Resource-Constrained Project Scheduling: Heuristics

Alessandro Hill

rev. 1.0(AH) – 2024

Topological Sorting Algorithm for the Project Scheduling Problem (PSP, no resources)

- Jobs J: L'insieme delle attività da pianificare.
- Precedence Network N: Un grafo diretto aciclico (DAG) che rappresenta i vincoli di precedenza tra le attività, dove i nodi sono le attività e gli archi rappresentano i vincoli di precedenza.

INPUT: Jobs J, Precedence Network N

Algoritmo di ordinamento topologico per il problema di programmazione dei progetti (Project Scheduling Problem, PSP) senza considerare le risorse.

1. Initialize:
   - Empty schedule S → Creare una schedule vuota S, che verrà popolata con le attività.
   - Auxiliary precedence network N':=N → Creare una copia ausiliaria della rete di precedenza N':=N, che sarà aggiornata durante l'algoritmo.

L'algoritmo segue una strategia iterativa per pianificare le attività nel rispetto dei vincoli di precedenza.

2. Repeat the following until all jobs are scheduled:
   a) Pick a job j that has not been scheduled yet and has NO predecessors in the CURRENT precedence network N'.
      Scegliere un'attività j che:
      - Non è stata ancora pianificata (non in S).
      - Non ha predecessori rimanenti nella rete di precedenza N' (Significa che tutti i suoi predecessori sono già stati completati).
   b) Remove j from the CURRENT precedence network N'. → Rimuovere il nodo j e i suoi archi dalla rete di precedenza N'.
   c) Schedule j in S to begin at the earliest possible start time. This is the latest end time in S of all predecessors of j in N. → Aggiungere j alla schedule S alla prima posizione disponibile, nel rispetto del suo tempo di inizio più presto possibile. Il tempo di inizio di j è determinato dal tempo di termine dei suoi predecessori in N.

This is an exact algorithm for the Project Scheduling Problem without resources.

Alternative description:
- Find a topological ordering Σ of the nodes in N.
- Schedule jobs in the order given by Σ at the earliest possible times.

- Trovare un ordinamento topologico sigma dei nodi nella rete di precedenza N (Un ordinamento topologico è una sequenza lineare delle attività che rispetta i vincoli di precedenza).
- Pianificare le attività nell'ordine dato da sigma, iniziando ogni attività al tempo più presto possibile.

La base della stategia TopoSort sono le Precedenze

Topological Sorting Heuristic for the RCPSP

TopoSort

Jobs J: L'insieme delle attività da pianificare.
Precedence Network N: Un grafo diretto che rappresenta i vincoli di precedenza tra le attività.
Resources R: L'insieme delle risorse disponibili, con vincoli di capacità.

INPUT: Jobs J, Precedence Network N, Resources R

1. Initialize:
   - Empty schedule S → Creare una schedule vuota S, che sarà popolata con le attività pianificate.
   - Auxiliary precedence network N' → Creare una copia ausiliaria della rete di precedenza N':=N, che sarà aggiornata iterativamente.

Ripetere i seguenti passi finché tutte le attività non sono pianificate:

2. Repeat the following until all jobs are scheduled:
   a) Pick a job j that has not been scheduled yet and has NO predecessors in the CURRENT precedence network N'.
   b) Remove j from the CURRENT precedence network N'.
   c) Schedule j to begin at the earliest possible start time. This is at least the latest end time of all predecessors of j in N. In general, it is later due to resources limitations in the CURRENT schedule.

Selezionare un'attività j che:
- Non è stata ancora pianificata.
- Non ha predecessori nella rete di precedenza corrente N' (Tutti i predecessori sono già stati completati).

Rimuovere il nodo j e i suoi archi associati dalla reta N'.

Pianificare j al tempo di inizio più presto possibile, tenendo conto di:
- Vincoli di precedenza: j può iniziare solo dopo il completamento di tutti i suoi predecessori.
- Vincoli di risorse: La pianificazione deve rispettare la disponibilità delle risorse in ogni momento (cioè in quel periodo la risorsa disponibile non può essere negativa).

This is a heuristic method for the RCPSP (PSP with resources).

Questa è una strategia euristica avanzata che sfrutta la rilassazione lineare (LP) del problema di programmazione lineare intera (IP) per determinare una pianificazione iniziale.

2 α - Point Heuristic for the RCPSP

Jobs J: L'insieme delle attività da pianificare.
Precedences A: L'insieme dei vincoli di precedenza che indicano l'ordine delle attività.
Resources R: L'insieme delle risorse disponibili con vincoli di capacità.

Scheduling Threshold alfa: Una soglia definita dall'utente per guidare l'euristica. Indica il livello di "confidenza" richiesto per pianificare un'attività in un determinato momento.

INPUT: Jobs J, Precedences A, Resources R, Scheduling Threshold α

1. Solve the linear relaxation (LP) for an RCPSP formulation (IP).
   This gives you a fractional value $x_{j,t}$ for each potential start time* t for each job j.

Risolvi la versione rilassata del problema RCPSP, che consente soluzioni frazionarie (non intere). Questo fornisce un valore frazionario xj,t che rappresenta la probabilità che il lavoro j inizi al tempo t.

2. Initialize empty job ordering Σ.

Crea un ordine vuoto sigma, che conterrà l'ordine finale delle attività.

Per ogni possibile tempo di inizio t, da 0 fino a MAX(T)

3. For potential job start time* t form 0 to MAX(T) do:
   - For each job that is NOT in Σ, calculate the start potential $a_{j,t}$ at time t: $a_{j,t} = \sum_{t'=0}^{t} x_{j,t}$
   - Let candidate set C_t contain all jobs that are not in Σ and for which $a_{j,t} \geq \alpha$.
   - Append jobs in C_t that are "precedence-feasible" to Σ**

Calcola il valore cumulativo aj,t: Questo valore rappresenta la probabilità cumulativa che il Job j inizi entro t.

Se ci sono più attività in Ct che possono essere aggiunte contemporaneamente e rispettano i vincoli di precedenza, aggiungere prima le attività con il valore aj,t più alto.

Ct contiene tutte le attività j che non sono ancora in sigma e hanno un valore aj,t >= ad alfa

4. Schedule jobs according to the ordering Σ.

Pianifica le attività secondo l'ordine sigma

*Can also be done using job end times.

**Try appending in decreasing order of $a_{j,t}$. All predecessors of the job must already be in Σ.

The α – point (in time) for a job j is the time t when $a_{j,t} \geq \alpha$.

Il punto alfa(t) è il primo momento in cui il valore cumulativo aj,t supera o è uguale ad alfa.

Strategy: Run this heuristic for various values of α and return the best schedule.
Example: $\alpha \in \{0.1, 0.2, \ldots, 1.0\}$

For very large RCPSPs, solving the LP can be time consuming!

Es. Capitol Construction (2a. Re-revisited)

La Capitol Construction Company deve completare la ristrutturazione del suo attuale ufficio il più rapidamente possibile…

Compito	Simbolo	Precedenca	Durata	Persone	Costi (in 1000)
Preparare opzioni di finanziamento	A	-	2	**3**	**3**
Preparare schizzi preliminari	B	-	3	**2**	**1**
Delineare le specifiche	C	-	1	**1**	**3**
Preparare disegni	D	A	4	**3**	**4**
Scrivere le specifiche	E	C, D	5	**3**	**1**
Eseguire le stampe	F	B	1	**1**	**1**

Resource availability: $q_{Persone} = 4$, $q_{Costi} = 5$

a) Trova una soluzione con il algoritmo TopoSort
b) Trova una soluzione con il algoritmo α - Point

Exercise 1)

In Table 1.1, a RCPSP instance is given with $n = 10$ real activities and $|\mathcal{R}| = 2$ resources with availabilities $B_1 = 7$ and $B_2 = 4$.

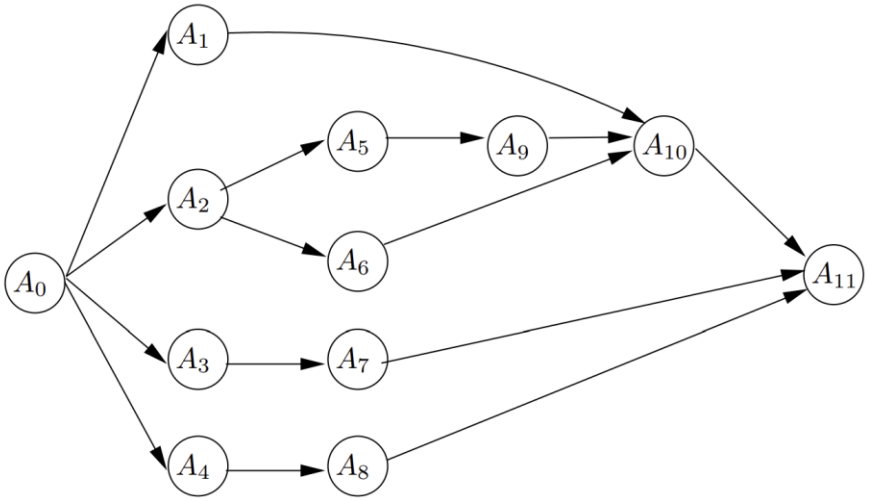
A_i	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}
p_i	0	6	1	1	2	3	5	6	3	2	4	0
b_{i1}	0	2	1	3	2	1	2	3	1	1	1	0
b_{i2}	0	1	0	1	0	1	1	0	2	2	1	0

1. Find a schedule using the TopoSort Heuristic

2. Find a schedule using the α - Point Heuristic

3. Find an optimal schedule using the IP or CP

3. Visualize all schedules using Gantt charts
Verify your results visually.
Compare results.



Exercise 2)

1. **Implement an RCPSP Heuristic**
   Option 1: TopoSort Heuristic (TopoSort)Build the Excel IP model.
   Option 2: α –Point Heuristic.
   Option 3: Both heuristics.
   Use the language/system of your choice.

2. **Run your algorithm on larger test instances**
   Data on Virtuale.
   Careful: Different data formats!
   Feel free to use other large RCPSP instances.

3. **Can you visualize your results?**

4. **How good are your results compared to an exact method?**
   Use an exact IP Solver or MiniZinc?