Advances in Metaheuristics on GPU

Thé Van Luong, El-Ghazali Talbi and Nouredine Melab

DOLPHIN Project Team

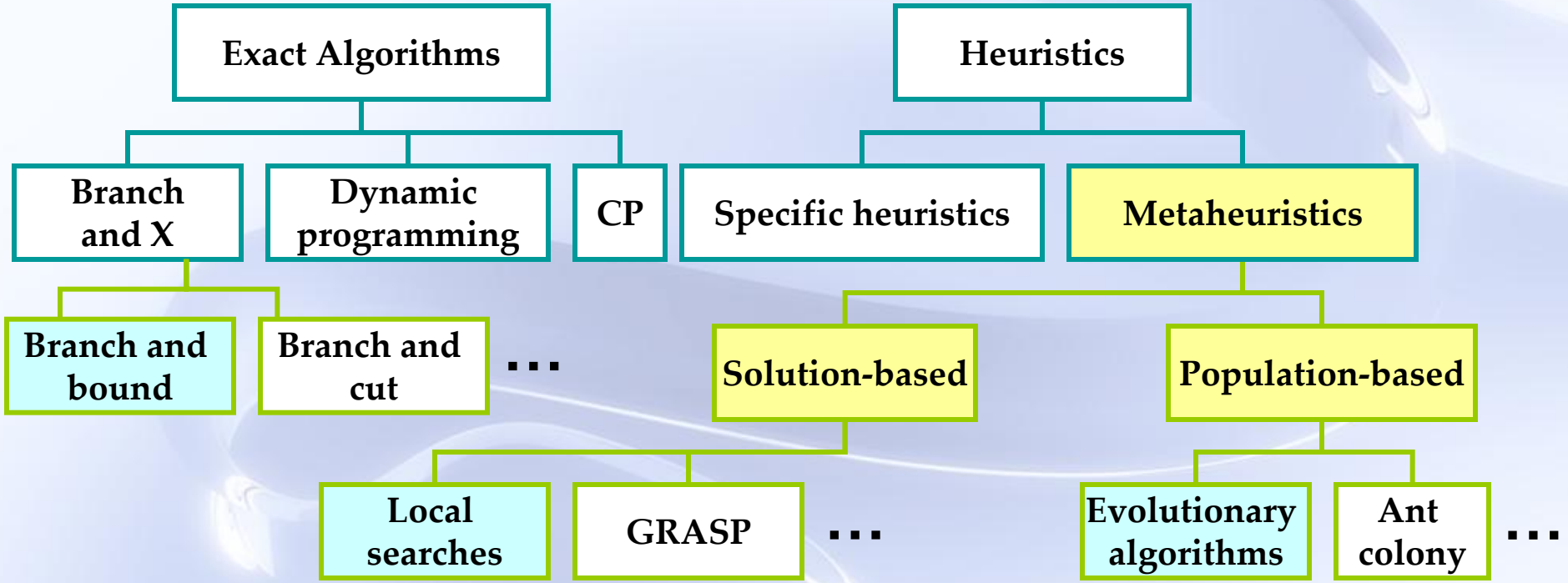
May 2011

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

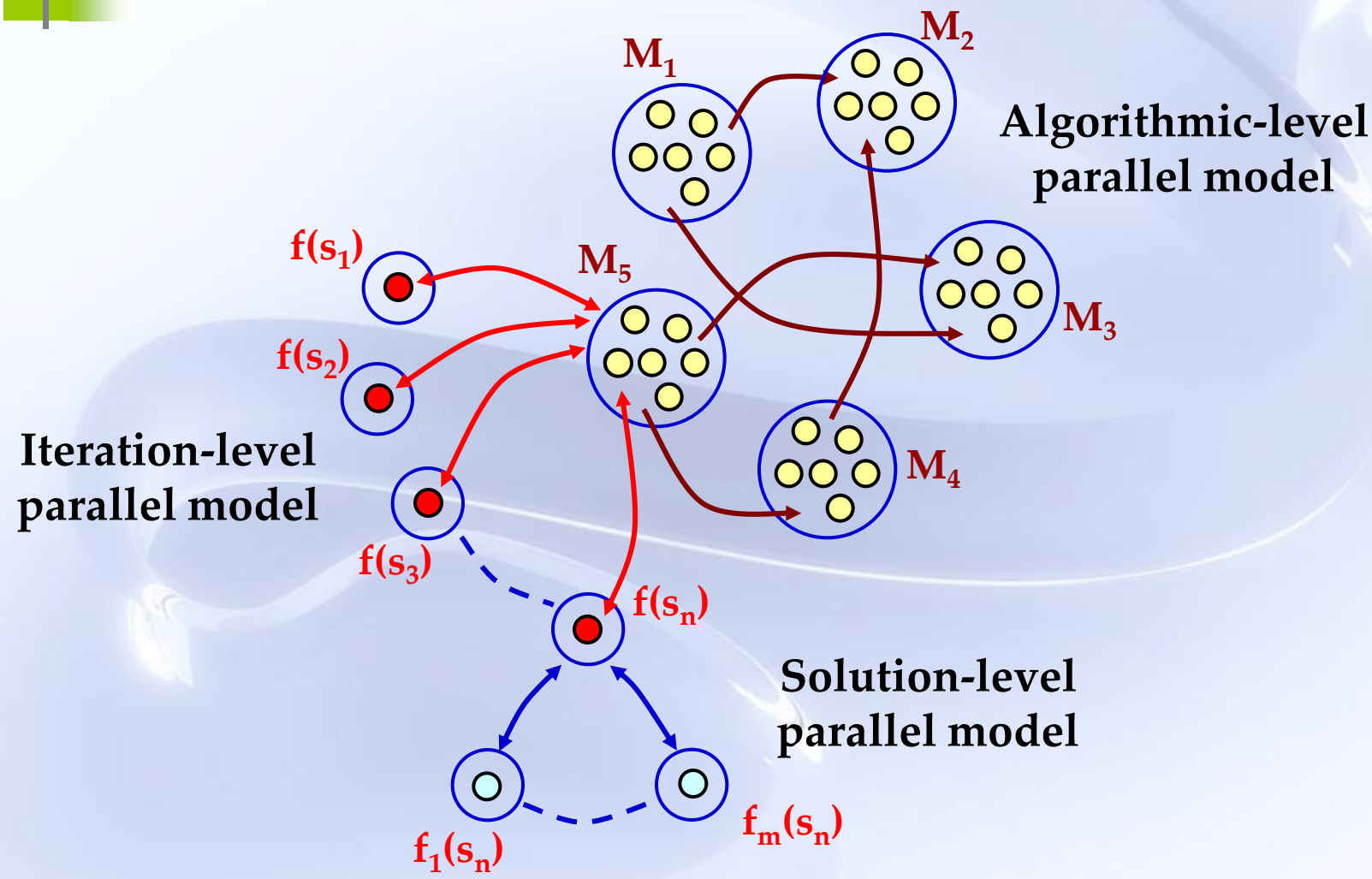
INRIA

centre de recherche LILLE - NORD EUROPE

Interests in optimization methods



Parallel models for metaheuristics



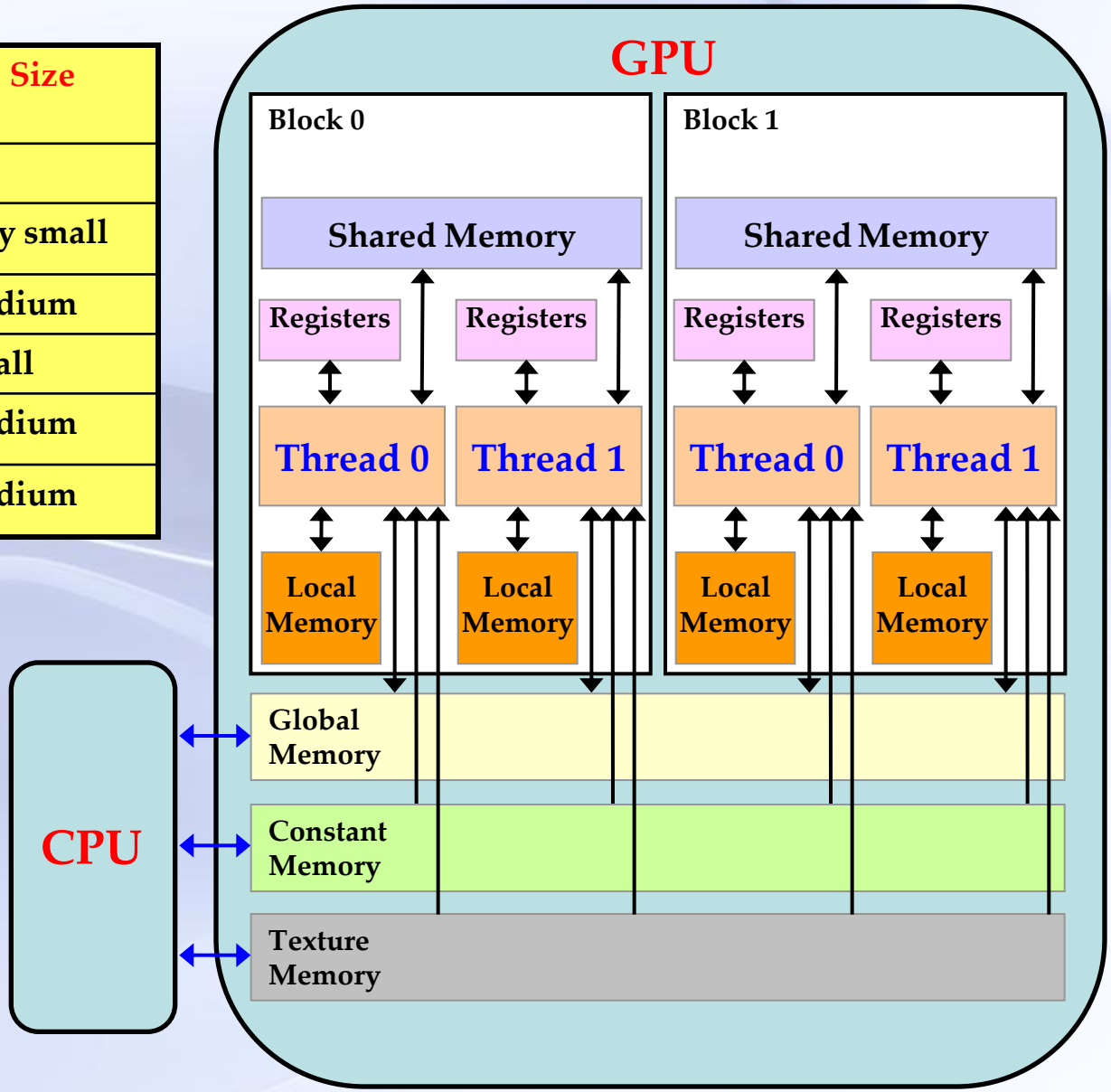
GPU Computing

- Used in the past for graphics and video applications …
- … but now popular for many other applications such as scientific computing [Owens *et al*. 2008]
- In the metaheuristics field:
  - Few existing works (Genetic algorithms [T-T. Wong 2006], Genetic programming [Harding *et al*. 2009], …), light tentative for the Tabu search algorithm [Zhu *et al*. 2009]
- Popularity due to the publication of the CUDA development toolkit allowing …
  - … GPU programming in a C-like language [Garland *et al*. 2008]

Many-cores and a hierarchy of memories

Memory type	Access latency	Size
Global	Medium	Big
Registers	Very fast	Very small
Local	Medium	Medium
Shared	Fast	Small
Constant	Fast (cached)	Medium
Texture	Fast (cached)	Medium

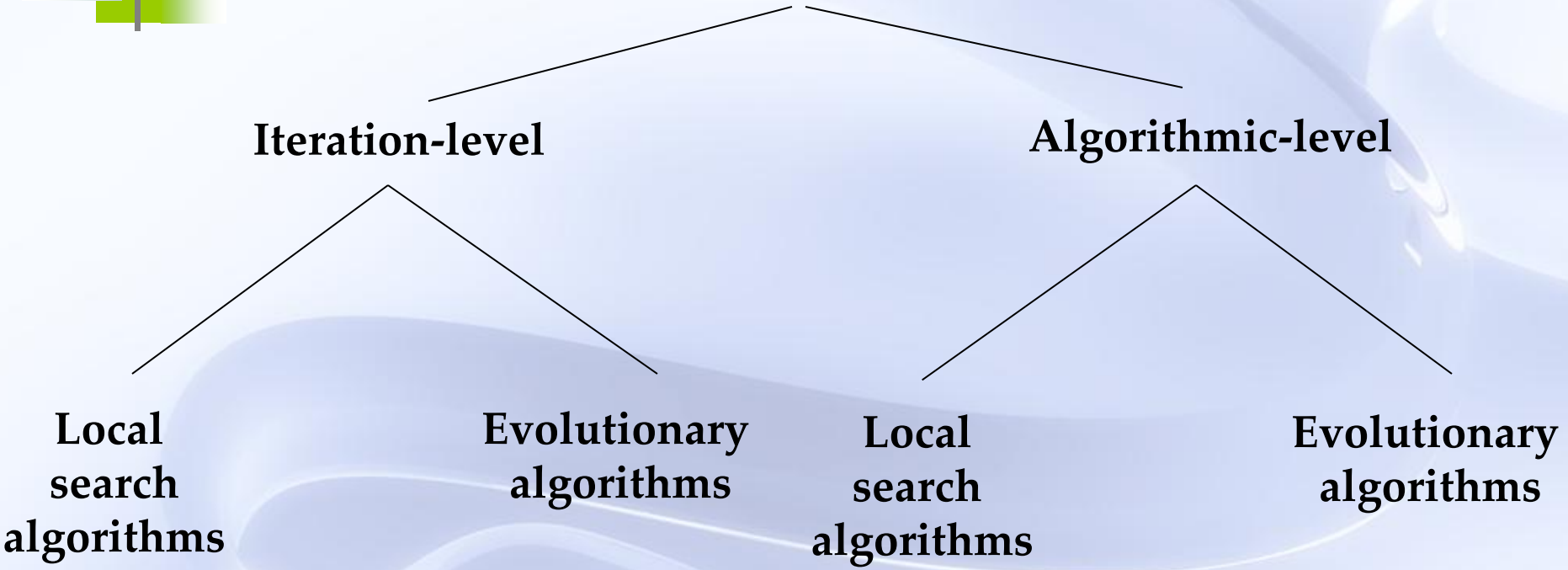
- Highly parallel multi-threaded many core
- High memory bandwidth compared to CPU
- Different levels of memory (different latencies)



Objective and challenging issues

- Re-think the parallel models to take into account the characteristics of GPU
  - Three major challenges …
- Challenge 1: efficient CPU-GPU cooperation
  - Work partitioning between CPU and GPU, data transfer optimization
- Challenge 2: efficient parallelism control
  - Threads generation control (memory constraints)
  - Efficient mapping between work units and threads Ids
- Challenge 3: efficient memory management
  - Which data on which memory (latency and capacity constraints) ?

Works completed



- Re-design of parallel models on GPU …
- … which allows to produce a bunch of general methods:
  - Hill climbing, tabu search, VNS, multi-start LS, …
  - GAs, EDAs, island model for GAs, …

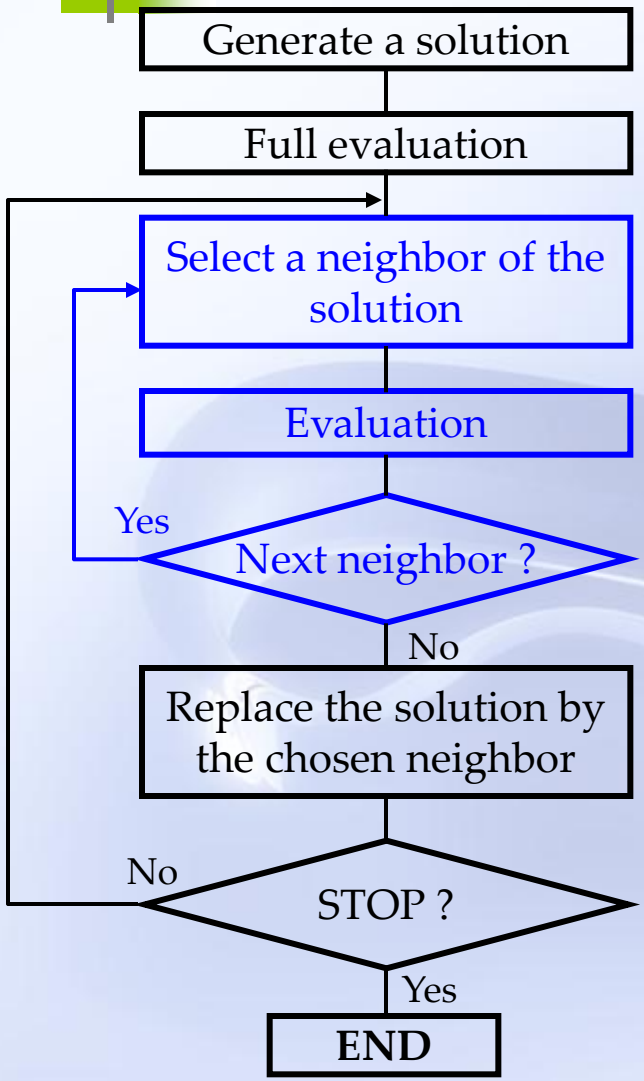
Works completed (2)

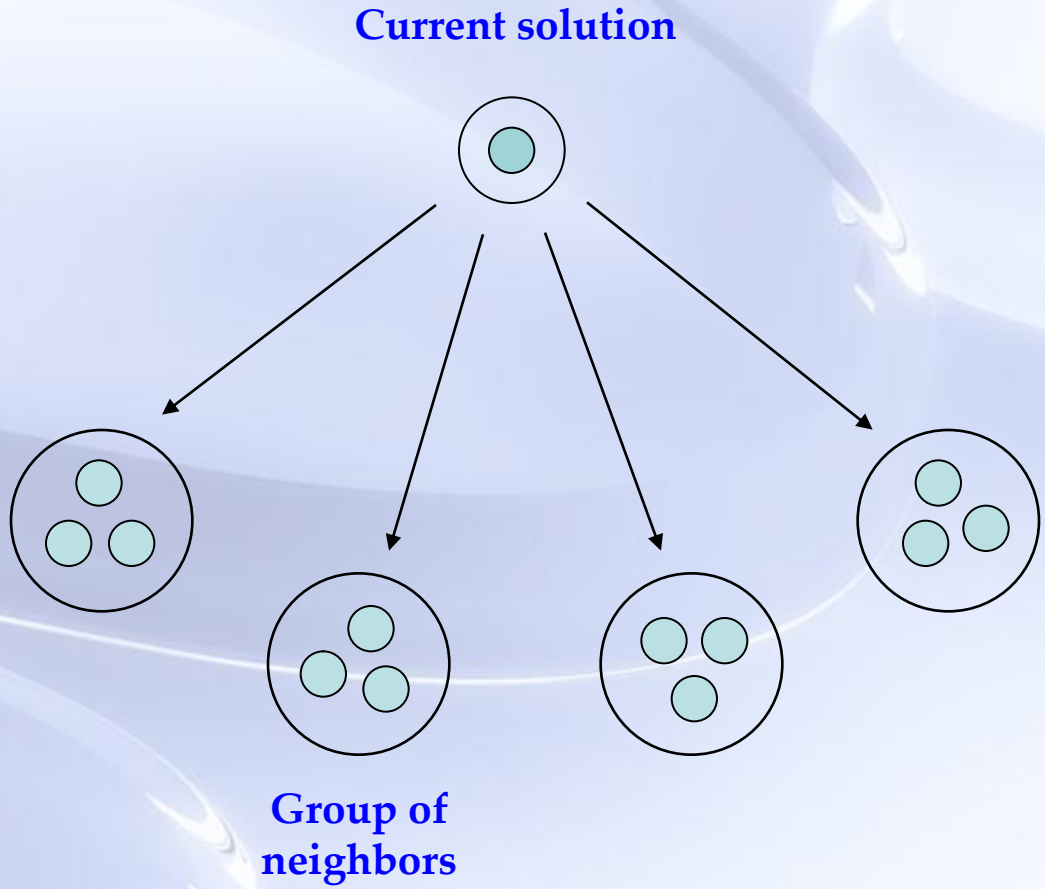
- Application to 9 combinatorial problems and 10 continuous tests functions.
- Speed-ups from experiments provide promising results :
  - Up to x50 for combinatorial problems
  - Up to x2000 for continuous test functions

GTX 280
(2008 configuration)

- 1 international journal.
- 9 international conference proceedings.
- 1 national conference proceeding.
- 2 conference abstracts.
- 7 workshops and talks.
- 1 research report.

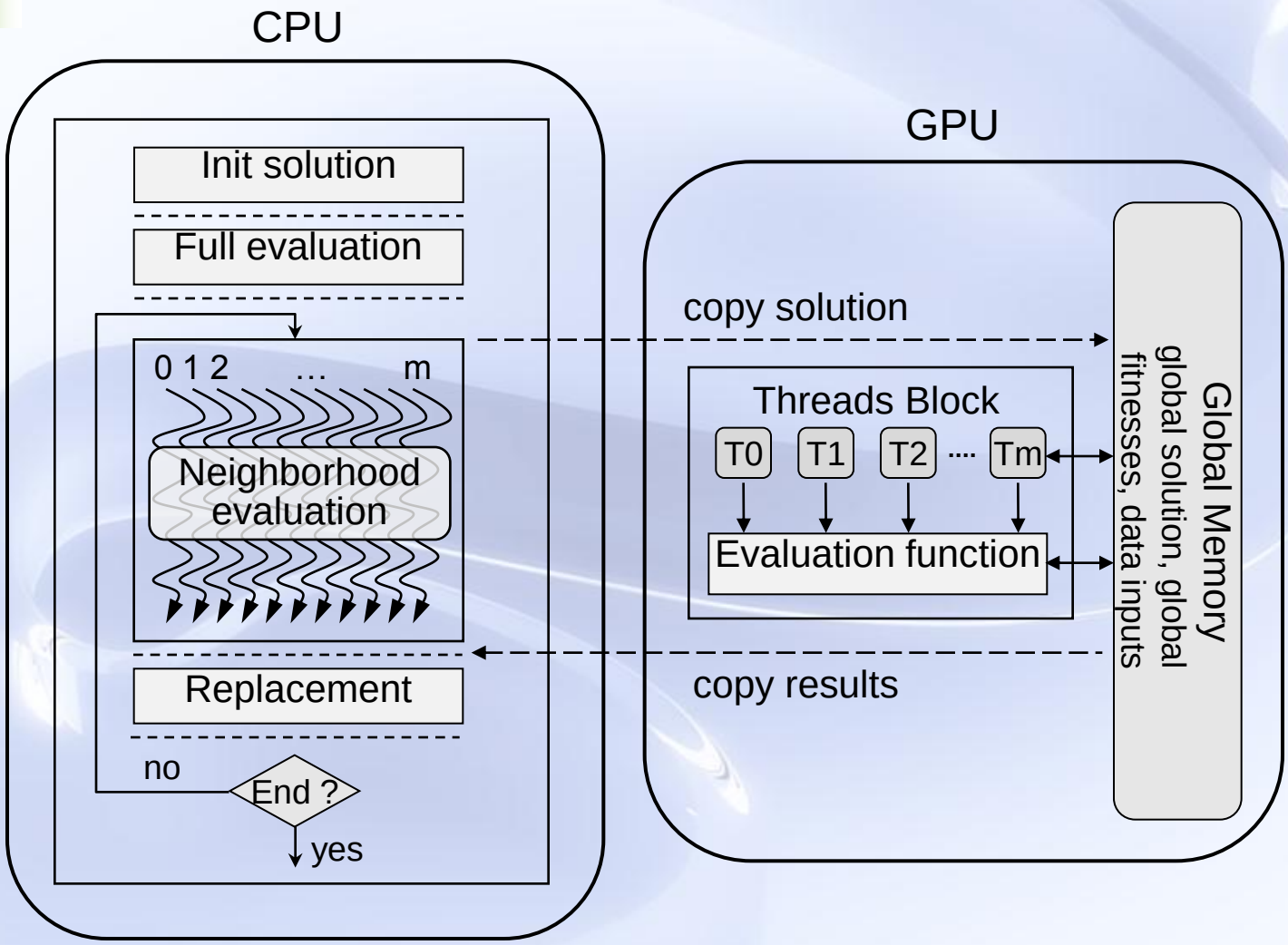
Iteration-level parallel model





- … Need of **massively parallel computing** on very **large** neighborhoods

Parallelization scheme on GPU (1)



Parallelization scheme on GPU (2)

- Keypoints:
  - Generation of the neighborhood on the GPU side to minimize the CPU→GPU data transfer
  - If possible, thread reduction for the best solution selection to minimize the GPU→CPU data transfer
  - Efficient parallelism control control: mapping neighboring onto threads ids, efficient kernel for fitness evaluation – incremental evaluation
  - Efficient memory management (e.g. use of texture memory or L2 cache for Fermi cards)

Parallelization scheme on GPU (3)

- Cons:
  - Data transfers at each iteration
  - Definitely not better than an optimized problem-dependent implementation (data reorganization, alignment requirements, register pressure, …)
- Pros:
  - Common parallelization to many local search algorithms
  - Application to many class of problems and local search algorithms
  - Clear separation of concepts specific to the local search and specific to the parallelization -> framework

GPU Computing for Parallel Local Search Metaheuristic Algorithms
IEEE Transactions on Computers (under revision)

Outline

- Application to Combinatorial Problems
- Application to Continuous Problems
- Thread Control Optimization
- Comparison with Other Parallel and Distributed Architectures
- Application to Other Algorithms
- Application to Multiobjective Optimization Problems
- Integration of GPU-based Concepts in the ParadisEO Platform

Application to Combinatorial Problems

Parameter settings

- Hardware configurations
  - Configuration 1: laptop (2006)
    - Core 2 Duo 2 Ghz + 8600M GT 4 multiprocessors (32 cores)
  - Configuration 2 : desktop-computer (2007)
    - Core 2 Quad 2.4 Ghz + 8800 GTX 16 multiprocessors (128 cores)
  - Configuration 3 : workstation (2008)
    - Intel Xeon 3 Ghz + GTX 280 30 multiprocessors (240 cores)
- Tabu Search and parameters
  - Neighborhood generation and evaluation on GPU
  - 10.000 iterations, 30 runs

Quadratic Assignment Problem

- Swap neighborhood: n*(n-1)/2 neighbors
- Delta evaluation: O(n)
- Permutation representation

Instance	Core 2 Duo 2Ghz GeForce 8600M GT 4 multiprocessors			Core 2 Quad 2.4Ghz GeForce 8800 GTX 16 multiprocessors			Intel Xeon 3Ghz GeForce GTX 280 30 multiprocessors		
	CPU	GPU	GPU_{Tex}	CPU	GPU	GPU_{Tex}	CPU	GPU	GPU_{Tex}
tai30a	2.3	$4.2_{\times 0.5}$	$1.3_{\times 1.8}$	1.9	$2.3_{\times 0.8}$	$1.0_{\times 1.9}$	1.6	$1.1_{\times 1.4}$	$0.8_{\times 2.0}$
tai35a	3.3	$6.5_{\times 0.5}$	$1.6_{\times 2.1}$	2.7	$2.9_{\times 0.9}$	$1.2_{\times 2.3}$	2.4	$1.2_{\times 1.9}$	$0.9_{\times 2.6}$
tai40a	5.0	$9.7_{\times 0.5}$	$1.8_{\times 2.6}$	4.2	$3.7_{\times 1.1}$	$1.5_{\times 2.9}$	3.6	$1.3_{\times 2.7}$	$1.1_{\times 3.3}$
tai50a	9.9	$16_{\times 0.6}$	$3.0_{\times 3.2}$	8.4	$5.7_{\times 1.4}$	$1.8_{\times 4.6}$	6.9	$1.7_{\times 4.1}$	$1.3_{\times 5.3}$
tai60a	17	$28_{\times 0.6}$	$4.9_{\times 3.4}$	14	$8.4_{\times 1.6}$	$2.0_{\times 7.1}$	11	$2.0_{\times 5.6}$	$1.6_{\times 7.3}$
tai80a	42	$63_{\times 0.7}$	$10_{\times 4.2}$	36	$19_{\times 1.9}$	$4.5_{\times 8.1}$	30	$3.2_{\times 9.1}$	$2.8_{\times 10.8}$
tai100a	108	$139_{\times 0.8}$	$19_{\times 5.6}$	76	$33_{\times 2.3}$	$8.7_{\times 8.8}$	60	$5.5_{\times 10.9}$	$3.7_{\times 16.5}$

Permuted Perceptron Problem (1)

- 1-Hamming distance neighborhood: n neighbors
- Delta evaluation: O(n)
- Binary encoding

Instance	Core 2 Duo 2Ghz GeForce 8600M GT 4 multiprocessors			Core 2 Quad 2.4Ghz GeForce 8800 GTX 16 multiprocessors			Intel Xeon 3Ghz GeForce GTX 280 30 multiprocessors		
	CPU	GPU	GPU_{Tex}	CPU	GPU	GPU_{Tex}	CPU	GPU	GPU_{Tex}
73-73	1.5	$6.1_{\times 0.2}$	$5.0_{\times 0.3}$	1.1	$3.3_{\times 0.3}$	$3.0_{\times 0.4}$	1.1	$3.5_{\times 0.3}$	$2.8_{\times 0.5}$
81-81	2.0	$6.7_{\times 0.3}$	$5.3_{\times 0.3}$	1.4	$3.9_{\times 0.4}$	$3.5_{\times 0.4}$	1.3	$3.8_{\times 0.3}$	$3.1_{\times 0.5}$
101-117	3.3	$8.9_{\times 0.4}$	$6.6_{\times 0.5}$	2.5	$4.8_{\times 0.5}$	$4.3_{\times 0.6}$	2.2	$4.9_{\times 0.4}$	$3.9_{\times 0.6}$
201-217	11	$19_{\times 0.6}$	$12_{\times 0.9}$	8.8	$10_{\times 0.8}$	$8.1_{\times 1.1}$	8.1	$8.8_{\times 0.9}$	$7.7_{\times 1.1}$
401-417	43	$53_{\times 0.8}$	$24_{\times 1.7}$	33	$28_{\times 1.2}$	$18_{\times 1.8}$	29	$16_{\times 1.9}$	$13_{\times 2.2}$
601-617	189	$169_{\times 1.1}$	$98_{\times 1.9}$	151	$96_{\times 1.6}$	$68_{\times 2.2}$	105	$47_{\times 2.2}$	$43_{\times 2.4}$
801-817	373	$244_{\times 1.4}$	$120_{\times 3.1}$	280	$120_{\times 2.4}$	$84_{\times 3.4}$	201	$54_{\times 3.6}$	$49_{\times 4.2}$
1001-1017	599	$345_{\times 1.8}$	$147_{\times 4.1}$	456	$145_{\times 3.1}$	$100_{\times 4.5}$	338	$63_{\times 5.3}$	$60_{\times 5.7}$
1301-1317	1180	$571_{\times 2.2}$	$273_{\times 4.3}$	808	$227_{\times 3.6}$	$169_{\times 4.8}$	690	$93_{\times 7.4}$	$82_{\times 8.1}$

Permuted Perceptron Problem (2)

- 2-Hamming distance neighborhood
- n*(n-1)/2 neighbors

Instance	Core 2 Duo 2Ghz GeForce 8600M GT 4 multiprocessors			Core 2 Quad 2.4Ghz GeForce 8800 GTX 16 multiprocessors			Intel Xeon 3Ghz GeForce GTX 280 30 multiprocessors		
	CPU	GPU	GPU_{Tex}	CPU	GPU	GPU_{Tex}	CPU	GPU	GPU_{Tex}
73-73	2.9	$3.9_{\times 0.7}$	$0.8_{\times 3.6}$	2.2	$1.3_{\times 1.7}$	$0.2_{\times 10.1}$	2.1	$0.2_{\times 9.9}$	$0.2_{\times 10.9}$
81-81	4.2	$5.2_{\times 0.8}$	$1.1_{\times 3.8}$	2.8	$1.7_{\times 1.7}$	$0.3_{\times 10.4}$	2.7	$0.3_{\times 10.3}$	$0.2_{\times 12.2}$
101-117	11	$12_{\times 0.9}$	$2.5_{\times 4.4}$	7.3	$4.1_{\times 1.8}$	$0.6_{\times 12.4}$	7.0	$0.6_{\times 11.8}$	$0.4_{\times 18.1}$
201-217	70	$75_{\times 0.9}$	$15_{\times 4.7}$	51	$25_{\times 2.0}$	$3.3_{\times 15.4}$	48	$3.0_{\times 16.3}$	$1.9_{\times 25.3}$
401-417	573	$570_{\times 1.0}$	$103_{\times 5.4}$	441	$123_{\times 3.5}$	$24_{\times 18.3}$	403	$20_{\times 20.1}$	$14_{\times 28.8}$
601-617	3210	$1881_{\times 1.7}$	$512_{\times 6.3}$	2520	$351_{\times 7.1}$	$89_{\times 28.3}$	2049	$67_{\times 30.5}$	$51_{\times 40.1}$
801-817	8615	$4396_{\times 2.0}$	$1245_{\times 6.9}$	6929	$817_{\times 8.5}$	$212_{\times 32.8}$	5410	$154_{\times 35.3}$	$128_{\times 42.3}$
1001-1017	17521	$8474_{\times 2.1}$	$2421_{\times 7.2}$	14461	$1474_{\times 9.8}$	$409_{\times 35.2}$	11075	$292_{\times 37.9}$	$252_{\times 43.9}$
1301-1317	39465	$17910_{\times 2.2}$	$4903_{\times 8.0}$	33155	$3041_{\times 10.9}$	$911_{\times 36.2}$	25016	$651_{\times 38.5}$	$568_{\times 44.1}$

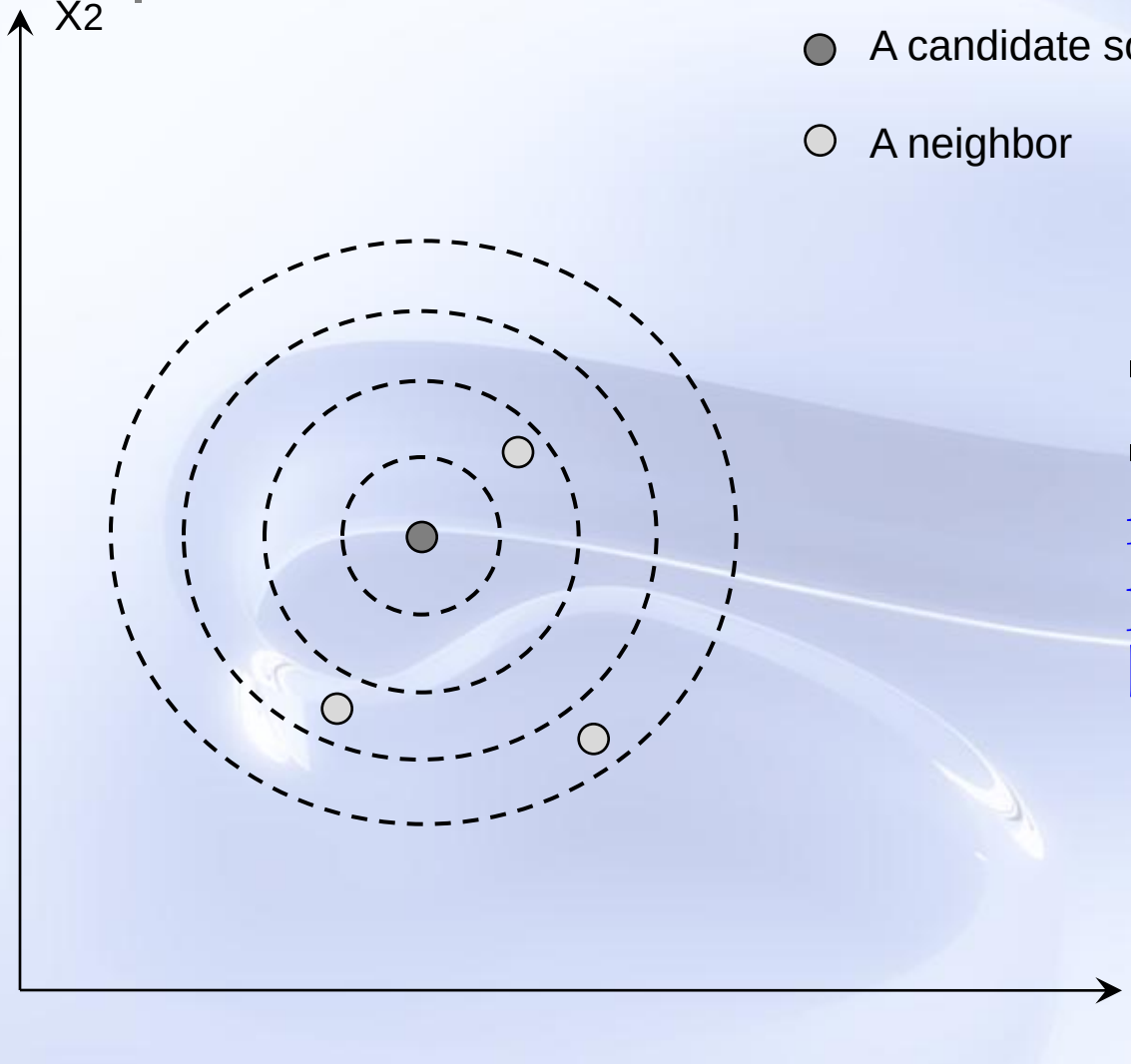
Permuted Perceptron Problem (3)

- 3-Hamming distance neighborhood
- n*(n-1)*(n-2)/6 neighbors

Instance	Core 2 Duo 2Ghz GeForce 8600M GT 4 multiprocessors			Core 2 Quad 2.4Ghz GeForce 8800 GTX 16 multiprocessors			Intel Xeon 3Ghz GeForce GTX 280 30 multiprocessors		
	CPU	GPU	GPU_{Tex}	CPU	GPU	GPU_{Tex}	CPU	GPU	GPU_{Tex}
73-73	538	$742_{\times 0.7}$	$167_{\times 3.2}$	312	$11.1_{\times 28.1}$	$10.6_{\times 29.4}$	278	$9.8_{\times 28.4}$	$9.2_{\times 30.2}$
81-81	788	$1023_{\times 0.8}$	$242_{\times 3.3}$	472	$16_{\times 29.5}$	$15_{\times 31.4}$	415	$13.3_{\times 31.2}$	$12.9_{\times 32.2}$
101-117	3181	$3329_{\times 0.9}$	$883_{\times 3.6}$	1831	$57_{\times 32.1}$	$55_{\times 33.3}$	1782	$50_{\times 35.6}$	$47_{\times 37.9}$
201-217	78835	$70325_{\times 1.1}$	$15471_{\times 5.1}$	25718	$650_{\times 39.5}$	$615_{\times 41.8}$	22375	$510_{\times 43.9}$	$488_{\times 45.8}$

Application to Continuous Problems

Continuous Neighborhood



- Vector of real values
- Neighbors are taken by random selection of one point inside each crown [Siarry et al.]

Continuous Weierstrass Function (1)

- Neighbor evaluation: $O(n^2)$
- Vector of real values
- No data inputs
- Continuous neighborhood: 10000 neighbors

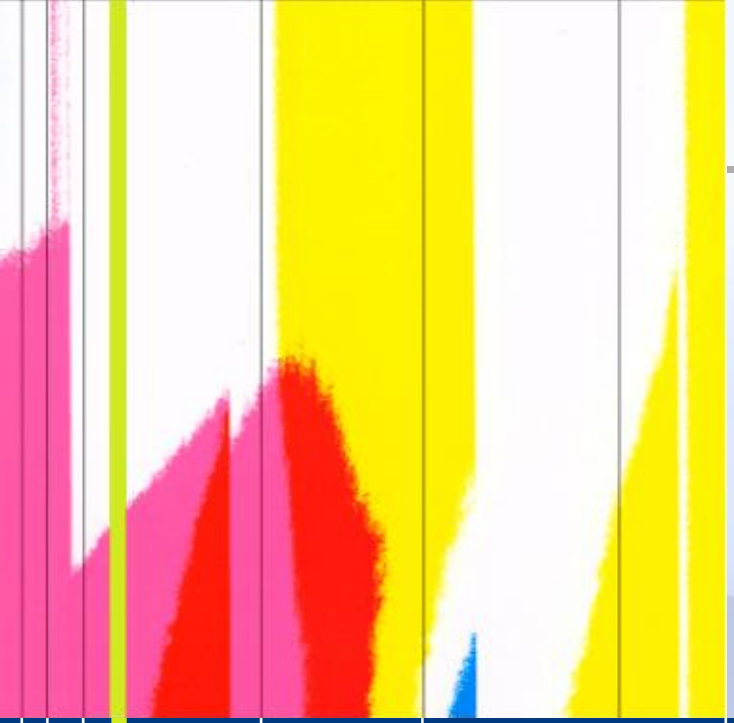
Instance	Core 2 Duo 2Ghz GeForce 8600M GT 4 multiprocessors		Core 2 Quad 2.4Ghz GeForce 8800 GTX 16 multiprocessors		Intel Xeon 3Ghz GeForce GTX 280 30 multiprocessors	
	CPU	GPU	CPU	GPU	CPU	GPU
1	3848	$98_{\times 39.2}$	2854	$27_{\times 105.7}$	2034	$16_{\times 127.1}$
2	10298	$247_{\times 41.7}$	5752	$52_{\times 110.6}$	4088	$32_{\times 127.8}$
3	15776	$354_{\times 44.6}$	8538	$77_{\times 110.9}$	6113	$47_{\times 130.0}$
4	20114	$440_{\times 45.7}$	11405	$102_{\times 111.8}$	8137	$61_{\times 133.4}$
5	23294	$501_{\times 46.5}$	14245	$127_{\times 112.1}$	10225	$76_{\times 134.5}$
6	28244	$603_{\times 46.8}$	17370	$151_{\times 115.0}$	12193	$90_{\times 135.5}$
7	33461	$712_{\times 47.0}$	20321	$173_{\times 117.4}$	14319	$104_{\times 137.7}$
8	36540	$774_{\times 47.2}$	23957	$203_{\times 118.0}$	16699	$120_{\times 139.2}$
9	42319	$889_{\times 47.6}$	27100	$229_{\times 118.3}$	19008	$134_{\times 141.9}$
10	51156	$1063_{\times 48.1}$	30709	$259_{\times 118.6}$	21095	$148_{\times 142.5}$

Continuous Weierstrass Function (2)

- Dimension Problem: 2
- Vary the neighborhood size

Neighborhood	Core 2 Duo 2Ghz GeForce 8600M GT 4 multiprocessors		Core 2 Quad 2.4Ghz GeForce 8800 GTX 16 multiprocessors		Intel Xeon 3Ghz GeForce GTX 280 30 multiprocessors	
	CPU	GPU	CPU	GPU	CPU	GPU
100	179	$17_{\times 10.5}$	57	$11_{\times 5.2}$	41	$11_{\times 3.7}$
500	482	$22_{\times 21.9}$	287	$15_{\times 19.1}$	203	$12_{\times 16.9}$
1000	1152	$31_{\times 37.1}$	576	$16_{\times 36.0}$	407	$13_{\times 31.3}$
5000	5317	$137_{\times 38.1}$	2874	$32_{\times 89.8}$	2034	$20_{\times 101.7}$
10000	10298	$247_{\times 41.7}$	5752	$52_{\times 110.6}$	4088	$32_{\times 127.8}$
20000	21330	$428_{\times 49.8}$	11488	$101_{\times 113.7}$	8508	$65_{\times 130.9}$
50000	53402	$1060_{\times 50.3}$	28718	$244_{\times 117.7}$	20364	$148_{\times 137.6}$
100000	104439	$2124_{\times 51.6}$	58857	$493_{\times 119.4}$	40700	$292_{\times 139.4}$

Thread Control Optimization

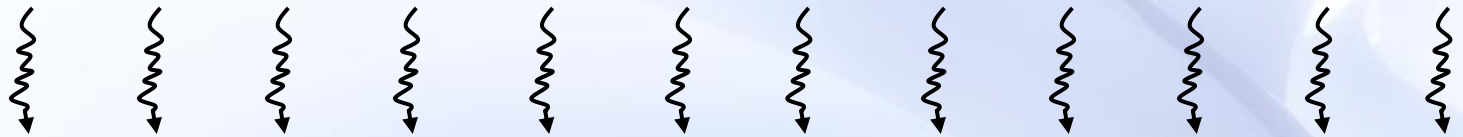


Traveling Salesman Problem

- Swap neighborhood: n*(n-1)/2 neighbors
- Delta evaluation: O(1)
- Permutation representation

Instance	Core 2 Duo 2Ghz GeForce 8600M GT 4 multiprocessors			Core 2 Quad 2.4Ghz GeForce 8800 GTX 16 multiprocessors			Intel Xeon 8 cores 3Ghz GeForce GTX 280 30 multiprocessors		
	CPU	GPU	GPU_{Tex}	CPU	GPU	GPU_{Tex}	CPU	GPU	GPU_{Tex}
eil101	2.2	$3.5_{\times 0.6}$	$1.8_{\times 1.2}$	1.6	$1.8_{\times 0.9}$	$1.1_{\times 1.5}$	1.4	$0.7_{\times 2.1}$	$0.6_{\times 2.3}$
d198	10	$12_{\times 0.8}$	$6.9_{\times 1.4}$	7.2	$5.1_{\times 1.4}$	$3.2_{\times 2.3}$	5.8	$1.5_{\times 3.9}$	$1.3_{\times 4.4}$
pcb442	55	$87_{\times 0.6}$	$36_{\times 1.5}$	31	$24_{\times 1.3}$	$14_{\times 2.2}$	27	$6.7_{\times 4.0}$	$6.0_{\times 4.5}$
rat783	199	$315_{\times 0.6}$	$144_{\times 1.4}$	117	$75_{\times 1.6}$	$42_{\times 2.8}$	93	$22_{\times 4.1}$	$20_{\times 4.7}$
d1291	695	$881_{\times 0.8}$	$503_{\times 1.4}$	491	$227_{\times 2.2}$	$140_{\times 3.5}$	365	$82_{\times 4.5}$	$71_{\times 5.1}$
pr2392	3370	–	–	2320	$874_{\times 2.6}$	$531_{\times 4.4}$	2394	$304_{\times 7.9}$	$286_{\times 8.4}$
fnl4461	14823	–	–	11721	–	–	11281	$1171_{\times 10.0}$	$1125_{\times 11.0}$
rl5915	27936	–	–	20939	–	–	20712	–	–

Thread Control (1)

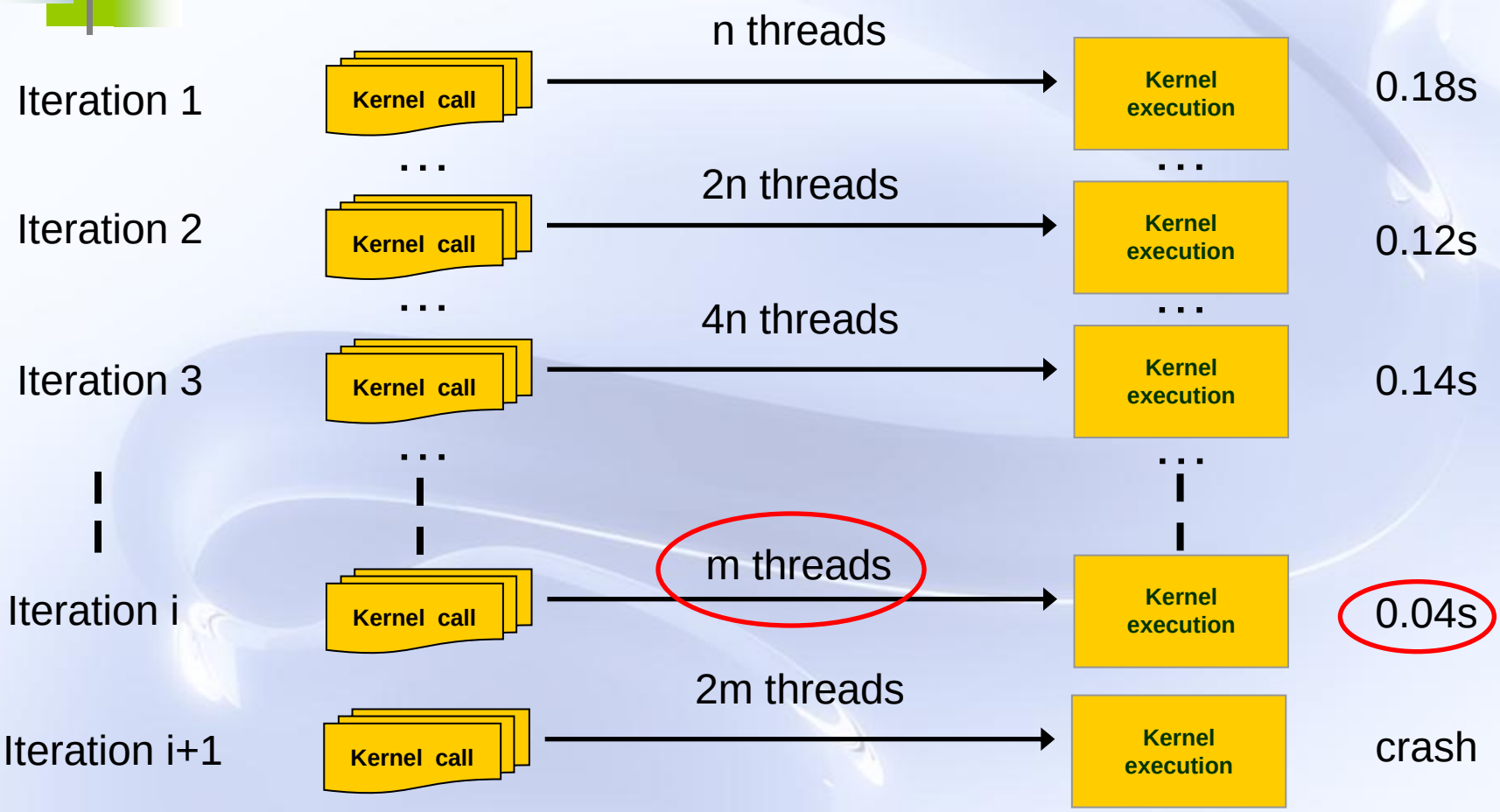


Neighborhood

	1	2	3	4	5	6	7	8	9	10	11	12
13	17	4	3	10	6	11	12	10	7	13	8	21
	13	14	15	16	17	18	19	20	21	22	23	24
Solution	11	5	8	19	12	19	13	11	13	14	19	29
	25	26	27	28	29	30	31	32	33	34	35	36
	14	13	16	13	4	15	11	19	6	18	5	24

- Increase the threads granularity …
  - … with associating each thread to MANY neighbors
  - … to avoid memory overflow (e.g. hardware register limitation)

Thread Control (2)



- Dynamic heuristic for parameters auto-tuning
  - To prevent the program from crashing
  - To obtain extra performance

Algorithm 3 Dynamic parameters tuning heuristic

```
Require: nb_trials;
 1: nb_threads := nearest_power_of_2 (solution_size);
 2: while nb_threads <= neighborhood_size do
 3:    nb_threads_block := 32;
 4:    while nb_threads_block <= 512 do
 5:       repeat
 6:          LS iteration pre-treatment on host side
 7:          Generation and evaluation kernel on GPU
 8:          if GPU kernel failure then
 9:             Restore (best_nb_threads);
10:             Restore (best_nb_threads_block);
11:             Exit procedure
12:          end if
13:          LS iteration post-treatment on host side
14:       until Time measurements of nb_trials
15:       if Best time improvement then
16:          best_nb_threads := nb_threads;
17:          best_nb_threads_block := nb_threads_block;
18:       end if
19:       nb_threads_block := nb_threads + 32;
20:    end while
21:    nb_threads := nb_threads * 2;
22: end while
23: Exit procedure
Ensure: best_nb_threads and best_nb_threads_block;
```

- Heuristic performed on the first iterations
- Vary the total number of threads (*2)
- Vary the number of threads per block (+32)
- Time measurements of each configuration for a certain number of trials
- Return the best configuration for tuning the rest of the algorithm
- If an error is detected, restore the previous best configuration found

Traveling Salesman Problem (2)

- Thread control
- Swap neighborhood: n*(n-1)/2 neighbors
- Delta evaluation: O(1)
- Permutation representation

Instance	Core 2 Duo 2Ghz GeForce 8600M GT 4 multiprocessors			Core 2 Quad 2.4Ghz GeForce 8800 GTX 16 multiprocessors			Intel Xeon 8 cores 3Ghz GeForce GTX 280 30 multiprocessors		
	CPU	GPU_{Tex}	GPU_{TexTC}	CPU	GPU_{Tex}	GPU_{TexTC}	CPU	GPU_{Tex}	GPU_{TexTC}
eil101	2.2	$1.8_{\times 1.2}$	$1.7_{\times 1.3}$	1.6	$1.1_{\times 1.5}$	$0.9_{\times 1.8}$	1.4	$0.6_{\times 2.3}$	$0.6_{\times 2.4}$
d198	10	$6.9_{\times 1.4}$	$6.8_{\times 1.5}$	7.2	$3.2_{\times 2.3}$	$3.0_{\times 2.4}$	5.8	$1.3_{\times 4.4}$	$1.3_{\times 4.5}$
pcb442	55	$36_{\times 1.5}$	$34_{\times 1.6}$	31	$14_{\times 2.2}$	$13_{\times 2.4}$	27	$6.0_{\times 4.5}$	$5.8_{\times 4.7}$
rat783	199	$144_{\times 1.4}$	$141_{\times 1.4}$	117	$42_{\times 2.8}$	$39_{\times 3.0}$	93	$20_{\times 4.7}$	$19_{\times 4.9}$
d1291	695	$503_{\times 1.4}$	$498_{\times 1.4}$	491	$140_{\times 3.5}$	$133_{\times 3.7}$	365	$71_{\times 5.1}$	$68_{\times 5.4}$
pr2392	3370	–	$1946_{\times 1.7}$	2320	$531_{\times 4.4}$	$519_{\times 4.5}$	2394	$286_{\times 8.4}$	$278_{\times 8.6}$
fnl4461	14823	–	$7133_{\times 2.1}$	11721	–	$1789_{\times 6.6}$	11281	$1125_{\times 10.0}$	$1110_{\times 10.2}$
rl5915	27936	–	$9142_{\times 3.1}$	20939	–	$2471_{\times 8.5}$	20712	–	$1461_{\times 14.2}$

Permuted Perceptron Problem

- 2-Hamming distance neighborhood
- n*(n-1)/2 neighbors
- Delta evaluation: O(n)
- Binary encoding
- Thread control

Instance	Core 2 Duo 2Ghz GeForce 8600M GT 4 multiprocessors			Core 2 Quad 2.4Ghz GeForce 8800 GTX 16 multiprocessors			Intel Xeon 3Ghz GeForce GTX 280 30 multiprocessors		
	CPU	GPU_{Tex}	GPU_{TexTC}	CPU	GPU_{Tex}	GPU_{TexTC}	CPU	GPU_{Tex}	GPU_{TexTC}
73-73	2.9	$0.8_{\times 3.6}$	$0.8_{\times 3.8}$	2.2	$0.2_{\times 10.1}$	$0.2_{\times 10.3}$	2.1	$0.2_{\times 10.9}$	$0.2_{\times 11.6}$
81-81	4.2	$1.1_{\times 3.8}$	$1.0_{\times 4.2}$	2.8	$0.3_{\times 10.4}$	$0.3_{\times 10.9}$	2.7	$0.2_{\times 12.2}$	$0.2_{\times 11.8}$
101-117	11	$2.5_{\times 4.4}$	$2.4_{\times 4.6}$	7.3	$0.6_{\times 12.4}$	$0.6_{\times 13.7}$	7.0	$0.4_{\times 18.1}$	$0.4_{\times 19.9}$
201-217	70	$15_{\times 4.7}$	$13_{\times 5.4}$	51	$3.3_{\times 15.4}$	$2.9_{\times 17.6}$	48	$1.9_{\times 25.3}$	$1.6_{\times 30.1}$
401-417	573	$103_{\times 5.4}$	$92_{\times 6.2}$	441	$24_{\times 18.3}$	$21_{\times 21.0}$	403	$14_{\times 28.8}$	$13_{\times 31.2}$
601-617	3210	$512_{\times 6.3}$	$446_{\times 7.2}$	2520	$89_{\times 28.3}$	$78_{\times 32.3}$	2049	$51_{\times 40.1}$	$45_{\times 45.3}$
801-817	8615	$1245_{\times 6.9}$	$1064_{\times 8.1}$	6929	$212_{\times 32.8}$	$182_{\times 38.1}$	5410	$128_{\times 42.3}$	$109_{\times 49.6}$
1001-1017	17521	$2421_{\times 7.2}$	$2087_{\times 8.4}$	14461	$409_{\times 35.2}$	$350_{\times 41.3}$	11075	$252_{\times 43.9}$	$216_{\times 51.3}$
1301-1317	39465	$4903_{\times 8.0}$	$4265_{\times 9.2}$	33155	$911_{\times 36.2}$	$779_{\times 42.5}$	25016	$568_{\times 44.1}$	$485_{\times 51.7}$

Comparison with Other Parallel and Distributed Architectures

Parallel Implementations

- Different parallel approaches and implementations have been proposed for local search algorithms on different architectures:
  - Massively Parallel Processors [e.g. Chakrapani et al. 1993]
  - Networks or Clusters of Workstations [e.g. T. Crainic et al. 1995]
  - Shared Memory or SMP machines [e.g. T. James et al. 2009].
  - Large-scale computational grids [N. Melab et al. 2007].

- Emergence of heterogeneous COWs and computational grids as standard platforms for high-performance computing.

COWs and grids

- Machines on COWs and grids have been selected to have the same computational power than the previous GPU configurations.
- A Myri-10G gigabit ethernet connects the different machines of the COWs .
- For the grid, the high-performance computing Grid'5000 has been used involving respectively two, five and seven French sites.

Architecture	Configuration 1		Configuration 2		Configuration 3	
	Machines	gflops	Machines	gflops	Machines	gflops
GPU	Core 2 Duo T5800 GeForce 8600M GT	76.8	Core 2 Quad Q6600 GeForce 8800 GTX	384	Intel Xeon E5450 GeForce GTX 280	981.12
COWs	Intel Xeon E5440 8 cores	90.656	4 Intel Xeon E5440 32 cores	362.624	11 Intel Xeon E5440 88 cores	995.236
Grid	2 Intel Xeon E5440 2 × 4 cores	90.656	Amd Opteron 2218 Intel Xeon E5520 2 Intel Xeon E5420 Intel Xeon E5440 40 cores	406.368	2 Intel Xeon E5520 2 AMD Opteron 2218 2 Intel Xeon E5520 4 Intel Xeon E5520 Intel Xeon X5570 Intel Xeon E5520 96 cores	979.104

Parallelization Scheme

- Parallelization: the neighborhood is decomposed into different partitions of equal size which are distributed among the cores of the different machines.

- The parallelization is synchronous and one has to wait for the termination of the exploration of all partitions.

- An hybrid OpenMP/MPI version has been produced to take advantage of both multi-core and distributed environments.

Cluster of Worstations (1)

- Permuted perceptron problem
- 2-Hamming distance neighborhood
- n*(n-1)/2 neighbors
- Delta evaluation: O(n)
- Binary encoding

Instance	Intel Xeon 2.83Ghz 8 cores			4 Intel Xeon 2.83 Ghz 32 cores			11 Intel Xeon 2.83Ghz 88 cores		
	CPU	GPU_{Tex}	COW	CPU	GPU_{Tex}	COW	CPU	GPU_{Tex}	COW
73-73	2.9	$0.8_{\times 3.6}$	$0.9_{\times 3.5}$	2.2	$0.2_{\times 10.1}$	$1.4_{\times 1.6}$	2.1	$0.2_{\times 10.9}$	$5.4_{\times 0.4}$
81-81	4.2	$1.1_{\times 3.8}$	$1.2_{\times 3.6}$	2.8	$0.3_{\times 10.4}$	$1.6_{\times 1.8}$	2.7	$0.2_{\times 12.2}$	$5.6_{\times 0.5}$
101-117	11	$2.5_{\times 4.4}$	$2.9_{\times 3.8}$	7.3	$0.6_{\times 12.4}$	$1.9_{\times 3.8}$	7.0	$0.4_{\times 18.1}$	$6.0_{\times 1.2}$
201-217	70	$15_{\times 4.7}$	$18_{\times 3.9}$	51	$3.3_{\times 15.4}$	$6.2_{\times 8.2}$	48	$1.9_{\times 25.3}$	$7.6_{\times 6.3}$
401-417	573	$103_{\times 5.4}$	$139_{\times 4.1}$	441	$24_{\times 18.3}$	$39_{\times 11.3}$	403	$14_{\times 28.8}$	$21_{\times 19.2}$
601-617	3210	$512_{\times 6.3}$	$966_{\times 3.3}$	2520	$89_{\times 28.3}$	$258_{\times 9.8}$	2049	$51_{\times 40.1}$	$115_{\times 17.8}$
801-817	8615	$1245_{\times 6.9}$	$2828_{\times 3.0}$	6929	$212_{\times 32.8}$	$737_{\times 9.4}$	5410	$128_{\times 42.3}$	$322_{\times 16.8}$
1001-1017	17521	$2421_{\times 7.2}$	$6307_{\times 2.8}$	14461	$409_{\times 35.2}$	$1708_{\times 8.5}$	11075	$252_{\times 43.9}$	$793_{\times 14.0}$
1301-1317	39465	$4903_{\times 8.0}$	$15257_{\times 2.6}$	33155	$911_{\times 36.2}$	$3968_{\times 8.4}$	25016	$568_{\times 44.1}$	$1807_{\times 13.8}$

Cluster of Worstations (2)

- Analysis of the data transfers including synchronizations

Instance	Intel Xeon 2.83Ghz 8 cores			4 Intel Xeon 2.83 Ghz 32 cores			11 Intel Xeon 2.83Ghz 88 cores		
	process	transfers	workers	process	transfers	workers	process	transfers	workers
73-73	4.2%	45.7%	50.1%	1.8%	93.0%	5.2%	0.7%	98.8%	0.5%
81-81	3.3%	45.3%	51.4%	2.3%	91.9%	5.8%	0.7%	98.7%	0.6%
101-117	2.6%	43.1%	54.3%	3.9%	83.8%	12.3%	1.2%	97.4%	1.4%
201-217	1.9%	42.4%	55.7%	5.7%	67.8%	26.5%	4.6%	88.2%	7.2%
401-417	1.8%	39.6%	58.6%	6.5%	57.1%	36.4%	12.1%	63.8%	24.1%
601-617	0.9%	52.0%	47.1%	3.5%	64.9%	31.6%	7.8%	71.7%	20.5%
801-817	0.6%	56.5%	42.9%	2.3%	67.4%	30.3%	5.3%	75.4%	19.3%
1001-1017	0.5%	59.4%	40.1%	1.9%	70.7%	27.4%	4.0%	79.9%	16.1%
1301-1317	0.4%	62.5%	37.1%	1.6%	71.3%	27.1%	3.5%	80.6%	15.9%

Computational Grid

Instance	2 Intel Xeon QC 2.83Ghz 8 cores			Configuration 5 machines 40 cores			Configuration 12 machines 96 cores		
	CPU	GPU_{Tex}	Grid	CPU	GPU_{Tex}	Grid	CPU	GPU_{Tex}	Grid
73-73	2.9	$0.8_{\times 3.6}$	$1.1_{\times 2.6}$	2.2	$0.2_{\times 10.1}$	$1.8_{\times 1.2}$	2.1	$0.2_{\times 10.9}$	$7.3_{\times 0.3}$
81-81	4.2	$1.1_{\times 3.8}$	$1.4_{\times 3.0}$	2.8	$0.3_{\times 10.4}$	$2.0_{\times 1.4}$	2.7	$0.2_{\times 12.2}$	$7.6_{\times 0.4}$
101-117	11	$2.5_{\times 4.4}$	$3.5_{\times 3.1}$	7.3	$0.6_{\times 12.4}$	$2.4_{\times 3.0}$	7.0	$0.4_{\times 18.1}$	$8.1_{\times 0.9}$
201-217	70	$15_{\times 4.7}$	$22_{\times 3.2}$	51	$3.3_{\times 15.4}$	$7.8_{\times 6.5}$	48	$1.9_{\times 25.3}$	$10_{\times 4.7}$
401-417	573	$103_{\times 5.4}$	$167_{\times 3.4}$	441	$24_{\times 18.3}$	$49_{\times 9.0}$	403	$14_{\times 28.8}$	$28_{\times 14.4}$
601-617	3210	$512_{\times 6.3}$	$1159_{\times 2.8}$	2520	$89_{\times 28.3}$	$323_{\times 7.8}$	2049	$51_{\times 40.1}$	$155_{\times 13.2}$
801-817	8615	$1245_{\times 6.9}$	$3394_{\times 2.5}$	6929	$212_{\times 32.8}$	$922_{\times 7.5}$	5410	$128_{\times 42.3}$	$425_{\times 12.7}$
1001-1017	17521	$2421_{\times 7.2}$	$7568_{\times 2.3}$	14461	$409_{\times 35.2}$	$2135_{\times 6.8}$	11075	$252_{\times 43.9}$	$1071_{\times 10.3}$
1301-1317	39465	$4903_{\times 8.0}$	$18308_{\times 2.2}$	33155	$911_{\times 36.2}$	$4960_{\times 6.7}$	25016	$568_{\times 44.1}$	$2439_{\times 10.2}$

Application to Other Algorithms

Iterated Local Search for the Q3AP

Instance	Nug12	Nug13	Nug15	Nug18	Nug22
Best known value	**580**	**1912**	2230	17836	42476
# solutions	**49/50**	**37/50**	**50/50**	**31/50**	**50/50**
CPU time	256 s	1879 s	1360 s	17447 s	16147 s
GPUTex time	15 s	64 s	38 s	415 s	353 s
Acceleration	x 17.3	x 29.2	x 36.0	x 42.0	x 45.7

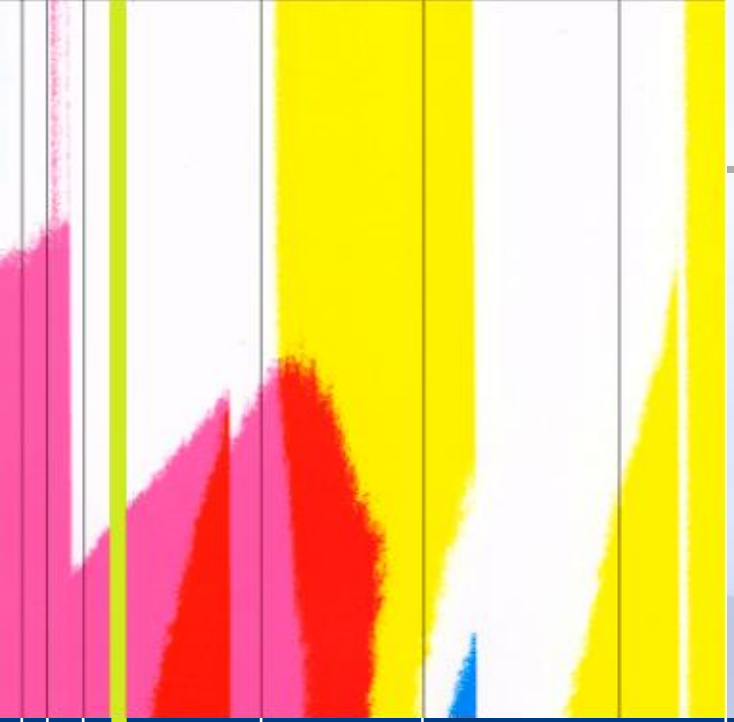
- Iterative local search (100 iters) + tabu search (5000 iters)
- Competitive algorithm
- Configuration: GTX 280

Hybrid Genetic Algorithm for the QAP

Instance	tai30a	tai35a	tai40a	tai50a	tai60a	tai80a	tai100a
Best known value	**1818146**	**2522002**	3139370	4938796	7205962	13511780	21052466
# solutions	27/30	23/30	18/30	10/30	6/30	4/30	2/30
CPU time	1h15min	2h24min	3h54min	10h2min	20h17min	66h	177h
GPUTex time	8min50s	12min56s	18min16s	45min	1h30min	4h45min	12h6min
Acceleration	x 8.5	x 11.1	x 12.8	x 13.2	x 13.4	x 13.8	x 14.6

- 10 individuals – 10 generations
- Evolutionary algorithm + iterative local search (3 iters) + tabu search (10000 iters)
- Neighborhood based on a 3-exchange operator
- Competitive algorithm
- Configuration: GTX 280

Multiobjective Optimization Problems



INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
LILLE - NORD EUROPE

Aggregated Tabu Search

- Multiobjective Flowshop Scheduling
  - 3 objectives (makespan, total tardiness and number of jobs delayed with regard to their due date)
  - Taillard instances extended

- 10000 global iterations per algorithm and 30 runs
- Configuration: GTX 280

Instance	20-10	20-20	50-10	50-20	100-10	100-20	200-10	200-20
CPU time	0.9s	1.9s	16s	32s	144s	271s	1190s	2221s
GPUTex time	1.7s	4.3s	6s	7.8s	11.5s	21s	77s	139s
Acceleration	x 0.6	x 0.7	x 3.8	x 4.1	x 12.6	x 12.9	x 15.4	x 16.0

PLS-1 (dominating neighbors)

- Pareto Multiobjective Local Search
- Only the dominating neighbors are added in the archive
- Archiving on CPU

Instance	20-10	20-20	50-10	50-20	100-10	100-20	200-10	200-20
CPU time	1.0s	1.9s	17s	33s	140s	275s	1172s	2196s
GPUTex time	1.7s	3.0s	4.3s	7.8s	11.7s	21s	78s	140s
Acceleration	x 0.6	x 0.6	x 3.8	x 4.2	x 12.0	x 13.1	x 15.1	x 15.7
# non-dominated solutions	11	13	19	20	31	34	61	71

PLS-1 (non-dominated neighbors)

- Pareto Multiobjective Local Search
- Only the non-dominated neighbors are added in the archive
- Archiving on CPU

Instance	20-10	20-20	50-10	50-20	100-10	100-20	200-10	200-20
CPU time	1.2s	2.0s	21s	39s	253s	312s	1361s	2672s
GPUTex time	1.8s	3.0s	8.2s	13.2s	49s	59s	218s	378s
Acceleration	x 0.7	x 0.7	x 2.5	x 3.0	x 5.1	x 5.3	x 6.4	x 7.1
# non-dominated solutions	77	83	396	596	1350	1530	1597	2061

PLS-1

- Analysis of the time spent for archiving

Instance	$PLS_{\succ}$			$PLS_{\not\prec}$		
	Evaluation	Archiving	LS process	Evaluation	Archiving	LS process
20-10	99.2%	0.7%	0.1%	95.2%	4.7%	0.1%
20-20	99.4%	0.5%	0.1%	98.2%	1.7%	0.1%
50-10	99.5%	0.4%	0.1%	89.7%	10.2%	0.1%
50-20	99.7%	0.2%	0.1%	94.2%	5.7%	0.1%
100-10	99.8%	0.1%	0.1%	88.2%	11.8%	0.1%
100-20	99.85%	0.1%	0.05%	93.6%	6.35%	0.05%
200-10	99.9%	0.05%	0.05%	92.3%	7.65%	0.05%
200-20	99.9%	0.05%	0.05%	94.2%	5.75%	0.05%

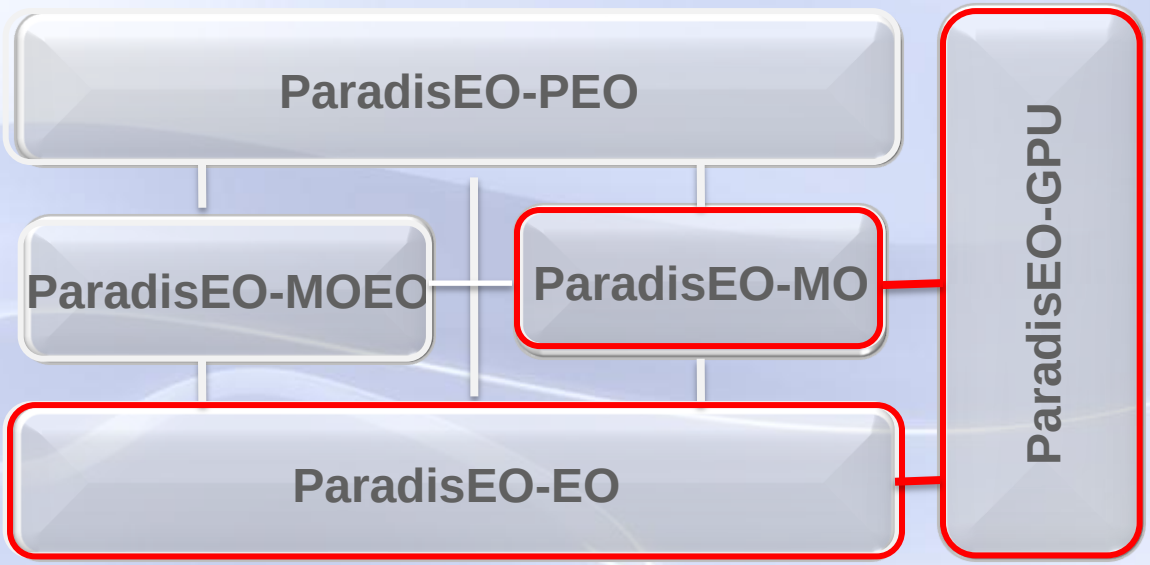
Integration of GPU-based Concepts in the ParadisEO Platform

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

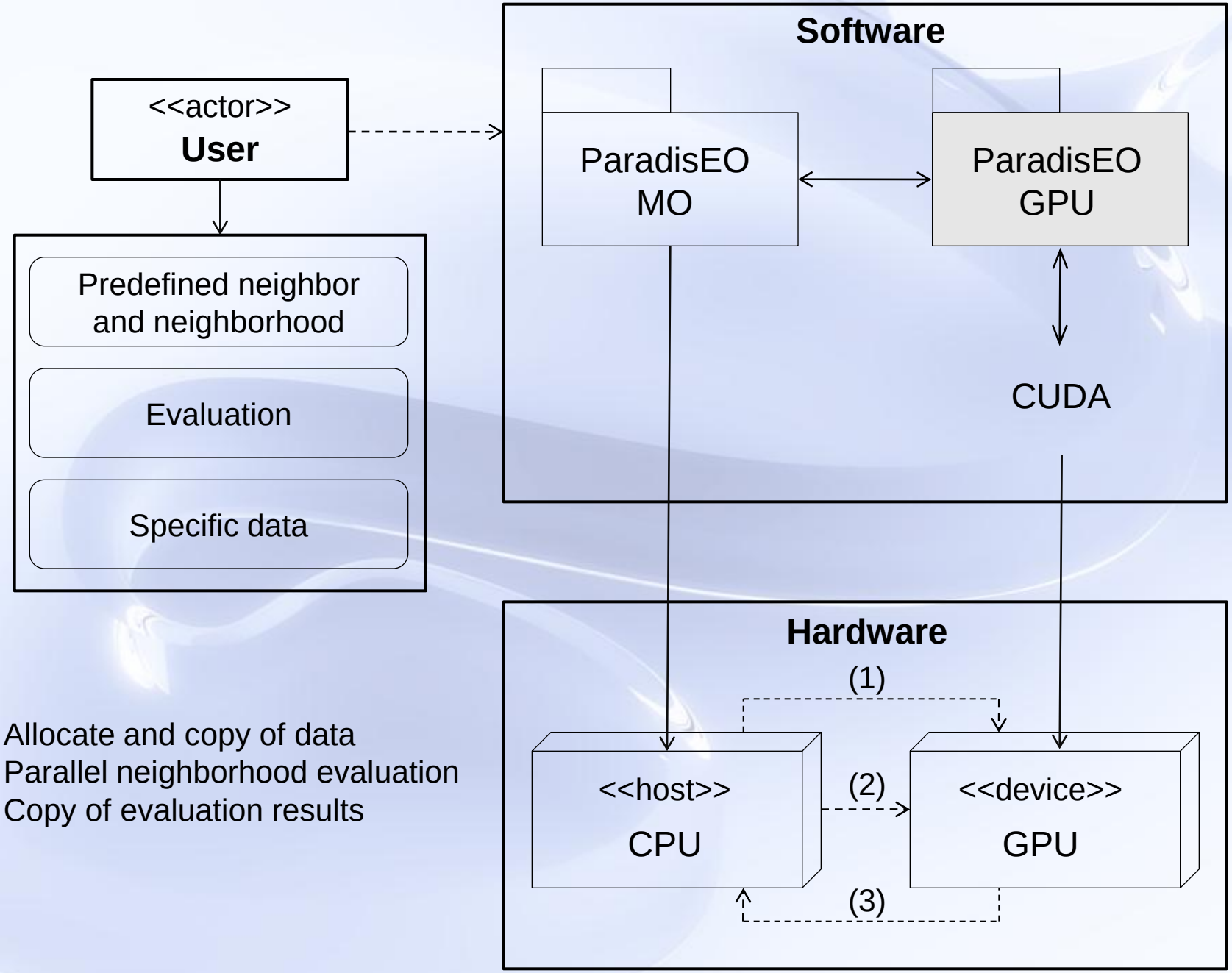
INRIA

centre de recherche
LILLE - NORD EUROPE

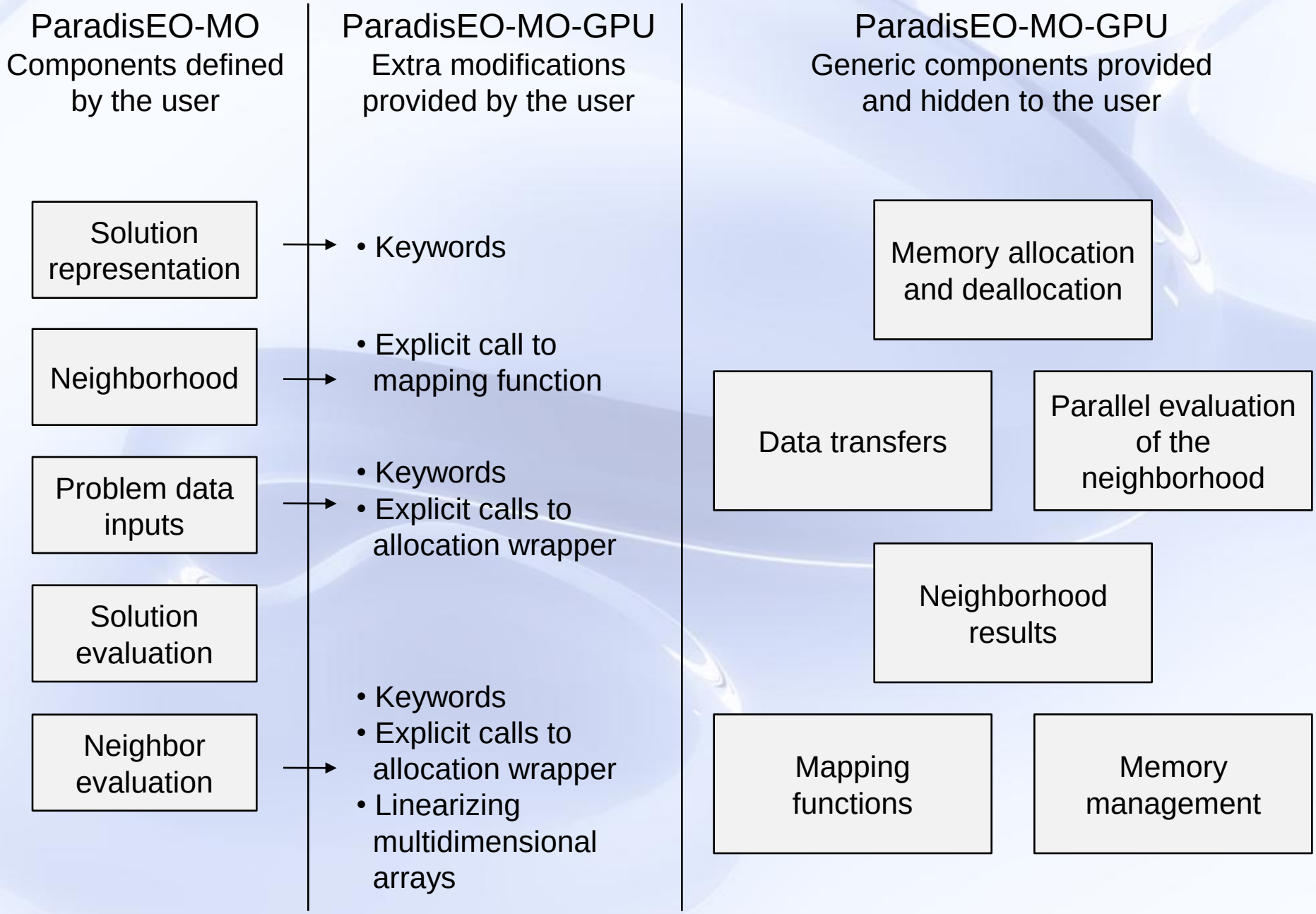
Integration in ParadisEO



One engineer position: Boufaras Karima
(Feb 2010 – Feb 2012)



(1) Allocate and copy of data
(2) Parallel neighborhood evaluation
(3) Copy of evaluation results


ParadisEO-MO
Components defined by the user
ParadisEO-MO-GPU
Extra modifications provided by the user
ParadisEO-MO-GPU
Generic components provided and hidden to the user
Solution representation
Neighborhood
Problem data inputs
Solution evaluation
Neighbor evaluation
• Keywords
• Explicit call to mapping function
• Keywords
• Explicit calls to allocation wrapper
• Keywords
• Explicit calls to allocation wrapper
• Linearizing multidimensional arrays
Memory allocation and deallocation
Data transfers
Parallel evaluation of the neighborhood
Neighborhood results
Mapping functions
Memory management

Results of the Platform (1)

- Quadratic assignment problem
- Neighborhood based on a 2-exchange operator
- Configuration: GTX 280
- Tabu search 10000 iterations

Instance	ParadisEO-MO			Optimized version			Perf. gap	
	CPU	GPU	Acc.	CPU	GPU	Acc.	CPU	GPU
tai30a	0.8	0.9	×0.9	0.7	0.7	×1.0	13%	23%
tai40a	1.9	1.2	×**1.6**	1.7	1.0	×**1.7**	11%	17%
tai50a	3.6	1.6	×**2.2**	3.0	1.3	×**2.3**	17%	19%
tai60a	6.0	2.0	×**3.0**	5.0	1.6	×**3.1**	17%	20%
tai80a	15.1	2.8	×**5.4**	12.3	2.1	×**5.8**	19%	25%
tai100a	31.7	3.8	×**8.3**	26.0	2.8	×**9.2**	18%	27%
tai150b	119.7	8.9	×**13.4**	98.1	6.5	×**15.1**	18%	27%

Results of the Platform (2)

- Permuted perceptron problem (7 smallest instances)
- Neighborhood based on a Hamming distance of two
- Configuration: GTX 480
- Tabu search 10000 iterations

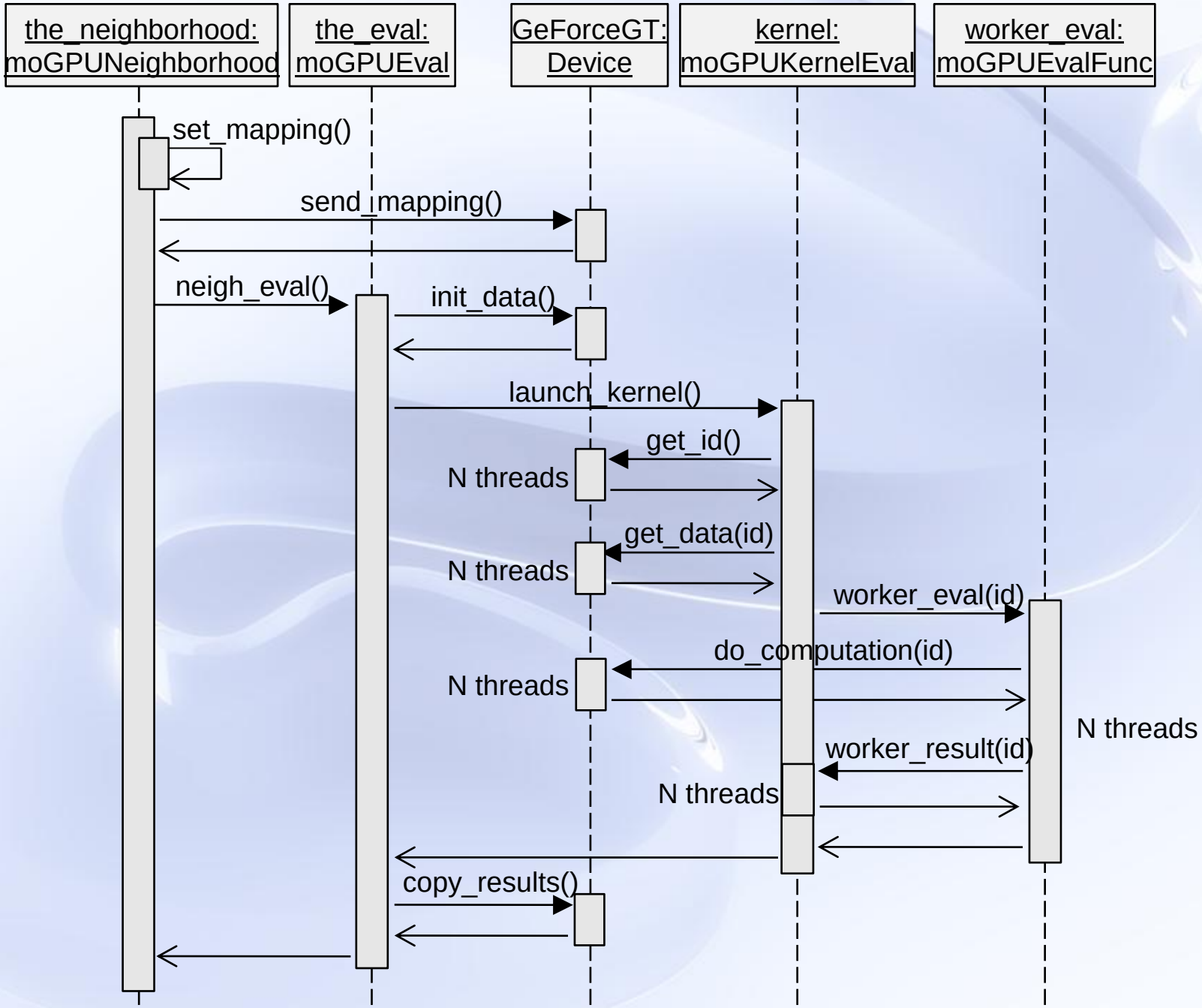
Instance	ParadisEO-MO			Optimized version			Perf. degradation	
	CPU	GPU	Acc.	CPU	GPU	Acc.	CPU	GPU
73-73	19.8	2.4	×**8.2**	17.1	1.3	×**13.1**	86%	54%
81-81	26	2.8	×**9.3**	22	1.6	×**13.8**	84%	57%
101-117	61	3.9	×**15.6**	51	2.2	×**23.2**	83%	56%
121-137	106	5.2	×**20.4**	91	2.9	×**31.3**	86%	56%
151-167	193	8.0	×**24.1**	134	4.1	×**32.7**	69%	51%
171-187	305	11.3	×**26.9**	208	5.6	×**37.1**	68%	49%
201-217	455	17.6	×**29.5**	343	8.2	×**41.8**	75%	46%

Actual Features

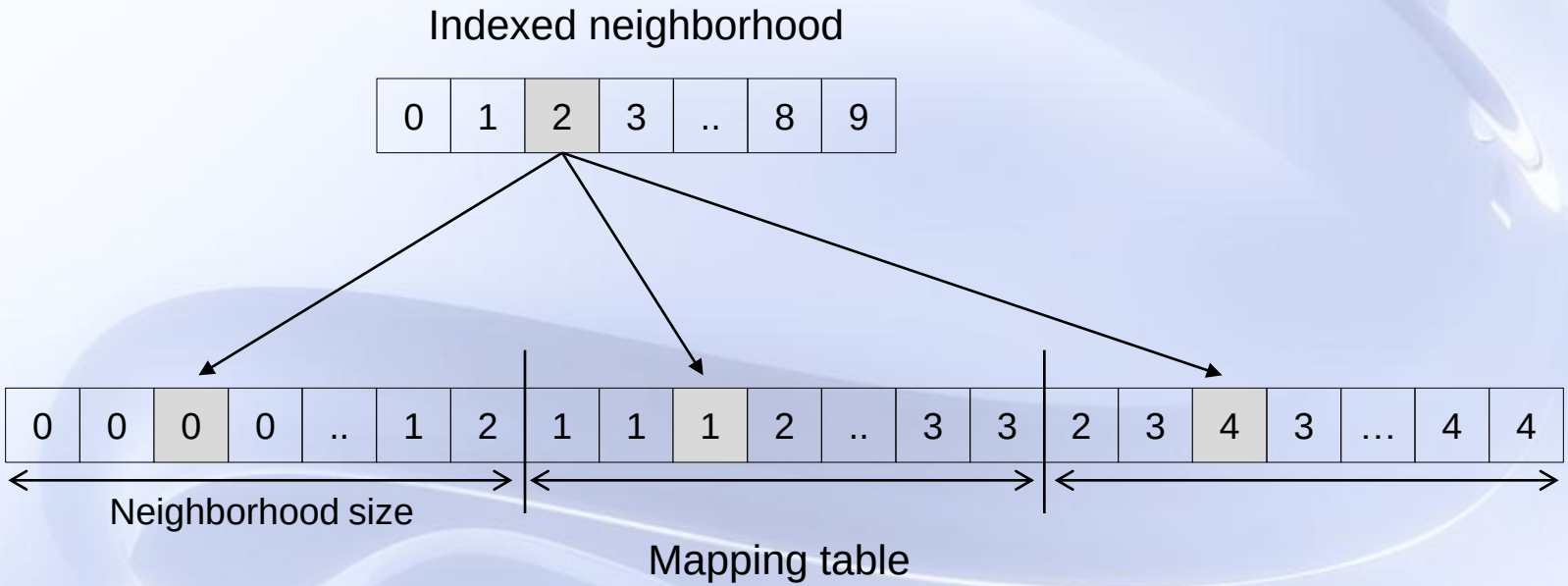
- First version will be released this month.
- Transparent use of GPU in local search algorithms.
- Flexibility and easiness of reuse at implementation.
- Support binary and permutation-based problems.

- http://paradiseo.gforge.inria.fr

THANK YOU FOR YOUR ATTENTION


the_neighborhood:
moGPUNeighborhood
the_eval:
moGPUEval
GeForceGT:
Device
kernel:
moGPUKernelEval
worker_eval:
moGPUEvalFunc
set_mapping()
send_mapping()
neigh_eval()
init_data()
launch_kernel()
get_id()
N threads
get_data(id)
N threads
worker_eval(id)
do_computation(id)
N threads
N threads
worker_result(id)
N threads
copy_results()

Mapping tables



- Pre-defined neighborhoods for binary and permutation representations.
- k-exchanges and k-Hamming distances neighborhoods.