Mapping filter services on heterogeneous platforms

To appear in IPDPS 2009

Anne Benoit,Fanny Dufossé,Yves Robert

January 9, 2009

Introduction

The problem:

- treatment of a data flow
- filter services with selectivity σ and cost c
- precedence constraints between services
- servers with speed s
- one-to-one mappings

The objective:

- minimize the period
- minimize the latency

Motivation

For services of selectivity less than one

- grep
- web services
- Select-Project-Join query optimization
- ...

Related problems:

- component testing
- unsupervised systems

The instances

The problems depend on:

- the criteria: MinPeriod, MinLatency or BiCriteria
- the platform: Hom or Het
- the dependence constraints: NoPrec or Prec

The instances

The problems depend on:

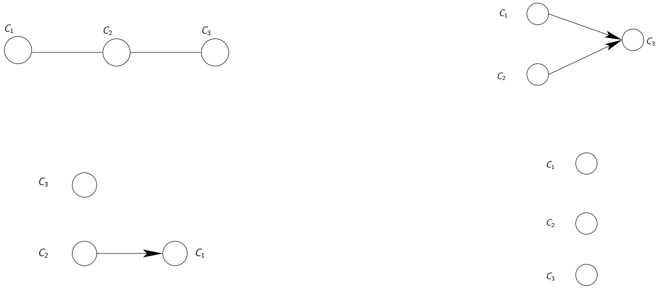
- the criteria: MinPeriod, MinLatency or BiCriteria
- the platform: Hom or Het
- the dependence constraints: NoPrec or Prec

The instances: $\mathcal{A} = (\mathcal{F}, \mathcal{G}, \mathcal{S})$ with:

- The services: $\mathcal{F} = \{C_1, C_2, \ldots, C_n\}$
- The precedence constraints: $\mathcal{G} \subset \mathcal{F} \times \mathcal{F}$
- The servers: $\mathcal{S} = \{S_1, S_2, \ldots, S_p\}$

The problem

Example for 3 independent services:
The plan?



The mapping?

$$(C_1, S_2), (C_2, S_1), (C_3, S_3)$$

$$(C_1, S_3), (C_2, S_2), (C_3, S_1)$$

Example

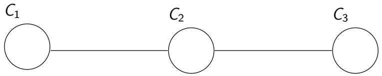


Figure: Chaining services.

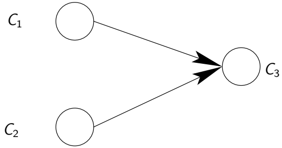


Figure: Combining selectivities

$$\mathcal{P} = \max\left(\frac{c_1}{s_1}, \frac{\sigma_1 c_2}{s_2}, \frac{\sigma_1\sigma_2 c_3}{s_3}\right) \qquad \mathcal{P} = \max\left(\frac{c_1}{s_1}, \frac{c_2}{s_2}, \frac{\sigma_1\sigma_2 c_3}{s_3}\right)$$

$$\mathcal{L} = \frac{c_1}{s_1} + \frac{\sigma_1 c_2}{s_2} + \frac{\sigma_1\sigma_2 c_3}{s_3} \qquad \mathcal{L} = \max\left(\frac{c_1}{s_1}, \frac{c_2}{s_2}\right) + \frac{\sigma_1\sigma_2 c_3}{s_3}$$

Example

- $c_1 = 1,\ c_2 = 4,\ c_3 = 10$
- $\sigma_1 = \frac{1}{2},\ \sigma_2 = \sigma_3 = \frac{1}{3}$
- $s_1 = 1,\ s_2 = 2$ and $s_3 = 3$

Example

- $c_1 = 1,\ c_2 = 4,\ c_3 = 10$
- $\sigma_1 = \frac{1}{2},\ \sigma_2 = \sigma_3 = \frac{1}{3}$
- $s_1 = 1,\ s_2 = 2$ and $s_3 = 3$

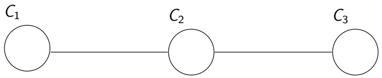


Figure: Optimal plan for period.

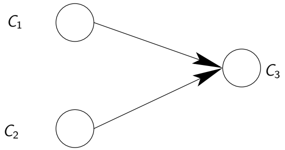


Figure: Optimal plan for latency

$$P = 1$$
$$L = \frac{5}{2}$$

$$L = \frac{13}{6}$$
$$P = \frac{4}{3}$$

General structure of optimal solutions

The instance : $C_1, ..., C_n, S_1, ..., S_n$ with

- $\sigma_1, ..., \sigma_p \leq 1$
- $\sigma_{p+1}, ..., \sigma_n \geq 1$

General structure of optimal solutions

The instance : $C_1, ..., C_n, S_1, ..., S_n$ with

- $\sigma_1, ..., \sigma_p \leq 1$
- $\sigma_{p+1}, ..., \sigma_n \geq 1$

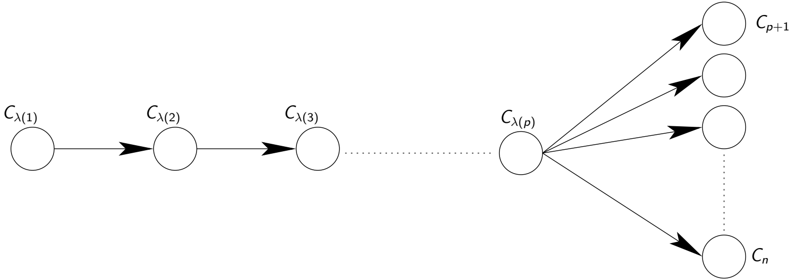


Figure: General structure

Homogeneous case without precedence constraints

The instance : $C_1, ..., C_n$ with

- $c_1 \leq c_2 \leq ... \leq c_p$
- $\sigma_1, ..., \sigma_p < 1$
- $\sigma_{p+1}, ..., \sigma_n \geq 1$

The matching: $C_1 \rightarrow C_2 \rightarrow ... \rightarrow C_p$

Homogeneous case with precedence constraints

Computing the optimal subgraph for C in the graph G:
Let $D = \max_i \{\log \sigma_i\}$.
We construct a network flow graph W with:

- a source s
- a node f_i by service in G
- a sink node t
- an edge $s -> f_i$ with capacity $+\infty$ if C_i is ancestor of C in $\mathcal{G}$, D else
- an edge $f_i -> f_j$ of capacity $+\infty$ if C_j is an ancestor of C_i in G
- an edge $f_i -> t$ with capacity $D + \log \sigma_i$

The set of services on the side of s in a min-cut is the optimal subset of predecessors for latency.

Homogeneous case with precedence constraints

Computing the optimal subgraph for C in the graph G:
Let $D = \max_i \{\log \sigma_i\}$.
We construct a network flow graph W with:

- a source s
- a node f_i by service in G
- a sink node t
- an edge $s -> f_i$ with capacity $+\infty$ if C_i is ancestor of C in $\mathcal{G}$, D else
- an edge $f_i -> f_j$ of capacity $+\infty$ if C_j is an ancestor of C_i in G
- an edge $f_i -> t$ with capacity $D + \log \sigma_i$

The set of services on the side of s in a min-cut is the optimal subset of predecessors for latency.
Optimal algorithm: at each step place the available service with minimal possible period.

Proof: NP-completeness of MinPeriod-Het

Problem (RN3DM)

Given an integer vector $A = (A[1], \dots, A[n])$ *of size n, does there exist two permutations* λ_1 *and* λ_2 *of* $\{1, 2, \dots, n\}$ *such that*

$$\forall 1 \leq i \leq n, \quad \lambda_1(i) + \lambda_2(i) = A[i]$$

Proof: NP-completeness of MinPeriod-Het

Problem (RN3DM)

Given an integer vector $A = (A[1], \ldots, A[n])$ of size n, does there exist two permutations λ_1 and λ_2 of $\{1, 2, \ldots, n\}$ such that

$$\forall 1 \leq i \leq n, \quad \lambda_1(i) + \lambda_2(i) = A[i]$$

The associated instance :

- $c_i = 2^{A[i]}$
- $\sigma_i = 1/2$
- $s_i = 2^i$
- $P = 2$

Proof: NP-completeness of MinPeriod-Het

Problem (RN3DM)

Given an integer vector $A = (A[1], \dots, A[n])$ of size n, does there exist two permutations λ_1 and λ_2 of $\{1, 2, \dots, n\}$ such that

$$\forall 1 \leq i \leq n, \quad \lambda_1(i) + \lambda_2(i) = A[i]$$

The associated instance :

- $c_i = 2^{A[i]}$
- $\sigma_i = 1/2$
- $s_i = 2^i$
- $P = 2$

$$\begin{aligned} & \forall 1 \leq i \leq n, \quad \lambda_1(i) + \lambda_2(i) \geq A[i] \\ \iff & \forall 1 \leq i \leq n, \quad \left(\frac{1}{2}\right)^{\lambda_1(i)-1} \times \frac{2^{A[i]}}{2^{\lambda_2(i)}} \leq 2 \end{aligned}$$

Inapproximability of MinPeriod-Het

Proposition

For any $K > 0$, there exists no K-approximation algorithm for MinPeriod-NoPrec-Het, *unless P=NP.*

Inapproximability of MinPeriod-Het

Proposition

For any $K > 0$, there exists no K-approximation algorithm for MinPeriod-NoPrec-Het, *unless P=NP.*

Reduction from RN3DM:

- $c_i = K^{A[i]-1}$
- $\sigma_i = 1/K$
- $s_i = K^i$
- $P = 1$

Integer linear program

The variables:

- $t_{i,u} = 1$ if service C_i is assigned to server S_u
- $s_{i,j} = 1$ if service C_i is an ancestor of C_j
- M is the logarithm of the optimal period

Integer linear program

The variables:

- $t_{i,u} = 1$ if service C_i is assigned to server S_u
- $s_{i,j} = 1$ if service C_i is an ancestor of C_j
- M is the logarithm of the optimal period

The constraints:

- $\forall i, \quad \sum_u t_{i,u} = 1$
- $\forall u, \quad \sum_i t_{i,u} = 1$
- $\forall i, j, k, \quad s_{i,j} + s_{j,k} - 1 \leq s_{i,k}$
- $\forall i, s_{i,i} = 0$
- $\forall i, \quad \log c_i - \sum_u t_{i,u} \log s_u + \sum_k s_{k,i} \log \sigma_k \leq M$

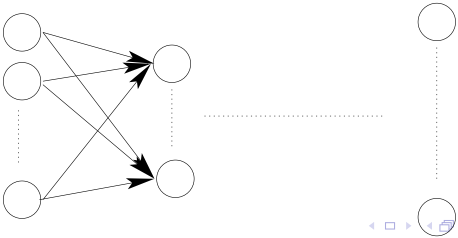
The objective function: Minimize M

Structure of the optimal plan

Proposition

Let $C_1, ..., C_n, S_1, ..., S_n$ be an instance of MinLatency. *Then, the optimal latency is obtained with a plan G such that, for any $v_1 = (C_{i_1}, S_{u_1})$, $v_2 = (C_{i_2}, S_{u_2})$,*

1. *If $d_{i_1}(G) = d_{i_2}(G)$, they have the same predecessors and the same successors in G.*
2. *If $d_{i_1}(G) > d_{i_2}(G)$ and $\sigma_{i_2} \leq 1$, then $c_{i_1}/s_{u_1} < c_{i_2}/s_{u_2}$.*
3. *All nodes with a service of selectivity $\sigma_i > 1$ are leaves ($d_i(G) = 0$).*



Algorithm without dependence constraint

Data: n services of cost $c_1 \leq \cdots \leq c_n$ and of selectivities $\sigma_1, ..., \sigma_n \leq 1$
Result: a plan G optimizing the latency
G is the graph reduced to node C_1;
for $i = 2$ **to** n **do**
    for $j = 0$ **to** $i - 1$ **do**
        Compute the completion time t_j of C_i in G with predecessors $C_1, ..., C_j$;
    end
    Choose j such that $t_j = \min_k\{t_k\}$;
    Add the node C_i and the edges $C_1 \rightarrow C_i, \ldots, C_j \rightarrow C_i$ to G;
end

Algorithm 1: Optimal algorithm for MinLatency-NoPrec-Hom.

G is the graph reduced to the node C of minimal cost with no predecessor in $\mathcal{G}$;

for $i = 2$ **to** n **do**

Let S be the set of services not yet in G and such that their set of predecessors in $\mathcal{G}$ is included in G;

for $C \in S$ **do**

for $C' \in G$ **do**

Compute the set S' minimizing the product of selectivities among services of latency less than $L_G(C')$, and including all predecessors of C in $\mathcal{G}$;

end

Let S_C be the set that minimizes the latency of C in G and L_C be this latency;

end

Choose a service C such that $L_C = \min\{L_{C'}, C' \in S\}$;

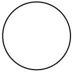
Add to G the node C, and $\forall C' \in S_C$, the edge $C' \rightarrow C$;

end

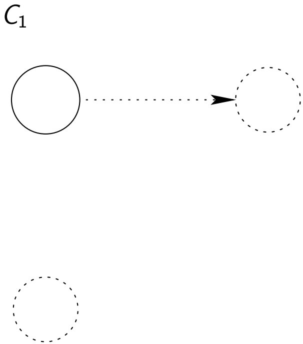
Algorithm 2: Optimal algorithm for MINLATENCY-PREC-HOM.

Example

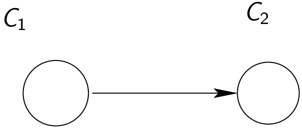
C_1



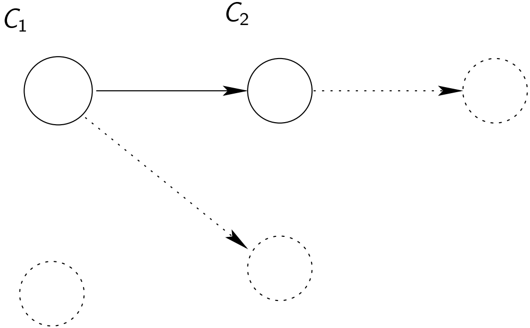
Example



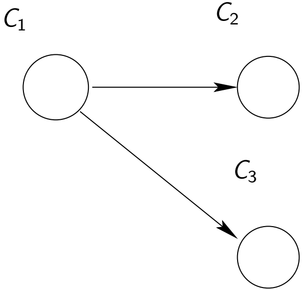
Example



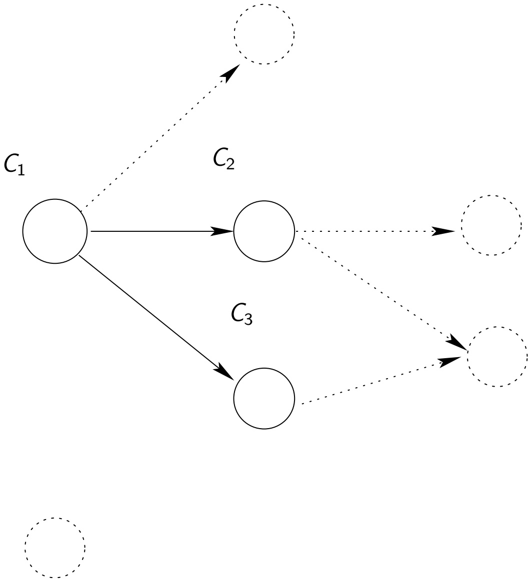
Example



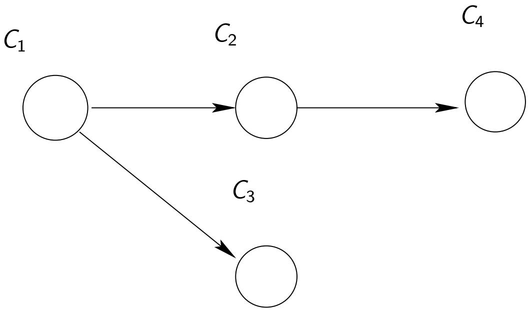
Example



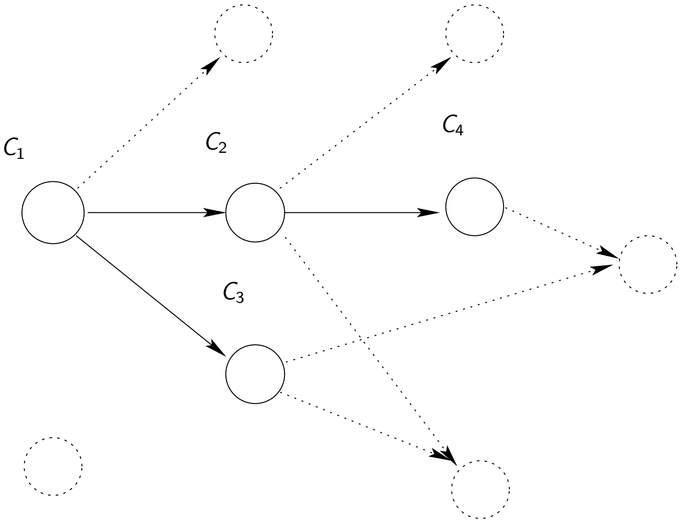
Example



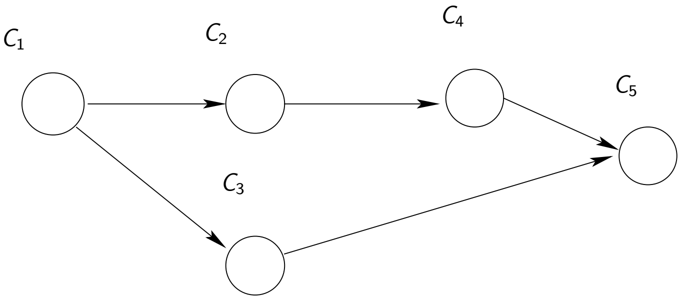
Example



Example



Example



Proof:NP-completeness of MinLatency-Het

Lemma

Let $C_1, \ldots, C_n, S_1, \ldots, S_n$ be an instance such that $\forall i$, c_i and s_i are integer power of 2 and $\sigma_i \leq \frac{1}{2}$. Then the optimal latency is obtained with a plan G such that

1. *Proposition 2 is verified;*
2. *for all nodes (C_{i_1}, S_{u_1}) and (C_{i_2}, S_{u_2}) with $d_{i_1}(G) = d_{i_2}(G)$, we have $\frac{c_{i_1}}{s_{u_1}} = \frac{c_{i_2}}{s_{u_2}}$.*

Proof:NP-completeness of MinLatency-Het

Lemma

Let $C_1, ..., C_n, S_1, ..., S_n$ be an instance such that $\forall i$, c_i and s_i are integer power of 2 and $\sigma_i \leq \frac{1}{2}$. Then the optimal latency is obtained with a plan G such that

1. *Proposition 2 is verified;*
2. *for all nodes (C_{i_1}, S_{u_1}) and (C_{i_2}, S_{u_2}) with $d_{i_1}(G) = d_{i_2}(G)$, we have $\frac{c_{i_1}}{s_{u_1}} = \frac{c_{i_2}}{s_{u_2}}$.*

- $c_i = 2^{A[i] \times n + (i-1)}$
- $\sigma_i = \left(\frac{1}{2}\right)^n$
- $s_i = 2^{n \times (i+1)}$
- $L = 2^n - 1$

Inapproximability of MinLatency-Het

Proposition

For any $K > 0$, there exists no K-approximation algorithm for MinLatency-NoPrec-Het, *unless P=NP.*

Inapproximability of MinLatency-Het

Proposition

For any $K > 0$, there exists no K-approximation algorithm for MinLatency-NoPrec-Het, *unless P=NP.*

Reduction from RN3DM

- $c_i = K^{A[i] \times n + (i-1)}$
- $\sigma_i = \left(\frac{1}{K}\right)^n$
- $s_i = K^{n \times (i+1)}$
- $L = \frac{K^n - 1}{K - 1}$

The variables:

- $z(i, u, e) = 1$ if the service C_i is associated to the server S_u and its set of predecessors is $e \subset \mathcal{C}$.
- $t(i)$ is the completion time of C_i
- M is the optimal latency

The variables:

- $z(i, u, e) = 1$ if the service C_i is associated to the server S_u and its set of predecessors is $e \subset \mathcal{C}$.
- $t(i)$ is the completion time of C_i
- M is the optimal latency

The constraints:

- $\forall u \in \mathcal{S}, \quad \sum_{i \in \mathcal{C}} \sum_{e \subset \mathcal{C}} z(i, u, e) = 1$
- $\forall i \in \mathcal{C}, \quad \sum_{u \in \mathcal{S}} \sum_{e \subset \mathcal{C}} z(i, u, e) = 1$

The variables:

- $z(i, u, e) = 1$ if the service C_i is associated to the server S_u and its set of predecessors is $e \subset \mathcal{C}$.
- $t(i)$ is the completion time of C_i
- M is the optimal latency

The constraints:

- $\forall u \in \mathcal{S}, \quad \sum_{i \in \mathcal{C}} \sum_{e \subset \mathcal{C}} z(i, u, e) = 1$
- $\forall i \in \mathcal{C}, \quad \sum_{u \in \mathcal{S}} \sum_{e \subset \mathcal{C}} z(i, u, e) = 1$
- $\forall i, i' \in \mathcal{C}, \forall u, u' \in \mathcal{S}, \forall e, e' \subset \mathcal{C}, e \not\subseteq e', i \in e', \quad z(i, u, e) + z(i', u', e') \leq 1$
- $\forall u \in \mathcal{S}, \forall e \subset \mathcal{C}, \forall i \in e, \ z(i, u, e) = 0$

The variables:

- $z(i,u,e) = 1$ if the service C_i is associated to the server S_u and its set of predecessors is $e \subset \mathcal{C}$.
- $t(i)$ is the completion time of C_i
- M is the optimal latency

The constraints:

- $\forall u \in \mathcal{S}, \quad \sum_{i\in\mathcal{C}} \sum_{e\subset\mathcal{C}} z(i,u,e) = 1$
- $\forall i \in \mathcal{C}, \quad \sum_{u\in\mathcal{S}} \sum_{e\subset\mathcal{C}} z(i,u,e) = 1$
- $\forall i, i' \in \mathcal{C}, \forall u, u' \in \mathcal{S}, \forall e, e' \subset \mathcal{C}, e \not\subseteq e', i \in e', \quad z(i,u,e) + z(i',u',e') \leq 1$
- $\forall u \in \mathcal{S}, \forall e \subset \mathcal{C}, \forall i \in e,\ z(i,u,e) = 0$
- $\forall i \in \mathcal{C}, \forall e \subset \mathcal{C}, \forall k \in e, \quad t(i) \geq \sum_{u\in\mathcal{S}} z(i,u,e) \left(\frac{c_i}{s_u} * \prod_{C_j\in e} \sigma_j + t(k) \right)$
- $\forall i \in \mathcal{C}, \quad t(i) \geq \sum_u z(i,u,e) \frac{c_i}{s_u} * \prod_{C_j\in e} \sigma_j$
- $\forall i \in \mathcal{C}, \quad t(i) \leq M$

The objective function: Minimize M

Data: n services of cost $c_1 \leq \cdots \leq c_n$ and of selectivities $\sigma_1, ..., \sigma_n \leq 1$ and a maximum throughput K

Result: a plan G optimizing the latency with a throughput less than K

G is the graph reduced to node C_1;

for $i = 2$ **to** n **do**

  for $j = 0$ **to** $i - 1$ **do**

    Compute the completion time t_j of C_i in G with predecessors $C_1, ..., C_j$;

  end

  Let $S = \{k | c_i \prod_{0 \leq k < i} \sigma_k \leq K\}$;

  Choose j such that $t_j = \min_{k \in S}\{t_k\}$;

  Add the node c_i and the edges $C_1 \rightarrow C_i, \ldots, C_j \rightarrow C_i$ to G;

end

Algorithm 3: Optimal algorithm for latency with a fixed throughput.

sigma-inc We place services on a chain in increasing order of σ.

short service/fast server We associate the service with shortest cost to the server with fastest speed.

long service/fast server We associate the service with largest cost with the server with fastest speed.

opt-homo We randomly associate services to servers.

greedy min This simple heuristic consists in running successively the four previous heuristics on the problem instance, and returning as a result the best of the four solutions.

random This last heuristic is fully random: we randomly associate services and servers, and we randomly place these pairs on a linear chain.

The instances:

- independent services
- the cost of services: $1 \leq c \leq 100$
- the selectivities: $0,01 \leq \sigma \leq 1$
- the speed of servers: $1 \leq s \leq 100$

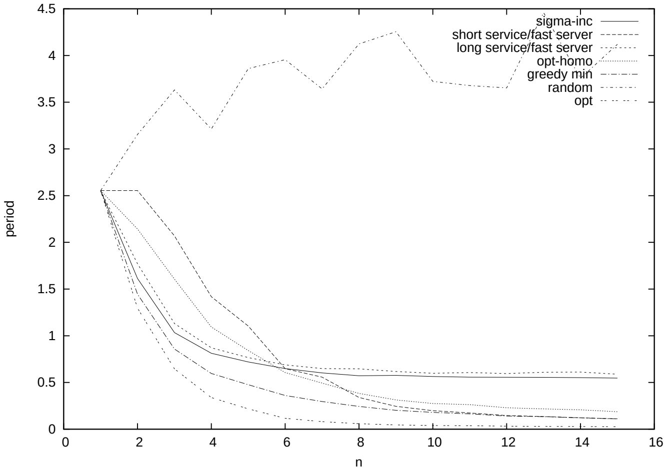


Figure: Experiment 1: general experiment.

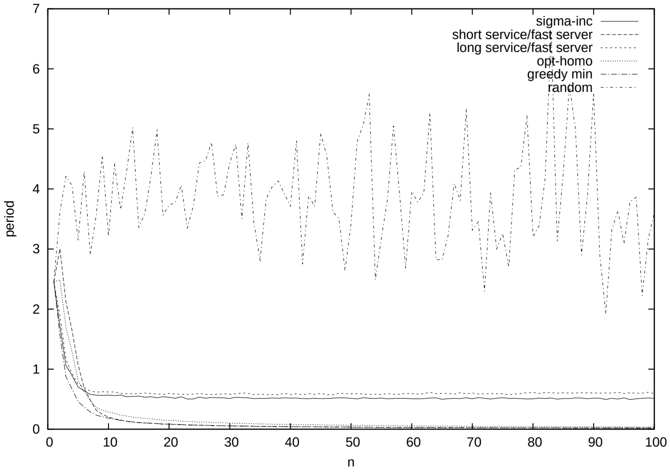


Figure: Experiment 1: general experiment.

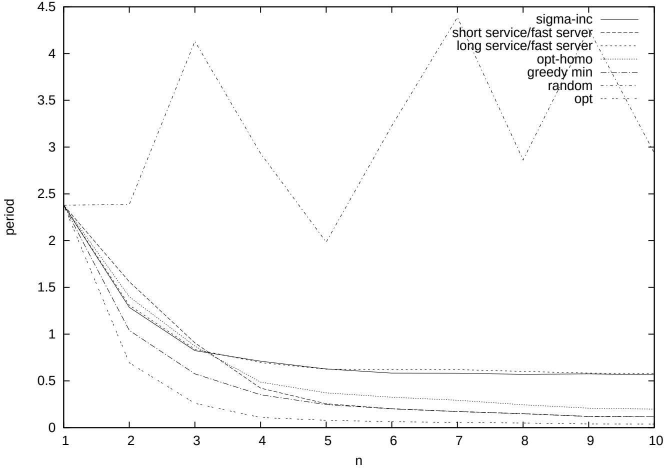


Figure: Experiment 2: with small selectivity.

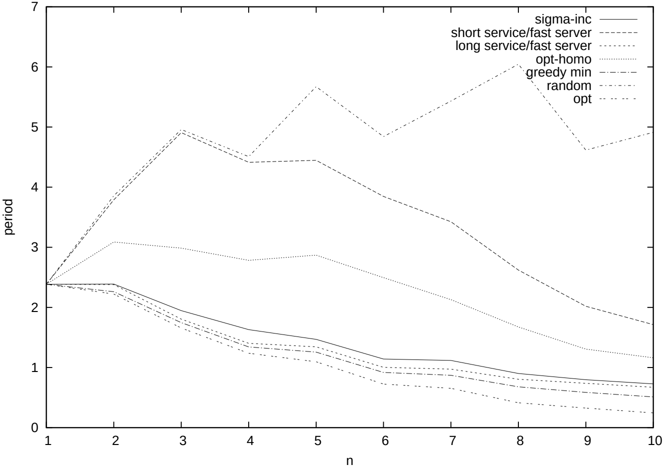


Figure: Experiment 3: with high selectivity.

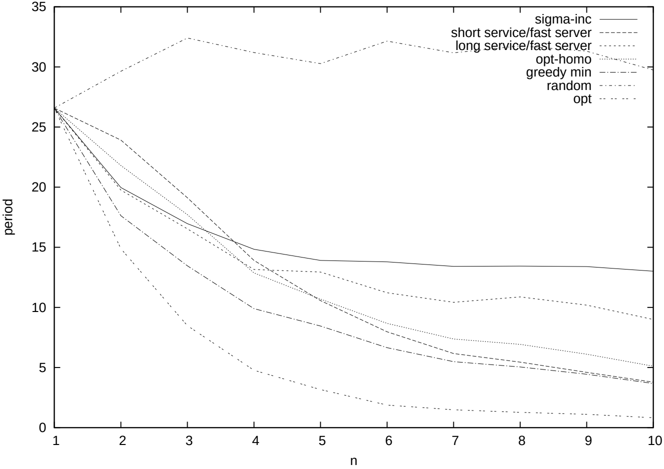


Figure: Experiment 4: with low speed.

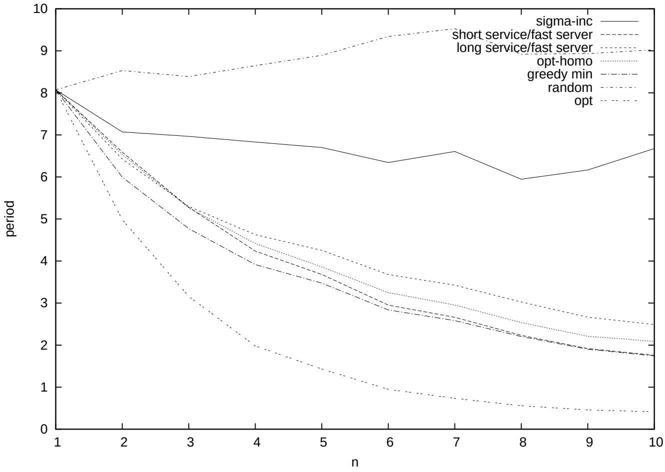


Figure: Experiment 5: with low heterogeneity.

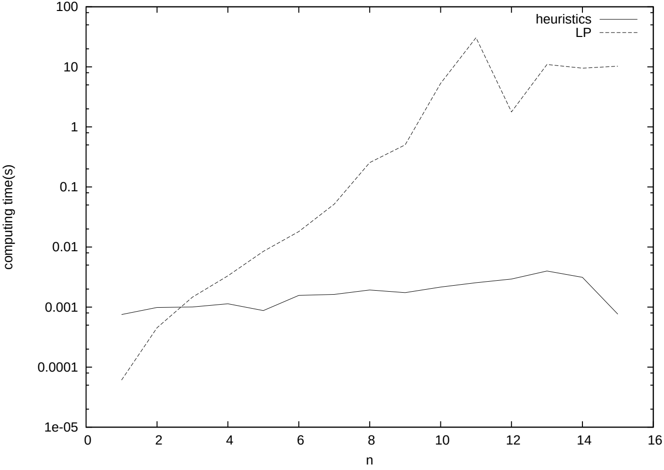


Figure: Experiment 1: Computing times.

References

M. P. Alessandro Agnetis, Paolo Detti and M. S. Sodhi.
Sequencing unreliable jobs on parallel machines.
Journal on Scheduling, May 2008.

U. Srivastava, K. Munagala, and J. Burge.
Ordering pipelined query operators with precedence constraints.
Technical report, Stanford University, November 2005.

U. Srivastava, K. Munagala, and J. Widom.
Operator placement for in-network stream query processing.
In *PODS '05: Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 250–258, New York, NY, USA, 2005. ACM.

U. Srivastava, K. Munagala, J. Widom, and R. Motwani.
Query optimization over web services.
In *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*, pages 355–366. VLDB Endowment, 2006.

Conclusion

The results:

- MinLatency-Hom is polynomial
- MinPeriod-Het is NP-complete
- MinLatency-Het is NP-complete
- BiCriteria-Hom is polynomial

The experiments on MinPeriod-NoPrec-Het:

- heuristics close to the optimal for small instances
- better performance than random

Future work:

- model with communication costs