授業準拠

経営科学

福山平成大学経営学部

福井　正康

はじめに

著者が福山大学・福山平成大学の経営系学科に赴任して３０年経ちました。その間、数理的に物事を見る目を養ってもらうにはどのように授業を進めるべきか、4 年生で分析的な卒業論文を書くためには何が必要か、いろいろ考えてみました。その結果生まれたのが C.Analysis (College Analysis) です。誰でも自由に使えて、初心者でも使い易い、それがこのソフト開発のコンセプトです。1997 年から C.Analysis を作り始めて、様々な分野の分析手法を加え、今では授業で困ることはほぼなくなりました。また、ネットを通じて多くの方からご意見をいただき、一般的な使用目的でもかなり使えるようになってきました。

このテキストは、福山平成大学の授業用に作ったものですが、統計の初心者の方が一通り気軽に勉強してみたいと思われたときには良い教材になると思います。このテキストの中には Youtube のビデオ教材へのリンクが含まれています。学生さんはそれを携帯で動かし、例題や問題をやっているようです。

このテキストは本当に初心者用なので、本気でやりたい方には不満足かも知れません。もっと数学的に知りたい、もっとソフトの機能を使いたいなど、ご希望がありましたら、総合マニュアルを当たってみて下さい。こちらには著者のつたない知識のすべてが含まれており、日々更新されています。これらが少しでも役に立てたら幸いです。

倉敷にて　福井正康

１．線形計画法　【動画】

1.1　LP の例

経営についての数学モデルで、ある条件のもとに、最大の利益を探る問題を数理計画問題といいます。特に利益を表す式と条件を表す式がいくつかの変数の 1 次式で表される場合を線形計画問題といいます。また、その問題を解く手法のことを線形計画法と呼びます。

簡単な例として、ある工場で２つの製品 1 と 2 を作っている場合を考えます。その製品を作るには 3 つの原料Ａ，Ｂ，Ｃを使用するとして、原料の供給量には上限があるとします。また、それぞれの製品を 1 単位（個でもよい）作るための原料にも必要量があり、それらは表 1 のようになっているとします。

表 1　製品と原料

製品 / 原料	１（x1）	２（x2）	原料の供給量（kg）
A	1	3	６０
B	3	4	１００
C	2	1	５０
製品 1 単位当りの利益（万円）	5	6	最大化

製品 1 を 1 単位作るには原料Ａを 1kg、原料Ｂを 3kg、原料Ｃを 2kg 使い、製品 2 を 1 単位作るには原料Ａを 3kg、原料Ｂを 4kg、原料Ｃを 1kg 使うとします。

原料の供給量の上限はＡが 60kg、Ｂが 100kg、Ｃが 50kg です。製品を作ると 1 単位作るごとに、製品 1 では 5 万円、製品 2 では 6 万円の利益が出るものとします。最大の利益をあげるためには製品 1 と 2 をいくらずつ作ればよいでしょうか。

これは条件の付いた利益最大化の問題です。

利益を表す式を目的関数、条件を表す式を制約条件といい、x_1 と x_2 をそれぞれ製品 1 と製品 2 の製造量として、この問題を式で表すと以下のようになります。

目的関数（objective function）

$z = 5x_1 + 6x_2$　　最大化　　利益が 5 万円と 6 万円

制約条件（constraints）

$x_1 + 3x_2 \le 60$

$3x_1 + 4x_2 \le 100$　　上から原料 A,B,C の制約、製造量は非負

$2x_1 + x_2 \le 50$

$x_1, x_2 \ge 0$

目的関数も制約条件もすべて 1 次式なので、この問題は線形計画問題と呼ばれます。

College Analysis を用いてこの問題を解いてみることにしましょう。ここでは問題を解くためのプログラムの使い方を簡単に説明します。メニュー［分析－ＯＲ－線形計画法」を選択すると、以下に示される分析実行画面が表示されます。。

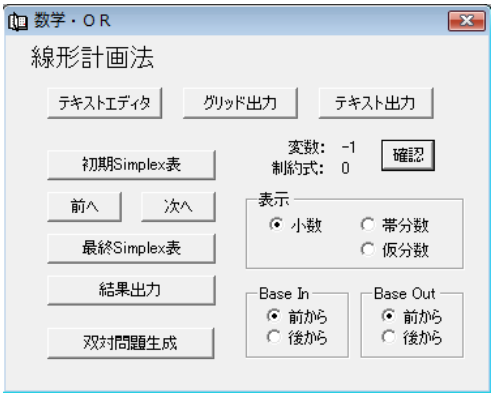


図１　線形計画法の実行画面

プログラムの実行には、グリッド（表入力画面）に書かれた表１によく似た形式のデータが必要ですが、「テキストエディタ」に式を書き込み、それをグリッドの形式に変換する方法もあります。ここでは後者の方法を採用し、まず「テキストエディタ」ボタンをクリックします。表示されたメモ帳のようなところに、以下のように半角英数で式を書きこみます。上の数式から意味は理解できると思います。

z=5*x1+6*x2　max

x1+3*x2<=60

3*x1+4*x2<=100

2*x1+x2<=50

x1, x2>=0

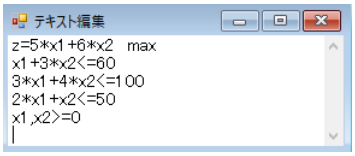


図２　テキスト入力

一番上が目的関数で「max」の前は１つ空欄を入れて下さい。２番目以降は制約条件です。「≦」は「<=」のように打ち込みます。「*」記号などを忘れないようにして下さい。

打ち終わったら、実行画面の「グリッド出力」ボタンをクリックします。そうすると、正しく打てていると以下のようなデータになります。これ以外だと、テキストの入力のどこかにミスがあります。訂正して再度「グリッド出力」ボタンをクリックします。

図３　グリッドデータ

正しく打てているようなら、「結果出力」ボタンをクリックします。以下のような計算結果が表示されます。

LP 最適解

	STEP数	最大値(Z)	
▶	3	160.000	
変数	値	被約費用	
x1	20.000	0.000	
x2	10.000	0.000	
行	スラック	双対価格	
1	10.000	0.000	
2	0.000	1.400	
3	0.000	0.400	
係数範囲			
変数	現在値	減少下限	増加上限
x1	5.000	-0.500	7.000
x2	6.000	-3.500	0.667
右辺範囲			

図４　計算結果

この中で最も重要なところは、黄色に網掛けされた部分で、目的関数の最大値（右上）と、それを実現する変数の値（左側）が表示されています。結果を以下にまとめておきます。

表２　最適解

x1	x2	Z
20	10	160

ここでは、答えを一気に出しましたが、コンピュータは段階を追って目的関数を大きくしていき、最終的に一番大きくなるところでそれを答え（最適解といいます）としています。その過程を見て行きましょう。

図３の段階ですぐに「結果出力」ボタンをクリックせずに、「初期シンプレックス表」ボタンをクリックします。すると以下のような表が表示されます。

シンプレックス表 STEP 0 (No. 0)

	x1	x2	SL1	SL2	SL3		
▶ <Z>	-5	-6	0	0	0	=	0
SL1	1	3	1	0	0	=	60
SL2	3	4	0	1	0	=	100
SL3	2	1	0	0	1	=	50

図５　初期シンプレックス表

これは、コンピュータが計算をしていくための最初の画面です。このくらいだったら手計算で以後の動きを追っていけるのですが、特に必要ないと思いますので省略します。一番上の行の右端の「0」は最初の目的関数 z の値です。ここに注目しながら、「次へ」ボタンをクリックします。すると次のような画面になります。

シンプレックス表 STEP 1 (No. 1)

	x1	x2	SL1	SL2	SL3		
▶ <Z>	-3	0	2	0	0	=	120
x2	0.333333	1	0.333333	0	0	=	20
SL2	1.666667	0	-1.333333	1	0	=	20
SL3	1.666667	0	-0.333333	0	1	=	30

図６　シンプレックス表（step 1）

これを見ると、右上の目的関数の値は 120 になっています。同じく、「次へ」ボタンをクリックすると次の画面になります。

シンプレックス表 STEP 2 (No. 2)

	x1	x2	SL1	SL2	SL3		
＜Z＞	0	0	-0.4	1.8	0	=	156
x2	0	1	0.6	-0.2	0	=	16
x1	1	0	-0.8	0.6	0	=	12
SL3	0	0	1	-1	1	=	10

図７　シンプレックス表（step 2）

右上の目的関数の値が 156 に上がりました。再度「次へ」をクリックします。すると今度は以下となります。

最終シンプレックス表 STEP 3 (No. 3)

	x1	x2	SL1	SL2	SL3		
＜Z＞	0	0	0	1.4	0.4	=	160
x2	0	1	0	0.4	-0.6	=	10
x1	1	0	0	-0.2	0.8	=	20
SL1	0	0	1	-1	1	=	10

図８　最終シンプレックス表（step 3）

これが最終ですが、目的関数の値は 160 になりました。これ以上上げることはできません。そして、答えは２行目と３行目を見て下さい。２行目の左端 x2 の値が右端の 10 です。３行目の左端 x1 の値が 20 になります。

このようにコンピュータは段階的に答えに近づいて行っているのです。このような方法をシンプレックス法と言って、後に説明する２段階シンプレックス法と合わせて、ノーベル賞を取った方法なのです。

問題１

以下の線形計画問題の最適解を求めよ。

目的関数

$z = x_1 + 2x_2 + x_3$　最大化

制約条件

$x_1 + 3x_2 + 2x_3 \le 600$

$2x_1 + x_2 + 2x_3 \le 300$

$x_1, x_2, x_3 \ge 0$

最適解

x1	x2	x3	Z

問題２

製品１と２の製造のためには３つの原料Ａ，Ｂ，Ｃが必要である。各製品１単位製造するために必要な各原料の量（kg）、各原料の供給量の上限（kg）及び、各製品１単位生産するごとの利益（万円）は以下の表の通りである。原料の供給量の範囲内で、利益が最大となる各製品の生産量（単位）はいくらか。解答は少数で表せ。

	製品１	製品２	原料供給量上限
原料Ａ	1.5	3.2	60
原料Ｂ	3.4	4.3	100
原料Ｃ	2.1	1.4	50
利益（万円）	5.2	6.3	

目的関数

制約条件

最適解

製品１	製品２	利益

問題１解答

最適解

x1	x2	x3	Z
60	180	0	420

問題２解答

目的関数

z=5.2*x1+6.3*x2　max

制約条件（入力形式）

1.5*x1+3.2*x2<=60

3.4*x1+4.3*x2<=100

2.1*x1+1.4*x2<=50

x1,x2>=0

最適解

製品１	製品２	利益
17.564	9.368	150.351

1.2 等号制約の線形計画問題　【動画】

標準的な線形計画問題とは？

前回は標準的な線形計画問題と呼ばれるものを解きました。標準的な線形計画問題とは以下の条件をすべて満たしているものをいいます。

1）最大化問題である

2）制約条件の右辺が非負（0 以上）

3）制約条件の変数が非負（0 以上）

4）不等号の向きが「≦」（変数の符号は除きます）

前回の例で見てみましょう。

目的関数

$z = 5x_1 + 6x_2$　　最大化

1）これは最大化問題を満たしています。

制約条件

$x_1 + 3x_2 \le 60$

$3x_1 + 4x_2 \le 100$

$2x_1 + x_2 \le 50$

$x_1, x_2 \ge 0$

2）右辺はすべて 0 以上ですので満たしています。

3）最後の条件から、変数は 0 以上ですので満たしています。

4）変数の符号を除いて、不等号の向きも「≦」ですので満たしています。

以上のことから、前回の問題は標準的な線形計画問題であるといえます。

等号制約の線形計画問題

標準的な線形計画問題には含まれないような問題で、特に重要なのが、制約式の不等号「≦」が等号になった、等号制約の線形計画問題です。以下に例を示します。

例

目的関数

$z = 3x_1 + 2x_2 + x_3$　　最大化

制約条件

$x_1 + 2x_2 + 2x_3 = 60$

$x_1 + 4x_2 + x_3 = 40$

$x_1, x_2, x_3 \ge 0$

この問題では標準的な線形計画法のような単純なシンプレックス法は使えません。標準的な問題を解くシンプレックス法を 1 段階シンプレックス法と呼びます。今回の等号制約の問題に対して考え出された手法は 2 段階シンプレックス法というものです。これは文字通りシンプレックス法を 2 回適用して解を求める方法です。1 段階目は問題を標準的なシン

プレックス法に書き換えるための段階で、2 段階目で解を求めて行きます。この方法が一般的な線形計画問題を解く鍵となります。2 段階シンプレックス法について詳細は省略します。

上の例を C.Analysis で解いてみましょう。入力は「テキストエディタ」を開いて、式を入力します。その後「グリッド出力」すると、以下の図のようになります。

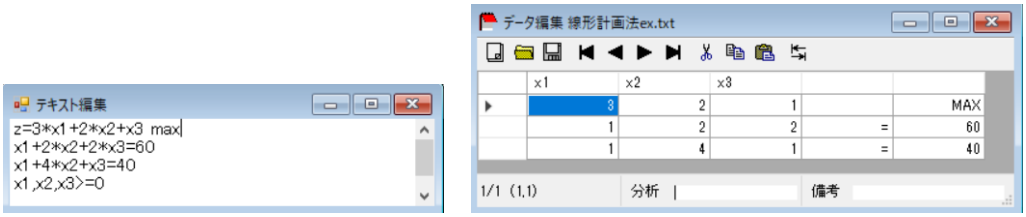


図 1　データの入力

この答えをすぐ求めることもできますが、計算の途中を示すために、「初期シンプレックス表」から「次へ」ボタンを押していくと、以下のように、解法の途中から 2 段階目に移ることが分かります。詳細は気にする必要はありません。

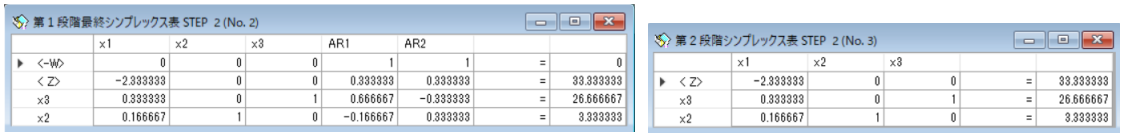


図 2　2 段階シンプレックス法

最終的に「結果出力」ボタンをクリックすると、以下のような結果になります。

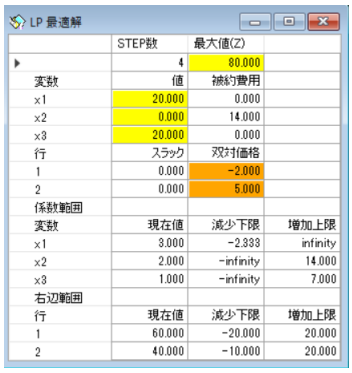


図 3　結果出力

これより、上の等号制約の線形計画問題の解は以下のようになります。

表 3　最適解

x1	x2	x3	Z
20	0	20	80

問題１（制約条件に等号と不等号の混じる例）

以下の線形計画問題を解け。

目的関数

$z = 2x_1 + 3x_2 + x_3$　　最大化

制約条件

$5x_1 + 2x_2 + 3x_3 \leq 10$

$2x_1 + x_2 + x_3 = 3$

$x_1 + 2x_2 - 2x_3 = 1$

$x_1, x_2, x_3 \geq 0$

最適解

x1	x2	x3	Z

問題２

以下の線形計画問題を解け。

目的関数

$z = 2x_1 + 3x_2 + 4x_3$　　最大化

制約条件

$5x_1 + 2x_2 + 3x_3 \leq 10$

$2x_1 + 3x_2 + x_3 = 5$

$x_1 + 2x_2 - 2x_3 = 3$

$x_1, x_2, x_3 \geq 0$

最適解

x1	x2	x3	Z

問題１解答

x1	x2	x3	Z
0	1.75	1.25	6.5

問題２解答

x1	x2	x3	Z
0	1.625	0.125	5.375

1.3 その他の線形計画問題　【動画】

これまで標準的な線形計画問題と等号制約の線形計画問題について見てきましたが、ここではさらに一般化するにはどうすべきか考えてみましょう。これは一見複雑そうに見えますが、うまく工夫することで、これまで勉強してきた２つの解法によって解くことが可能です。ここではその方法を、例を使って説明します。標準問題の制約を順番に取り除いていきましょう。

１）最小化問題の場合

$z = 2x_1 - x_2 + 3x_3$　最小化　の場合

$z' = -z = -2x_1 + x_2 - 3x_3$　最大化　として解く。

最小化問題は目的関数の符号を変えれば最大化問題になります。

２）右辺が負の場合

$-3x_1 + x_2 - 2x_3 = -5$　の場合、$3x_1 - x_2 + 2x_3 = 5$　として解く。

$-3x_1 + x_2 - 2x_3 \le -5$　の場合、$3x_1 - x_2 + 2x_3 \ge 5$　として４）へ。

右辺が負の場合も全体の符号を変えると、右辺は正になります。但し、不等号の向きが変わりますので、その時は４）で考えてやりましょう。

３）変数に非負条件がない場合

$x_1 \ge 0$　でない場合

$x_1 = x_1^+ - x_1^-,\ x_1^+ \ge 0, x_1^- \ge 0$　として変数を置き換えて解く。

変数に非負条件がない場合は、「非負の変数－非負の変数」というように、変数を２つに分けて考えます。これによって変数の数は増えますが、非負の条件はそのまま使えるようになります。

４）不等号の向きが逆の場合

$3x_1 - x_2 + 2x_3 \ge 5$　の場合、

$3x_1 - x_2 + 2x_3 - x_c = 5$　として、等号制約問題として解く。

２）からの場合も含めて、引き算のスラック変数を入れることで等号制約の問題に変更することができます。

これで標準問題のすべての制約を取り除くことができました。以上のことを考えると以下の例題も解けるようになります。

例１　以下の線形計画問題の最適解を求めよ。

$z = 2x_1 + 3x_2 + x_3$　　最大化

$3x_1 + 2x_2 + 5x_3 \le 20$

$4x_1 - 2x_2 + x_3 = 5$

$x_1 + x_2 + x_3 \ge 2$

$x_1, x_2, x_3 \ge 0$

最適解

x1	x2	x3	Z
3.571	4.643	0	21.071

例２

以下の線形計画問題の最適解を求めよ。

$$z = x_1 - 2x_2 \qquad \text{最大化}$$

$$2x_1 + x_2 \geq 2$$

$$x_1 - x_2 = 4$$

$$x_1 \geq 0$$

非負条件の付かない変数 x_2 について、グリッドの中では「x2!」のように、後ろに「!」記号が付きます。直接グリッドに入力する場合には特に覚えておいて下さい。

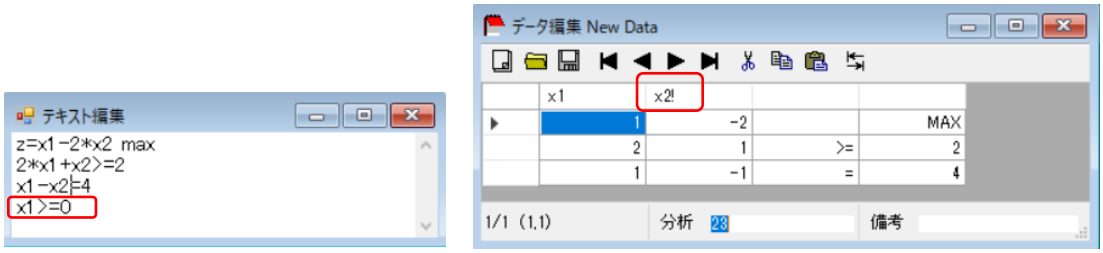


図１　非負条件が付かない場合の入力

結果は以下のようになります。

図２　結果出力

ここで、「x2_1」と「x2_2」というのが出ていますが、この結果は、x2=(x2_1)-(x2_2)=0-2=-2 のように読みます。

最適解

x1	x2	Z
2	-2	6

問題１　以下の線形計画問題の最適解を求めよ。

目的関数

$z = 2x_1 + 3x_2 + 5x_3$　　最大化

制約条件

$5x_1 + 2x_2 + 3x_3 \leq 10$

$2x_1 + 4x_2 + 2x_3 = 8$

$x_1 + 2x_2 + 2x_3 = 5$

$x_1, x_2, x_3 \geq 0$

最適解

x1	x2	x3	Z

問題２　以下の線形計画問題の最適解を求めよ。

目的関数

$z = 5x_1 - 2x_2$　　最大化

制約条件

$2x_1 + 3x_2 \leq 1$

$x_1 - x_2 = 5$

$x_1 \geq 0$

最適解

x1	x2	Z

問題１解答

x1	x2	x3	Z
1	1	1	10

問題２解答

x1	x2	Z
3.2	-1.8	19.6

1.4　輸送問題への応用　【動画】

これまで線形計画問題を考えてきましたが、ここではその応用を２つやってみましょう。１つはこの節の輸送問題への応用で、もう一つは包絡分析法という効率化測定手法です。後者は大きな内容ですので、章を改めて話をすることにします。

以下の例を考えてみます。

例　工場で生産した商品を各販売会社に配送する問題

今、3 つの工場から、4 つの会社に製品を運ぶ問題を考えます。各工場の生産数は決まっており、配送先の会社の需要も決まっているとします。下の表では、生産量は右端に、需要は一番下に書かれています。各工場から各会社に配送するのに輸送コストがかかりますが、下の表でそれぞれの枠の括弧の中にかかれた数字が、その工場から会社に単位数量運ぶのに必要な輸送コストです。変数 x_{ij} は求めるべき輸送量の値です。すべての輸送を最小のコストでおこなうには、どのように輸送量を設定したらよいでしょうか。

表　単位商品当りの輸送コストと輸送量

	会社１	会社２	会社３	会社４	生産量
工場１	(7) x_{11}	(3) x_{12}	(2) x_{13}	(10) x_{14}	18
工場２	(9) x_{21}	(3) x_{22}	(6) x_{23}	(8) x_{24}	22
工場３	(8) x_{31}	(7) x_{32}	(8) x_{33}	(6) x_{34}	26
需要	12	20	16	18	66

これは手短に言い換えると、( )内の数字を単位商品当りの輸送コストとして、生産高と需要との関係を満たしながら、総輸送コストを最小にする輸送量を求める、という問題です。

線形計画法による定式化

工場 i から会社 j への単位商品当りの輸送コストを c_{ij}、工場 i の生産高を a_i、会社 j の需要を b_j とすると、この問題は以下のような線形計画問題になります。

目的関数　$z=\sum_{i=1}^{3}\sum_{j=1}^{4}c_{ij}x_{ij}$　最小化

制約条件　$\sum_{j=1}^{4}x_{ij}=a_i,\quad \sum_{i=1}^{3}x_{ij}=b_j,\quad x_{ij}\geq 0$

これを具体的に書いてみると、以下のようになります。

$$z=7x_{11}+3x_{12}+2x_{13}+10x_{14}+9x_{21}+\cdots+6x_{34}\quad \text{最小化 (min)}$$

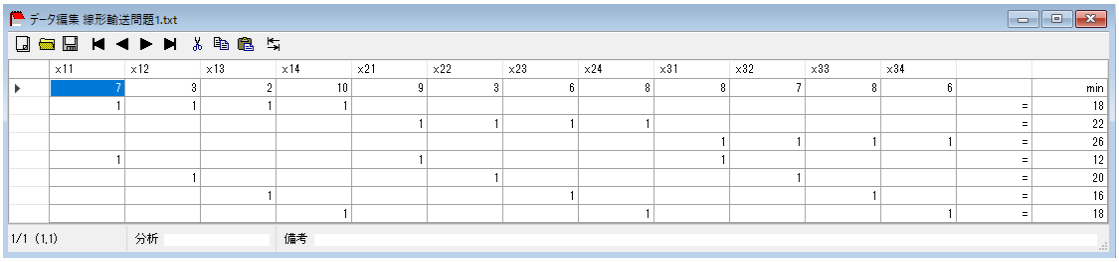
$$\begin{cases} x_{11}+x_{12}+x_{13}+x_{14}=18 \\ x_{21}+x_{22}+x_{23}+x_{24}=22 \\ x_{31}+x_{32}+x_{33}+x_{34}=26 \end{cases} \qquad \begin{cases} x_{11}+x_{21}+x_{31}=12 \\ x_{12}+x_{22}+x_{32}=20 \\ x_{13}+x_{23}+x_{33}=16 \\ x_{14}+x_{24}+x_{34}=18 \end{cases}$$

$$x_{11},x_{12},\cdots,x_{34}\geq 0$$

これをパソコンでやってみましょう。この例の場合、テキストで変数名をたくさん打ち込むより、グリッドに直接打ち込む方が楽になります。まず、メニュー［ファイル－新規作成］で出てくる以下の新規作成画面で、行数に制約数+1=7+1=8、列数に変数数+2=12+2=14 を入力し、「OK」ボタンをクリックします。

図 1　新規作成画面

グリッドエディタが大きく拡がりますので、以下のように入力します。



x11	x12	x13	x14	x21	x22	x23	x24	x31	x32	x33	x34		
7	3	2	10	9	3	6	8	8	7	8	6		min
1	1	1	1									=	18
				1	1	1	1					=	22
								1	1	1	1	=	26
1				1				1				=	12
	1				1				1			=	20
		1				1				1		=	16
			1				1				1	=	18

図 2　グリッドエディタへの直接入力（線形輸送問題 1.txt，1 頁にもあります）

ここで変数名は、エディタのメニュー［編集－列名入力］で、以下のように入力します。

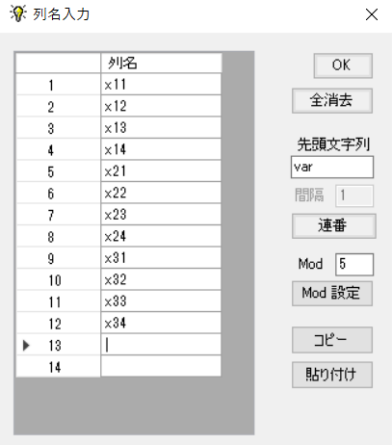


図 3　グリッドエディタの列名入力

このデータを用いて最適解を求めてみると以下になります。

LP 最適解

	STEP数	最小値(Z)
▶	16	296.000
変数	値	被約費用
x11	2.000	0.000
x12	0.000	2.000
x13	16.000	0.000
x14	0.000	5.000
x21	2.000	0.000
x22	20.000	0.000
x23	0.000	2.000
x24	0.000	1.000
x31	8.000	0.000
x32	0.000	5.000
x33	0.000	5.000
x34	18.000	0.000
行	スラック	双対価格
1	0.000	5.000
2	0.000	7.000

図４　結果出力

解答

	会社１	会社２	会社３	会社４
工場１	2	0	16	0
工場２	2	20	0	0
工場３	8	0	0	18

最適輸送費用［ 296 ］千円

問題１

３つの工場で生産した商品を３つの販売会社に輸送する。そのときの製品１単位当りの運送費用（括弧の付いた値，千円)、各工場の生産高（商品単位)、各会社への納入量（商品単位）は以下の表の通りである。全体の運送費用を最も安くするためには、各工場から各会社への輸送量をいくらにすればよいか。またそのときの輸送費用はいくらか。

	会社１	会社２	会社３	生産高
工場１	(4)	(4)	(5)	15
工場２	(3)	(4)	(5)	14
工場３	(6)	(5)	(4)	13
納入量	12	14	16	42

解答

	会社１	会社２	会社３
工場１			
工場２			
工場３			

輸送費用［　　］千円

問題２

上の問題で、工場３から会社３への輸送量は 8 単位を超えないものとすると、各工場から各会社への輸送量はどうなるか。またそのときの輸送費用はいくらか。

解答

	会社１	会社２	会社３
工場１			
工場２			
工場３			

輸送費用［　　］千円

問題１解答

	会社１	会社２	会社３
工場１	0	12	3
工場２	12	2	0
工場３	0	0	13

輸送費用［ 159 ］千円

問題２解答

	会社１	会社２	会社３
工場１	0	7	8
工場２	12	2	0
工場３	0	5	8

輸送費用［ 169 ］千円

ヒント　問題１の制約条件に、x33<=8 というのを付け加えます。

２．ＤＥＡ（Data Envelopment Analysis: 包絡分析法）　【動画】

2.1　DEA とは

通常、事業体（企業など）の生産や経営の効率は「産出÷投入」の形で考えられます。しかし、産出や投入に何を取るべきなのかはあいまいですし、だからといってそれぞれ複数の項目を取ったとしても、どの項目に重点を置くべきか疑問が残ります。つまり効率というものにはあいまいさが残り、分析者の好みが表れてきます。

ＤＥＡ（包絡分析法）は各事業体の得意な項目に重点を置いて効率を計るようにした手法で、なるべく利用者の好みやあいまいさが残らないように考えられた方法です。以下の例を見て下さい。

例　投入：従業員数，売場面積　産出：売上　として店舗の効率性を調べる。(DEA1.txt)

店舗	A	B	C	D	E	F	G	H	I
従業員数	20	42	20	26	20	15	45	55	42
売場面積	300	360	50	260	800	120	600	500	350
売上	20	24	10	26	40	12	30	40	28

これは店舗の効率性を、従業員数と売り場面積を投入に、売上を算出にして調べようというものです。店舗によって、従業員数で効率化を進めている場合と売り場面積で効率化を進めている場合があるとします。

比較の方法

今、売場面積÷売上と従業員数÷売上というデータを作ります。そのデータを店舗ごとにプロットしたのが以下の散布図です。

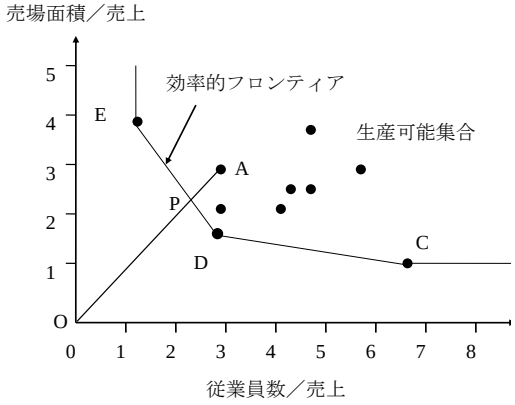


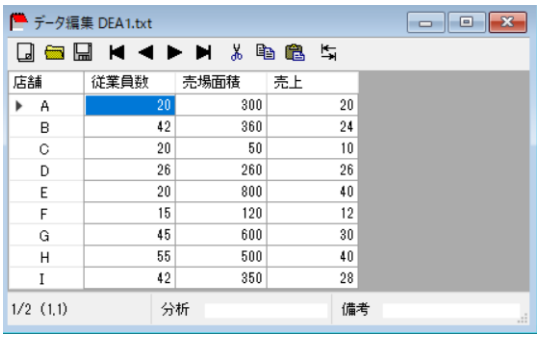
図１　店舗のプロット

このグラフで、どのような店舗（点）が効率的なのでしょうか。一般に、売上が大きく、売場面積や従業員数が小さい方が効率的と考えます。即ちこの図では、店舗が原点に近いほど効率的です。また、店舗の位置によって、何を努力しているのかも分かります。

次に、原点に近い店舗を結んで図のように線を引いてみます。線から店舗が原点側にはみ出さないようにします。端は y 軸に平行、x 軸に平行になるようにしてみます。この線のことを効率的フロンティアと呼びます。即ち、この線の上に並ぶ店舗が、最も効率的な店舗を表していると考え、この店舗に効率 1 を与えます。その他の店舗の効率は、原点とその店舗を結ぶ線が効率的フロンティアと交わる点を使って表します。例えば店舗 A では、OP/OA が効率値です。このような測り方だと、店舗が違えば、効率的フロンティアのどの部分と交わるかが変わりますので、この方法は店舗の良いところを考慮して効率を測定していることになります。

このような考えで効率性を求める分析法を DEA といいます。詳細な理論は省略しますが、ここでは前章で習った線形計画法が使われています。では実際にこの分析をやってみましょう。

まず、このデータを開いておきましょう。エディタのメニュー［ファイル－開く］で Samples の中の DEA1.txt を図のように開いて下さい。



店舗	従業員数	売場面積	売上
A	20	300	20
B	42	360	24
C	20	50	10
D	26	260	26
E	20	800	40
F	15	120	12
G	45	600	30
H	55	500	40
I	42	350	28

図 2　データ（DEA1.txt）

分析実行画面はメニュー［分析－OR－DEA］を選択すると、以下のように表示されます。

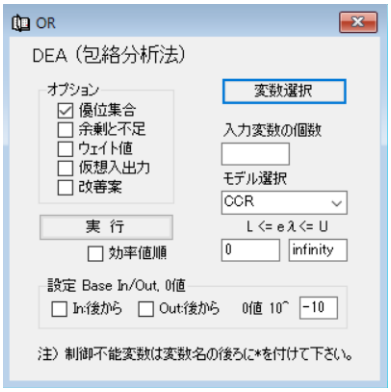


図 3　分析実行画面

まず「変数選択」で、入力変数、出力変数の順に指定します。ここでは順番に、従業員数、売り場面積、売上とします。次に「入力変数の個数」のテキストボックスに、この場合の 2 を（半角で）入力します。「実行」ボタンをクリックしてみましょう。以下のような結果が表示されます。

CCR モデル 計算結果

	D効率値	優位集合1(	優位集合2(
A	0.857	D(0.549)	E(0.143)
B	0.632	C(0.253)	D(0.826)
C	1.000	C(1.000)	
D	1.000	D(1.000)	
E	1.000	E(1.000)	
F	0.923	C(0.185)	D(0.391)
G	0.600	D(0.923)	E(0.150)
H	0.774	C(0.258)	D(1.439)
I	0.750	C(0.350)	D(0.942)

図 4　分析結果

ここで、各店舗名に続いて、右側に効率値、優位集合の店舗名が書いてあります。優位集合とは、効率を測定している店舗に類似した効率的フロンティア上の店舗のことです。図 1 の店舗 A では、D と E が優位集合です。

優位集合の名前の右に数字が書いてありますが、これはどの程度優位集合に近いかを表す数字で、大きいほど近いと考えます。以上の結果をまとめると以下になります。

解答（有意集合の数値は近さを表す値）

	効率値	優位集合
A	0.857	D(0.549), E(0.143)
B	0.632	C(0.253), D(0.826)
C	1.000	C(1.000)
D	1.000	D(1.000)
E	1.000	E(1.000)
F	0.923	C(0.185), D(0.391)
G	0.600	D(0.923), E(0.150)
H	0.774	C(0.258), D(1.439)
I	0.750	C(0.350), D(0.942)

問題 1

Samples¥DEA2.txt はあるグループの店舗についてのデータである。入力を従業員数と売場面積、出力を売上と会員数として（CCR モデル）で各店舗の効率性を調べ、それらの効率値とその優位集合（名前のみ）を求めよ。

店舗	A	B	C	D	E	F	G	H
従業員数	20	42	20	26	20	15	45	55
売場面積	300	360	50	260	800	120	600	500
売上	20	24	10	26	40	12	30	40
会員数	522	734	340	433	525	350	826	913

解答

	効率値	優位集合
A		
B		
C		
D		
E		

F		
G		
H		

問題２

Samples¥DEA3.txt は東京都各区の図書館のデータである。このデータを用いて、入力を蔵書数（千冊）と職員数（人）、出力を登録者数（人）と貸出冊数（千冊）として各図書館の効率を DEA を用いて検討し、いくつかの図書館について効率値と優位集合（名前のみ）答えよ。

	効率値	優位集合
千代田		
中央		
台東		
荒川		
港		
文京		
墨田		
渋谷		

問題１解答

	効率値	優位集合
A	1.000	A(1.000)
B	0.742	A(0.127), F(1.907)
C	1.000	C(1.000)
D	1.000	D(1.000)
E	0.933	A(1.400)
F	1.000	F(1.000)
G	0.721	A(1.237), F(0.515)
H	0.813	A(0.063), D(0.768), F(1.565)

問題２解答

	効率値	優位集合
千代田	0.226	世田谷(0.029)
中央	0.638	世田谷(0.095)
台東	0.540	世田谷(0.132)
荒川	0.593	世田谷(0.207)
港	0.911	板橋(0.256), 世田谷(0.162)
文京	0.745	世田谷(0.351)
墨田	0.650	杉並(0.170), 世田谷(0.162)
渋谷	0.539	板橋(0.322)

2.2　種々のモデル

これまで扱ってきた DEA で各事業体の効率と優位集合は分かりましたが、ここではその他の機能について説明します。以下の分析実行画面を見て下さい。

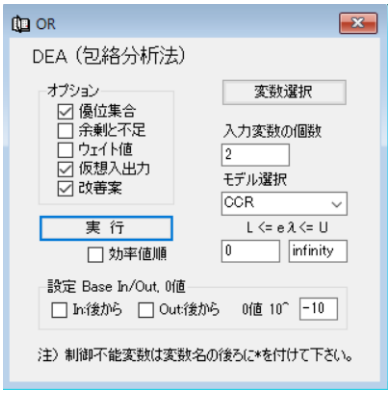


図 1　分析実行画面

オプションのところで、立ち上げた最初は「優位集合」のところだけにチェックが付いています。これが、実行結果に優位集合が表示された理由です。上の画面では「仮想入出力」と「改善案」のところにもチェックが付いています。仮想入出力は、事業体の効率がどの変数を評価して得られたか、ということを表します。また、改善案はどのように改善すると優位集合に含まれるような効率 1 の事業体になれるかを与える 1 つの案です。あまり現実的でない場合もありますが、分かり易い機能です。DEA1.txt のデータを元に、分析を実行してみましょう。実行結果を以下に示します。

CCR モデル 計算結果

	D効率値	優位集合1(	優位集合2(	v1×従業員	v2×売場面	u1×売上
A	0.857	D(0.549)	E(0.143)	0.571	0.429	0.857
B	0.632	C(0.253)	D(0.826)	0.368	0.632	0.632
C	1.000	C(1.000)		0.667	0.333	1.000
D	1.000	D(1.000)		0.333	0.667	1.000
E	1.000	E(1.000)		1.000	0.000	1.000
F	0.923	C(0.185)	D(0.391)	0.385	0.615	0.923
G	0.600	D(0.923)	E(0.150)	0.600	0.400	0.600
H	0.774	C(0.258)	D(1.439)	0.355	0.645	0.774
I	0.750	C(0.350)	D(0.942)	0.375	0.625	0.750

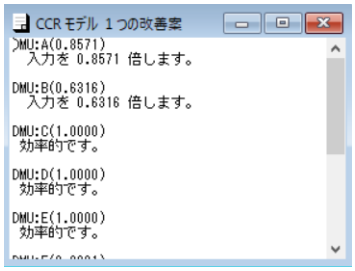


図 2　実行結果

仮想入出力ですが、投入と産出を分けて見ます。ここでは、投入の変数が 2 つありますので、見て下さい。一番上の A の店舗では、従業員数が 0.571、売り場面積が 0.429 となっています。この数値の大きさで、評価の度合いが決まります。この場合、ほとんど同じだけれど、どちらかというと従業員数の評価が高いことになります。改善案は読んでみればどうすればよいか分かると思います。中には、ほぼ無理でしょうという内容も含まれています。結果をまとめておきます。

各事業体の効率を知るには　　→　　D 効率値

各事業体に似た効率的な事業体は　　→　　優位集合

優位集合のうちどちらに近い　　→　　優位集合名のカッコ内の数字の大きい方

各事業体の入出力でどの項目が評価されたか　　→　　仮想入出力で大きい値のもの

注）効率 1 の場合、項目の評価は必ずしもそうとはいえない

次に、分析実行画面右側の「モデル選択」について見てみます。図 1 ではモデルの名前が CCR となっています。これは規模の収穫一定のモデルとも呼ばれ、事業体の大きさが大きくなっても、小さくなっても効率は同じように評価するモデルです。小さな商店と大きなスーパーとを同じように評価するので、不公平感のあるモデルとも考えられます。これに対して例えば、IRS モデルというのもあり、これは規模の収穫増加を仮定したモデルです。元々大きなスーパーは効率が良いのだから、その分差し引いて評価するというモデルです。他にもありますので以下のモデルの特徴を見て下さい。

モデルの特徴

CCR モデル　規模の収穫一定を仮定したモデル
（入出力が同比率で拡大すると効率は同じ）

IRS モデル　規模の収穫増加を仮定したモデル
（入出力が同比率で拡大すると効率は下がる）

DRS モデル　規模の収穫減少を仮定したモデル
（入出力が同比率で拡大すると効率は上がる）

BCC モデル　規模の小さいときは IRS、大きいときは DRS

この他にも、これに準じて CCRO, IRSO, DRSO, BCCO というモデルもあります。これらは、出力モデルと呼ばれ、効率値の値は同じですが、他の部分が異なってきます。詳細は省略します。これで問題はできると思いますので、やってみて下さい。

問題３

samples¥DEA2.txt はあるグループの店舗についてのデータである。入力を従業員数と売場面積、出力を売上と会員数として CCR, IRS, DRS, BCC の各モデルで各店舗の効率性を調べ、それらの効率値を求めよ。

店舗	A	B	C	D	E	F	G	H
従業員数	20	42	20	26	20	15	45	55
売場面積	300	360	50	260	800	120	600	500
売上	20	24	10	26	40	12	30	40
会員数	522	734	340	433	525	350	826	913

解答

効率値	CCR	IRS	DRS	BCC
A				
B				
C				
D				
E				
F				
G				
H				

問題 4

Samples¥DEA3.txt は東京都各区の図書館のデータである。このデータを用いて、入力を蔵書数（千冊）と職員数（人）、出力を登録者数（千人）と貸出冊数（千冊）として各図書館の効率を DEA を用いて検討し、以下の問いに答えよ。

１）CCR モデルを用いて港区、文京区の効率値とそれらの優位集合を求めよ。

	効率値	優位集合
港区		
文京区		

２）BCC モデルを用いて港区、文京区の効率値とそれらの優位集合を求めよ。

	効率値	優位集合
港区		
文京区		

３）入力と出力を同じ比率で大きくすると以下のモデルで効率はどうなるか。

CCR	一定・上がる・下がる・どちらともいえない
IRS	一定・上がる・下がる・どちらともいえない
DRS	一定・上がる・下がる・どちらともいえない
BCC	一定・上がる・下がる・どちらともいえない

４）BCC モデルは規模が小さいとき［IRS・DRS］モデルに似て、規模が大きくなると［IRS・DRS］モデルに似てくる。

<u>以後は CCR モデルを用いて質問に答えよ。</u>

５）文京区の入力では［蔵書数・職員数］が主に評価され、出力では［登録者数・貸出冊数］が主に評価されている。

６）文京区を効率的にする案として、まず入力を［　　］倍にする。その後、職員数を［　　］人減らし、登録者数を［　　］千人増やす。

<u>以後は CCRO モデルを用いて質問に答えよ。</u>

７）CCR モデルと比べて効率に差があるか。［ある・ない］

８）文京区を効率的にする案として、まず出力を［　　］倍にする。その後、職員数を［　　］人減らし、登録者数を［　　］千冊増やす。

問題３解答

効率値	CCR	IRS	DRS	BCC
A	1.000	1.000	1.000	1.000
B	0.742	0.742	1.000	1.000
C	1.000	1.000	1.000	1.000
D	1.000	1.000	1.000	1.000
E	0.933	0.933	1.000	1.000
F	1.000	1.000	1.000	1.000
G	0.721	0.721	1.000	1.000
H	0.813	0.813	1.000	1.000

問題４解答

１）CCR モデルを用いて港区、文京区の効率値とそれらの優位集合を求めよ。

	効率値	優位集合
港区	0.911	板橋，世田谷
文京区	0.745	世田谷

２）BCC モデルを用いて港区、文京区の効率値とそれらの優位集合を求めよ。

	効率値	優位集合
港区	1.000	港
文京区	0.885	江東，世田谷

３）入力と出力を同じ比率で大きくすると以下のモデルで効率はどうなるか。

CCR	一定・上がる・下がる・どちらともいえない
IRS	一定・上がる・下がる・どちらともいえない
DRS	一定・上がる・下がる・どちらともいえない
BCC	一定・上がる・下がる・どちらともいえない

４）BCC モデルは規模が小さいとき［IRS・DRS］モデルに似て、規模が大きくなると［IRS・DRS］モデルに似てくる。

以後は CCR モデルを用いて質問に答えよ。

５）文京区の入力では［蔵書数・職員数］が主に評価され、出力では［登録者数・貸出冊数］が主に評価されている。

６）文京区を効率的にする案として、まず入力を［ 0.745 ］倍にする。その後、職員数を［ 13.977 ］人減らし、登録者数を［ 1.006 ］千人増やす。

以後は CCRO モデルを用いて質問に答えよ。

７）CCR モデルと比べて効率に差があるか。［ある・ない］

８）文京区を効率的にする案として、まず出力を［ 1.342 ］倍にする。その後、職員数を［ 18.762 ］人減らし、登録者数を［ 1.351 ］千人増やす。

注）CCR モデルの改善案は規模を縮小する方向で進め、CCRO モデルの改善案は規模を拡大する方向で進めることが分かります。

３．待ち行列　【動画】

3.1　待ち行列とは

待ち行列とはスーパーのレジや銀行の ATM、駅の緑の窓口などで人が並ぶ状態だけでなく、ネットのアクセス待ちなど見えない状態も含めた言葉です。どのぐらいの人数がどれくらい並ぶか、どれくらい待てばよいか、様々な状況でそれらを予測する数学的なモデルが今回学ぶ待ち行列です。この理論は昔のアメリカの電話網普及の時代に、電話交換機の混雑を表すモデルとして、アーラン（A.K.Erlang）によって始められました。これはオペレーションズ・リサーチ（OR）と呼ばれる数学の一分野で、研究がよく進んだテーマの 1 つです。

待ち行列では理論により結果を求める方法とシミュレーションにより結果を求める方法がありますが、ここでは現実に使えるシミュレーションを使った待ち行列を考えることにします。

ここで扱う待ち行列は以下の図のようなものを考えます。

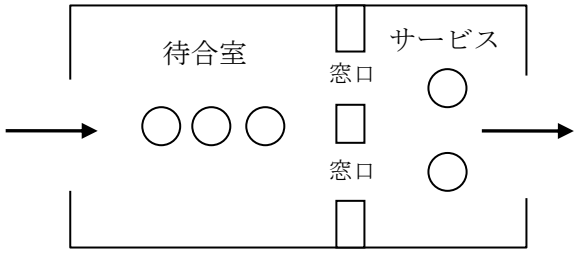


図 1　待ち行列のシステム

（窓口 2，滞在数（システム内）5 人，待ち数 3 人）

待ち行列にはその先でサービスを行う窓口が 1 つまたは複数あります。客はこの窓口の前で待ちますが、窓口が複数の場合、1 列で待つか、窓口ごとに並んで待つかします。前者は駅の緑の窓口の方式、後者はスーパーのレジで見かける方式です。待ち行列にはいくらでも並んでよい場合と人数制限を設ける場合がありますが、上の図ではこれを待合室に入れる人数というように表現しています。人数制限を設ける場合、やってきた人は制限人数を超えた場合帰ってしまい、機会損失が起こります。待っている人の人数を「待ち数」、待っている人とサービスを受けている人の合計を「滞在数」と呼ぶことにします。

待ち行列のシミュレーションに必要なデータ

待ち行列のシミュレーションを行う場合、以下の必要な情報があります。

１）客の到着の仕方は？

単位時間（1 時間でも 1 分でもよい）当たりの到着数の分布（どのぐらいの確率でどのぐらい来るかということ）とその平均が必要です。これは到着時間間隔の分布とその平均と言い換えてもよいでしょう。

２）サービス時間の分布とその平均は？

これは到着の場合と同じです。特にサービス時間で見ることが多いようです。

３）窓口の数は？

これは言い換えると、同時にサービスを受けられる人の数になります。

４）待合室の人数制限は？

人数制限があるかないか、ある場合は何人かという情報です。

待ち行列の求めたい主な結果

１）客の平均待ち時間

サービスが始まるまでの平均時間です。

２）待ち行列の平均的長さ

待っている人の平均の人数です。滞在者数ではありません（滞在者数は別途考えます）。

３）窓口の稼働率

サービスを行っている時間の割合です。これが低いと無駄になります。

４）ある一定時間以上待つ確率

この確率はソフトでは出力されませんが、必要になる場合もあります。

3.2　ケンドール（Kendall）の記号

上の状況を組み合わせるために、ケンドールという統計学者が考えた記号を紹介しておきます。どのような待ち行列かこれらの記号の並びで表します。

待ち行列を特徴付ける記号表示

待ち行列は以下のように 4 つの部分で表されます

A/B/s [(r)]

この中で、「A」と「B」は到着とサービスの分布です。到着では到着数に着目し、サービスはサービス時間に着目することが多いようですが、基本的には単位時間に何人到着し、何人サービスできるかですので同じです。分布は以下の記号で表します。

D：確定分布 M：指数分布（ポアソン分布）　E_k：k 次アーラン分布

H：超指数分布　　G：一般の分布

ここで特に重要なのは D と M でしょう。D は時間間隔が決まっている場合で、M は平均だけは決まっていますが、全くランダムに到着し、サービスが終わる場合です。

「s」はサービス窓口の数を数字で表します。

「r」は待合室に入れる人数（待ち行列の長さ制限）を、括弧を付けて表します。$r=\infty$（長さに制限がない）の場合は省略できます。以下の例を見て下さい。

例

M/M/1　　指数到着、指数サービス、窓口１つ（行列に制限なし）

D/M/2 (5)　　確定到着、指数サービス、窓口２つ、行列 5 人まで

3.3　定常的な待ち行列

実際にプログラムを動かして待ち行列のシミュレーションをしてみましょう。ソフトのメニュー［分析－OR－待ち行列］を選択すると以下のような実行画面が表示されます。

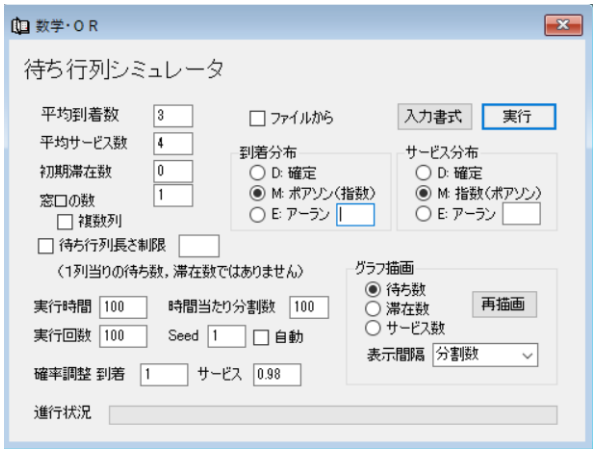
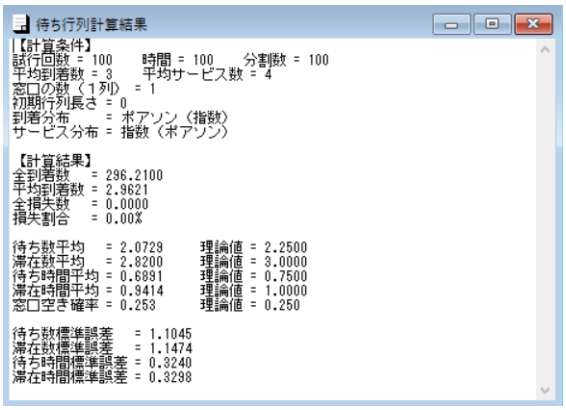


図 2　待ち行列シミュレーション実行画面

何の設定もせずにまず「実行」ボタンをクリックしてみて下さい。以下のような 2 つの Window が表示されます。



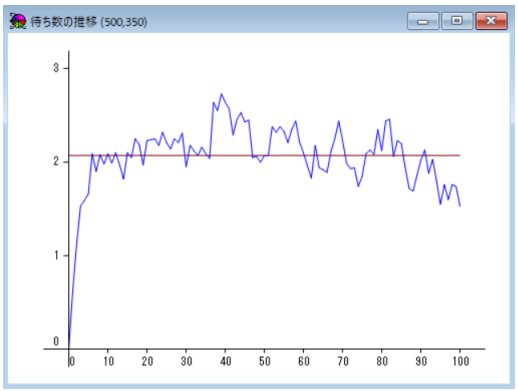


図 3　シミュレーション実行結果

このシミュレーションを図 2 の画面を使って説明します。画面左上から、「平均到着数」は 3、「平均サービス数」は 4、「窓口の数」は 1 つです。シミュレーションを開始する際の「初期滞在数」は 0 からです。「待ち行列の長さ制限」は、チェックボックスにチェックが入っていないので、ありません。「到着分布」と「サービス分布」はポアソン分布で、記号では M です。ポアソン分布は到着時間間隔やサービス時間に注目すると指数分布ということもできます。

「実行時間」は 100 単位時間、「1 単位時間分割数」を 100 に設定していることから、1 単位時間を 100 に分割して、その都度人が来たかどうかをチェックし、シミュレーションを実行しています。また、1 回だけのシミュレーションではばらつきが大きく上の右図のような結果にならないので、「実行回数」を 100 として、シミュレーションを 100 回行い、その平均をとっています。これが例えば 1 回だと以下のような結果になります。

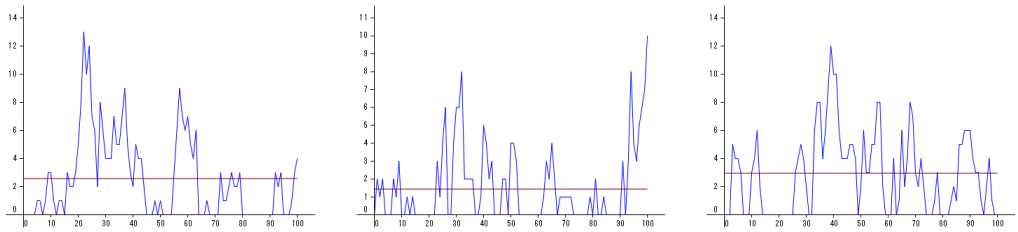


図 4　1 回のシミュレーション結果

３つの図が違うのは、シミュレーションに使う乱数の種類を「Seed」で 1, 2, 3 と変えているからです。このようなシミュレーションを 100 回繰り返し、図 3 の結果を得ています。ちなみに図 3 の乱数の「Seed」は 1 です。この段落で述べた、実行回数や 1 単位時間分割数、Seed などについてはこのテキストでは変更しません。

次に図 3 に戻って、文字の実行結果の方を見てみましょう。「待ち数平均」、「待ち時間平均」などは特に重要です。この結果には理論的に求めた結果も表示され、シミュレーション結果と比較することができます。シミュレーション結果の平均は、時間の最初から終わりまで、理論の結果は待ち数が安定した状態（上のグラフでいえば平らになった部分）での平均ですので、少し違っています。以上のことを確かめた上で例題をやってみることにしましょう。今回は解答を分けてありますので、できるだけ自分でやってみて下さい。

例

以下の待ち行列は定常的であるか。<u>定常的ならば</u>、平均待ち数と平均待ち時間をシミュレーション実行結果から求めよ。但し、乱数は Seed を 1、実行時間や分割数などはデフォルト（変更しないまま）とせよ。

１）M/M/1，到着数 4，サービス数 5
　定常的で［ある・ない］，平均待ち数［　　］，平均待ち時間［　　］

２）M/M/1，到着数 5，サービス数 4
　定常的で［ある・ない］，平均待ち数［　　］，平均待ち時間［　　］

３）M/M/2，到着数 6，サービス数 4，窓口に全体で 1 列に並ぶ場合
　定常的で［ある・ない］，平均待ち数［　　］，平均待ち時間［　　］

４）M/M/2，到着数 6，サービス数 4，窓口ごとに複数列で並ぶ場合
　定常的で［ある・ない］，平均待ち数［　　］，平均待ち時間［　　］

５）複数窓口の場合、どちらが効率は良いか。
　［1 列に並ぶ場合・複数列で並ぶ場合］

６）M/M/1(5)，到着数 3，サービス数 4（待ち人数が 5 人までの場合）
　平均待ち数［　　］，平均待ち時間［　　］

７）上の場合、行列に並ばずに帰ってしまった人は全体の何%か。
　[　　]%

８）M/M/1，到着数 3，サービス数 4，最初にシステム内に 10 人並んでいる場合
定常的で［ある・ない］，平均待ち数［　　］，平均待ち時間［　　］

９）前問の場合、システムが定常状態になるのにおよそ何単位時間かかるか。
約［　　］単位時間

例題解答

１）M/M/1，到着数 4，サービス数 5
定常的で［ある・ない］，平均待ち数［ 3.180 ］，平均待ち時間［ 0.790 ］
ソフトでは、平均待ち数が待ち数平均に、平均待ち時間が待ち時間平均になっています。

２）M/M/1，到着数 5，サービス数 4
定常的で［ある・ない］，平均待ち数［　　］，平均待ち時間［　　］
これは待ち数がどんどん大きくなっているので定常的とはいえません。

３）M/M/2，到着数 6，サービス数 4，窓口に全体で 1 列に並ぶ場合
定常的で［ある・ない］，平均待ち数［ 1.999 ］，平均待ち時間［ 0.331 ］
窓口の数を 2 に設定します。到着数＜2*サービス数で定常的です。

４）M/M/2，到着数 6，サービス数 4，窓口ごとに複数列で並ぶ場合
定常的で［ある・ない］，平均待ち数［ 2.221 ］，平均待ち時間［ 0.368 ］
窓口の数の複数窓口にチェックを入れて下さい。

５）複数窓口の場合、どちらが効率は良いか。
［1 列に並ぶ場合・複数列で並ぶ場合］
平均待ち数や平均待ち時間が 1 列に並ぶ方が少なくなっています。

６）M/M/1，到着数 3，サービス数 4　　平均待ち数［ 2.073 ］

７）M/D/1，到着数 3，サービス数 4　　平均待ち数［ 1.064 ］

８）$M/E_2/1$，到着数 3，サービス数 4　　平均待ち数［ 1.624 ］
この３つは分布を変えて試しています。E はアーラン分布と呼ばれる分布です。特に気にする必要はありません。

９）上の３つの待ち行列の記号を平均待ち数が少ない順に並べて書け。
［ M/D/1 ］［ $M/E_2/1$ ］［ M/M/1 ］

10）M/M/1(5)，到着数 3，サービス数 4（待ち人数が 5 人までの場合）
平均待ち数［ 1.177 ］，平均待ち時間［ 0.396 ］
待ち行列長さ制限にチェックを入れ、その数を 5 に設定します。

11）上の場合、行列に並ばすに帰ってしまった人は全体の何%か。　［ 4.74 ］%
結果の損失割合のところを見ます。

12）M/M/1，到着数 3，サービス数 4，最初にシステム内に 10 人並んでいる場合
定常的で［ある・ない］，平均待ち数［ 2.733 ］，平均待ち時間［ 0.908 ］
初期滞在数を 10 にしてシミュレーションします。

13）前問の場合、システムが定常状態になるのにおよそ何単位時間かかるか。
約［ 20 ］単位時間
グラフが平らになったところのおよその時間を見ます。

3.4　時間とともに変化する待ち行列　【動画】

これまでは定常的な待ち行列で、理論だけでもかなりの部分が説明可能でした。しかし、これからは昼休み等で時間とともに来客数が変化したり、突発的な変化に対応したり、本当の意味でシミュレーションにしかできないことを勉強します。以下の例を見て下さい。

例

単位時間当たりの到着数・サービス数、窓口数が以下のように時間的に変化するときの待ち行列の状態を調べる。（Samples¥待ち行列 1.txt）

時刻	到着数	サービス数	窓口数
0	3	4	1
50	6	4	1
55	6	4	2
100			

これは時刻 0 から 100 分（単位時間）までのシミュレーションで、サービス数が窓口当たり 1 分で平均 4 人は変わりません。到着数は初め 1 分平均 3 人だったのですが、50 分後 6 人に増えました。その状態では待ち行列が伸びて行ったので、55 分に窓口を 2 つに増やしました。この状態を見てみようというというシミュレーションです。

解答

データは待ち行列 1. txt に入っています。それを図のように開いてみます。

図 1　シミュレーションデータ

ここでファイルでは、「時刻」、「到着数」、「サービス数」、「窓口数」の順番に入力します。分析プログラムでは、上中央のところにある「ファイルから」チェックボックスにチェックを入れます。これで「平均到着数」、「平均サービス数」、「窓口数」は、ファイルからの指示通りになります。その他の設定は画面通りです。

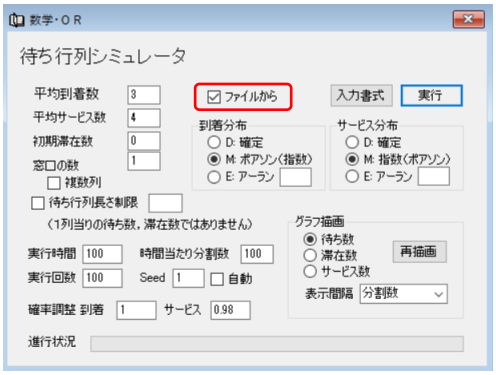


図 1　分析実行画面

これで「実行」ボタンをクリックすると以下のような結果が表示されます。

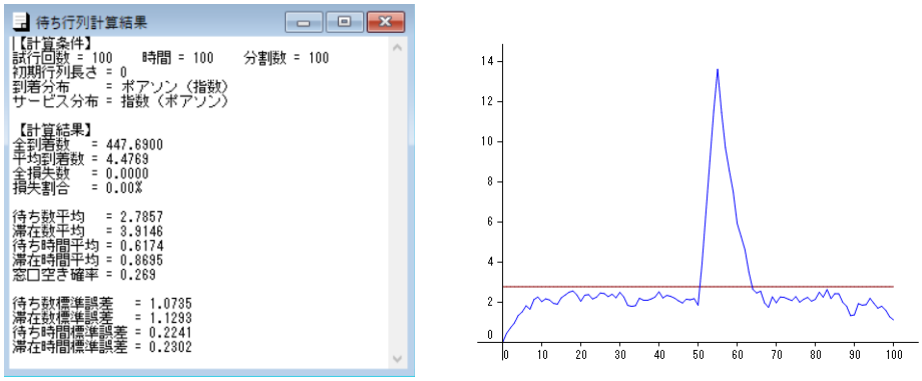


図 2　シミュレーション実行結果

ここで、右のグラフに引かれている横線は待ち数の平均値です。

一度シミュレーションを実行したのち、グラフ描画のグループボックスで、「滞在数」や「サービス数」のラジオボタンを選択し、「再描画」をクリックすると以下のような指標を変えたグラフの描画もできます。

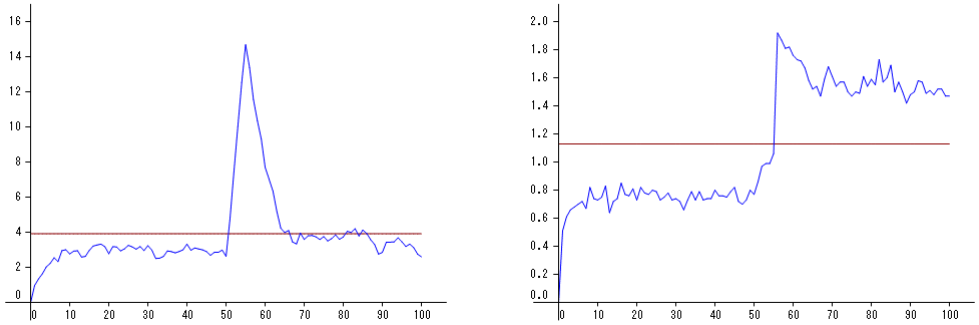


図 2　シミュレーション結果の別指標表示

それでは、以下の問題を考えてみて下さい。シミュレーションでは乱数が異なると結果が違ってきますので、乱数の seed を 1（デフォルト）として下さい。

問題 1

ショッピングセンターのＡＴＭに、昼 12 時（時刻 0）から 12 時 20 分（時刻 20）までに 8 人、12 時 20 分から 40 分（時刻 40）までに 16 人、12 時 40 分から 13 時（時刻 60）までに 12 人ランダムに到着する。1 分当たりの処理人数が平均 0.7 人であるとき、単位時間を 1 分とし、M/M/1 を仮定して以下を求めよ。

1）各時間帯で 1 分当たりの到着数は何人か

12 時～12:20	12:20～12:40	12:40～13 時

2）最大の待ち人数はおよそいくらか　［　　］人

3）平均の待ち人数は何人か　［　　］人

4）平均の待ち時間は何分か　［　　］分

問題２

みどりの窓口で 60 分間に 72 人の客がランダムに到着するとする。サービス時間は 1 人平均 2 分で、窓口は 3 つ開いているものとする。単位時間を 1 分とし、M/M/n の 1 列を仮定して以下の問いに答えよ。シミュレーションの時間は 100 分とする。

1）1 分当たりの客の到着人数は何人か　［　　］人
2）1 分当たりのサービス人数は何人か　［　　］人
3）平均の待ち人数は何人か　［　　］人
4）平均の待ち時間は何分か　［　　］分
5）窓口の空き確率はいくらか［　　］
6）開始から 90 分後、1 台が故障して窓口が 2 つになった。その 10 分後にはおよそ何人の行列ができるか。　およそ［　　］人

問題１解答

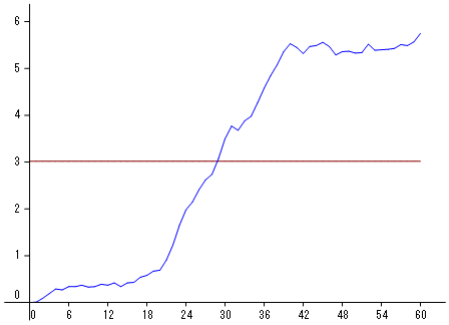
1）各時間帯で 1 分当たりの到着数は何人か

12 時～12:20	12:20～12:40	12:40～13 時
0.4	0.8	0.6

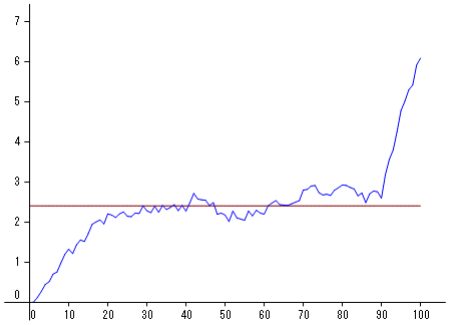
2）最大の待ち人数はおよそいくらか　［ 6 ］人
3）平均の待ち人数は何人か　［ 3.016 ］人
4）平均の待ち時間は何分か　［ 4.691 ］分

問題２解答

1）1 分当たりの客の到着人数は何人か　［ 1.2 ］人
2）1 分当たりのサービス人数は何人か　［ 0.5 ］人
3）平均の待ち人数は何人か　［ 2.258 ］人
4）平均の待ち時間は何分か　［ 1.822 ］分
5）窓口の空き確率はいくらか［ 0.388 ］
6）その 10 分後にはおよそ何人の行列ができるか。　およそ［ 6 ］人



問題１待ち数グラフ



問題２待ち数グラフ

４．在庫管理　【動画】

4.1　在庫管理とは

在庫管理問題とは例えば以下の図のように供給元（問屋や工場など）から出荷された原料や製品などを一時的に保管し、需要先（顧客や商店など）の要求に応じて出荷する、店舗や倉庫、物流センターなどでの商品管理の問題です。在庫量が多ければ保管費用がかさみ、在庫量が少なければ需要先からの要求に欠品が生じます。欠品が起こらないで、費用もできるだけ抑えられる、適正な在庫量はいくらしょうか。このような問題に対して、どのような条件でどれだけの量を供給元に発注するのかを決めるのが在庫管理の問題です。

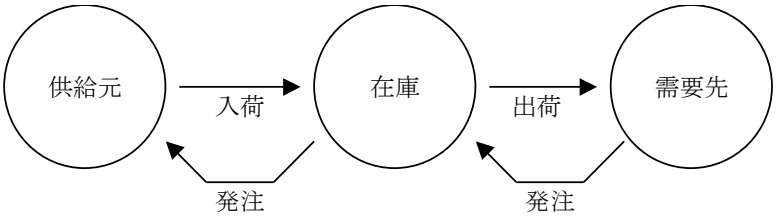


図１　原料・中間原料・製品・商品などの在庫管理

在庫が少なくなったら、供給元に発注しますが、商品はすぐに届くわけではありません。発注してから届くまでには時間がかかります。これを調達期間といいます。また、需要先からの要求も決まった量がくるとは限りません。多いときもあれば少ないときもあります。理論的にはこの出庫量を正規分布とみなし、平均と標準偏差（ばらつき）を考えます。このような情報を元に在庫管理の方法が決められます。

まず在庫管理に必要な情報を見てみましょう。

必要な情報

調達期間（リードタイム）L 日

これは発注から入庫までの間隔です。

出庫の量（確定的か確率的か？）

一般に正規分布で近似しますが、確定的な場合は標準偏差 0 と考えます。

我々が決定する事項は以下のような発注時期や発注量です。

決定事項（重要な意思決定）

発注時期（いつの時点で発注するか？）在庫の量 I または発注間隔 R

発注量（どれだけ発注するか？）Q

ここでは単位として日を使っていますが、これは分かり易さのためで特に日に限るわけではありません。。

この問題を解決する方法として、ここで取り上げるのは基本的な２つの方法です。１つは発注が必要になった時に一定の量を発注する定量発注方式で、もう１つは一定の期間をおいて必要な量を発注する定期発注方式です。これらは初歩的な方法ですが、複雑な方法の基礎として、考え方を学んでおくことは意味があります。それではそれぞれの方法について分けて見て行きましょう。

4.2　定量発注方式

定量発注方式というのは倉庫の在庫量がある基準以下になったとき、決まった一定の発注量を発注する方式です。定められた一定の発注量は管理者が需要量を見て決めておきます。基本は在庫量をできるだけ少なく保つことですが、発注した製品が届くまでの間に需要によって欠品が生じないようにすることが重要です。定められた一定の発注量、出庫の平均と標準偏差、発注から入荷までの調達期間が分かっているとき、欠品の危険率（許容確率）を定めて、在庫量がいくつになったら発注するかという発注点を求めます。定量発注方式に必要な情報とそれから決定される事項は以下の通りです。

必要な情報

発注量　Q

調達期間（リードタイム）L（日）

１日当たりの出庫（確率的）　$N(\mu,\sigma^2)$

平均 μ（個）、分散 σ^2（標準偏差 σ 個）の正規分布を仮定

欠品の危険率（許容範囲）$\alpha\times100\%$

決定事項

発注点（在庫量がどこまで減ったら発注するか？）

在庫の量をグラフにしてみると、以下のような変動になります。

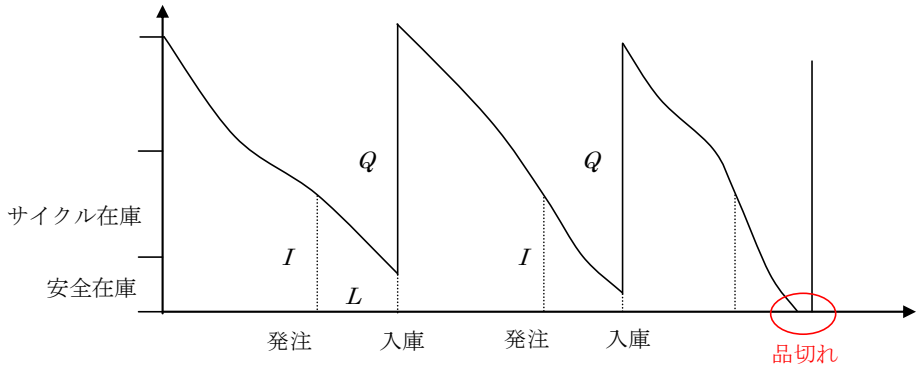


図 2　定量発注方式イメージ図

発注点を求めるための公式を書いておきましょう。

公式　定量発注方式

１日当たりの出庫　平均 μ（個）、分散 σ^2（標準偏差 σ 個）の正規分布

発注量：Q　調達期間（リードタイム）：L（日）　品切れの危険率：$\alpha\times100\%$

安全係数を $\lambda = normsinv(1-\alpha)$ として、在庫が $I = L\mu + \lambda\sigma\sqrt{L}$ になった時点（発注点）で、Q の量を発注する。

サイクル在庫：$Q/2$、安全在庫：$\lambda\sigma\sqrt{L}$、理論在庫：$Q/2+\lambda\sigma\sqrt{L}$

ここでサイクル在庫は発注量の半分、安全在庫は発注が多かった場合に品切れを起こさないため上乗せする在庫量、理論在庫はその合計です。安全在庫は、欠品の危険率から求まる

安全係数λを、調達期間全体での需要の標準偏差$\sigma\sqrt{L}$に掛けて求めます。安全係数は標準正規分布の上側確率αの値です。

例

1 日当たりの出庫の平均 20 個、標準偏差 5 個、欠品の危険率 5%以下、調達期間 7 日のとき、発注量 200 個の定量発注方式として以下の問いに答えよ。

1 日当たりの出庫の平均 20 個、標準偏差 5 個、欠品の危険率 5%以下、調達期間 7 日のとき、発注量 200 個の定量発注方式として以下の問いに答えよ。

１）安全係数を求めよ。

$\lambda = normsinv(1-\alpha) =$ ［ 1.645 ］

２）サイクル在庫、安全在庫、理論在庫を求めよ。

サイクル在庫：$Q/2 =$ ［ 100 ］（個）

安全在庫：$\lambda\sigma\sqrt{L} =$ ［ 21.76 ］（個）

理論在庫：$Q/2 + \lambda\sigma\sqrt{L} =$ ［ 121.76 ］（個）

３）発注点在庫量を求めよ。

$I = L\mu + \lambda\sigma\sqrt{L} =$ ［ 161.76 ］（個）

これまでは Excel などを意識して説明をしてきましたが、C.Analysis でこれを実現することができます。[分析－ＯＲ－在庫管理]を選択すると、以下のような分析実行画面が表示されます。

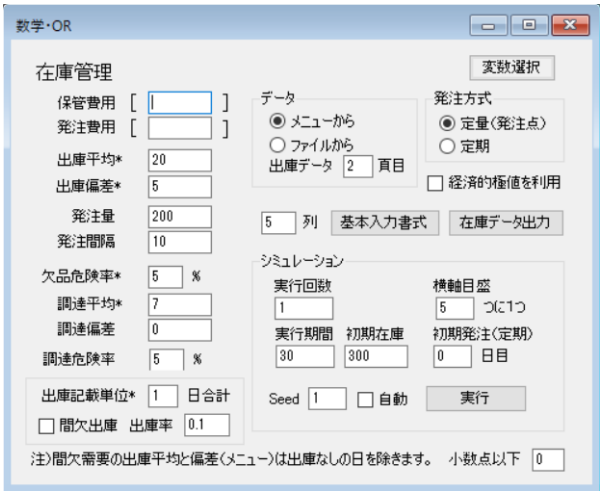


図 3　在庫管理実行画面

この画面に手入力する場合は、与えられたデータを左のテキストボックスに入力します。特に必要なデータは、左の項目名の後ろに「*」印を付けています。間違いのないよう入力し、右上の発注方式を「定量」として、「在庫データ出力」ボタンをクリックします。結果は以下のようになります。

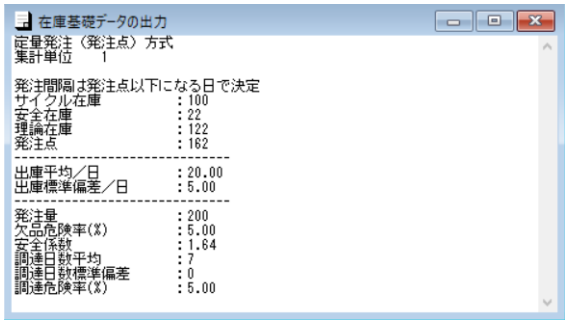


図４　在庫データ出力

以下の定期発注方式も同様に計算できます。

4.3　定期発注方式

定期発注方式は、予め発注間隔で定めた一定期間ごとに、必要な発注量を発注する方式です。基本は在庫量をできるだけ少なく保つことですが、発注した製品が届くまでの間に需要によって欠品が起こらないようにすることが重要なのは定量発注方式と同じです。予め決めた一定の発注間隔、出庫の平均と標準偏差、発注から入荷までの調達期間が分かっているとき、欠品の危険率（許容確率）を決めて、発注量をいくらにすればよいのかを求めます。定期発注方式に必要な情報とそれから決定される事項は以下の通りです。

必要な情報

発注間隔　　R（日）

調達期間（リードタイム）L（日）

１日当たりの出庫（確率的）　$N(\mu,\sigma^2)$

平均μ（個）、分散σ^2（標準偏差σ個）の正規分布を仮定

決定事項

<u>最大在庫量Mを求める。</u>

<u>発注量は　最大在庫量－発注時在庫量－発注残量</u>

ここで注意することは、在庫の量は今回発注を済ませたら次回入庫まで調整できない、ということです。まず今回の発注から次回の発注分の納品まで期間が$L+R$であることに注意します。この間に在庫不足の起こらないように、最大在庫量Mというものを求めます。発注量はそれから、現在の在庫量や発注残量を引いて求めます。

$M=(L+R)\mu+\lambda\sigma\sqrt{L+R}$　　λ：安全係数

発注量　$Q=M-I$　　$L>R$の場合は計算値から未納分を引いておく。

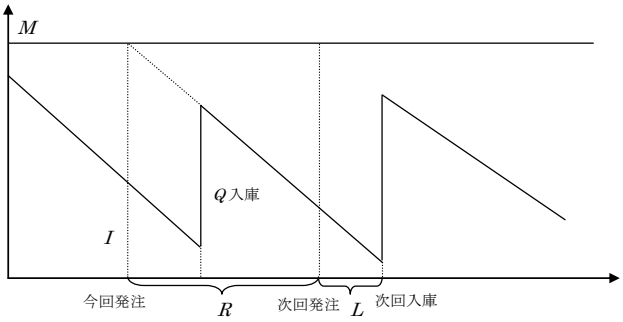


図 3　定期発注方式イメージ図 $L \le R$ の場合（$L \le R$ の場合）

公式　定期発注方式

１日当たりの出庫　平均 μ（個）、分散 σ^2（標準偏差 σ 個）の正規分布

発注間隔：R（日）　調達期間（リードタイム）：L（日）　品切れの危険率 $\alpha \times 100\%$

安全係数を $\lambda = normsinv(1-\alpha)$、最大在庫量を $M = (L+R)\mu + \lambda\sigma\sqrt{L+R}$ として、発注間隔ごとに、最大在庫量－現在の在庫量－現在の発注残量 を発注する。

サイクル在庫：$R\mu/2$、安全在庫：$\lambda\sqrt{L+R}\sigma$、理論在庫：$R\mu/2 + \lambda\sqrt{L+R}\sigma$

ここでサイクル在庫は発注間隔での平均需要の半分、安全在庫は発注が多かった場合に品切れを起こさないため上乗せする在庫量、理論在庫はその合計です。安全在庫は、欠品の危険率から求まる安全係数 λ を、発注から納品までの期間全体での需要の標準偏差 $\sigma\sqrt{L+R}$ に掛けて求めます。安全係数は標準正規分布の上側確率 α の値です。

例

1 日当たりの出庫の平均 20 個、標準偏差 5 個、欠品の危険率 5%以下、調達期間 7 日のとき、<u>発注間隔 10 日の定期発注方式</u>として以下の問いに答えよ。

１）安全係数を求めよ。

$\lambda = normsinv(1-\alpha)$ ＝［ 1.645 ］

２）最大在庫量を求めよ。

$M = (L+R)\mu + \lambda\sigma\sqrt{L+R}$ ＝［ 373.91 ］（個）

３）サイクル在庫、安全在庫、理論在庫を求めよ。

サイクル在庫：$R\mu/2$ ＝［ 100 ］（個）

安全在庫：$\lambda\sigma\sqrt{L+R}$ ＝［ 33.91 ］（個）

理論在庫：$R\mu/2 + \lambda\sigma\sqrt{L+R}$ ＝［ 133.91 ］（個）

４）発注日に前の発注の残りはあるか。　［ある・ない］

<u>C.Analysis でこれを実行する場合、図 3 の分析実行メニューで、「発注方式」を「定期」ラジオボタンに変えて、「在庫データ出力」ボタンをクリックします。</u>

問題

1 日当たりの出庫の平均 30 個、標準偏差 4 個、欠品の危険率 1%以下、調達期間 3 日のとき、以下の問いに答えよ。

<u>発注量 150 個の定量発注方式</u>

１）安全係数を求めよ。（定量でも、定期でも同じ）

$\lambda = normsinv(1-\alpha)$ ＝［　　］

２）サイクル在庫、安全在庫、理論在庫を求めよ。

サイクル在庫：$Q/2$ ＝［　　］（個）

安全在庫：$\lambda\sigma\sqrt{L}$ ＝［　　］（個）

理論在庫：$Q/2+\lambda\sigma\sqrt{L}$ ＝［　　］（個）

３）発注点在庫量を求めよ。

$I = L\mu + \lambda\sigma\sqrt{L}$ ＝［　　］（個）

<u>発注間隔 7 日の定期発注方式</u>

４）サイクル在庫、安全在庫、理論在庫を求めよ。

サイクル在庫：$R\mu/2$ ＝［　　］（個）

安全在庫：$\lambda\sigma\sqrt{L+R}$ ＝［　　］（個）

理論在庫：$R\mu/2+\lambda\sigma\sqrt{L+R}$ ＝［　　］（個）

５）最大在庫量を求めよ。

$M = (L+R)\mu + \lambda\sigma\sqrt{L+R}$ ＝［　　］（個）

６）発注日に前の発注の残りはあるか。　［ある・ない］　　ヒント：L<R

問題解答

１）安全係数を求めよ。

$\lambda = normsinv(1-\alpha)$ ＝［ 2.326 ］

２）サイクル在庫、安全在庫、理論在庫を求めよ。

サイクル在庫：$Q/2$ ＝［ 75 ］（個）

安全在庫：$\lambda\sigma\sqrt{L}$ ＝［ 16.12 ］（個）

理論在庫：$Q/2+\lambda\sigma\sqrt{L}$ ＝［ 91.12 ］（個）

３）発注点在庫量を求めよ。

$I = L\mu + \lambda\sigma\sqrt{L}$ ＝［ 106.12 ］（個）

４）サイクル在庫、安全在庫、理論在庫を求めよ。

サイクル在庫：$R\mu/2$ ＝［ 105 ］（個）

安全在庫：$\lambda\sigma\sqrt{L+R}$ ＝［ 29.43 ］（個）

理論在庫：$R\mu/2+\lambda\sigma\sqrt{L+R}$ ＝［ 134.43 ］（個）

５）最大在庫量を求めよ。

$M = (L+R)\mu + \lambda\sigma\sqrt{L+R}$ ＝［ 329.43 ］（個）

６）発注日に前の発注の残りはあるか。　［ある・ない］

在庫管理演習

演習 1

1 日当たりの出庫の平均 25 個、標準偏差 5 個、欠品の危険率 3%以下、調達期間 7 日のとき、以下の問いに答えよ。

発注量 250 個の定量発注方式

１）安全係数λを求めよ。　［　　］

２）サイクル在庫、安全在庫、理論在庫を求めよ。

サイクル在庫　［　　］（個）

安全在庫　［　　］（個）

理論在庫　［　　］（個）

３）発注点在庫量を求めよ。　［　　］（個）

発注間隔 12 日の定期発注方式

４）サイクル在庫、安全在庫、理論在庫を求めよ。

サイクル在庫　［　　］（個）

安全在庫　［　　］（個）

理論在庫　［　　］（個）

５）最大在庫量を求めよ。　［　　］（個）

６）発注日に前の発注の残りはあるか。　［ある・ない］　ヒント：L<R

演習２

出庫は 1 日当たり平均 30 個、標準偏差 6 個、欠品の危険率は 1%以下、調達期間 10 日として、以下の問いに答えよ。

発注量 400 の定量発注方式

１）安全係数λを求めよ。　［　　］

２）サイクル在庫、安全在庫、理論在庫を求めよ。

サイクル在庫　［　　］（個）

安全在庫　［　　］（個）

理論在庫　［　　］（個）

３）発注点在庫量を求めよ。　［　　］（個）

発注間隔 7 日の定期発注方式

４）サイクル在庫、安全在庫、理論在庫を求めよ。

サイクル在庫　［　　］（個）

安全在庫　［　　］（個）

理論在庫　［　　］（個）

５）最大在庫量を求めよ。　［　　］（個）

６）発注日に前の発注の残りはあるか。　［ある・ない］　ヒント：L>R

演習３

出庫は 1 週間当たり平均 200 個、標準偏差 20 個、欠品の危険率 5%以下、調達期間 1 週間として、以下の問いに答えよ。

１）1 日当たりの出庫の平均と標準偏差を求めよ。

ヒント：標準偏差の場合の計算は　1 週間の値÷$\sqrt{7}$

平均［　　］　　標準偏差［　　］

発注量 500 個の定量発注方式

２）安全係数λを求めよ。　［　　］

３）サイクル在庫、安全在庫、理論在庫を求めよ。

サイクル在庫　［　　］（個）

安全在庫　　［　　］（個）

理論在庫　　［　　］（個）

４）発注点在庫量を求めよ。　［　　］（個）

発注間隔 10 日間の定期発注方式

５）サイクル在庫、安全在庫、理論在庫を求めよ。

サイクル在庫　［　　］（個）

安全在庫　　［　　］（個）

理論在庫　　［　　］（個）

６）最大在庫量を求めよ。　［　　］（個）

７）発注日に前の発注の残りはあるか。　［ある・ない］

演習１解答

１）安全係数λを求めよ。　［ 1.88 ］

２）サイクル在庫、安全在庫、理論在庫を求めよ。

サイクル在庫　［ 125 ］（個）

安全在庫　　［ 25 ］（個）

理論在庫　　［ 150］（個）

３）発注点在庫量を求めよ。　［ 200 ］（個）

４）サイクル在庫、安全在庫、理論在庫を求めよ。

サイクル在庫　［ 150 ］（個）

安全在庫　　［ 41 ］（個）

理論在庫　　［ 191 ］（個）

５）最大在庫量を求めよ。　［ 516 ］（個）

６）発注日に前の発注の残りはあるか。　［ある・ない］　　ヒント：L<R

演習２解答

１）安全係数λを求めよ。　［ 2.33 ］

２）サイクル在庫、安全在庫、理論在庫を求めよ。
　サイクル在庫　［ 200 ］（個）
　安全在庫　　［ 44 ］（個）
　理論在庫　　［ 244 ］（個）
３）発注点在庫量を求めよ。　［ 344 ］（個）
４）サイクル在庫、安全在庫、理論在庫を求めよ。
　サイクル在庫　［ 105 ］（個）
　安全在庫　　［ 58 ］（個）
　理論在庫　　［ 163 ］（個）
５）最大在庫量を求めよ。　［ 568 ］（個）
６）発注日に前の発注の残りはあるか。　［ある・ない］　ヒント：L>R

演習３解答
１）１日当たりの出庫の平均と標準偏差を求めよ。
　平均［ 28.57 ］　標準偏差［ 7.56 ］
２）安全係数λを求めよ。　［ 1.64 ］
３）サイクル在庫、安全在庫、理論在庫を求めよ。
　サイクル在庫　［ 250 ］（個）
　安全在庫　　［ 33 ］（個）
　理論在庫　　［ 283 ］（個）
４）発注点在庫量を求めよ。　［ 233 ］（個）
５）サイクル在庫、安全在庫、理論在庫を求めよ。
　サイクル在庫　［ 143 ］（個）
　安全在庫　　［ 51 ］（個）
　理論在庫　　［ 194 ］（個）
６）最大在庫量を求めよ。　［ 537 ］（個）
７）発注日に前の発注の残りはあるか。　［ある・ない］

７．スケジュール管理

7.1　スケジュール管理問題とは

製品の製造や建築、イベントの準備など、１つのプロジェクトは多くの作業（工程）で組み立てられています。このようなプロジェクトに対して、完成まで最短何日かかるのか、各作業はいつから始められるのか、どのくらい遅れが許されるのか、各作業の所要日数に統計的なバラツキがある場合はどう考えるのか、各作業に使える人員に制限がある場合はどう扱うのか、各作業に費用をかけて完成までの日数を短くできる場合の費用と日数の関係はどうなるのかなど、いろいろな疑問について考えるのがスケジュール問題です。

このスケジュール問題の解決法には有名な PERT や CPM（費用を含む問題）、またそれらを拡張した方法が考案されていますが、ここではその中で最も基本的で重要な PERT について詳しく見て行くことにします（私自身不勉強でこれしか知らない）。

以下に例として家屋建築の作業リストを示します。

表１　家屋建築の作業リスト

作業	先行作業	所要日数	作業内容
A		7	設計
B	A	3	地盤工事
C	B	5	基礎工事
D	A	6	資材調達
E	C	3	屋根工事
F	C,D	6	外壁・防水工事
G	E	4	床面工事
H	F,G	5	内壁工事
I	E	3	ガス・水道工事
J	H,I	2	電気工事
K	F,G	10	仕上工事

この作業リストは、ある作業を始める直前にどの作業が終了していなければならないかを表す先行作業と、それぞれの作業の所要日数で表されています。PERT ではこれら２つの要素の他は考えません。右端に書いた作業内容は理解を助けるためのものです。

7.2　アローダイアグラム

スケジューリング問題の手法では、多くの場合、事業をアローダイアグラムと呼ばれる図で表します。アローダイアグラムでは作業（job）を１つの矢印で表し、作業から他の作業へ移る点を結合点（node）と呼び、○印で表します。以下にこれらの図と名前を描いておきます。

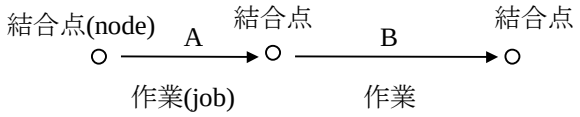


図１　アローダイアグラムの名称

次に、このアローダイアグラムを描くための規則について説明します。アローダイアグラ

ムを描く際には以下のことを守って下さい。

１）矢印の長さと作業時間は無関係である。

２）隣り合った２つの結合点間は１つの作業で結ぶ。（図 2）

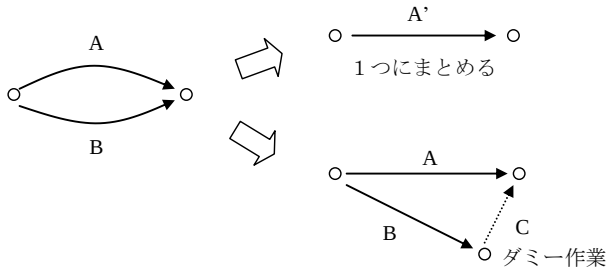


図 2　同一結合点間の 2 つの作業の表示法

同一結合点間の 2 つの作業は左のように 2 重に描かず、1 つにまとめるか、ダミー作業を含めて表示します。ダミー作業は作業時間（所要日数）0 の形式的な作業で、点線で表示します。ダミー作業は作業リストには書きません。

３）同じ結合点から出る（に入る）作業は共通の先行作業（後続作業）を持つ（図 3）。

例えば右のようなダイアグラムでは、C と D が同じ結合点から出ていますので、同じ先行作業 A と B を持つことになります。これは作業リストからすると、同じ先行作業を持つ作業は同じ結合点から出してよいとも解釈できます。

作業	先行作業	後続作業
A		C,D
B		C,D
C	A,B	
D	A,B	

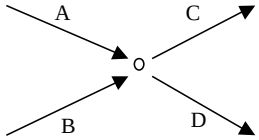


図 3　結合点と先行・後続作業

この規則を言い換えると（対偶に当たります）、同じ先行作業を持たない作業は同じ結合点から出してはいけない、となります。以下の作業リストのように、C と D は違った先行作業を持っていますので、ダイアグラムでは同じ結合点から出ることはできません。以下のようにダミー作業を入れるなどの処置が必要です。

作業	先行作業
A	
B	
C	A
D	A,B

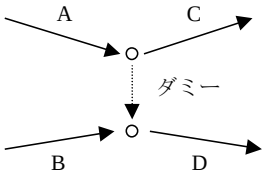


図 4　結合点と先行・継続作業（対偶）

４）ループがあってはならない。

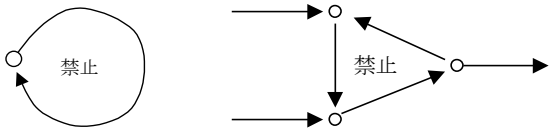


図 5　ループの禁止

５）プロジェクトの始点と終点を１つの結合点にまとめる（図 6)。

作業	先行作業
A	
B	
C	A
D	A,B

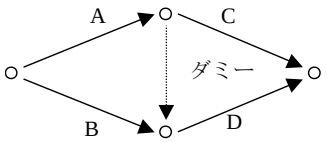


図 6　開始と終了の処理

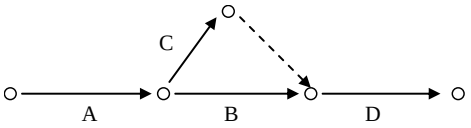
開始と終了は１つずつの結合点で表します。

これらの規則を使って、以下の問題を解いてみましょう。

問題　以下のプロジェクトをアローダイアグラムで表せ。

１）

作業	先行作業
A	
B	A
C	A
D	B,C



これは B と C が同じ開始点と終了点にならないような処理です。

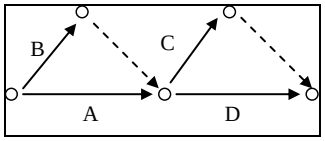
２）

作業	先行作業
A	
B	
C	A,B
D	A,B

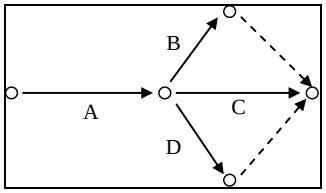
３）

作業	先行作業
A	
B	A
C	A
D	A

問題 1 解答

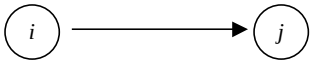


問題 2 解答



3. アローダイアグラムの例

始点を 1 として各作業の両端の結合点に対して、



$i < j$ となるように順番に番号を付けます（ソフトを使う際にはあまりこだわりません）。アローダイアグラムは以下のような表示になります。

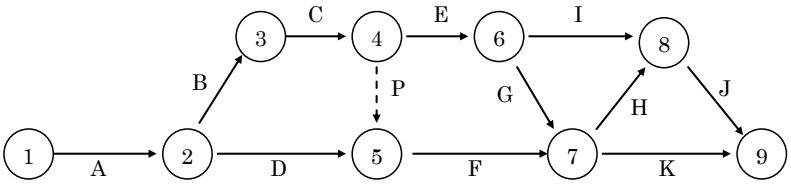


図 7　アローダイアグラム

7.3　PERT の手法

ここではスケジュール管理法としてよく利用されている PERT（Program Evaluation and Review Technique）について考え方を学びます。PERT は作業時間に基づくスケジュール管理手法です。1 節で説明に利用した家屋建築の作業リストを元に PERT の用語と手法について説明して行きます。確認が楽なように、以下に再度作業リストを示しておきます。

表 1　家屋建築の作業リスト

作業	先行作業	作業時間	作業内容
A		7	設計
B	A	3	地盤工事
C	B	5	基礎工事
D	A	6	資材調達
E	C	3	屋根工事
F	C,D	6	外壁・防水工事
G	E	4	床面工事
H	F,G	5	内壁工事
I	E	3	ガス・水道工事
J	H,I	2	電気工事
K	F,G	10	仕上工事

これらの作業をアローダイアグラムで表すと以下のようになります。

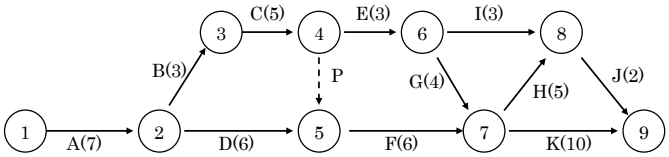


図 1　家屋建築のアローダイアグラム

これから各結合点に作業の開始や終了を表す時刻を記入して行きます。結合点に与えられる時刻で重要なものは、最早結合点時刻（Earliest Node Time）と最遅結合点時刻（Latest Node Time）と呼ばれるものです。前者は各結合点で最も早く次の作業に移れる時刻です。その結合点から出る作業は、最早結合点時刻になると始めることができます。後者はプロジェクトを遂行時間で仕上げるために、各結合点で遅くとも次の作業に移らなければならない時刻です。つまり先行作業は遅くともこの時刻までに終わっていなければなりません。また、この時刻を過ぎて次の作業を始めたら、最終的な納期に遅れが出ます。これらは以下のように求めることができます。ここではまず、どんな感じに描かれるかイメージを掴んでもらえば結構です。詳細は問題でやり、余力があればここに戻って詳しく見て行って下さい。。

1)　最早結合点時刻

始点での開始時刻 0 から始めて、順次時刻を足していきますが、これは各結合点に入る作業（ダミー作業も含む）のうちの最も遅い終了時刻をとるようにして求めます。これを使うと各結合点の最早結合点時刻は以下のようになります。これについては問題 1 のところで詳しく説明します。

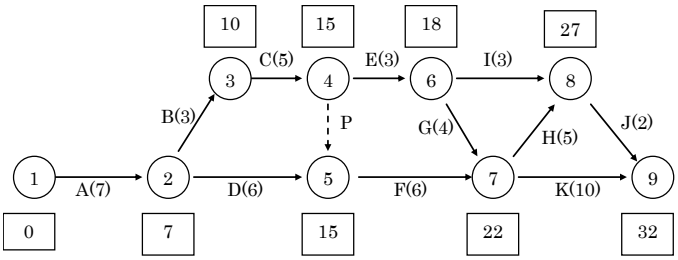


図 2　最早結合点時刻

ここで、終点の最早結合点時刻 32 はプロジェクト遂行時間と呼ばれます。

2)　最遅結合点時刻（Latest Node Time）

これは出て行く作業（ダミー作業も含む）について、開始時刻の最小のものをとります。これを使って最遅結合点時刻を最早結合点時刻の下に描くと、以下のようになります。これは結構難しいので詳細は問題 1 を見て下さい。

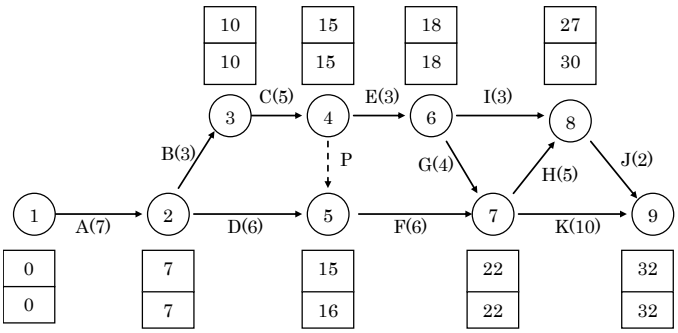


図 3　最遅結合点時刻

3)　クリティカル パス

これらの作業の中で、プロジェクト遂行時間でプロジェクトを終了するために、遅れることのできない作業があります。これをつないだパスをクリティカル パスといいます。これは以下のように求めることができます。即ち、最遅結合点時刻＝最早結合点時刻となる２つの結合点を通り、始点の最早結合点時刻＋作業時間＝終点の最早結合点時刻、になる作業を選んでつないでいきます。以下にクリティカル パスを太線で描いて示します。

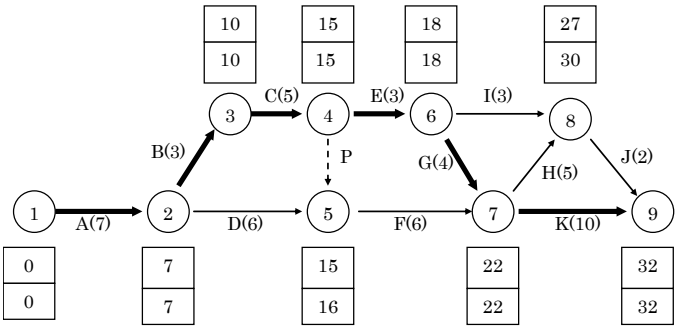
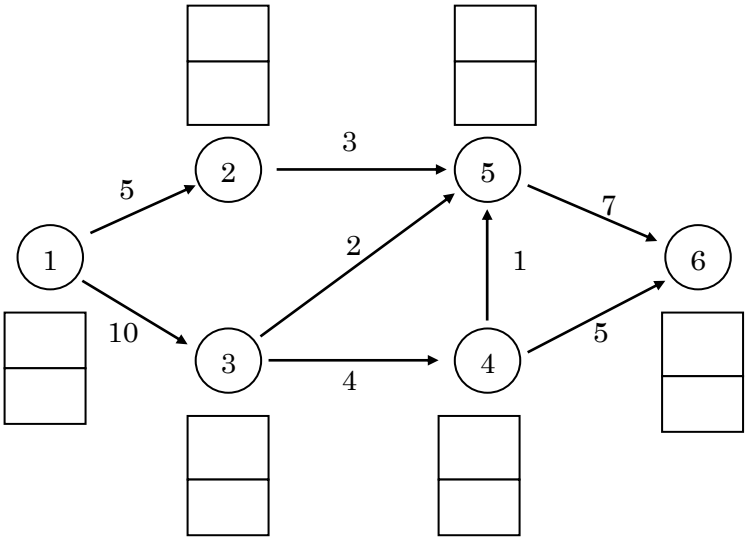


図 4　クリティカル パス

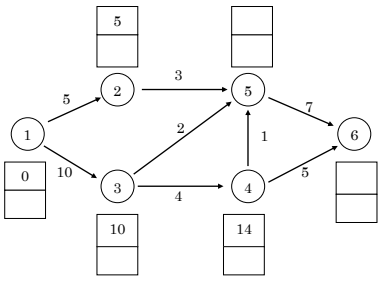
問題 1 （PERT2.txt）

以下のプロジェクトの最早結合点時刻と最遅結合点時刻を求め、クリティカルパスを示せ。

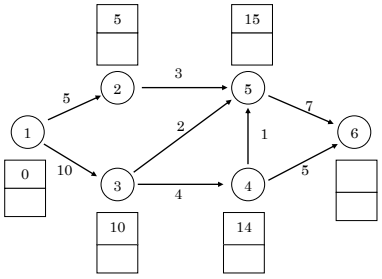


問題 1 解答

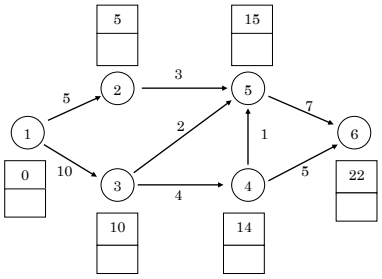
上の四角の最早結合点時刻から書き込んでいきます。まず始点①の四角に 0 を書きます。次に②の結合点は、①からの作業時間 5 の作業だけが入ってきますので、0+5=5 を記入します。また同じく③の結合点は、0+10=10 を記入します。さらに④は 10+4=14 です。これらはそれぞれ 1 つの作業が入ってくる結合点です。



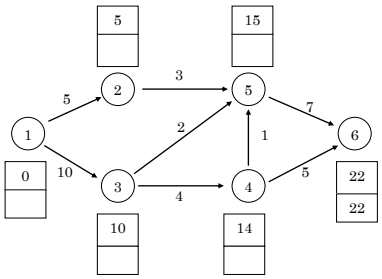
次は⑤ですが、これは３つの作業が入ってきます。②からは 5+3=8、③からは 10+2=12、④からは 14+1=15 です。これらがすべて済んでいないと次の作業は始められませんので、一番大きな 15 をとります。



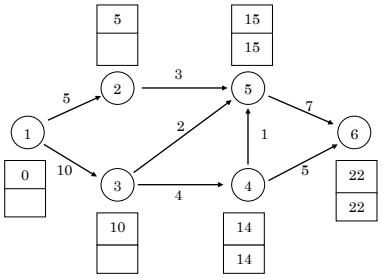
最後に、⑥については、⑤から 15+7=22、④から 14+5=17=19 であり、大きい値をとって 22 となります。これでプロジェクト遂行時間が決まりました。



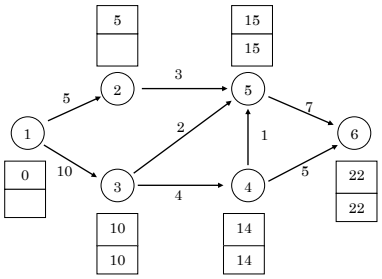
今度はこれを逆にたどって最遅結合点時刻を求めてみましょう。まず終点⑥の最早結合点時刻 22 の値を下の四角に移します。



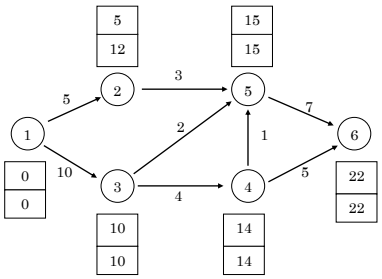
⑤の結合点は⑨への作業しか出て行きませんので 22-7=15 です。④の結合点は⑥への作業から 22-5=17、⑤への作業から 15-1=14 となります。ここで、17 まで待っていたら、⑤の 15 には間に合わないので、この場合は一番小さな値をとります。結局④の最遅結合点時刻は 14 です。



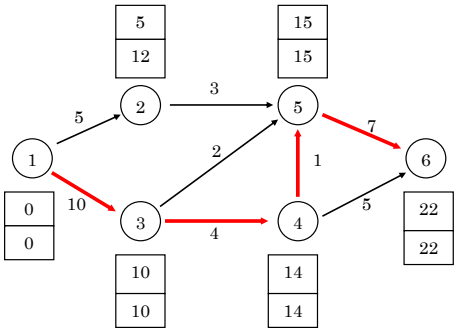
次に③ですが、これは 2 つの作業が出て行っています。⑤への作業では 15-2=13、④への作業では 14-4=10 となり、小さい方を選んで 10 となります。



次に②では、⑤への作業だけですので、15-3=12 となります。最後に①ですが、②への作業から 12-5=7、③への作業から 10-10=0 となり、小さい方をとって 0 となります。始点①の最遅結合点時刻が 0 にならなかったら、どこかに間違いがありますので再検討して下さい。

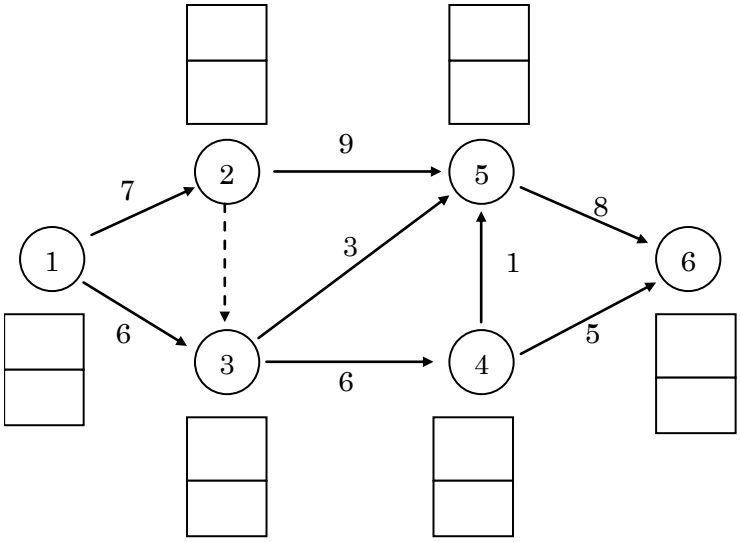


最後はクリティカルパスです。これは、最早結合点時刻と最遅結合点時刻が等しい結合点を通る作業の中で、始点の最早結合点時刻＋作業時間＝終点の最早結合点時刻となる作業を選びます。例えば、③から出る作業では、⑤へ向かう作業は 10+2≠15、④へ向かう作業は 14+1=15 となり、④へ向かう作業がクリティカルパスとなります。このようにして求めたクリティカルパス上の作業は、遅れることのできない大事な作業とみなされます。

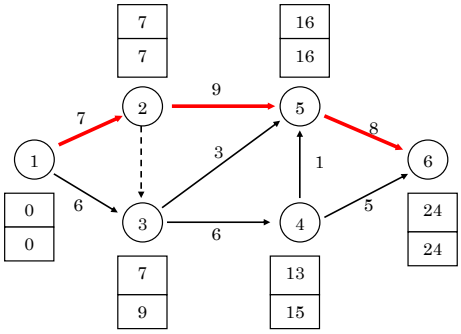


問題２（PERT3.txt）

以下のプロジェクトの最早結合点時刻と最遅結合点時刻を求め、クリティカルパスを示せ。



問題２解答

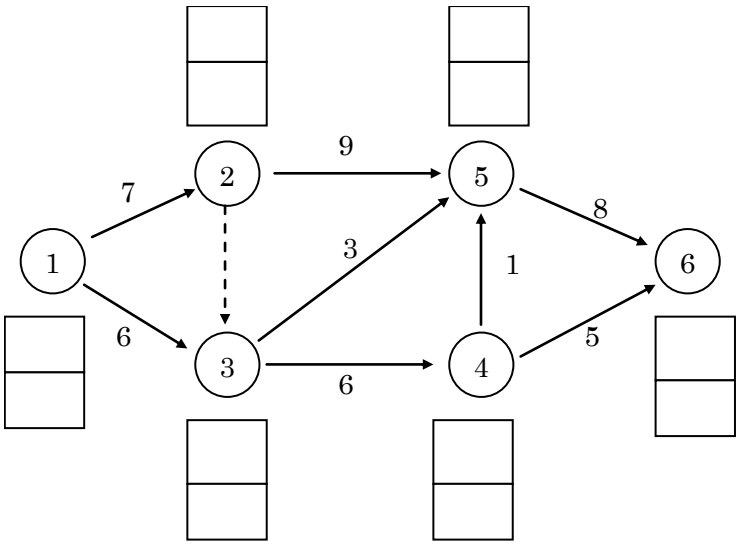


パソコンの利用　【動画】

これまでは PERT の考え方を知るために、紙面上で計算する方法を学んできました。しかし、複雑な問題が出てくるかも知れません。そのような問題に対処できるように、パソコンを使って解く方法も学んでおきましょう。ここでは、アローダイアグラムが分かっているものとして実行しますが、作業リストからアローダイアグラムを生成することもできます。

例（パソコンを使って）

以下のプロジェクトの最早結合点時刻と最遅結合点時刻を求め、クリティカルパスを示せ。



これは前回やった問題です。C.Analysis のメニュー［分析－OR－RERT］を選択すると、実行画面が表示されますが、今回はグラフィックエディタという機能で図を描いて、結果を求めるので、「グラフィックデータから」ラジオボタンをクリックします。結果は以下のようになります。

図 1　分析実行画面

ここで、「グラフィックエディタ」ボタンをクリックすると以下のような画面になります。

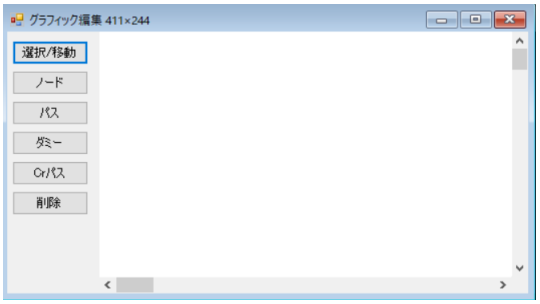


図 2　グラフィックエディタ

まず「ノード」ボタンをクリックした後、画面上をクリックしてノード（結合点）を配置します。

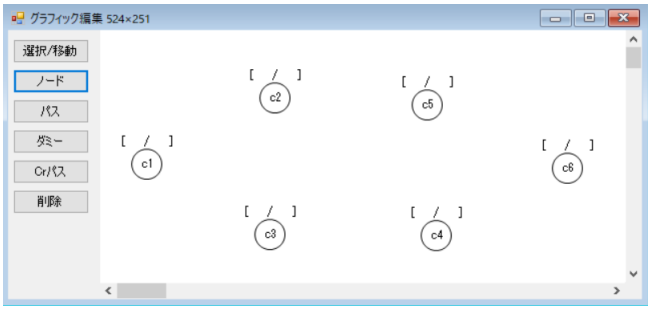


図 3　結合点の配置

Window 画面は大きくすることもできますし、ノードはドラッグすることで動かすこともできます。次に、「パス」ボタンをクリックし、ノード間をドラッグして図のようにパスを引いて行きます。失敗してノードやパスを消したい場合は、「削除」ボタンをクリックし、削除したい部品をクリックします。

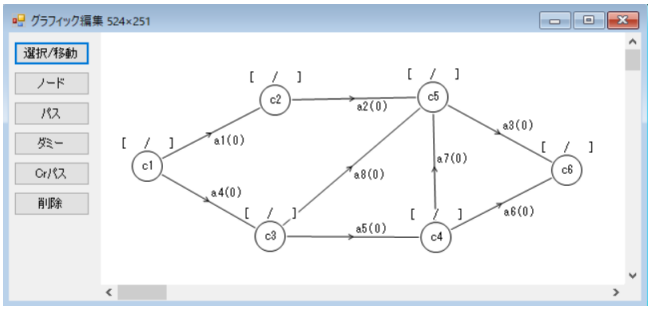


図 4　パスの追加

「パス」ボタンが選択されているときは、ノードが動かせませんので、「選択/移動」ボタンなどに切り替えて動かしながら、きれいに配置して下さい。「選択/移動」ボタンが選択されているときは、範囲をドラッグするとその中に含まれるノードがすべて赤色に変わり、一度に移動させることができます。試してみて下さい。最後に「ダミー」ボタンをクリックし、パスと同様にダミー作業を追加します。

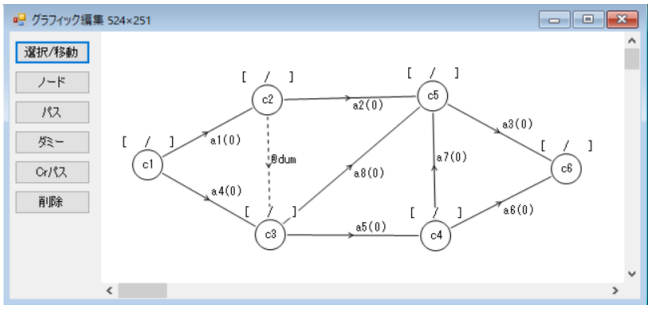


図 5　ダミーの追加

次に作業時間を入れてみましょう。パスの途中に、「a1(0)」などというのが描かれていますが、これは、作業名(作業時間)です。作業名はパスの上（文字の上ではない）をダブルクリックして、作業時間はパスの上を右クリックして変更できます。作業名は何でも構わないので、作業時間を変更してみます。

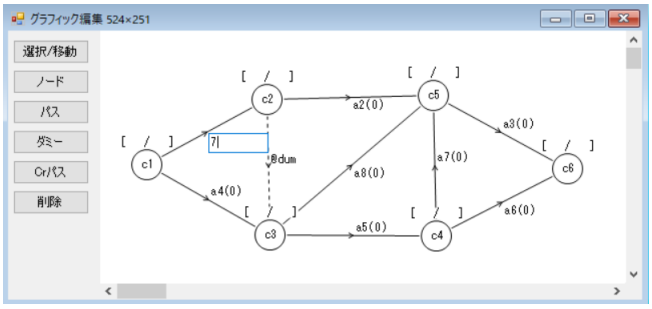


図 6　作業時間の編集

訂正が終わったら、他の部分をクリックします。このようにしてすべての作業時間を以下のように変更します。

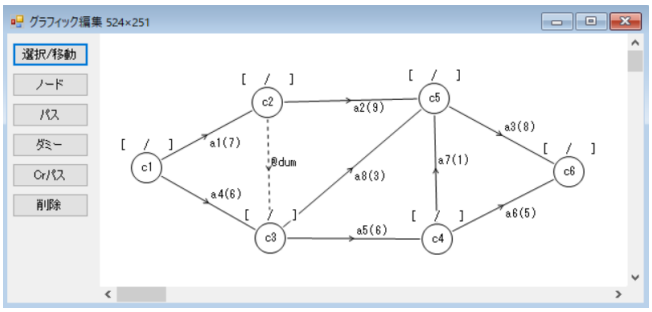


図 7　アローダイアグラム

今までは、Window ごとコピーして貼り付けてきましたが、グラフィックエディタのメニュー［編集－画面コピー］で中の図だけをコピーすることができます。以後はその方法を使います。

　ダイアグラムが完成したら、実行画面の「最早結合点時刻」をクリックして最早結合点時刻を表示してみましょう。ノードの上に表示されます。また、同様に「最遅結合点時刻」をクリックして最遅結合点時刻も表示してみましょう。

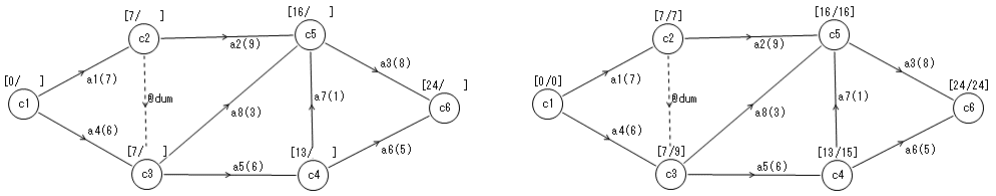


図 8　最早結合点時刻と最遅結合点時刻

最後に、「クリティカルパス」ボタンをクリックすると以下のように表示されます。

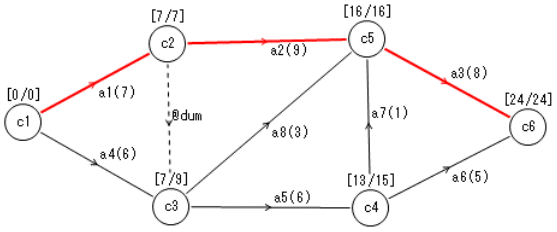
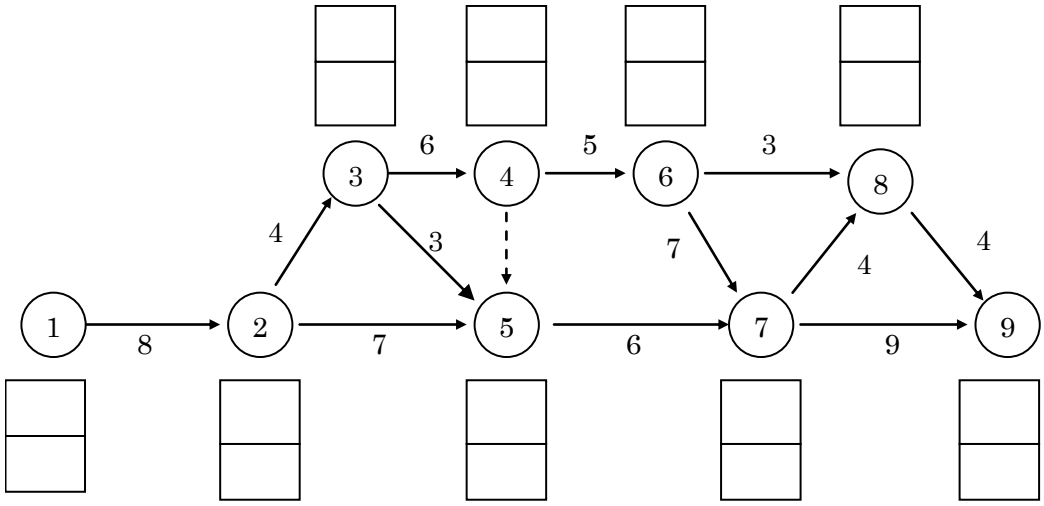


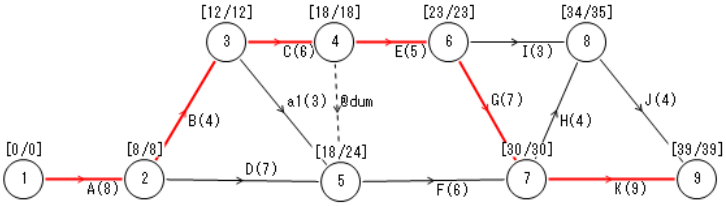
図 9　クリティカルパス

問題

以下のプロジェクトの最早結合点時刻と最遅結合点時刻を求め、クリティカルパスを示せ。



問題解答



4)　日程計画　【動画】

求められたアローダイアグラムから、工程を日程表として 1 つの表にまとめることは、一目で工程を見渡すために重要なことです。今回はその日程計画の話をします。

日程表に書く指標について、一般的に書くと難しく見えるので、仮にアローダイアグラムの一部を抜き出して、ノード⑤から⑨をつなぐ作業 C について説明します。

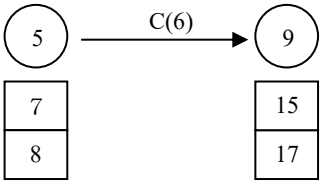


図 1　結合点の関係の例

これまでは結合点の時刻について話をしましたが、日程表は作業についての指標です。これを忘れないで下さい。以下は作業 C についての指標です。

最早開始時刻　　7

作業 C を最も早く開始できる時刻＝⑤の最早結合点時刻＝7

最早終了時刻　　7+6=13

作業 C を最も早く終了できる時刻＝⑤の最早結合点時刻＋作業時刻

最遅開始時刻　　17-6=11

作業 C を遅くとも開始しなければならない時刻＝⑨の最遅結合点時刻－作業時間

最遅終了時刻　　17

作業 C を遅くとも終了しなければならない時刻＝⑨の最遅結合点時刻

フリーフロート　　15-7-6=2

後続の作業に影響を与えないゆとり

＝⑨の最早結合点時刻－⑤の最早結合点時刻－作業時間

トータルフロート　　17-7-6=4

最大限許されるゆとり＝⑨の最遅結合点時刻－⑤の最早結合点時刻－作業時間

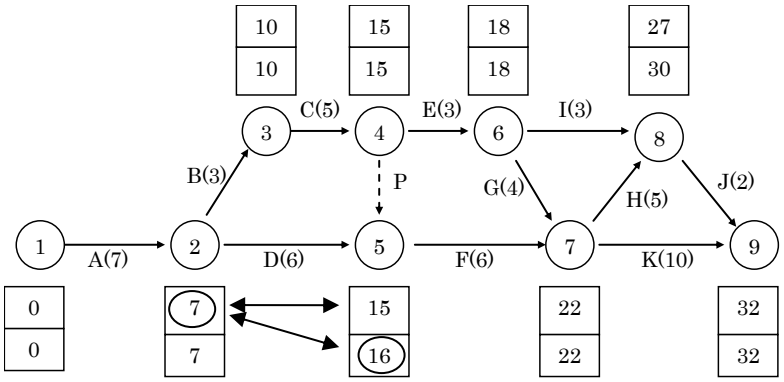
フリーフロート≦トータルフロート

ここでトータルフロート（最大限許されるゆとり）が 0 になるパスがクリティカルパスになります。

最後に、以下の例を用いた日程表を示しておきます。

例

作業	作業内容	先行作業	作業時間
A	設計		7
B	地盤工事	A	3
C	基礎工事	B	5
D	資材調達	A	6
E	屋根工事	C	3
F	外壁・防水工事	C,D	6
G	床面工事	E	4
H	内壁工事	F,G	5
I	ガス・水道工事	E	3
J	電気工事	H,I	2
K	仕上工事	F,G	10



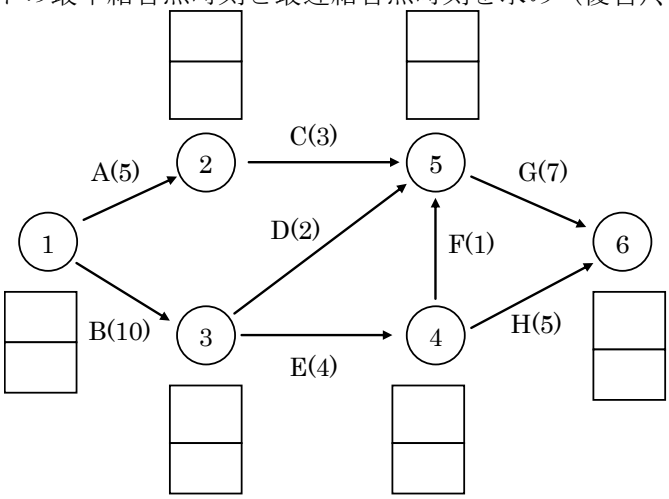
日程表

作業	最早開始時刻	最早終了時刻	最遅開始時刻	最遅終了時刻	フリーフロート	トータルフロート	クリティカル・パス
A(7)	0	7	0	7	0	0	1
B(3)	7	10	7	10	0	0	1
C(5)	10	15	10	15	0	0	1
D(6)	7	13	10	16	2	3	0
E(3)	15	18	15	18	0	0	1
F(6)	15	21	16	22	1	1	0
G(4)	18	22	18	22	0	0	1
H(5)	22	27	25	30	0	3	0
I(3)	18	21	27	30	6	9	0
J(2)	27	29	30	32	3	3	0
K(10)	22	32	22	32	0	0	1

クリティカル・パス上の作業は遅れに十分注意する。

問題（PERT2.txt）

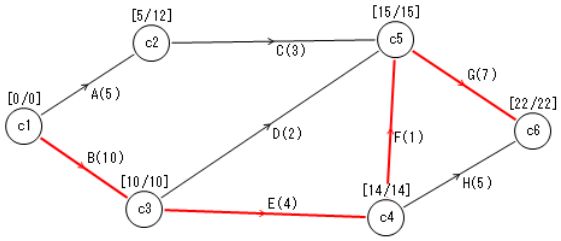
以下のプロジェクトの最早結合点時刻と最遅結合点時刻を求め（復習）、日程表を作れ。



日程表

作業	最早開始時刻	最早終了時刻	最遅開始時刻	最遅終了時刻	フリーフロート	トータルフロート	クリティカル・パス
A(5)							
B(10)							
C(3)							
D(2)							
E(4)							
F(1)							
G(7)							
H(5)							

例解答　パソコンでの解答を示します。



最後の日程表は、「作業名順」ラジオボタンを選んで、「日程表」ボタンをクリックして得られます。

日程表

	作業時間	最早開始	最早終了	最遅開始	最遅終了	Free Float	Total Float	Critical Path
A	5	0	5	7	12	0	7	0
B	10	0	10	0	10	0	0	1
C	3	5	8	12	15	7	7	0
D	2	10	12	13	15	3	3	0
E	4	10	14	10	14	0	0	1
F	1	14	15	14	15	0	0	1
G	7	15	22	15	22	0	0	1
H	5	14	19	17	22	3	3	0