Homogeneous versus hierarchical communication delay model:complexity and approximation

Rodolphe Giroudeau & Jean-Claude König
{rgirou@,konig}lirmm.fr

LIRMM, University of Montpellier II,
161 rue Ada,34392 Montpellier Cedex 5, France, UMR 5056

8 juin 2009

- We are interesting in one of the most important problem in Parallel computation : Scheduling problem

- We are interesting in one of the most important problem in Parallel computation : Scheduling problem
- An application is represented by a precedence graph

- We are interesting in one of the most important problem in Parallel computation : Scheduling problem
- An application is represented by a precedence graph
- PRAM, coarse grain, communication delays is not considered

- We are interesting in one of the most important problem in Parallel computation : Scheduling problem
- An application is represented by a precedence graph
- PRAM, coarse grain, communication delays is not considered
- Communication delays must be not neglected

Model

- Several scheduling models taking account communication

Model

- Several scheduling models taking account communication
- The most Popular : Homogeneous Communication Delay Model

Model

- Several scheduling models taking account communication
- The most Popular : Homogeneous Communication Delay Model
- Advantages :

- Several scheduling models taking account communication
- The most Popular : Homogeneous Communication Delay Model
- Advantages :
- Abstraction of the topology of the Parallel machine

Model

- Several scheduling models taking account communication
- The most Popular : Homogeneous Communication Delay Model
- Advantages :
- Abstraction of the topology of the Parallel machine
- Simplicity of the model

Model

- Several scheduling models taking account communication
- The most Popular : Homogeneous Communication Delay Model
- Advantages :
- Abstraction of the topology of the Parallel machine
- Simplicity of the model
- Not specified for a particular machine

Homogeneous model

Let be $G = (V, E)$ a graph

Homogeneous model

Let be $G = (V, E)$ a graph
$t_i =$ the starting time of the task i

Homogeneous model

Let be $G = (V, E)$ a graph
t_i = the starting time of the task i
π_i = processor on which the task i is executed

Homogeneous model

Let be $G = (V, E)$ a graph
t_i = the starting time of the task i
π_i = processor on which the task i is executed
p_i = duration of the task i

Homogeneous model

Let be $G = (V, E)$ a graph
t_i = the starting time of the task i
π_i = processor on which the task i is executed
p_i = duration of the task i
A Feasible schedule :

Homogeneous model

Let be $G = (V, E)$ a graph
t_i = the starting time of the task i
π_i = processor on which the task i is executed
p_i = duration of the task i
A Feasible schedule :
$\forall (i, j) \in E$ a value c_{ij} can be associated

Homogeneous model

Let be $G = (V, E)$ a graph
t_i = the starting time of the task i
π_i = processor on which the task i is executed
p_i = duration of the task i
A Feasible schedule :
$\forall (i,j) \in E$ a value c_{ij} can be associated

- If $\pi_i = \pi_j$ then $t_i + p_i \leq t_j$

Homogeneous model

Let be $G = (V, E)$ a graph

t_i = the starting time of the task i

π_i = processor on which the task i is executed

p_i = duration of the task i

A Feasible schedule :

$\forall (i, j) \in E$ a value c_{ij} can be associated

- If $\pi_i = \pi_j$ then $t_i + p_i \leq t_j$
- else $t_i + p_i + c_{ij} \leq t_j$

Homogeneous model

Let be $G = (V, E)$ a graph
t_i = the starting time of the task i
π_i = processor on which the task i is executed
p_i = duration of the task i
A Feasible schedule :
$\forall (i,j) \in E$ a value c_{ij} can be associated

- If $\pi_i = \pi_j$ then $t_i + p_i \leq t_j$
- else $t_i + p_i + c_{ij} \leq t_j$

The Objective : minimization of $C_{max} = \max_{i \in V}\{t_i + p_i\}$

Homogeneous model

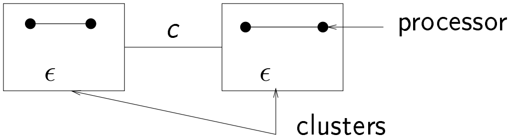
Let be $G = (V, E)$ a graph
t_i = the starting time of the task i
π_i = processor on which the task i is executed
p_i = duration of the task i
A Feasible schedule :
$\forall (i,j) \in E$ a value c_{ij} can be associated

- If $\pi_i = \pi_j$ then $t_i + p_i \leq t_j$
- else $t_i + p_i + c_{ij} \leq t_j$

The Objective : minimization of $C_{max} = \max_{i \in V}\{t_i + p_i\}$
If $c_{ij} = 0, \forall (i,j) \in E$ then we have the model without communication (PRAM model)

An investigation of the scheduling problem from a application on the parallel machine constituted by a set of clusters of processors
To require taking account hierarchical communication
Communications :

- intra-clusters (denoted by ϵ_{ij})
- extra-clusters (denoted by c_{ij})



Hierarchical Formulation

Let be a graph $G = (V, E)$

Hierarchical Formulation

Let be a graph $G = (V, E)$
t_i = the starting time of the task i

Hierarchical Formulation

Let be a graph $G = (V, E)$

t_i = the starting time of the task i

(Π^i, π^i_k) = cluster and processor on which the task i is executed

Hierarchical Formulation

Let be a graph $G = (V, E)$
t_i = the starting time of the task i
(Π^i, π_k^i) = cluster and processor on which the task i is executed
p_i = duration of the task i

Hierarchical Formulation

Let be a graph $G = (V, E)$
t_i = the starting time of the task i
(Π^i, π^i_k) = cluster and processor on which the task i is executed
p_i = duration of the task i
Feasible schedule :

Hierarchical Formulation

Let be a graph $G = (V, E)$
t_i = the starting time of the task i
(Π^i, π^i_k) = cluster and processor on which the task i is executed
p_i = duration of the task i
Feasible schedule :
$\forall (i, j) \in E$ a couple a values (ϵ_{ij}, c_{ij}) can be associated with
$\epsilon_{ij} \leq c_{ij}$

Hierarchical Formulation

Let be a graph $G = (V, E)$
t_i = the starting time of the task i
(Π^i, π^i_k) = cluster and processor on which the task i is executed
p_i = duration of the task i
Feasible schedule :
$\forall (i, j) \in E$ a couple a values (ϵ_{ij}, c_{ij}) can be associated with
$\epsilon_{ij} \leq c_{ij}$

- $\Pi^i = \Pi^j$ and if $\pi^i_k = \pi^j_k$ then $t_i + p_i \leq t_j$

Hierarchical Formulation

Let be a graph $G = (V, E)$
t_i = the starting time of the task i
(Π^i, π^i_k) = cluster and processor on which the task i is executed
p_i = duration of the task i
Feasible schedule :
$\forall (i, j) \in E$ a couple a values (ϵ_{ij}, c_{ij}) can be associated with
$\epsilon_{ij} \leq c_{ij}$

- $\Pi^i = \Pi^j$ and if $\pi^i_k = \pi^j_k$ then $t_i + p_i \leq t_j$
- else if $\Pi^i = \Pi^j$ and if $\pi^i_k \neq \pi^j_{k'}$ with $k \neq k'$ then $t_i + p_i + \epsilon_{ij} \leq t_j$

Hierarchical Formulation

Let be a graph $G = (V, E)$
t_i = the starting time of the task i
(Π^i, π^i_k) = cluster and processor on which the task i is executed
p_i = duration of the task i
Feasible schedule :
$\forall (i,j) \in E$ a couple a values (ϵ_{ij}, c_{ij}) can be associated with $\epsilon_{ij} \leq c_{ij}$

- $\Pi^i = \Pi^j$ and if $\pi^i_k = \pi^j_k$ then $t_i + p_i \leq t_j$
- else if $\Pi^i = \Pi^j$ and if $\pi^i_k \neq \pi^j_{k'}$ with $k \neq k'$ then $t_i + p_i + \epsilon_{ij} \leq t_j$
- else $\Pi^i \neq \Pi^j$ then $t_i + p_i + c_{ij} \leq t_j$

Possible configurations

- Hierarchical Problems

Possible configurations

- Hierarchical Problems
- number machines per clusters 2, or m

Possible configurations

- Hierarchical Problems
- number machines per clusters 2, or m
- number of clusters ∞, M

Possible configurations

- Hierarchical Problems
- number machines per clusters 2, or m
- number of clusters ∞, M
- values of $c \in \{1, 2, \dots, \}$ and of $\epsilon \in \{0, 1, \dots\}$ with $\epsilon \leq c$

Possible configurations

- Hierarchical Problems
- number machines per clusters 2, or m
- number of clusters ∞, M
- values of $c \in \{1, 2, \ldots, \}$ and of $\epsilon \in \{0, 1, \ldots\}$ with $\epsilon \leq c$
- Heterogeneous or Homogeneous processors

Hierarchical model is a generalization of homogeneous model

- Several ways to find the homogeneous model from the hierarchical model :

Hierarchical model is a generalization of homogeneous model

- Several ways to find the homogeneous model from the hierarchical model :
  - Only one processor per cluster, $m = 1$

Hierarchical model is a generalization of homogeneous model

- Several ways to find the homogeneous model from the hierarchical model :
  - Only one processor per cluster, $m = 1$
  - Unbounded of processors in one cluster $M = 1, m = \infty$

Hierarchical model is a generalization of homogeneous model

- Several ways to find the homogeneous model from the hierarchical model :
  - Only one processor per cluster, $m = 1$
  - Unbounded of processors in one cluster $M = 1, m = \infty$
  - $\forall (i, j) \in E, c_{ij} = \epsilon_{ij}$

Using the three fields $\alpha|\beta|\gamma$ a schedule problem in Homogeneous model can be denoted in what follows :

$$\bar{P}|prec, c_{ij} = c \geq 1; p_i = 1|C_{max}$$

where

Using the three fields $\alpha|\beta|\gamma$ a schedule problem in Homogeneous model can be denoted in what follows :

$$\bar{P}|prec, c_{ij} = c \geq 1; p_i = 1|C_{max}$$

where

- $\bar{P}$ signifies unbounded number of processors

Notation for scheduling problem

Using the three fields $\alpha|\beta|\gamma$ a schedule problem in Homogeneous model can be denoted in what follows :

$$\bar{P}|prec, c_{ij} = c \geq 1; p_i = 1|C_{max}$$

where

- $\bar{P}$ signifies unbounded number of processors
- *prec* signifies that we have a not specified precedence graph

Notation for scheduling problem

Using the three fields $\alpha|\beta|\gamma$ a schedule problem in Homogeneous model can be denoted in what follows :

$$\bar{P}|prec, c_{ij} = c \geq 1; p_i = 1|C_{max}$$

where

- $\bar{P}$ signifies unbounded number of processors
- *prec* signifies that we have a not specified precedence graph
- $c_{ij} = c \geq 1$ the communication delay is equal to $c, \forall (i,j) \in E$

Notation for scheduling problem

Using the three fields $\alpha|\beta|\gamma$ a schedule problem in Homogeneous model can be denoted in what follows :

$$\bar{P}|prec, c_{ij} = c \geq 1; p_i = 1|C_{max}$$

where

- $\bar{P}$ signifies unbounded number of processors
- *prec* signifies that we have a not specified precedence graph
- $c_{ij} = c \geq 1$ the communication delay is equal to $c, \forall (i,j) \in E$
- $p_i = 1$ the duration is equal to $1, \forall i \in V$

Notation for scheduling problem

Using the three fields $\alpha|\beta|\gamma$ a schedule problem in Homogeneous model can be denoted in what follows :

$$\bar{P}|prec, c_{ij} = c \geq 1; p_i = 1|C_{max}$$

where

- $\bar{P}$ signifies unbounded number of processors
- *prec* signifies that we have a not specified precedence graph
- $c_{ij} = c \geq 1$ the communication delay is equal to $c, \forall (i,j) \in E$
- $p_i = 1$ the duration is equal to $1, \forall i \in V$
- C_{max} the objective is the minimization of the length of the schedule

Notation for hierarchical scheduling problem

As the same as previously, a schedule problem in Hierarchical model can be denoted in what follows :
$\bar{P}(Pl \geq 2)|prec; (c_{ij}, \epsilon_{ij}) = (c, c'); p_i = 1|C_{max}$ where

Notation for hierarchical scheduling problem

As the same as previously, a schedule problem in Hierarchical model can be denoted in what follows :
$\bar{P}(Pl \geq 2)|prec; (c_{ij}, \epsilon_{ij}) = (c, c'); p_i = 1|C_{max}$ where

- $\bar{P}(Pl \geq 2)$ signify unbounded number of clusters with at most two processors

Notation for hierarchical scheduling problem

As the same as previously, a schedule problem in Hierarchical model can be denoted in what follows :
$\bar{P}(Pl \geq 2)|prec; (c_{ij}, \epsilon_{ij}) = (c, c'); p_i = 1|C_{max}$ where

- $\bar{P}(Pl \geq 2)$ signify unbounded number of clusters with at most two processors
- *prec* signify that we have a not specified precedence graph

Notation for hierarchical scheduling problem

As the same as previously, a schedule problem in Hierarchical model can be denoted in what follows :
$\bar{P}(Pl \geq 2)|prec; (c_{ij}, \epsilon_{ij}) = (c, c'); p_i = 1|C_{max}$ where

- $\bar{P}(Pl \geq 2)$ signify unbounded number of clusters with at most two processors
- *prec* signify that we have a not specified precedence graph
- $(c_{ij}, \epsilon_{ij}) = (c, c')$ the couple of communication delay

Notation for hierarchical scheduling problem

As the same as previously, a schedule problem in Hierarchical model can be denoted in what follows :
$\bar{P}(Pl \geq 2)|prec; (c_{ij}, \epsilon_{ij}) = (c, c'); p_i = 1|C_{max}$ where

- $\bar{P}(Pl \geq 2)$ signify unbounded number of clusters with at most two processors
- *prec* signify that we have a not specified precedence graph
- $(c_{ij}, \epsilon_{ij}) = (c, c')$ the couple of communication delay
- $p_i = 1$ the duration is equal to $1, \forall i \in V$

Notation for hierarchical scheduling problem

As the same as previously, a schedule problem in Hierarchical model can be denoted in what follows :
$\bar{P}(Pl \geq 2)|prec; (c_{ij}, \epsilon_{ij}) = (c, c'); p_i = 1|C_{max}$ where

- $\bar{P}(Pl \geq 2)$ signify unbounded number of clusters with at most two processors
- *prec* signify that we have a not specified precedence graph
- $(c_{ij}, \epsilon_{ij}) = (c, c')$ the couple of communication delay
- $p_i = 1$ the duration is equal to $1, \forall i \in V$
- C_{max} the objective is the minimization of the length of the schedule

Another hierarchical model

- Malleable tasks

Another hierarchical model

- Malleable tasks
  - A malleable tasks is a computational unit which may be executed on any arbitrary number of processors,

Another hierarchical model

- Malleable tasks
  - A malleable tasks is a computational unit which may be executed on any arbitrary number of processors,
  - its execution time depending on the amount of resources allotted to it

Another hierarchical model

- Malleable tasks
  - A malleable tasks is a computational unit which may be executed on any arbitrary number of processors,
  - its execution time depending on the amount of resources allotted to it

1. Compute the allotment of all tasks

Another hierarchical model

- Malleable tasks
  - A malleable tasks is a computational unit which may be executed on any arbitrary number of processors,
  - its execution time depending on the amount of resources allotted to it

1. Compute the allotment of all tasks
2. Schedule the obtained multiprocessor task graph

- Malleable tasks
  - A malleable tasks is a computational unit which may be executed on any arbitrary number of processors,
  - its execution time depending on the amount of resources allotted to it

  1. Compute the allotment of all tasks
  2. Schedule the obtained multiprocessor task graph

- Polynomial-time approximation algorithms are existed according to the type of precedence graph (independent tasks, chains, trees, general precedence graphs) [Trystram,Mounie, Rapine, Dutot]

The aim

Determinate

The aim

Determinate

- Theoretical aspect of scheduling problems

Determinate

- Theoretical aspect of scheduling problems
- Upper bounds (Positives results)

The aim

Determinate

- Theoretical aspect of scheduling problems
- Upper bounds (Positives results)
- Lower bounds (Negatives results)

Homogeneous model : Negative result (1)

- $\forall i \in V, p_i = 1, \forall (i,j) \in E, c_{ij} = 1$
- Unbounded number of processors

$\nexists$ ρ-approximation with $\rho < \frac{7}{6}$, unless $\mathcal{P} = \mathcal{NP}$
[Hoogeveen,Lenstra,Veltman 94]

- $\forall i \in V, p_i = 1, \forall (i,j) \in E, c_{ij} = 1$
- bounded number of processors

$\nexists$ ρ-approximation with $\rho < \frac{5}{4}$, unless $\mathcal{P} = \mathcal{NP}$
[Hoogeveen,Lenstra,Veltman 94]

Homogeneous model : Negative result (2)

- $\forall i \in V, p_i = 1, \forall (i,j) \in E, c_{ij} = c \geq 2$
- bounded number of processors

$\nexists$ ρ-approximation with $\rho < 1 + \frac{1}{c+3}$, unless $\mathcal{P} = \mathcal{NP}$
[Bampis,Giannakos,König96]

- $\forall i \in V, p_i = 1, \forall (i,j) \in E, c_{ij} = c \geq 2$
- unbounded number of processors

$\nexists$ ρ-approximation with $\rho < 1 + \frac{1}{c+4}$, unless $\mathcal{P} = \mathcal{NP}$[Giroudeau,König,Palaysi,Moulaï08]

Hierarchical model : Negative result (1)

- $\forall i \in V, p_i = 1, \forall (i,j) \in E, c_{ij} = 1$ and $\epsilon_{ij} = 0$
- Unbounded number of clusters, 2 identical processors per cluster
- $\bar{P}(P_2)|prec;(c_{ij},\epsilon_{ij}) = (1,0); p_i = 1|C_{max}$

$\nexists$ ρ-approximation with $\rho < \frac{5}{4}$, unless if $\mathcal{P} = \mathcal{NP}$
[Bampis,Giroudeau,König 99]

- $\forall i \in V, p_i = 1, \forall (i,j) \in E, c_{ij} = 2$ and $\epsilon_{ij} = 1$
- Unbounded number of clusters, 2 identical processors per cluster

$\nexists$ ρ-approximation with $\rho < \frac{6}{5}$, unless if $\mathcal{P} = \mathcal{NP}$ [Giroudeau 04]

- $\forall i \in V, p_i = 1, \forall (i,j) \in E, c_{ij} = c$ and $\epsilon_{ij} = c'$ with $c \geq c'$
- Unbounded number of clusters, $m \geq 2$ identical processors per cluster

$\nexists$ ρ-approximation with $\rho < 1 + \frac{1}{c+3}$, unless if $\mathcal{P} = \mathcal{NP}$
[Giroudeau,König 04]

Complexity results

Number of processors	Com. Delay (c_{ij}, ϵ_{ij})	Complexity
$\bar{P}$	$(1,1)$	$\rho \geq \frac{7}{6}$
$\bar{P}$	(c,c)	$\rho \geq \frac{c+5}{c+4}$
P	$(1,1)$	$\rho \geq \frac{4}{3}$
P	(c,c)	$\rho \geq \frac{c+4}{c+3}$
$\bar{P}(P_2)$	$(1,0)$	$\rho \geq \frac{4}{3}$
$\bar{P}(P_2)$	$(c,c'),\ c > c' \geq 1$	$\rho \geq \frac{c+4}{c+3}$
$P(P_2)$ *bipartite*	$(1,1)$	$\rho \geq \frac{4}{3}$

Polynomial-time reduction

- For an bounded number of machines (processors, clusters)

Polynomial-time reduction

- For an bounded number of machines (processors, clusters)
  - Graph theory : subgraphs and supergraphs

Polynomial-time reduction

- For an bounded number of machines (processors, clusters)
  - Graph theory : subgraphs and supergraphs
    - *Clique* and *Balanced Bipartite Complete Graph*

Polynomial-time reduction

- For an bounded number of machines (processors, clusters)
  - Graph theory : subgraphs and supergraphs
    - *Clique* and *Balanced Bipartite Complete Graph*
- For an unbounded number of machines (processors, clusters)

Polynomial-time reduction

- For an bounded number of machines (processors, clusters)
  - Graph theory : subgraphs and supergraphs
    - *Clique* and *Balanced Bipartite Complete Graph*
- For an unbounded number of machines (processors, clusters)
  - Logic : Propositional logic

- For an bounded number of machines (processors, clusters)
  - Graph theory : subgraphs and supergraphs
    - *Clique* and *Balanced Bipartite Complete Graph*
- For an unbounded number of machines (processors, clusters)
  - Logic : Propositional logic
    - 3*SAT* and a variant of 3*SAT* in where the occurrence of variables is limited (the definition is more sophisticated as 3*SAT*)

Polynomial-time reduction

- For an bounded number of machines (processors, clusters)
  - Graph theory : subgraphs and supergraphs
    - *Clique* and *Balanced Bipartite Complete Graph*
- For an unbounded number of machines (processors, clusters)
  - Logic : Propositional logic
    - $3SAT$ and a variant of $3SAT$ in where the occurrence of variables is limited (the definition is more sophisticated as $3SAT$)

A lower bound implies that there is non $\mathcal{PTAS}$ for all previous scheduling problems

Homogeneous model : Positive result

- $\exists \frac{4}{3}$-approximation algorithm for the problem $\bar{P}|p_i = 1, c_{ij} = 1|C_{max}$[Munier,König 97]

Homogeneous model : Positive result

- $\exists \frac{4}{3}$-approximation algorithm for the problem $\bar{P}|p_i = 1, c_{ij} = 1|C_{max}$[Munier,König 97]
- Integer Linear Programming Formulation

$$PL_I \quad \left\{ \begin{array}{lr} & min C_{max} \\ \forall (i,j) \in E, & x_{ij} \in \{0,1\} \\ \forall i \in V, & t_i \geq 0 \\ \forall (i,j) \in E, & t_i + 1 + x_{ij} \leq t_j \\ \forall i \in V - U, & \sum_{j \in \Gamma^+(i)} x_{ij} \geq |\Gamma^+(i)| - 1 \\ \forall i \in V - Z, & \sum_{j \in \Gamma^-(i)} x_{ji} \geq |\Gamma^-(i)| - 1 \\ \forall i \in V & t_i + 1 \leq C_{max} \end{array} \right.$$

Procedure

- An integrity constraint is relaxed $x_{ij} \in \{0,1\} \Rightarrow e_{ij} \in [0,1]$, $PL_I \Rightarrow PL_I^{inf}$

Procedure

- An integrity constraint is relaxed $x_{ij} \in \{0,1\} \Rightarrow e_{ij} \in [0,1]$, $PL_I \Rightarrow PL_I^{inf}$
- and feasible schedule is produced by rounding

Procedure

- An integrity constraint is relaxed $x_{ij} \in \{0, 1\} \Rightarrow e_{ij} \in [0, 1]$, $PL_I \Rightarrow PL_I^{inf}$
- and feasible schedule is produced by rounding
- $$\begin{cases} \text{if } e_{ij} < 0.5 & \Longrightarrow & x_{ij} = 0 \\ \text{if } e_{ij} \geq 0.5 & \Longrightarrow & x_{ij} = 1 \end{cases}$$

Procedure

- An integrity constraint is relaxed $x_{ij} \in \{0,1\} \Rightarrow e_{ij} \in [0,1]$, $PL_I \Rightarrow PL_I^{inf}$
- and feasible schedule is produced by rounding
- $$\begin{cases} \text{if } e_{ij} < 0.5 & \Longrightarrow \quad x_{ij} = 0 \\ \text{if } e_{ij} \geq 0.5 & \Longrightarrow \quad x_{ij} = 1 \end{cases}$$
- By induction, $t_i \leq \frac{4}{3} t_i^*$ with t_i^* the starting time of i given by the solution of PL_I^{inf}

Procedure

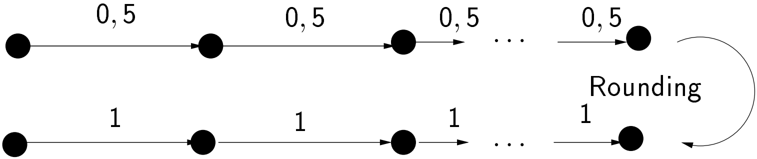
- An integrity constraint is relaxed $x_{ij} \in \{0,1\} \Rightarrow e_{ij} \in [0,1]$, $PL_I \Rightarrow PL_I^{inf}$
- and feasible schedule is produced by rounding
- $$\begin{cases} \text{if } e_{ij} < 0.5 & \Longrightarrow & x_{ij} = 0 \\ \text{if } e_{ij} \geq 0.5 & \Longrightarrow & x_{ij} = 1 \end{cases}$$
- By induction, $t_i \leq \frac{4}{3} t_i^*$ with t_i^* the starting time of i given by the solution of PL_I^{inf}
- So,

$$C_{max} \leq \frac{4}{3} C_{max}^{opt}$$

Tightness of the bound

The bound is tight :

a path with k vertices



$C_{max} = 2k - 1$ $\qquad$ $C_{opt} = k + \frac{k-1}{2} = \frac{3k}{2} - \frac{1}{2}$

Dilatation of length by a factor : $\frac{4}{3}$

Hierarchical model : Positive result

- $(2 - \frac{2}{2m+1})$-approximation algorithm for the problem $\bar{P}(P_m)|prec;(c_{ij},\epsilon_{ij}) = (1,0);p_i = 1|C_{max}$ with $m \geq 1$
  [Bampis,Giroudeau,König 99]

- $(2-\frac{2}{2m+1})$-approximation algorithm for the problem $\bar{P}(P_m)|prec;(c_{ij},\epsilon_{ij})=(1,0);p_i=1|C_{max}$ with $m\geq 1$ [Bampis,Giroudeau,König 99]
- $$\begin{cases} & \min C_{max} \\ \forall (i,j)\in E, & x_{ij}\in\{0,1\} \\ \forall i\in V, & t_i\geq 0 \\ \forall (i,j)\in E, & t_i+1+x_{ij}\leq t_j \\ \forall (i,j)\in E-U, & \sum_{j\in\Gamma^+(i)} x_{ij}\geq |\Gamma^+(i)|-2 \\ \forall (i,j)\in E-Z, & \sum_{j\in\Gamma^-(i)} x_{ji}\geq |\Gamma^-(i)|-2 \\ \forall i,j,k,l,m\in V, & x_{ji}+x_{jk}+x_{lk}+x_{lm}\geq 1 \\ (j,i),(j,k),(l,k),(l,m)\in E, & \\ \forall i,j,k,l,m\in V, & x_{ij}+x_{kj}+x_{kl}+x_{ml}\geq 1 \\ (i,j),(k,j),(k,l),(m,l)\in E, & \\ \forall i\in V, & t_i+1\leq C_{max} \end{cases}$$

Procedure

- An integrity constraint is relaxed $x_{ij} \in \{0, 1\} \Rightarrow e_{ij} \in [0, 1]$, $PL_I \Rightarrow PL_I^{inf}$

Procedure

- An integrity constraint is relaxed $x_{ij} \in \{0,1\} \Rightarrow e_{ij} \in [0,1]$, $PL_I \Rightarrow PL_I^{inf}$
- and feasible schedule is produced by rounding

Procedure

- An integrity constraint is relaxed $x_{ij} \in \{0, 1\} \Rightarrow e_{ij} \in [0, 1]$, $PL_I \Rightarrow PL_I^{inf}$
- and feasible schedule is produced by rounding
- $$\begin{cases} \text{if } e_{ij} < 0.25 & \Longrightarrow \quad x_{ij} = 0 \\ \text{if } e_{ij} \geq 0.25 & \Longrightarrow \quad x_{ij} = 1 \end{cases}$$

Procedure

- An integrity constraint is relaxed $x_{ij} \in \{0,1\} \Rightarrow e_{ij} \in [0,1]$, $PL_I \Rightarrow PL_I^{inf}$
- and feasible schedule is produced by rounding
- $$\begin{cases} \text{if } e_{ij} < 0.25 & \Longrightarrow \quad x_{ij} = 0 \\ \text{if } e_{ij} \geq 0.25 & \Longrightarrow \quad x_{ij} = 1 \end{cases}$$

- Another step is needed in order to get a feasible schedule

Procedure

- An integrity constraint is relaxed $x_{ij} \in \{0,1\} \Rightarrow e_{ij} \in [0,1]$, $PL_I \Rightarrow PL_I^{inf}$
- and feasible schedule is produced by rounding
- $$\begin{cases} \text{if } e_{ij} < 0.25 & \Longrightarrow \quad x_{ij} = 0 \\ \text{if } e_{ij} \geq 0.25 & \Longrightarrow \quad x_{ij} = 1 \end{cases}$$

- Another step is needed in order to get a feasible schedule
- By induction, $t_i \leq (2 - \frac{2}{2m+1})t_i^*$ with t_i^* the starting time of i given by the solution of PL_I^{inf}

- An integrity constraint is relaxed $x_{ij} \in \{0,1\} \Rightarrow e_{ij} \in [0,1]$, $PL_I \Rightarrow PL_I^{inf}$
- and feasible schedule is produced by rounding
- $$\begin{cases} \text{if } e_{ij} < 0.25 & \Longrightarrow & x_{ij} = 0 \\ \text{if } e_{ij} \geq 0.25 & \Longrightarrow & x_{ij} = 1 \end{cases}$$

- Another step is needed in order to get a feasible schedule
- By induction, $t_i \leq (2 - \frac{2}{2m+1})t_i^*$ with t_i^* the starting time of i given by the solution of PL_I^{inf}
- So,

$$C_{max} \leq (2 - \frac{2}{2m+1})C_{max}^{opt}$$

Procedure

- An integrity constraint is relaxed $x_{ij} \in \{0,1\} \Rightarrow e_{ij} \in [0,1]$, $PL_I \Rightarrow PL_I^{inf}$
- and feasible schedule is produced by rounding
- $$\begin{cases} \text{if } e_{ij} < 0.25 & \Longrightarrow \quad x_{ij} = 0 \\ \text{if } e_{ij} \geq 0.25 & \Longrightarrow \quad x_{ij} = 1 \end{cases}$$

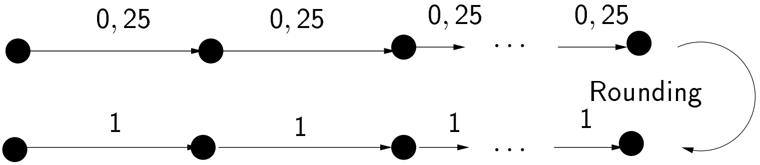
- Another step is needed in order to get a feasible schedule
- By induction, $t_i \leq (2 - \frac{2}{2m+1})t_i^*$ with t_i^* the starting time of i given by the solution of PL_I^{inf}
- So,

$$C_{max} \leq (2 - \frac{2}{2m+1})C_{max}^{opt}$$

- The bound is tight

The bound is tight :

a path with k vertices



$C_{max} \leq 2k$ $C_{opt} = k + \frac{k-1}{4} = \frac{5k}{4} - \frac{1}{4}$

Dilatation of length by $\frac{8}{5}$

Bounded version

- Using the Brent's Lemma (74)(The validity of this algorithm is based on the fact there is at most one matching between the tasks executed at time t_i and the tasks processed at time $t_i + 1$)

Bounded version

- Using the Brent's Lemma (74)(The validity of this algorithm is based on the fact there is at most one matching between the tasks executed at time t_i and the tasks processed at time $t_i + 1$)
- $\exists(1 + \frac{4}{3})$-approximation algorithm for the $P|p_i = 1, c_{ij} = 1|C_{max}$ (Improved by [Munier,Hanen 96] $\exists(1 + \frac{4}{3} - \frac{4}{3m})$-approximation algorithm)

- Using the Brent's Lemma (74)(The validity of this algorithm is based on the fact there is at most one matching between the tasks executed at time t_i and the tasks processed at time $t_i + 1$)
- $\exists(1 + \frac{4}{3})$-approximation algorithm for the $P|p_i = 1, c_{ij} = 1|C_{max}$ (Improved by [Munier,Hanen 96] $\exists(1 + \frac{4}{3} - \frac{4}{3m})$-approximation algorithm)
- $(1 + (2 - \frac{1}{m})\frac{8}{5})$-approximation algorithm for the problem $P(P_m)|prec; (c_{ij}, \epsilon_{ij}) = (1, 0); p_i = 1|C_{max}$

Analysis

- For the central problems $(UET - UCT)$ in scheduling Homogeneous/Hierarchical models are in class $\mathcal{APX}$

- For the central problems $(UET - UCT)$ in scheduling Homogeneous/Hierarchical models are in class $\mathcal{APX}$
- For the large communication delays $c \geq 2$ (or $(c_{ij} = c, \epsilon_{ij} = c')$ with $c > c' \geq 1$), it exists an-approximation algorithm A such that $A \in \mathcal{APX}$?

Approximation algorithm for Large communication delay in homogeneous model

- We consider the case of $c \geq 2$

Approximation algorithm for Large communication delay in homogeneous model

- We consider the case of $c \geq 2$
- $\bar{P}|prec; c_{ij} \geq 2; p_i = 1|C_{max} \in \mathcal{APX}$?

- We consider the case of $c \geq 2$
- $\bar{P}|prec; c_{ij} \geq 2; p_i = 1|C_{max} \in \mathcal{APX}$?
- We develop a new algorithm based on the solution given the problem $\bar{P}|prec; c_{ij} = 1; p_i = 1|C_{max}$

Strategy

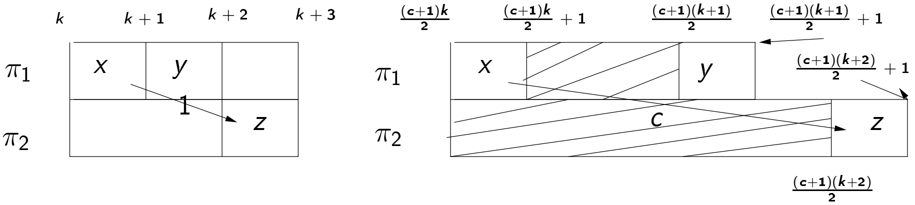
- First Step "Good" feasible schedule with unbounded number of processors σ^∞

Strategy

- First Step "Good" feasible schedule with unbounded number of processors σ^{∞}
- Second Step An assignment of the tasks is keeping

Strategy

- First Step "Good" feasible schedule with unbounded number of processors σ^{∞}
- Second Step An assignment of the tasks is keeping
- Third Step An expansion of the makespan, while preserving communication delays ($t_j^c \geq t_i^c + 1 + c$) for i and j with $(i, j) \in E$, processing on two different processors



Coefficient of expansion

- Let be two tasks i and j such that $(i,j) \in E$, which are processed on two different processors in the feasible schedule σ^{∞}

Coefficient of expansion

- Let be two tasks i and j such that $(i,j) \in E$, which are processed on two different processors in the feasible schedule σ^{∞}
- $t_i^c = d \times t_i$ and $t_j^c = d \times t_j$

Coefficient of expansion

- Let be two tasks i and j such that $(i,j) \in E$, which are processed on two different processors in the feasible schedule σ^{∞}
- $t_i^c = d \times t_i$ and $t_j^c = d \times t_j$
- After an expansion, $t_j^c \geq t_i^c + 1 + c$, and so $d \times t_i - d \times t_j \geq c + 1, \; d \geq \frac{c+1}{t_i - t_j}, \; d \geq \frac{c+1}{2} \Rightarrow d = \frac{(c+1)}{2}$

Ratio

- An expansion algorithm gives a feasible schedule

Ratio

- An expansion algorithm gives a feasible schedule
- An expansion algorithm gives a $\frac{2(c+1)}{3}-$approximation algorithm

Ratio

- An expansion algorithm gives a feasible schedule
- An expansion algorithm gives a $\frac{2(c+1)}{3}-$approximation algorithm
- $C^h_{max} \leq \frac{4}{3} C^{opt}_{max}$ by the First Step of an algorithm

Ratio

- An expansion algorithm gives a feasible schedule
- An expansion algorithm gives a $\frac{2(c+1)}{3}$ −approximation algorithm
- $C^h_{max} \leq \frac{4}{3} C^{opt}_{max}$ by the First Step of an algorithm
- $C^{h^*}_{max} = \frac{(c+1)}{2} C^h_{max}$ by the Third Step of an algorithm
- $C^{opt,c}_{max} \geq C^{opt}_{max}$

Ratio

- An expansion algorithm gives a feasible schedule
- An expansion algorithm gives a $\frac{2(c+1)}{3}-$approximation algorithm
- $C^h_{max} \leq \frac{4}{3} C^{opt}_{max}$ by the First Step of an algorithm
- $C^{h^*}_{max} = \frac{(c+1)}{2} C^h_{max}$ by the Third Step of an algorithm
- $C^{opt,c}_{max} \geq C^{opt}_{max}$
- $\frac{C^{h^*}_{max}}{C^{opt,c}_{max}} = \frac{\frac{(c+1)}{2} C^h_{max}}{C^{opt,c}_{max}} \leq \frac{\frac{(c+1)}{2} C^h_{max}}{C^{opt}_{max}} \leq \frac{\frac{(c+1)}{2} \frac{4}{3} C^{opt}_{max}}{C^{opt}_{max}} \leq \frac{2(c+1)}{3}$

Ratio

- An expansion algorithm gives a feasible schedule
- An expansion algorithm gives a $\frac{2(c+1)}{3}$−approximation algorithm
- $C^h_{max} \leq \frac{4}{3} C^{opt}_{max}$ by the First Step of an algorithm
- $C^{h*}_{max} = \frac{(c+1)}{2} C^h_{max}$ by the Third Step of an algorithm
- $C^{opt,c}_{max} \geq C^{opt}_{max}$
- $\frac{C^{h*}_{max}}{C^{opt,c}_{max}} = \frac{\frac{(c+1)}{2} C^h_{max}}{C^{opt,c}_{max}} \leq \frac{\frac{(c+1)}{2} C^h_{max}}{C^{opt}_{max}} \leq \frac{\frac{(c+1)}{2} \frac{4}{3} C^{opt}_{max}}{C^{opt}_{max}} \leq \frac{2(c+1)}{3}$
  - $\rho \leq \frac{2(c+1)}{3}$

- The previous algorithm may used to derive a polynomial-time with performance ratio equal to $(c+1)(1-\frac{1}{2m+1})$-approximation algorithm for the problem $\bar{P}(P_m)|prec;(c_{ij},\epsilon_{ij})=(c,c');p_i=1|C_{max}$ with $m \geq 1$

Approximation

# processors	Com. Delay (c_{ij}, ϵ_{ij})	Complexity & Approximation
$\bar{P}$	$(1,1)$	$\frac{7}{6} \leq \rho \leq \frac{4}{3}$
$\bar{P}$	(c,c)	$\frac{c+5}{c+4} \leq \rho \leq \frac{2(c+1)}{3}$
P	$(1,1)$	$\frac{4}{3} \leq \rho \leq \frac{7}{3}$
P	(c,c)	$\frac{c+4}{c+3} \leq \rho \leq \frac{7(c+1)}{6}$
$\bar{P}(P_2)$	$(1,0)$	$\frac{4}{3} \leq \rho \leq \frac{8}{5}$
$\bar{P}(P_2)$	$(c,c'),\ c > c' \geq 1$	$\frac{c+4}{c+3} \leq \rho \leq (c+1)(1-\frac{1}{2m+1})$

Conclusion on homogeneous versus hierarchical model

- Techniques used in polynomial-time transformations (complexity) resist to passage to hierarchical communication model

Conclusion on homogeneous versus hierarchical model

- Techniques used in polynomial-time transformations (complexity) resist to passage to hierarchical communication model
- Techniques used for approximation algorithm also resist to passage to hierarchical communication model

Future work ?

- Complexity results :

Future work ?

- Complexity results :
  - The lower bounds are the best lower bound ? It given by an impossibility theorem ? Another way to find the best lower bound ?
  - Approximation algorithms :

- Complexity results :
  - The lower bounds are the best lower bound ? It given by an impossibility theorem ? Another way to find the best lower bound ?
  - Approximation algorithms :
  - There exists a $\rho-$approximation with $\rho < 4/3$ for the scheduling problem $\bar{P}|p_i = 1, c_{ij} = 1|C_{max}$?

Future work ?

- Complexity results :
  - The lower bounds are the best lower bound ? It given by an impossibility theorem ? Another way to find the best lower bound ?
  - Approximation algorithms :
  - There exists a $\rho-$approximation with $\rho < 4/3$ for the scheduling problem $\bar{P}|p_i = 1, c_{ij} = 1|C_{max}$?
  - There exists a ρ-approximation algorithm with $\rho \in \mathbb{R}$ for the scheduling problem $\bar{P}|p_i = 1, c_{ij} = c \geq 2|C_{max}$?

Conclusion

Any questions ?