

Investigación Operativa: Trabajo Final de Curso



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Yimin Pan
Adnan Akortal
2019 Q2

Índice de Páginas

1. Introducción	3
2. Descripción de los modelos	6
2.1. CPM Básico (cpm.mod)	6
Juego de Prueba	7
2.2. CPM con costes (cpm2.mod)	8
Juego de Prueba	10
3. Usos y Aplicaciones	13
4. Juegos de Prueba en Casos Realistas	14
4.1. Construcción de una casa	14
4.2. Estabilización de 300 hectáreas de uva.	17
5. Frontera de Pareto	19
6. Salida del Modelo	20
7. Cambios a lo largo del proyecto	21
8. Competencia	22
8.1 Uso solvente de los recursos de la información	22
8.1.1 Case study de CPM-PERT	22
8.1.2 Caso de aplicación del modelo implementado	22
8.1.3 Otras Extensiones	23
8.2 Bibliografía	23

1. Introducción

La tarea de gestión y control de proyecto siempre ha sido un tema importante para las empresas, pues permiten que los trabajos se puedan llevar a cabo correctamente en el tiempo preestablecido. En esta práctica intentaremos resolver este problema formulándolo mediante programación lineal, traduciéndolo y resolverlo en AMPL.

Para esto, el proyecto lo veremos como un conjunto de actividades a realizar, cada uno con un tiempo de finalización diferente y con dependencias entre actividades, es decir, existen actividades que necesitan que otras actividades se acaben antes de comenzar este. De este modo, podemos representar el proyecto como un grafo dirigido, siendo las actividades los nodos y los arcos las dependencias.



Para entender este método, definiremos unos conceptos:

- Si pasa que en una actividad $S = 0$, entonces esta es crítica, por lo que formará parte del camino crítico (Critical Path). De allí también el nombre del método, buscaremos este camino ya que es la que nos indica el procedimiento por el cual podemos acabar el proyecto de forma más eficiente y rápida. La suma del tiempo que tarda en completarse las actividades en dicho camino nos marca el tiempo mínimo para completar el proyecto. Un cambio de tiempo de finalización en una de estas actividades producirá un cambio en el tiempo total del proyecto; si disminuye, entonces disminuirá el tiempo del proyecto, mientras que si aumenta entonces tendremos que volver a hacer los cálculos. Sin embargo, la situación es más flexible

para los nodos no críticos, si el tiempo de ejecución disminuyera para alguno de estos, este cambio no tendrá efecto en el tiempo del proyecto, mientras que un incremento puede haber consecuencias dependiendo de si supera su S (slack) o no, en caso afirmativo tendríamos que recalcular el camino .

2. Descripción de los modelos

2.1. CPM Básico (cpm.mod)

El modelo básico de CPM no considera demasiadas cosas, tan solo tiene un objetivo: encontrar el tiempo mínimo que necesitamos para completar todo el proyecto. La formulación es la siguiente:

$$\begin{aligned} & \text{minimize : } x_f = \text{final dummy activity} \\ & \text{st : } \forall (i \rightarrow j) \ x_j \geq x_i + t_i \\ & \quad \forall i \ x_i \geq 0 \end{aligned}$$

x_i tiempo de comienzo de actividad i

t_i tiempo para finalizar una actividad i

En este modelo tenemos que cada nodo representa una actividad y cada arco representa las precedencias, un arco de forma $(i \rightarrow j)$ implica que j ha de empezar cuando i esté terminado.

Hemos definido un set de nodos **Nodes** como la unión entre las actividades **Activities** a_i y un nodo dummy a_f llamado **Final_Activity** (un set de un solo elemento) que lo hemos conectado con las últimas actividades del proyecto, las que no tienen arcos salientes. De tal modo nos sirve para marcar el final del proyecto, si este acaba implicará que las últimas actividades también están acabadas. Por lo que nuestra optimización se basa en minimizar su tiempo de inicio x_f que corresponde también al tiempo total del proyecto.

Definimos un set de arcos **Arcs** para representar las precedencias de las actividades.

Tenemos un parámetro **Completion_Time** t_i definido sobre el set de nodos, que indica el tiempo que tardamos en acabar cada actividad, obviamente t_f es 0..

La variable de decisión será x_i que indica el tiempo el cual empezamos una actividad a_i .

La función a minimizar es: **Final_Activity_Time**: $\text{sum}\{i \text{ in } \text{Final_Activity}\} \ x[i]$;

- Esta representa que queremos minimizar el tiempo en que empieza la actividad final **Final_Activity** o lo que es lo mismo, el tiempo en el que terminamos el proyecto.

En este modelo solo tenemos una restricción: **Precedence_Relationship**:

- Esta se aplica a todos los arcos($i \rightarrow j$) de manera que $x[j] \geq x[i] + Completion_Time[i]$;
- Esto significa que toda actividad j debe empezar como mínimo después del inicio de la actividad i (el requisito de j) más el tiempo que se necesita para terminar la actividad i .

En la versión básica solo tenemos un objetivo a minimizar por lo que existe noción de solución óptima, con el cual obtenemos el mínimo para la función objetivo z cumpliendo con las restricciones de precedencia. Como hemos indicado anteriormente esta z coincide con la suma de los tiempos de las actividades en el camino crítico o que es lo mismo que cuándo empezamos la actividad dummy, x_f . Esto no implica que tengamos un único camino crítico, pero en ese caso el valor de la función objetivo también es el mismo.

Juego de Prueba

Para ver que el modelo funciona correctamente, escogemos un caso de proyecto reducido para facilitar el análisis y comprobar que funciona correctamente.

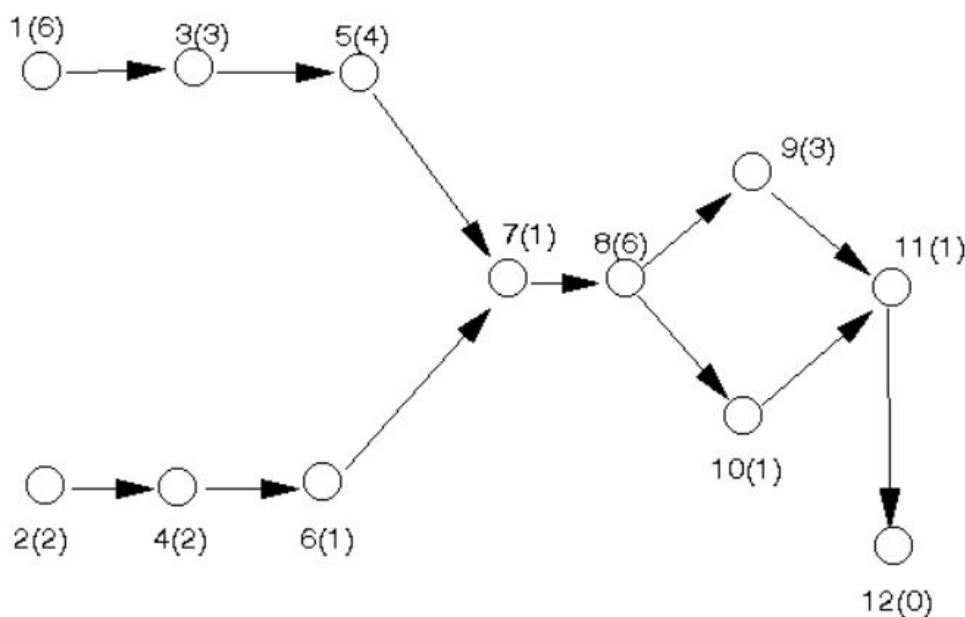


Figura 2: Juego de Prueba para CPM Básico

Como podemos ver fácilmente a ojo, el camino crítico viene marcada por 1-3-5-7-8-9-11-12, pues cuando tenemos varios caminos paralelos (caso de actividad que depende de más de una actividad previa, caso del nodo 7) elegimos el de mayor tiempo, así nos aseguramos suficiente tiempo para que el otro camino acabe también. De este modo, el tiempo mínimo para acabar el proyecto es

$6+3+4+1+6+3+1 = 24$. Que coincide con la salida del solver, por lo que parece que el modelo funciona correctamente.

```
MINOS 5.5: optimal solution found.
2 iterations, objective 24
x[total] = 24
```

Figura 3: Salida AMPL de CPM Básico

2.2. CPM con costes (cpm2.mod)

En la práctica se nos pide implementar un modelo más elaborado y cercano a la realidad. Desde del CPM básico añadimos un objetivo más, minimizar el coste del proyecto. Cabe decir que a parte de lo descrito en la práctica, el modelo que implementamos permite hacer crash de las actividades que es una situación bastante común en control y gestión de proyectos. Project Crashing quiere decir añadir más recursos (más costes) para acelerar el proceso de terminar una actividad o reducir requerimientos de las actividades, en nuestro caso solamente consideramos la primera posibilidad. De este modo el coste total incluye:

1. Coste de ejecución de las actividades. Calculado como sumatorio de tiempo crash * coste crash + tiempo restante * coste ejecución normal.
2. Coste de dirección del proyecto. Que es proporcional al tiempo total del proyecto, multiplicando a éste por el coste de dirección por unidad de tiempo.
3. Coste por interés de préstamo. En caso de que la suma de los costes anteriores superen al presupuesto inicial, podemos pedir préstamo del dinero que nos falta, pagando un tipo de interés. Por lo tanto se calcula multiplicando este interés a la cantidad prestada.

La formulación es el siguiente:

$$\text{Minimize : } w * x_f + (1 - w) * (CEje + CDir + CPre)$$

$$\text{CosteEjecución} = \sum ce_i + c_i * (cc_i - c_i) / (t_i - tc_i)$$

$$\text{CosteDirección} = x_f * cd$$

$$\text{CostePréstamo} = \max(0, (CEje + CDir - b_0) * r)$$

$$st : \forall (i \rightarrow j) x_j \geq x_i + t_i - c_i$$

$$\forall i c_i \leq t_i - mc_i$$

$$\forall i x_i \geq 0, c_i \geq 0$$

x_i tiempo de comienzo de actividad i

t_i tiempo para finalizar una actividad i

c_i tiempo de crash que decidimos para una actividad i

tc_i tiempo para terminar el proyecto si se aplica crash

ce_i coste de ejecución de toda la actividad i

cc_i coste de crash por unidad de tiempo de una actividad i

cd coste de dirección por unidad de tiempo del proyecto

b_0 presupuesto inicial

r tipo de interés del préstamo

w peso para equilibrar los dos objetivos

En el fichero cpm2.mod simplemente seguimos esta formulación. A partir de lo que tenemos en el cpm.mod añadimos los siguientes parámetros:

- **executionCost.** Definido sobre todos los nodos, que indica el coste de ejecución de una actividad.
- **crashCost.** Definido también sobre todos los nodos, que indica el coste de crash para una actividad.
- **crashCompletionTime.** Definido también sobre todos los nodos, indica el tiempo mínimo que tardamos en hacer esa actividad cuando le aplicamos crash (El tiempo mínimo se consigue haciendo el crash máximo).
- **directioningCost.** Coste de dirección del proyecto por unidad de tiempo.
- **b0.** Presupuesto establecido para el proyecto.
- **r.** Tipo de interés en caso de pedir un préstamo.
- **w.** Peso para balancear los diferentes objetivos.

Con la comprobación $crashCompletionTime[i] \leq completionTime[i]$; nos aseguramos que el tiempo de ejecución de crash de una actividad sea menor que el tiempo de finalización original de la actividad.

Tenemos una variable de decisión más **c** que nos indica cuanto “crasheamos” a cada actividad.

Las otras “variables” simplemente son para facilitar la lectura y sencillez del código, lo pudiera haber metido todo a la función objetivo. Son los diferentes costes:

- **Activity_Cost.** Es el coste asociado a la ejecución de todas las actividades. Se calcula como el sumatorio de los costes de todas las actividades, donde el coste de cada actividad viene definido de la siguiente manera:
 - Si la actividad no se puede crashear: $executionCost[i]$;
 - Sino, viene definido como coste de ejecutar la actividad + días reducidos * coste de reducir 1 día esa actividad: $(executionCost[i] + c[i]*(crashCost[i] - executionCost[i]) / (completionTime[i] - crashCompletionTime[i]))$;
- **Project_Directioning_Cost.** Coste de dirección del proyecto, calculado tiempo total requerido por el proyecto por el coste de dirección por unidad de tiempo.
 $sum\{f \text{ in } Final_Activity\} x[f]*directioningCost;$

- **Interest_Cost.** Coste del préstamo en caso de que lo necesitamos, es decir que la suma de los costes anteriores superen al presupuesto. En este caso aplicamos un tipo de interés a la diferencia entre coste y presupuesto, el dinero prestado. Cabe decir que en lugar de usar condicionales, es lo mismo implementarlo la función max, si **b0** es superior a los costes, entonces este término será negativo, y por max será 0.
 $\text{max}(0, (\text{Activity_Cost} + \text{Project_Directioning_Cost} - b0) * r);$
- **Total_Cost.** La suma de estos tres costes.
 $\text{Activity_Cost} + \text{Project_Directioning_Cost} + \text{Interest_Cost};$
- **Total_Time.** Simplemente es el tiempo total requerido por el proyecto.
 $\text{sum } \{f \text{ in } \text{Final_Activity}\} x[f];$

De restricciones solamente tenemos 2:

- Asociado a la dependencia de las actividades como habíamos explicado en el modelo básico, solo que ahora hay que considerar la reducción de tiempo debido a crashing.
Precedence_Relationship $\{(i,j) \text{ in } \text{Arcs}\}: x[j] \geq x[i] + \text{completionTime}[i] - c[i];$
- Asociado al upper bound del crash de las actividades.
Crash_Time_Constraint $\{i \text{ in } \text{Nodes}\}: c[i] \leq (\text{completionTime}[i] - \text{crashCompletionTime}[i]);$

Finalmente la función objetivo será tan simple como la suma de los dos objetivos, tiempo para terminar el proyecto y el coste total, aplicando los pesos para equilibrarlos ya que sus métricas pueden ser muy diferentes. De unidad de tiempo podríamos estar hablando de años mientras que de coste puede ser monetario, entonces sin equilibrarlos la aportación del tiempo será muy poco en comparación con coste, de este modo nos estamos centrando básicamente en minimizar coste.

minimize Time_Cost_Objective: $w * \text{Total_Time} + (1-w) * \text{Total_Cost};$

Juego de Prueba

Seguiremos el proyecto del juego de prueba para la versión básica añadiendo los datos que falta para esta versión.

Activity Number	Activity Name	Immediate Predecessor (list number/name, separated by ',')	Normal Time	Crash Time	Normal Cost	Crash Cost
1	1		6	4	\$100	\$240
2	2		2	1	\$100	\$150
3	3	1	3	3	0	0
4	4	2	2	2	0	0
5	5	3	4	2	\$100	\$180
6	6	4	1	1	0	0
7	7	5,6	1	1	0	0
8	8	7	6	3	\$100	\$160
9	9	8	3	2	\$100	\$140
10	10	8	1	1	0	0
11	11	9,10	1	1	0	0

Figura 3: Juego de Prueba para CPM con Costes

$w = 0.5$, $r = 0$, $b0 = 10000$, **directioningCost** = 10; Se fijan valores triviales para algunos de estos parámetros, para que sea más fácil de analizar la salida.

Dado que hemos puesto $w = 0.5$, no estamos equilibrando los objetivos, y como hemos dicho antes, estos tienen métricas diferentes. En este ejemplo el tiempo está en semanas, por lo que el valor será bastante más pequeño que los costes, de este modo, obviamente se intentará minimizar el coste consiguiendo un tiempo de proyecto sin hacer crash (más coste), el resultado deberá ser 24 semanas, mismo camino crítico que en el caso de versión básica sin crashing.

```

ampl: include ../IO_TrabajoFinal/CPM2.run;
MINOS 5.5: optimal solution found.
10 iterations, objective 382
Nonlin evals: obj = 7, grad = 6.
Total_Time = 24

Activity_Cost = 500

Project_Directioning_Cost = 240

Interest_Cost = 0

Total_Cost = 740

c [*] :=
1 0
2 0
3 0
4 0
5 0
6 0
7 0
8 0
9 0
10 0
11 0
12 0
;
```

Figura 4: Salida 1 para CPM con Costes

Sin embargo, si ahora modificamos el peso, $w = 0.99$; estamos dando mucha más prioridad a minimizar el tiempo total del proyecto. De esta forma intentará hacer todo el crash posible sobre las actividades del camino crítico.

```
ampl: include ../IO_TrabajoFinal/CPM2.run;
MINOS 5.5: optimal solution found.
6 iterations, objective 25.64
Nonlin evals: obj = 3, grad = 2.
Total_Time = 16

Activity_Cost = 820

Project_Directioning_Cost = 160

Interest_Cost = 0

Total_Cost = 980

c [*] :=
1 2
2 0
3 0
4 0
5 2
6 0
7 0
8 3
9 1
10 0
11 0
12 0
;
```

Figura 5: Salida 2 para CPM con Costes

Como hemos supuesto, se ha hecho el máximo de crashing posible sobre 1-5-8-9 que es el camino crítico. El valor de c para estas actividades ya no vale 0, sino que tiene como valor la cantidad de unidades de tiempo reducida en esa actividad. De este modo conseguimos terminar el proyecto en tan solo 16 semanas, pero el coste total ha subido de 740 a 980 por los crashings.

En realidad cuando estamos modificando el valor de w , nos estamos moviendo entre la frontera de Pareto. En el caso de optimización multiobjetivo no podemos decir que una solución sea mejor que otra si no hay mejoras en todos los objetivos. Cosa que pasa en las situaciones anteriores, cuando mejoramos uno, el otro empeora por los crashings, la prioridad dependerá de las necesidades del usuarios, motivo por el cual hemos puesto un parámetro de peso para los objetivos.

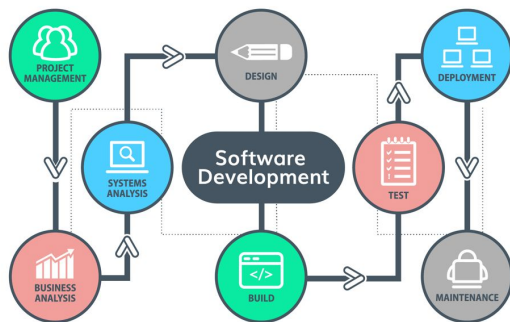
3. Usos y Aplicaciones

El método CPM es usado para el cálculo de plazos en la planificación de proyectos. El principal objetivo es determinar la duración de un proyecto, donde las actividades del mismo tienen una duración estimada y posibles precedencias.

De tal modo que si representamos el proyecto como un grafo dirigido, podemos encontrar un camino (critical path) que con la suma del tiempo requerido por las actividades en este camino, sabemos el tiempo mínimo requerido para completar el proyecto.

Este método fue desarrollado en el año 1950 por Morgan R. y James E. Kelley, y la primera aplicación fue para la reforma de una planta química. Basándose en casos de años anteriores, pudieron aplicar CPM, ya que era necesario saber el tiempo estimado para cada actividad, de ésta forma se redujo sustancialmente el tiempo de completar la reforma comparado con casos anteriores.

Hoy en día, este método tiene una extensa aplicación en todo tipos de proyectos:



Desarrollo de Software



Construcción



Aeronáutica



Proceso Industria

...etc.

4. Juegos de Prueba en Casos Realistas

4.1. Construcción de una casa

En este primer ejemplo **CPM2-House.dat** vamos a aplicar el modelo en la planificación de la construcción de una casa. Este esta basado en la empresa Angel Estates and Construction Ltd. en Ghana.

Esta es la tabla de actividades, el coste de cada una de ellas, sus predecesores y la duración estimada de cada una.

Activity Code	Activity Description	Normal cost (GH¢)	Immediate Predecessors	Estimated Duration
A	Site clearing	1,841.52	—	2 days
B	Foundation	1,264.34	A	4 days
C	Block Laying	5,281.16	B	10 days
D	Roofing	4,165.82	C	6 days
E	Plumbing	5,017.92	C	4 days
F	Electrical work	5,268.11	E	5 days
G	Plastering	6,389.43	D	7 days
H	Fixing up of doors and windows	3,590.55	E, G	9 days
I	Ceiling	4,099.73	C	7 days
J	Flooring	4,239.48	F, I	8 days
K	Interior Fixtures	4,764.31	J	4 days
L	Exterior fixtures	3,388.74	J	5 days
M	Painting	4,153.91	H	2 days
N	Landscaping	3,691.33	K, L	6 days

Figura 6: Información del proyecto (1)

A continuación tenemos la tabla donde para actividad tenemos el tiempo de finalización, el coste de ejecución, el tiempo de finalización si le aplicamos crash y el coste de la ejecución de la actividad al aplicar el crash.

Activity Code	Normal		Crash	
	Time (Days)	Cost (GH¢)	Time (Days)	Cost (GH¢)
A	2	1,841.52	1	2,433.37
B	4	1,264.34	2	2,113.74
C	10	5,281.16	7	7,049.89
D	6	4,165.82	4	4,996.76
E	4	5,017.92	3	5,597.21
F	5	5,268.11	3	6,978.68
G	7	6,389.43	4	8,281.45
H	9	3,590.55	6	5,352.24
I	7	4,099.73	5	4,987.63
J	8	4,239.48	6	5,316.11
K	4	4,764.31	3	5,221.87
L	5	3,388.74	3	4,709.21
M	2	4,153.91	1	4,780.65
N	6	3,691.33	3	4,968.25

Figura 7: Información del proyecto (2)

A continuación podemos ver como es el grafo de precedencias de estas actividades, donde la actividad **O** representa el final del proyecto.

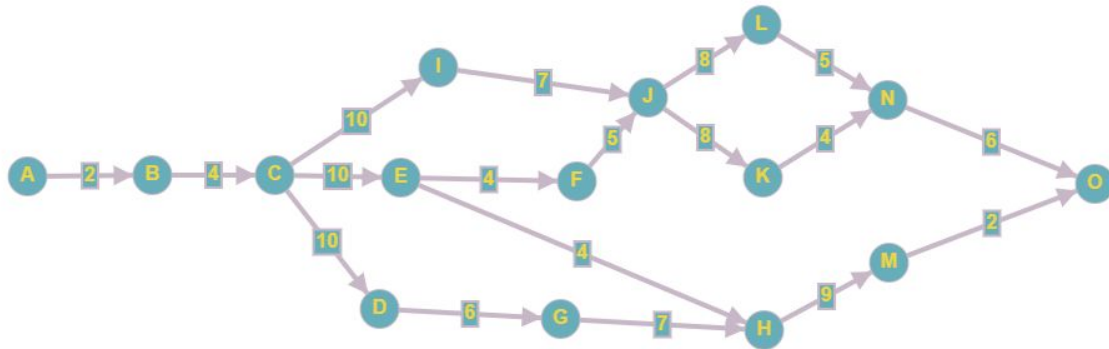


Figura 8: Grafo de precedencia

Para este proyecto no pondremos ni costes de direccionamiento ni un límite de presupuesto ya que no están estipulados. Además, realizaremos varias ejecuciones con **w** distintas para poder ver como se mueve el problema en la frontera de pareto. Recordemos que tal y como habíamos definido la función de minimización, una $w=0$ minimizaba el coste mientras que $w=1$ minimizaba el tiempo de ejecución y cualquier valor $0 < x < 1$ hace un balance entre ambos. En este caso, al ser el coste muy superior al tiempo de ejecución, hay que usar valores muy cercanos al 1 para ponderar mucho más el tiempo de ejecución.

W	0	0.997655	0.9977	0.999	1
Tiempo (días)	44	42	39	32	28
Coste (GH¢)	57156.3	58005.7	59282.7	64130.1	70455.1

Cantidad de días reducidos para cada actividad en funcion de la w:

W	0	0.997655	0.9977	0.999	1
1(A)	0	0	0	1	1
2(B)	0	2	2	2	2
3(C)	0	0	0	3	3
4(D)	0	0	0	2	2
5(E)	0	0	0	1	1

6(F)	0	0	0	0	2
7(G)	0	0	0	0	1
8(H)	0	0	0	0	3
9(I)	0	0	0	0	1
10(J)	0	0	0	2	2
11(K)	0	0	0	0	1
12(L)	0	0	0	0	2
13(M)	0	0	0	0	0
14(N)	0	0	3	3	3

4.2. Estabilización de 300 hectáreas de uva.

En este segundo ejemplo **CPM2-Grapes.mod** vamos a aplicar el modelo de planificación para la estabilización de 300 hectáreas de uva.

Appendix 1: The Activities list of Project Scheduling of 300 hectares Grape Garden Stabilization in Agricultural Research Center of

Activity	Predecessor	Normal time	Crash time	Crash time cost
1 Land Preparation, Soil Sampling and Analysis	...	36	19	4600000
2 Location Identification and Land Map Preparation	1	10	6	400000
3 Determination of Irrigation System Type	1, 2	7	4	3000000
4 Land Digging and Initial Fixation of Pipes and Links	1, 2, 3	33	17	450000
5 Continue to Fix Pipes and Links	1, 2, 3, 4	147	74	10950000
6 Main Pipe Burying and Primary Welding of Secondary Pipes	1, 2, 3, 4, 5	10	6	720000
7 Continue to Secondary Pipes Welding	1, 2, 3, 4, 5, 6	147	73	888000
8 Pipes Checkout and Burying	1, 2, 3, 4, 5, 6	21	10	420000
9 Land Surveying and Ripping for Windbreak Fixation	1, 2, 3	10	5	1648000
10 Ditch Fix and Windbreaks Drenching	1, 2, 3, 4, 6, 8, 9	14	8	125000
11 Plantation of Windbreak Seedlings	1, 2, 3, 4, 6, 8, 9, 10	5	3	120000
12 Plantation Lines Primary Survey	1, 2, 3, 9	2	1	39000
13 Continue to Plantation lines Survey	1, 2, 3, 9, 12	148	74	2886000
14 Primary Heavy and Semi Heavy Ripping	1, 2, 3, 9, 12	2	1	320000
15 Continue to Heavy and Semi Heavy Ripping	1, 2, 3, 9, 12, 14	198	99	31680000
16 Primary Location Survey and Digging of Pits	1, 2, 3, 9, 12, 14	4	2	439000
17 Continue to Pits Location Survey	1, 2, 3, 9, 12, 16	148	74	2886000
18 Continue to Pits Digging	1, 2, 3, 9, 12, 14, 16	148	74	29600000
19 Arable soil, Gravel and Fertilizer Primary Transfer to Farmland	1, 2, 3, 9, 12, 16	2	1	600000
20 Continue to Arable soil, Gravel and Fertilizer Transfer to Farmland	1, 2, 3, 9, 12, 16, 19	298	149	89400000
21 Primary Preparation of Mixed Soil	1, 2, 3, 9, 12, 16, 19	2	1	200000
22 Continue to Mixed Soil Preparation	1, 2, 3, 9, 12, 16, 19, 21	98	49	9800000
23 Mixed Soil Primary Transfer to Farmland and Primary Filling of Pits with them	1, 2, 3, 9, 12, 16, 19, 21	4	2	1200000
24 Continue to Mixed Soil Transfer to Farmland	1, 2, 3, 9, 12, 16, 19, 21, 23	298	149	119200000
25 Continue to Pits Filling with Mixed Soil	1, 2, 3, 9, 12, 16, 19, 21, 23	298	149	59600000
26 Primary Setting of Pipes (16inches) and Droppers	1, 2, 3, 9, 12, 16, 19, 21, 23	2	1	48000
27 Continue to Setting of Pipes (16inches) and Droppers	1, 2, 3, 9, 12, 16, 19, 21, 23, 26	73	36	1776000
28 Heavy Irrigation	1, 2, 3, 4, 6, 8, 9, 12, 16, 19, 21, 23, 26	60	60	0
29 Plantation of Grape Seedlings	1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 16, 19, 21, 23, 26, 28	38	19	9120000

Para esta planificación, los autores han puesto como coste de cada actividad = 0. Esto es debido a que el coste de la planificación no puede reducirse, y solo aumentar debido al crashing.

Veamos los costes monetarios y temporales asociados a esta planificación en función de W.

W	0	0.99999	0.999999	0.9999995	1
Tiempo (días)	419	376	343	234	214
Coste adicional (Rial)	0	713823.529	10374941.17	214454000	264082000

Como podemos ver, el tiempo mínimo que tardaríamos en este proyecto sin añadir dinero adicional es de 419. A partir de aquí, si ponemos un valor alto de la W vemos como con un poco mas de 700 mil riales podemos reducir el tiempo en 43 días. Si seguimos aumentando la W podemos reducir otros 33 días a cambio de aumentar el presupuesto adicional en mas de 14 veces. Si seguimos aumentando el presupuesto extra, vemos que con 214 millones conseguimos reducir el tiempo inicial en 185 días, un coste de crash diario mucho mas elevado que si el proyecto durara 376, donde cada día tendria una influencia mucho menor. Finalmente, si queremos reducir el tiempo del proyecto al máximo sin importarnos el presupuesto, vemos que el tiempo mínimo es de 214 días con un coste de más de 264 millones de riales.

5. Frontera de Pareto

Dado que tenemos varios objetivos a minimizar, tiempo y coste, pues no existe noción de solución óptima. Es difícil en este caso decir que una solución es mejor que otra si no hay mejoras en ambos objetivos, ya que a unos usuarios les importará más reducir el tiempo total y a otros gastar lo mínimo. Para esto hemos introducido una variable w (weight) que pondera esta preferencia. Pues se multiplica w al primer objetivo y $(1-w)$ al segundo, por este motivo w ha de estar entre $(0,1)$. Dicho esto, si modificamos el valor de w , estamos “viajando” en la frontera de Pareto, donde se encuentran las soluciones óptimas.

Por los experimentos, podemos ver que normalmente hay que asignar valores extremos a w , para movernos de una solución a otra. Esto puede que se deba a la gran diferencia entre métrica de tiempo (meses, semanas, días...) y el coste (suele ser valores elevados). Para el ejemplo default (cpm2.dat), no se consigue producir cambios con w entre $(0, 0.95)$ por la razón anterior, pues la unidad de tiempo está en semanas, por lo que suele ser valores como 1, 2, 3... mientras que los costes como mínimo es 100.

6. Salida del Modelo

Como el objetivo principal del problema, pretendemos que la salida del modelo nos indique claramente cual es el tiempo mínimo requerido para terminar el proyecto y los recursos económicos a gastar. De este modo, la salida del modelo consiste en dos partes.

La primera indica el tiempo de comienzo asignado a cada actividad y el tiempo en completar el proyecto.

```
Time: -----  
Activity 1 starts at: 0  
Activity 2 starts at: 8  
Activity 3 starts at: 6  
Activity 4 starts at: 10  
Activity 5 starts at: 9  
Activity 6 starts at: 12  
Activity 7 starts at: 13  
Activity 8 starts at: 14  
Activity 9 starts at: 17  
Activity 10 starts at: 19  
Activity 11 starts at: 20  
==> Minimum completion time for the project: 21
```

Figura 9: Screenshot de la salida (1)

La segunda nos da información sobre el crash de cada actividad y los costes económicos debidos a: ejecución de la actividad, dirección del proyecto y préstamos.

```
Crushes: -----  
Activity 1 crushes: 0 time unit  
Activity 2 crushes: 0 time unit  
Activity 3 crushes: 0 time unit  
Activity 4 crushes: 0 time unit  
Activity 5 crushes: 0 time unit  
Activity 6 crushes: 0 time unit  
Activity 7 crushes: 0 time unit  
Activity 8 crushes: 3 time unit  
Activity 9 crushes: 0 time unit  
Activity 10 crushes: 0 time unit  
Activity 11 crushes: 0 time unit  
Total cost of the activities: 560.000000  
Cost of directioning the project: 210.000000  
Cost of interest of the loan: 0.000000  
==> Total monetary cost: 770.000000
```

Figura 10: Screenshot de la salida (2)

7. Cambios a lo largo del proyecto

El modelo final que implementamos se basa en extender el modelo básico de CPM para que tenga en cuenta los aspectos económicos que se indica en el enunciado de la práctica, tal como se especifica en la sección 2 del informe. Sin embargo, este modelo no es el que habíamos pensado al inicio. La diferencia principal se basa a cómo calculamos los costes de ejecución de las actividades.

Al principio lo planteamos como:

$$\sum (t_i - c_i) * ce_i + c_i * cc_i$$

siendo:

t_i tiempo para finalizar una actividad i

c_i tiempo de crash que decidimos para una actividad i

tc_i tiempo mínimo en acabar una actividad si crash máximo

ce_i coste de ejecución por unidad de tiempo para la actividad i

cc_i coste de crash por unidad de tiempo de una actividad i

Es decir, el coste de hacer crash (tiempo * coste por unidad de tiempo de crash) más el tiempo restante después de crash (la diferencia) por el coste de ejecución normal por unidad de tiempo. Pero en varios ejemplos de papers vemos que la información de los proyectos que se nos proporciona es diferente a nuestro modelo, por lo que habrá que hacer cálculos previos (manualmente) para transformarlos a las entradas correspondientes. De esta forma, modificamos este modelo para que se adapte a los ejemplos que salen en las fuentes de información en que nos basamos. Ahora cc_i pasa de ser coste por unidad de tiempo en crash, a coste de la actividad acabándola en tc_i , que es el tiempo requerido para terminar la actividad si hacemos crash. Entonces, la fórmula pasa a ser:

$$\sum ce_i + c_i * (cc_i - c_i) / (t_i - tc_i)$$

De esta forma $(cc_i - c_i) / (t_i - tc_i)$ es el coste extra (debido a crash) por unidad de tiempo, que multiplicándolo por c_i tenemos el coste extra de la actividad por hacer crash, y a esto le sumamos el coste de ejecución normal ce_i . Pero hay que tener cuidado para el caso de $t_i = tc_i$ que resulta en una división por cero, pero sabemos que si se da el caso (tardamos igual haciendo crash), pues entonces simplemente el coste será ce_i , ya que lógicamente no vamos a pagar dinero extra para algo igual, esto en ampl se resuelve añadiendo una condicional. Dicho esto, ahora tenemos un modelo que se adapta bastante bien a la información de entrada de ejemplos que encontramos. No es necesario hacer cálculos extras para transformar la entrada, simplemente tenemos que copiarlo al .dat.

8. Competencia

Nuestra principal fuente de información se basa en el link que se da en el pdf, que nos lleva a los apuntes del profesor Beasley. Pues la información está bastante completa y fácil de entender, con diferentes ejemplos y extensiones. Para los case studies, nos basamos principalmente en el buscador de la biblioteca para encontrar documentos, y en algunos casos también Google para resolver dudas que tenemos sobre ampl.

8.1 Uso solvente de los recursos de la información

8.1.1 Case study de CPM-PERT

El siguiente case study lo encontramos con el buscador de revistas de la biblioteca de UPC, y se basa en comparar la eficacia de los métodos CPM y PERT en el contexto de construcción de edificios. El resultado es que aplicando cualquiera de los dos métodos se obtiene mejoras importantes comparado con los 173 días de experiencias anteriores. Con CPM calculamos que podemos completar el proyecto en 131 y con PERT 136 días, la diferencia se basa en que para CPM estamos usando tiempos definitivos para completar la actividad, mientras que en PERT éste no se conoce, por lo que tenemos una probabilidad de 75% (muy alta) en completarlo en 136 días.

Enlace:

<https://iopscience-iop-org.recursos.biblioteca.upc.edu/article/10.1088/1755-1315/140/1/012010>

8.1.2 Caso de aplicación del modelo implementado

Este documento es un case study de un modelo muy parecido al que hemos implementado en este trabajo, pues tienen que considerar no sólo recursos temporal, sino económicos también (posibilidad de hacer crashing). Pues se basan en datos obtenidos de gerentes de proyectos de Chowdhury Construction Company, Dhaka y Bangladesh. Y el resultado del análisis indica que podemos reducir el tiempo desde 140 días a 120 días aplicando este método.

Enlace: http://www.riejournal.com/article_51839.html

8.1.3 Otras Extensiones

Esta extensión la encontramos en un paper que buscamos también por la biblioteca de la UPC, y se basa en aplicar FCPM-FPERT (F de fuzzy) en el ámbito educativo. Pues, el diseño de las líneas de aprendizaje, reducción de los costes de enseñanza y incremento de la efectividad son objetivos importantes para las escuelas y universidades. Pero como sabemos, existe varios factores que influye en el aprendizaje del estudiante, por este motivo, extiende el modelo básico con fuzzy sets ya que son más eficientes para actividades con incertezas.

Enlace: <https://eric.ed.gov/?id=EJ1235798>

8.2 Bibliografía

Ejemplo PERT-CPM en la planificación de la construcción de una casa.

<https://www.ijstr.org/final-print/aug2015/Project-Planning-And-Scheduling-Using-Pert-And-Cpm-Techniques-With-Linear-Programming-Case-Study.pdf>

Ejemplo PERT-CPM en la planificación de la estabilización de 300 hectáreas de uva:

[http://idosi.org/aejaes/jaes5\(3\)/3.pdf](http://idosi.org/aejaes/jaes5(3)/3.pdf)

CPM-PERT en construcción.

<https://theconstructor.org/construction/uses-of-cpm-and-pert-in-construction-projects/9162/>

Transparencias del profesor J.E. Beasley sobre IO.

<http://people.brunel.ac.uk/~mastijb/jeb/or/contents.html>

Manual AMPL.

<https://ampl.com/BOOK/CHAPTERS/24-refman.pdf>

Case Studies (indicados anteriormente):

<https://iopscience-iop-org.recursos.biblioteca.upc.edu/article/10.1088/1755-1315/140/1/012010>

http://www.riejournal.com/article_51839.html

<https://eric.ed.gov/?id=EJ1235798>

<https://www-sciencedirect-com.recursos.biblioteca.upc.edu/science/article/pii/S026378631100127X?via%3Dihub>