A New Parallel Asynchronous Cellular Genetic Algorithm for Mapping in Grids

Frédéric Pinel, Bernabé Dorronsoro, Pascal Bouvry

NIDISC 2010

UNIVERSITÉ DU LUXEMBOURG

Outline

- Contribution
- Problem description
- Algorithms
- Results
- Future work

Contribution

- Apply a new multi-core model for independent task scheduling on grids
- New local search operator
- Improve previous results

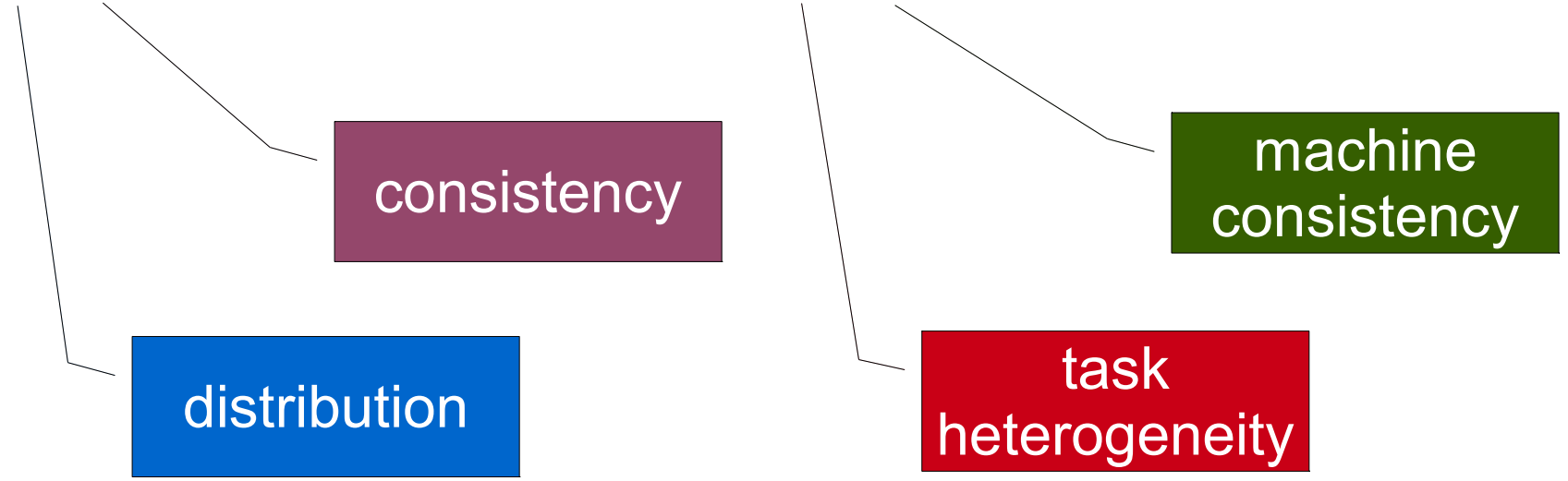
Problem description (1)

- Map heterogeneous independent tasks to heterogeneous machines
  - 512 tasks, 16 machines
- Expected Time to Compute (ETC) model
- Minimize makespan
- Limited execution time (90 s)

Problem description (2)

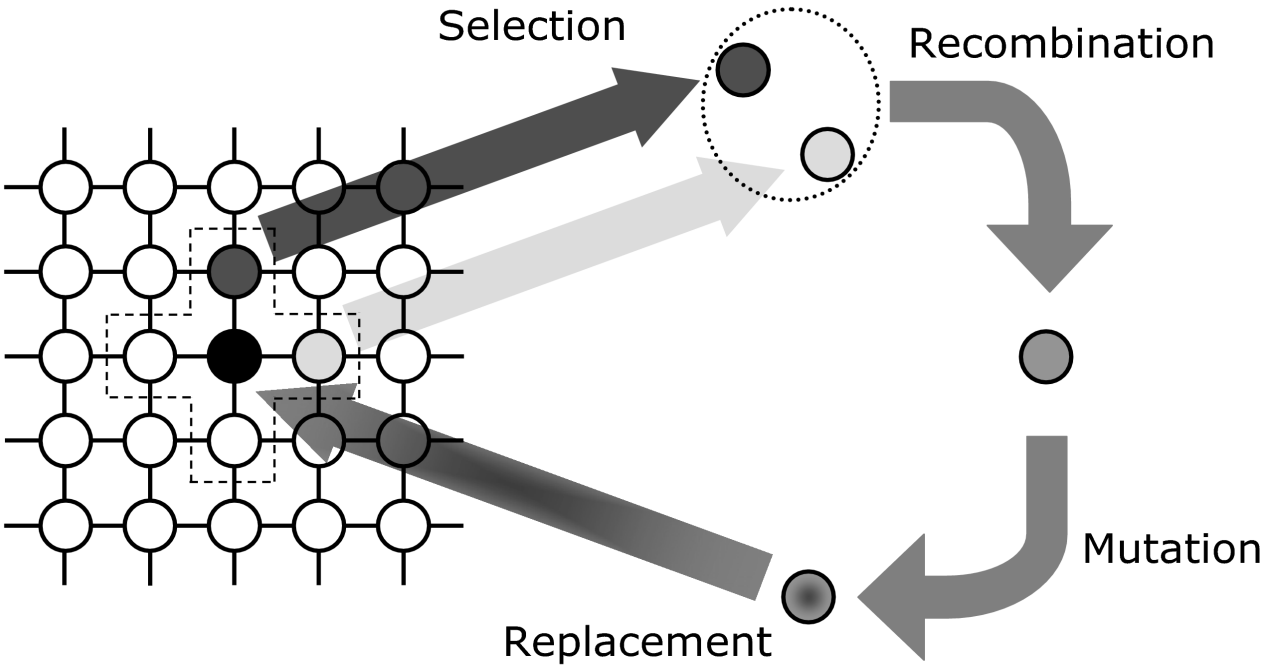
12 ETC instances used:

u_c_hihi.0	u_s_hihi.0	u_i_hihi.0
u_c_hilo.0	u_s_hilo.0	u_i_hilo.0
u_c_lohi.0	u_s_lohi.0	u_i_lohi.0
u_c_lolo.0	u_s_lolo.0	u_i_lolo.0



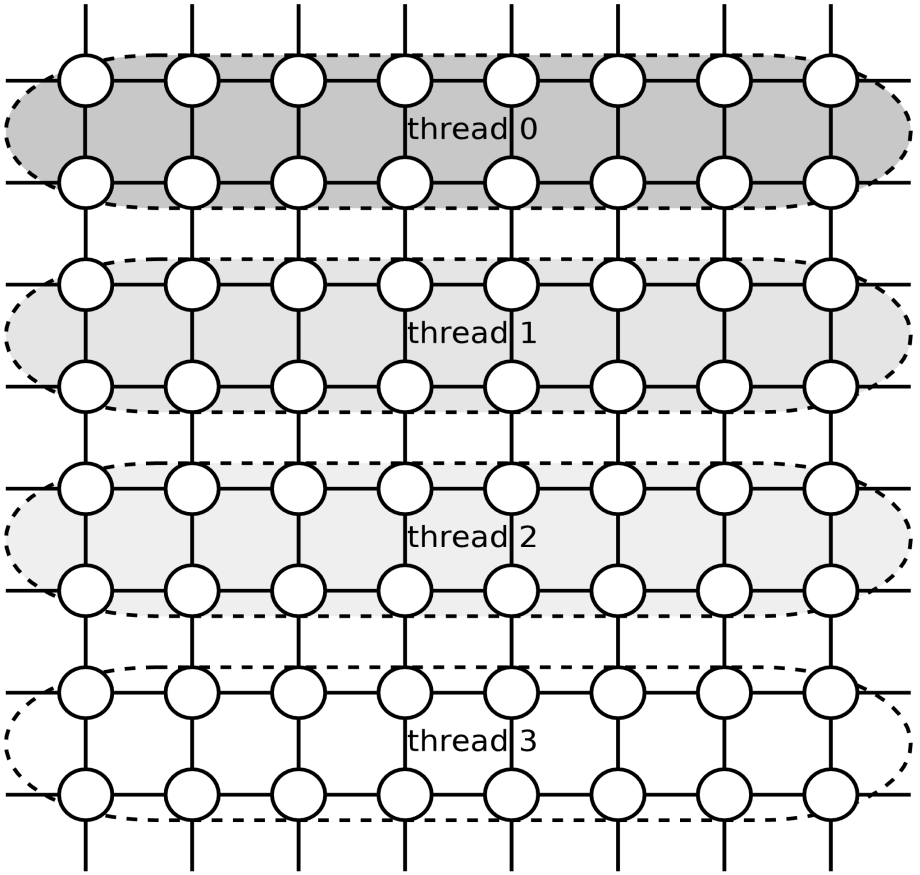
Algorithms (1)

- Cellular genetic algorithm
- Asynchronous



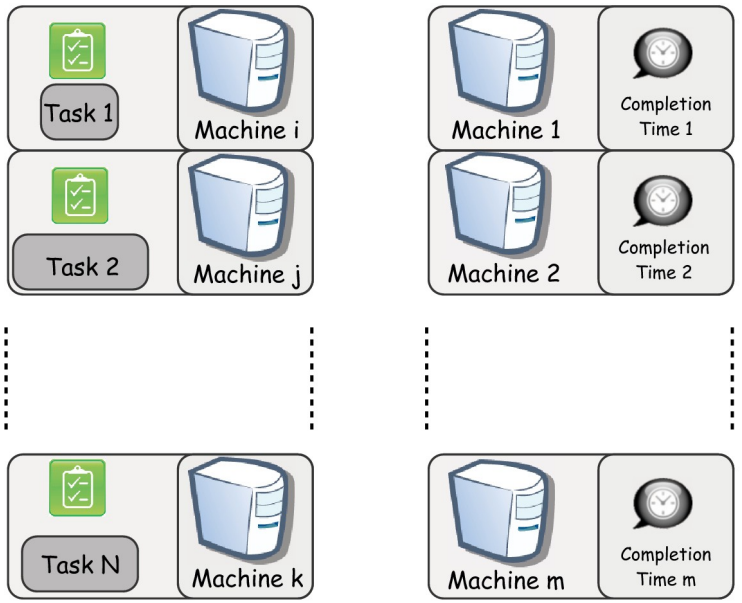
Algorithms (2)

Parallelism



Algorithms (3)

Representation

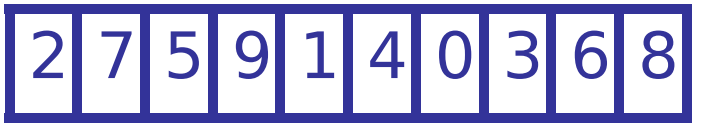


ETC

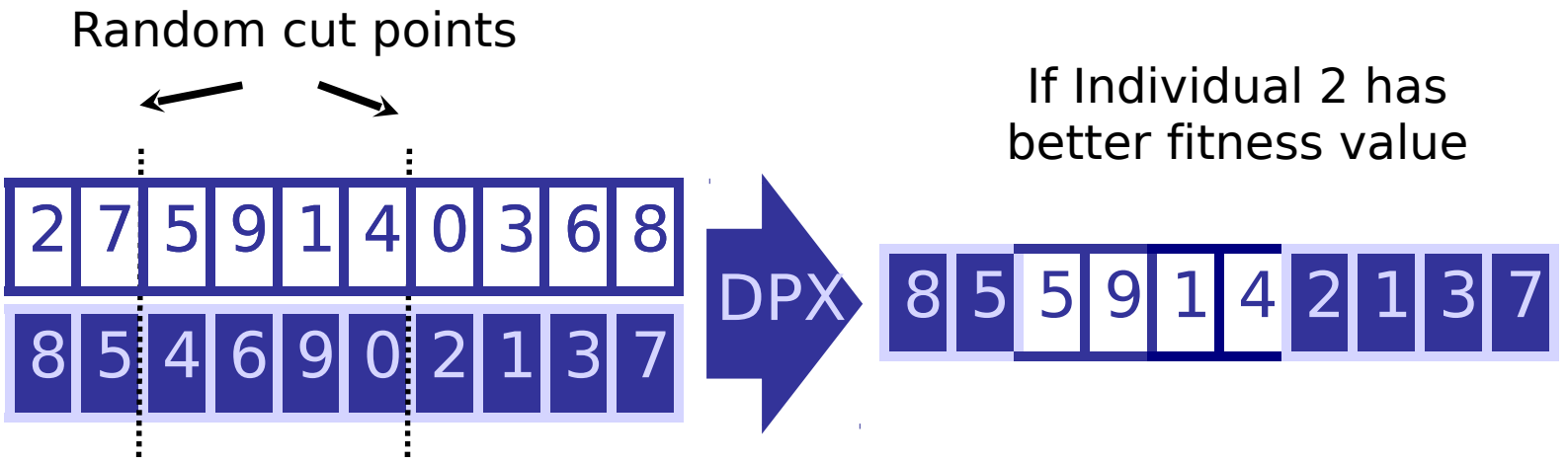
...	...
machine i	task i
machine i+1	task i+1
...	...
...	...
machine i	task i
machine i+1	task i+1
...	...

Algorithms (4)

- Representation



- Crossover : 2 point cross-over



Algorithms (5)

Local search

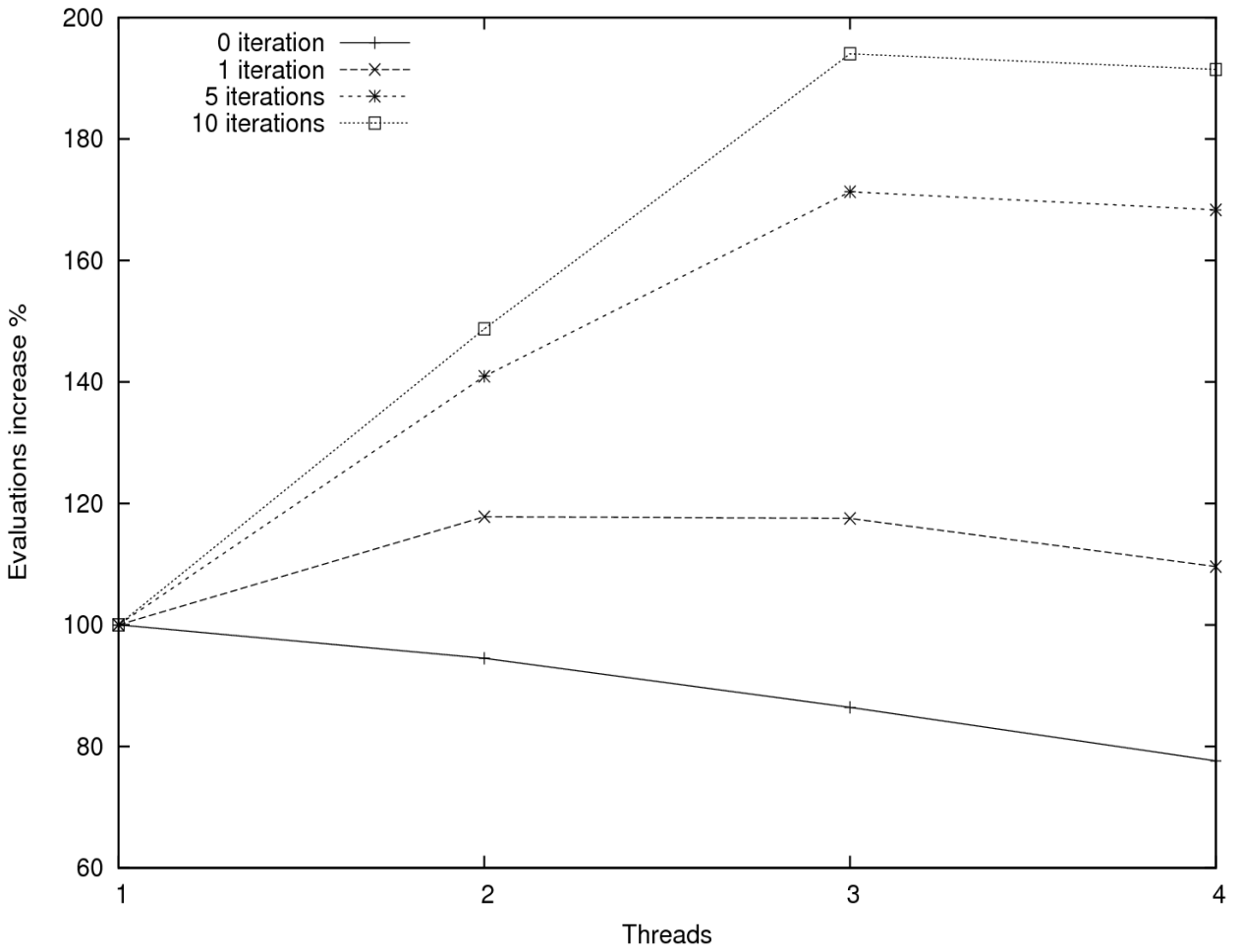
- Select a random task from most loaded machine
- Move to one of the least loaded machines, whose new completion time is smallest
- Iterate

Algorithms (6)

- Population: 16 x 16
- Initialize 1 individual with Min-Min
- Threads: 1-4
- Recombination: 1 or 2 point cross-over
- Mutation: move random task to random machine
- Local search iterations: 5-10
- Replace if better
- Processor: Xeon 2.8 GHz, 4 cores (2007)

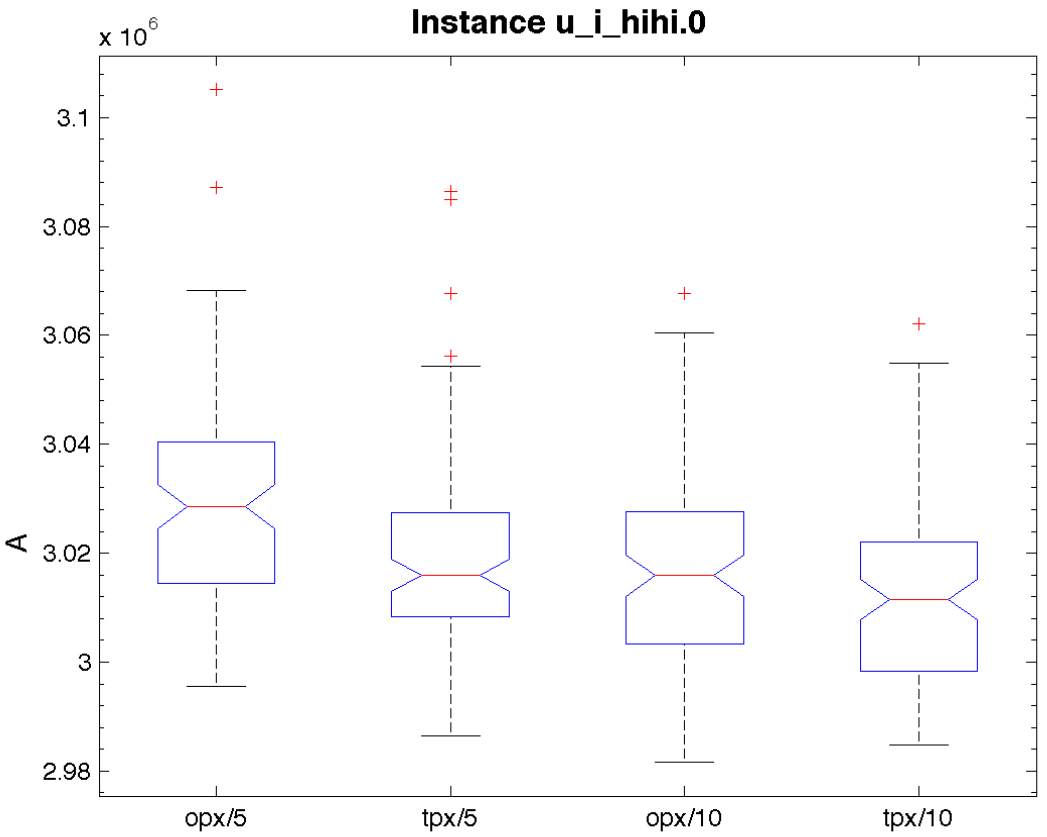
Results (1)

Speed-up



Results (2)

- Recombination
- Local search iterations



UNIVERSITÉ DU LUXEMBOURG

Results (3)

Comparison of mean makespan

instance	Struggle GA	CMA + LTH	PA-CGA
u_c_hihi.0	7,752,349.4	7,554,119.4	7,437,591.3
u_c_hilo.0	155,571.5	154,057.6	154,392.8
u_c_lohi.0	250,550.9	247,421.3	242,061.8
u_c_lolo.0	5,240.1	5,148.8	5,247.9
u_s_hihi.0	4,371,324.5	4,337,494.6	4,229,018.4
u_s_hilo.0	98,334.6	97426.2	97,424.8
u_s_lohi.0	127,762.5	128,216.1	125,579.3
u_s_lolo.0	3,539.4	3,488.3	3,526.6
u_i_hihi.0	3,080,025.8	3,054,137.7	3,011,581.3
u_i_hilo.0	76,307.9	75,005.5	74,476.8
u_i_lohi.0	107,294.2	106,158.7	104,490.1
u_i_lolo.0	2,610.2	2,597.0	2,602.5

Results (4)

Comparison of mean makespan

instance	Struggle GA	CMA + LTH	PA-CGA 10s	PA-CGA
u_c_hihi.0	7,752,349.4	7,554,119.4	7,518,600.7	7,437,591.3
u_c_hilo.0	155,571.5	154,057.6	154,963.6	154,392.8
u_c_lohi.0	250,550.9	247,421.3	245,012.9	242,061.8
u_c_lolo.0	5,240.1	5,148.8	5,261.4	5,247.9
u_s_hihi.0	4,371,324.5	4,337,494.6	4,277,497.3	4,229,018.4
u_s_hilo.0	98,334.6	97426.2	97,841.6	97,424.8
u_s_lohi.0	127,762.5	128,216.1	126,397.9	125,579.3
u_s_lolo.0	3,539.4	3,488.3	3,535.0	3,526.6
u_i_hihi.0	3,080,025.8	3,054,137.7	3,030,250.8	3,011,581.3
u_i_hilo.0	76,307.9	75,005.5	74,752.8	74,476.8
u_i_lohi.0	107,294.2	106,158.7	104,987.8	104,490.1
u_i_lolo.0	2,610.2	2,597.0	2,605.5	2,602.5

Summary

- Parallel asynchronous CGA for multi-core
- Applied to independent task mapping on grids
- Evaluated on benchmark instances
- Improved most results

Future work

- Paper extension:
  - Experiment with more instances of each ETC class
  - Study performance of algorithm with # threads (outside runtime considerations)
  - Heuristics & population initialization
  - Heterogeneous algorithms (parameters)