Université Polytechnique HAUTS-DE-FRANCE

RAILENIUM TEST & RESEARCH CENTRE

New Mathematical Programming Models for Scheduling Unrelated Parallel Machines with Heterogeneous Delays

By Abdessamad AIT EL CADI
UPHF

LAMIH UMR CNRS 8201

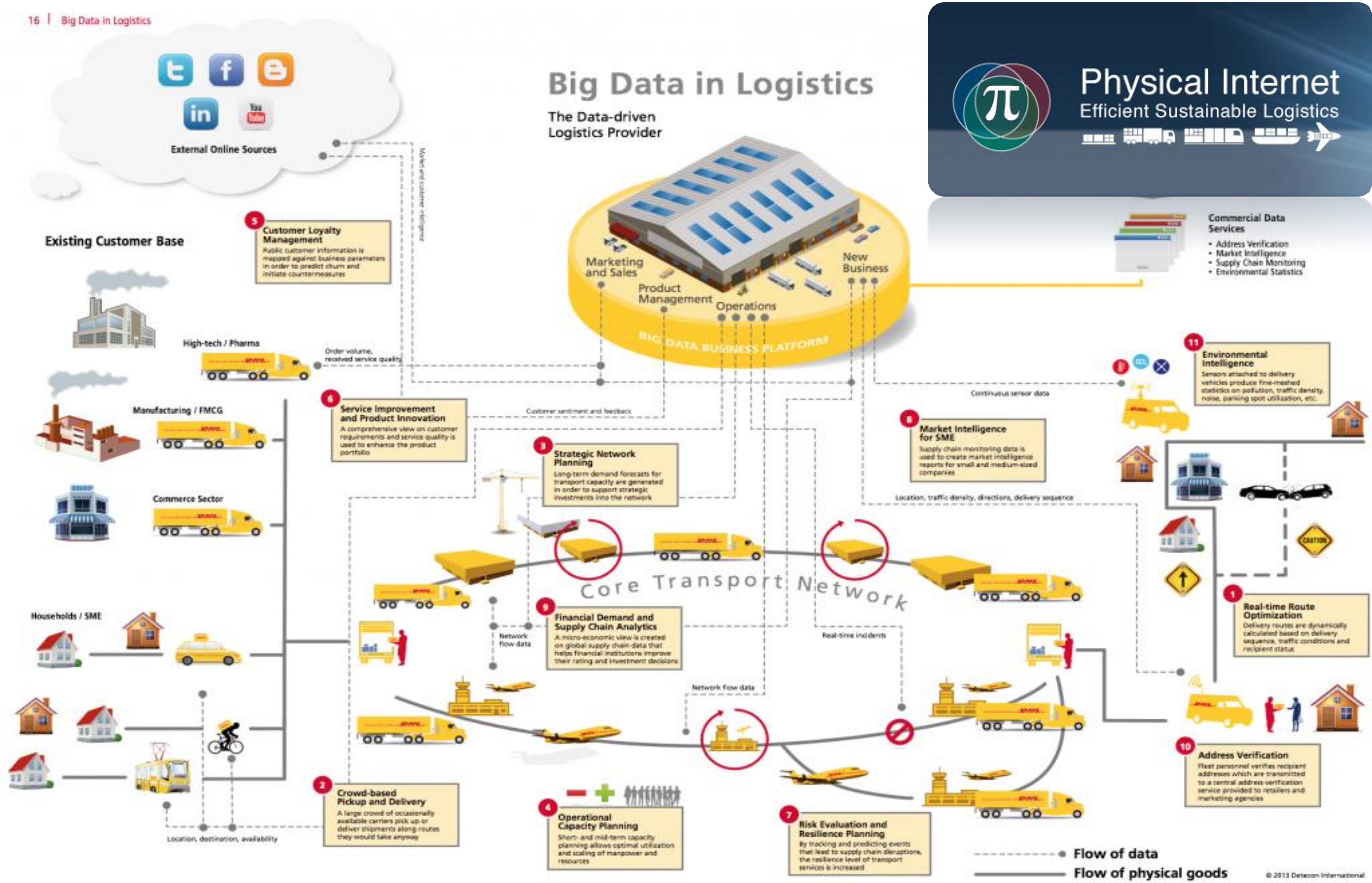
cnrs

INSTITUT CARNOT ARTS

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES HAUTS-DE-FRANCE

Outline

1. Introduction: Transport & Logistic



1. Introduction: Scheduling

- Machine scheduling: assigning a group of ordered jobs to an individual machine
- Work-force scheduling: assigning a group of ordered tasks to an individual worker
- Task Mapping in distributed systems
- Making a good scheduling decision requires understanding specific tradeoffs
  - job shop: control inventory while shipping orders on time
  - assembly line: promote resource efficiency and maintain adequate finished goods

1. Introduction: Scheduling

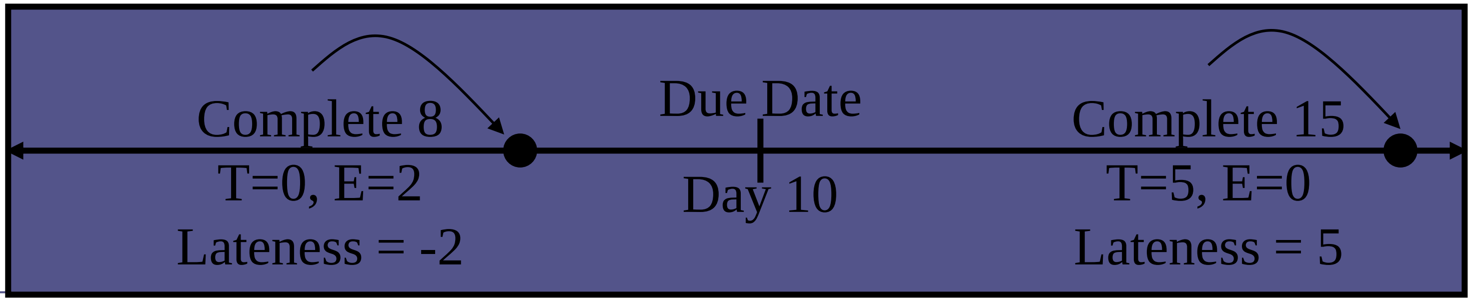
Efficiency-based Scheduling Criteria

- Job flow time: the amount of time elapsed from a job's entry into the shop until the job completes all processing.
- Makespan: the amount of time required to complete a pre-identified group of jobs.
- WIP: the amount of inventory in process.
- Inventory: the amount of raw material, WIP, and finished goods in stock.
- Utilization: the percentage of time a capacitated resource is used productively.

1. Introduction: Scheduling

Customer Service-based Scheduling Criteria

- Lateness: the difference between a job's completion date and its due date.
- Tardiness: the amount of time it takes to complete a job once its due date has passed.
- Earliness: the amount of time until a job's due date arrives once the job has been completed.



1. Introduction: Scheduling

- Benefits of scheduling:
  - ***Lower Cost***: less money in inventory
  - ***More Flexibility***: less disruptive to change backlog that work in process
  - ***Better Quality***: faster defect detection
  - ***Less Reliance on Forecasts***: cycle times below frozen zone allow make to order
  - ***Better Forecasts***: distant (inaccurate) forecasts are no longer needed

1. Introduction: Example

Problematic :

Complex systems, such as aeronautic, avionic, robotic and intelligent transportation systems are more and more complex:

- Computing demand is growing;
- One single processor is inadequate;
- Technology, Real-time, Flexibility and Energy efficiency constraints.

So : Heterogeneous architecture CPU & FPGA is a great choice.

Huong, G.N.T., Na, Y. and Kim, S.W., Applying frame layout to hardware design in FPGA for seamless support of cross calls in CPU-FPGA coupling architecture, Microprocessors and Microsystems, 35:462-472, 2011.

1. Introduction

Why is this a problem :

The main difficulty faced by designers and engineers using these complex systems:

- The separation of the application tasks between these resources (CPU and FPGA).
- They need methods and tools that help to do this mapping efficiently while considering all the constraints.
- The problem is how to assign tasks to the available resources in order to optimize some performance criterion such as the makespan, the load balance, energy consumption, etc.

1. Introduction

Literature Overview :

Scheduling (sequencing, planning, …) appears in different field such as :

- Assembly line balancing
- Resource-Constrained Project Scheduling
- Load Balancing problem
- Task Scheduling in parallel and distributed systems

Baker, K. R. and Trietsch, D., Principles of Sequencing and Scheduling, Wiley, 2009.

2. The Problem

Specific Problem

Separate a set of tasks, subject to precedence constraints (graph), over a set of heterogeneous processors with heterogeneous communication delays

2. The Problem

Problem classification

- MIMD (Multiple Instructions Multiple Data) architecture according to Flynn's taxonomy – 1966.

- R / pred, c_{ikjl} / C_{max}, based on the notation proposed by Graham et al. 1979.

M. J. Flynn "Very high-speed computing systems", Proc. IEEE, vol. 54, pp.1901 -1909 1966

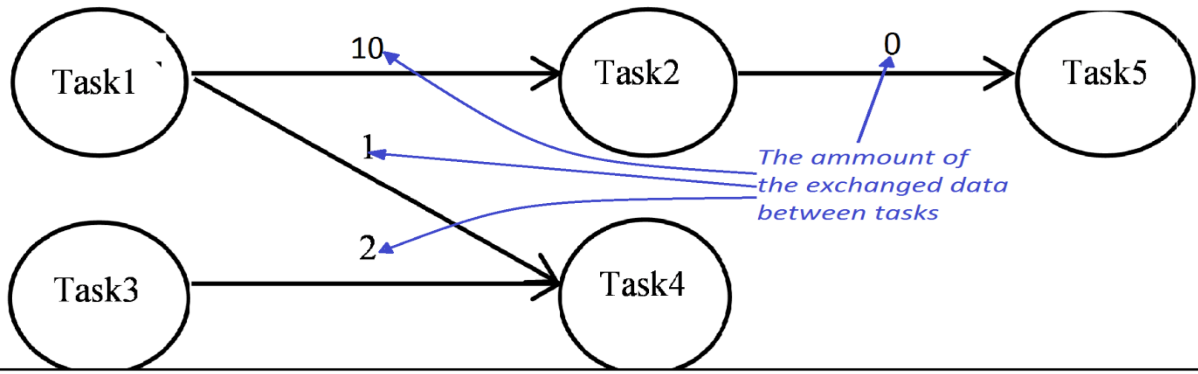
R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan (1979): Optimization and approximation in deterministic sequencing and scheduling: a survey, Ann. Discrete Math. 4, 287-326.

2. The Problem

A good solution would:

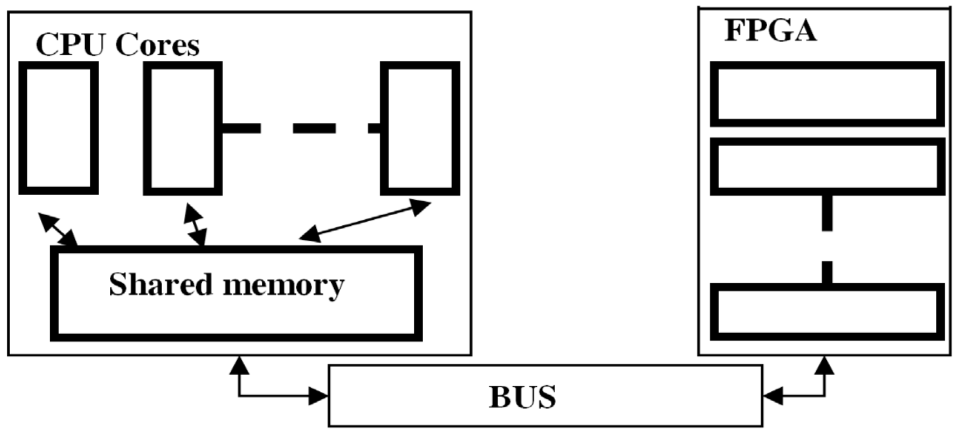
- Speed up the distribution and the design cycle
- Guarantee parallel execution
- Take into account all possible parallelism opportunities, then determine the “best” parallel execution
- Take into account all precedence constraints and communication delays.

Tasks graph



	Processing time			
Tasks	CPU1	CPU2	CPU3	FPGA
Task1	10	11	21	10
Task2	21	13	14	10
Task3	31	21	21	15
Task4	10	21	15	18
Task5	15	21	31	10

Computing network characteristics



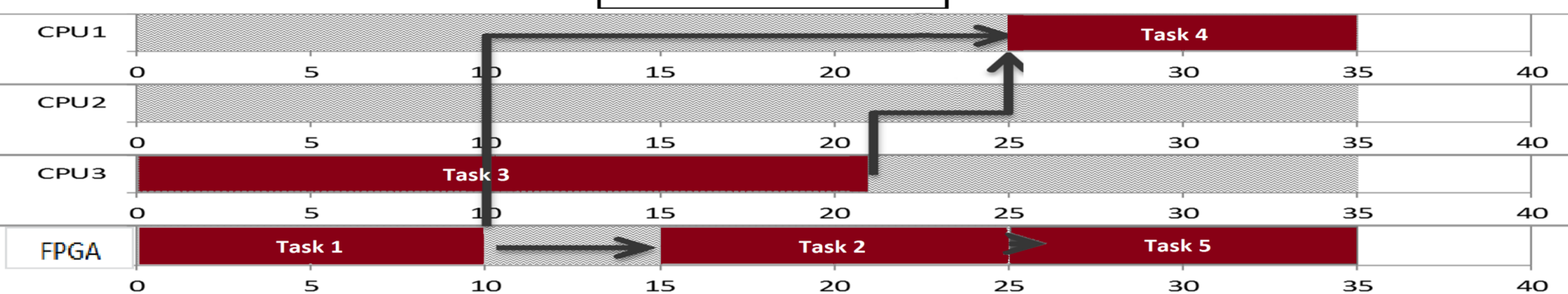
Communication access cost

	CPU1	CPU2	CPU3	FPGA
CPU1	1	3	3	4
CPU2	3	1	3	4
CPU3	3	3	1	4
FPGA	4	4	4	0

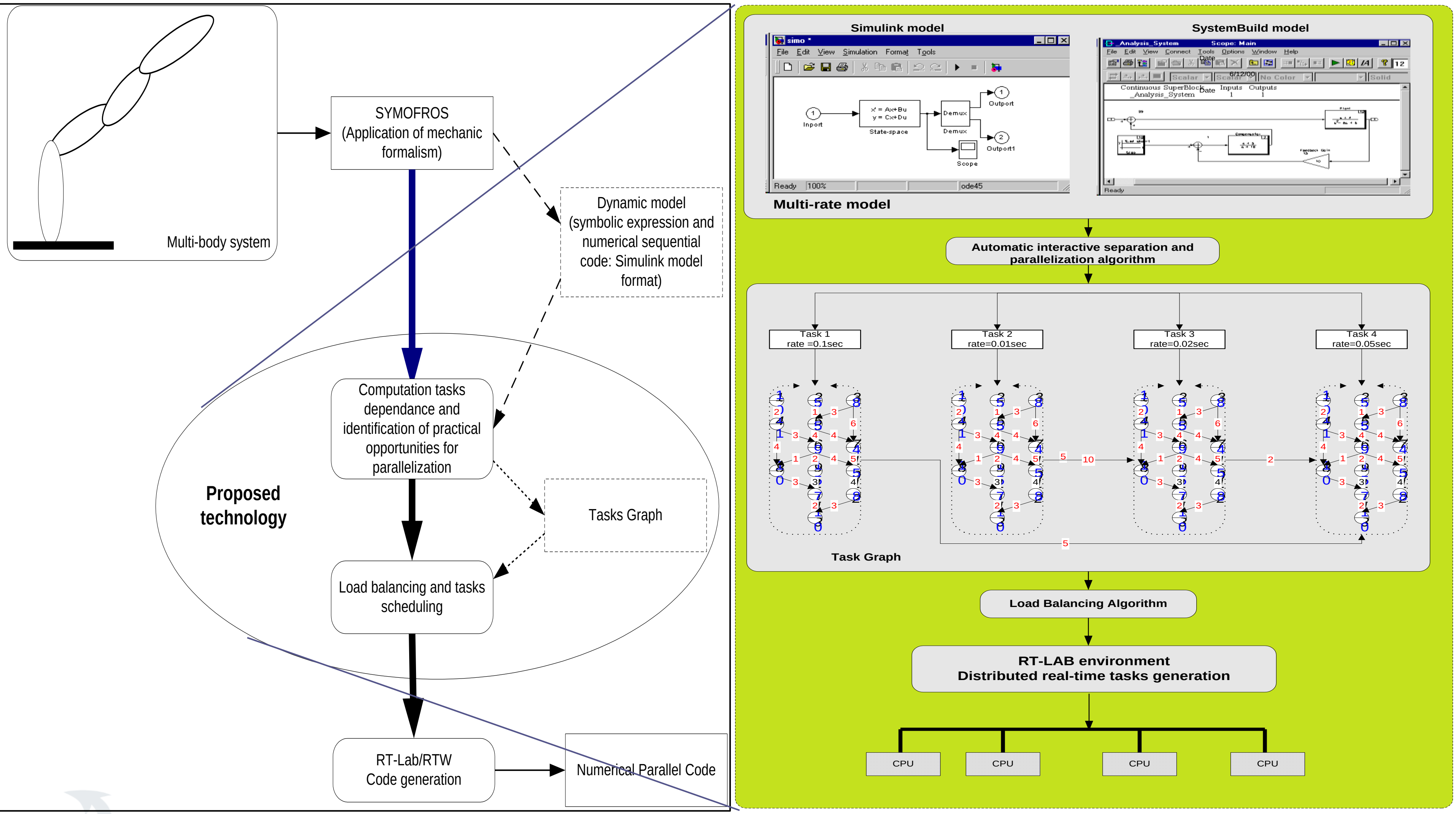
Communication rate

	CPU1	CPU2	CPU3	FPGA
CPU1	1	2	2	1
CPU2	2	1	2	1
CPU3	2	2	1	1
FPGA	1	1	1	2

Solution



Scheduling & Load Balancing over a Distributed Network



3. Mathematical Models

Approach

- We build a model for the general case, model that is a MIQCP;
- We linearize the model so that it become a MILP.
- We reduce the size (number of variables and constraints) of the linear model by exploiting the precedence graph and pruning any unnecessary constraints or variables.
- We add bounds on C_{max} and some general cuts to improve the running time.

3. Mathematical Model

Objectives :

- Minimize the makespan C_{max}.
- What is the minimal execution time using *m* processing units, where *m* is fixed.

Constraints :

- Precedence constraints
- Communication constraints
- Disjunctive constraints
- ...

3. Mathematical Models

Notation:

- N: Set of n tasks;
- M: Set of m processing units (CPUs/FPGAs);
- G=(N, A): a given directed acyclic graph, where N is the set of tasks and A set of arcs representing the precedence between tasks, i.e. (i, j) in A means that the task i must be performed before the task j.
- Pred(i): Set of tasks that precede the task i;
- t_{ik}: Processing time of the task i on processing unit k;
- $c_{ik,jl}$: The cost of direct communication between task I on processing unit k and the task j on processing unit l;
- Fk : Set of tasks that should not be assigned to the processing unit k.
- B: very large scalar value.

- Decision variables: s_i & x_{ik}

3. Mathematical Models

MIQCP formulation

$$Min\ C_{max} \quad (1)$$

Subject to:

$$\sum_{k=1}^{m} x_{ik} = 1 \quad \forall\, i \in N \quad (2)$$

$$s_i + \sum_{k=1}^{m} t_{ik}\, x_{ik} \le C_{max} \quad \forall\, i \in N \quad (3)$$

$$s_i + \sum_{k=1}^{m} t_{ik}\, x_{ik} + \sum_{k=1}^{m} \sum_{l=1}^{m} c_{ik,jl}\, x_{jl}\, x_{ik} \le s_j, \forall j \in N, \forall i \in Pred(j) \quad (4)$$

$$\begin{cases} s_i + t_{ik} - s_j \le B(1 - x_{ik}\, x_{jk}) \\ or \\ s_j + t_{jk} - s_i \le B(1 - x_{ik}\, x_{jk}) \end{cases} \quad \forall i, j \in N,\ \forall k \in M \quad (5)$$

$$x_{ik} = 0 \quad \forall i \in F_k, \forall k \in M \quad (6)$$

$$x_{ik} \in \{0, 1\};\ s_i \in R^{+} \quad \forall i \in N, \forall k \in M \quad (7)$$

MIQCP: Mixed-Integer Quadratically-Constrained Program.

3. Mathematical Models

Linearization of communication constraints

(4) $s_i + \sum_{k=1}^{m} t_{ik} x_{ik} + \sum_{k=1}^{m} \sum_{l=1}^{m} c_{ik,jl} x_{jl}\, x_{ik} \leq s_j, \forall j \in N, \forall i \in Pred(j)$

(4) $\Leftrightarrow s_i + t_{ik} x_{ik} + c_{ik,jl} x_{jl} x_{ik} \leq s_j \quad \forall k, l \in M$

$$\Leftrightarrow \begin{cases} s_i + t_{ik^*} + c_{ik^*,jl^*} \leq s_j, \\ s_i + t_{ik*} \leq s_j, \\ s_i \leq s_j. \end{cases}$$

$$\Leftrightarrow s_i + t_{ik*} + c_{ik*,jl*} \leq s_j$$

(4-a) $s_i + t_{ik} x_{ik} + c_{ik,jl}\left(x_{jl} + x_{ik} - 1\right) \leq s_j \quad \forall k, l \in M; \forall j \in N, \forall i \in Pred(j)$

(4-a) $\Leftrightarrow \begin{cases} s_i + t_{ik*} + c_{ik*,jl*} \leq s_j, \\ s_i + t_{ik*} \leq s_j, \\ s_i \leq s_j, \\ s_i - c_{ik,jl} \leq s_j. \end{cases}$

$$\Leftrightarrow s_i + t_{ik*} + c_{ik*,jl*} \leq s_j$$

3. Mathematical Models

MILP formulation

$Min\ C_{max}$ Subject to:		(1)
$\sum_{k=1}^{m} x_{ik} = 1$	$\forall\, i \in N$	(2)
$s_i + \sum_{k=1}^{m} t_{ik} x_{ik} \leq C_{max}$	$\forall\, i \in N$	(3)
$s_i + t_{ik} x_{ik} + c_{ik,jl}\left(x_{jl} + x_{ik} - 1\right) \leq s_j$	$\forall k,l \in M, \forall j \in N, \forall i \in Pred(j)$	(4-a)
$s_i + t_{ik} - s_j \leq B\left(3 - x_{ik} - x_{jk} - \delta_{ij}\right)$	$\forall i,j \in N,\ \forall k \in M$	(5-a)
$s_j + t_{jk} - s_i \leq B\left(2 - x_{ik} - x_{jk} + \delta_{ij}\right)$	$\forall i,j \in N,\ \forall k \in M$	(5-b)
$x_{ik} = 0\ \forall i \in F_k, \forall k \in M$		(6)
$x_{ik}, \delta_{ij} \in \{0,1\};\ s_i \in R^+$	$\forall i,j \in N, \forall k \in M$	(7)

MILP: Mixed-Integer Linear Program.

3. Mathematical Models

Reduced MILP formulation

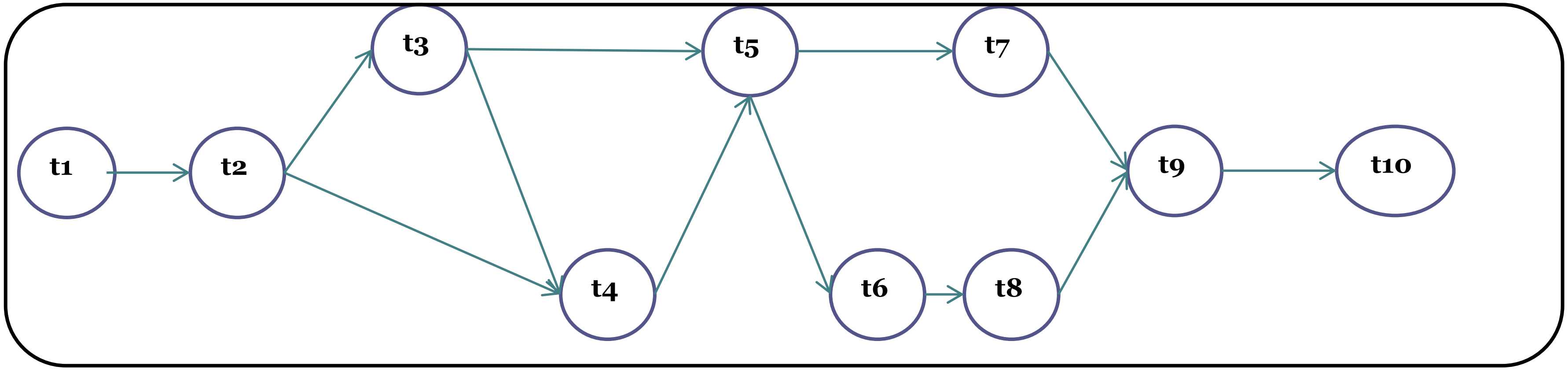
$Min\ C_{max}$		(1)
Subject to:		
$\sum_{k=1}^{m} x_{ik} = 1$	$\forall\, i \in N \backslash F_k$	(2-a)
$s_i + \sum_{k=1}^{m} t_{ik} x_{ik} \leq C_{max}$	$\forall\, i \in FT$	(3-a)
$s_i + t_{ik} x_{ik} + c_{ik,jl}\left(x_{jl} + x_{ik} - 1\right) \leq s_j$	$\forall k, l \in M, \forall j \in N \backslash F_l, \forall i \in Pred(j) \backslash F_k$	(4-b)
$s_i + t_{ik} - s_j \leq B\left(3 - x_{ik} - x_{jk} - \delta_{ij}\right)$	$\forall k \in M, \forall i \in N \backslash F_k, \forall j \in N \backslash P(i)$	(5-c)
$s_j + t_{jk} - s_i \leq B\left(2 - x_{ik} - x_{jk} + \delta_{ij}\right)$	$\forall k \in M, \forall i \in N \backslash F_k, \forall j \in N \backslash P(i)$	(5-d)
$x_{ik}, \delta_{ij} \in \{0,1\};\ s_i \in R^{+}$	$\forall k \in M, \forall i, j \in N \backslash F_k$	(7)

P(i) is the set of tasks that can be reached from i using a path in the graph G or the inverse graph G^{-1}.

P(i) & Disjunctive constraints

Example of graph G

Density	26.67%
Average degree	2.4
Minimum degree	0
Maximum degree	3



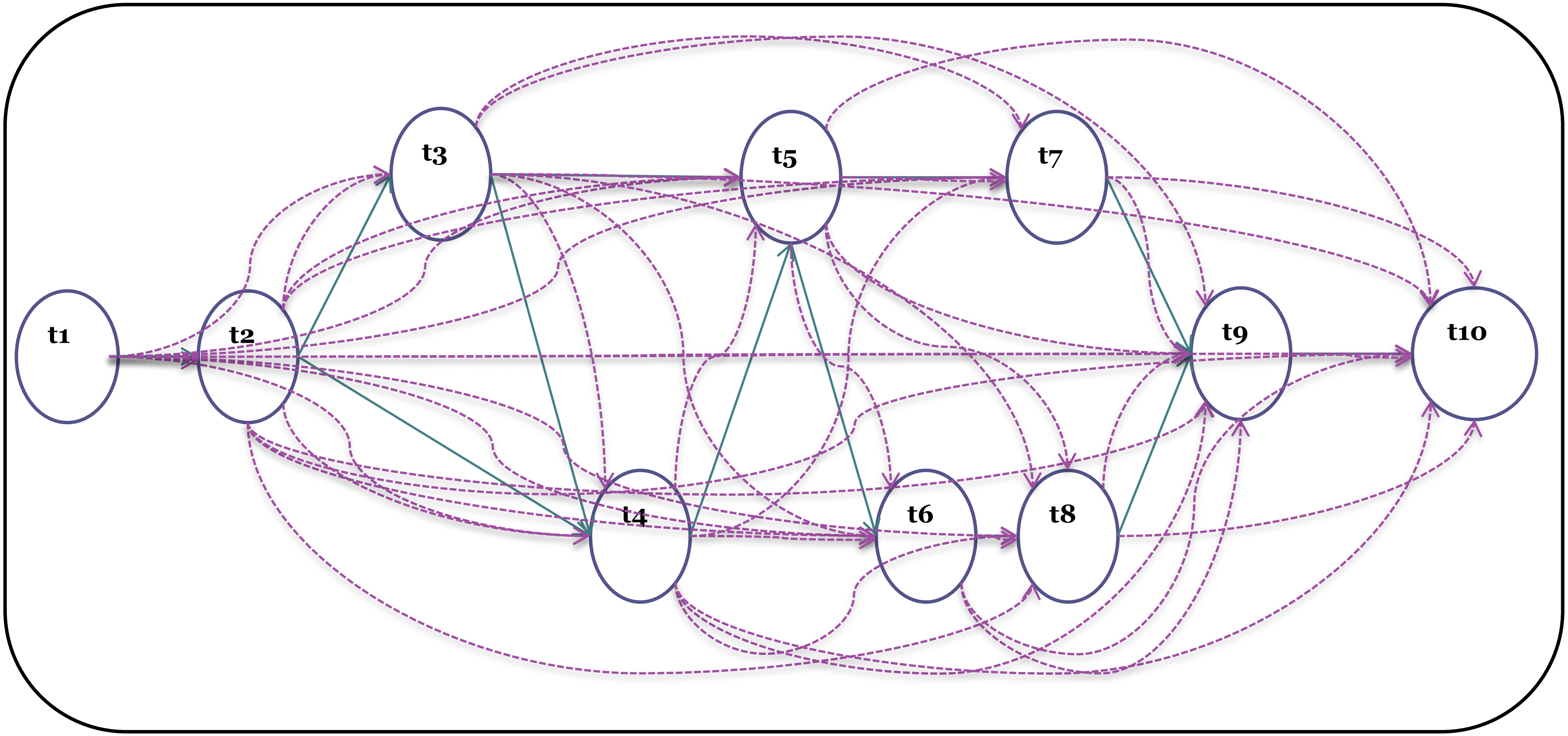
For 3 CUs:

Nb disjunctive constraints 600

Nb binary variables due to disjunctive constraints 100

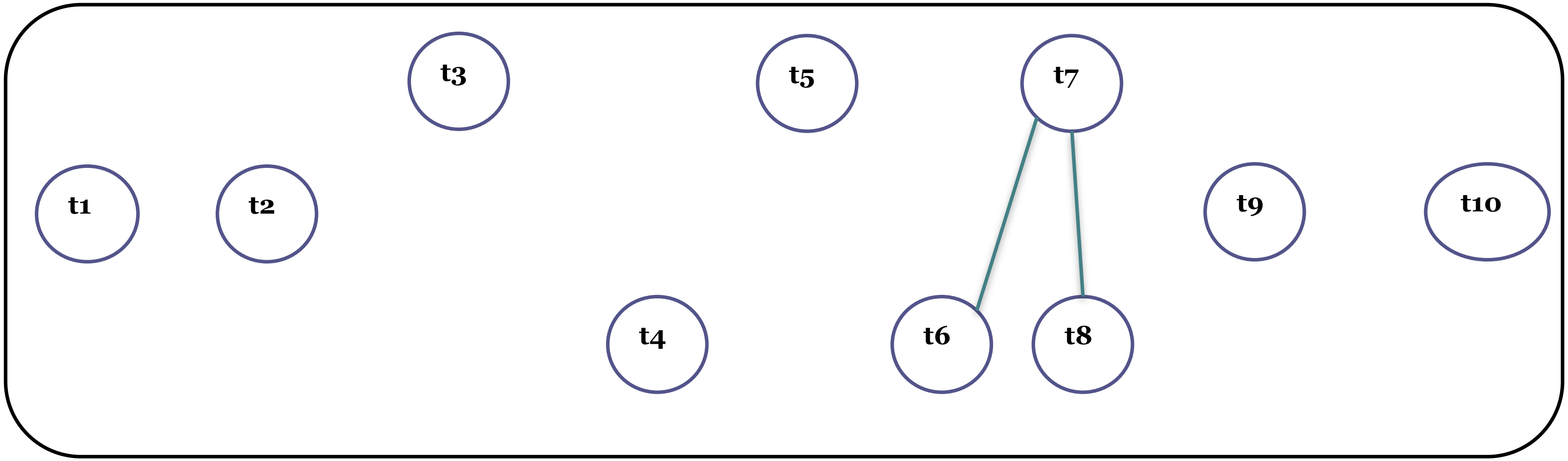
P(i) & Disjunctive constraints

The associated n power graph G^n



P(i) & Disjunctive constraints

The associated graph H



For 3 CUs:

Nb disjunctive constraints 12 (vs. 600)
Nb binary variables due to disjunctive constraints 2 (vs. 100)

P(i) & Disjunctive constraints

$$|A_H| = \sum_{\substack{1 \le i < n \\ i < j \le n}} \left[\neg \left(\bigvee_{i=1}^{n} A^i \right) \right] (i,j) = \sum_{\substack{1 \le i < n \\ i < j \le n}} [\neg((A+I)^n)](i,j)$$

In script m :

```
>> An=I; for i=1:n,  An=logical(An+An*A);end,
>> sum(sum(logical(An==0)))-(n*(n-1)/2)

>> [i, j] = find(An==0);
>> [i(find(j>i)),  j(find(j>i))]
```

gives the number of disjunctive constraints

gives the indices of concerned tasks

3. Mathematical Models

MILP formulation

Number of binary variables $= nm + n^2,$
Number of continuous variables $= n,$
Number of constraints $= 2n + |A|(m^2) + 2mn^2$

Reduced MILP formulation

Number of binary variables $= nm + 2|A_H|,$
Number of continuous variables $= n,$
Number of constraints $= n + |N^+| + m^2|A| + 2m|A_H|$

Where:

- $|A_H|$ *=cardinality of the set of edges in the complement graph of the undirected graph associated with the n-th power G^n of graph G.*
- $|N^+|$ *=cardinality on the set of tasks with no successors*

5. Performance and Test Results

Illustrative data sets

Data set	m	n	|A(G)|
Dataset 1	4	5	4
Dataset 2	4	20	29
Dataset 3	4	20	22
Dataset 4	4	20	24
Dataset 5	5	49	67
Dataset 6	5	49	67

5. Performance and Test Results

Key Results on Illustrative data sets

Data set	MILP Model			MIQCP Model		
	C_{max}	Time (sec)	Gap	C_{max}	Time (sec)	Gap
Dataset 1	35	**0,265**	0,00%	35	0,561	0,00%
Dataset 2	97,25	**0,998**	0,00%	97,25	600,401	2,59%
Dataset 3	76,00	**5,819**	0,00%	91,56	619,137	24,12%
Dataset 4	49,00	**5,897**	0,00%	64,07	603,069	33,57%

The results for comparing the non-linear and the linear models (time limit for the solver: 600 sec).

5. Performance and Test Results

Key Results on Illustrative data sets

Data Set	Linear Model					Reduced Model				
	Cmax	Time (sec)	Gap	nb cols	nb rows	Cmax	Time (sec)	Gap	nb cols	nb rows
Dataset 1	35	**0,28**	0,00%	47	234	35	**0,23**	0,00%	32	111
Dataset 2	97,25	**1,00**	0,00%	482	3544	97,25	**0,53**	0,00%	239	1589
Dataset 3	76	**5,82**	0,00%	482	3432	76	**2,32**	0,00%	248	1544
Dataset 4	49	**5,90**	0,00%	482	3464	49	**2,45**	0,00%	232	1446
Dataset 5	152	3000,55	25,62%	2648	25293	145	3000,41	**15,63 %**	1138	10145
Dataset 6	191,5	3000,15	0,78%	2648	25346	190	**30,61**	**0,00%**	1051	6299

The results for comparing the linear and the reduced models (time limit for the solver: 3000 sec).

5. Performance and Test Results

Key Results on Benchmark sets from Davidović et al.

Mod File	Time (sec)	Deterministic Time	Improvement comparing to M3 (%)
M3: reduced MILP	3,2210	1807,96	0,00%
M3 + Cuts	1,6428	1011,65	49,00%
M3 + Cuts + Bounds	1,8360	1166,77	43,00%

The results for the effects of cuts and bounds (Average cpu time for 584 instances).

Davidović, T., Crainic, T.G., Benchmark-problem instances for static scheduling of task graphs with communication delays on homogeneous multiprocessor systems. Comput. Oper. Res. 33, 2155–2177 (2006)

Davidović, T.; Liberti, L.; Maculan, N. & Mladenović, N. Towards the optimal solution of the multiprocessor scheduling problem with communication delays MISTA Proceedings, 2007

5. Performance and Test Results

Key Results on Benchmark sets from Davidović et al.

		Average results for the 17 common solved instances		
Model	Nb solved	CPU time (sec)	Nb Var	Nb Constraints
M6	19	3,5036	602	8965
M5	22	6,9955	6783	8925
M4	0	---	---	---
M3	34	0,6725	107	2019
M3 + Cuts	**35**	**0,4027**	**107**	**2035**

Average solving time for 45 instances with n=10 to 50 and m =2 to 8 (time limit for the solver: 120 sec).

M4: classical formulation (Davidović et al.)

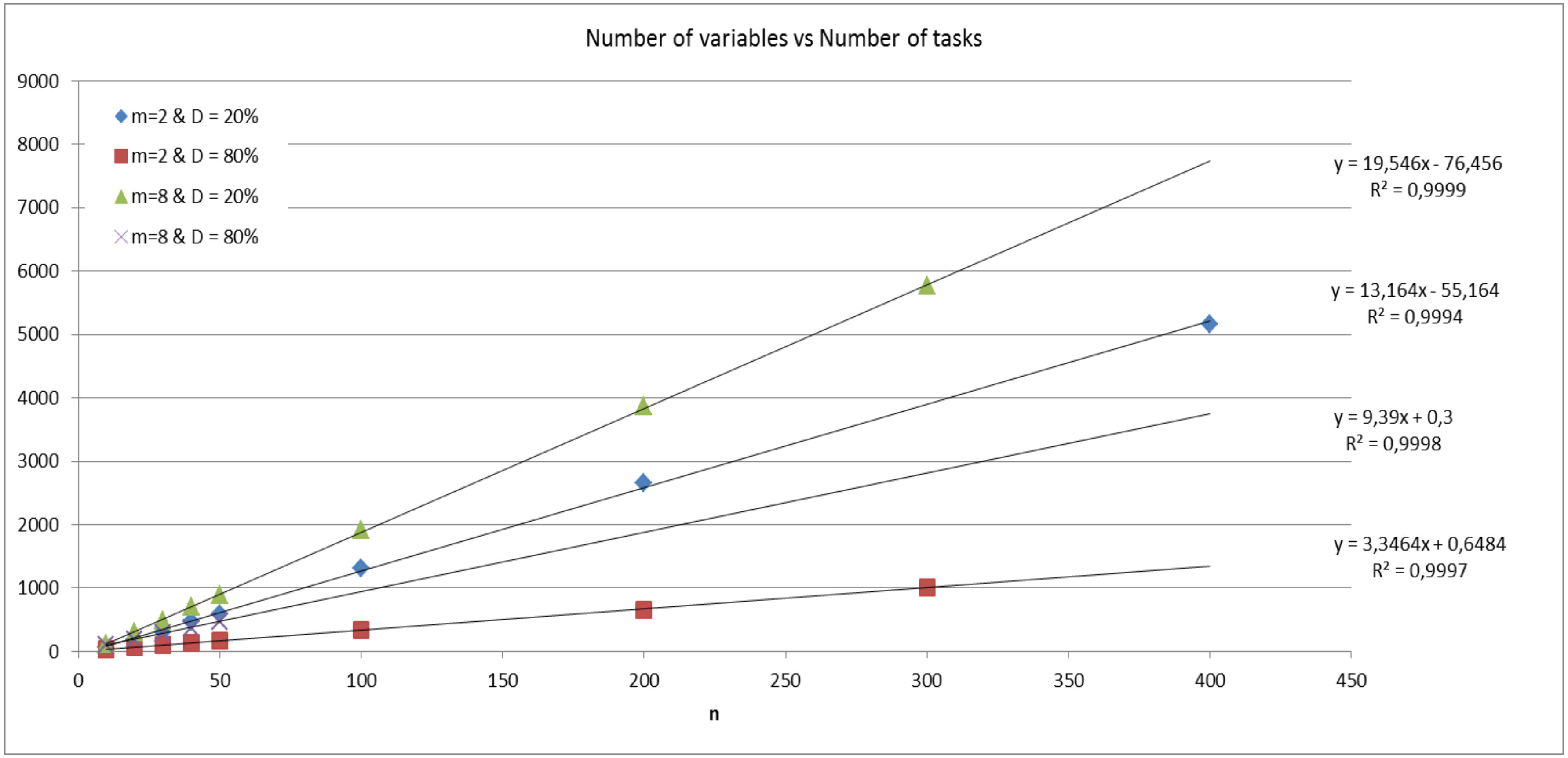
M5: packing formulation (Davidović et al.)

M6: ILP-Transitivity-Clause model (Venugopalan et al.)

Venugopalan, S. & Sinnen, O. Xiang, Y.; Stojmenovic, I.; Apduhan, B.; Wang, G.; Nakano, K. and Zomaya, A. (Eds.) Optimal Linear Programming Solutions for Multiprocessor Scheduling with Communication Delays Algorithms and Architectures for Parallel Processing, Springer Berlin Heidelberg, 2012, 7439, 129-138

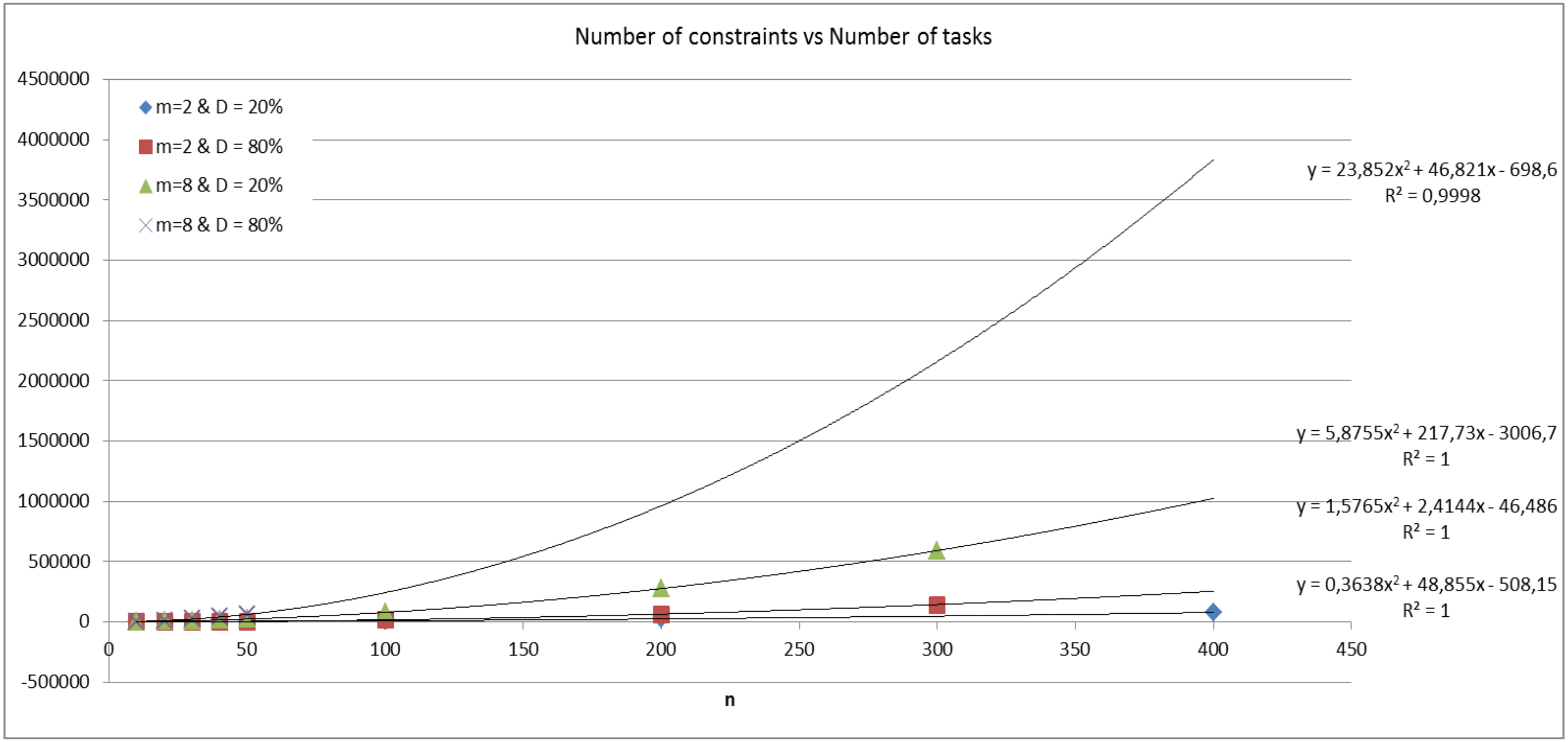
5. Performance and Test Results

Key Results on Benchmark sets from Davidović et al.



5. Performance and Test Results

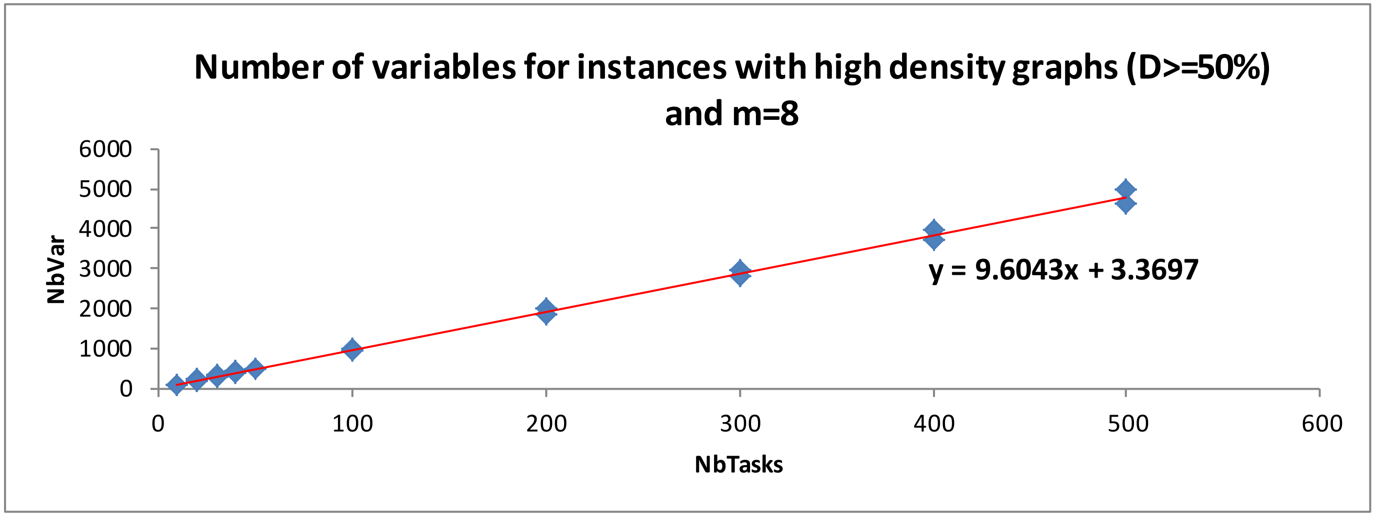
Key Results on Benchmark sets from Davidović et al.



5. Performance and Test Results

Regression between number of variables and tasks

Regression Statistics	
Multiple R	0.999236084
R Square	0.998472752
Adjusted R Square	0.998465994
Standard Error	45.43191537
Observations	228



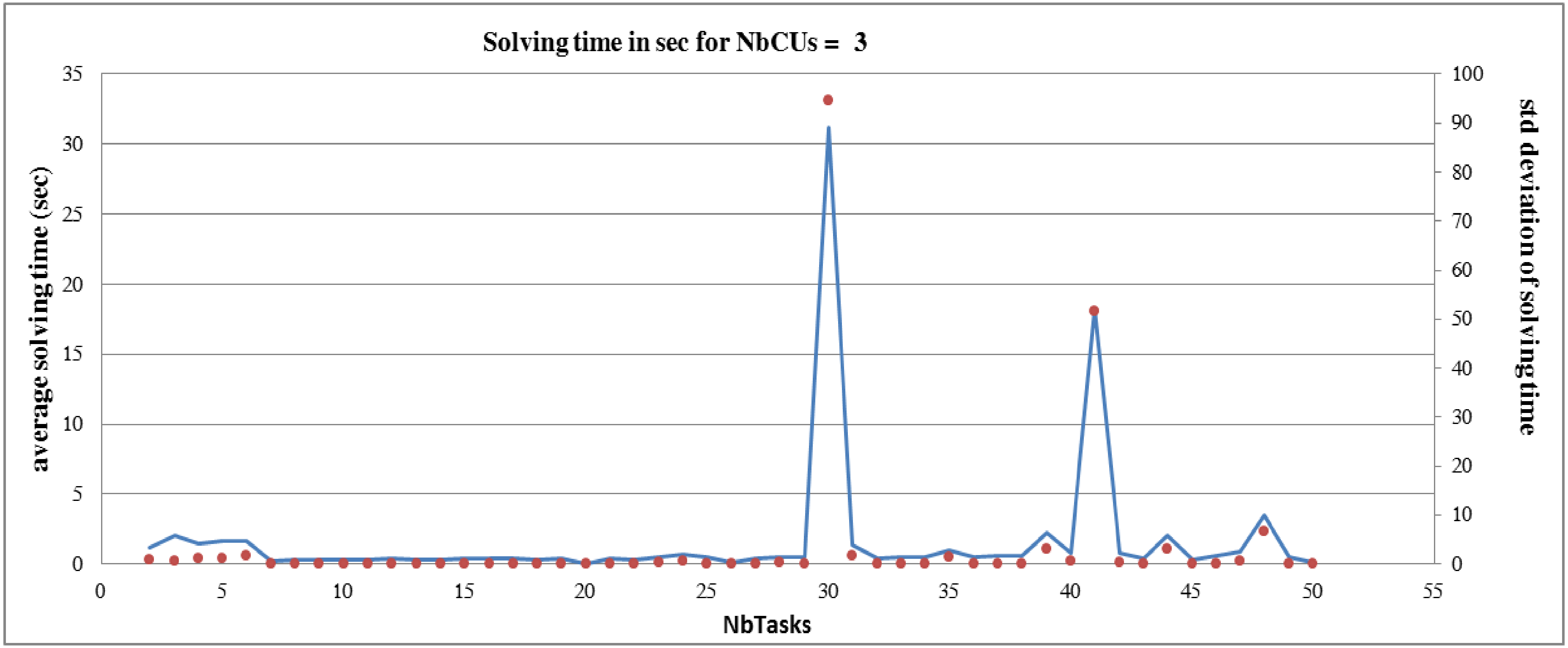
ANOVA

	df	*SS*	*MS*	*F*	*P-value*
Regression	1	304970070.9	304970070.9	147752.5985	0
Residual	226	466477.3191	2064.058934		
Total	227	305436548.2			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	3.370	3.530	0.955	0.341	-3.586	10.325
NbTasks	9.604	0.025	384.386	0.000	9.555	9.653

5. Performance and Test Results

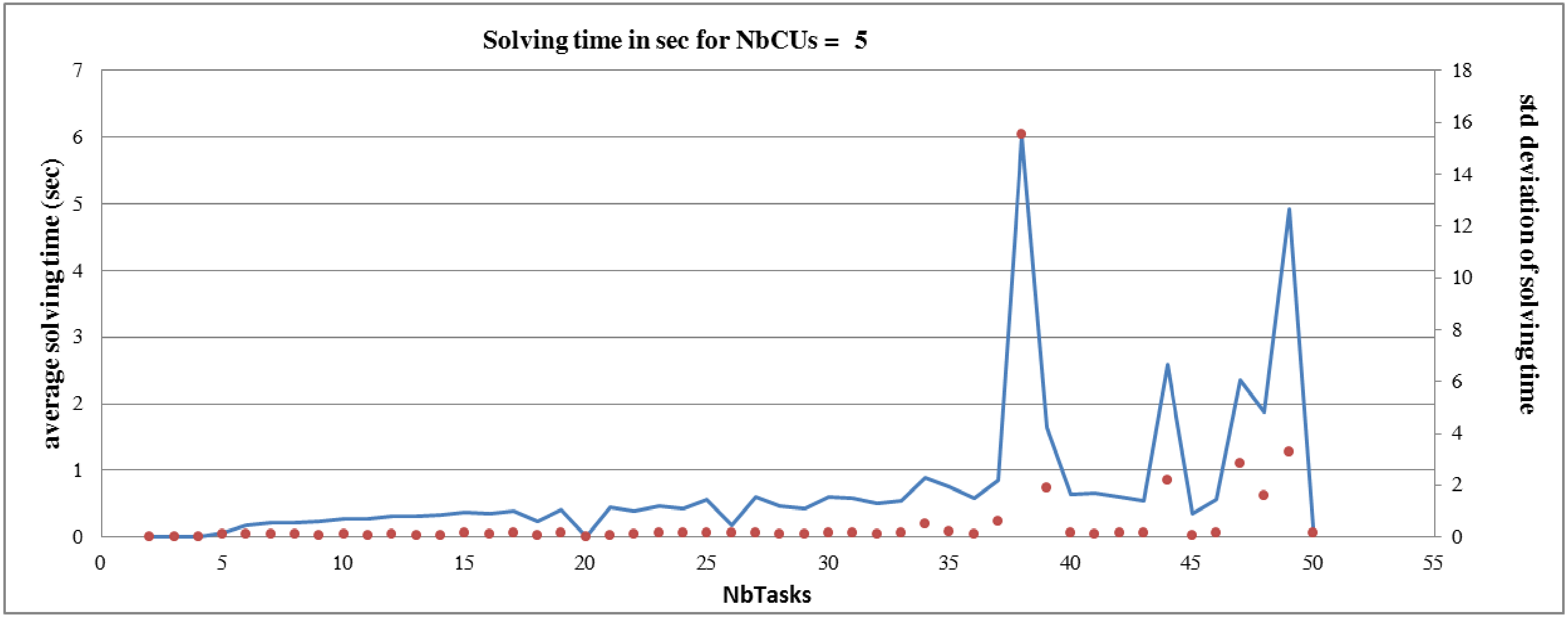
Other Results



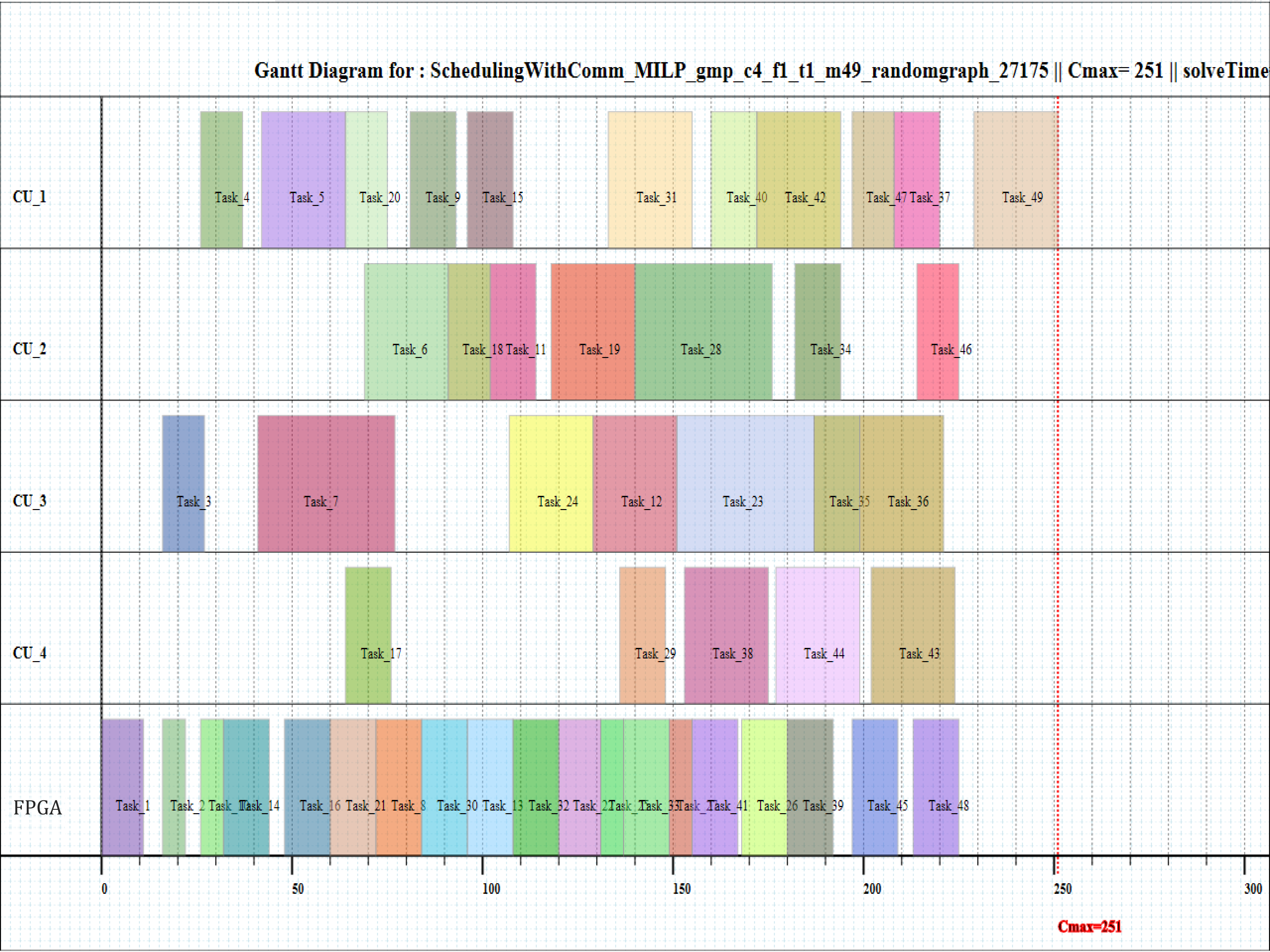
Solving time for instances with 3 computing units and number of tasks from 2 to 50.

5. Performance and Test Results

Other Results



Solving time for instances with 5 computing units and number of tasks from 2 to 50.


Gantt Diagram for : SchedulingWithComm_MILP_gmp_c4_f1_t1_m49_randomgraph_27175 || Cmax= 251 || solveTime
CU_1
Task_4
Task_5
Task_20
Task_9
Task_15
Task_31
Task_40
Task_42
Task_47
Task_37
Task_49
CU_2
Task_6
Task_18
Task_11
Task_19
Task_28
Task_34
Task_46
CU_3
Task_3
Task_7
Task_24
Task_12
Task_23
Task_35
Task_36
CU_4
Task_17
Task_29
Task_38
Task_44
Task_43
FPGA
Task_1
Task_2
Task_14
Task_16
Task_21
Task_8
Task_30
Task_13
Task_32
Task_41
Task_26
Task_39
Task_45
Task_48
0
50
100
150
200
250
300
Cmax=251

6. Conclusions

- We were able to take into account the general case of the communication delay in an heterogeneous environment;
- The MIQCP is convex but still too hard to solve in a reasonable amount of time;
- The linearization is possible and beneficial, the solving time drastically reduced;
- Linearization of the communications constraints is done without any additional variables;

6. Conclusions

- Constraints and variables pruning, by exploiting the precedence graph, cuts the solving time almost by half.
- The proposed model is promising and could handle average problem with size up to 50 tasks and 8 CPU/FPGA units in few seconds.
- In our case, we use this model to minimize the C_{max}, but it could be easily adapted to other objectives.

7. Questions

5. Performance and Test Results

Parameters used for the testing

- The entire testing is conducted on a laptop with 8 GB of RAM and an Intel processor i7-3740QM with 8 cores. The operating system is a 64-bit Windows 7 professional.
- For solving the mathematical models, we use Cplex 12.5x
- For coding we use OPL scripting language.

Initialization:

Let $I = \{i \in N : Pred(i) = \emptyset\}$;
Set $U = -\infty$;
for $i \in N dos_i = -1$; // negative to tag unprocessed task
for $j \in M doa(j) = 0$;

Main loop:

while ($I \neq \emptyset$) do {
Let x(i,j) = max{a(j), max{s_j+t_j+ $c_{jH(j),ij}$: $j \in Pred(i)$}}
Set x(i*,j*) = min{x(i,j) : $i \in I, j \in M$}
H(i*) = j*;
s_{i*} = x(i*,j*) - $t_{i*,j*}$;
a(j*) = x(i*,j*);
I = I - {i*};
U = max{U, s_{i*}+ $t_{i*,j*}$};
I = I + {r $\in Succ(i*) : s_j < 0 forall j \in Pred(r)$};
}

Algorithm 1: The greedy constructive heuristic (GCH).

Initialization:

Let I = $\{i \in N : Pred(i) = \emptyset \}$;

Set L = $+\infty$;

for i N do e_i = -1; // negative to tag unprocessed task;

Main loop:

while $I \neq \emptyset$ do {

Let $c^-(i,j) = \min\{c_{ikjh} : k, h \in M\}$

$e_i = \max\{0, \max\{e_j + t_j + c^-(i,j) : j \in Pred(i)\}\}$

$L = \max(L, e_i + t_i)$;

$I = \{r \in Succ(i) : e_j < 0 forall j \in Pred(r)\}$;

}

Algorithm 2: The ForwardPass for computing the earliest start time.

Initialization:
Let I = $\{i \in N : Succ(i) = \emptyset\}$;
Set U = C_{max} of the greedy constructive heuristic;
for i N do l_i = -1; // negative to tag unprocessed task;
Main loop:
while $I \neq \emptyset$ do {
    Let $c^-(i,j) = \min\{c_{ikjh} : k, h \in M\}$
    Set $U = \max\{U, \min\{l_j - c^-(i,j) : j \in Succ(i)\}\}$;
    $l_i = U - t_i$;
    $I = \{r \in Pred(i) : l_j < 0 forall j \in Succ(r)\}$;
}
Algorithm 3: The BackwardPass for computing the tardiest start time.