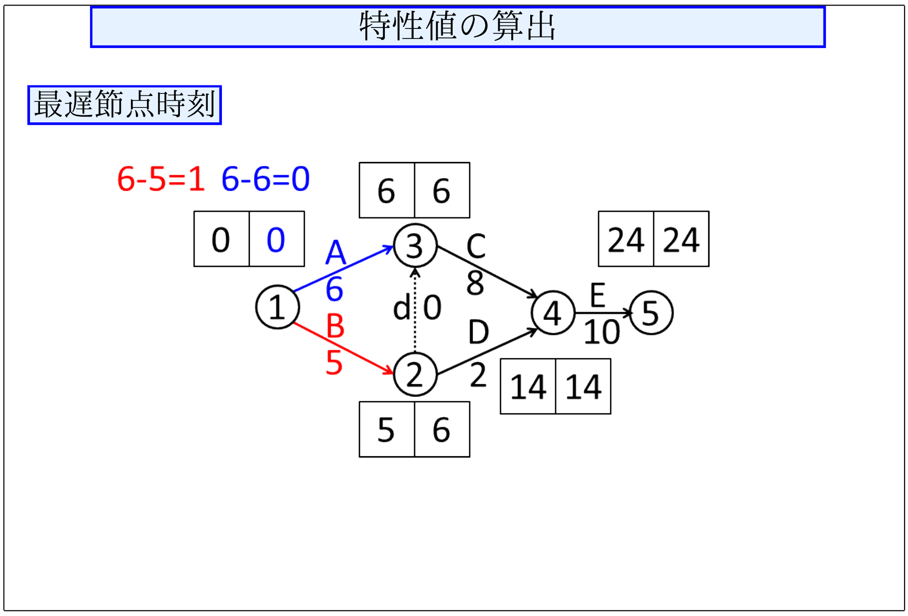
- 次に点 4 の最遅節点時刻を算出します．
- 点 4 から所要時間 10 分の作業 E が開始されています．
- 点 5 の最遅節点時刻 24 分に間に合わせるためには，作業 E は時刻 24 分には終了していなければなりません．
- したがって，点 4 の最遅節点時刻は $24 - 10 = 14$(分) となります．



- 次に点 3 の最遅節点時刻を算出します.
- 点 3 から所要時間 8 分の作業 C が開始されています.
- 点 4 の最遅節点時刻 14 分に間に合わせるためには，作業 C は 14 分には終了していなければなりません.
- したがって，点 3 の最遅節点時刻は $14 - 8 = 6$(分) となります.
-



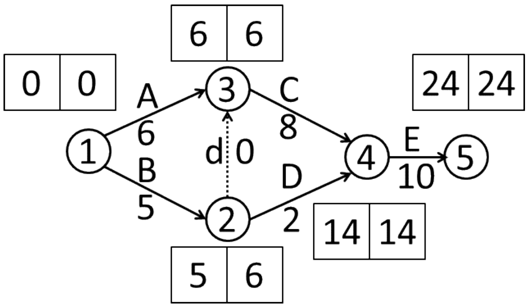
- 次に点 2 の最遅節点時刻を算出します．
- 点 2 からは，所要時間 0 のダミー作業スモール d と，所要時間 2 分の作業 D が開始されています．
- 点 2 の最遅節点時刻は，ダミー作業スモール d の終了点である点 3 の最遅節点時刻，作業 D の終了点である点 4 の最遅節点時刻の，両方に間に合う必要があります．
- ダミー作業スモール d の終了点である点 3 の最遅節点時刻は 6 分ですので，点 3 の最遅節点時刻 6 分に間に合わせるためには，ダミー作業スモール d は $6-0=6$(分) には開始していなければなりません．
- 一方，所要時間 2 分の作業 D の終了点である点 4 の最遅節点時刻は 14 分ですので，点 4 の最遅節点時刻 14 分に間に合わせるには，作業 D は $14-2=12$(分) には開始していなければなりません．
- これら 2 つの要請を同時に満たす，最遅節点時刻は 6 分となります．



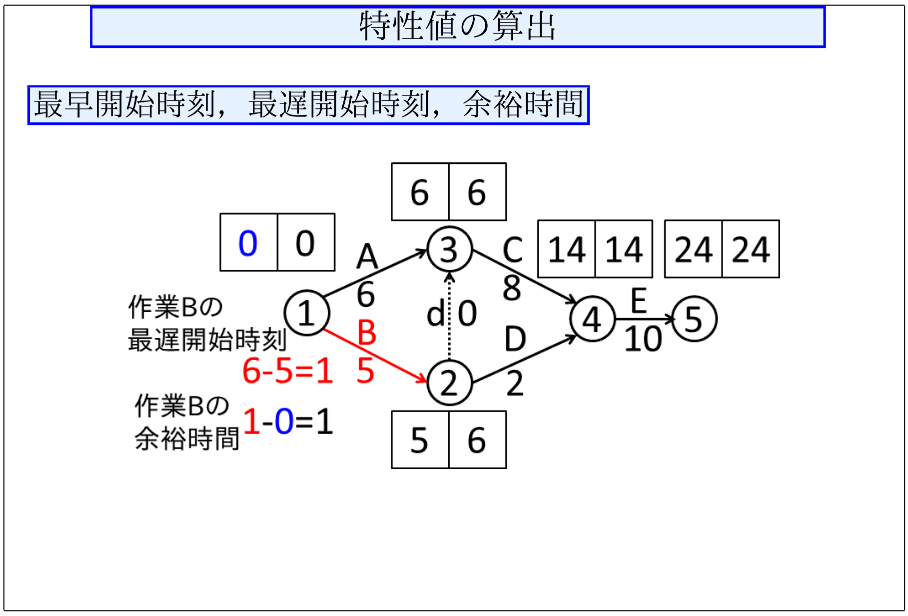
- 最後に点 1 の最遅節点時刻を算出します．
- 点 1 からは，所要時間 6 分の作業 A と，所要時間 5 分の作業 B が開始されています．
- 点 1 の最遅節点時刻は，作業 A の終了点である点 3 の最遅節点時刻，作業 B の終了点である点 2 の最遅節点時刻の，両方に間に合う必要があります．
- 作業 A の終了点である点 3 の最遅節点時刻は 6 分ですので，点 3 の最遅節点時刻 6 分に間に合わせるためには，作業 A は $6-6=0$(分) には開始していなければなりません．
- 一方，所要時間 5 分の作業 B の終了点である点 2 の最遅節点時刻は 6 分ですので，点 2 の最遅節点時刻 6 分に間に合わせるには，作業 B は $6-5=1$(分) には開始していなければなりません．
- これら 2 つの要請を同時に満たす，最遅節点時刻は 0 分となります．

特性値の算出

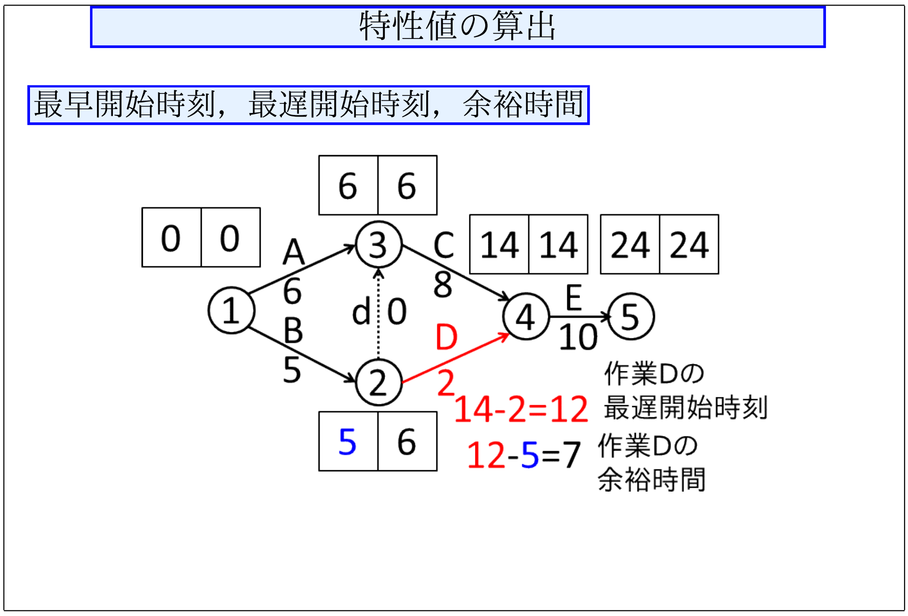
最早開始時刻，最遅開始時刻，余裕時間



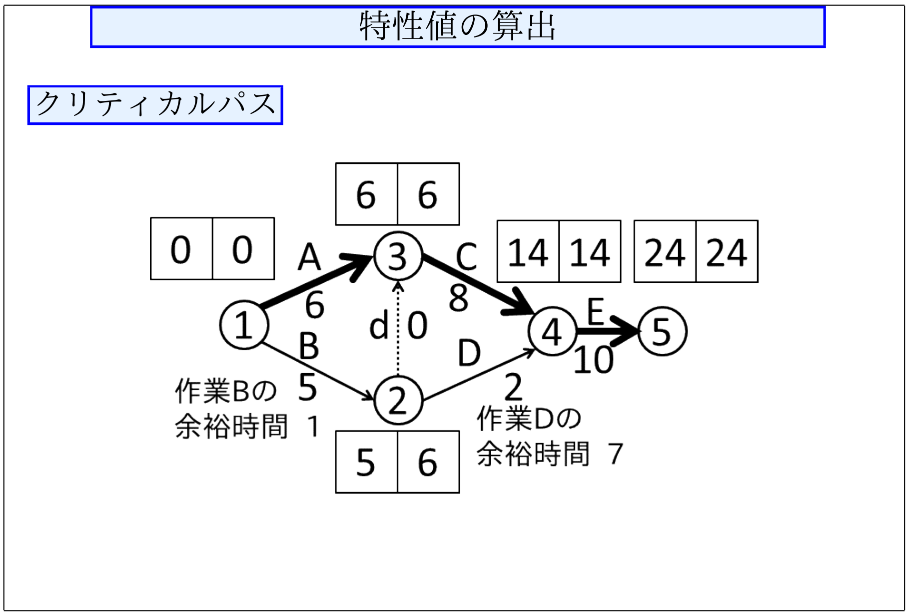
- 以上で各点の最遅節点時刻が算出されました．
- ここまで，点における最遅節点時刻を算出しましたが，作業によっては，さらに開始時刻を遅らせてもプロジェクトの終了時刻を遅らせないものもあります．
- そこで，今度は作業の最遅開始時刻を算出します．
- また，作業の最早開始時刻という指標もありますが，これは作業の開始点における最早節点時刻と同じです．



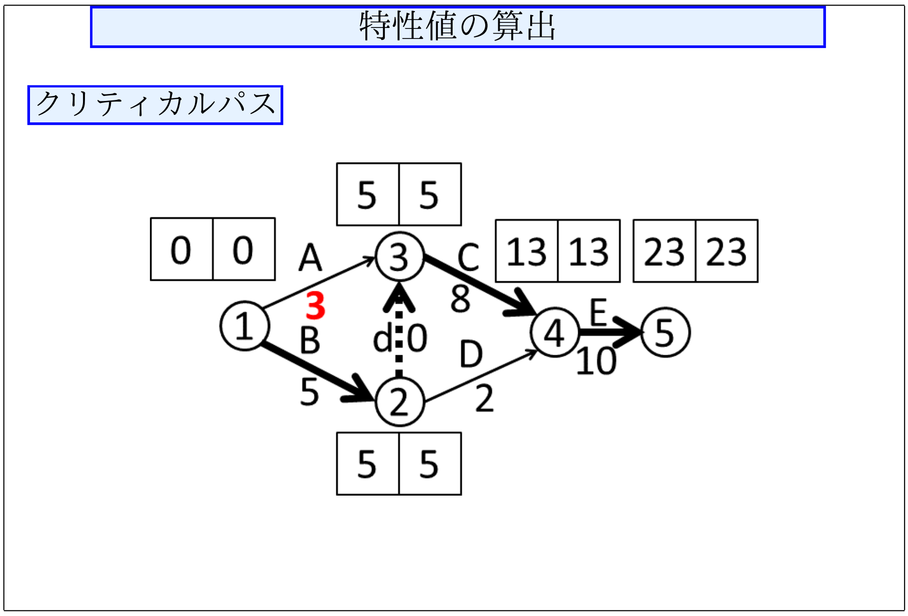
- まず，作業 B は所要時間が 5 分で，点 2 の最遅開始時刻 6 分までに終了させればいいことから，作業 B は $6-5=1$(分) に作業を開始すれば間に合います．
- したがって，作業 B の最遅開始時刻は 1 分となります．
- 作業 B の最早開始時刻は，作業 B の開始点 1 の最早節点時刻 0 分であることから，作業 B の最遅開始時刻には $1-0=1$(分) の余裕時間があります．



- また，作業 D は所要時間が 2 分で，点 4 の最遅開始時刻 14 分までに終了させればいいことから，作業 D は $14 - 2 = 12$(分) に作業を開始すれば間に合います．
- したがって，作業 D の最遅開始時刻は 12 分となります．
- 作業 D の最早開始時刻は 5 分であることから，作業 D の最遅開始時刻には $12 - 5 = 7$(分) の余裕時間があります．
- 作業 B, D 以外の作業は，最早開始時刻と最遅開始時刻が等しく，余裕時間はありません．
- これについては，各自で確認しておいて下さい．
-

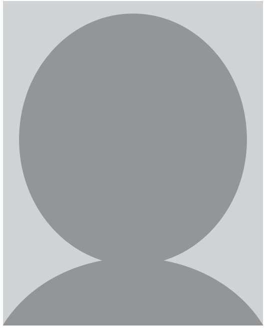


- プロジェクトの開始点から終了点に至る経路の中で，余裕時間のない作業からなる経路はクリティカルパスと呼ばれます．
- このプロジェクトにおけるクリティカルパスは，作業 A, C, E の経路です．
- クリティカルパス上の作業の所要時間の和は，プロジェクトの完了に要する時間と等しくなります．
- したがって，クリティカルパス上の作業に遅れが生じると，プロジェクトの終了時刻は遅れてしまいます．
- また，費用をかけてプロジェクトを最早完了時刻よりさらに早く終わらせるには，クリティカルパス上の作業の所要時間を短縮するようにしなければなりません．
- クリティカルパス上の作業の所要時間を短縮すると，クリティカルパスが変化することがあります．

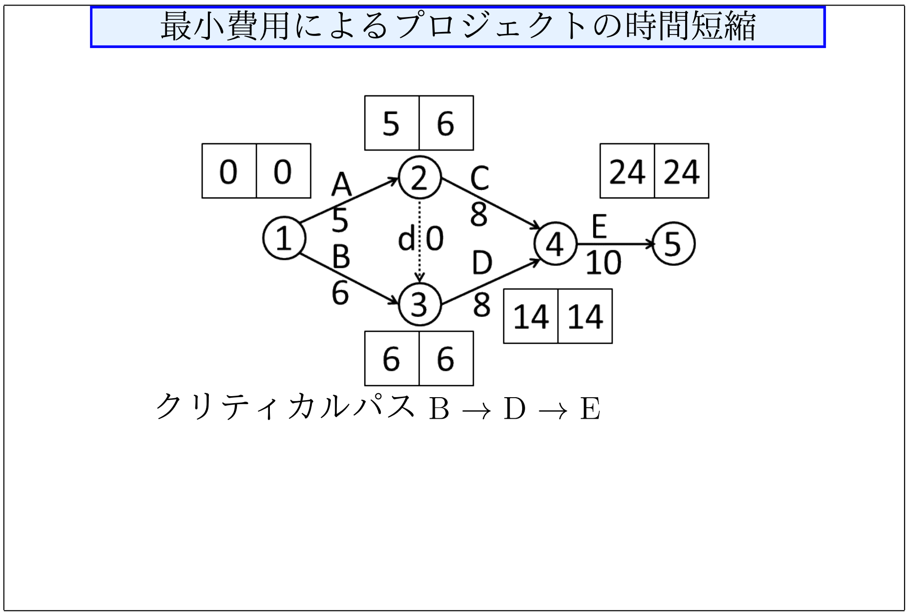


- 例えば，作業 A の所要時間を 6 分から 3 分に短縮したとします．
- すると，クリティカルパスは，作業 B, スモール d, C, E となります．
- プロジェクトの終了時刻は 23 分となり，1 分だけ短縮されます．
- 作業 A を 3 分短縮しても，全体では 1 分しか短縮されないので，あまり効率的ではありません．
- 例えば，作業 E は作業時間を短縮しただけ，プロジェクトの終了時刻が短縮されますので，効率的と言えます．
- ただし，作業により作業時間を 1 分短縮するためのコストが異なるのが一般的ですから，コストも含めて考える必要も出てきます．
- これについては，次にお話します．
-

最小費用によるプロジェクトの時間短縮



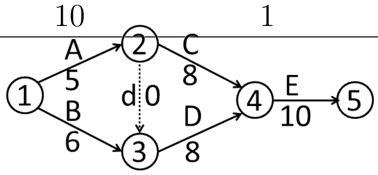
- ＣＰＭは，ＰＥＲＴとほぼ同時期に開発された，プロジェクトのスケジューリング手法です.
- 目的，方法はＰＥＲＴと類似していますが，時間短縮にかかる費用を考慮しているところに特徴があります.



- まず，時間短縮の策を施す前の，標準時間で作業を行う場合のプロジェクトのアロー・ダイアグラムを示します．
- このプロジェクトは完了まで 24 時間かかります．
- 各作業は費用をかけることにより作業時間を短縮することができます．
- 最小の費用でプロジェクト完了までの時間を 20 時間以内にするには，どの作業をどれだけ短縮すればよいか，という問題を考えます．

最小費用によるプロジェクトの時間短縮

記号	先行作業	標準時間 (時間)	短縮可能時間 (時間)	費用/時間 (万円/時間)
A	-	5	2	2
B	-	6	2	3
C	A	8	2	5
D	A, B	8	1	1
E	C, D	10	1	4



- 作業時間は無制限に短縮できるわけではありません．
- 表の「短縮可能時間」の列に最大何時間短縮できるかが記されています．
- 作業 A, B, C は最大 2 時間，作業 D, E は最大 1 時間短縮可能です．
- 作業時間を 1 時間短縮するためにかかる費用も作業ごとに異なります．
- 表の費用パー時間の列に，作業時間を 1 時間短縮するのにかかる費用が記されています．
- ただし，短縮時間は 1 時間毎とします．
- 0.5 時間とか 1.5 時間の短縮は認められないとします．

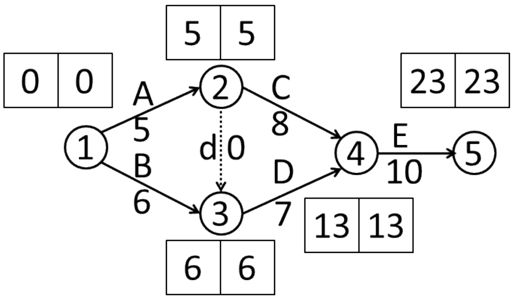
最小費用によるプロジェクトの時間短縮

記号	先行作業	標準時間 (時間)	短縮可能時間 (時間)	費用/時間 (万円/時間)
B	-	6	2	3
D	A, B	8	1	1
E	C, D	10	1	4

- プロジェクト完了までの時間を短縮可能な作業の組み合わせ … カット
  ({B}, 3), ({D}, 1), ({E}, 4)
- 費用が最小なカット … 最小カット ({D}, 1)

- 先ほども説明しましたように，プロジェクト完了までの時間を短縮するには，クリティカルパス上にある作業の作業時間を短縮する必要があります.
- クリティカルパス上にある短縮可能な作業のうち，時間当たりの費用が最小の作業を短縮することが基本になります.
- ただし，1 時間作業を短縮すると，別の作業の系列が新たにクリティカルパスに加わることがあります.
- 複数のクリティカルパスが存在する場合には，同時に全てのクリティカルパスにおいて，作業時間を短縮しなければ，プロジェクト完了までの時間を短縮することができません.
- そこで，1 時間作業を短縮するたびに，最早節点時刻等の指標を更新して，次に短縮すべき作業を探します.
- クリティカルパス上の作業時間を短縮可能な作業の組み合わせのことをカットと呼びます.
- 現在のクリティカルパスは B → D → E で，これらの作業はいずれも短縮可能ですから，カットは B, D, E の 3 つです.
- 作業名と組になっている数字は，その作業を 1 時間短縮するのに必要な経費です.
- カットのうち，費用が最小な最小カットの作業時間を 1 時間短縮して，最早節点時刻等の指標を更新することを繰り返すことにより最小費用でプロジェクト完了までの時間を短縮することができます.
- ここでは，作業 D の時間短縮のための費用が最小なので，1 万円の費用を払って作業 D を 1 時間短縮します.

最小費用によるプロジェクトの時間短縮

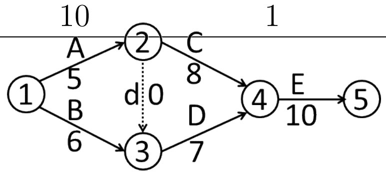


クリティカルパス B → D → E,　A → C → E

- その結果，プロジェクト完了までの時間は 1 時間短縮され，23 時間になりました．
- 指標を計算し直すと，図のようになり，クリティカルパスは B → D → E に加え，A → C → E もクリティカルパスとなりました．

最小費用によるプロジェクトの時間短縮

記号	先行作業	作業時間 (時間)	短縮可能時間 (時間)	費用/時間 (万円/時間)
A	-	5	2	2
B	-	6	2	3
C	A	8	2	5
D	A, B	7	0	-
E	C, D	10	1	4



- 現在の作業時間と短縮可能時間を示します.
- 作業 D は短縮可能時間を使い切りましたので，これ以上短縮することはできません.

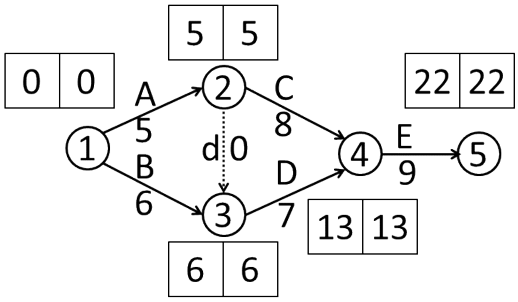
最小費用によるプロジェクトの時間短縮

記号	先行作業	作業時間 (時間)	短縮可能時間 (時間)	費用/時間 (万円/時間)
A	-	5	2	2
B	-	6	2	3
C	A	8	2	5
D	A, B	7	0	-
E	C, D	10	1	4

- クリティカルパス　$B \to D \to E$, $A \to C \to E$,
- カット　$(\{B, A\}, 3+2)$, $(\{B, C\}, 3+5)$, $(\{E\}, 4)$
- 最小カット　$(\{E\}, 4)$

- クリティカルパスが　$B \to D \to E$，と $A \to C \to E$，の 2 つありますので，両方のクリティカルパスの作業時間を短縮しないと，プロジェクト完了までの時間を短縮することはできません．
- そこで，カットは，$\{B, A\}$, $\{B, C\}$, $\{E\}$ の 3 つとなります．
- クリティカルパス　$B \to D \to E$　上の作業と，クリティカルパス　$A \to C \to E$ 上の作業が組み合わされています．
- 作業 D は短縮不可能なので，カットに含まれません．
- また，作業 E は両方のクリティカルパスに含まれますので，作業 E の作業時間を短縮すれば，プロジェクト完了までの時間を短縮できます．
- 最小カットは E なので，費用 4 万円をかけて作業 E を 1 時間短縮します．

最小費用によるプロジェクトの時間短縮

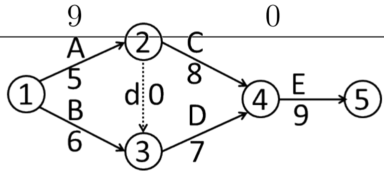


クリティカルパス $\mathrm{B} \to \mathrm{D} \to \mathrm{E}$, $\mathrm{A} \to \mathrm{C} \to \mathrm{E}$

- その結果，プロジェクト完了までの時間は 1 時間短縮され，22 時間になりました．
- クリティカルパスには変化がありません．

最小費用によるプロジェクトの時間短縮

記号	先行作業	作業時間 (時間)	短縮可能時間 (時間)	費用/時間 (万円/時間)
A	-	5	2	2
B	-	6	2	3
C	A	8	2	5
D	A, B	7	0	-
E	C, D	9	0	-



- 現在の作業時間と短縮可能時間を示します.
- 作業Eも短縮可能時間を使い切りましたので，これ以上短縮することはできません.

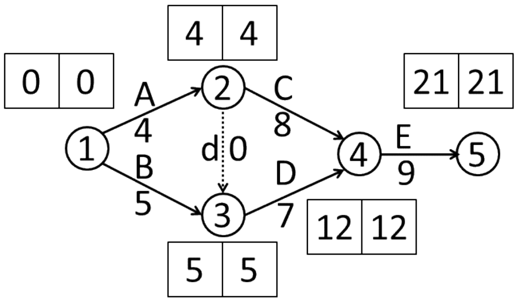
最小費用によるプロジェクトの時間短縮

記号	先行作業	作業時間 (時間)	短縮可能時間 (時間)	費用/時間 (万円/時間)
A	-	5	2	2
B	-	6	2	3
C	A	8	2	5
D	A, B	7	0	-
E	C, D	9	0	-

- クリティカルパス　B $\to$ D $\to$ E, A $\to$ C $\to$ E,
- カット　({B, A}, 3 + 2), ({B, C}, 3 + 5)
- 最小カット　({B, A}, 5)

- クリティカルパスが　B $\to$ D $\to$ E，と A $\to$ C $\to$ E，の 2 つありますので，カットは，{B, A} と {B, C}, となります．
- 最小カットは {B, A} なので，費用 5 万円をかけて作業 B と A をそれぞれ 1 時間短縮します．

最小費用によるプロジェクトの時間短縮

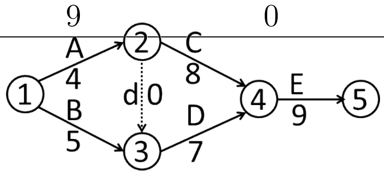


クリティカルパス B → D → E,　A → C → E

- その結果，プロジェクト完了までの時間は 1 時間短縮され，21 時間になりました．
- クリティカルパスには変化がありません．

最小費用によるプロジェクトの時間短縮

記号	先行作業	作業時間 (時間)	短縮可能時間 (時間)	費用/時間 (万円/時間)
A	-	4	1	2
B	-	5	1	3
C	A	8	2	5
D	A, B	7	0	-
E	C, D	9	0	-



- 現在の作業時間と短縮可能時間を示します.
- 作業 A, B, C はさらに時間短縮が可能です.

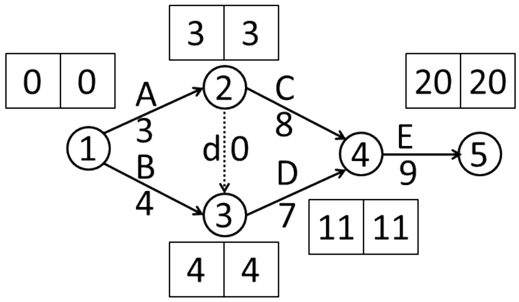
最小費用によるプロジェクトの時間短縮

記号	先行作業	作業時間 (時間)	短縮可能時間 (時間)	費用/時間 (万円/時間)
A	-	4	1	2
B	-	5	1	3
C	A	8	2	5
D	A, B	7	0	-
E	C, D	9	0	-

- クリティカルパス　B $\rightarrow$ D $\rightarrow$ E, A $\rightarrow$ C $\rightarrow$ E,
- カット　({B, A}, 3 + 2), ({B, C}, 3 + 5)
- 最小カット　({B, A}, 5)

- カットは，{B, A} と {B, C} となります.
- 最小カットは ({B, A}, 3 + 2) なので，さらに費用 5 万円をかけて作業 B と A をそれぞれ 1 時間短縮します.

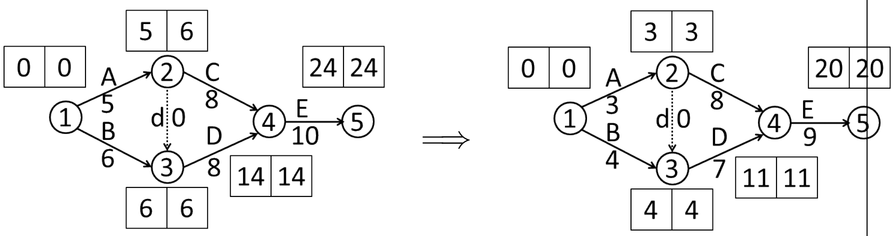
最小費用によるプロジェクトの時間短縮



クリティカルパス B → D → E, A → C → E

- その結果，プロジェクト完了までの時間は 1 時間短縮され，20 時間になりました．
- これで目標達成という訳です．
- ちなみに，さらに短縮可能な作業は C のみで，C の作業時間を短縮しても，プロジェクト完了までの時間は短縮できません．

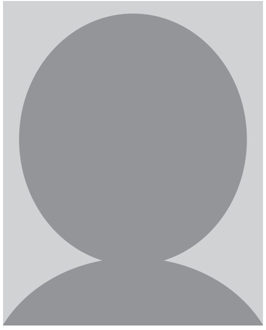
最小費用によるプロジェクトの時間短縮



- ({D}, 1), ({E}, 4), ({B, A}, 3 + 2) , ({B, A}, 3 + 2) の作業を費用をかけて短縮
- 費用は $1 + 4 + 5 + 5 = 15$ (万円)

- 以上の一連の時間短縮で，標準時間では 24 時間かかっていたプロジェクトが 20 時間で完了するようになりました．
- 時間短縮に要した費用は合計 15 万円でした．

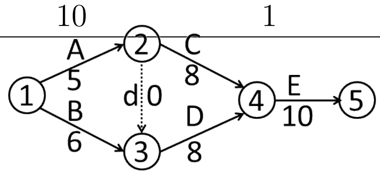
線形最適化問題としての定式化



- 今度は，作業の短縮時間に連続値を許し，最小費用でプロジェクト完了までの時間を短縮することを考えます．
- この問題を線形最適化問題として定式化を行います．

線形最適化問題としての定式化

記号	先行作業	標準時間 (時間)	短縮可能時間 (時間)	費用/時間 (万円/時間)
A	-	5	2	2
B	-	6	2	3
C	A	8	2	5
D	A, B	8	1	1
E	C, D	10	1	4



- 先ほどと同じプロジェクトを例にします.
- 最小の費用で，プロジェクト完了までの時間を 24 時間から 20 時間に短縮します.

線形最適化問題としての定式化

- 点 $i \in \{1,2,3,4,5\}$ の最早節点時刻を x_i
- 作業 $j \in \{A,B,C,D,E\}$ の短縮時間を s_j
- 短縮費用の合計 z (万円) は

$$z = \sum_j s_j = 2s_A + 3s_B + 5s_C + s_D + 4s_E$$

- 点 i の最早節点時刻を x_i とします．
- また，作業 j の短縮時間を s_j とします．
- 時間短縮の費用の合計 z は，1 時間当たりの短縮費用と短縮時間 s_j の積の和となります．
- z が目的関数で，x_i と s_j が決定変数です．

線形最適化問題としての定式化

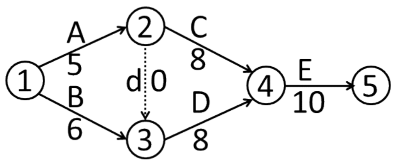
点 1 の最早節点時刻

$$x_1 = 0$$

点 2 の最早節点時刻

- 作業 A の作業時間が $5 - s_A$

$$x_2 = x_1 + (5 - s_A)$$



- 制約条件を定式化していきますが，最早節点時刻を求めるのと同じ要領です.
- まず，点 1 の最早節点時刻は 0 ですから，$x_1 = 0$ となります.
- 点 2 の最早節点時刻は，作業時間が $5 - s_A$ ですので，このような式になります.

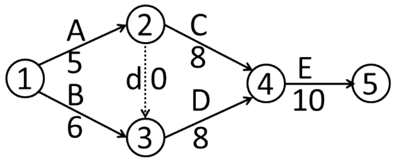
線形最適化問題としての定式化

点 3 の最早節点時刻

$$x_3 = \max\{x_2, x_1 + (6 - s_B)\}$$

$$\Downarrow$$

$$x_3 \geq x_2,\ x_3 \geq x_1 + (6 - s_B)$$



- 点 3 の最早節点時刻は，作業 A, B の終了後ですから，この式のようになります．
- しかし，これでは線形最適化問題にならないので，これら 2 つの不等式で表します．
- max の式と 2 つの不等式は，これだけでは等価ではありませんが，z の最小化との組み合わせで，等価な制約として機能します．
- これは線形最適化法を適用するために，よく使われるテクニックです．
- この科目内でも，印刷教材の第 8 章でこのテクニックが出てきます．

線形最適化問題としての定式化

点 4 の最早節点時刻

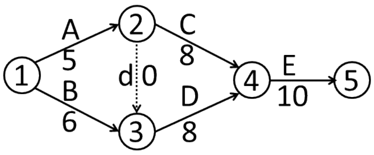
$$x_4 \geq x_2 + (8 - s_C),\ x_4 \geq x_3 + (8 - s_D)$$

点 5 の最早節点時刻

$$x_5 = x_4 + (10 - s_E)$$

プロジェクトは 20 時間以内で完了

$$x_5 \leq 20$$



- 点 4 の最早節点時刻についても，同様に，2 つの不等式で表します．
- 点 5 の最早節点時刻は，点 4 の最早節点時刻が x_4，作業 E の作業時間が $10-s_E$ であることから，この式になります．
- プロジェクトは 20 時間以内で完了する必要があることから，$x_5 \leq 20$ とします．

線形最適化問題としての定式化

$$
\begin{array}{rl}
\text{最小化} & z = 2s_A + 3s_B + 5s_C + s_D + 4s_E \\
\text{制約条件} & x_1 = 0 \\
& x_2 = x_1 + (5 - s_A) \\
& x_3 \geq x_2,\ x_3 \geq x_1 + (6 - s_B) \\
& x_4 \geq x_2 + (8 - s_C),\ x_4 \geq x_3 + (8 - s_D) \\
& x_5 = x_4 + (10 - s_E) \\
& x_5 \leq 20 \\
& 0 \leq s_A \leq 2,\ 0 \leq s_B \leq 2,\ 0 \leq s_C \leq 2, \\
& 0 \leq s_D \leq 1,\ 0 \leq s_E \leq 1
\end{array}
$$

- さらに，s_j に関して短縮可能時間の制約を記述することで，この問題を線形最適化問題として定式化することができました．
- ここでは，作業の短縮時間に連続値を許しましたが，整数値のみを許す場合は，さらに s_j に整数であるという制約条件を加えれば，整数最適化問題として定式化することができます．