Diagrammi PERT (rev 0.3)

Michele Schimd

6 febbraio 2022

1 Schedulazione delle attività di progetto

Compito del project manager è quello di predisporre un piano per indicare *chi fa cosa e quando*. La prima cosa da stimare è il **tempo** che ogni singola attività richiede. Altro aspetto fondamentale della schedulazione è l'individuazione delle **dipendenze** cioè cosa è necessario fare prima di poter procedere con una data attività. Nel classico esempio della costruzione di un edificio, le fondamenta devono essere fatte prima delle colonne portanti. D'altro canto prima di poter gettare le fondamenta è necessario scavare il terreno. Nella Figura 1 possiamo vedere una sequenza di attività necessarie alla costruzione di un edificio. Come è ovvio, queste attività possono avvenire solo nell'ordine indicato dalle frecce.

Figura 1: La sequenza di dipendenze nel progetto di costruzione di un edificio.

Sebbene tale esempio risulti semplice, è facile immaginare come, nella realtà, la pianificazione non sia così diretta e *lineare*. Si pensi ad esempio alle attività necessarie per la posa delle finestre, la dipintura, l'installazione degli impianti elettrici e idraulici, ... Senza una pianificazione formale ed effettuata mediante tecniche collaudate non sarebbe possibile l'architettura moderna per costruire il *Burj Khalida*, l'edificio più alto del mondo (828 metri) in 5 anni.

1.1 Cosa sono PERT e CPM

La tecnica **PERT** (*Program Evaluation and Review Technique*) è utilizza nel project management per gestire ed analizzare l'esecuzione dei task di un progetto (es. working package). Si tratta di una tecnica basata sull'**analisi dei grafi** che può essere facilmente realizzata con un programma software.

Il **CPM** (*Critical Path Method*) è un algoritmo su grafi che permette di individuare in un *grafo delle attività* il percorso più lungo detto anche **percorso critico**. Per questo motivo l'algoritmo CPM viene spesso utilizzato insieme alla tecnica PERT per la creazione di un *piano delle attività di progetto*

1.2 Stima dei tempi con PERT

Uno dei più importanti problemi della pianificazione di progetto è la **stima dei tempi di esecuzione di un'attività**. Tutta la schedulazione delle attività (e quindi della consegna dei *deliverable*) dipende in maniera fondamentale da una buona stima dei tempi. Per questo motivo la metodologia PERT prevede una formula specifica per la stima dei tempi che tiene conto di tre aspetti.

1. Tempo **ottimistico** t_o che indica *nel migliore di casi* quanto ci si impiega a completare l'attività.

2. Tempo **pessimistico** t_p che indica *nel peggiore dei casi* quanto ci si impiega a completare l'attività.

3. Tempo **molto probabile** t_m che indica *la migliore stima* di quanto ci si impiega a completare l'attività.

La formula indicata dalla metodologia PERT per il **tempo di esecuzione** atteso t_e è

$$t_e = \frac{t_o + 4t_m + t_p}{6}. \tag{1}$$

La stessa metodologia PERT indica come calcolare una stima della **variabilità** di tale tempo

$$\sigma_{t_e} = \frac{t_p - t_o}{6}. \tag{2}$$

Esempio Facciamo un esempio di stima del tempo per l'attività *Progettazione schema del database*. Supponiamo che il tempo peggiore sia $t_p = 16$ ore, quello ottimista $t_o = 4$ ore e la stima sia di $t_m = 8$ ore. Usando le formule (1) e (2)

$$t_e = \frac{16 + 4 \cdot 8 + 4}{6} = \frac{52}{6} = 8.67h$$
$$\sigma_{t_e} = \frac{16 - 4}{6} = \frac{12}{6} = 2h$$

Esercizio Riempire la Tabella 1 con i valori ti t_e e σ_{t_e} calcolati mediante le formule (1) e (1).

Attività	t_p	t_o	t_m	t_e	σ_{t_e}
A	1	3	4		
B	4	8	12		
C	3	3	3		
D	10	15	18		
E	2.5	3	3.5		

Tabella 1: Esercizio per stima della durata attività con PERT.

Rispondere alle seguenti domande.

- Qual'è l'attività che richiede meno tempo? Quale quella che richiede più tempo?
- Spiega il valore di σ_{t_e} per l'attività C.
- Se le attività fanno svolte una dopo l'altra (senza possibilità di sovrapposizione) quant'è la stima del tempo necessario?

2 Grafi (ripasso)

Vale la pena richiamare i concetti importanti che riguardano i **grafi** prima di procedere con lo studio dei grafi delle attività di progetto. Il modo più naturale di rappresentare un grafo è attraverso una *rete* di **nodi** collegati da **archi**. Un arco collega due nodi e può o meno essere orientato, nel caso di archi orientati, uno dei nodi *precede* l'altro, questo fatto è indicato con

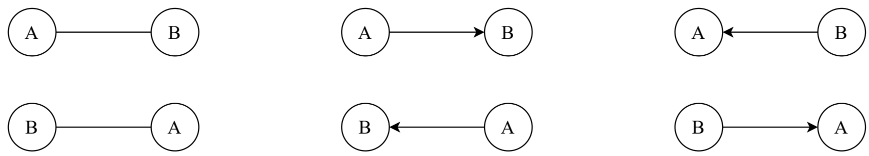

Figura 2: Tre casi di nodi direttamente connessi in un grafo. A sinistra abbiamo il caso di grafo non orientato che connette A e B. Nel caso centrale abbiamo un caso di grafo orientato in cui B succede ad A. Nel caso a destra abbiamo un caso di grafo orientato in cui A succede B. Per ognuno dei tre esempi vengono mostrati due modi **equivalenti** di mostrare la situazione.

una freccia che punta al nodo che *succede*. i vari casi di come un grafo può o meno essere orientato, sono mostrati in Figura 2.

È importante sottolineare che la **rappresentazione grafica è solo uno dei modi possibili di indicare un grafo**. Altri modo sono possibili ed, almeno due sono di interesse: rappresentazione mediante relazione e rappresentazione mediante *matrici*.

Rappresentazione di grafi mediante relazione Possiamo immaginare i grafi come un modo per rappresentare il "collegamento" tra due entità, che nella rappresentazione grafica sono i nodi del grafo. Ad esempio, nella Figura 2, la parte centrale indica un collegamento da A a B, mentre la parte sinistra un collegamento tra A e B (senza una direzione particolare).

Se ci intendiamo sul significato, allora possiamo esprimere il collegamento come una *coppia* (A, B). Ad esempio, nel caso sinistra della Figura 2, questo significa collegamento non orientato tra A e B, nel caso della parte centrale della figura questo significa collegamento da A a B. La coppia (B, A) sarebbe uguale ad (A, B) nel caso non orientato, ma diverso nel caso orientato perché indicherebbe collegamento da B ad A (si noti la direzione della freccia nella Figura 2).

Rappresentazione di grafi mediante matrici È possibile rappresentare un grafo anche utilizzando una tabella contenente informazioni numeriche, cioè una **matrice**. La matrice in questione avrà lo stesso numero di righe e colonne e questo numero è esattamente il numero di nodi del grafo. Possiamo quindi pensare ogni riga come associata ad un nodo così come ogni colonna può essere associata ad un nodo. Lo stesso nodo, inoltre, compare sia nelle righe che nelle colonne. La tabella corrispondente a due nodi A e B sarà

	A	B
A	$x_{0,0}$	$x_{0,1}$
B	$x_{1,0}$	$x_{1,1}$

Se da tale tabella estraiamo i numeri $x_{\dots}$ abbiamo la matrice seguente (si noti che la matrice si indica come una tabella racchiusa tra due parantesi tonde, spesso si usano anche le parantesi quadre)

$$\mathbf{X} = \begin{pmatrix} x_{0,0} & x_{1,0} \\ x_{1,0} & x_{1,1} \end{pmatrix}.$$

Ogni *cella* della matrice $\mathbf{X}$ contiene un valore numerico, nel caso più semplice questi valori sono 0 oppure 1, ma in generale possono essere valori *reali* (a volte si usano anche i valori $\pm\infty$).

Come nel caso della rappresentazione mediante relazione, anche nella rappresentazione *matriciale* è necessario mettersi d'accordo sul significato dei numeri. Nel caso della matrice in cui righe e colonne sono nodi, si parla di **matrice di adiacenza** (*adjacency matrix*) ed il numero indica la presenza o meno di un "collegamento". La seguente tabella indica il caso centrale della Figura 2.

	A	B
A	0	1
B	0	0

Nel caso di sinistra della della Figura 2, la tabella diventa

	A	B
A	0	1
B	1	0

la differenza sta nel fatto che A a B sono collegati senza direzione particolare (nella forma relazione questo vuol dire che (A, B) e (B, A) sono la stessa cosa) quindi vediamo che sia riga A colonna B, sia riga B colonna A sono impostati ad 1.

Self-loop Un caso particolare di collegamento o relazione tra nodi è quando un nodo è collegato a se stesso, in questo caso si parla di *self-loop* in quanto nella rappresentazione grafica questo è un circolo tra un nodo e se stesso e viene rappresentato con un arco (con o senza freccia dipende dal tipo di grafo) che esce ed entra nello stesso nodo.

Riassumendo

Esempio Per capire meglio il concetto di grafo, consideriamo l'esempio di grafo **orientato** la cui rappresentazione grafica è data nella Figura 3. Questo grafo contiene $n = 5$ nodi da A a B, vediamo come costruire la

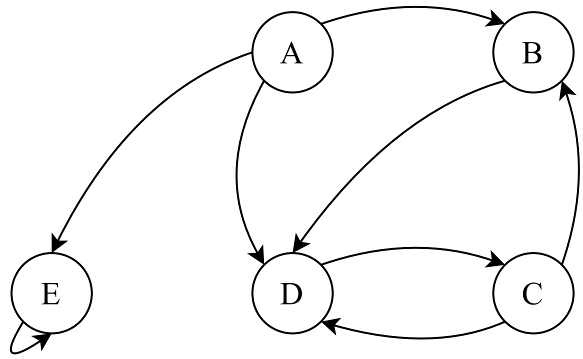

Figura 3: Un esempio di grafo orientato con 5 nodi.

rappresentazione matriciale.

1. *Creiamo la tabella* (e la matrice di adiacenza), avendo $n = 5$ nodi la nostra tabella avrà 5 righe e 5 colonne

	A	B	C	D	E
A					
B					
C					
D					
E					

2. Per *riempire la tabella* possiamo farlo una riga alla volta. Partendo dalla riga A ci chiediamo *quali sono i nodi con un collegamento proveniente da A?* (Ricordiamo che il grafo è orientato). Dalla figura vediamo che questi nodi sono B, D ed E e quindi nella riga A mettiamo 1 in corrispondenza delle colonne B, D ed E e 0 nelle altre celle della riga.

	A	B	C	D	E
A	0	1	0	1	1
B					
C					
D					
E					

3. Procedendo allo stesso modo *si riempie l'intera* tabella, notando che il nodo E a un (unico) arco che esce e che rientra in se stesso, questo significa semplicemente che nella riga E avremo l'unico 1 in corrispondenza della colonne E.

	A	B	C	D	E
A	0	1	0	1	1
B	0	0	0	1	0
C	0	1	0	1	0
D	0	0	1	0	0
E	0	0	0	0	1

4. Se viene richiesta la *matrice di adiacenza* basta prendere la tabella ed *eliminare le intestazioni* (usando la notazione con parentesi tonde, come sopra).

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Esercizio Data la seguente rappresentazione mediante matrici del grafo

	A	B	C	D	E
A	1	1	0	0	1
B	1	0	1	1	0
C	0	0	0	0	0
D	0	1	0	1	0
E	0	0	1	0	1

produrne la rappresentazione grafica.

2.1 Pesatura di un grafo

I grafi che abbiamo considerato finora, descrivono relazioni tra i nodi e più precisamente *relazioni binarie* cioè collegamento presente o assente. In molti casi è utile sapere qualcosa di più su tali relazioni ovvero il loro **peso**. Un classico esempio è rappresentato dal grafo che descrive le città come nodi ed i collegamenti (es. aerei o ferroviari) come gli archi. La Figura 4 mostra

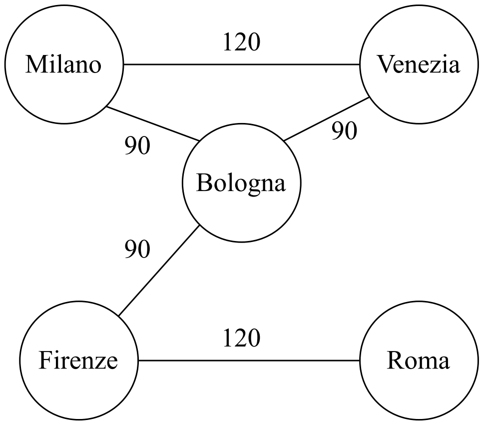

Figura 4: Un grafo non orientato e pesato per rappresentare i collegamenti ferroviari con la durata dei viaggi.

un grafo *non orientato* (non ci sono frecce negli archi) che potrebbe essere interpretato come i collegamento ferroviari ad alta velocità e la durata del viaggio nelle singole tratte. Ad esempio, da Venezia a Bologna esiste un collegamento la cui durata è di 90 minuti, mentre non esiste un collegamento tra Venezia e Roma.

Nella rappresentazione dei grafi pesati, il peso viene solitamente messo in evidenza in corrispondenza dell'arco (come si vede nella Figura 4). Nella rappresentazione matriciale, il peso viene messo come elemento della matrice (cioè al posto di 0 o 1). In questo caso lo 0 non viene messo perché un arco di peso 0 è diverso dall'assenza dello stesso arco. A volte si esprime l'assenza di un arco mettendo il valore $+\infty$ ad indicare una "distanza" infinita, ma questo può non avere senso in alcuni casi. Mettendo in ordine alfabetico le città (B, F, M, R, V) per il grafo di Figura 4 abbiamo la seguente matrice di adiacenza

$$\begin{pmatrix} - & 90 & 90 & - & 90 \\ 90 & - & - & 120 & - \\ 90 & - & - & - & 120 \\ - & 120 & - & - & - \\ 90 & - & 120 & - & - \end{pmatrix}$$

in cui abbiamo deciso di indicare con il simbolo - l'assenza di un collegamento.

Rappresentare i pesi nella notazione relazionale risulta scomodo, di solito si usa una *funzione* il cui argomento è l'arco. Ad empio la se la funzione $f(\cdot)$ rappresenta tale funzione di pesatura, nel caso in esempio avremmo

$$\begin{aligned} &f(\text{Milano}, \text{Bologna}) = 90 \\ &f(\text{Milano}, \text{Venezia}) = 120 \\ &f(\text{Bologna}, \text{Venezia}) = 90 \\ &\ldots \end{aligned}$$

come si può vedere questa rappresentazione risulta molto scomoda ed è di solito non utilizzata.

2.2 Percorsi in un grafo

Consideriamo nuovamente l'esempio di grafo non orientato della Figura 4. Benché non ci sia un collegamento tra Venezia e Roma, possiamo immaginare che, passando da Bologna e Firenze, possiamo attraversare un **percorso** (*path*) da Venezia a Roma.

Un **percorso** è una sequenza valida di archi; per valida si intende che ogni arco del percorso deve esistere ed uscire dal nodo in cui mi trovo. Ad esempio il percorso Venezia - Bologna - Roma, non è valido perché non c'è un arco tra Bologna e Roma, inoltre non posso "saltare" da Bologna direttamente a Roma, ma devo prima passare per il nodo Firenze.

Ovviamente, il tempo richiesto per attraversare un percorso dipende dai tempi dei singoli archi. Ad esempio, tra Venezia e Roma il tempo sarà

$$90 + 90 + 120 = 300.$$

In generale parliamo di **peso** di un percorso intendendo la somma dei pesi di ogni arco del percorso.

È naturale chiedersi, tra tutti i percorsi possibili, se ce ne sia uno di peso minore (o maggiore) di tutti gli altri. Ad esempio, per andare da Venezia a Roma è possibile seguire il percorso Venezia - Milano - Bologna - Firenze - Roma. Tuttavia tale percorso ha peso

$$120 + 90 + 90 + 120 = 420$$

ed è quindi meno veloce del percorso visto sopra.

Nell'esempio visto sopra, risulta ovvio che il percorso attraverso Milano sia più lungo rispetto a quello che non passa per Milano (visto che tutti e due passano per Bologna), nella pratica, tuttavia, non è sempre così ovvio trovare i percorsi migliori ed è quindi necessario utilizzare *algoritmi* appositi per l'individuazione di tali percorsi.

3 Grafo delle attività

In un progetto reale, le attività hanno delle *interdipendenze* più complesse di quelle illustrate in Figura 1. Infatti è possibile condurre un progetto utilizzando numerose persone che, contemporaneamente, svolgono delle attività di progetto portando a termine dei *work package*. Tuttavia è altrettanto normale che le varie attività non siano completamente *indipendenti* dalle altre, bensì che la maggior parte di esse possano essere iniziate solo dopo che una o più altre attività sono terminate. Per questo motivo, la gestione delle attività in un progetto richiede strumenti di analisi sofisticate, uno dei quali è il **grafo delle attività**.

3.1 Nodi e archi del grafo delle attività

Il **grafo delle attività** è un grafo *orientato* e *pesato* in cui ogni nodo rappresenta un'attività. Ogni arco indica una precedenza tra attività (se c'è l'arco $A \to B$, allora A deve terminare prima che B possa iniziare). Inoltre, la pesatura sull'arco indica la durata dell'attività da cui l'arco esce.

Nella Figura 5, se vede un esempio di grafo delle attività in cui 8 nodi (corrispondenti alle 8 attività $A1, \ldots, A2$) sono collegati da archi che ne indicano precedenza e durata.

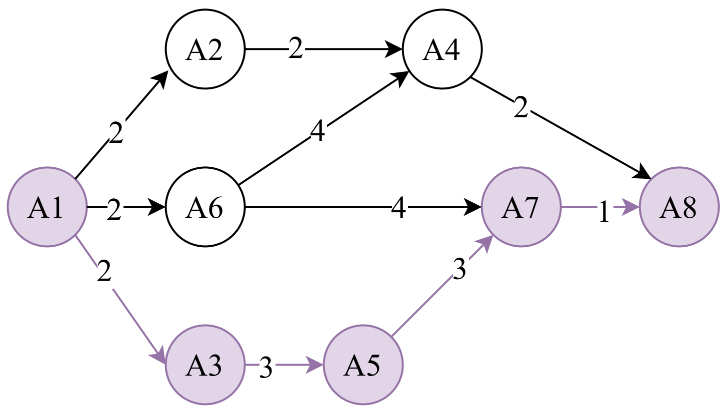

Figura 5: Esempio di un grafo delle attività con 8 nodi. Il *percorso critico* (più lungo) è evidenziato in viola.

3.2 Critical Path Method (CPM)

Uno dei più importanti utilizzi del grado delle attività è l'identificazione del **percorso critico** definito come il percorso con la durata (somma dei pesi degli archi) maggiore. Infatti il percorso critico rappresenta il *tempo minimo* di durata del progetto ed è quindi un valore molto importante da stimare in modo quanto più preciso possibile

Nella Figura 5, il percorso critico è stato evidenziato colorando i nodi e gli archi che vi fanno parte. Si può notare come tale percorso abbia lunghezza 9 e che nessun altro percorso abbia lunghezza maggiore (nell'esempio questo è anche l'unico percorso di lunghezza 9). Per completezza, la seguente tabella riporta tutti i percorsi possibili e le corrispondenti durate.

Percorso	**Durata**
$A1 - A2 - A4 - A8$	6
$A1 - A3 - A5 - A7 - A8$	9
$A1 - A6 - A4 - A8$	8
$A1 - A6 - A7 - A8$	7

3.3 Stima dei tempi di progetto PERT e CMP

Il meccanismo di stima dei tempi con PERT vista nella Sezione 1.1 ed la tecnica del CPM vista nella Sezione 3.2 sono solitamente usate insieme per avere una visione più dettagliata e realistica dei tempi di esecuzione di un progetto.

La prima cosa che ogni *project manager* deve conoscere e tenere sotto controllo è il tempo totale di esecuzione del progetto. Senza conoscere tale tempo, non è possibile fornire stime realistiche sui tempi di consegna del prodotto finito. La tecnica PERT vista nella Sezione 1.1 basata sui valori *ottimistici*, *medi* e *pessimistici* di stima dei tempi, vale per il calcolo del tempo della singola attività. Ovviamente, è interessante usare PERT per la stima del tempo di tutto il percorso critico.

Supponiamo che la stima PERT degli archi ottenuta con la formula (1) a pagina 2 siano $t_1, t_2, \ldots, t_m$ (quindi ci sono m archi sul percorso critico). Per ottenere una stima della durata delle attività sul percorso critico (che fornisce una stima del tempo di esecuzione del progetto), basta sommare le stime su tutti gli archi del percorso

$$t_{PC} = t_1 + t_2 + \ldots + t_m = \sum_{i=1}^{m} t_i.$$

Per la stima della *varianza*, si parte dalle varianze $\sigma_1, \ldots, \sigma_m$ trovate con la formula (2) a pagina 2, ma anziché sommarli, si *prende la radice quadrata della somma dei quadrati*

$$\sigma_{PC} = \sqrt{\sigma_1^2 + \sigma_2^2 + \ldots + \sigma_m^2} = \sqrt{\sum_{i=1}^{m} \sigma_i^2}.$$

3.4 Utilizzo del percorso critico e calcolo dello *slack*

Oltre ad indicare il minimo tempo di completamento di un progetto, il percorso critico può essere utilizzato per meglio schedulare e controllare le attività che si trovano al di fuori del percorso critico. Infatti è possibile che alcune di queste attività abbiano dei tempi di inizio e fine *flessibili* in quanto non coinvolte nel percorso critico. Avendo questa informazione è possibile, ad esempio, distribuire le risorse in modo che **le attività critiche non subiscano ritardi visto che questo vorrebbe dire ritardare tutto il progetto**.

Per quantificare numericamente la quantità di flessibilità del progetto, dobbiamo introdurre i seguenti concetti.

Early Start (ES) il primo momento possibile per l'inizio di un'attività;

Late Start (LS) l'ultimo momento possibile per l'inizio di un'attività;

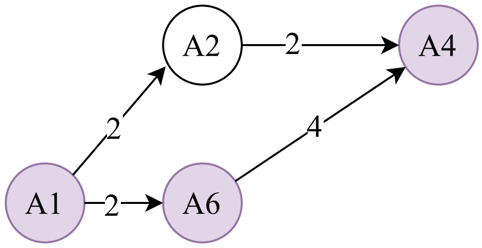

Figura 6: Semplice esempio di grafo delle attività con percorso critico.

Early Finish (EF) il primo momento possibile di completamento di un'attività;

Late Finish (LF) l'ultimo momento possibile per il completamento di un'attività.

Il concetto di primo e ultimo momento possibile va riferito al tempo di esecuzione del progetto. Semplificando, questi sono quei tempi di inizio e fine che sono possibili senza che il progetto subisca ritardi.

Cerchiamo di capire questi concetti con il semplice esempio di grafo delle attività mostrato in Figura 6. Iniziamo con l'osservare che il percorso critico ($A1-A6-A4$) ha durata 6. L'unico altro percorso possibile ($A1-A2-A4$) ha durata 4. Mentre ritardare la fine e/o l'inizio di una qualsiasi delle attività sul percorso critico farà aumentare la sua durata, quindi ritarderà il progetto, per quanto riguarda l'attività $A2$ questo non è necessariamente vero. Vediamo infatti che, anche se $A2$ inizia con un ritardo di 1 (l'unità di misura qui può essere qualsiasi, es. giorni o settimane, purché si usi sempre la stessa) questo non ritarda il progetto. Infatti in questo caso il percorso $A1-A2-A4$ avrà durata 5, ma il percorso critico rimarrà $A1-A6-A4$ con durata 6. Diciamo quindi che l'attività $A2$ può iniziare, al più tardi al tempo 4 anche se al più presto poteva iniziare al tempo 2.

Riferimenti bibliografici

[1] Wikipedia contributors. Critical path method — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Critical_path_method&oldid=1055667689`, 2021. [Online; accessed 6-January-2022].

[2] Wikipedia contributors. Program evaluation and review technique — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Program_evaluation_and_review_technique&oldid=1053710479, 2021. [Online; accessed 6-January-2022].