Systems of Uniform Recurrence Equations

(Hand-Made slides)

Alain Darte
LIP, ENS-Lyon, 46, Allée d'Italie, 69007 Lyon, France.
E-mail: `Firstname.Lastname@ens-lyon.fr`

SYSTEMS OF UNIFORM RECURRENCE EQUATIONS

from

- "The organization of computations for uniform recurrence equations"
  KARP, MILLER & WINOGRAD, 1967
- Works by
  RAO, 1985
  Roychowdhury, 1988
  DARTE, KHACHIYAN, ROBERT, 1991
  DARTE, VIVIEN, 1995

OUTLINE

- Definition, example
- Link with different models or problems
- Computability, principles
- Computability & Linear programming
- Scheduling
    - One equation.
    - A system of equations.

Definition

Structure of regular computations:

$$a_i(p) = f\left(a_{i_1}(p - d_1^i), \ldots, a_{i_k}(p - d_k^i)\right)$$

$\forall p \in \mathcal{P} \subset \mathbb{Z}^n$.

Example: $1 \leq i,j \leq N$

$$\begin{cases} a(i,j) = a(i-1,j) + b(i,j-1) \\ b(i,j) = b(i+1,j) + a(i,j) \end{cases}$$

Semantics:

- Single assignment. Systems of equations.
- Compute right-hand side before left-hand side.
- If $p - d \notin \mathcal{P}$, data is given, otherwise $a_j(p)$ depends on $a_i(p-d)$.

$\Rightarrow$ Explicit dependences, but implicit execution order driven by the semantics.

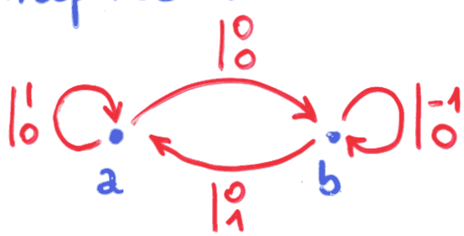
Reduced Dependence Graph

- One vertex per equation
- One edge per dependence

So:

$$\begin{cases} a(i,j) = a(i-1,j) + b(i,j-1) \\ b(i,j) = b(i+1,j) + a(i,j) \end{cases}$$

Corresponds to



vertices, edges

$G = (V, E, \omega)$

dependences

$e = u \xrightarrow{\omega(e)} v$ means

$\forall p \in \mathcal{P}$, $p - \omega(e) \in \mathcal{P}$: $v(p)$ must be computed after $u(p - \omega(e))$.

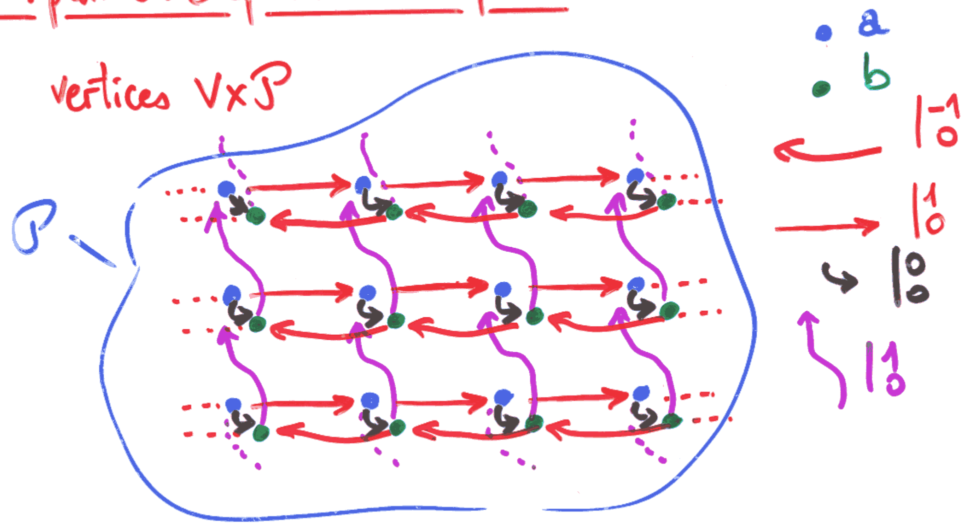
In terms of scheduling:

$$\sigma : V \times \mathcal{P} \longrightarrow \mathbb{N}$$

$$\sigma(v, p) \geq \sigma(u, p - \omega(e)) + 1$$

Expanded Dependence Graph

vertices $V \times \mathcal{P}$



Questions

- Given $v \in V$ and $p \in \mathcal{P}$, can we "compute" $v(p)$ in a finite number of steps?

  $\Rightarrow$ Computable equations

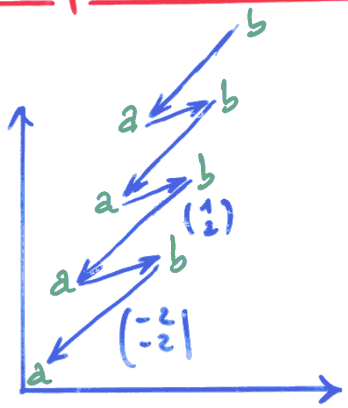
- If $v(p)$ is computable, can we evaluate the minimal number of steps required to compute it?

  $\Rightarrow$ Free schedule –

  As soon as possible schedule –

  (ASAP)

Computable RDGs



in $\mathbb{N}^n$, computable
$\Leftrightarrow$ No cycle C such that $\omega(C) \leq 0$.

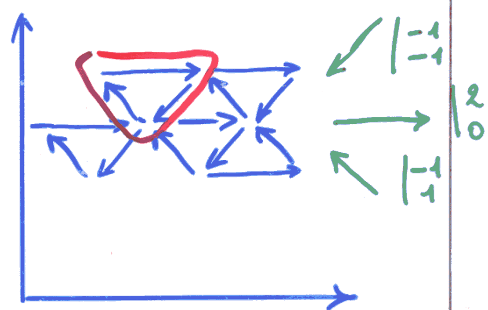
Proof of $\Rightarrow$:

if $\omega(C) \leq 0$

$\sigma(u) \geq \sigma(u-\omega(e)) + 1$
$\geq \dots$ and

$u \in \mathbb{N}^n \Rightarrow u - \omega(C) \in \mathbb{N}^2$

$\Leftarrow$: More difficult



in a bounded $\mathcal{P}$, computable
$\Leftrightarrow$ no cycle C such that $\omega(C) = 0$.

$\Rightarrow$ only if $\mathcal{P}$ is large enough

Proof of $\Rightarrow$:

$\sigma(u) \geq \sigma(v - \omega(e)) + 1$
$\sigma(v - \omega(e)) \geq \dots + 1$

$+ \quad \overline{\sigma(u) \geq \sigma(u) + \ell(C)}$

length of C

G is computable $\overset{\text{def}}{\Leftrightarrow}$ $\forall \mathcal{P}$ bounded domain, the SURE is computable.

$\Leftrightarrow$ G has no cycle C s.t. $\omega(C) = 0$

Related Models (1/3)

Nested Loops in imperative languages.

```
DO i=1,n
  DO j=1,n
    a(i,j) = a(i-1,j) + b(i,j-1)
    b(i,j) = b(i+1,j) + a(i,j)
  ENDDO
ENDDO
```

⚠ Explicit order, the sequential order $<_{seq}$.

Implicit dependences.

All distance vectors are lexicopositive $\succ_{lex} 0$.

$\Rightarrow$ No computability problems

BUT

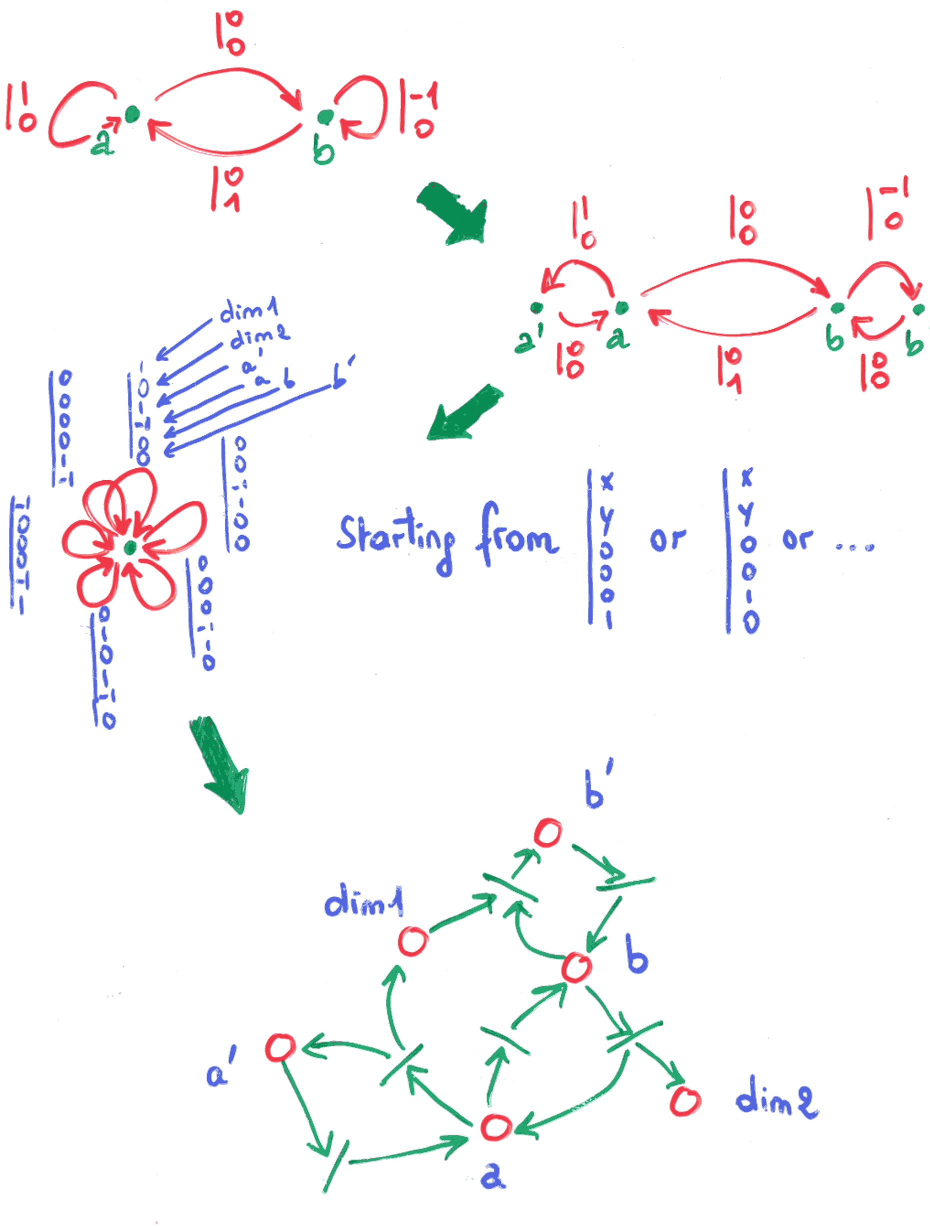
SIMILAR TECHNIQUES

ITERATION DOMAIN, RDG, EDG, ...

NOT ALWAYS UNIFORM DEPENDENCES

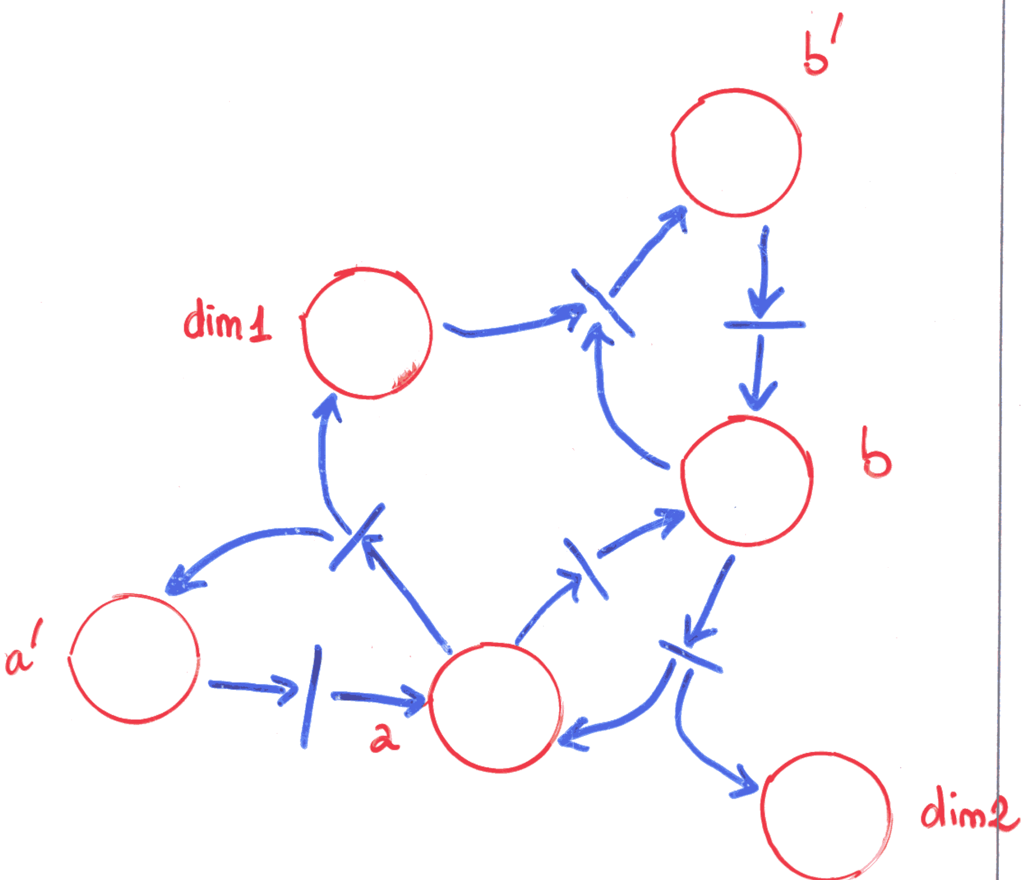
Related Models (2/3)

Petri Nets

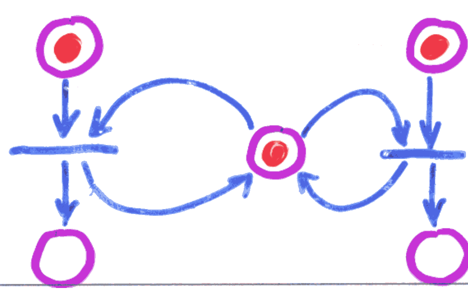


Accessibility Problem.

Petri Nets



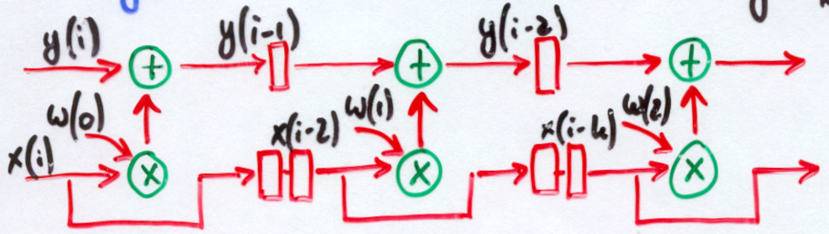
Example of resource constrained problem



Related models (3/3)

- Registers in VLSI circuits

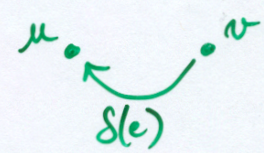
$$y_i = \sum_{k=0}^{K} w_k x_{i-k}$$



- Scheduling tasks with deadlines

"after" $t(v) \geq t(u) + \delta(e)$

"before" $t(v) \leq t(u) + \delta(e)$



⇒ Cyclic graphs labeled by integers.
Particular case in dim 1.

⇒ "easier": Bellman-Ford
Leiserson-Saxe

Linear scheduling & 1 uniform recurrence equation

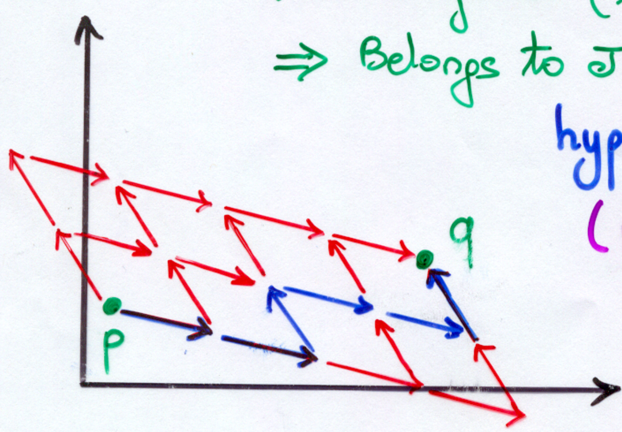
- A URE is completely defined by:
  - a set of dependence vectors $D = \{d_i, 1 \le i \le m\}$
  - an iteration domain $\mathcal{J} = \{x \mid Ax \le b\}$ (polyhedron)

- Dependence path:

$$p \rightsquigarrow q \quad : \quad q = p + \sum_{i=1}^{m} \lambda_i d_i \qquad \lambda_i \in \mathbb{N}$$

uses λ_i times the dependence vector d_i –

Conversely: Study the path closest to the the diagonal (straight line).
$\Rightarrow$ Belongs to $\mathcal{J}$ under suitable hypotheses.
(p & q not too close to the boundary)



Computability for one equation

- Not computable iff there is a cycle in the EDG, iff $\exists p$ s.t. $p = p + D\lambda$ iff $\exists \lambda \in \mathbb{N}, \lambda \neq 0$ & $D\lambda = 0$.

  $\Rightarrow$ always true

  $\Leftarrow$ if J is "large enough".

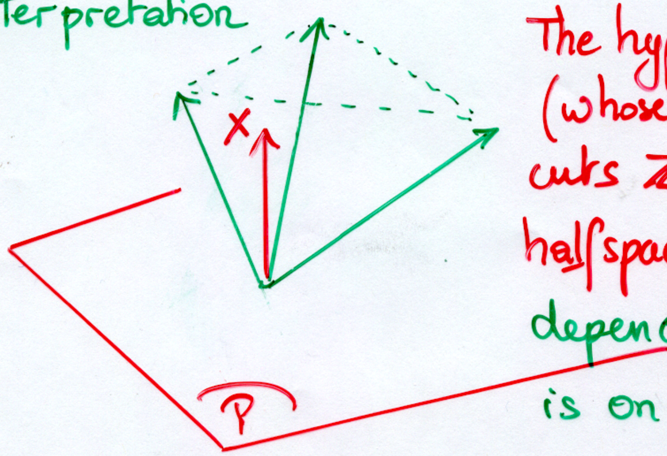
- Computable $\Leftrightarrow \{\lambda \in \mathbb{Z}^n \mid \lambda \geqslant 0, \lambda \neq 0, D\lambda = 0\} = \emptyset$

$\Leftrightarrow \{\lambda \in \mathbb{Q}^n \mid \lambda \geqslant 0, \lambda \neq 0, D\lambda = 0\} = \emptyset$

$\Leftrightarrow \{X \in \mathbb{Q}^n \mid XD \geqslant 1\} \neq \emptyset$

↳ Farkas' Lemma.

- Interpretation



The hyperplane P (whose normal is X) cuts $\mathbb{Z}^n$ in two halfspaces: the dependence cone is on one side.

Schedule for one equation:

- Linear schedule:

$\sigma(p) = X.p$ (or $\lfloor X.p \rfloor$ if $X \notin \mathbb{Z}^n$)

- Constraint:

$Xd \geq 1 \iff Xp \geq X(p-d)+1 \quad \forall p \in J$

- Latency:

$\max \{ \lfloor X.p \rfloor - \lfloor X.q \rfloor \mid p \in J, q \in J \}$

$\leq 2 + \max \{ X(p-q) \mid Ap \leq b, Aq \leq b, p \in \mathbb{Q}^n, q \in \mathbb{Q}^n \}$

- "Optimal" linear schedule:

$$\min_{XD \geq 1} \; \max_{\substack{Ap \leq b \\ Aq \leq b}} X(p-q)$$

$\Rightarrow$ Minmax problem.

Linear schedule with minimal latency

$$\mathrm{Max}\{X(p-q) \mid Ap \le b,\ Aq \le b\}$$

$$= \mathrm{Min}\ \{(X_1+X_2)b \mid X_1 \ge 0,\ X_2 \ge 0,\ X_1A = X,\ X_2A = -X\}$$

duality thm.

Thus:

$$\min_{XD \ge 1}\ \max_{\substack{Ap \le b\\ Aq \le b}}\ X(p-q) =$$

$$\mathrm{Min}\{(X_1+X_2)b \mid X_1 \ge 0,\ X_2 \ge 0,\ X_1A = X,\ X_2A = -X,\ XD \ge 1\}$$

$\Rightarrow$ one simple linear program.

Rmq: $N\, \lambda_{opt}(A, b) = \lambda_{opt}(A, Nb)$

$\Rightarrow$ linearity in the domain size.

- If b is parameterized, use parametric linear programming.

Interpretation of the dual

$$\text{Min}\left\{(X_1+X_2)b \mid X_1 \geq 0, X_2 \geq 0, X_1A = X, X_2A = -X, XD \geq 1\right\}$$

$$=$$

$$\text{Max}\left\{\sum_{i=1}^{m} \lambda_i \mid \lambda \geq 0, Aq \leq b, Ap \leq b, q = p + D\lambda\right\}$$

$\Rightarrow$ "path" of length $\Sigma\lambda_i$ = latency !!

But : * rounding effects $(\lambda \in \mathbb{Q})$

* boundary effects

Proof principle:

$$L_{min}(Nb) \geq dep(Nb) \geq L_{min}((N-1)b) = \frac{N-1}{N} L_{min}(Nb)$$

$$\Rightarrow \quad dep \sim L_{min} \text{ when } N \longrightarrow +\infty$$

$$dep(Nb) \geq L_{min}(Nb) - \underbrace{L_{min}(b)}_{\text{"constant"}}$$

$$\Rightarrow \quad dep(Nb) \leq L_{min}(Nb) \leq dep(Nb) + K$$

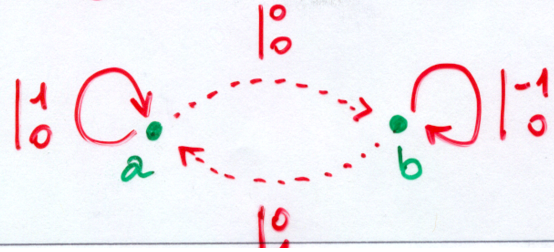
Looking for zero weight cycles

Remarks :

- Computability of conditional SUREs.
  $\Rightarrow$ UNDECIDABLE !
- Existence of an elementary cycle of zero weight.
  $\Rightarrow$ NP_complete !
- Existence of a multicycle of zero weight.
  $\Rightarrow$ Polynomial ☺
- Existence of a cycle of zero weight.
  $\Rightarrow$ Polynomial.

$\boxed{\text{key structure } G'}$

G' subgraph induced by all edges that belong to a multicycle (i.e union of cycles) of zero weight.



Karp, Miller & Winograd's technique

3 key-remarks:

(i) a zero weight cycle is a zero weight multicycle
⇒ Searching in G' is sufficient.

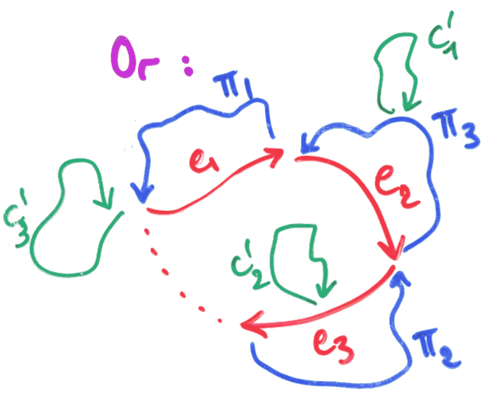
(ii) a cycle belongs to a strongly connected comp.
⇒ Searching in each SCC is sufficient.

(iii) if G' is strongly connected, there exists a cycle of zero weight.

For (iii):

Either, use Euler's Theorem.

Or:



$\text{cycle} = \sum_i C_i \Rightarrow$ zero weight.

$e_i \in C_i$

$C_i = \underbrace{e_i + \pi_i}_{\text{cycle}} + C_i'$

—— cycle that traverses all vertices.

Karp, Miller & Winograd's decomposition

KMW(G) {

- Build G' subgraph of zero weight multicycles.
- Compute the SCC of G': $G'_1, \dots, G'_k$

  (i) if $k=0$, G' is empty, return TRUE.

  (ii) if $k=1$, G' is strongly connected, return FALSE

  (iii) otherwise return $\text{AND}_{i=1}^{k} \text{KMW}(G'_i)$.

}

Then $\text{KMW}(G) = \text{TRUE} \iff$ No cycle of zero weight.

Def: d is the depth of the decomposition.

$d=0$ if G is acyclic {CONVENTION}.

$d=1$ all SCC of G have an empty G'.

$\vdots$

Property:

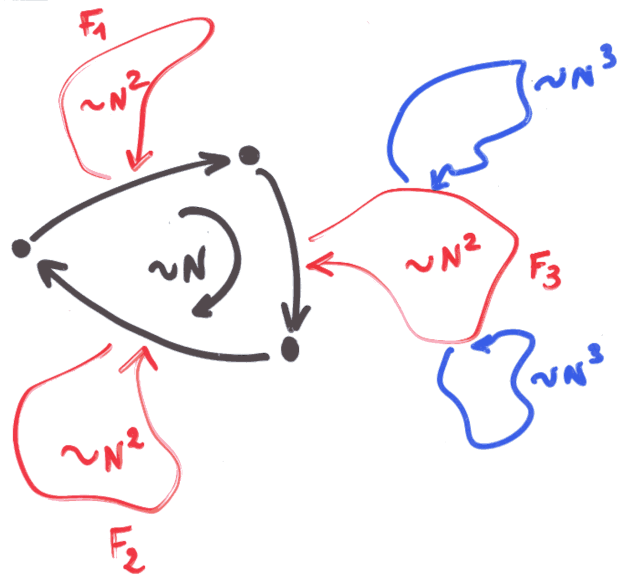
$$\boxed{\begin{array}{l} d \leq n \text{ if } G \text{ is computable.} \\ d \leq n+1 \text{ otherwise.} \end{array}}$$

proof is not straightforward. Wait for schedules...

Length of the longest path

$\Box$ n-cube of size 1.
if $\lambda N \Box \subset \mathcal{P}$ then there exists a dependence path of length $\Omega(N^d)$.

INTUITIVE PROOF:



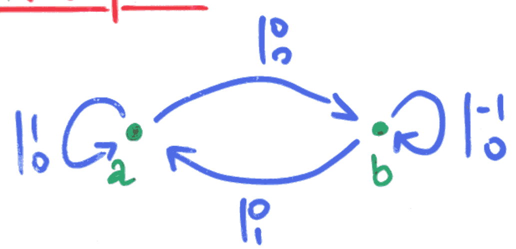
in black: one SCC of G: G_1.
in red: the SCCs of G' for G_1, F_1, F_2 & F_3.
in blue: the G' for F_3.

...

CLEAN PROOF:

Be careful that the path lies inside the domain $\mathcal{P}$.

Example:

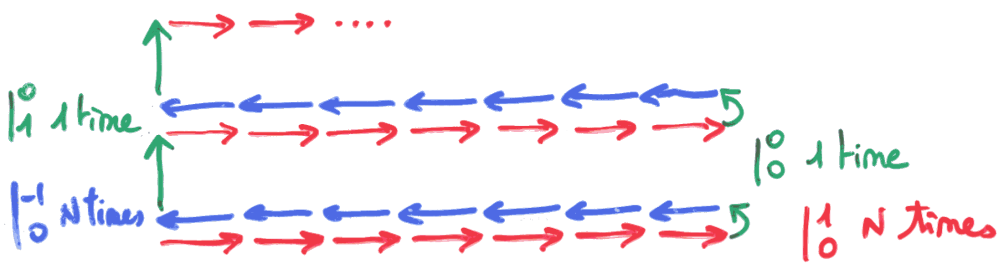


G

G'

$\emptyset$ $\emptyset$ $d=2$

Dependence path:



$\Rightarrow$ path of length $\Omega(N^2)$.

Computing G' in polynomial time

For a path q, let q_e the number of occurrences of an edge e in q. $\Rightarrow$ Vector of size $\#E$.

Euler's thm:

q is a union of cycles $\Leftrightarrow$ in & out-edges are equally used $\Leftrightarrow$ $q \geq 0$ & $Cq = 0$

connection matrix

$$\sum_{e \to v} q_e = \sum_{v \xrightarrow{e}} q_e \quad \forall v.$$

(NON TRIVIAL) Multicycle of zero weight:

$$\{ q \geq 0 ; q \neq 0 ; Cq = 0 ; Wq = 0 \}$$

weight matrix

$e \in G' \Leftrightarrow \exists q$ integral, $q \geq 0$, $q_e \geq 1$, $Cq = Wq = 0$

$\Leftrightarrow \exists q$ rational, $q \geq 0$, $q_e \geq 1$, $Cq = Wq = 0$

$\Rightarrow$ Polynomial for each edge.

This is KMW's technique:

$\Rightarrow$ $\#E$ linear programs.

With a single linear program

$e \in G' \iff v_e = 0$ for any optimal solution of

$$\min \left\{ \sum_e v_e \;\middle|\; q \geqslant 0,\ v \geqslant 0,\ q+v \geqslant 1,\ Cq=0,\ Wq=0 \right\} \quad (*)$$

lemma 1: for any optimal solution (q,v) of $(*)$

- $q_e \neq 0 \iff v_e = 0$
- $q_e = 0 \iff v_e = 1$

proof: a) $q_e = 0 \Rightarrow v_e \geqslant 1 \Rightarrow v_e = 1$ (minimization).

b) $v_e = 0 \Rightarrow q_e \neq 0$.

c) Let $q_0 = \min \{ q_e \mid q_e \neq 0 \}$

Let $q'_e = q_e / q_0$, $v'_e = 0$ if $q_e \neq 0$, $v'_e = v_e$ otherwise.
Better solution except if $q_e \neq 0 \Rightarrow v_e = 0$.

lemma 2: for any optimal solution (q,v) of $(*)$

- $v_e = 0 \iff e \in G'$

proof: a) q is a multicycle of zero weight. Thus,
$v_e = 0 \Rightarrow q_e \neq 0 \Rightarrow e \in q \Rightarrow e \in G'$.

b) Let $e \in G'$, $\exists \tilde{q}_e \geqslant 0$, $C\tilde{q}_e = W\tilde{q}_e = 0$ & $\tilde{q}_e \neq 0$.

Form $q' = q + \tilde{q}$, $v'_f = v_f$ if $f \neq e$ & $v'_e = 0$.
Better solution except if $v_e = 0$.

$\Rightarrow$ (i) A single linear program.

(ii) An interesting interpretation in the dual.

Considering the dual problem

$$\max \left\{ \sum_e z_e \;\middle|\; 0 \leq z \leq 1,\ X w(e) + p_{y_e} - p_{x_e} \geq z_e \right\}$$

lemma 3: for any optimal solution (X, p, z)

- $e \in G' \Leftrightarrow X.w(e) + p_{y_e} - p_{x_e} = 0$
- $e \notin G' \Leftrightarrow X.w(e) + p_{y_e} - p_{x_e} \geq 1$

proof: Use complementary slackness theorem

$$\max\{cx \mid Ax \leq b\} = \min\{yb \mid y \geq 0,\ yA = c\}$$

and either there is an opt. sol. x_0 with $a_i x_0 < b_i$

or there is an opt. sol. y_0 with $y_0 > 0$.

In other words:

$$\boxed{\exists \text{ opt. sol. } x_0 \text{ with } a_i x_0 < b_i \Leftrightarrow y_0 = 0 \text{ in any opt. sol.}}$$

Correspondance:

$z_e \geq 0 \rightsquigarrow w_e$ where $q_e + v_e = 1 + w_e$

$z_e \leq 1 \longrightarrow v_e$

$Xw(e) + p_{y_e} - p_{x_e} \geq z_e \rightsquigarrow q_e$

Back to scheduling

We look for $\sigma : V \times \mathcal{P} \longrightarrow \mathbb{N}$

or any totally ordered set.

such that:

$$\sigma(v,p) > \sigma(u, p-\omega(e)) \quad \text{if} \quad u \xrightarrow{\omega(e)} v$$

We schedule with $\sigma(u,p) = X.p + \rho_u$

Then:

$$(X.p + \rho_v) - (X.(p-\omega(e)) + \rho_u) = X.\omega(e) + \rho_v - \rho_u$$

$$\geqslant 1 \quad \text{if } e \notin G'$$

$$= 0 \quad \text{if } e \in G'$$

$\Rightarrow$ Multi dimensional scheduling

(lexicographic order: "hours", "days", "seconds", ...)

$e \notin G' \Rightarrow$ two operations computed at different hours.

$e \in G' \Rightarrow$ Same hour, need to consider minutes...

d dimensions $\Rightarrow O(N^d)$ time steps.

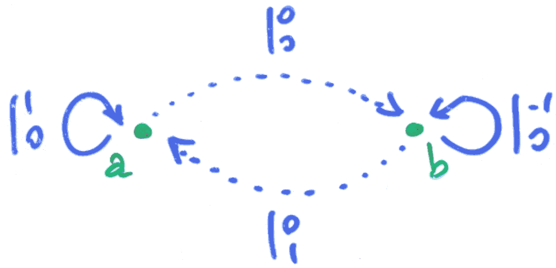
$\Rightarrow$ Asymptotically optimal.

(since dependence path is $\Omega(N^d)$).

Back to example 1

$e = (u,v)$

$x\, w(e) + \rho_v - \rho_u$



First level: $x_1\binom{1}{0} = x_1\binom{-1}{0} = 0$

$x_1\binom{0}{0} + \rho_b - \rho_a \geq 1$ & $x_1\binom{0}{1} + \rho_a - \rho_b \geq 1$

$\Rightarrow \quad x_1 = \binom{0}{2} \quad \rho_b = 1 \quad \rho_a = 0$

Second level:

for a: $x_{21}\binom{1}{0} \geq 1 \quad \Rightarrow \quad x_{21} = \binom{1}{0}$

for b: $x_{22}\binom{-1}{0} \geq 1 \quad \Rightarrow \quad x_{22} = \binom{-1}{0}$

Meaning:

a(i,j) computed at step $(2j, i)$

b(i,j) computed at step $(2j+1, -i)$

Schedule for example 1

- SURE: $1 \le i,j \le N$

$$\begin{cases} a(i,j) = a(i-1,j) + b(i,j-1) \\ b(i,j) = b(i+1,j) + a(i,j) \end{cases}$$

- Schedule: $a(i,j) \rightsquigarrow (2j, i)$
  $b(i,j) \rightsquigarrow (2j+1, -i)$

- Code with loops:

```
DO j=1, N
   DO i=1, N
      a(i,j) = a(i-1,j) + b(i,j-1)
   ENDDO
   DO i=N, 1, -1
      b(i,j) = b(i+1,j) + a(i,j)
   ENDDO
ENDDO
```

2 "1" 1 2

two levels: $e \in G' \Rightarrow$ level 2 dependence
$e \notin G' \Rightarrow$ level 1 dependence
or ∞ at first depth.

Linear independence of the scheduling vectors

For each vertex v, $(X_v^1, \dots X_v^{d_v-1})$ are linearly independent & $(X_v^1, \dots, X_v^{d_v})$ also if G is computable.

Proof:
Suppose $X_v^1, \dots, X_v^{k-1}$ are linearly independent and $X_v^1, \dots, X_v^k$ are linearly dependent, $1 \leq k < d_v$

$\Rightarrow X_v^1 = 0$ if $k=1$ or $X_v^k = \sum_{i=1}^{k-1} \lambda_i X_v^i$ if $k>1$.

Let C be a cycle of G_k, the SCC at depth k that contains v.

$\forall i<k,\ X_v^i\,\omega(C) = 0 \Rightarrow X_v^k\,\omega(C) = 0$.

since $\forall e \in C,\ X_v^i\,\omega(e) + \rho_{y_e}^i - \rho_{x_e}^i = 0$ if $i<k$.

BUT $X_v^k\,\omega(e) + \rho_{y_e}^k - \rho_{x_e}^k \geq z_e^k \Rightarrow X_v^k\,\omega(e) \geq \sum z_e^k$.

Thus, $e \in C \Rightarrow z_e^k = 0 \Rightarrow e \in G_k'$.

$\Rightarrow G_k' = G_k$ and G_k is not computable.

Consequences :

$$\boxed{\begin{array}{l} \bullet\ d_v \leq n+1 \text{ in general.} \\ \bullet\ d_v \leq n \text{ if } G \text{ is computable.} \end{array}}$$

- "hours" & "minutes" & "seconds" are linearly independent.

Optimality questions

- dependence path $\Omega(N^d)$ with schedule $O(N^d)$.

  $\Rightarrow$ optimal in terms of degree (d).

OPEN

- if dependence path $= \lambda N^d + o(N^d)$ can we find λ and schedule in $O(\lambda N^d)$?
- is dependence path $\sim \lambda N^d$ for some λ?

- when $d = 1$,

  * One equation:

  constant

  $|\text{dependence path} - \text{schedule}| \leq K$

  $\Rightarrow$ dependence path $= \lambda N + o(1)$.

  * Several equations:

  dependence path $\sim$ schedule

  $\Rightarrow$ dependence path $= \lambda N + o(N)$

BUT: dependence path $-$ schedule $\geq \log(N)$ dim 2

$\geq \sqrt[3]{N}$ dim 3

OPEN $\rightarrow$ $\geq \cdots$