§16 Evaluating the distribution of the objective value of a policy

Policies and detailed project analysis

- First step:
  - determine best/good policy Π w.r.t. expected cost $E(\kappa^{\Pi})$
- Second step:
  - use Π for detailed analysis
    - » cost distribution
    - » time-cost tradeoff

Early Start policies $\Rightarrow$ methods from this part applicable

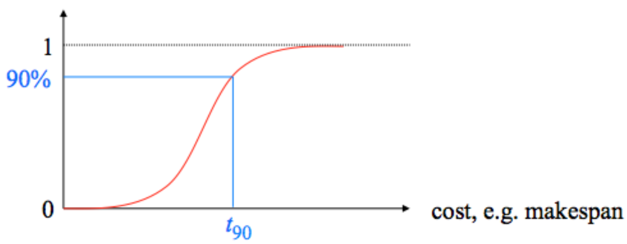
ES-policies just do early start scheduling w.r.t. precedence constraints

$\Rightarrow$ many algorithms from combinatorial optimization available

Consider now makespan $\kappa = C_{max}$ as objective

Detailed analysis of cost distribution

Ideal: distribution function F of κ^{Π}



Modest: percentiles

$$t_{90} = \inf\{t \mid \underbrace{\Pr\{\kappa^{\Pi} \le t\}}_{F(t)} \ge 0.90\}$$

Obtaining stochastic information is hard

Hagstrom '88

MEAN

Given: Stochastic project network with discrete independent processing times

Wanted: Expected makespan

DF

Given: Stochastic project network with discrete independent processing times, time t

Wanted: $\Pr\{\text{ makespan} \le t\ \}$

Problems to solve

Hagstrom '88

- MEAN and DF are polynomially equivalent for 2-point distributions
- DF and the 2-point versions of DF and MEAN are #P-complete
- Unless P = NP, MEAN and DF cannot be solved in time polynomial in the number of values that the makespan attains

16.1 Theorem

Reduction from computing network reliability

#P = complexity class for counting problems (called "sharp NP" or "number NP")

A problem is in #P if one can compute in non-deterministic polynomial time the number of "yes"-answers for every instance of that problem, i.e.

$\mathcal{P} \in \#P :\Longleftrightarrow \exists$ polynomial $p(n)$, $\exists$ non-deterministic algorithm A such that $\forall\ I \in \mathcal{P}$ $A(I)$ gives the number of "yes"-answers to instance I in time $p(<I>)$

Examples

(1) SAT $\in$ NP : does a CNF formula have a fulfilling assignment?

#SAT $\in$ #P : how many fulfilling assignments has a CNF formula?

(2) HAMILTON CYCLE $\in$ NP : Does graph G have a Hamiltonian cycle?

HAMILTON CYCLE $\in$ #P : How many ...

(3) LINEAR EXTENSION $\in$ P : does partial order P have a linear extension?

#LINEAR EXTENSION $\in$ #P : how many ...

shows that polynomially solvable problems also lead to #P problems

Similar to proving NP-completeness, one defines #P-complete and shows:

$\mathcal{P} \in$ #P is #P-complete

$\iff \exists$ polynomial Turing reduction from $\mathcal{P}'$ to $\mathcal{P}$, where $\mathcal{P}'$ is #P-complete, such that counting on $\mathcal{P}$ solves the counting problem on $\mathcal{P}'$

Examples (1)-(3) above are #P-complete

$\Rightarrow$ #P-complete problems can arise from polynomially solvable problems!

Proof of Theorem 16.1 needs a different representation of the project network G that will also be usefull for other algorithms

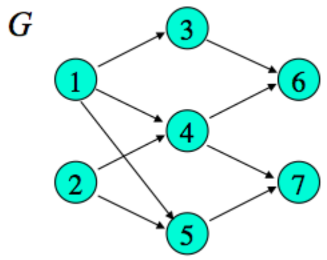
Transform $G = (V, E)$ into an s,t-dag $D = (N, A)$

D has a unique source s and a unique sink t

Every job $j \in V$ is an arc of D such that precedences are preserved, i.e. $(i,j) \in E \Rightarrow \exists$ path from the end of i to the start of j in D

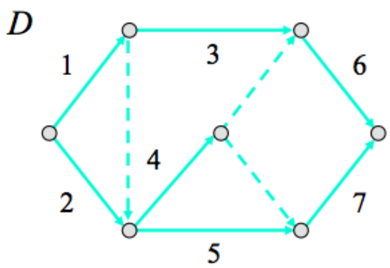
One may use additional arcs (dummy arcs) to properly represent precedence constraints

Arc diagrams



Node diagram

jobs are nodes of digraph G



Arc diagram

jobs are arcs of digraph D

dummy arcs - - - → may be necessary

standard graph algorithms apply

makespan = longest path

D is called an arc diagram of G

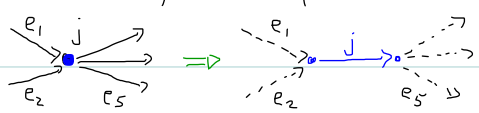
Note:

(1) arc diagrams are not unique

(2) a standard (polynomial) construction is as follows:

(i) take the edges of G as dummy arcs

(ii) blow up every vertex of G to an arc



(iii) identify sources and sinks,

contract "superfluous" dummy arcs

(3) Construction with the minimum number of dummy arcs is NP-hard

Proof of Theorem 16.1 (only for the 2-point version of DF).

Use reduction from RELIABILITY (is #P-complete)

Given: s,t-dag D with failure probabilities q_j on arcs j

Wanted: $\underbrace{\text{the probability}}_{\text{the reliability of } D}$ that there is an s,t-path without arc failures

Let I be an instance of RELIABILITY

Construct instance I' of 2-point DF s.t.

$$\text{reliability of } D = 1 - \text{Prob}\{C_{max} \le L-1\} \text{ for some appropriate } L$$
$$= 1 - F(L-1)$$

$\Rightarrow$ can compute reliability(D) from $F(L-1)$

Construction of I':

Take s,t-dag D as arc diagram of G

Every job (arc) j has processing time $X_j = \begin{cases} 0 & \text{with prob } q_j \\ p_j > 0 & \text{with prob } 1-q_j \end{cases}$

with p_j integer such that all s,t paths have equal length L

(this can be done in polynomial time)

Then: $\text{Prob}(\text{some path is reliable}) = 1 - \text{Prob}(\text{all paths fail})$

$$= 1 - \text{Prob}(\cap_i \{\text{path } P_i \text{ fails}\})$$
$$= 1 - \text{Prob}(\cap_i \{\text{some arc fails on path } P_i\})$$
$$= 1 - \text{Prob}(\cap_i \{\text{path } P_i \text{ has length} < L\})$$
$$= 1 - \text{Prob}(C_{max} \le L-1)$$

This shows #P-hardness. See Hagstrom 88 for DF $\in$ #P $\square$

Exercises:

16.1: Characterization of partial orders that have an arc diagram without dummy arcs

A partial order is N-free if its Hasse diagram does not contain N as subgraph (so $\forall i,j$ $ImPred(i) = ImPred(j)$ or $ImPred(i) \cap ImPred(j) = \emptyset$)

A partial order has the CAC-property if every maximal chain and every maximal antichain have a non-empty intersection (also called the chain-antichain property

Show that the following are equivalent for a partial order $G = (V, E)$:

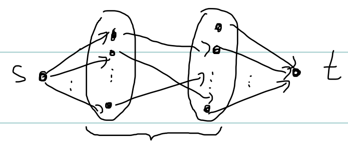
(1) G has an arc-diagram without dummy-arcs.

(2) G is N-free

(3) G has the CAC-property

16.2: Strengthen the proof of Theorem 16.1 to processing times $X_j \in \{0,1\}$.

Hint: Use the fact that RELIABILITY is already #P-complete for s,t-dags of the form



bipartite graph

16.3 Show that the 2-point versions of MEAN and DF with $X_j \in \{0,1\}$ are polynomially equivalent