INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Laboratoire d'Informatique Fondamentale de Lille

Parallel & Hybrid Metaheuristics

E-G. Talbi, S. Cahon, N. Melab

E-G. Talbi, « Parallel Combinatorial Optimization », John Wiley & Sons, 2006.

Parallel Cooperative Optimization Research Group

Paradiseo

Motivations

- High-dimensional and complex optimization problems in many areas of industrial concern
  → Telecommunication, genomics, transport, …
- Problems of increasing size (combinatorial explosion)
  - Getting near-optimal solutions in a tractable time
- Issues
  - Hybridization features
  - Large scale parallelism (GRID Computing)
- Objectives
  - Speedup the search, Improve quality of solutions, Improve robustness, solving large problems

Outline

- Design issues
  - Parallelization
    - Hierarchical classification
  - Hybridization
    - Hierarchical vs. flat classifications
- Implementation issues
  - Hardware platforms
  - Programming models, …

Parallel metaheuristics

E. Alba, E-G. Talbi, G. Luque, N. Melab, « ***Metaheuristics and parallelism*** », in Parallel Metaheuristics, pp. 79-104, Edited by E. Alba, John Wiley & Sons, 2005.

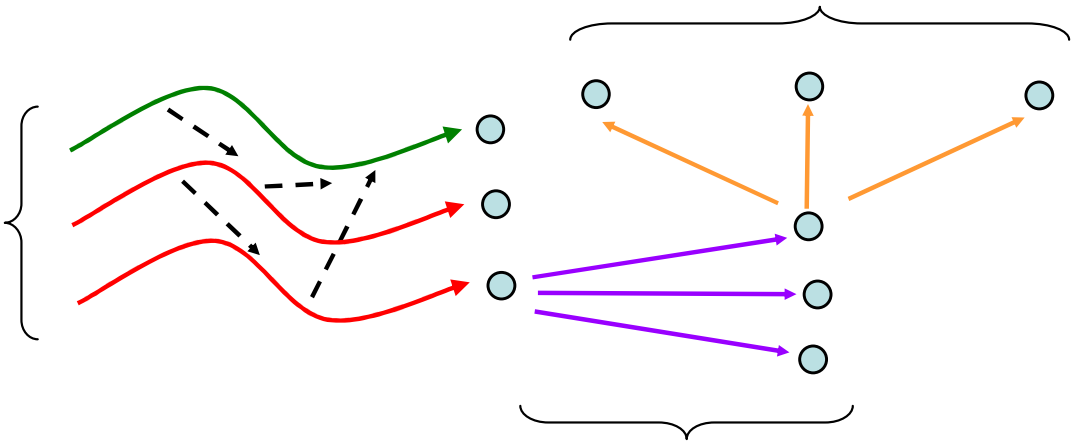
An unifying view of three hierarchical levels

- For both the population-based and solution-based metaheuristics,
  - The deployment of concurrent independent/cooperative metaheuristics
  - The parallelization of a single step of the metaheuristic (based on distribution of the handled solutions)
  - The parallelization of the processing of a single solution

Design of several levels of parallelization/hybridization

Processing of a single solution
(Objective / Data partitioning)

Independent walks,
Multi-start model,
Hybridization/Cooperation
of metaheuristics

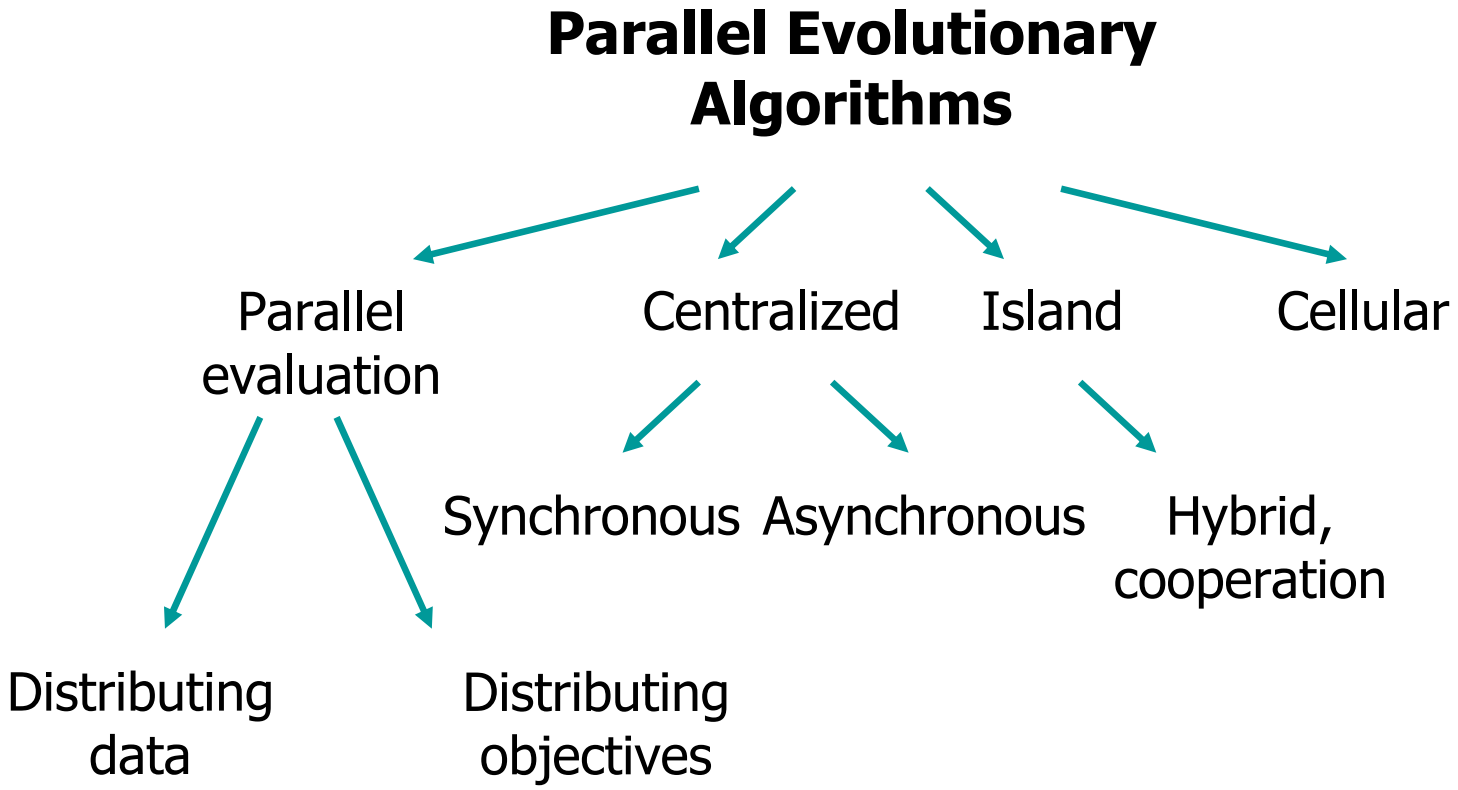


Parallel evaluation of
the neighborhood/population

Scalability $|H| \times |P| \times |S|$

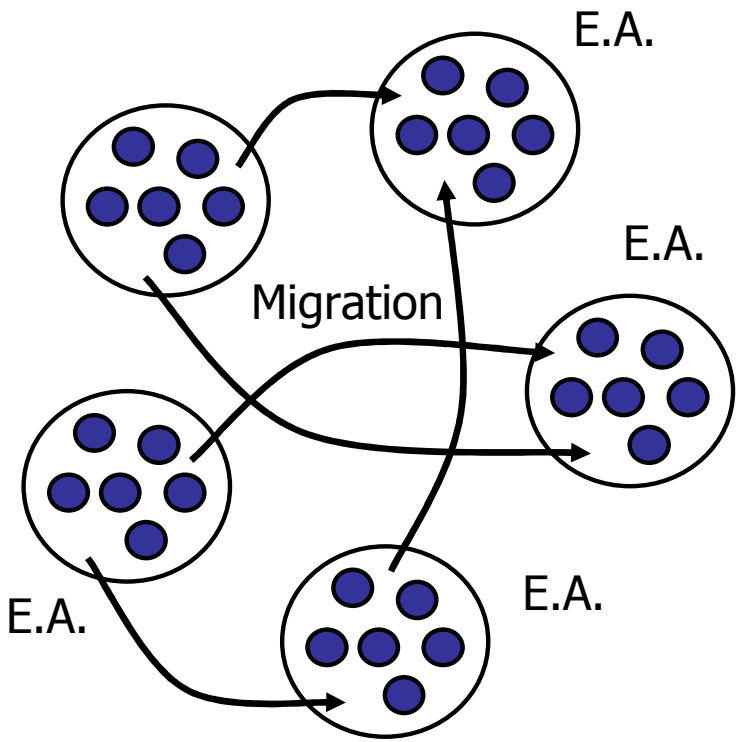
Heuristic — Population / Neighborhood — Solution

Parallel models of Evolutionary Algorithms



- The cooperative island model
- The parallelization of the evaluation/transformation step
- The parallelization of the process of evaluation

The cooperative island model



- A hybridization scheme !
  - Distribution of the population in a set of islands in which semi-isolated EAs are executed
  - Sparse individual exchanges are performed among these islands with the goal of introducing more diversity into the target populations (thus avoiding falling in local optima!)
  - Improvement of robustness

Some references

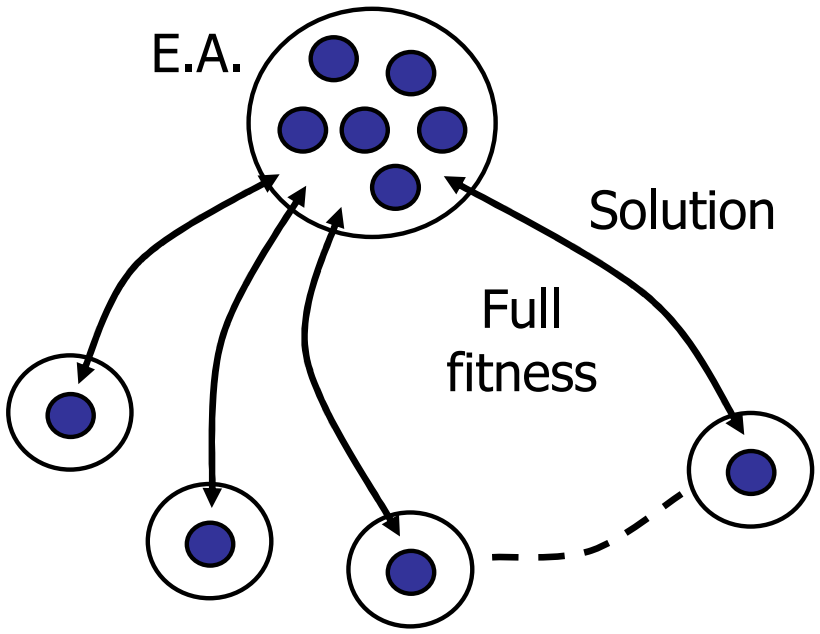
- E-G. Talbi, V. Bachelet, «**COSEARCH: A parallel cooperative metaheuristic**», Journal of Mathematical Modeling and Algorithms JMMA, 2006.

- L. Vermeulen-Jourdan, C. Dhaenens, E-G. Talbi,"**Linkage disequilibrium study with a parallel adaptive GA**", International Journal of Foundation of Computer Science IJFCS, Vol.16, No.2, pp.241-260, Apr 2005.

- N. Jozefowiez, F. Semet, E-G. Talbi, «**Parallel and hybrid models for multi-objective optimization»,**Parallel Problem Solving from Nature – PPSN VII, LNCS No.2439, Springer Verlag, pp.271-280, Granada, Spain, Sept 2002.

- N. Jozefowiez, F. Semet, E-G. Talbi, «**Enhancements of NSGA II and its application to the vehicle routing problem with route balancing**», 7th International Conference EA'2005 (Artificial Evolution), LNCS No.3871, Springer, pp.131-142, Lille, France, Oct 2005.

The parallel evaluation/transformation step



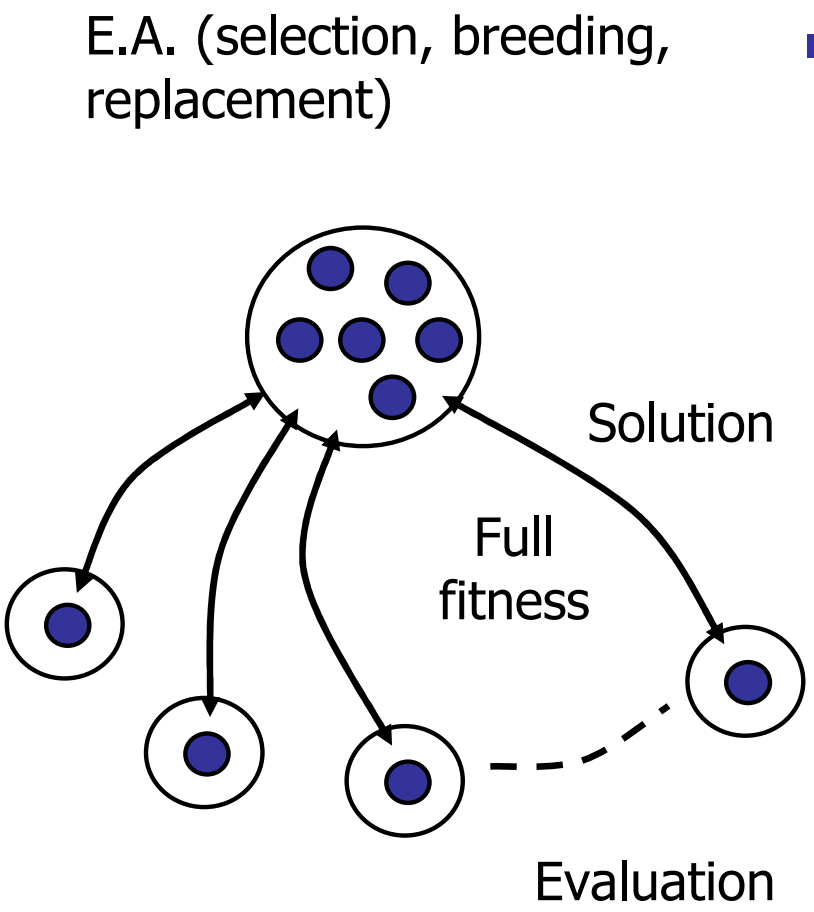
- The evaluation of the offspring is generally the most computationally step of the EA.
- Parallelization bases on the distribution of the evolving population
- It doesn't change the behavior of the E.A. but speeds up the search

Some implementations

- E-G. Talbi, H. Meunier, «**Hierarchical parallel approach for GSM mobile network design**», Journal of Parallel and Distributed Computing (JDPC), Vol.66, No.2, pp.274-290, Feb 2006.

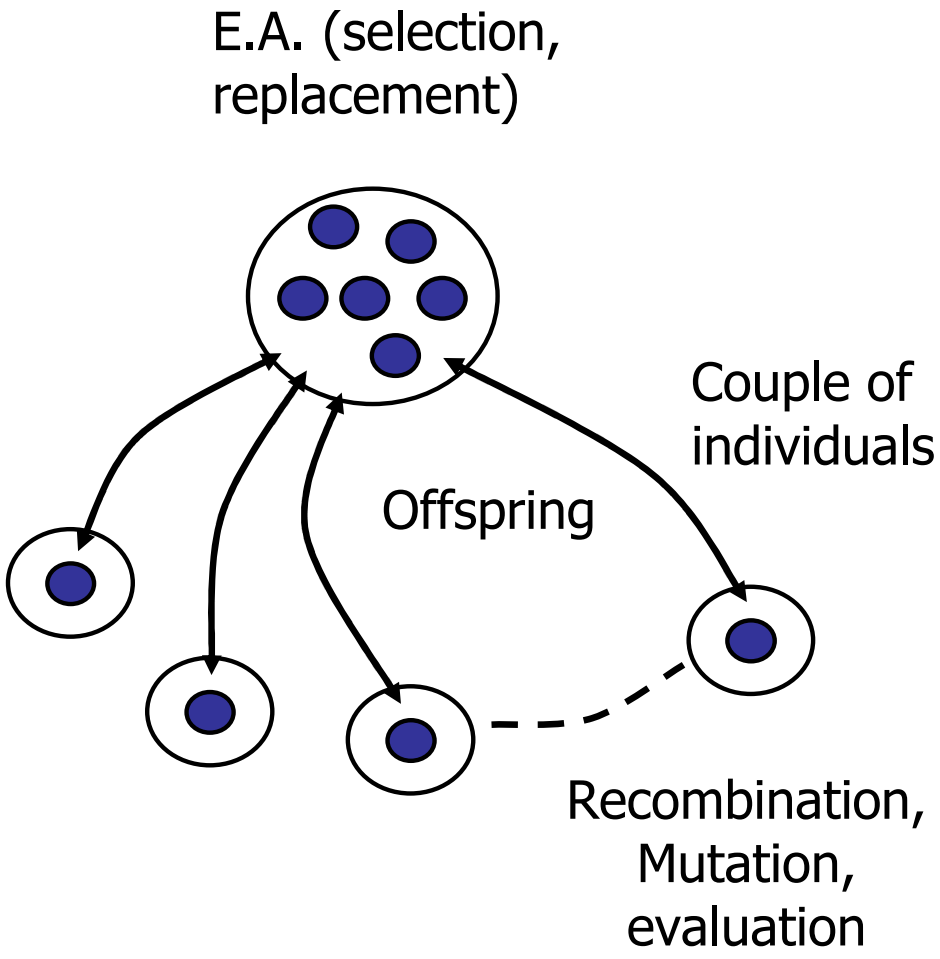
- E-G. Talbi, O. Roux, C. Fonlupt, D. Robillard, «**Parallel ant colonies for the quadratic assignment problem**», Future Generation Computer Systems, Vol.17, No.4, pp.441-449, Jan 2001.

Two schemes: The synchronous and asynchronous parallel models



- Synchronous E.A.
  - All the solutions are scattered between the working nodes. The computing fitness values are gathered by the E.A. that resumes the process of evolution.
  - Rather intended for the parallelization of homogeneous evaluations on homogeneous computational resources

Two schemes: The synchronous and asynchronous parallel models



- Asynchronous Steady-State E.A.
  - Transformation/evaluation step is desynchronized from the other steps of the E.A. Each worker is given a couple of parents, applies the variation operators, the evaluation of the offspring, sends back the resulting individual(s), is given a new couple of solutions and so on …

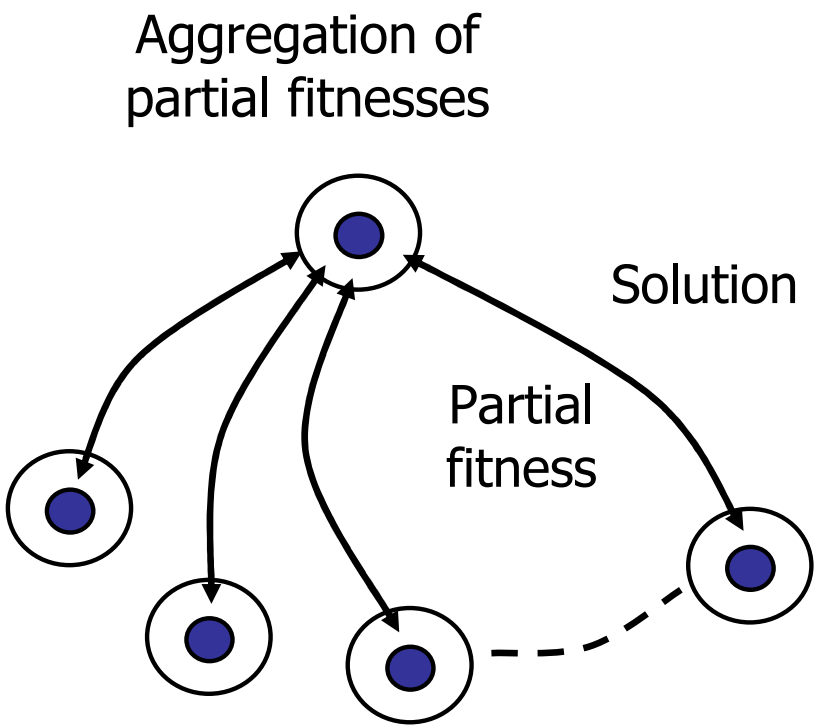
Some implementations

- E-G. Talbi, S. Cahon, N. Melab, «**Designing cellular networks using a parallel hybrid metaheuristic**», Journal of Computer Communications, 2006.

- P.Bessière, J-M.Ahactzin, E-G.Talbi, E.Mazer,**"The Ariadne's clew algorithm: global planning with local methods",** in Algorithmic Foundations of Robotics, Edited by K.Goldberg, D.Halperin, J-C.Latombe and R. Wilson, A K Peters Publishers, Wellesley, Massachusetts, pp.39-47, USA, 1995.

- A. J. Nebro, F. Luna, E-G. Talbi, E. Alba, "**Parallel multi-objective optimization**", in Parallel Metaheuristics, Edited by E. Alba, pp. 371-394, John Wiley & Sons, 2005.

The parallelization of the evaluation of a single solution

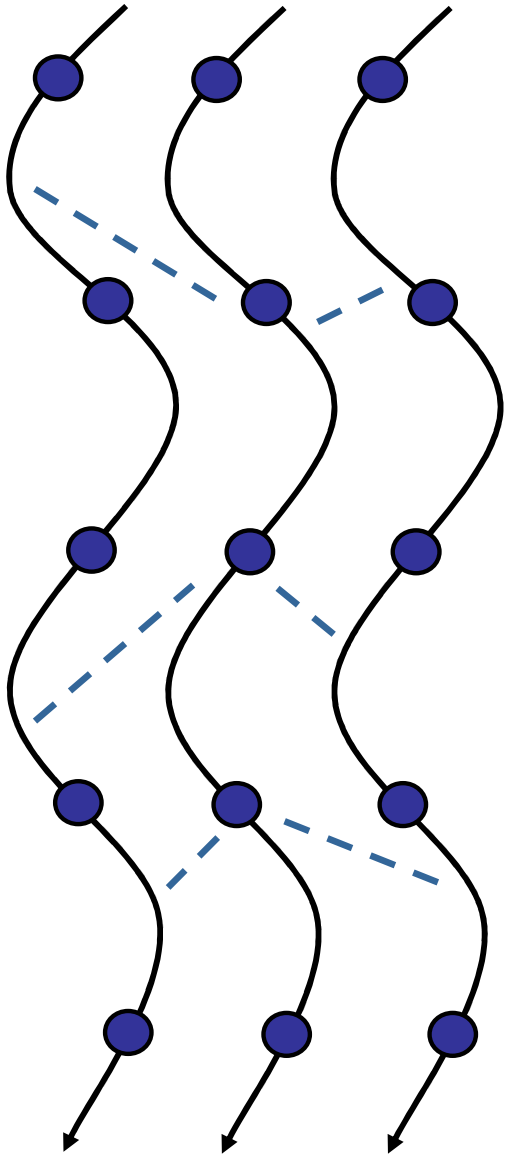


- The evaluation of a single solution may be parallelized !
  - Partitioning data, objectives.
- To be used in conjunction with the parallelization of the population to enhance scalability
- Most relevant with the synchronous parallel evaluation step

Parallel models of solution-based metaheuristics

- Parallel multi-start model
- Parallel moves model
- Move acceleration model

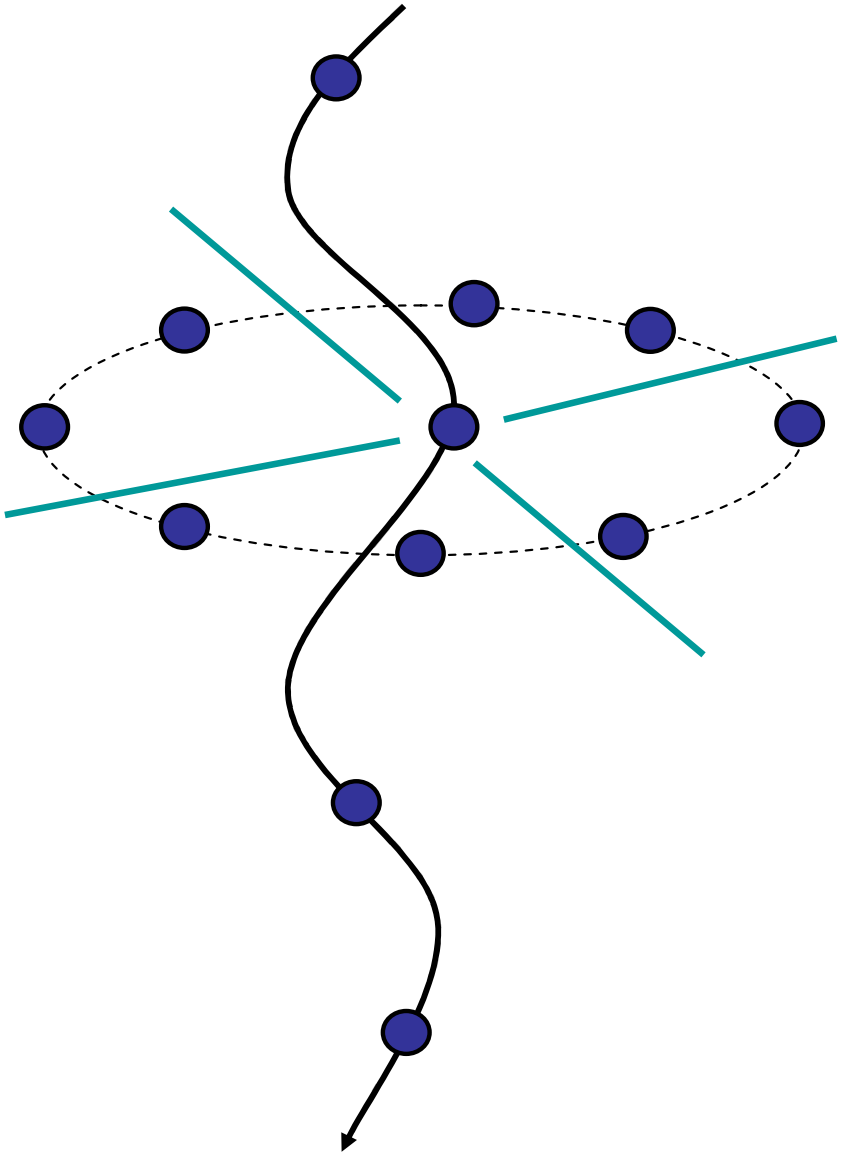
The multi-start model



- It consists in concurrently launching several searches for computing better and robust solutions.
- Searches may be homogeneous/heterogeneous, independent/cooperative, may start from the same or different solution(s), etc.

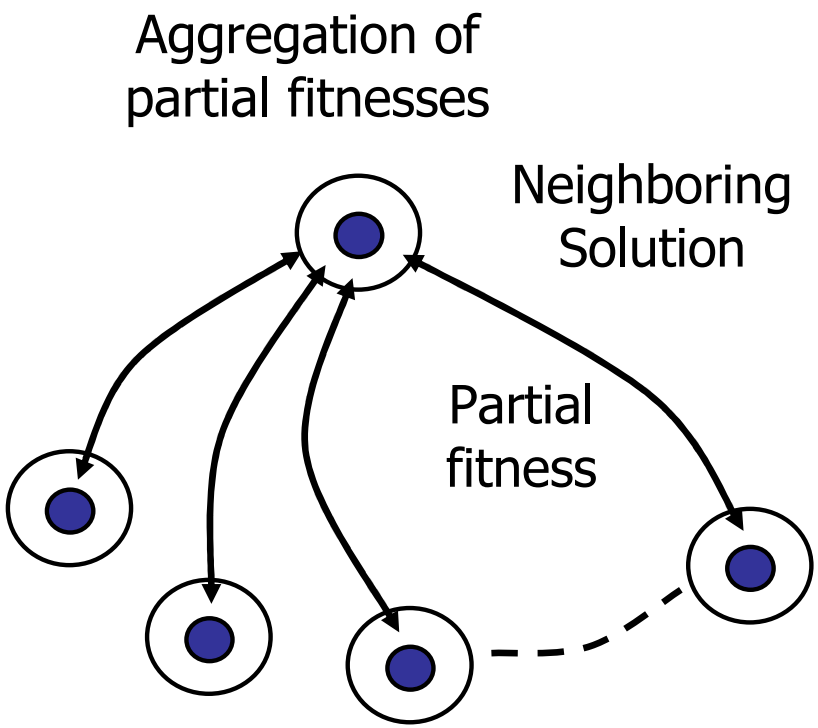
E-G. Talbi, Z. Hafidi, J-M.Geib, **"A parallel adaptive tabu search approach",** Parallel Computing, Vol.24, No.14, pp.2003-2019, Dec 1998.

The parallel moves model



- It does not modify the behavior of the search but speeds-up the search.
- A set of working nodes process a same initial solution but explore different subsets of neighboring candidates.
- The master node then applies a selection strategy to replace the current solution

Move acceleration model



- The quality of each move is evaluated in a parallel centralized way
- The function can be viewed as an aggregation of a certain number of partial functions
- The same semantic as the parallelization of the evaluation of one solution in the E.A.

Hybrid metaheuristics

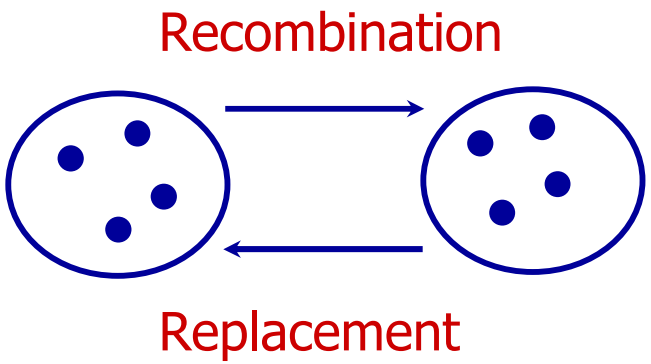
E-G. Talbi, «**A taxonomy of hybrid metaheuristics**», Journal of Heuristics, Vol.8, No 2, pp.541-564, Sept 2002.

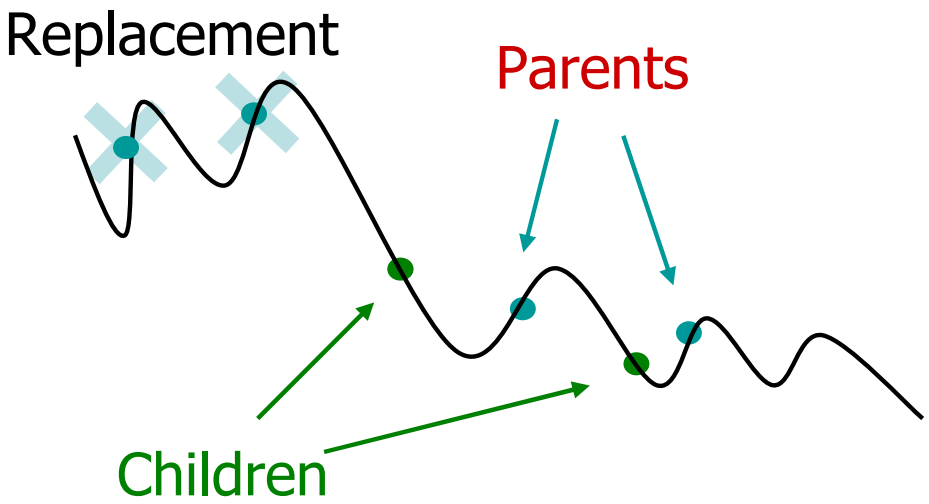
Motivations

- Considerable interest in recent years (combinatorial optimization, data mining, ...).
- Best found solutions for academic and real applications are obtained by hybrid algorithms (QAP, JSP, TSP, ...).
- A wide variety of hybrid approaches in the literature.
- Common terminology and classification mechanisms (Unifying view).
- Comparison of hybrid algorithms in a qualitative way.
- Areas in need of future work.

Population-based metaheuristics

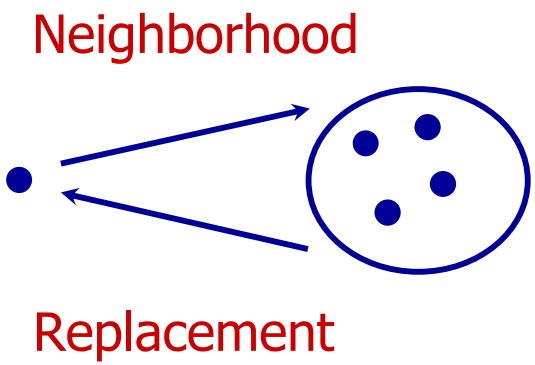
- "Improvement" of a population
- Exploration oriented
  - Solutions distributed in the search space
  - Recombination is explorative

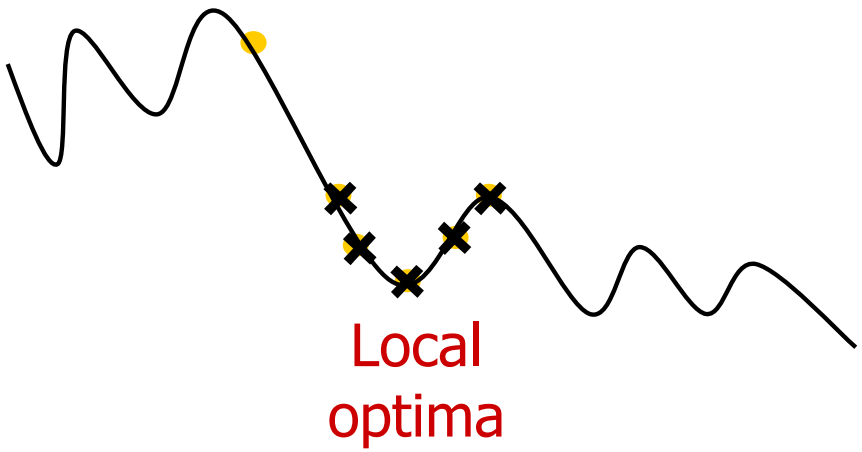




Solution-based metaheuristics

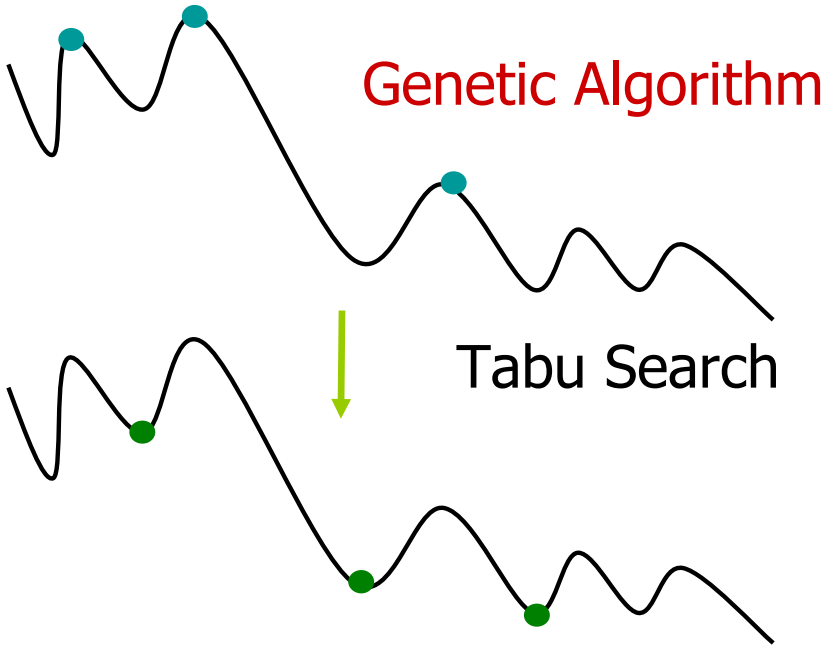
- “Improvement” of a solution
- Exploitation oriented
  - Based on the descent
  - Exploration of the neighborhood (intensification)



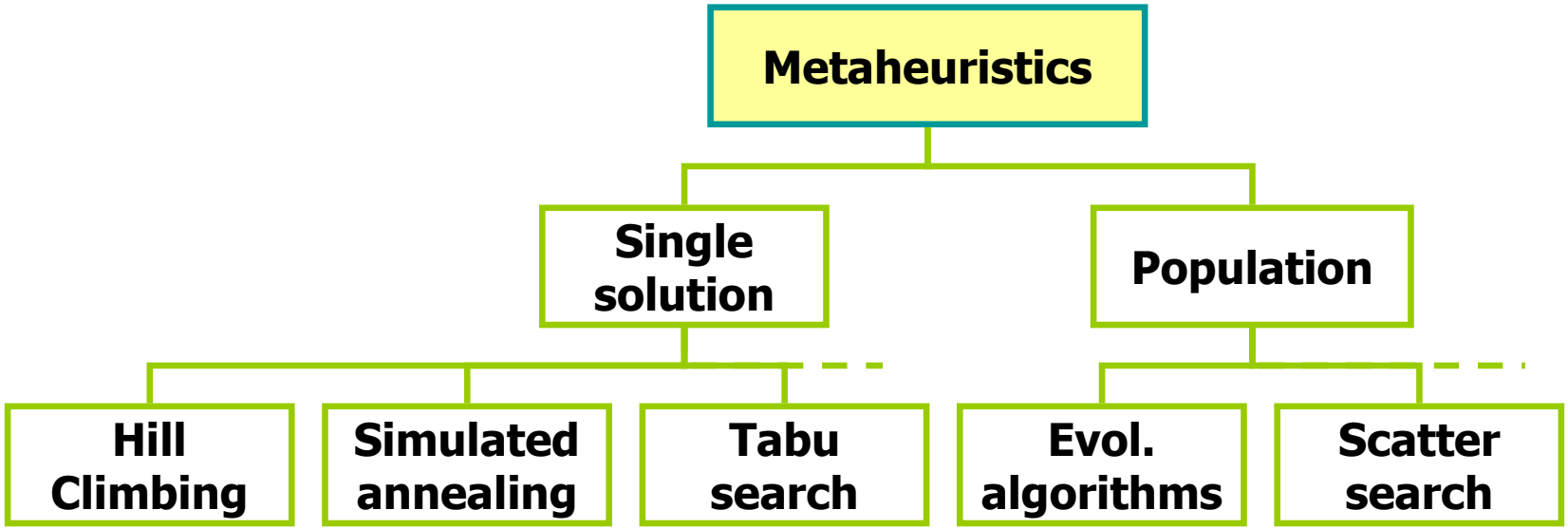


Hybridization

- Power of intensification and diversification
- To improve both quality of computed solutions and the robustness of solvers



Taxonomy of metaheuristics

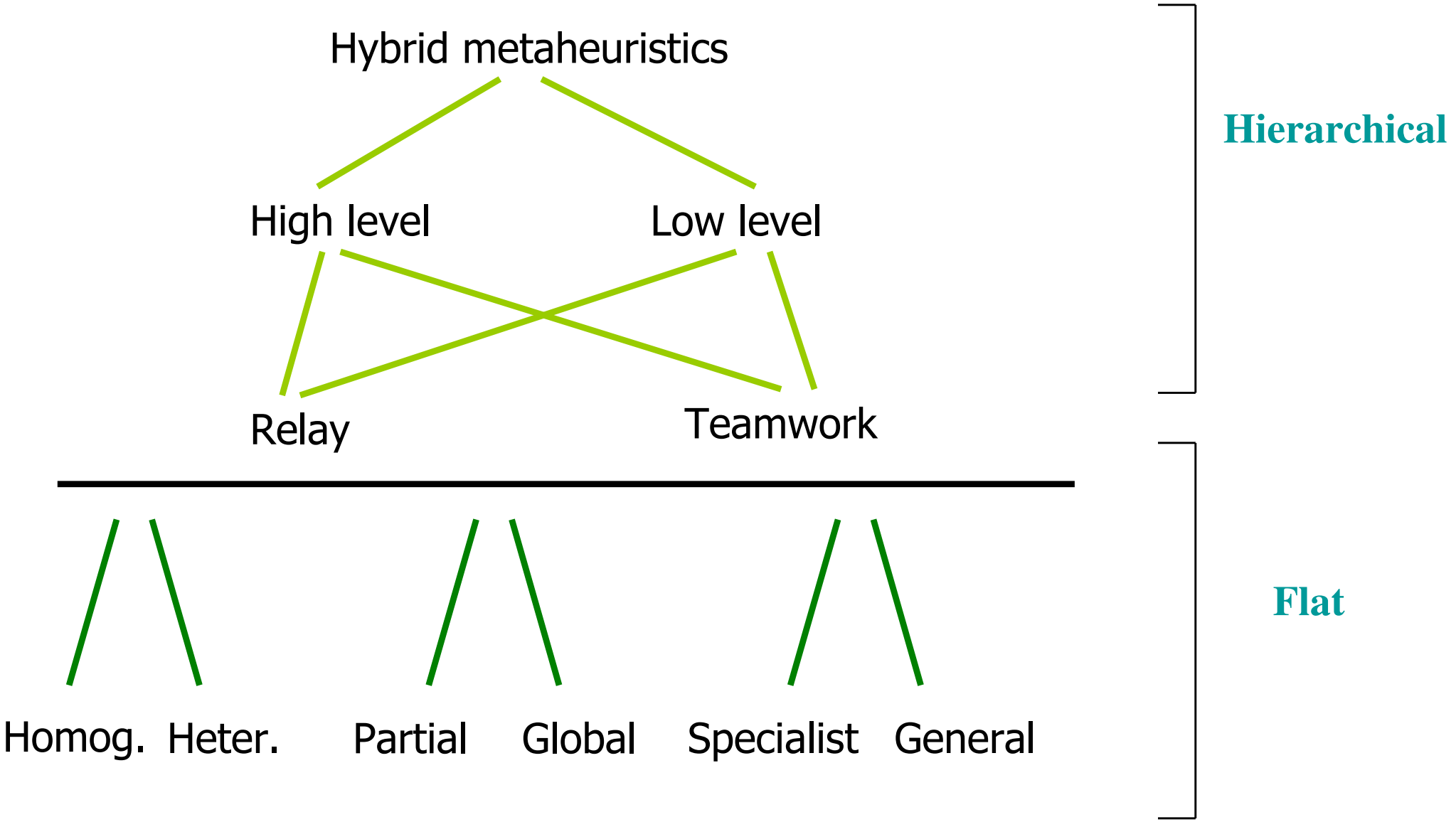


- Population-based metaheuristics: Evolutionary Algorithms, Scatter Search, Ant systems, …
- Solution-based metaheuristics: Hill Climbing, Simulated Annealing, Tabu Search, …

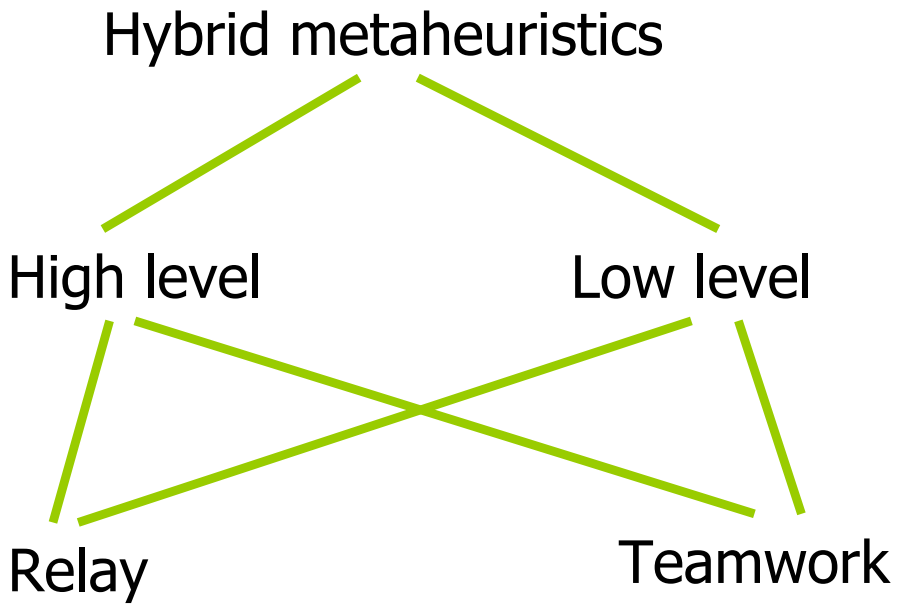
Design issues

- Two main independent but complementary classifications
  - Hierarchical
  - Flat

Design issues



Hierarchical classification

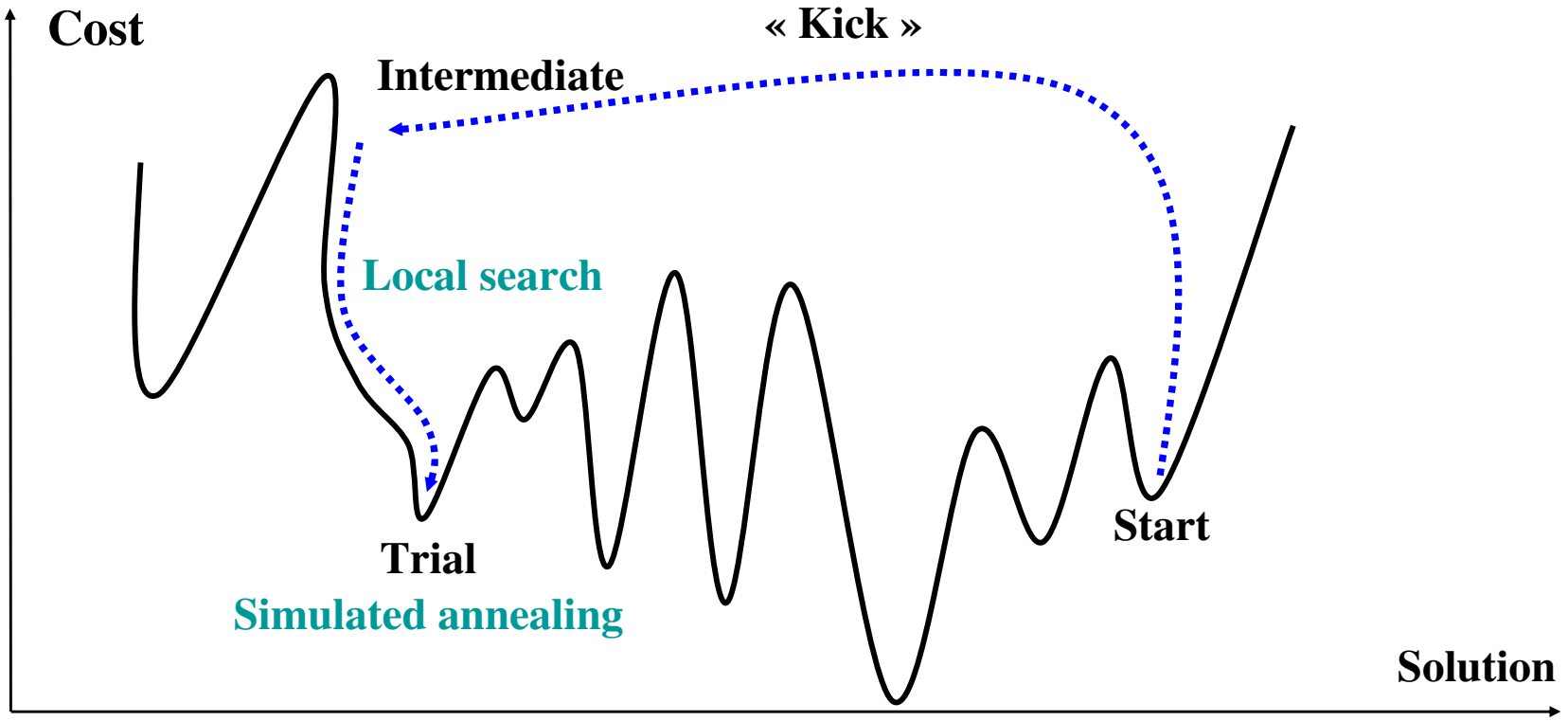


- Four classes
  - Low-level Relay Hybrid (LRH)
  - Low-level Teamwork Hybrid (LTH)
  - High-level Relay Hybrid (HRH)
  - High-level Teamwork Hybrid (HTH)

Low-level Relay Hybrid

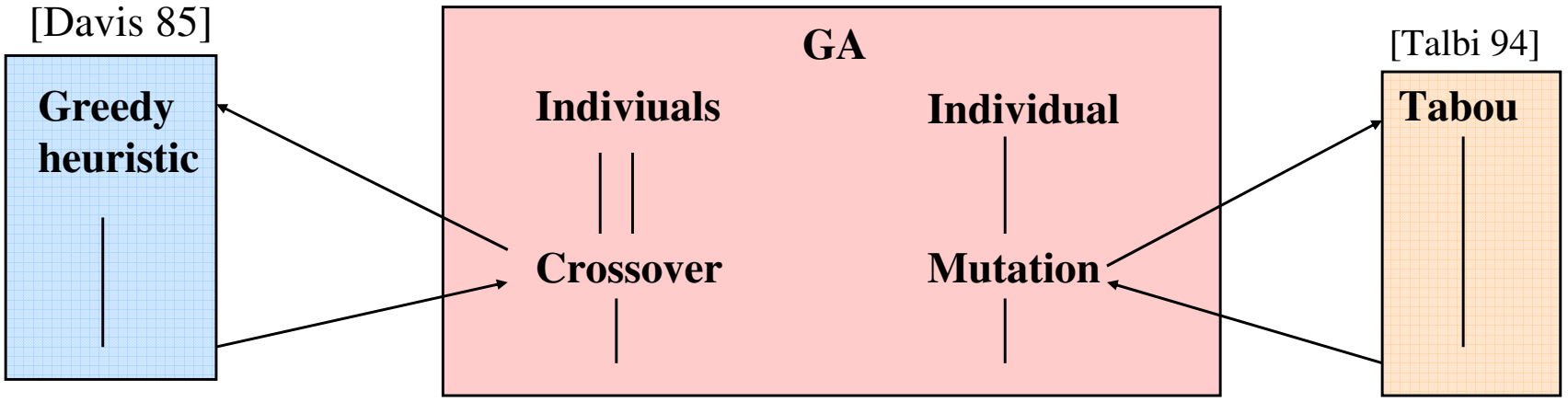
- It denotes algorithms in which a given metaheuristic is embedded into a single solution metaheuristic
- Example. Local search embedded in Simulated Annealing

[Martin et Otto 92]



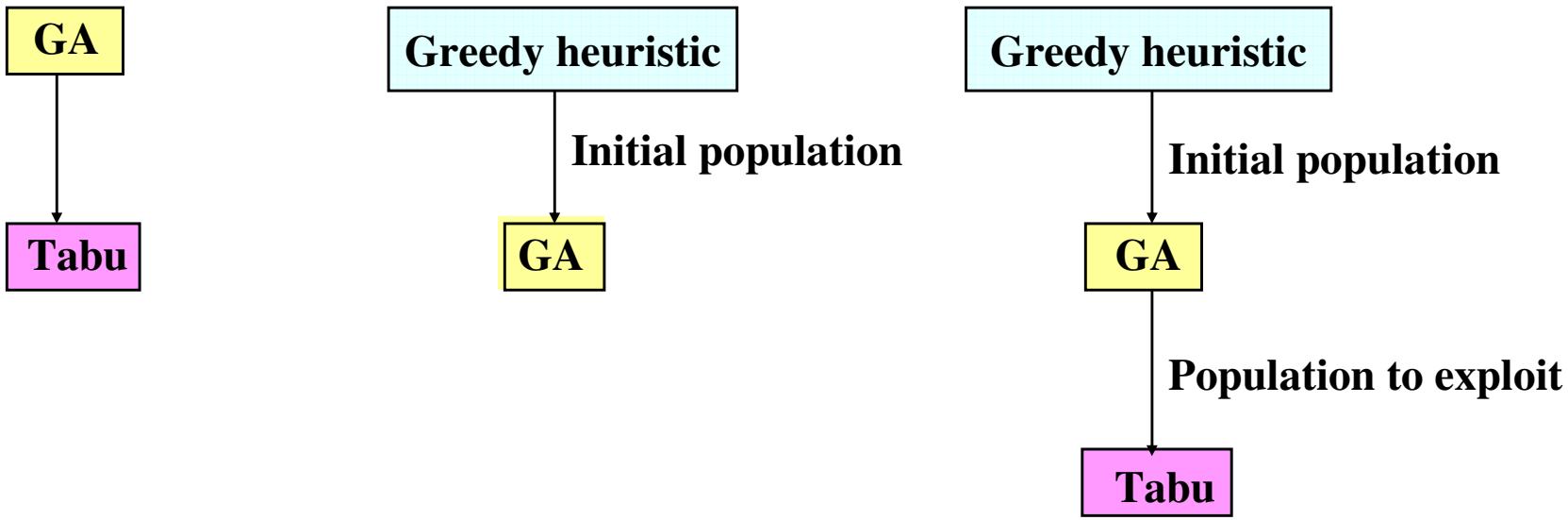
Low-level Teamwork Hybrid

- An optimization method is embedded in a population-based metaheuristic
- Examples
  - Greedy crossover [Grefenstette 85], local search for mutation [Fleurant 94, Chu 97] (lamarckian, baldwin, …) in GAs
  - Local search in Ant colonies [Taillard 97, Talbi et al. 99], Scatter search [Van Dat 97], and Genetic programming [O'Reilly 95].



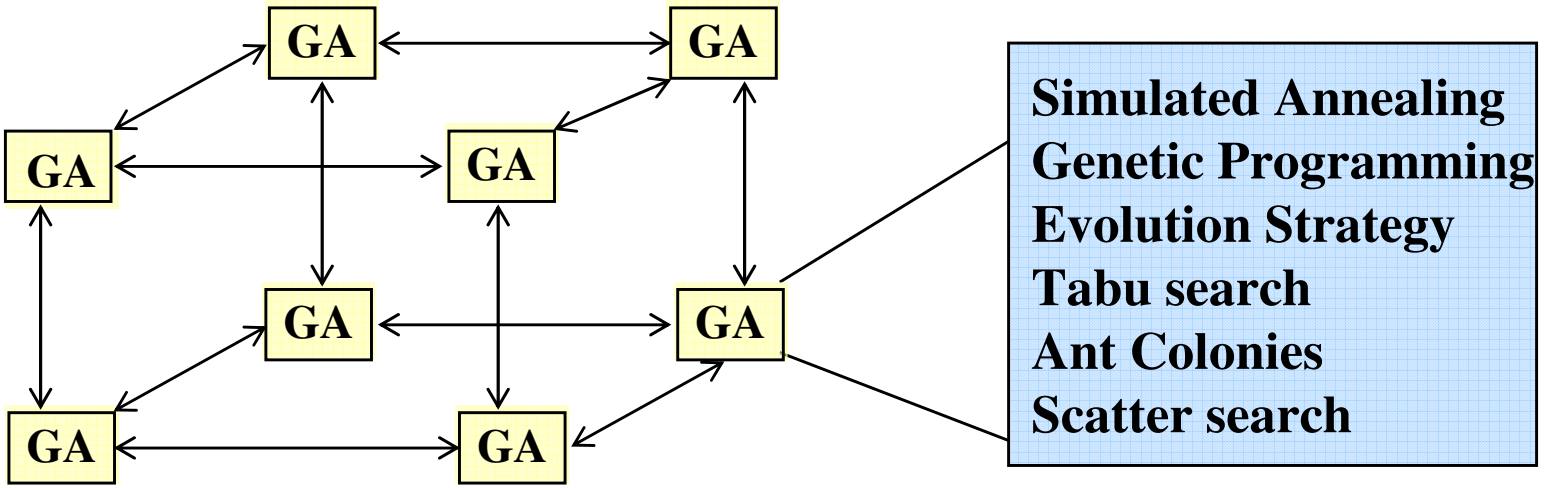
High-level Relay Hybrid

- Self-contained optimization methods are executed in a sequence (each using the output of the previous as its input, acting a pipeline fashion)
- Example
  - GA + Simulated annealing [Mahfoud 95]. GA + Tabu search [Talbi 94].
  - ES + Local search [Nissen 94]. Simulated annealing + GA [Lin 91]

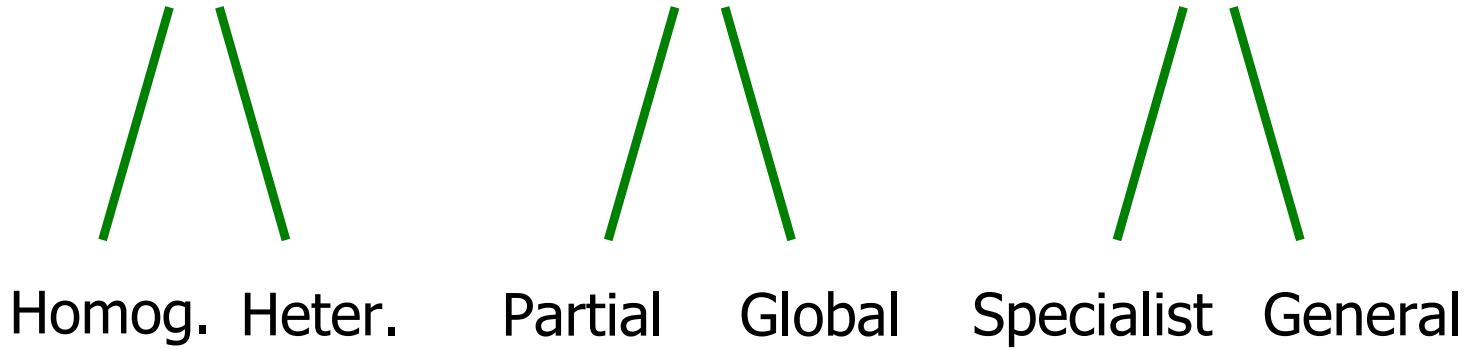


High-Level Teamwork Hybrid

- Parallel cooperating self-contained algorithms
- Examples
  - Island model for GAs [Tanese 87], Genetic programming [Koza 95], Evolution strategies [Voigt 90].
  - Tabu search [Rego 96], Simulated annealing [De Falco 95], Ant colonies [Mariano 98], Scatter search [Van Dat et al. 99].



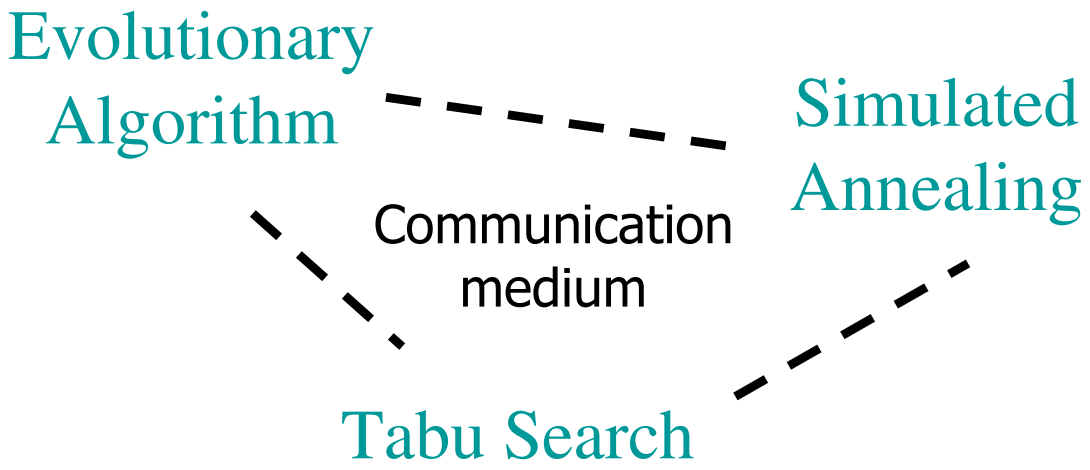
Flat classification



- Heterogeneous/Homogeneous
- Global/Partial
- Specialist/General

Homogeneous/Heterogeneous

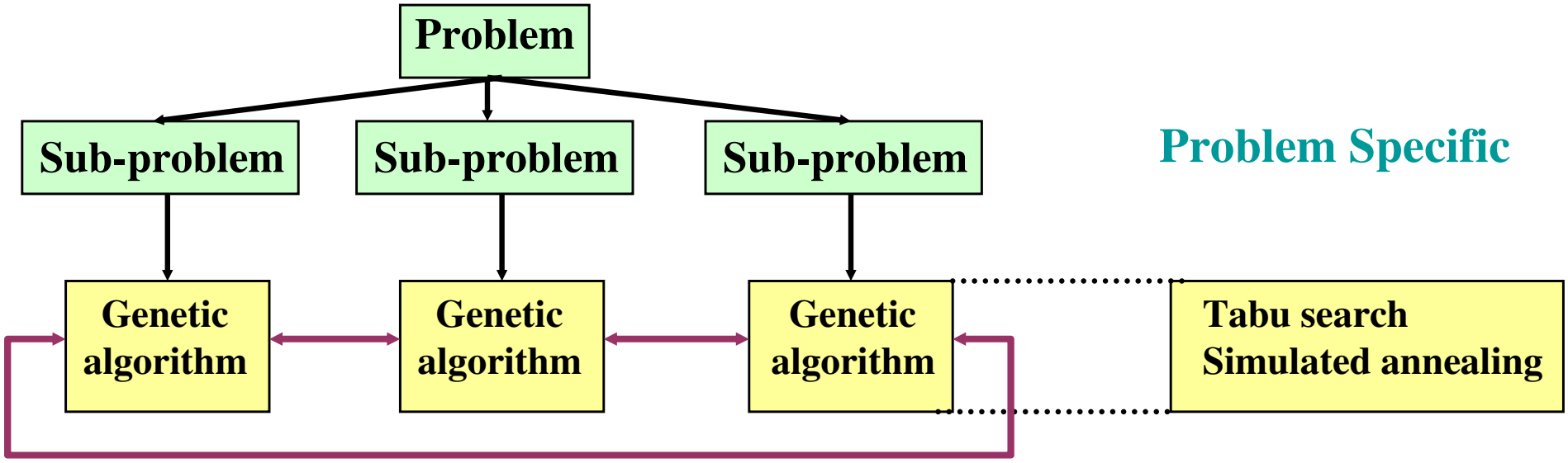
- In homogeneous hybrids, all the combined methods are the same metaheuristic (different metaheuristics are employed in heterogeneous hybridizations)
- Example



Several different metaheuristics cooperate and co-evolve some solutions

Global/Partial hybrids

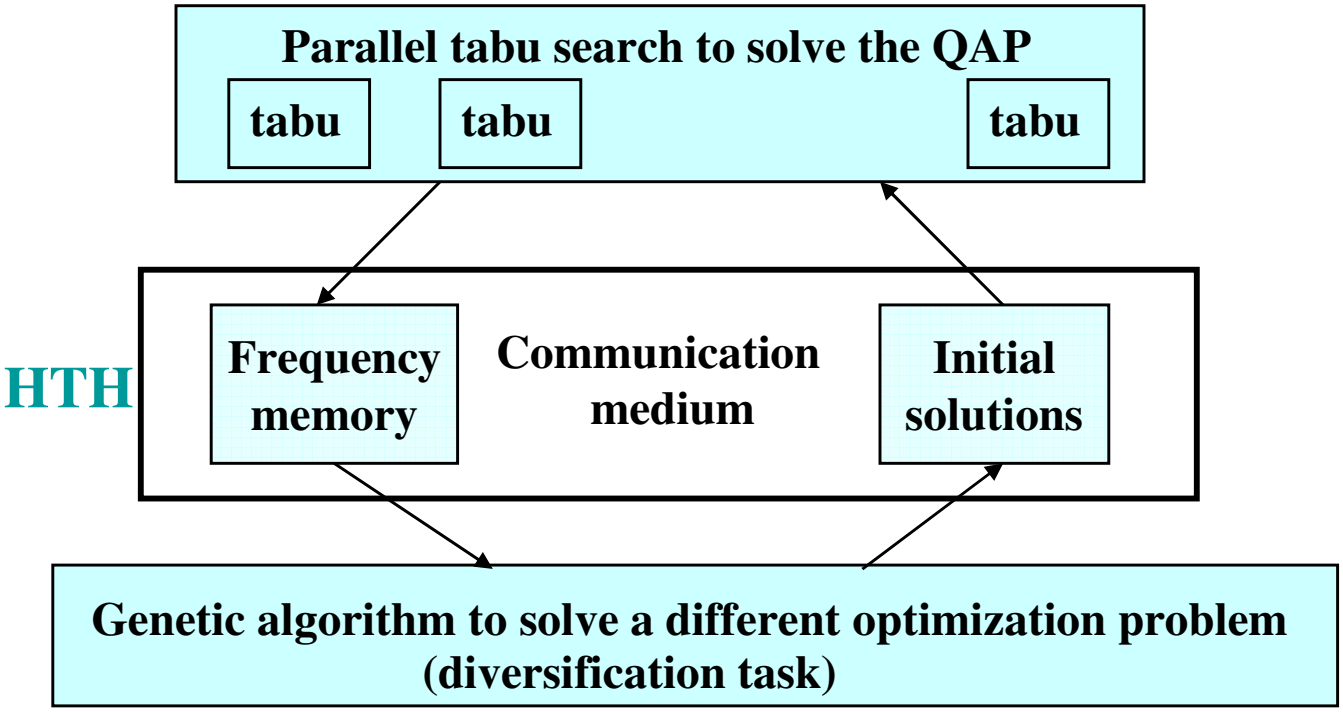
- Partial : Problem is decomposed in sub-problems. Each algorithm is dedicated to solve one sub-problem
- Example
  - HTH : Tabu search (Vehicle routing) [Taillard 93], Simulated annealing (placement of macro-cells) [Casoto et al. 86].
  - HTH : GA (Job-shop scheduling) [Husbands et al. 90].

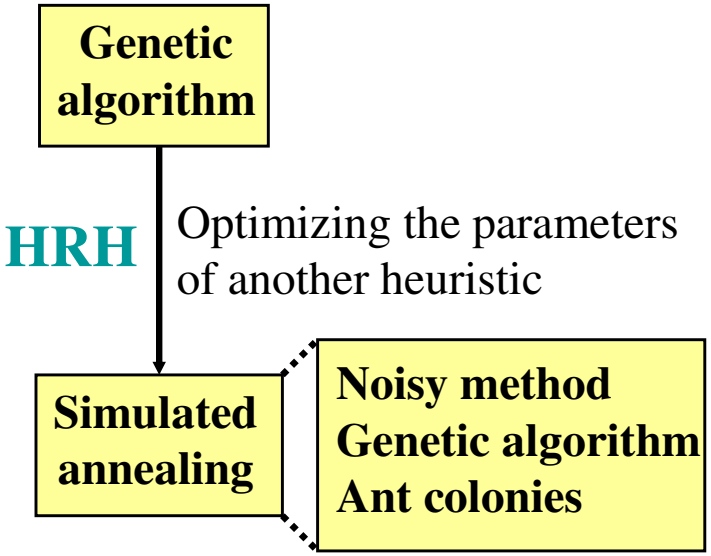


Synchronization : Build a global viable solution

Specialist/General hybrids

- Specialist: Algorithms which solve different problems
- Examples
  - **HTH** : GA + Tabu (QAP) [Talbi et al. 98]
  - **HRH** : GA + (SA | NM | GA | AC) [Krueger 93][Shahookar et al. 90][Abbattista et al. 95]

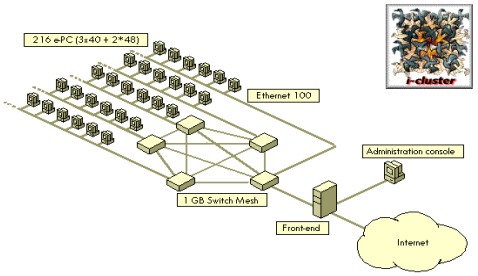




Implementation issues in the parallel/distributed execution of parallel/hybrid metaheuristics

N. Melab, M. Mezmaz and E-G. Talbi, “**Parallel Cooperative Meta-heuristics on the Computational Grid. A Case Study: The Bi-objective Flow-Shop Problem**”, Elsevier Science, Parallel Computing 32, Pages 643-659, 2006.

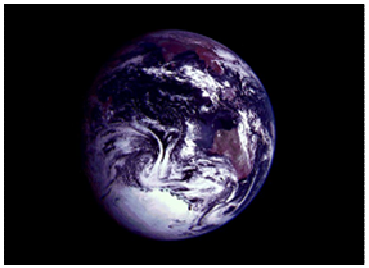
Different execution platforms



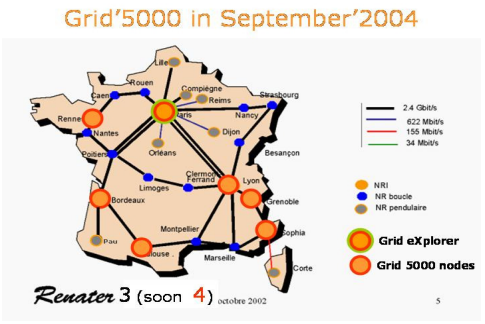
Cluster Computing

Parallel Computing

Grid Computing



Internet Computing



Main issues

- Homogeneity/Heterogeneity of the resources CPU
- Robustness/volatility of resources
- Dedicated/non-dedicated resources
- Communication
- Scalability

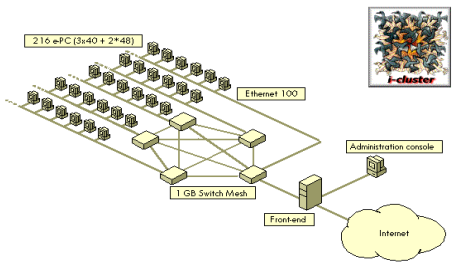
Solutions for the deployment of parallel hybrid metaheuristics

- Unpredictable CPU load, heterogeneity of the CPU resources, network latency, …
  → Asynchronism
- Volatility
  → Application checkpointing
- Scalability
  → Design of hierarchical models of parallelism/hybridization in conjunction
- Heterogeneity
  → Use of an underlying communication library (e.g. MPI/PVM)

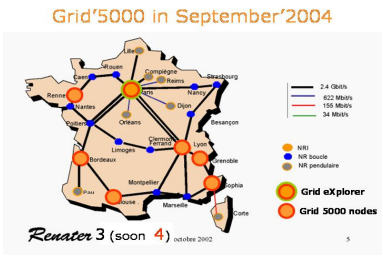
Two main axis

Grid Computing

High Performance Computing
Cluster Computing



Cluster'Lille



Grid'5000

High Throughput Computing/ Global Computing

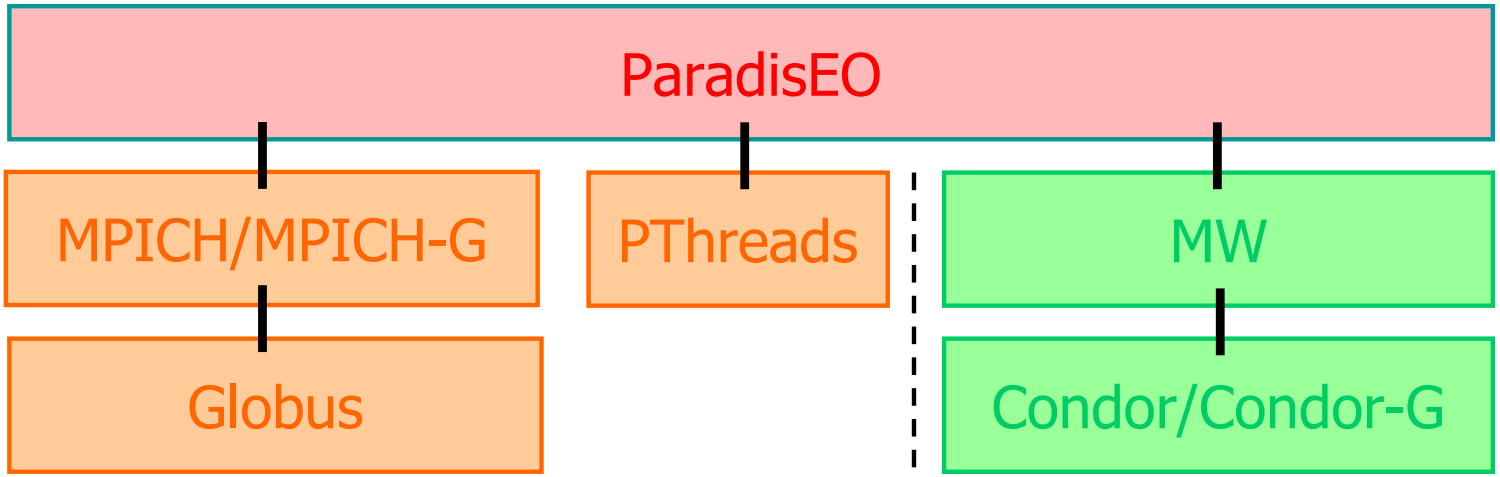
Non dedicated workstations

HPC vs. HTC

- High Performance Computing environments
  → To deliver a tremendous amount of power over a short period of time
- High Throughput Computing environments
  → To make an efficient use of the wasted computation time (universities, administrations, …)
  - Inexpensive, potentially very powerful but more difficult to program than traditional parallel computers

Architecture (implementation)

http://paradiseo.gforge.inria.fr



- Parallelism and distribution
  - Communication libraries (LAM-MPI, PVM) for message passing
    → Deployment on network of workstations (COWs, NOWs)
  - Support for multi-programming (Posix threads)
    → Machines with multiple processors and shared memory (SMPs)
  - Support of non dedicated parallel platforms (Condor/MW)
  - Transparence for the user

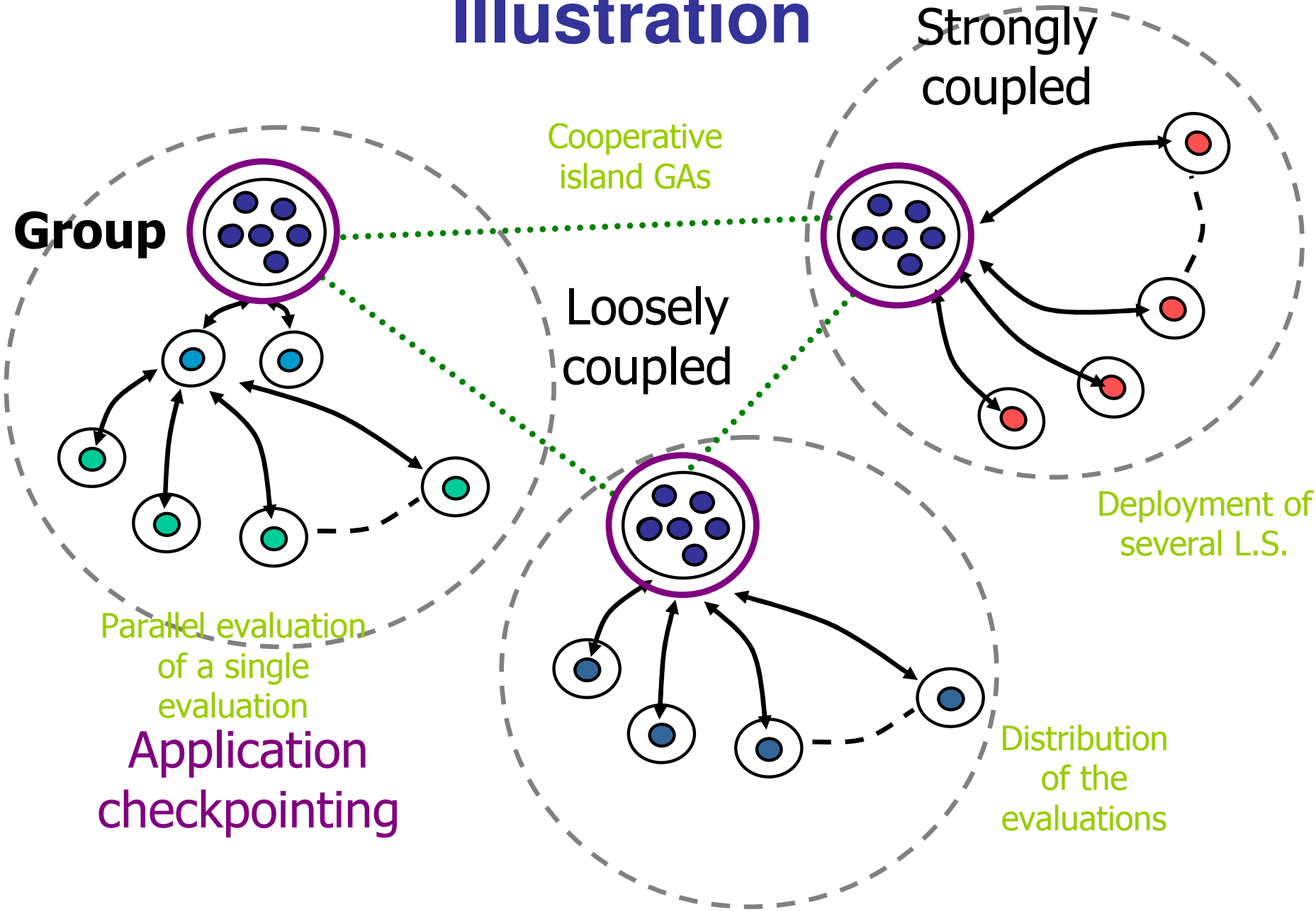
ParadisEO for High Performance Computing

S. Cahon, N. Melab and E-G. Talbi, «**ParadisEO: A Framework for the Reusable Design of Parallel and Distributed Metaheuristics**”, Journal of Heuristics, Vol.10, No.3, pp.357-380, May 2004.

Main features

- Intended to the deployment on safe (reliable) and dedicated computational resources
- Free location of the running metaheuristics, support of distributed scheduling to avoid bottlenecks and enhance parallel efficiency
- Automatic periodic checkpointing
- But not fault-tolerant !
- Underlying tools and libraries (MPICH and Posix threads)

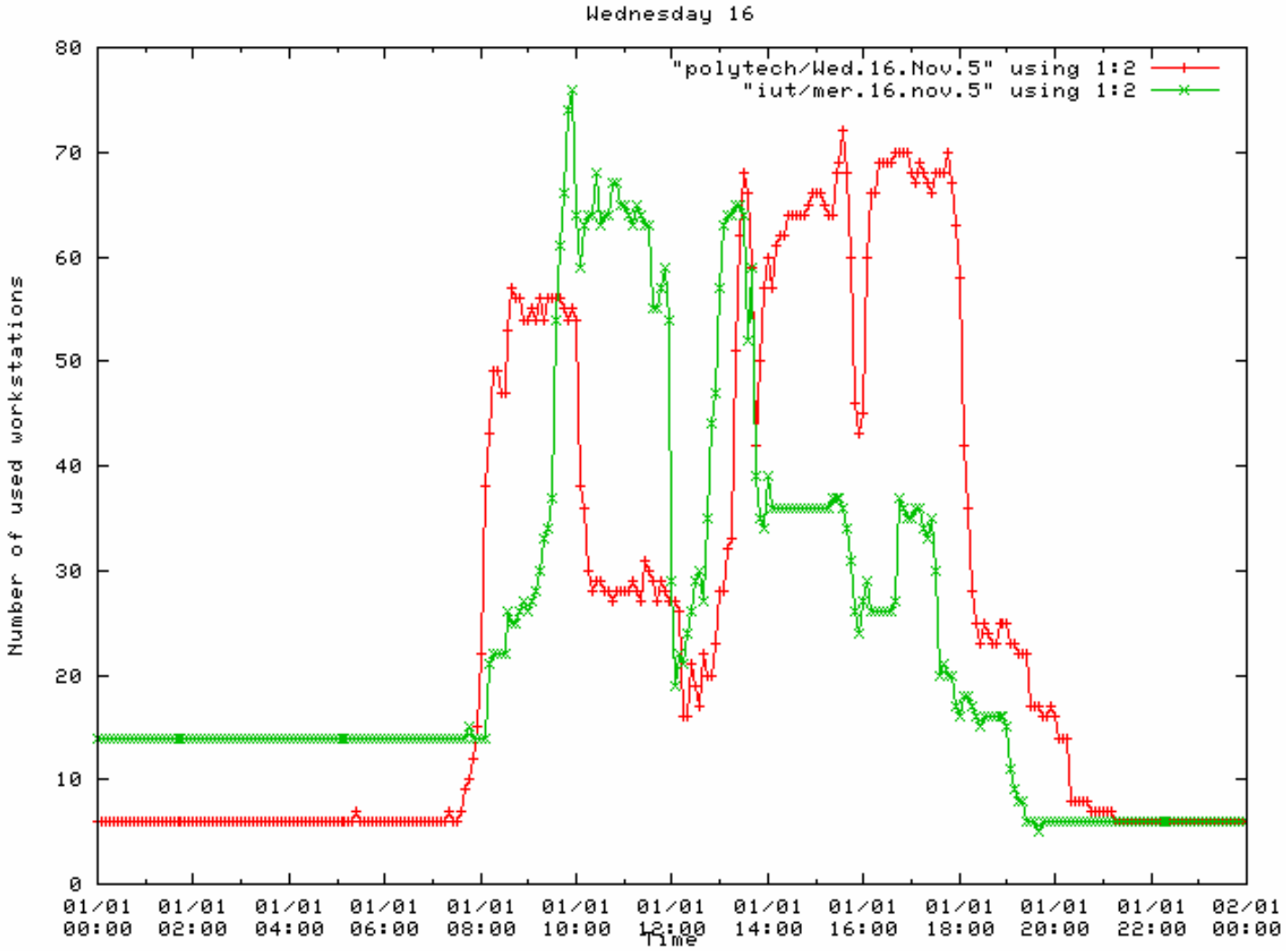
Illustration



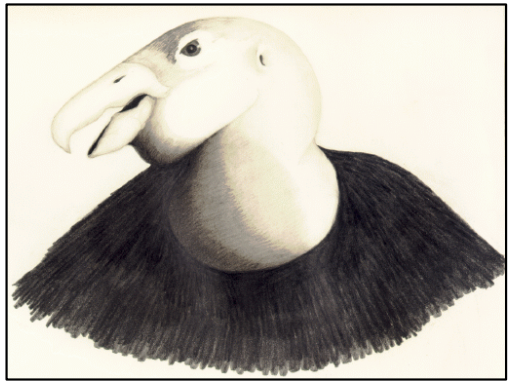
ParadisEO for High Throughput Computing

S. Cahon, N. Melab, E-G. Talbi, «**Building with ParadisEO reusable parallel and distributed evolutionary algorithms**», Parallel Computing, Vol.30, No.5-6, pp.677-697, June 2004.

Illustration with the education network at Polytech-Lille and IUT “A” during a full day

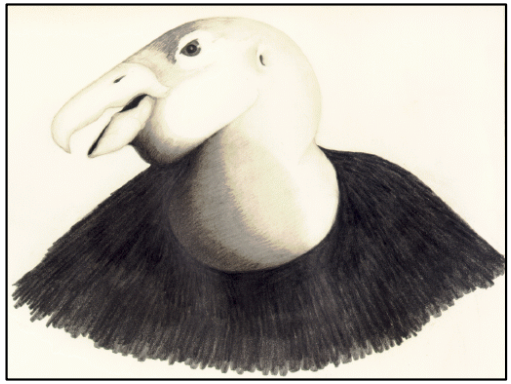


Why Condor ?



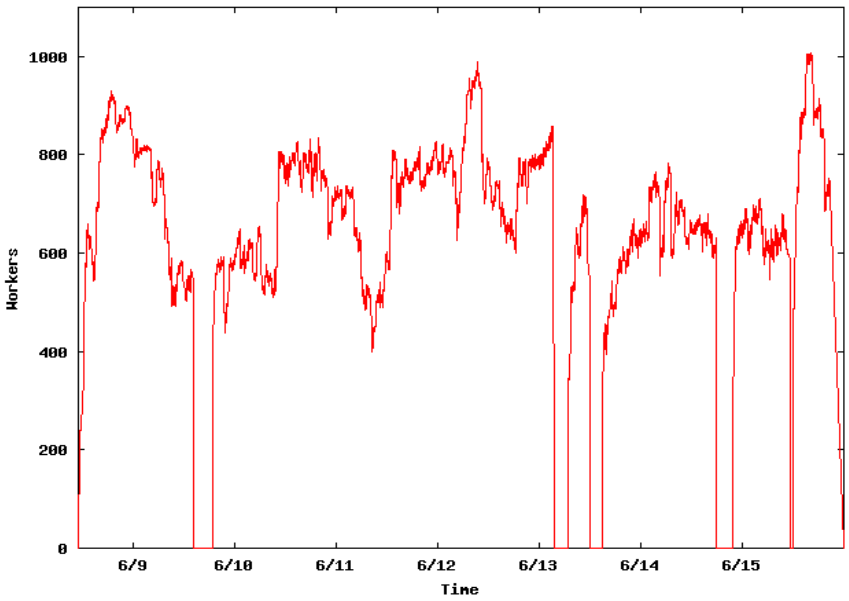
- Condor
  - Management and exploitation of distributed computational resources that are dynamic in space and time
- Main features
  - Matchmaking, Check-pointing, Security, …

Additional features



- **MW** for making a parallel master-worker style application that works in the distributed, opportunistic environment of Condor
- **Condor's "flocking"** technology allows multiple pools to work together in order to build Grid-style computing environments that cross administrative boundaries
- **Condor-G** is fully interoperable with resources managed by Globus

Success story. Solving NUG30

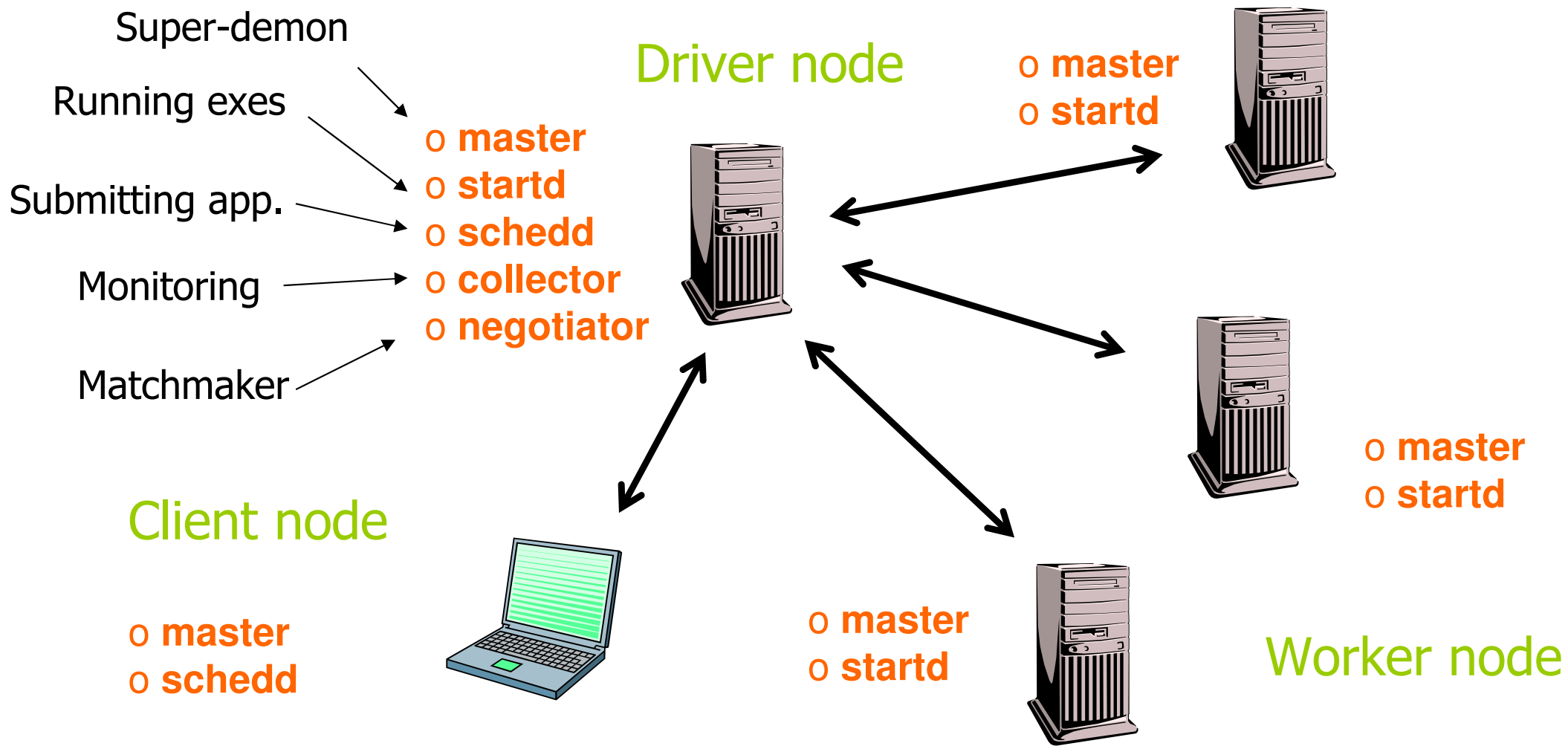


Number	Arch/OS	Location
414	Intel/Linux	Argonne
96	SGI/Irix	Argonne
1024	SGI/Irix	NCSA
16	Intel/Linux	NCSA
45	SGI/Irix	NCSA
246	Intel/Linux	Wisconsin
146	Intel/Solaris	Wisconsin
133	Sun/Solaris	Wisconsin
190	Intel/Linux	Georgia Tech
94	Intel/Solaris	Georgia Tech
54	Intel/Linux	Italy (INFN)
25	Intel/Linux	New Mexico
12	Sun/Solaris	Northwestern
5	Intel/Linux	Columbia U.
10	Sun/Solaris	Columbia U.

The MetaNEOS Project

http://www-unix.mcs.anl.gov/metaneos/index.html

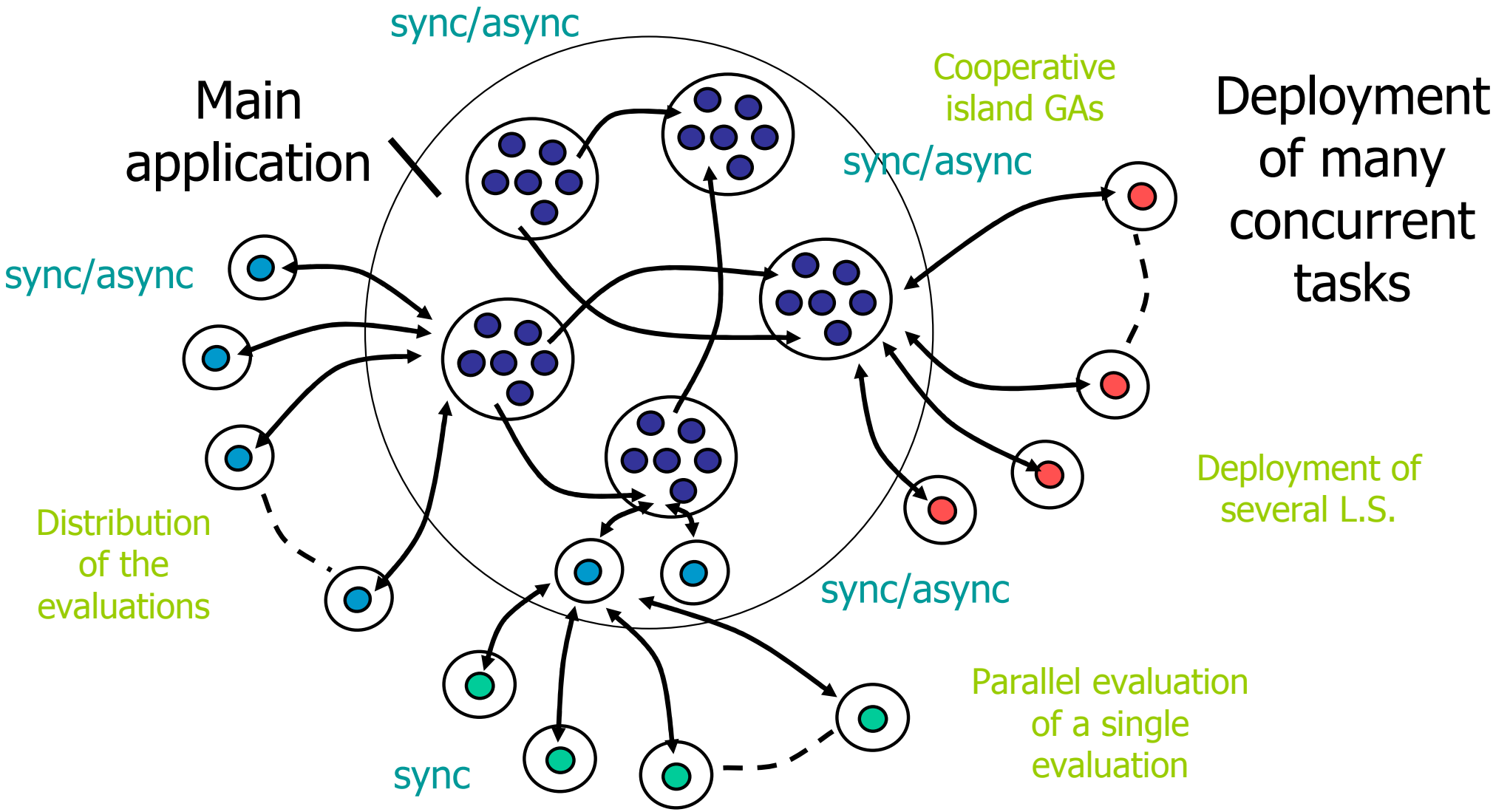
Condor in practice …



Setting policies (ClassAd)

```
NonCondorLoadAvg     = (LoadAvg - CondorLoadAvg)
BackgroundLoad       = 0.3
HighLoad             = 0.5
StartIdleTime        = 15 * $(MINUTE)
ContinueIdleTime     =  5 * $(MINUTE)
MaxSuspendTime       = 10 * $(MINUTE)
MaxVacateTime        = 10 * $(MINUTE)
KeyboardBusy         = (KeyboardIdle < $(MINUTE))
ConsoleBusy          = (ConsoleIdle  < $(MINUTE))
CPUIdle              = ($(NonCondorLoadAvg) <= $(BackgroundLoad))
CPUBusy              = ($(NonCondorLoadAvg) >= $(HighLoad))
KeyboardNotBusy      = ($(KeyboardBusy) == False)
UWCS_START    = ( (KeyboardIdle > $(StartIdleTime)) \
                && ( $(CPUIdle) || (State != "Unclaimed" && State
!= "Owner")) )
```

Illustration

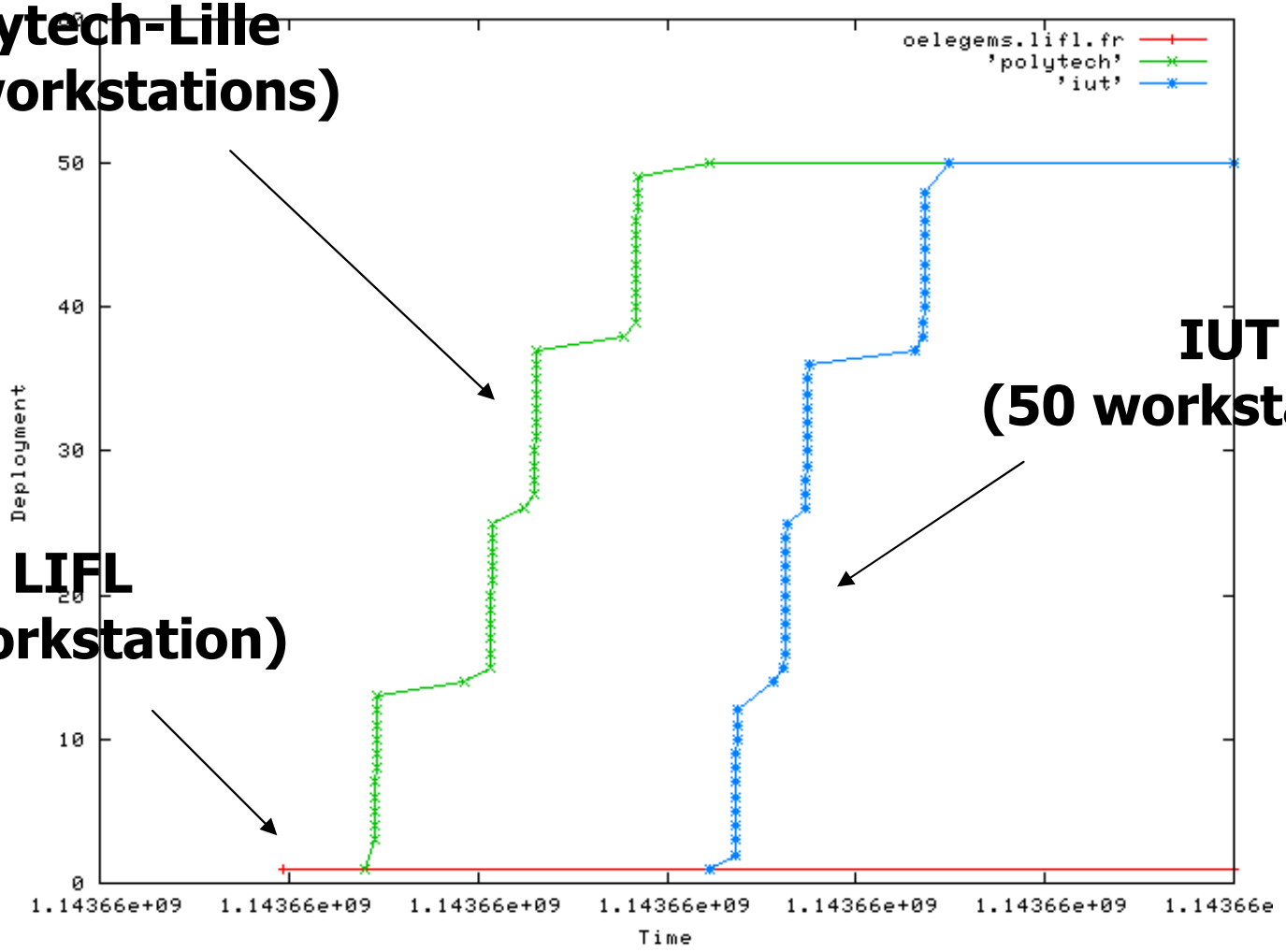


Deployment with Condor / Flocking at University of Lille1 (Polytech-Lille et IUT “A”)

Polytech-Lille (50 workstations)

LIFL (1 workstation)

IUT (50 workstations)



An example. Design of a parallel hybrid metaheuristic for radio network optimization

N. Melab, S. Cahon, E-G. Talbi, «**Grid computing for parallel bioinspired algorithms**», Journal of Parallel and Distributed Computing (JDPC), accepted, to appear 2006.

Design of radio networks in mobile telecommunication

- Financial context
  (cost of the network)
  - Number of sites
  - Quality of Service
- Network design
  - Positioning sites
  - Fixing a set of parameters for each antenna
- Very highly combinatorial (NP-hard)

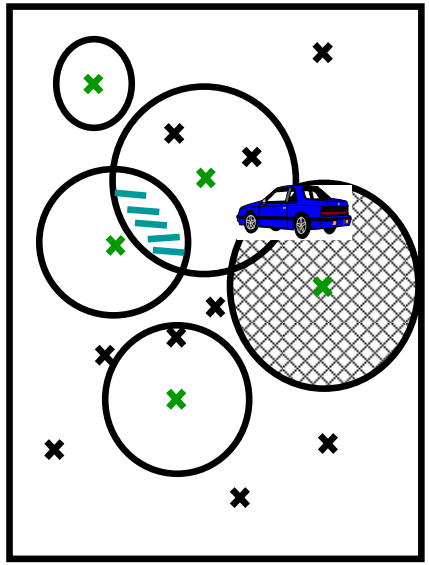
A brief description

- A set of base stations that satisfy the following constraints …
  - Cover
  - Handover
- … and optimizes the following criterion
  - Min. the number of sites
  - Min. interferences
  - Max. yield traffic

Parameters of antennas

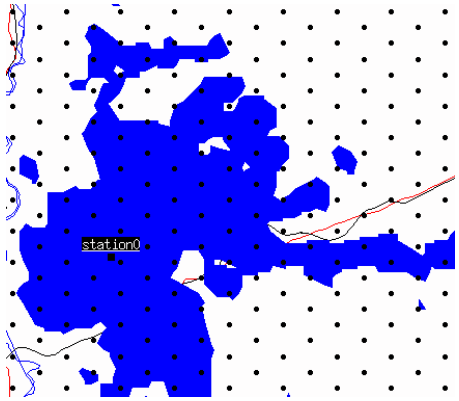
Données	bornes
Ps	[26, 55] dBm
Diagram	3 types
Hauteur	[30, 50] m
Azimut	[0, 359] °
Inclinaison	[-15, 0] °
TRX	[1, 7]

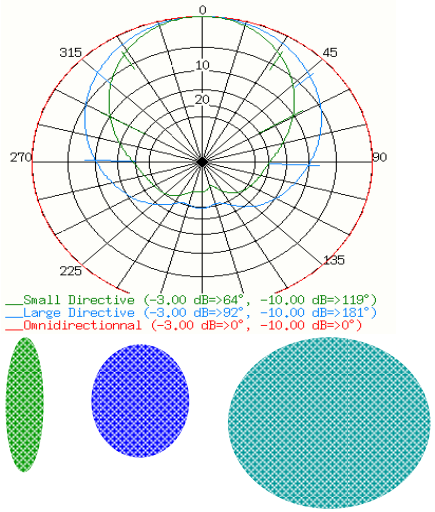
Working area



- Used sites
- Useless sites
- Handover area
- Cover area

Propagation model (Free spaceOkumura-Hata)

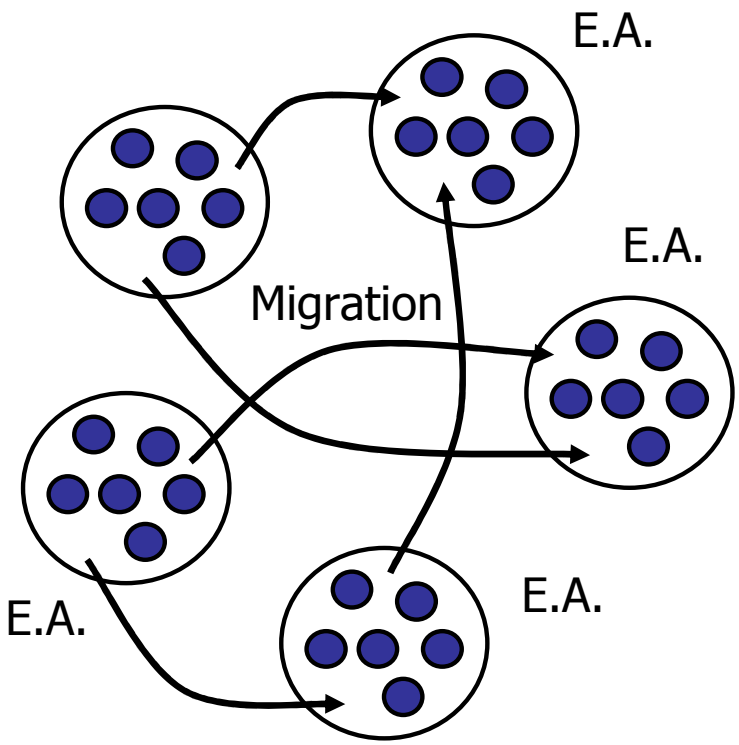




A quick view of the hybrid metaheuristic

- Cooperative island of Evolutionary Algorithms
  $\rightarrow$ Diversifying the search along the Pareto frontier and delaying convergence
- Coupling with Local Searches techniques
  $\rightarrow$ Improving sets of non-dominated solutions in isolated regions

Cooperative island E.As



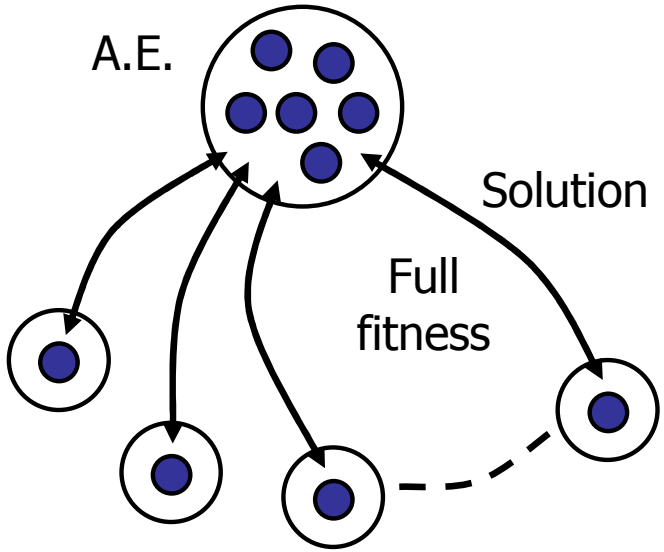
- Concurrent evolution of populations distributed in space
- Adapted to Multi-criterion problems
  → Exchanging a part of the local archive
- Improvement of the robustness, better results, …

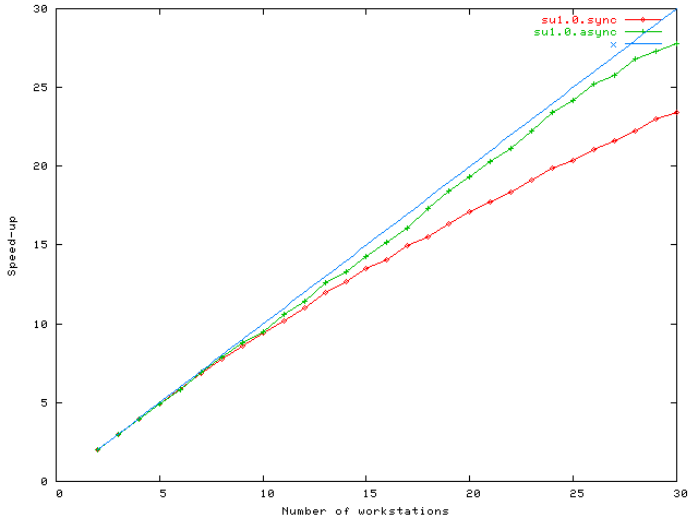
The parallel evaluation step

- Synchronous/asynchronous
- Deploying irregular tasks

$\rightarrow$ The computation time is dependent of the number of activated sites

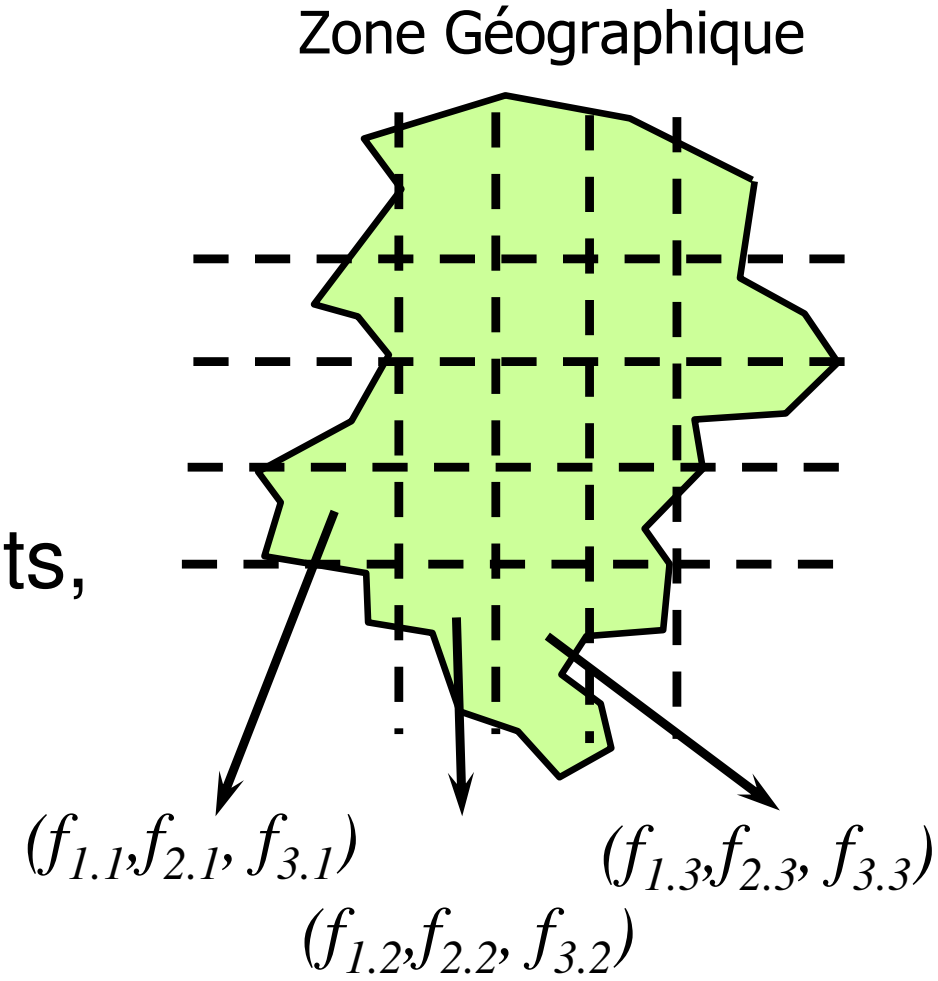
- Limited scalability of the synchronous model (size of the pop., *e.g.* 100)





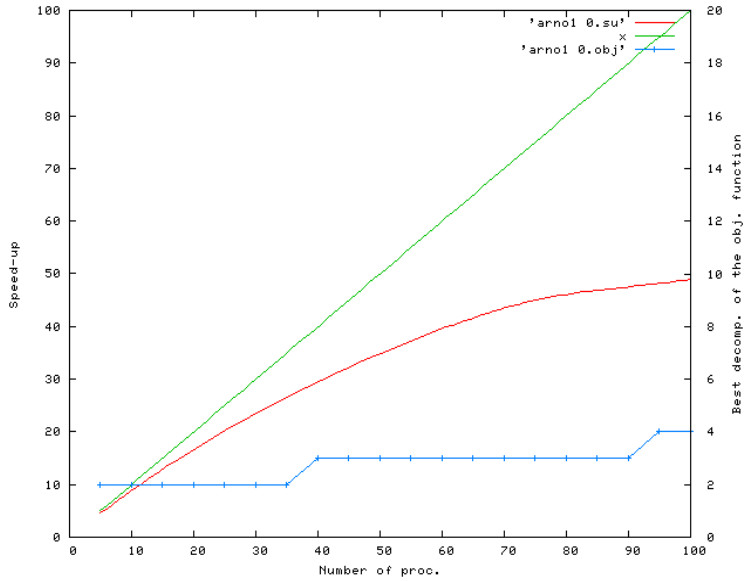
Parallel synchronous decomposition model (1)

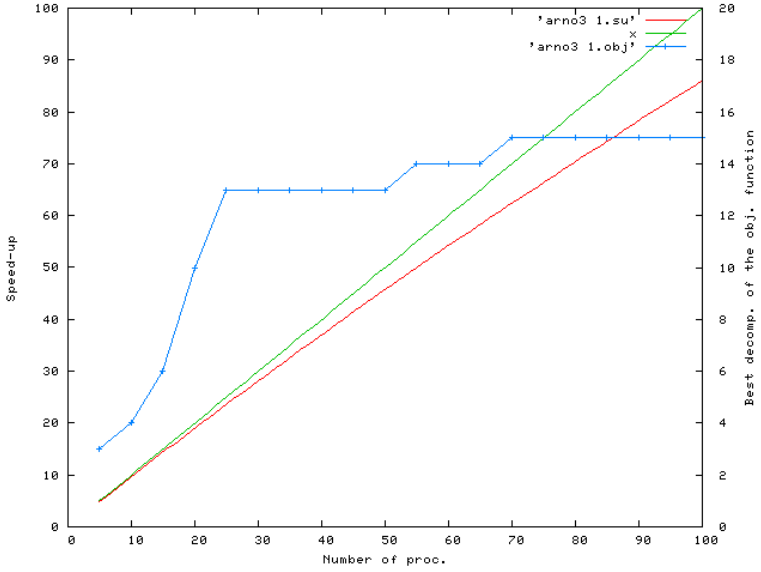
- Parallel evaluation of objectives
  - Partitioning the workspace area,
  - Gathering partial results, aggregating them, …
- Synchronous, fine-grained !

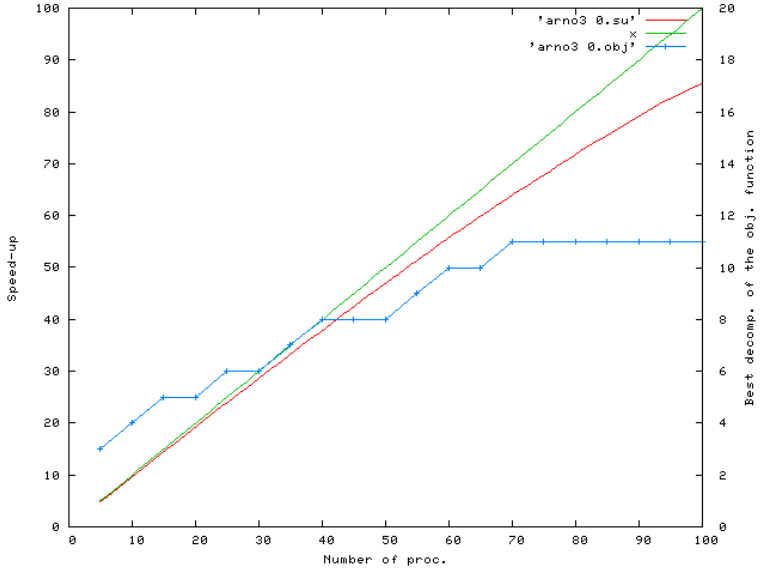


Parallel synchronous decomposition model (2)

- Should be used in conjunction with the parallel evaluation model to strongly enhance scalability !

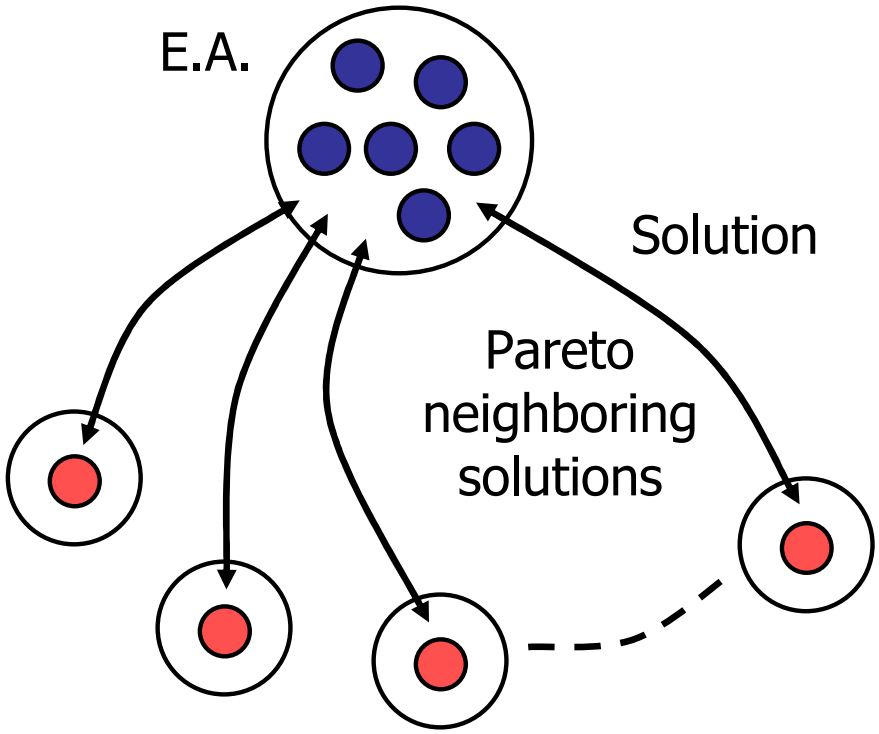






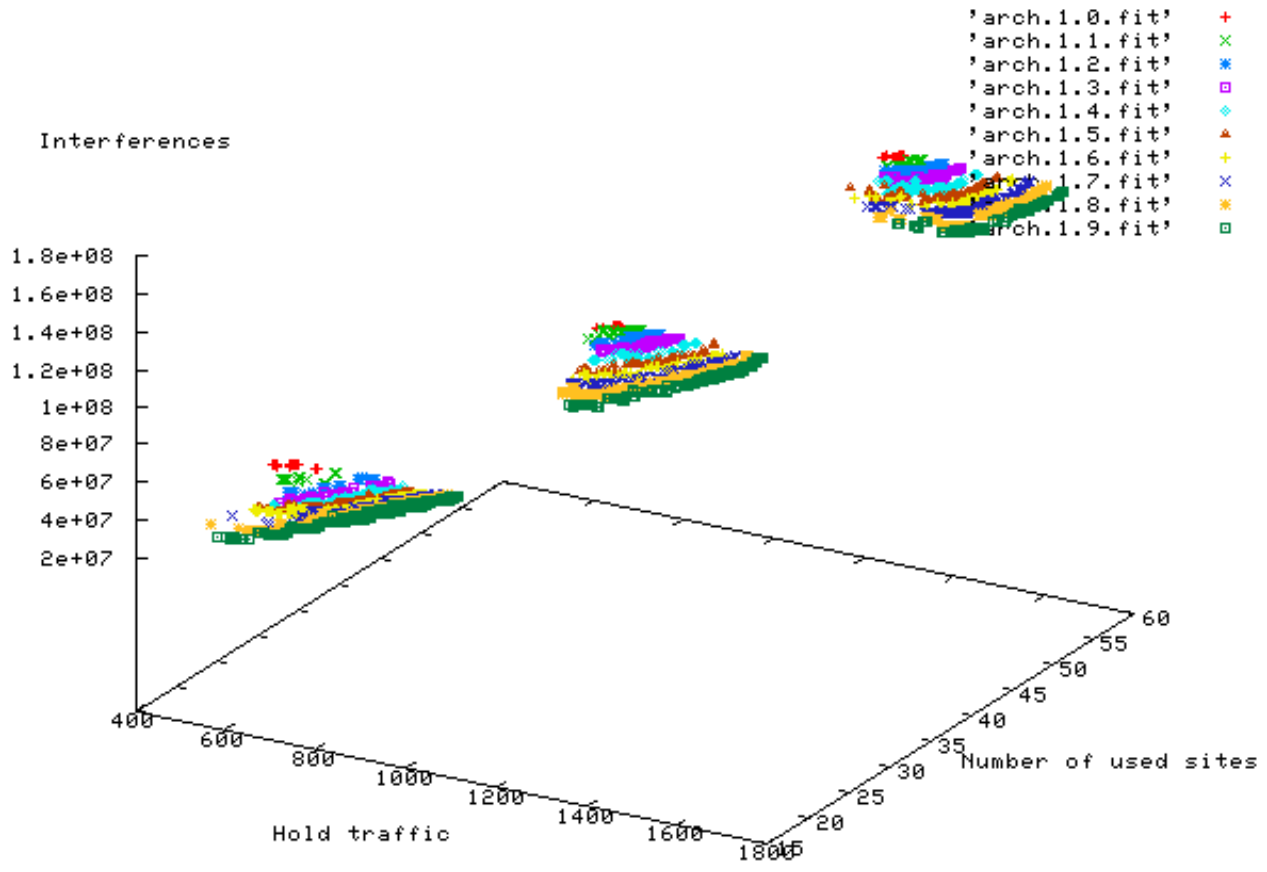
Paradiseo

Hybridization with Local Searches



- E.A. are exploration oriented
  - Efficient to find good solutions scattered in the Pareto frontier
  - Weaker to exploit those regions
- L.S. are intended to carefully tune the configuration of base stations of a given network.

Application of the L.S. starting from three random solutions (Arno 1.0)



The main issue

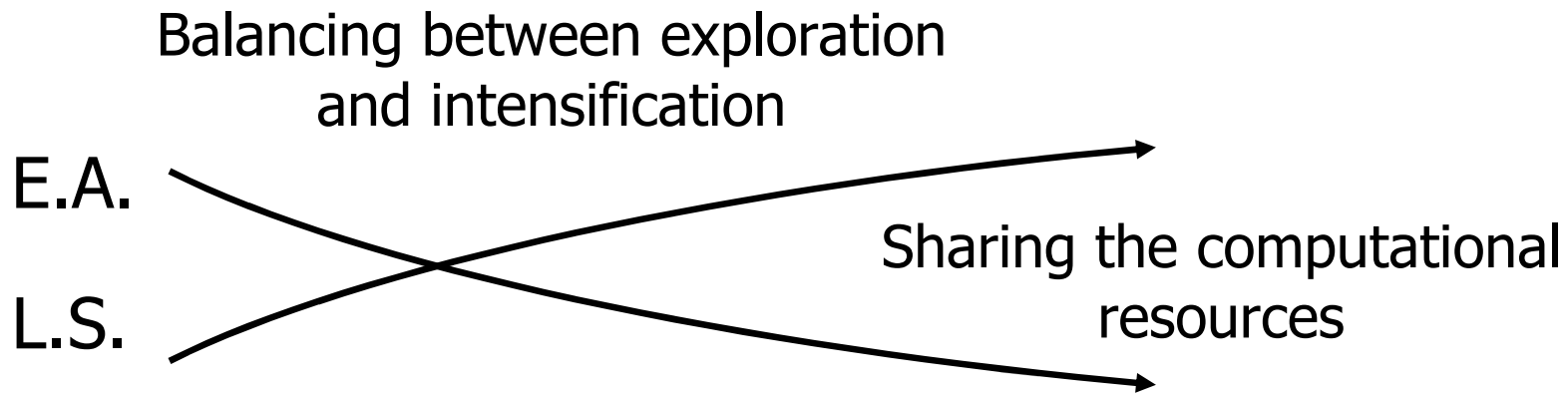
- Size of the local archive dramatically increases as an exploration of the neighborhood is iteratively applied to any new PO solution found

$\rightarrow$ Setting a max. width ($e.g.$ 5) and depth ($e.g.$ 10) for the iterative search.

Some mean statistics for one iteration of the L.S.

Instance	Average number of neighboring solutions processed	Average number of neighboring P.O. solutions	Mean CPU time
Arno 1.0	10661.8	11.29	511.763 s.
Arno 3.0	16467.5	15.55	461.44 s.
Arno 3.1	33232.5	17.23	2497.51 s.

Hybridization with the E.A.

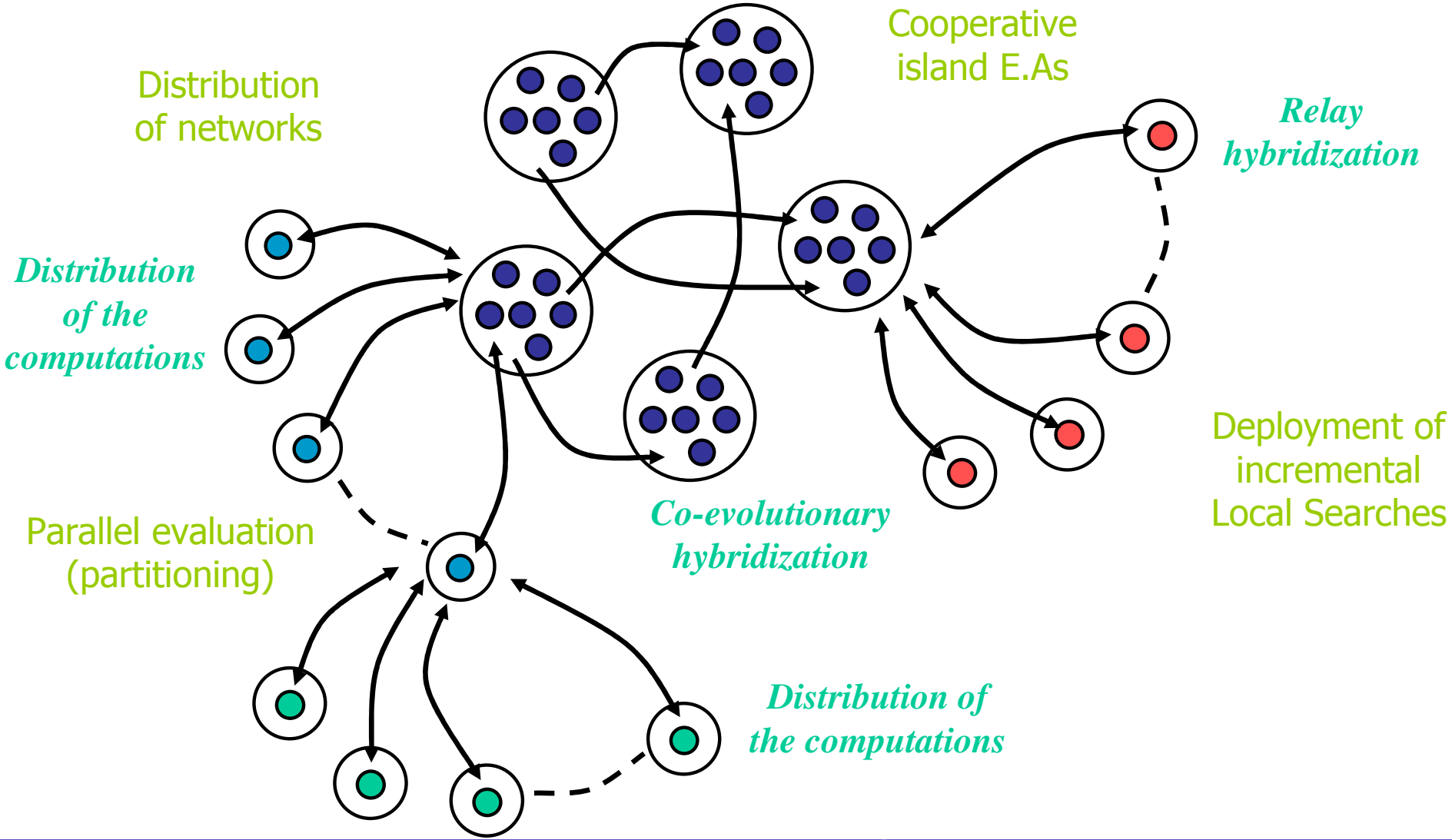


- Co-evolutionary mode
  - EA and LS concurrently run and share the same elites
- Beginning of the optimization process
  - EA are more powerful to quickly improve the archive of PO solutions than the LS
- EAs will gradually be less and less efficient
- Conversely, the LS will be more powerful to exploit such already good solutions and compute new better ones

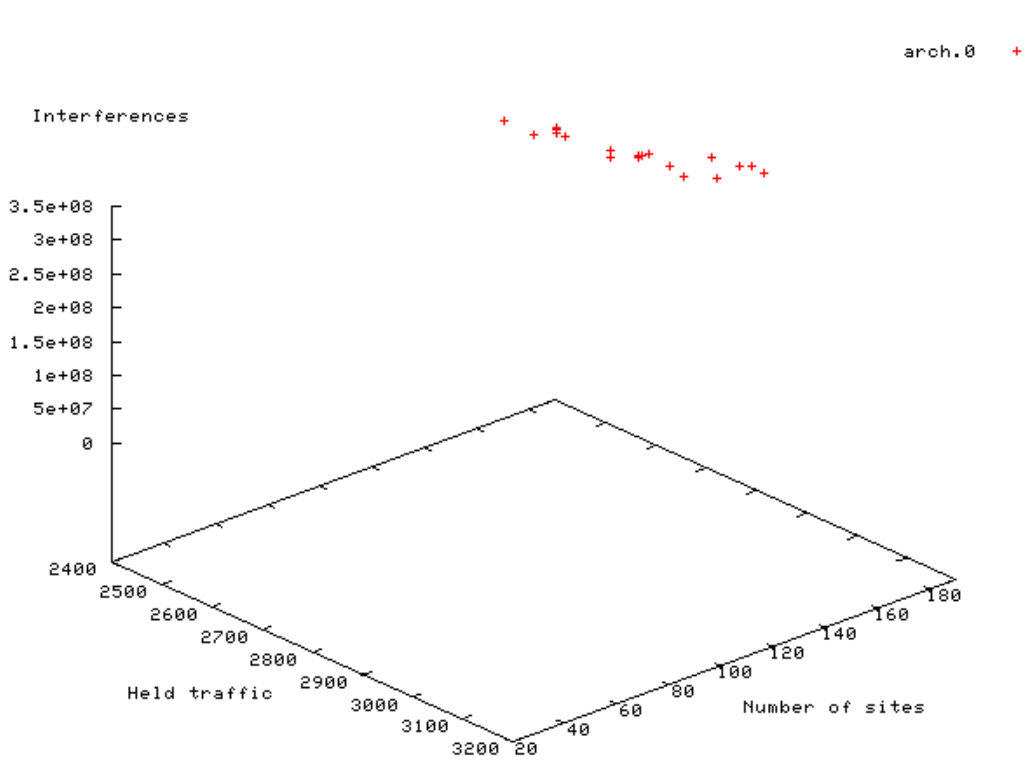
Models of parallelization

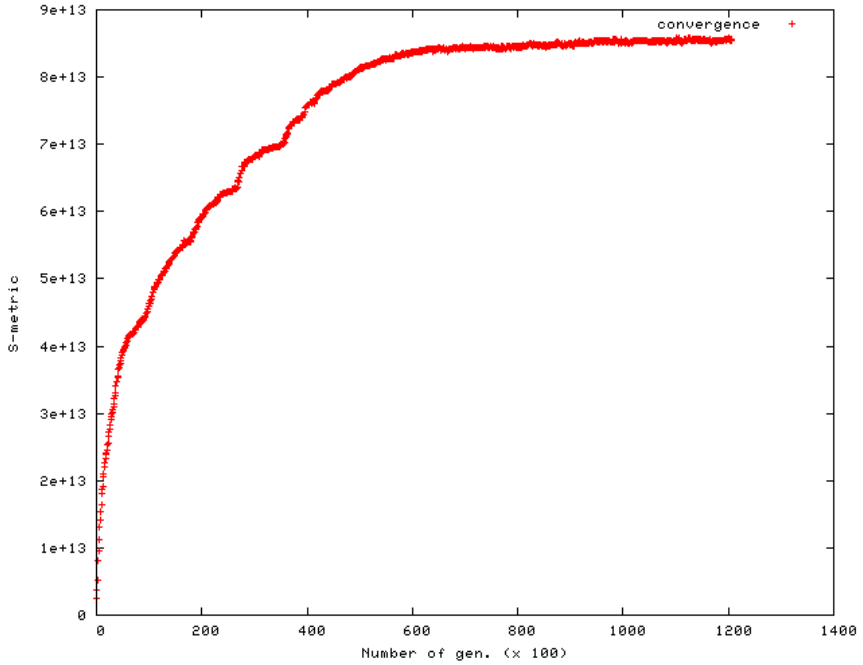
- High computational cost consecutive to the processing of networks at different stages of the optimization process
  $\rightarrow$ harnessing parallel/distributed platforms to get the results in a tractable time
- Parallel multi-start of LS
- Parallel (a)synchronous evaluation model
- Parallel synchronous decomposition model
- What has been done: the parallelization of the exploration of the neighborhood

A multi-layer hierarchical parallel/hybrid metaheuristic



Convergence of the elite of the PO solutions found during the search (Arno 1.0)

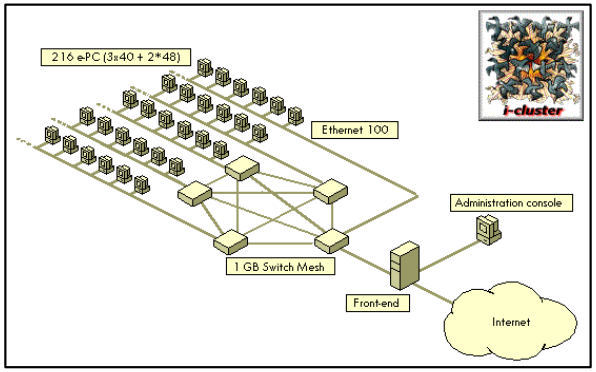




Efficiency bases on the S-metric

Solving instance Arno 1.0

Platform	Cluster Grid'5000 - Lille
Wave. Prop. precalculated	Yes
Number of proc.	52 (AMD Opteron 2.2 GHz)
Cumulative wall clock time	2112 h.
Wall clock time	44 h.
Parallel efficiency	0.92



Number of tasks	1038010
Min. cost	0.939 s.
Max. cost	5.92 s.
Mean cost	2.08 s.
Cumulative time	509.5 h.

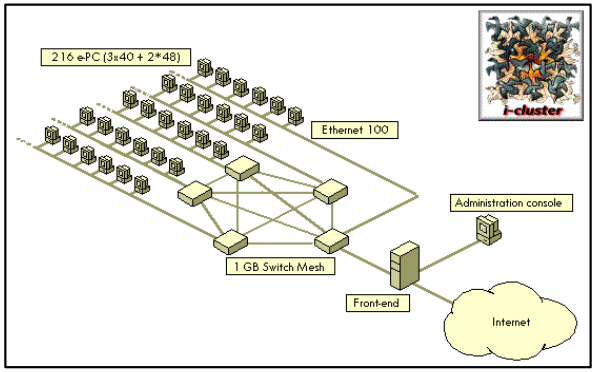
Parallelization of evaluations

Number of tasks	31359
Min. cost	0.173 s.
Max. cost	561.303 s.
Mean cost	173.637 s.
Cumulative time	1512.52 h.

Parallelization of Local Searches

Solving instance Arno 3.0

Platform	Cluster Grid'5000 - Lille
Wave. Prop. precalculated	Yes
Number of proc.	52 (AMD Opteron 2.2 GHz)
Cumulative wall clock time	1596 h.
Wall clock time	39 h.
Parallel efficiency	0.75



Number of tasks	856859
Min. cost	0.31 s.
Max. cost	11.34 s.
Mean cost	1.14 s.
Cumulative time	270.43 h.

Parallelization of evaluations

Number of tasks	56071
Min. cost	12.5257 s.
Max. cost	520.415 s.
Mean cost	85.132 s.
Cumulative time	1325.95 h.

Parallelization of Local Searches

Solving instance Arno 3.1

Platform	Metacomputing
Wave. Prop. precalculated	No
Parallel decomp. Model	Yes (x 2)
Number of proc.	100 (heterog. and non dedicated)
Cumulative wall clock time	30681 h.
Wall clock time	Almost 15 days
Parallel efficiency	0.98

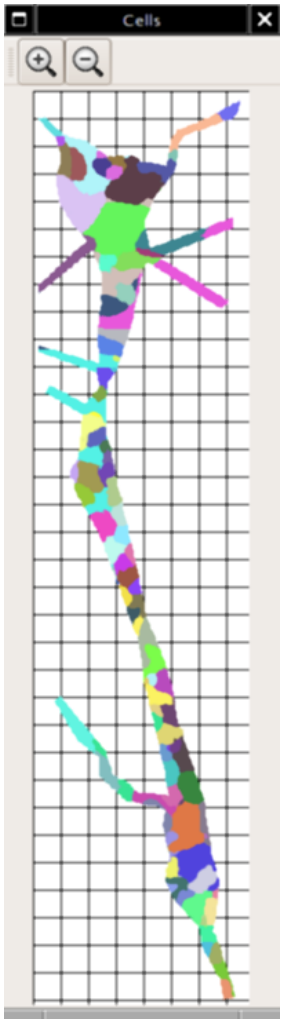
Number of tasks	1897046
Min. cost	7.815 s.
Max. cost	247.967 s.
Mean cost	21.51 s.
Cumulative time	11340 h.

Parallelization of evaluations

Number of tasks	15299
Min. cost	1832.57 s.
Max. cost	24380.7 s.
Mean cost	4593.52 s.
Cumulative time	19521 h.

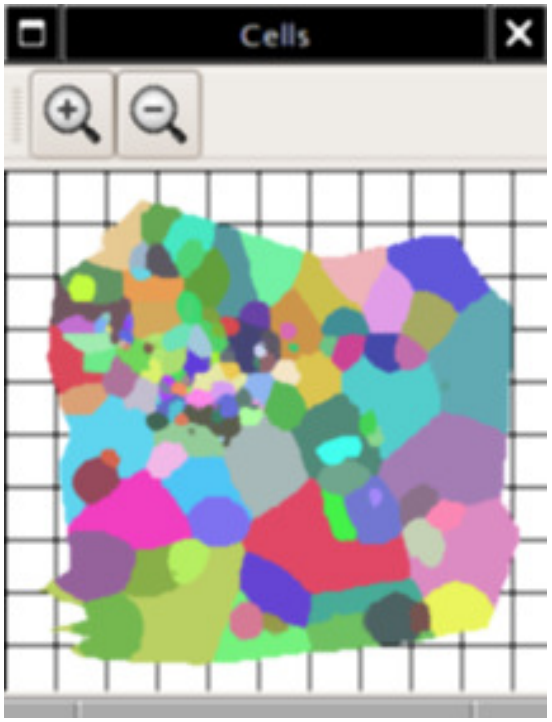
Parallelization of Local Searches

Decomposition in cells of some optimized networks taken from the final archive



Arno 1.0 (highway)

Arno 3.0 (urban)



Arno 3.1 (urban)