Reusable Design of Parallel and Distributed Evolving Objects

Nouredine MELAB
El-Ghazali TALBI

http://paradiseo.gforge.inria.fr
paradiseo-help@lists.gforge.inria.fr

Outline

- Contributions
- An unifying view of three parallel hierarchical levels
- ParadisEO-PEO Evolutionary Algorithm Components
- Low Level Relay GA ↔ HC
- High Level Relay GA + HC
- High Level Coevolutionary Island Model
  - HL Coevolutionary – Asynchronous Island Model
  - HL Coevolutionary – Synchronous Island Model
- ParadisEO-PEO – Parallel Models
  - The Parallel Evaluation of the Population
  - Parallel Evaluation of the Objective Function
- Conclusions

Contributions

ParadisEO-EO - a framework of Evolutionary Algorithms

ParadisEO-MO

solution-based metaheuristics

- Hill Climbing
- Simulated Annealing
- Tabu search

ParadisEO-PEO

parallel and distributed metaheuristics

- **Parallelism**
  partitioning solutions at several steps
- **Cooperation / Hybridization**
  algorithms exchange solutions
  Ex. island cooperation

ParadisEO-MOEO

techniques related to multi-objective optimization

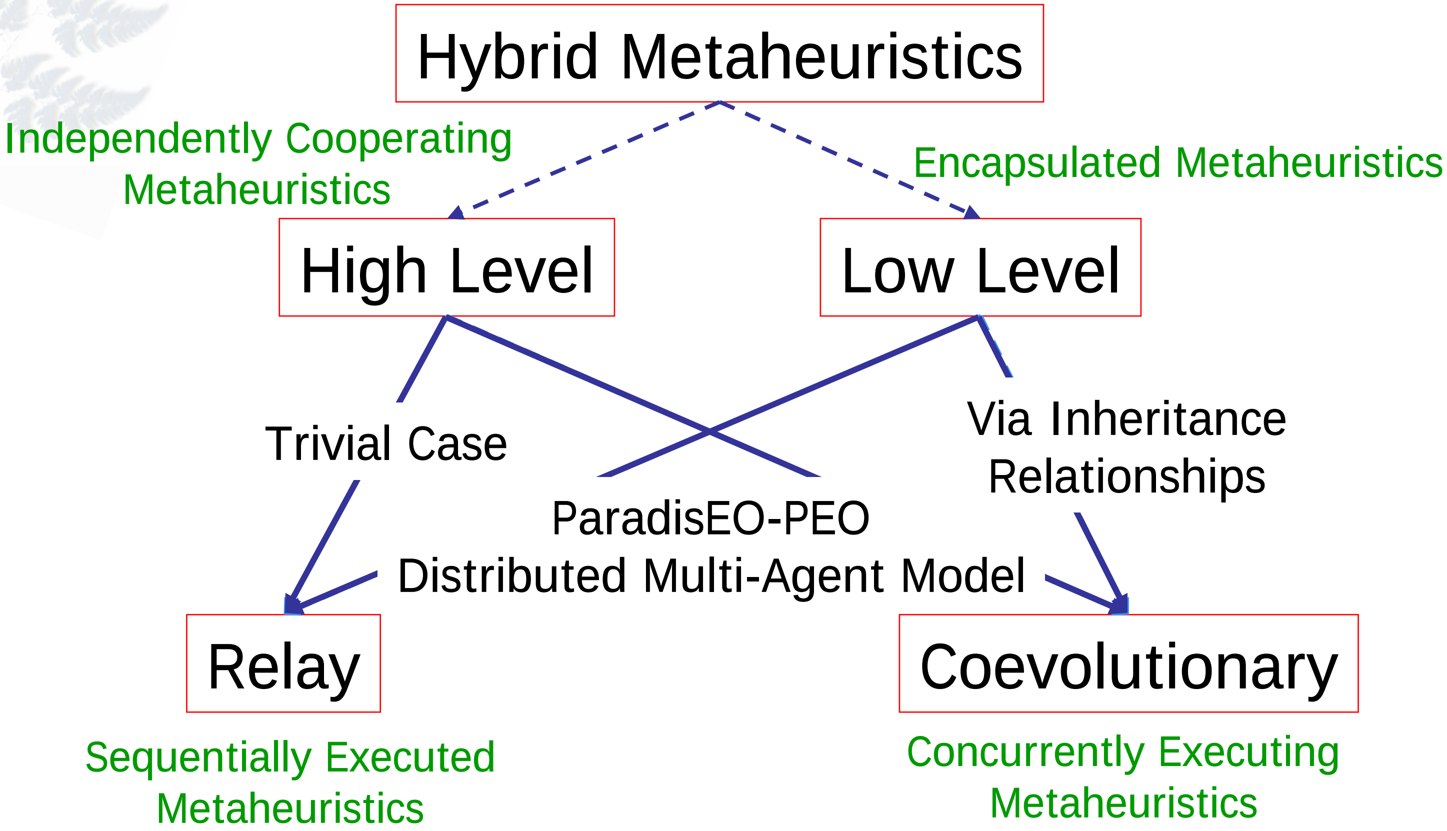
http://paradiseo.gforge.inria.fr

An unifying view of three parallel 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

Hybrid Metaheuristics Development

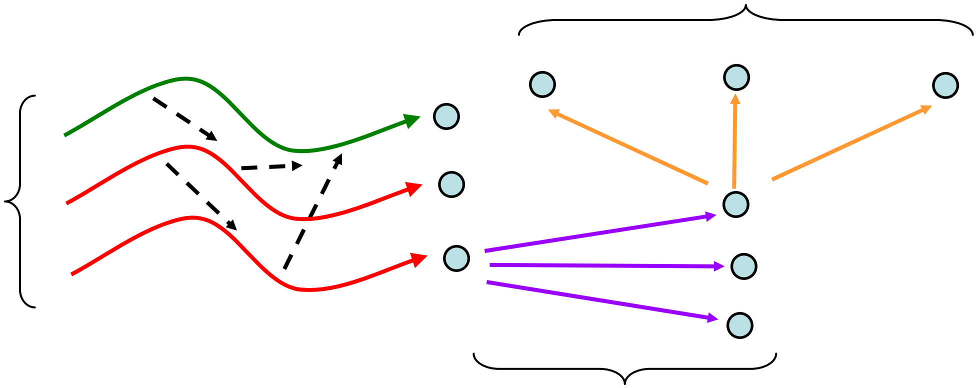


E-G. Talbi, **A taxonomy of hybrid metaheuristics**, Journal of Heuristics, 2002.

Design of several levels of parallelization/hybridization

Processing of a single solution
(Objective / Data partitioning)

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

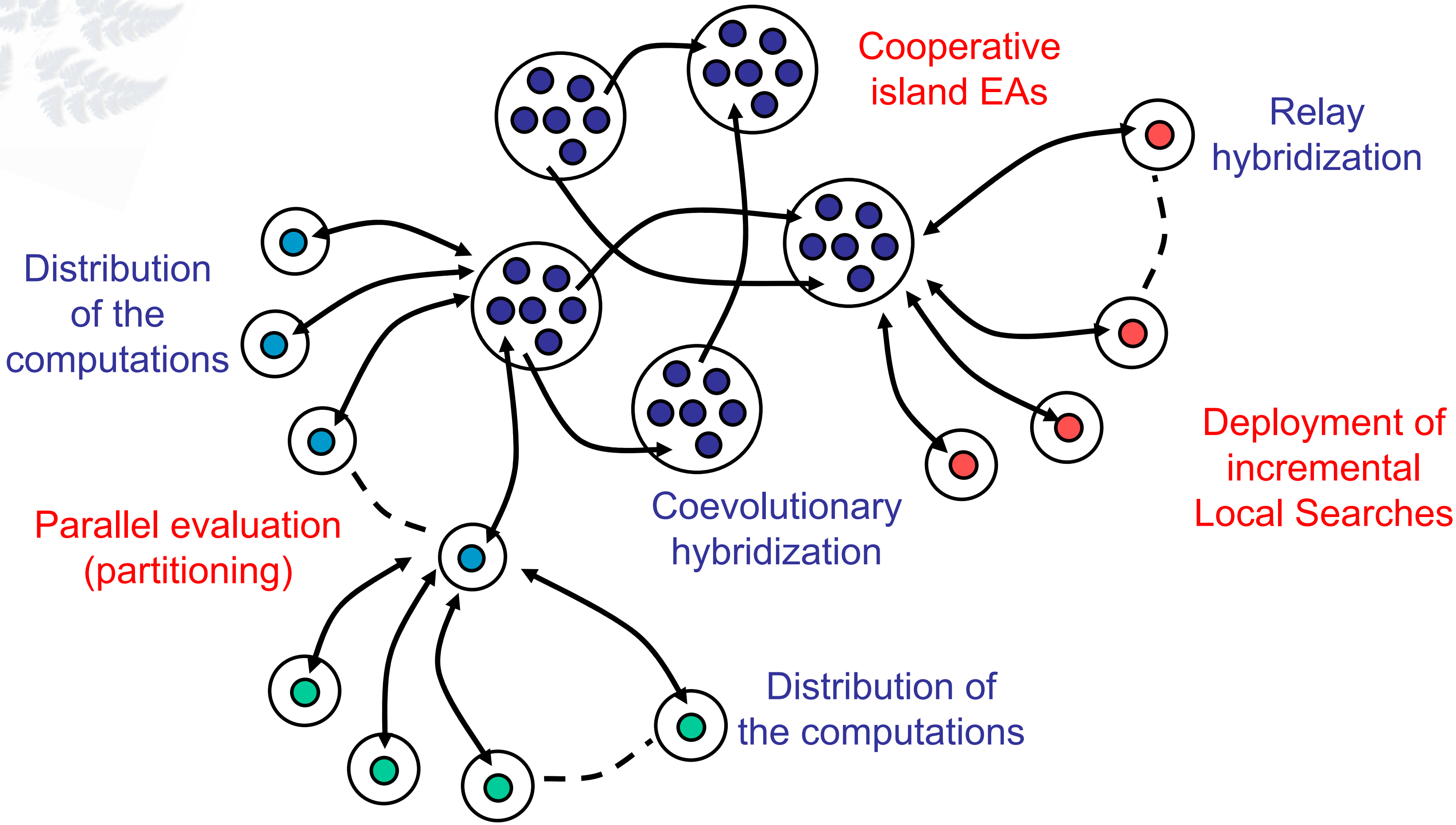


Parallel evaluation of
the neighborhood/population

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

Heuristic   Population / Neighborhood   Solution

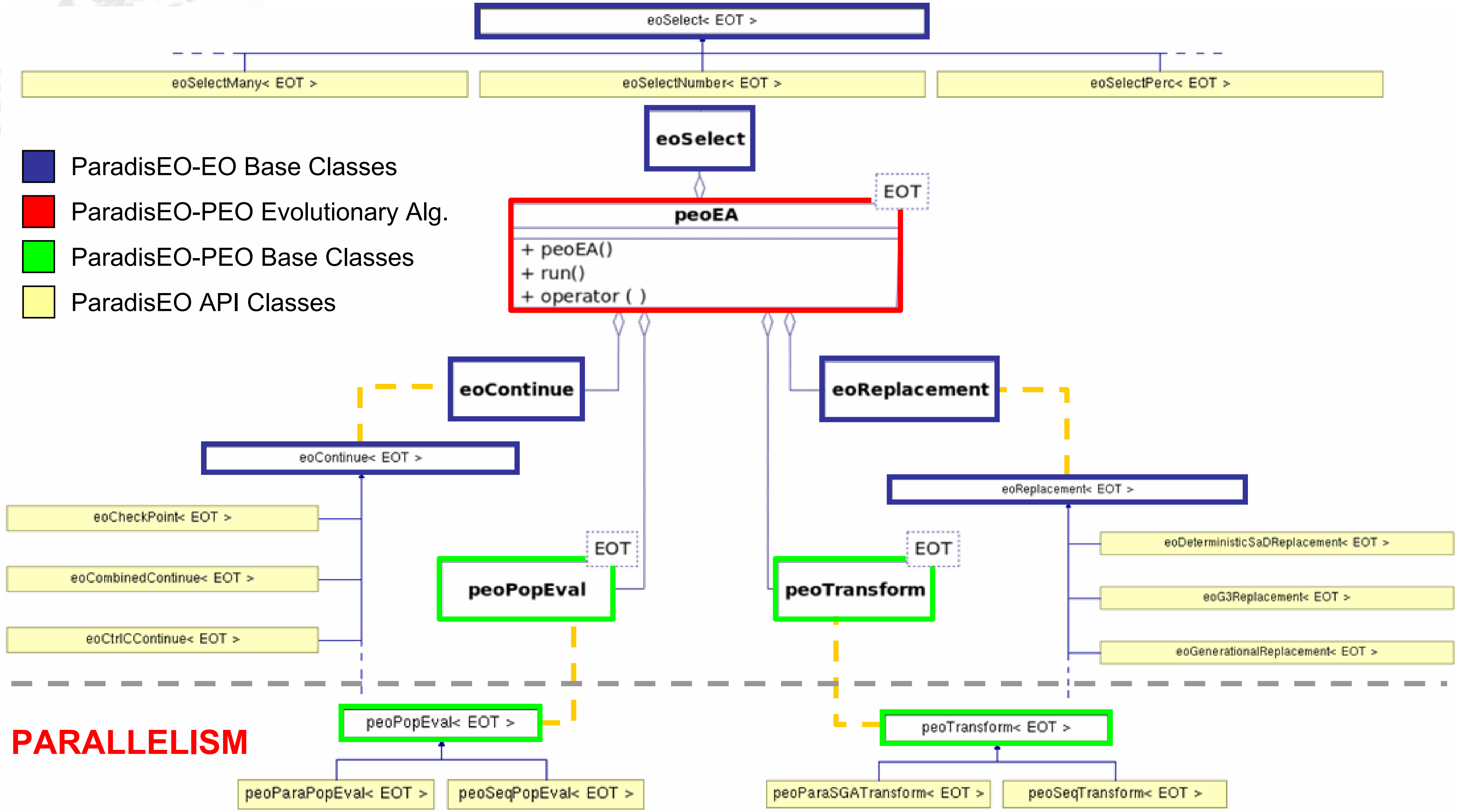
A multi-layer hierarchical parallel/hybrid metaheuristic



Outline

- Contributions
- An unifying view of three parallel hierarchical levels
- **ParadisEO-PEO Evolutionary Algorithm Components**
- Low Level Relay GA ↔ HC
- High Level Relay GA + HC
- High Level Coevolutionary Island Model
  - HL Coevolutionary – Asynchronous Island Model
  - HL Coevolutionary – Synchronous Island Model
- ParadisEO-PEO – Parallel Models
  - The Parallel Evaluation of the Population
  - Parallel Evaluation of the Objective Function
- Conclusions

ParadisEO-PEO Evolutionary Algorithm Components



Outline

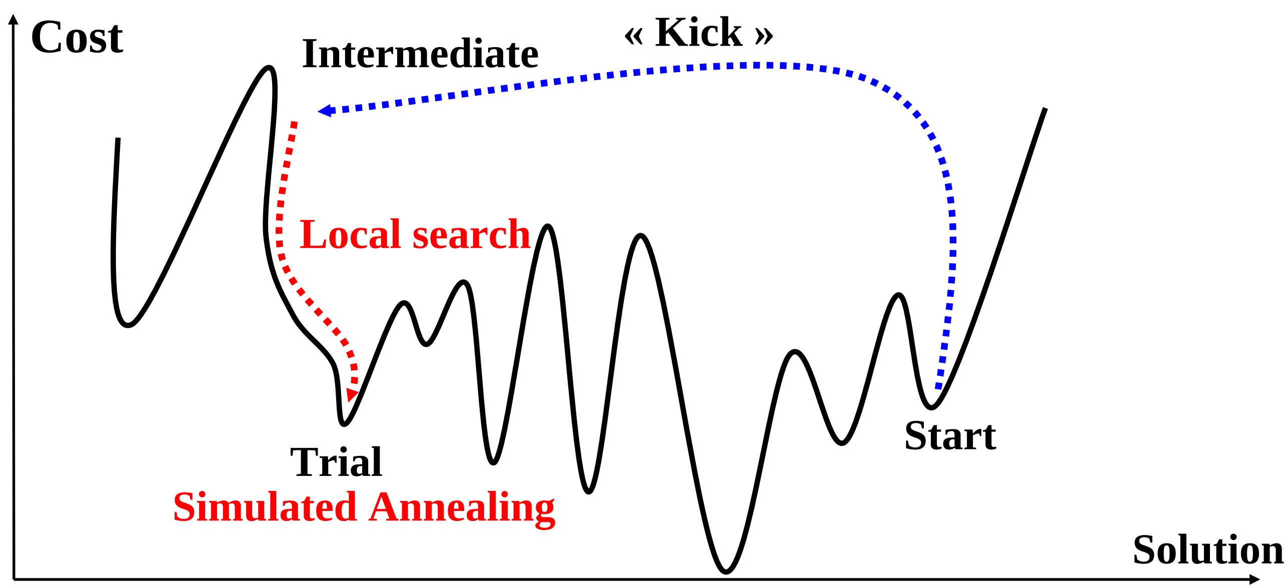
- Contributions
- An unifying view of three parallel hierarchical levels
- ParadisEO-PEO Evolutionary Algorithm Components
- **Low Level Relay GA ↔ HC**
- High Level Relay GA + HC
- High Level Coevolutionary Island Model
  - HL Coevolutionary – Asynchronous Island Model
  - HL Coevolutionary – Synchronous Island Model
- ParadisEO-PEO – Parallel Models
  - The Parallel Evaluation of the Population
  - Parallel Evaluation of the Objective Function
- Conclusions

Low-level Relay Hybrid

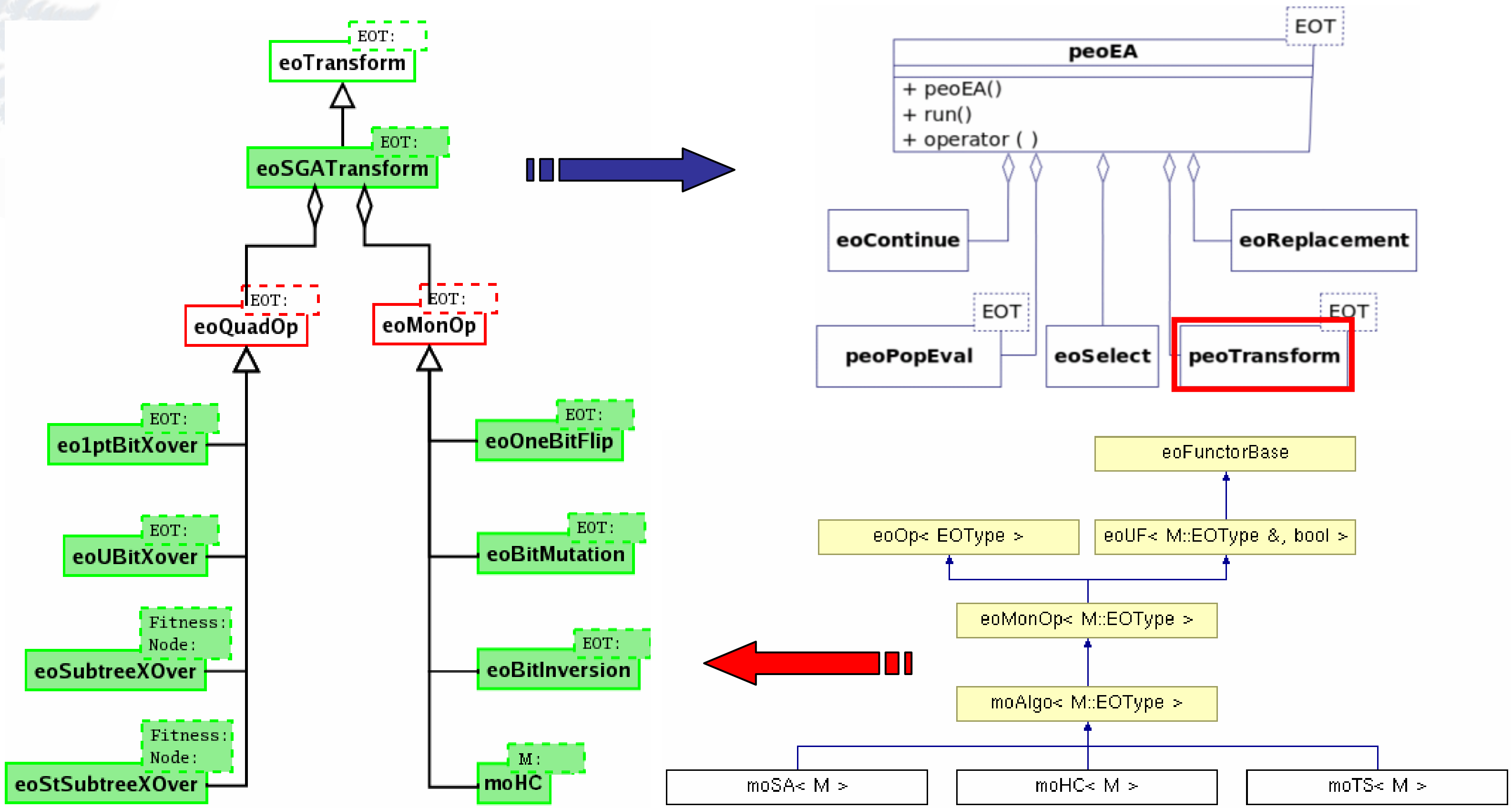
It denotes algorithms in which a given metaheuristic is embedded into a single solution metaheuristic.

Example: Local Search embedded in Simulated Annealing

[Martin et Otto 92]



Low Level Relay - GA ↔ HC (1)



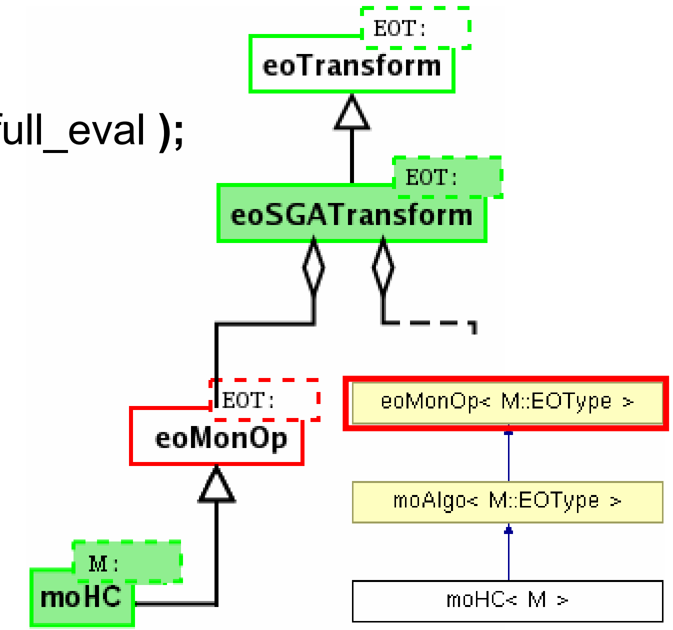
A single solution method is a unary operator !!

Low Level Relay - GA ↔ HC (2)

LS

```
...
moHC<TwoOpt> hc( pmx_two_opt_init, ..., full_eval );
...
```

```
...
eoSGATransform<Route> ox_transform(
                order_cross, CROSS_RATE,
                hc, MUT_RATE );
```



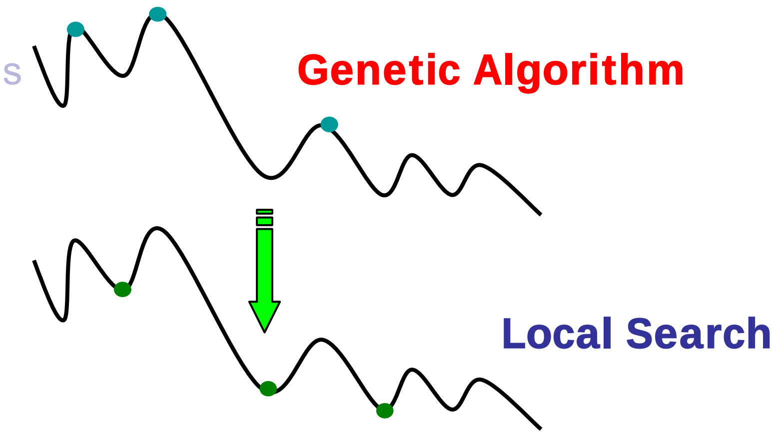
EA

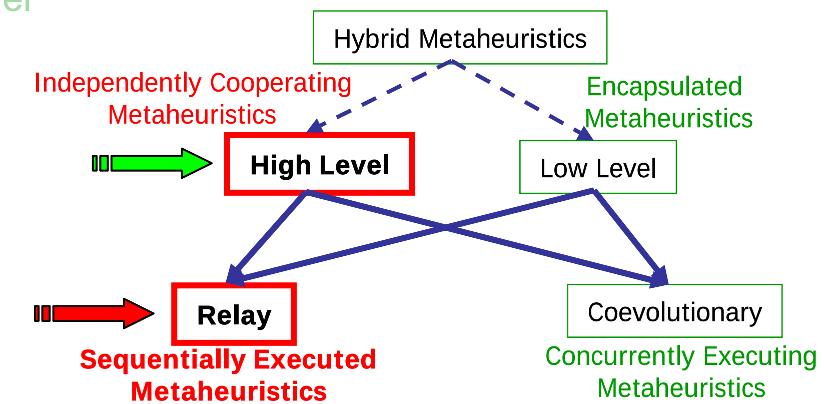
```
...
peoEA<Route> ox_ea( ox_checkpoint, ..., ox_transform, ox_replace);
...
```

The mutation operator can be replaced by a local search method as all the MO algorithms have as base class in the hierarcy the eoMonOp class.

Outline

- Contributions
- An unifying view of three parallel hierarchical levels
- ParadisEO-PEO Evolutionary Algorithm Components
- Low Level Relay GA ↔ HC
- **High Level Relay GA + HC**
- High Level Coevolutionary Island Model
    - HL Coevolutionary – Asynchronous Island Model
    - HL Coevolutionary – Synchronous Island Model
- ParadisEO-PEO – Parallel Models
    - The Parallel Evaluation of the Population
    - Parallel Evaluation of the Objective Function
- Conclusions



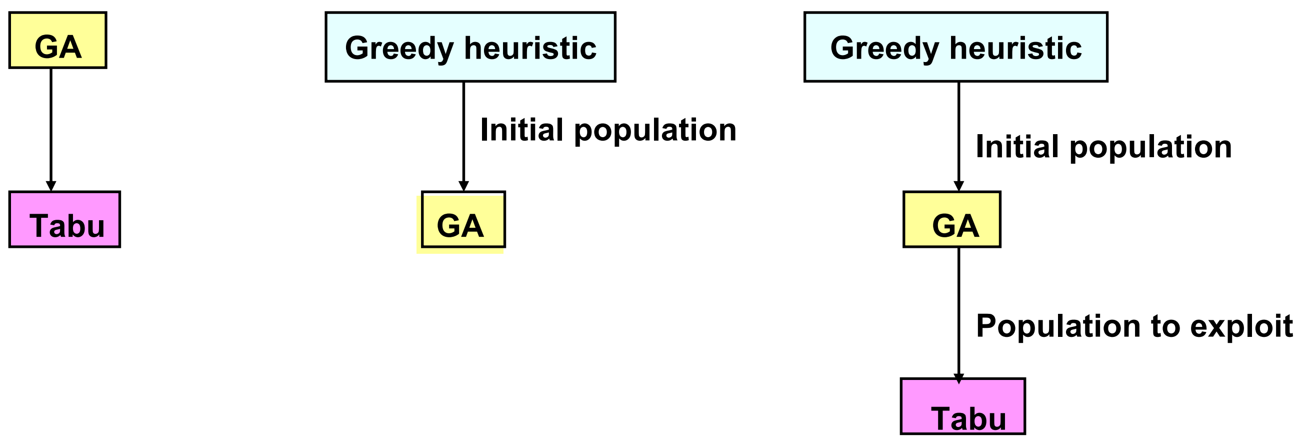


High-level Relay Hybrid

Self-contained optimization methods are executed in a sequence (each using the output of the previous as its input, acting in 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 Relay - GA + HC

LS

```
…
moHC<TwoOpt> hc( pmx_two_opt_init, ..., full_eval );
…
```

EA

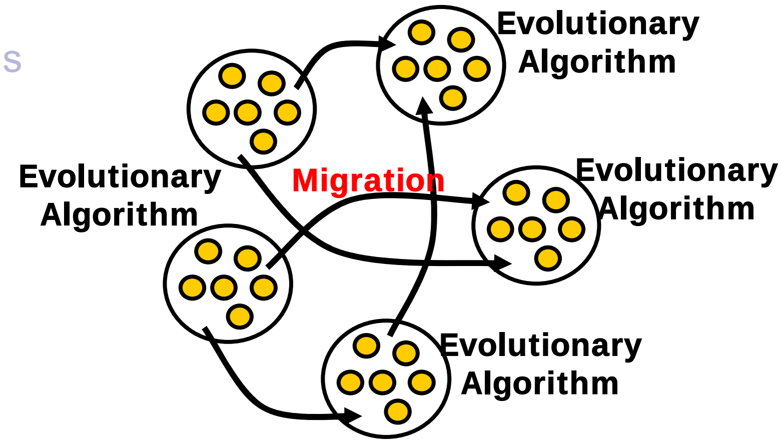
```
…
peoEA<Route> ox_ea( ox_checkpoint, ..., ox_transform, ox_replace);
ox_ea( ox_pop );
…

for ( int index = 0; index < ox_pop.size(); index++ ) {

        std :: cout << "Initial value: " << ox_pop[ index ].fitness() << std :: endl;
        hc( ox_pop[ index ] );
        std :: cout << "Final value: " << ox_pop[ index ].fitness() << std :: endl;
}
…
```

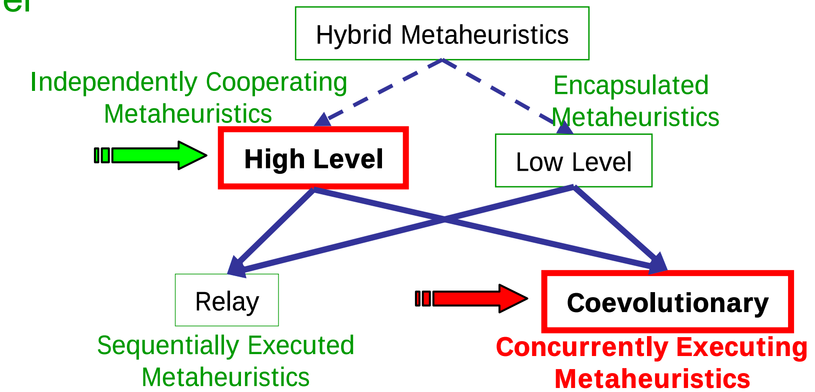
+

The Hill Climbing local search operator is applied on the resulting individuals.

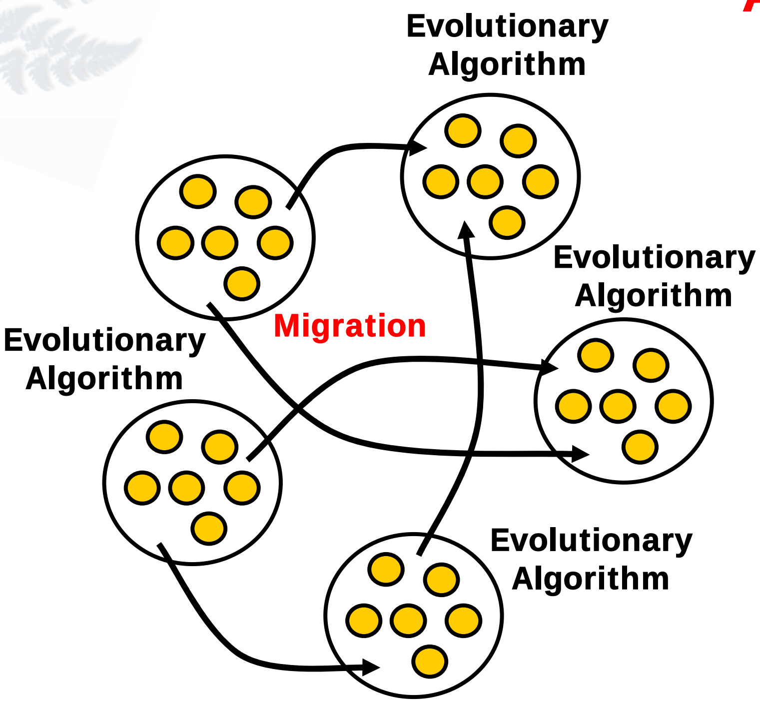
High Level Coevolutionary Island Model

- Contributions
- An unifying view of three parallel hierarchical levels
- ParadisEO-PEO Evolutionary Algorithm Components
- Low Level Relay GA ↔ HC
- High Level Relay GA + HC
- **High Level Coevolutionary Island Model**
  - HL Coevolutionary – Asynchronous Island Model
  - HL Coevolutionary – Synchronous Island Model
- ParadisEO-PEO – Parallel Models
  - The Parallel Evaluation of the Population
  - Parallel Evaluation of the Objective Function
- Conclusions





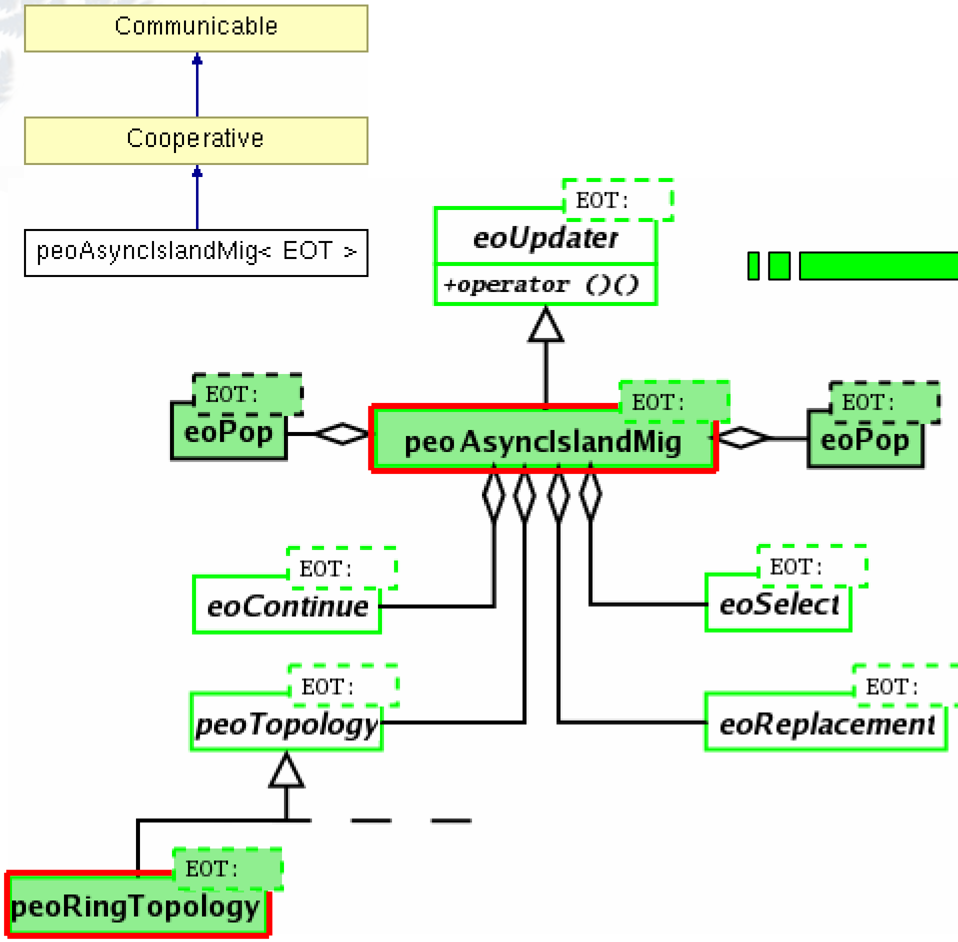
High Level Coevolutionary Island Model

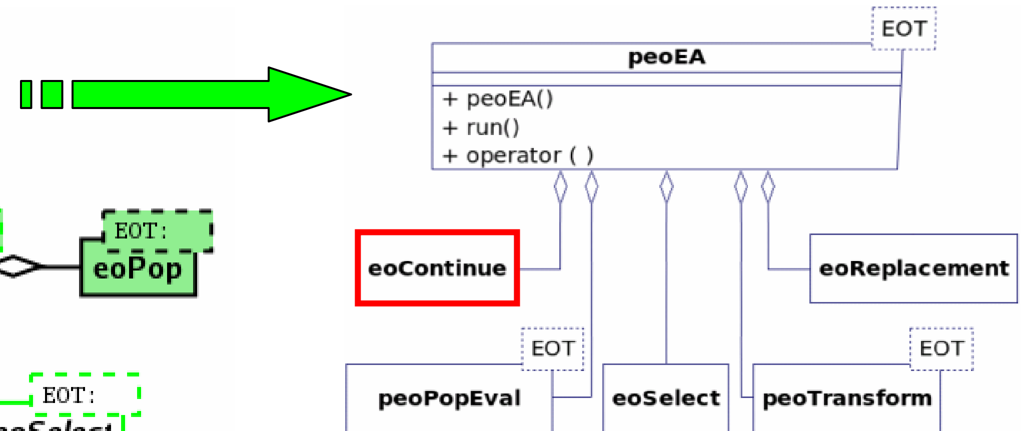


A hybridization scheme!

- 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

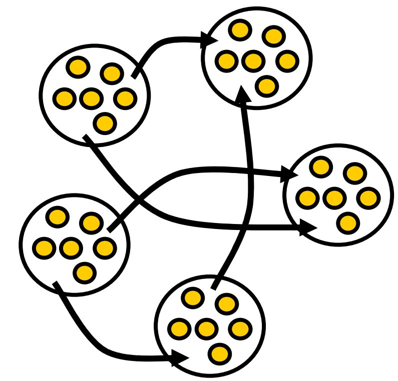
HL Coevolutionary – Async. Island Model (1)





HL Coevolutionary – Async. Island Model (2)

#1 Migration Flow - a Migration Topological Model



```
RingTopology topo;
```

#2 Asynchronous Migration Objects are part of a Topology

```
...
eoPeriodicContinue<Route>        ox_mig_cont ( MIG_FREQ );

eoRandomSelect<Route>            ox_mig_select_one;
eoSelectNumber<Route>            ox_mig_select( ox_mig_select_one, MIG_SIZE );
eoPlusReplacement<Route>         ox_mig_replace;

peoAsyncIslandMig<Route> ox_mig(
        ox_mig_cont,             // migrations occur at every MIG_FREQ generations
        ox_mig_select,           // strategy of selection for obtaining the emigrants
        ox_mig_replace,          // strategy of replacement for integrating the immigrants
        topo,                    // topology – migrations follow a pre-defined topology
        ox_pop, ox_pop );        // the source & destination populations
```

HL Coevolutionary – Async. Island Model (3)

#3 The Asynchronous Migration Object is linked to the EA as a CHECKPOINT!

- A - the Migration Object is called at every generation

```
...
eoCheckPoint<Route> ox_checkpoint( ox_cont );
ox_checkpoint.add( ox_mig );
...
```

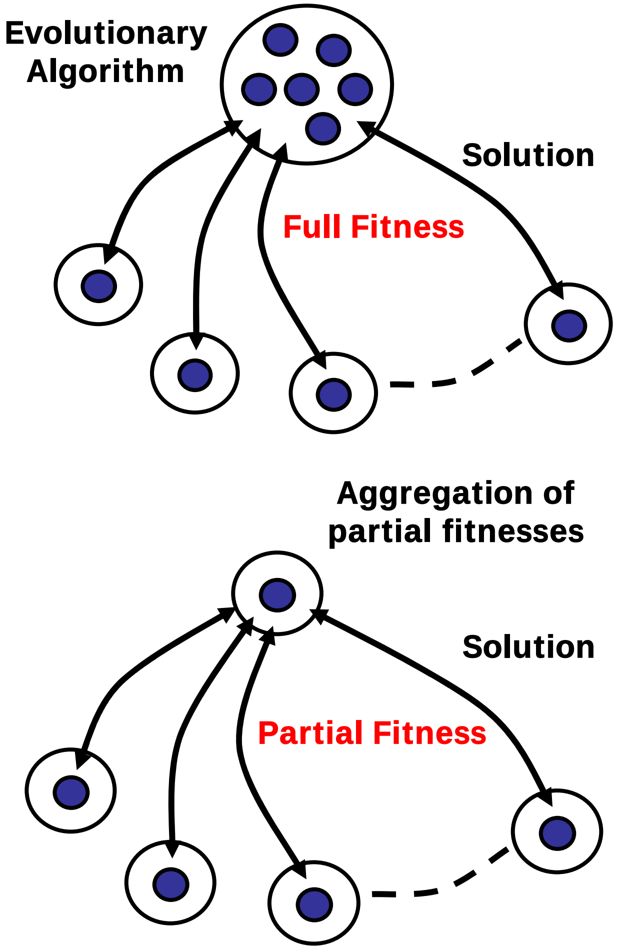
- B - the **Evolutionary Object** has to be set as **OWNER** of the **Migration Object !**

```
...
peoEA<Route> ox_ea( ox_checkpoint, ox_pop_eval, …, ox_replace );
ox_mig.setOwner( ox_ea );
ox_ea( ox_pop );
...
```

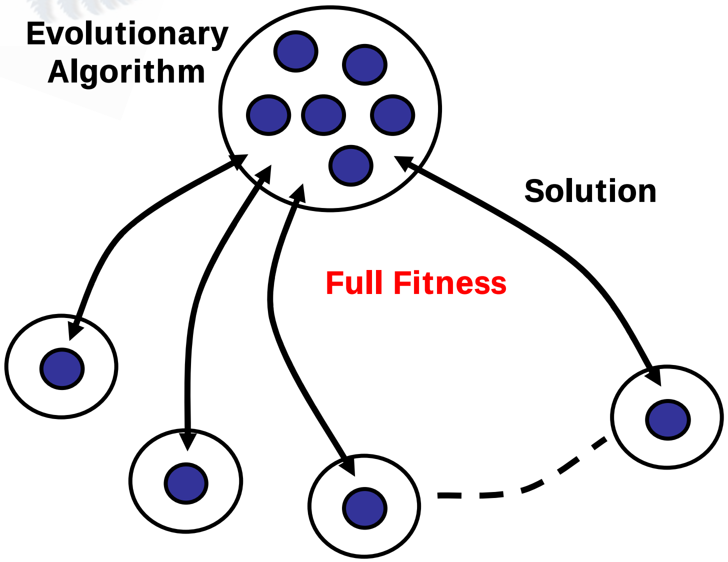
#4 Two or more Evolutionary Algorithms have to be part of the same topology!

ParadisEO-PEO – Parallel Models

- Contributions
- An unifying view of three parallel hierarchical levels
- ParadisEO-PEO Evolutionary Algorithm Components
- Low Level Relay GA ↔ HC
- High Level Relay GA + HC
- High Level Coevolutionary Island Model
    - HL Coevolutionary – Asynchronous Island Model
    - HL Coevolutionary – Synchronous Island Model
- **ParadisEO-PEO – Parallel Models**
    - The Parallel Evaluation of the Population
    - Parallel Evaluation of the Objective Function
- Conclusions



The Parallel Evaluation of the Population



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

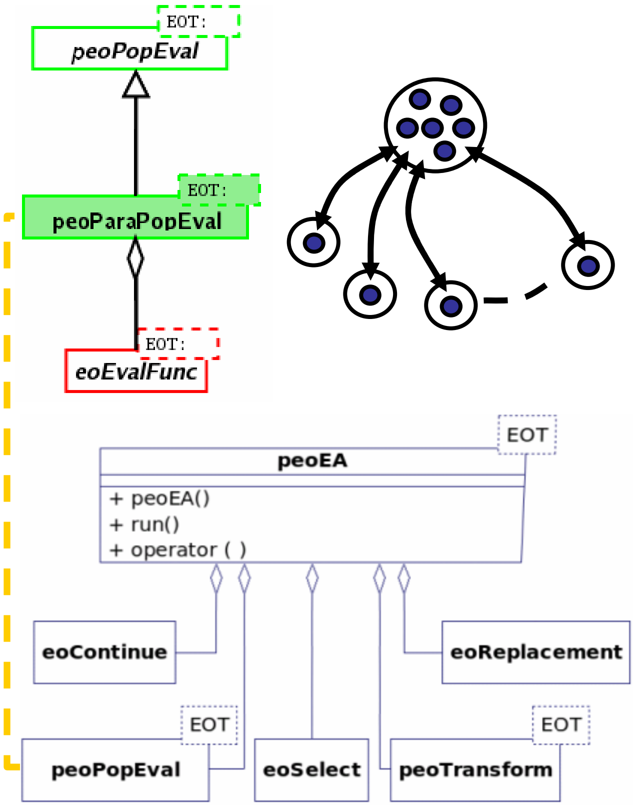
The Parallel Evaluation of the Population

#1 Full fitness evaluator – no parallelism mechanisms enclosed

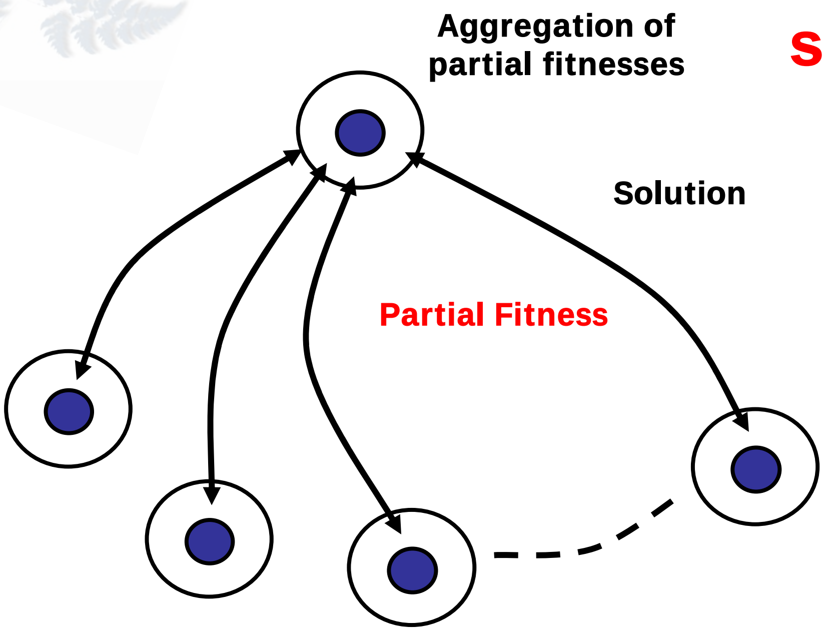
```
…
RouteEval full_eval;
…
```

#2 A Parallel Evaluator WRAPS the full evaluator:

```
…
peoParaPopEval<Route> ox_pop_eval( full_eval );
…

peoEA<Route> ox_ea( ox_checkpoint, ox_pop_eval, … );
ox_ea (ox_pop);
```



Parallel Evaluation of the Objective Function

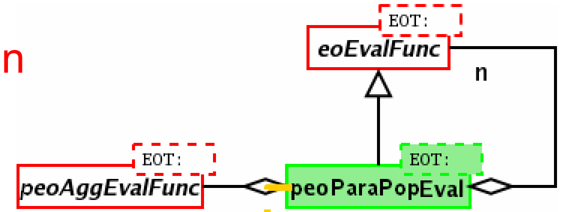


The evaluation of a single solution may be parallelized !

- To be used in conjunction with the parallelization of the population to enhance scalability
- Most relevant with the synchronous parallel evaluation step
- Partitioning data, objectives.

Parallel Evaluation of the Objective Function

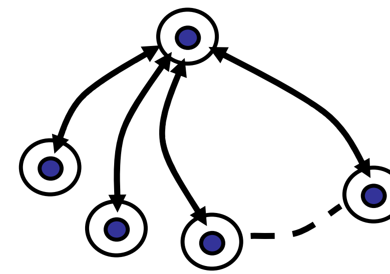
#1 Fitness computed as an aggregation function of partial fitness function evaluators:

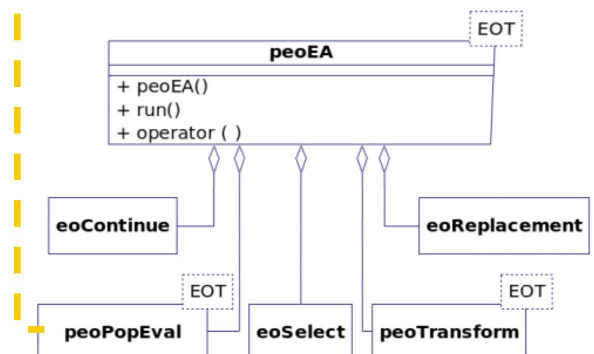


MERGE F.

```
…
MergeRouteEval merge_eval;

std :: vector< eoEvalFunc<Route>* > part_eval;

for (int i = 1; i <= NUM_PART_EVALS; i ++) {

    part_eval.push_back (
          new PartRouteEval( (i-1) / (float) NUM_PART_EVALS,
                      i / (float) NUM_PART_EVALS)
    );
}
…
```

PARTIAL





#2 Wrap the partial fitness eval. and merge functions:

PARALLEL

```
…
peoParaPopEval<Route> ox_pop_eval( part_eval, merge_eval );
```

Outline

- Contributions
- An unifying view of three parallel hierarchical levels
- ParadisEO-PEO Evolutionary Algorithm Components
- Low Level Relay GA ↔ HC
- High Level Relay GA + HC
- High Level Coevolutionary Island Model
  - HL Coevolutionary – Asynchronous Island Model
  - HL Coevolutionary – Synchronous Island Model
- ParadisEO-PEO – Parallel Models
  - The Parallel Evaluation of the Population
  - Parallel Evaluation of the Objective Function
- **Conclusions**

Conclusions

ParadisEO-PEO is a white-box Object Oriented Framework

- **clear conceptual separation of solution methods and problems**
- **maximum design and code reuse**
- **high flexibility - fine-grained EO objects**

ParadisEO-PEO provides a broad range of features

- **evolutionary algorithms, local searches and natural hybridization mechanisms (invariant parts provided).**
- **various transparent and portable parallel/distributed models**

Experimental evaluation on academic problems and industrial applications

- **high reuse capabilities, efficacy of the parallel/distributed models**