

Algorithmes génétiques et calcul haute performance

Jean-Charles Boisson CReSTIC-SysCom – EA 3804



- Algorithmes génétiques (AGs) :
 - Positionnement
 - Coopération avec d'autres méthodes

- AGs et parallélisme :
 - Schéma de parallélisation
 - Architectures utilisées

- Impact du multi-objectif

- Aide à la mise en place d'AGs : ParadisEO

Google (scholar) fight

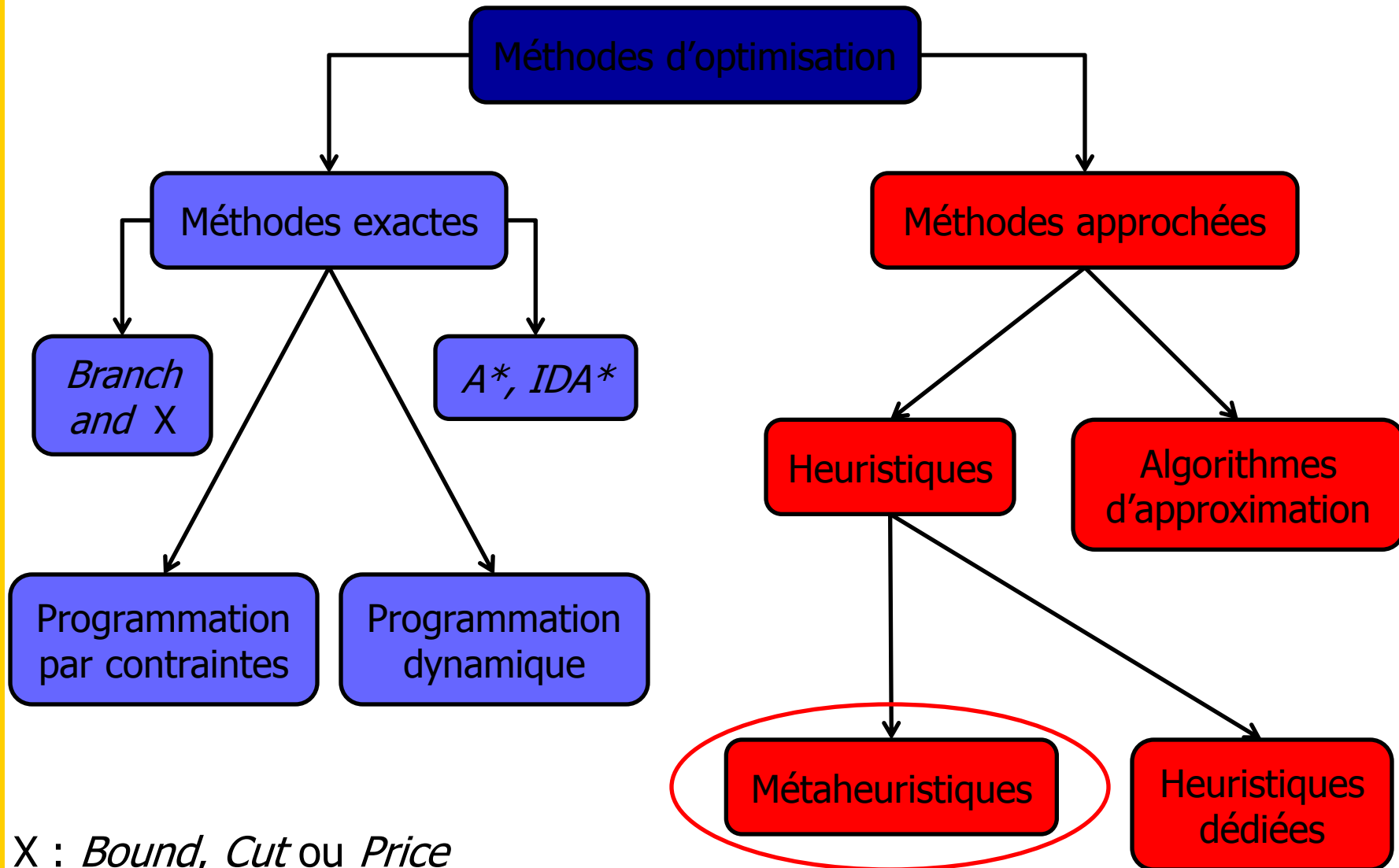
■ Hill climbing	Local search 3 150 000	191 000
■ Tabu search		104 000
■ Simulated annealing		717 000
■ Particle Swarm Optimization		68 700
■ Ant Colony Optimization		291 000

Genetic Algorithm → 1 590 000

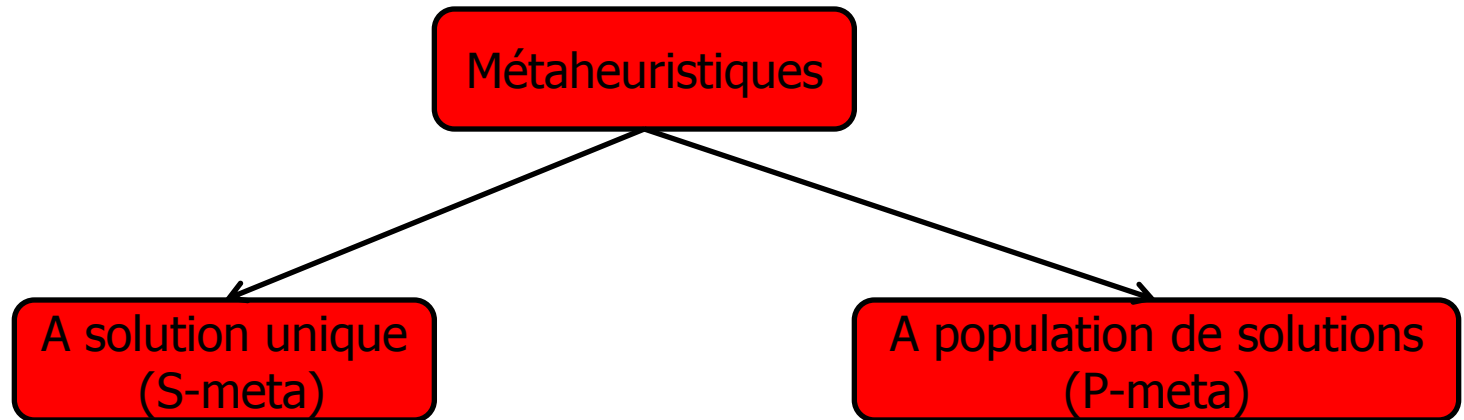
La (merveilleuse) histoire des algorithmes génétiques

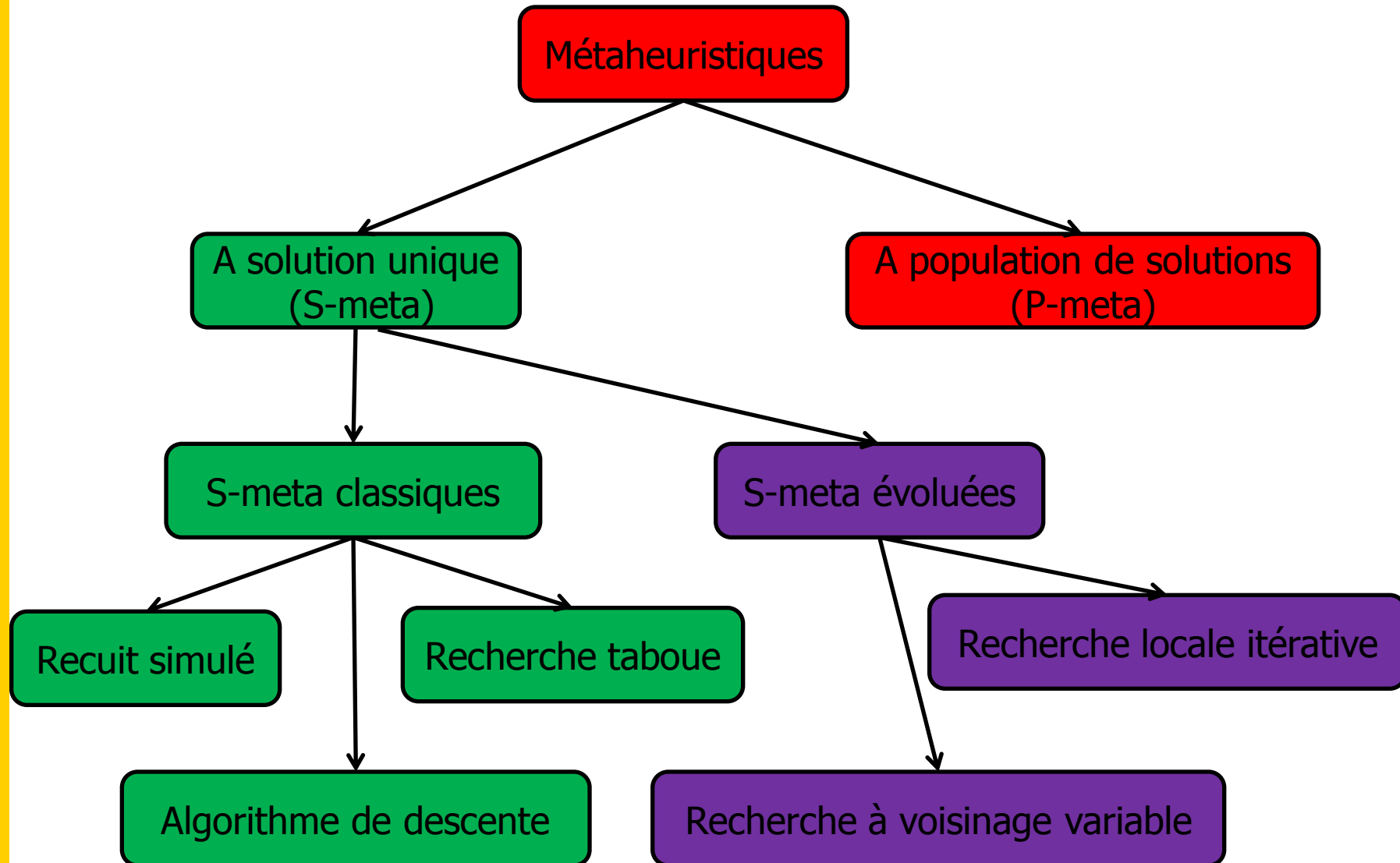


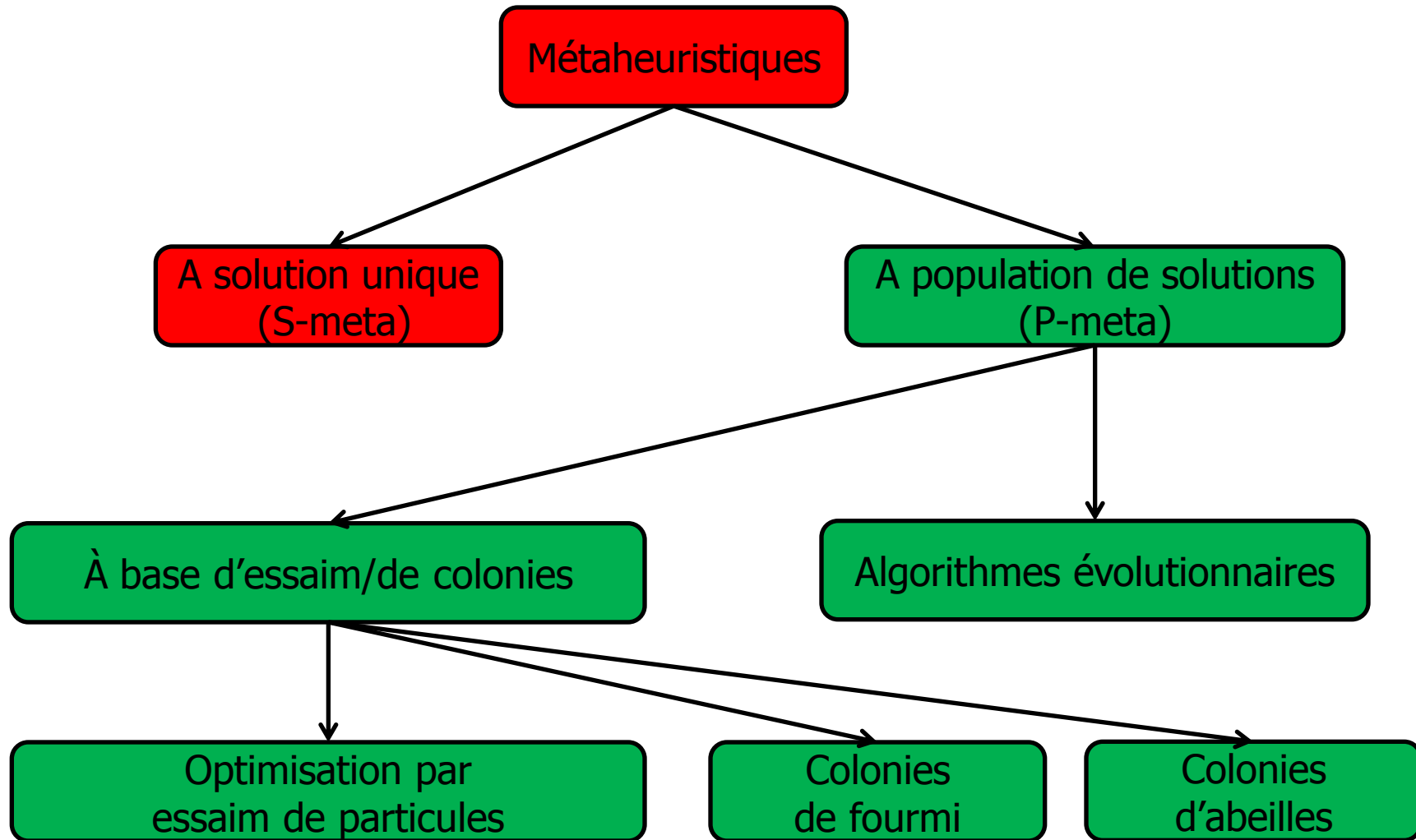
Méthodes d'optimisation

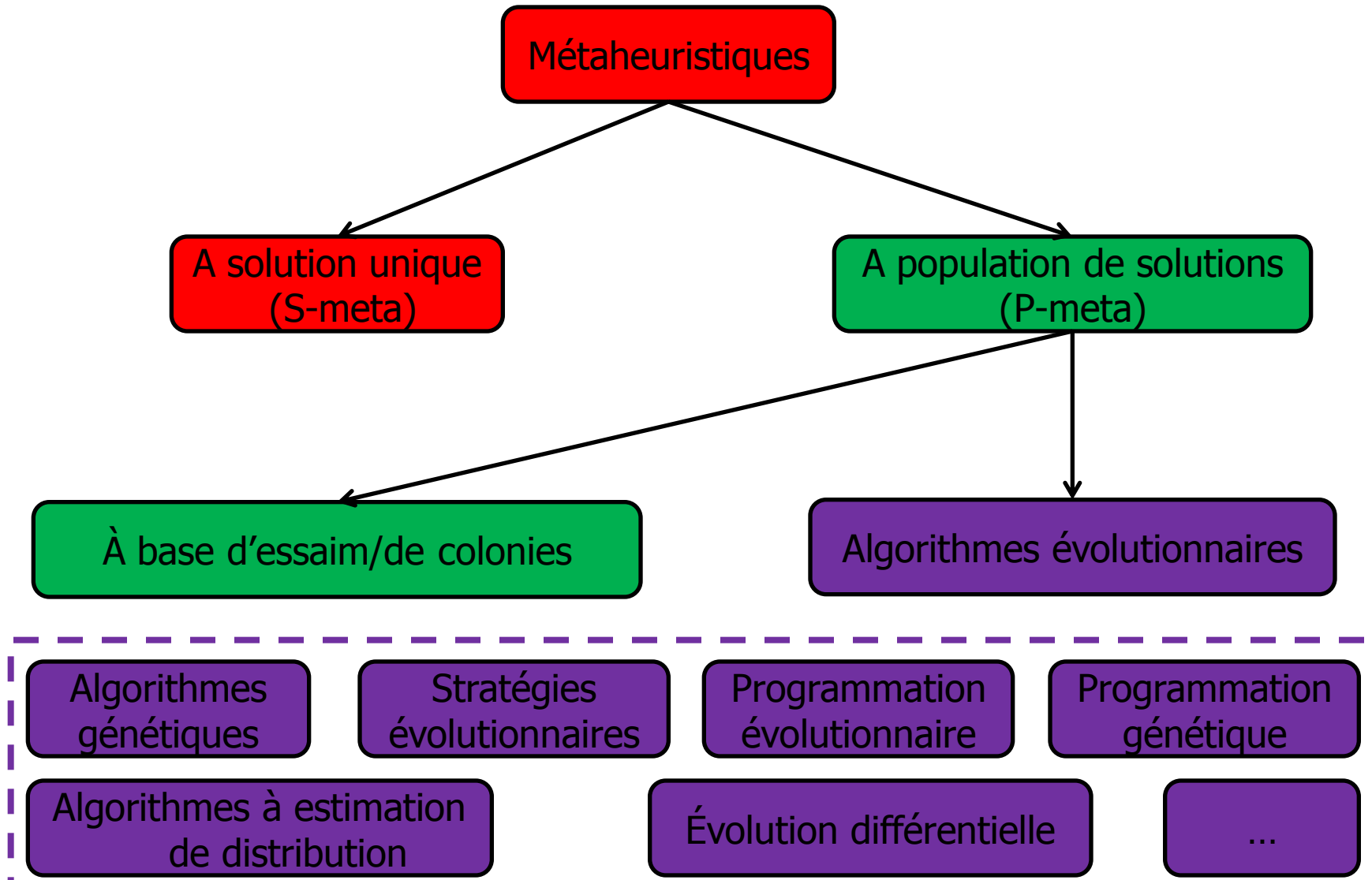


X : Bound, Cut ou Price









Choix de la méthode d'optimisation

- Comment choisir **objectivement** la méthode à utiliser ?
- Étape 1 : définir la représentation d'une solution
 - Mot binaire
 - Ensemble de valeur discrètes/réelles
 - Permutation
 - Spécifique au problème ?
 - Taille fixe ou variable ?

Choix de la méthode d'optimisation

- Comment choisir **objectivement** la méthode à utiliser ?
- Étape 2 : définir l'évaluation d'une solution
 - Coût temps/espace d'une évaluation standard
 - Coût temps/espace d'une « delta » évaluation
 - Connaissant l'évaluation d'une solution et la modification apportée pour obtenir une nouvelle solution, peut-on déduire efficacement la nouvelle évaluation ?

Choix de la méthode d'optimisation

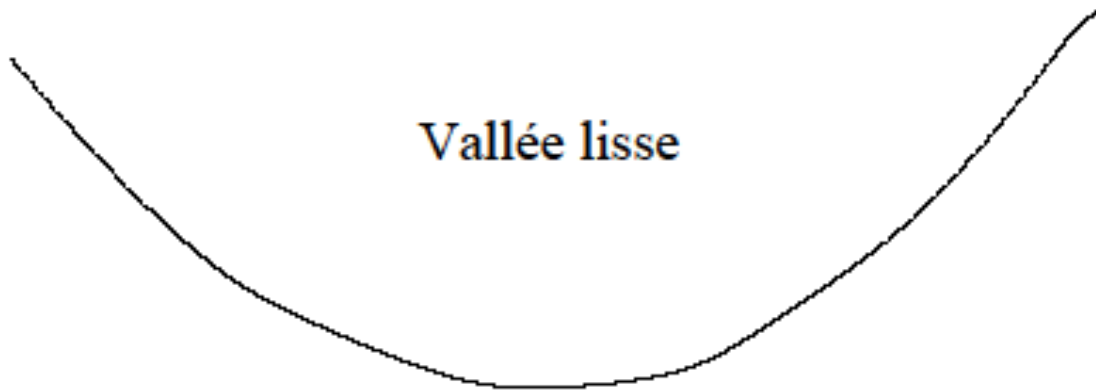
- Comment choisir **objectivement** la méthode à utiliser ?
- Représentation + évaluation → **paysage**
- Basé sur les notions de :
 - voisinage d'une solution
 - distance entre solutions

Exemple de paysages (2D)

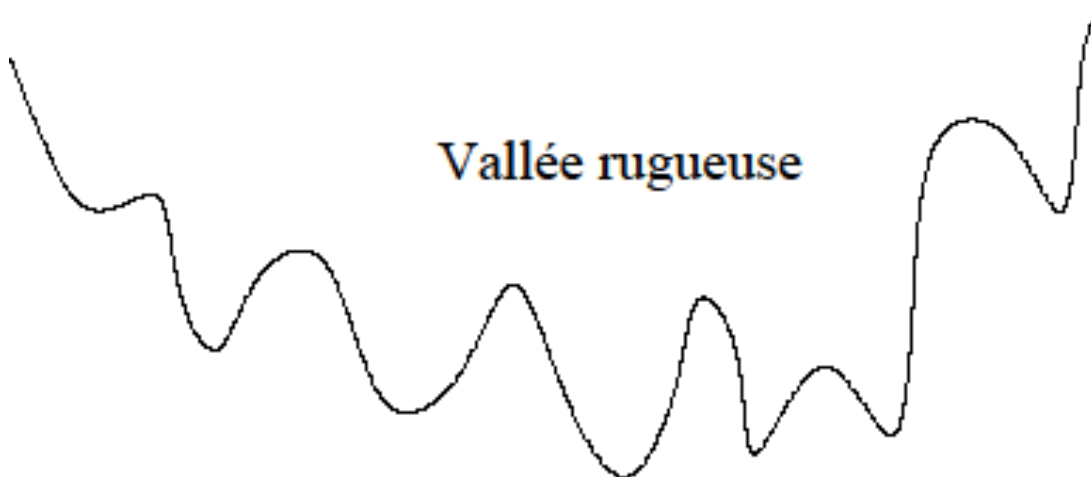
Plaine lisse



Vallée lisse



Vallée rugueuse



- Métaheuristiques à base de population de solutions (P-Meta).

- Défini par J.-H. Holland (1962) :

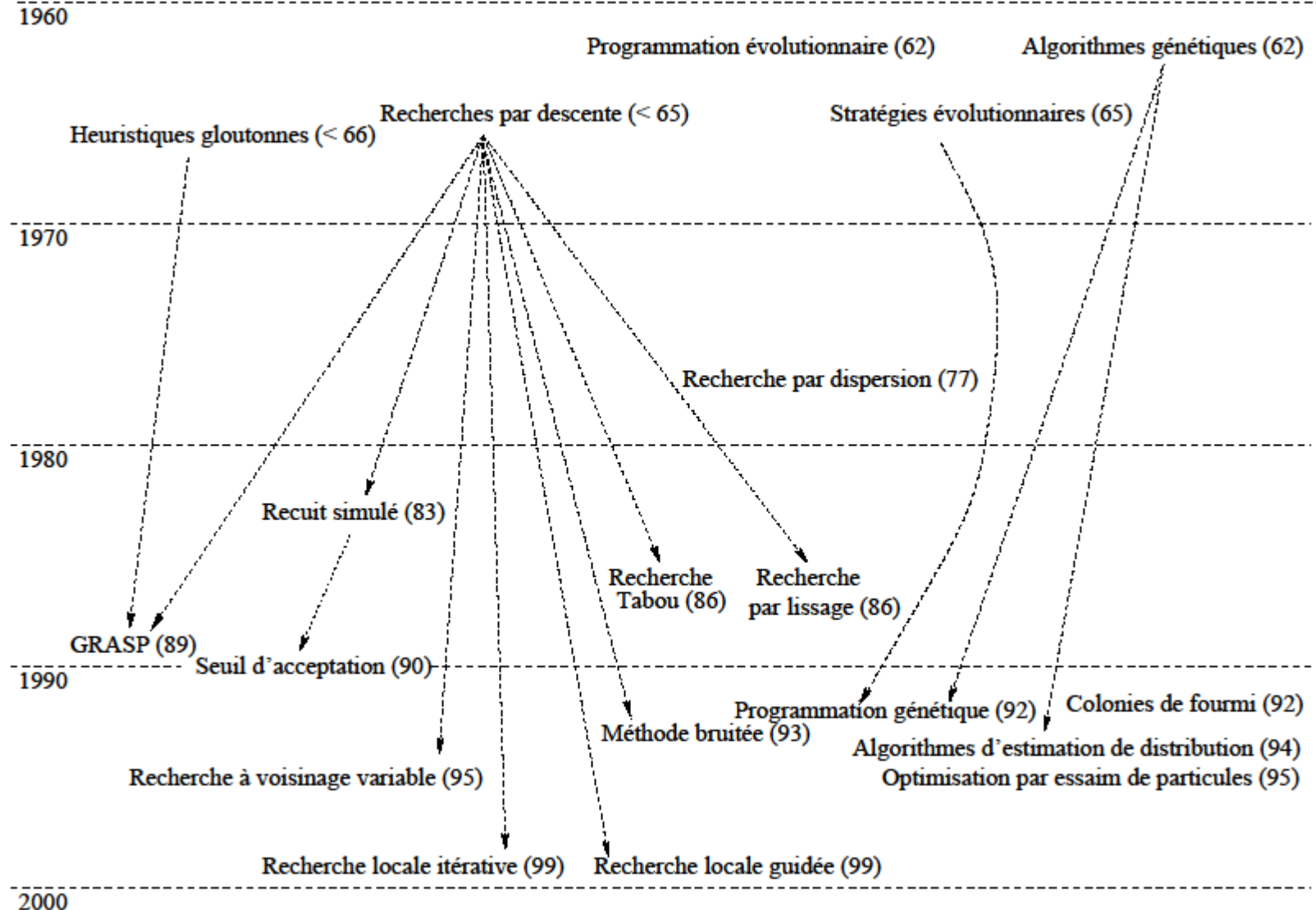
J.-H. Holland. **Outline for a logical theory of adaptive systems.** Journal of the ACM , 3:297-314, 1962.

- Référence (1975) :

J.-H. Holland. **Adaptation in Natural and Artificial Systems.** University of Michigan Press, Ann Arbor, MI, 1975.

- Autres P-Meta > 1990

Historique des méthodes d'optimisation

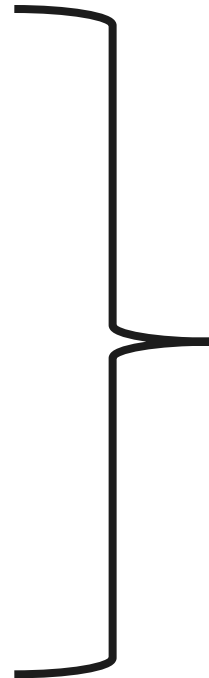


E.-G. Talbi **Metaheuristics : From Design to Implementation**, Wiley, 2009

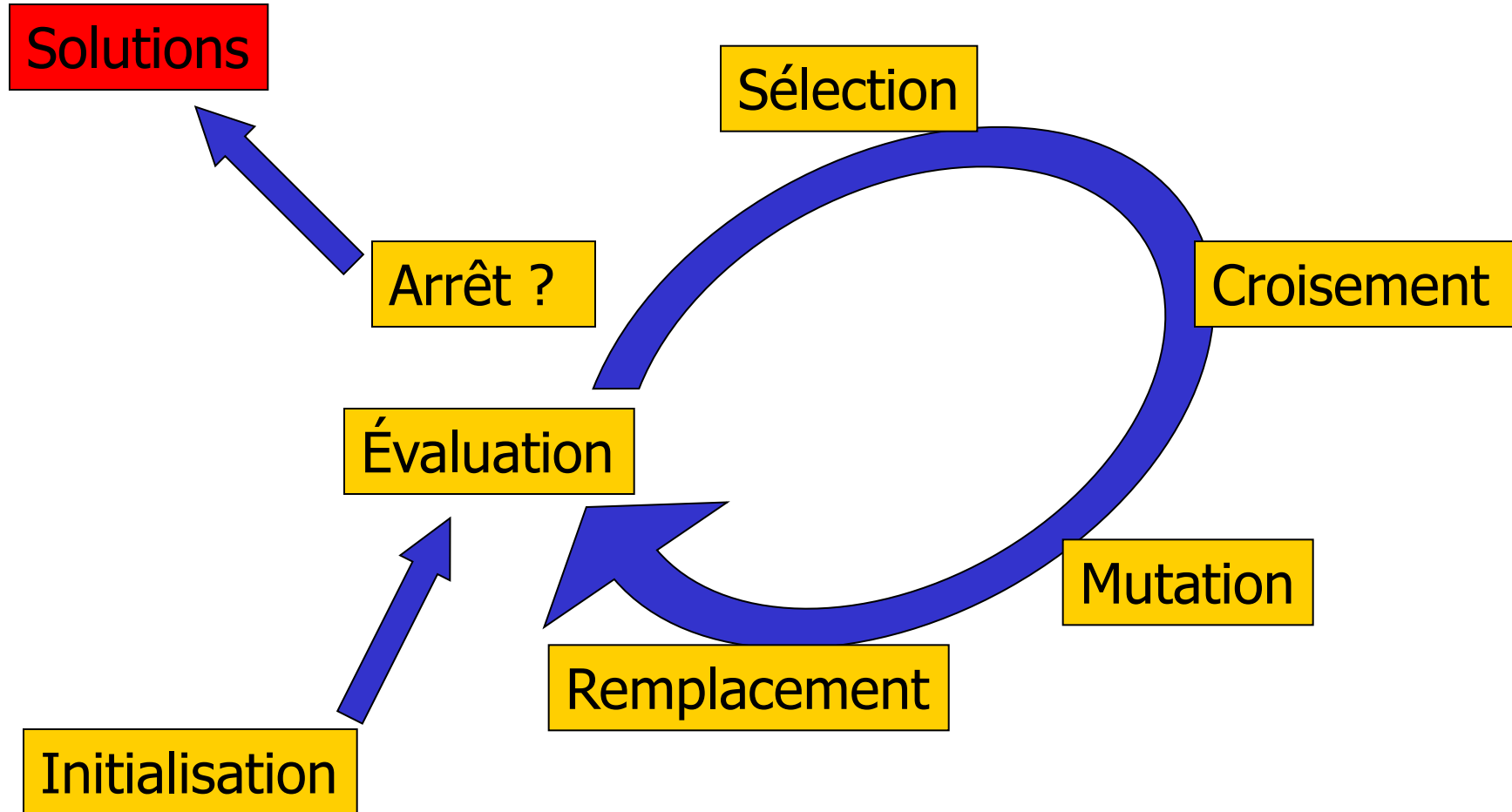
- Basés sur les mécanismes évolutionnistes de Darwin.

- Notions de :

- Sélection
- Croisement
- Mutation
- Remplacement



EVOLUTION



- Donner un ensemble de solutions réalisables.
- Être capable d'explorer de très grands espaces de recherche
- Ne pas être gêné par des optimaux locaux
- Être robuste

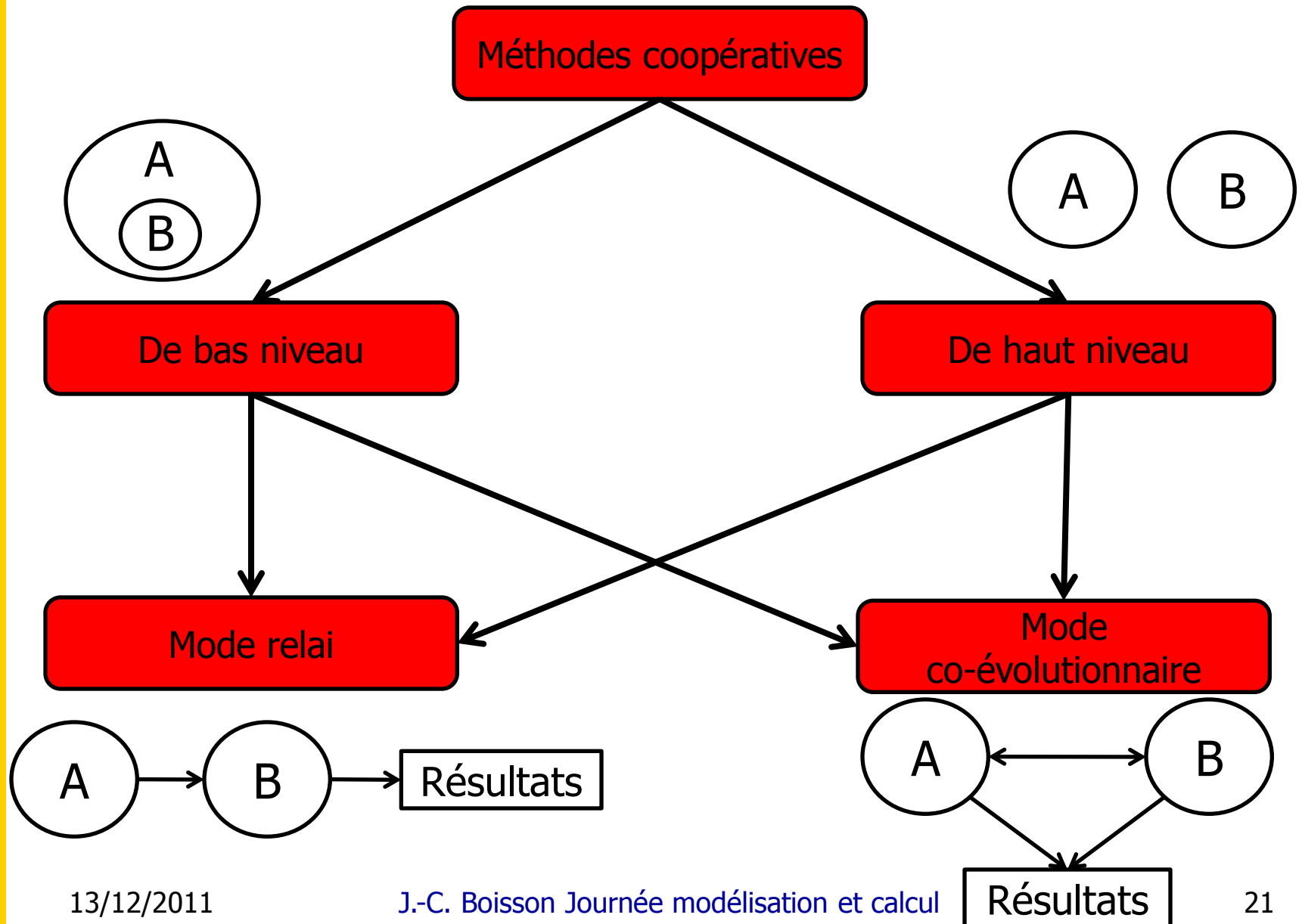
- Être difficilement paramétrables
- Peuvent être très long à converger
- Inversement peuvent converger prématurément
- Manquent de pouvoir d'intensification

Améliorer le comportement d'un AG

- Être difficilement paramétrable
 - « tuning » des paramètres sur un ensemble d'instances du problème.
- Long à converger
 - Accélération du processus d'évaluation
- Convergence prématurée
 - Amélioration de la diversité
- Manque de pouvoir d'intensification
 - Coopération avec d'autres méthodes

Coopération de méthodes

Exemple de taxonomie

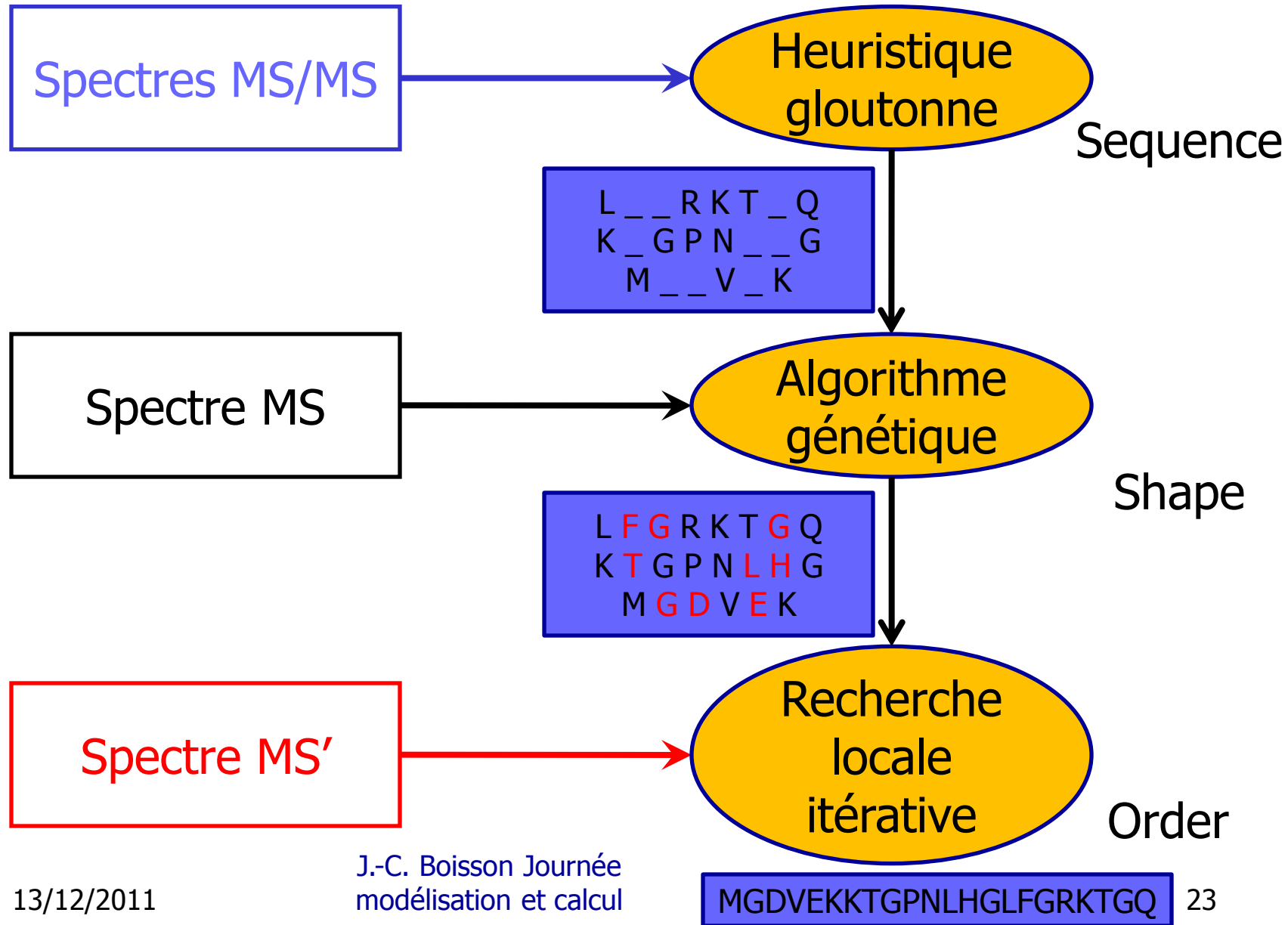


- En mode relai / haut niveau :
 - Génération de la population initiale
 - Optimisation de la population finale

- En mode co-évolutionnaire / bas niveau :
 - Associé une (des) méthode(s) d'intensification dans les opérateurs d'évolution.

- Attention au coût associé !!!

Coopération relai de haut niveau



« Vers l'infini et au-delà »
*(réflexion concernant le temps d'obtention des
solutions en séquentiel)*



Pourquoi paralléliser un AG ?

- Pour une configuration type
 - améliorer le temps de résolution

- Pour une limite de temps donnée
 - Être capable de gérer de plus grandes quantités de données :
 - Population plus grande
 - Plus grand nombre de générations
 - Opérateur d'évolution plus coûteux

Sur quelle architecture paralléliser ?

- Sur ordinateur multi-cœur :
 - Partage natif de la mémoire et du système de fichiers.
 - Une tâche (ou plus cf multi-threading) par cœur.
- Sur plusieurs nœuds de calcul :
 - Dans une même machine \Leftrightarrow cluster :
 - Chaque nœud a sa mémoire propre et son système de fichier
 - Un système de fichier réseau peut être monté
 - Sur des machines différentes \Leftrightarrow grille :
 - Chaque machine a sa mémoire propre et son système de fichier
 - Communication réseau entre les machines

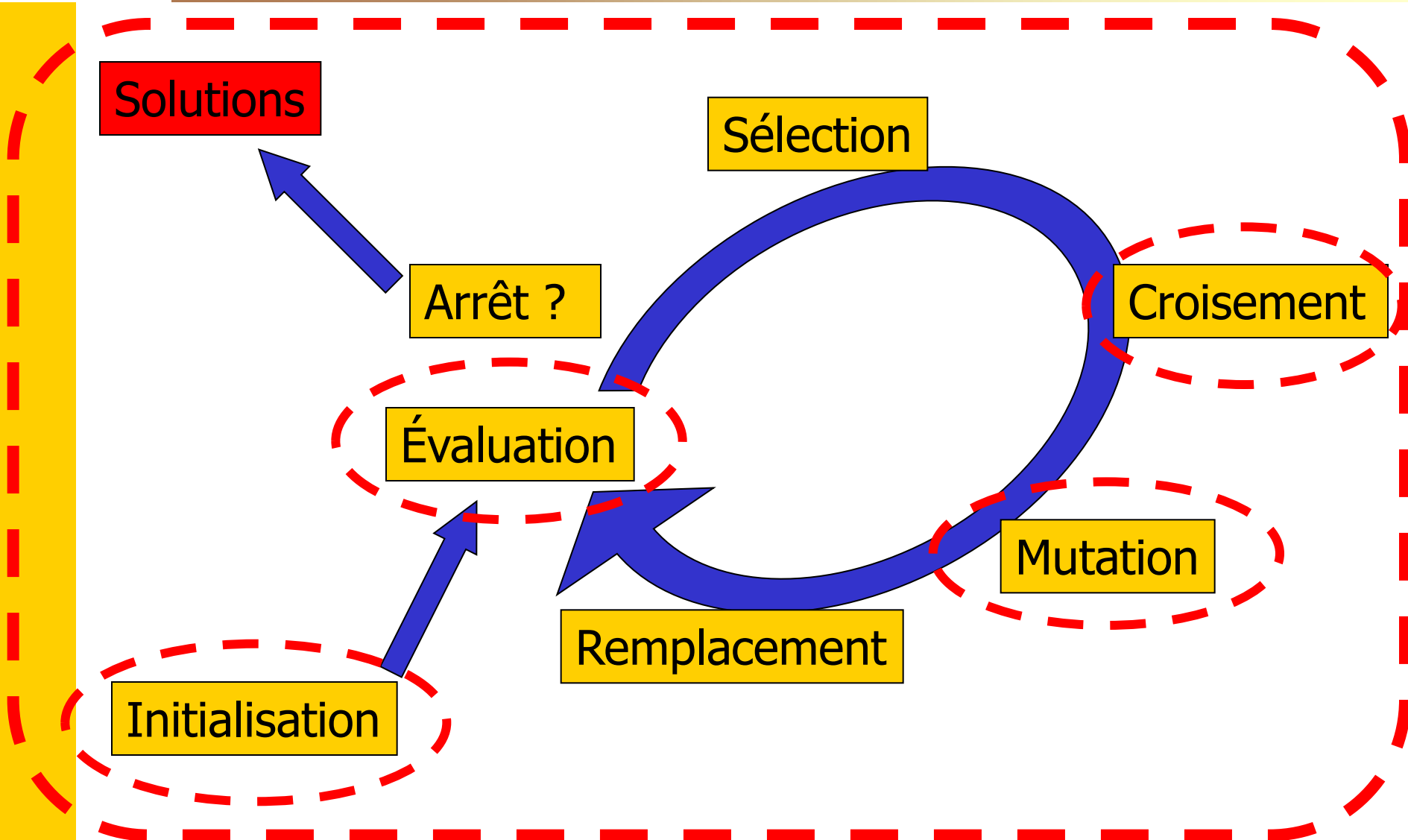
Sur quelle architecture paralléliser ?

- Graphic Processing Unit (GPU) :
 - Composant initialement dédiée à la gestion des calculs graphique
 - Très proche du matériel pour une exploitation optimale
 - Pas forcément dédié à toutes les tâches
 - Fortement limitée en mémoire
 - Échanges (très) lent avec le CPU et la mémoire de la machine hôte de la carte

Sur quelle architecture paralléliser ?

- Field Programmable Gate Arrays (FPGA) :
 - Programmer directement ses circuits pour le traitement que l'on désire faire.
 - Instruction indépendante en parallèle
 - Nécessite une connaissance spécifique de ce type d'architecture \Leftrightarrow langage de description matériel
 - Dispose de peu de mémoire embarqué
 - N'est pas adaptable à tout type de traitement.

Que paralléliser dans un AG ?



Les bonnes questions à se poser

- Ma fonction d'évaluation est-elle coûteuse ?
- Mes données prennent-elles de la place en mémoire ?
- Est-ce qu'il y a beaucoup de lecture/écriture sur le système de fichier ?
- Quel est l'objectif de mon approche ?

Les bonnes questions à se poser

- Ma fonction d'évaluation est-elle coûteuse ?
 - Savoir identifier les parties les plus coûteuse d'une approche
 - Paralléliser l'évaluation de la population :
 - Distribution des individus sur un ensemble de ressources
 - Paralléliser l'évaluation d'un seul individu :
 - Découpage en sous-tâche d'évaluation
 - Distribution des sous-tâches et calcul
 - Récupération des résultats reconstruction du résultat global

Les bonnes questions à se poser

- Mes données prennent-elles de la place en mémoire ?
 - Suivant l'architecture la mémoire disponible n'est pas la même.
 - Impossible de charger autant de données sur une carte GPU que dans la mémoire d'un ordinateur
 - Coût important de transférer les données de la mémoire d'une machine vers une autre machine

Les bonnes questions à se poser

- Est-ce qu'il y a beaucoup de lecture/écriture sur le système de fichier ?
 - Si tous les cœurs d'un processeur lisent/écrivent sur le système de fichier en même temps
 - Si différentes nœuds lisent/écrivent sur un système de fichier réseau
 - Si différentes machines écrivent sur leur système de fichier avant de faire une synchronisation

→ Chute des performances

Les bonnes questions à se poser

- Quel est l'objectif de mon approche ?
 - Trouver la solution d'un problème non encore résolu
 - Produire une solution logicielle à finalité de production
 - Mettre en place une méthode plus rapide/efficace que l'existant

Les bonnes questions à se poser

- Quel est l'objectif de mon approche ?
 - Trouver la solution d'un problème non encore résolu :
 - Instance très grande à résoudre
 - Peu d'exécution mais a priori très longue
 - Gourmande en moyen de calcul
 - Ratio temps de développement/utilisation : 10/90

Les bonnes questions à se poser

- Quel est l'objectif de mon approche ?
 - Produire une solution logicielle à finalité de production :
 - Respect de contraintes (temps de calcul, taille mémoire, ...)
 - Nombreuses exécution plus ou moins longue
 - Adaptée à un niveau de ressources disponibles
 - Ratio temps de développement/utilisation : 50/50

Les bonnes questions à se poser

- Quel est l'objectif de mon approche ?
 - Mettre en place une méthode plus rapide/efficace que l'existant :
 - Exploitation au mieux des ressources disponibles
 - Très nombreuses exécutions
 - Schéma de déploiement adaptable aux ressources
 - Ratio temps de développement/utilisation : 90/10

Le cas multi-objectif
*(ou comment avoir tout ce que l'on a toujours
voulu d'un seul coup)*



Pourquoi du multi-objectif ?

- Classiquement :
évaluer une solution \Leftrightarrow évaluer un critère
- Problème réels :
 - Ensemble de critères pouvant être :
 - Antagonistes :
 - Maximiser une distance parcourue tout en limitant la consommation de carburant.
 - Minimiser le retard de traitement en augmentant le nombre de commandes.
 - Complémentaires :
 - Améliorer l'aérodynamisme et la durée de vol sans moteur
 - Minimiser l'énergie total d'un système moléculaire ainsi que sa surface accessible au solvant.

- Évaluation d'une solution selon plusieurs critères
→ agrégation de terme pondérés
- Avantages :
 - Les méthodes d'optimisation utilisées ne changent pas
 - Nombre de critères à gérer non limité
- Inconvénients :
 - Paramétrage complexe des poids
 - Non rigoureusement valable

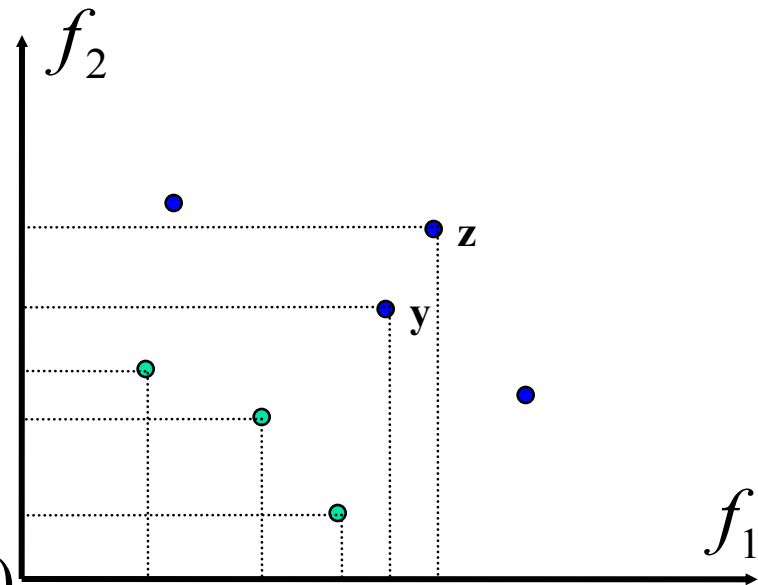
■ Problème multi-objectif (PMO)

$$(\text{PMO}) \begin{cases} \min F(x) = (f_1(x), f_2(x), \dots, f_n(x)) \\ \text{s.c. } x \in C \end{cases} \quad n \geq 2$$

■ Notion de dominance

y domine z ssi $\forall i \in [1..n]$,

$f_i(y) \leq f_i(z)$ et $\exists i \in [1..n] / f_i(y) < f_i(z)$.



● Solution Pareto optimale

● Solution réalisable dominée

- Pareto optimalité \Leftrightarrow une solution $x^* \in C$ est Pareto optimale, ssi il n'existe pas de solutions $x \in C$ telle que x domine x^* .

Qualité d'un algorithme multi-objectif ?

- De par sa définition, le multi-objectif complique un grand nombre de traitement :
 - Comparaison des solutions entre deux approches
 - Qualité de la :
 - Convergence
 - Diversification

➔ notion de front

- Mesures particulières pour la convergence et la diversification :
 - Indicateur de convergence :
 - Contribution
 - e-indicateur
 - Cardinalité
 - Distance
 - Indicateurs de diversité, mesure :
 - De l'espacement (entre les solutions)
 - Du maximum de dispersion
 - De l'entropie
 - Indicateur hybride : hypervolume

- Les AGs sont (une) des méthodes dont les modèles multi-objectif sont les mieux codifiés.
- Modèles connus :
 - NSGA-II

K. Deb, A. Pratap, S. Agarwal and T. Meyarivan. **A fast and elitist multi-objective genetic algorithm : NSGA-II.** IEEE Transaction on Evolutionary Computation, 6(2):181-197, 2002.

- IBEA

E. Zitzler and S. Kunzli. **Indicator-based selection in multiobjective search.** Parallel Solving from Nature, PPSN VIII, 3242:832-842, 2004.

- SPEA2

E. Zitzler, M. Laumanns, and L. Thiele. **SPEA2: Improving the strength Pareto evolutionary algorithm.** Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001



Les AGs pour les nuls
*(mais connaissant quand même le C++, les
templates, la POO, ...)*



- Exemple (**subjectif**) de plateforme : ParadisEO
 - Boite « blanche »
 - Extension des bases placées dans **Evolving Object** (1999) \Leftrightarrow P-Meta
 - Ajouts des S-Meta + parallélisation du tout (2001-2005) \rightarrow travail de Sébastien Cahon
 - Ajout du **multi-objectif** (2006-2009)
 - Refonte de la partie S-Meta + S-Meta multi-objectif (2008-2010)
 - Parallélisation sur GPU des S-Meta (2009-2011)

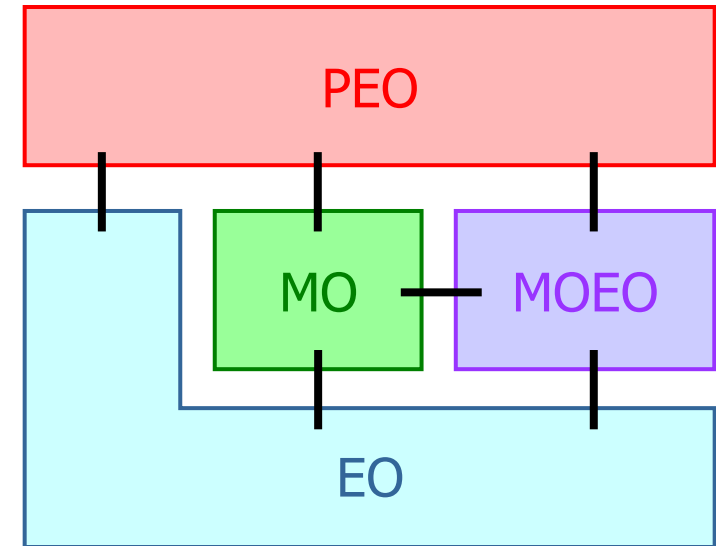
PARAllel and DIStributed Evolving Objects

→ **E**volving **O**bject (**EO**), développement d'algorithmes à base de population de solutions: **EA**, **PSO**.

→ **M**oving **O**bjects (**MO**), mise en place de recherches locales : **HC**, **SA**, **TS**, **ILS**, **VNS**.

→ **M**ulti-**O**bjective **EO** (**MOEO**), développement d'algorithmes évolutionnaires multi-objectifs : **NSGA-II**, **IBEA**, ...

→ **P**aradis**EO** (**PEO**), mise en place de métaheuristiques parallèles.



Comment faire un AG sous ParadisEO

- Pour son problème, définir :
 - Sa représentation de solution (standard ou exotique)
 - L'évaluation d'une solution

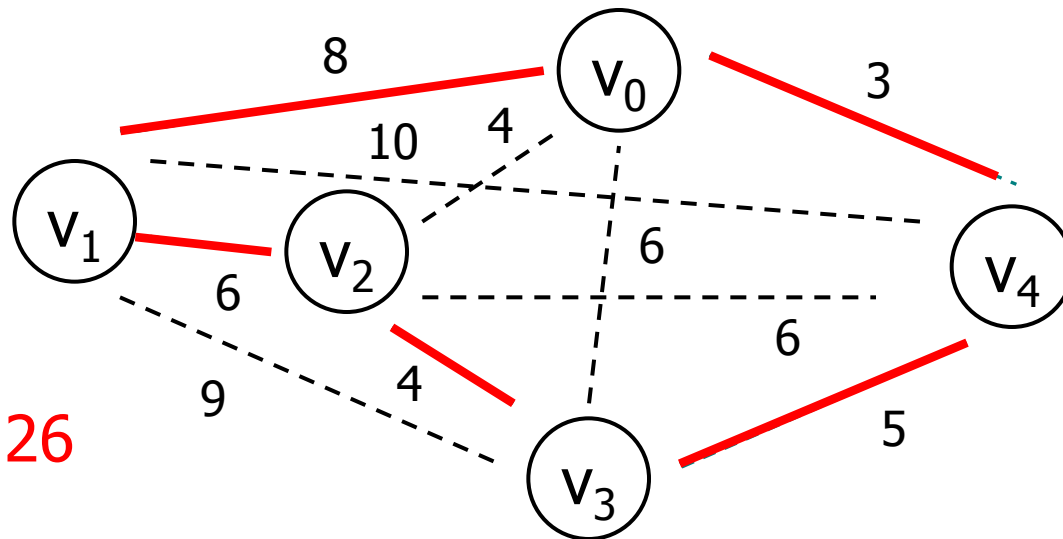
- Sous ParadisEO-EO :
 - Choisir la P-Meta que l'on veut utiliser

 - Comprendre ce que l'on doit définir

 - Choisir les boîtes qui nous intéressent dans les parties génériques.

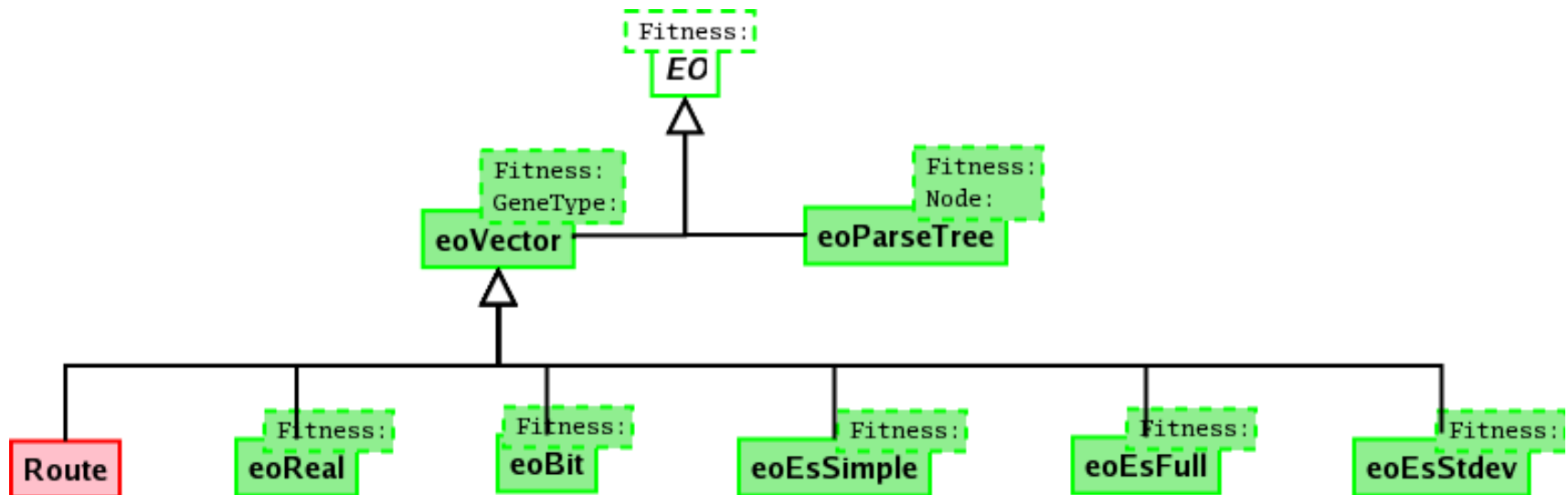
Exemple : le voyageur de commerce

- Soit un ensemble de N villes, connaissant la distance entre toutes ces villes, le problème du voyageur de commerce consiste à trouver le plus court chemin pour toutes les parcourir
- Exemple en version symétrique :

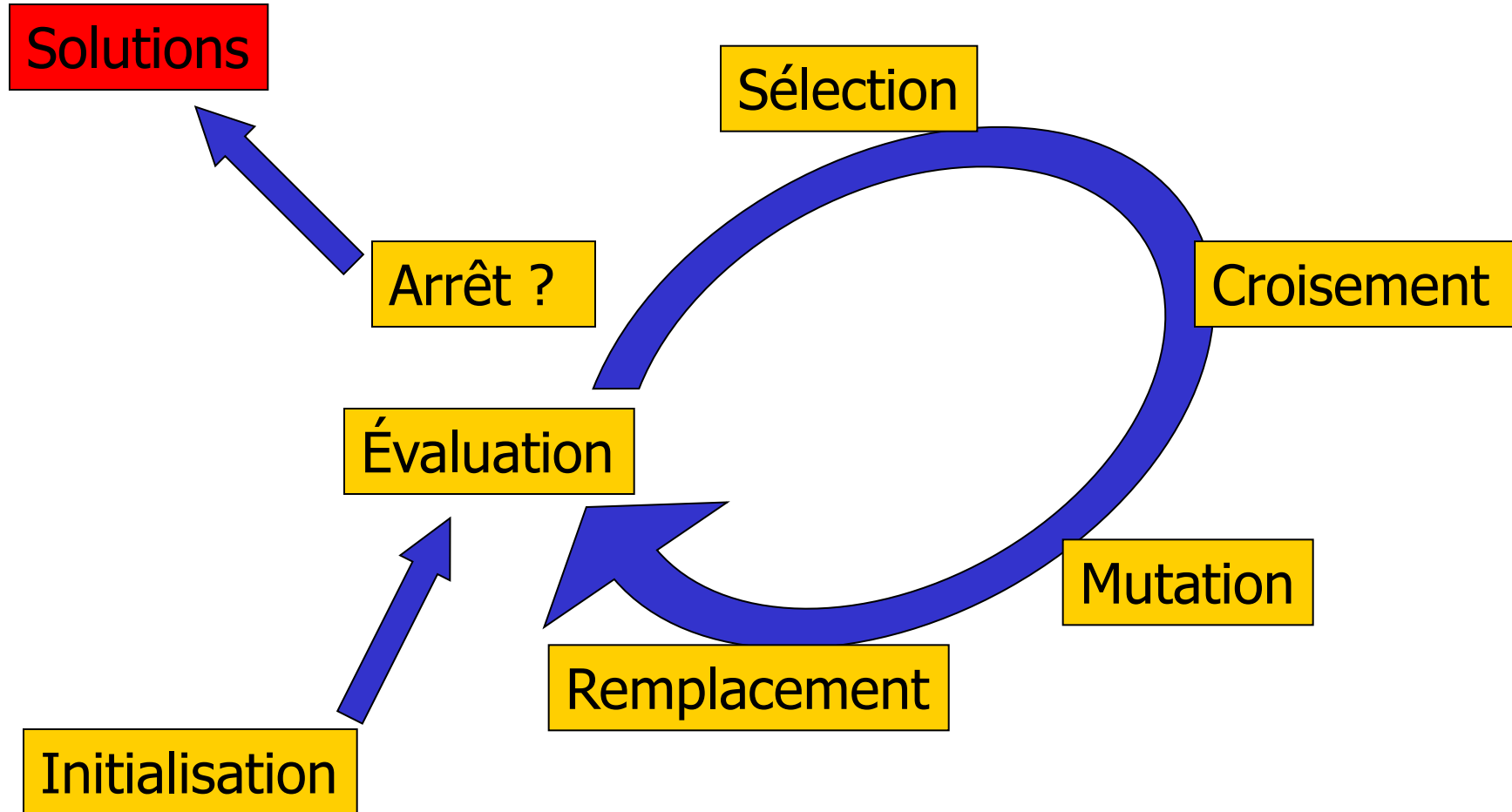


Longueur: 26

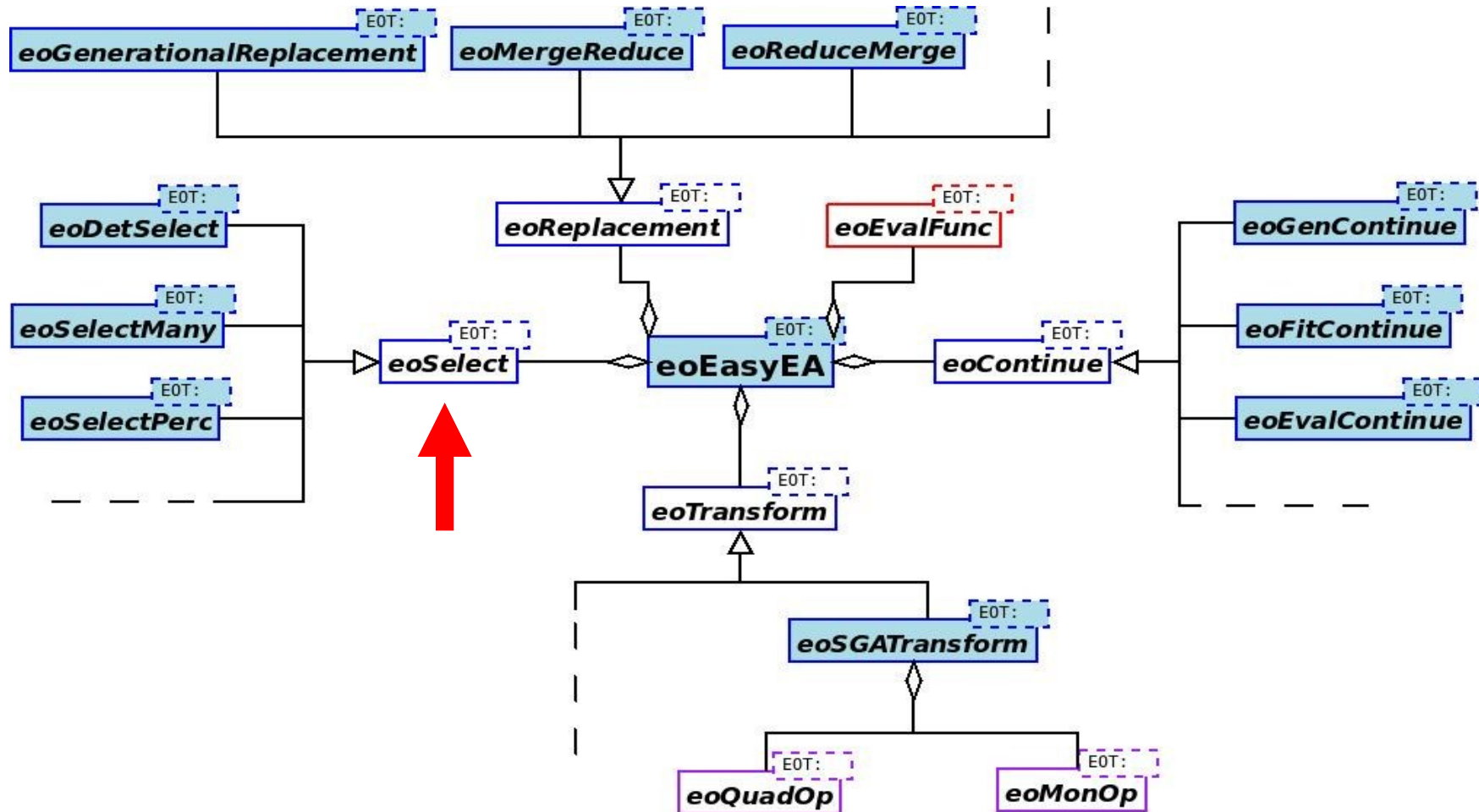
Définition d'une nouvelle représentation



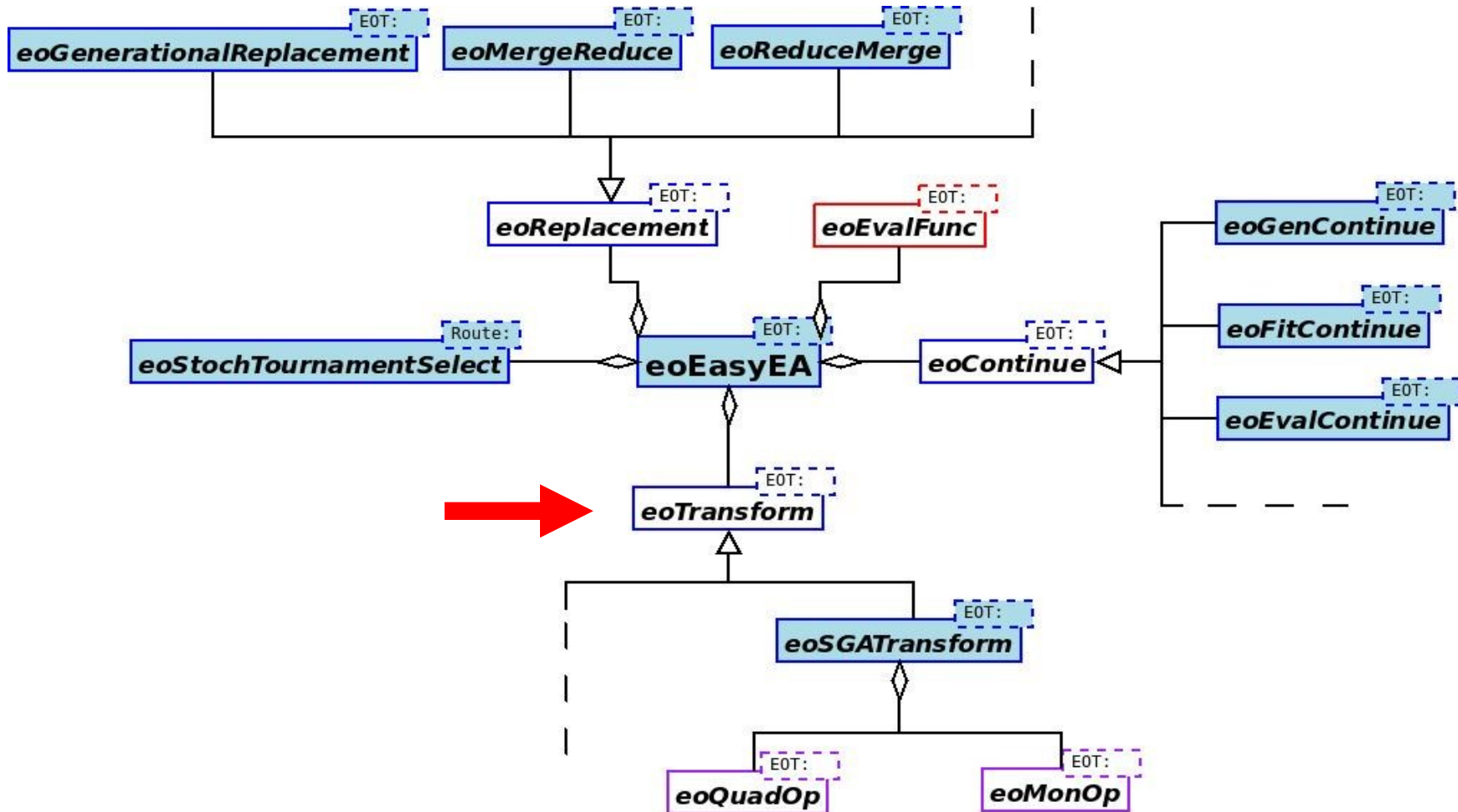
J.-C. Boisson and E.-G. Talbi. **Software Framework for Metaheuristic**. 22nd European Conference on Operational Research (EURO XXII), page 45, July 8-11, Prague, Czech Republic, 2007.



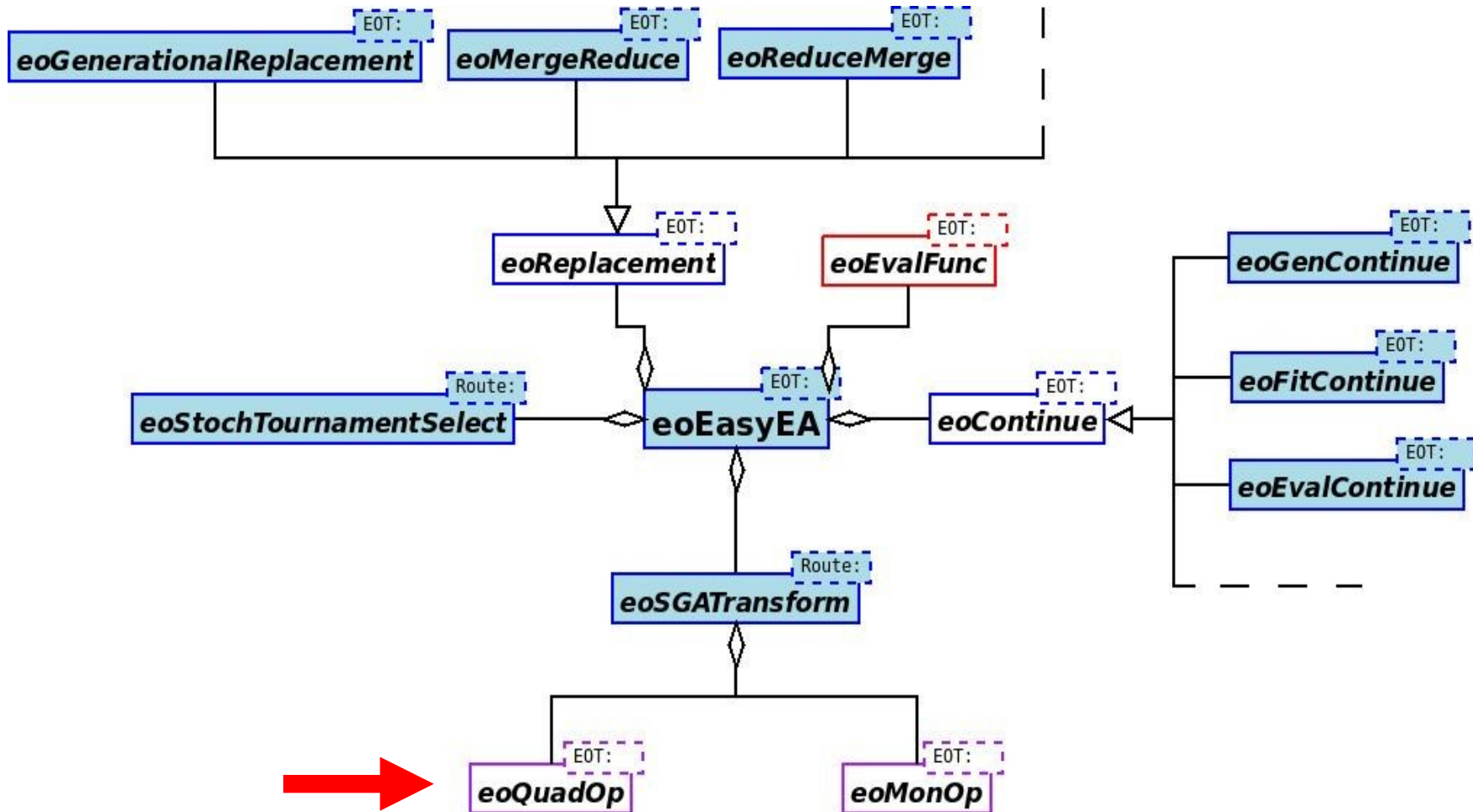
Construction d'un AG sous ParadisEO-EO



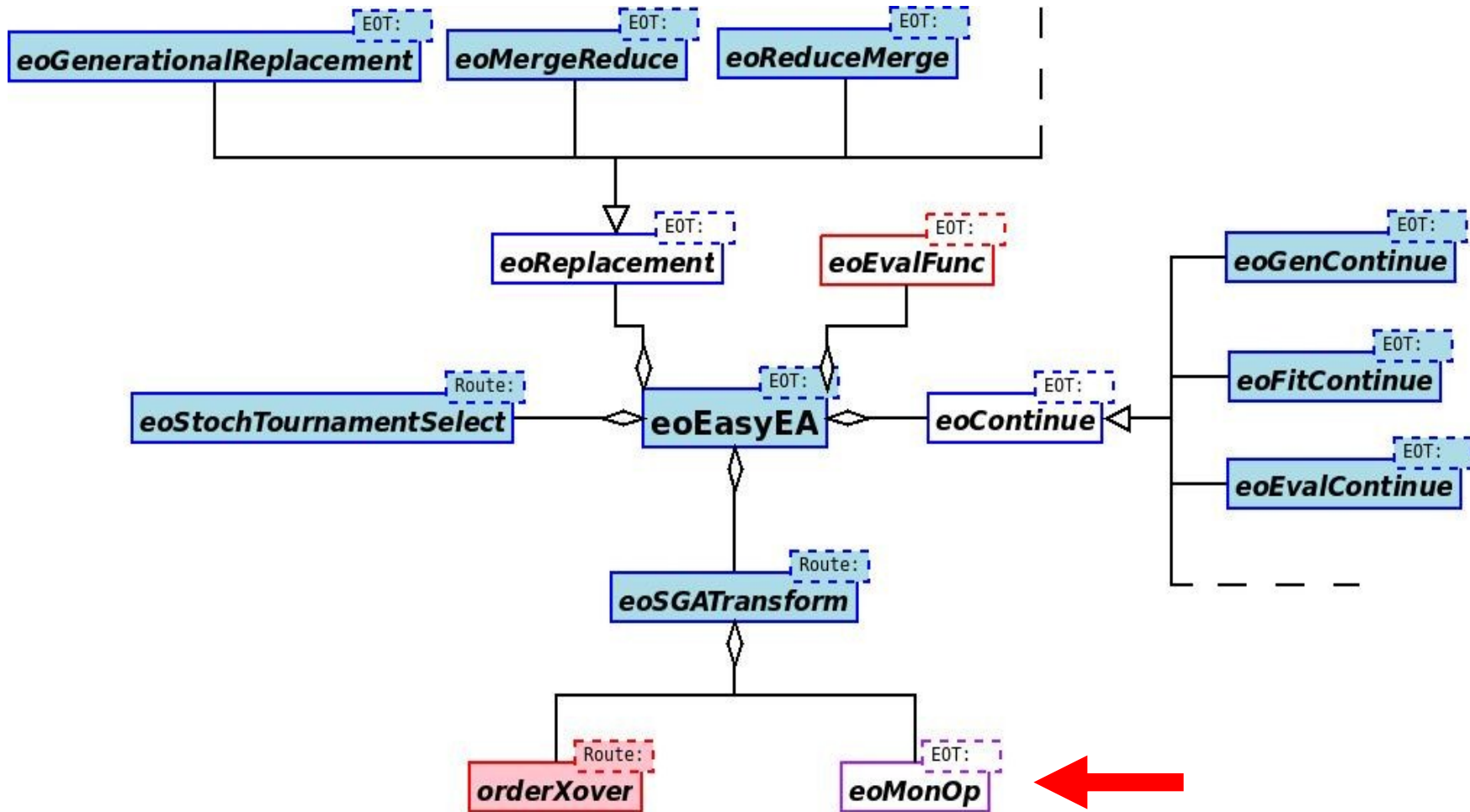
Construction d'un AG sous ParadisEO-EO



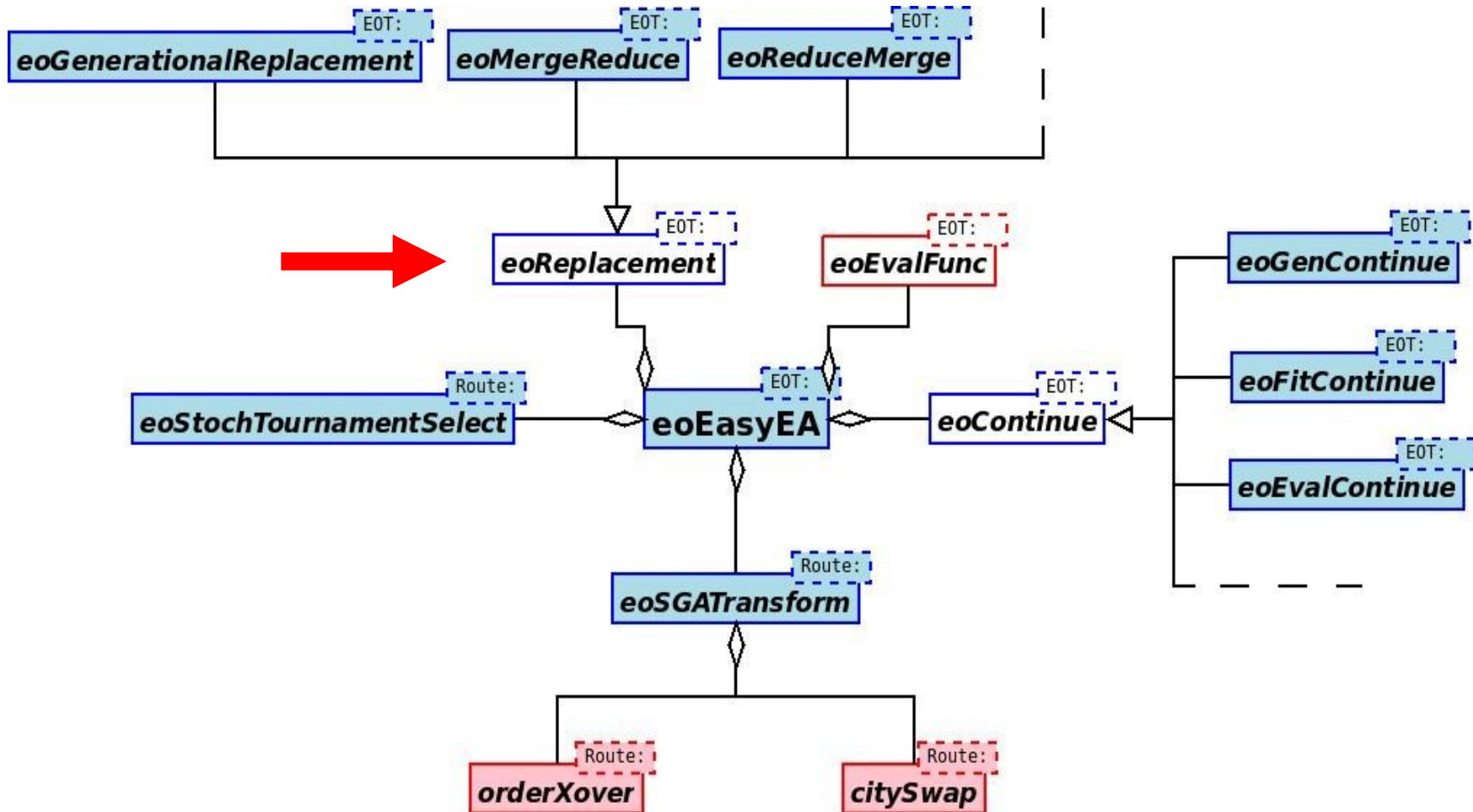
Construction d'un AG sous ParadisEO-EO



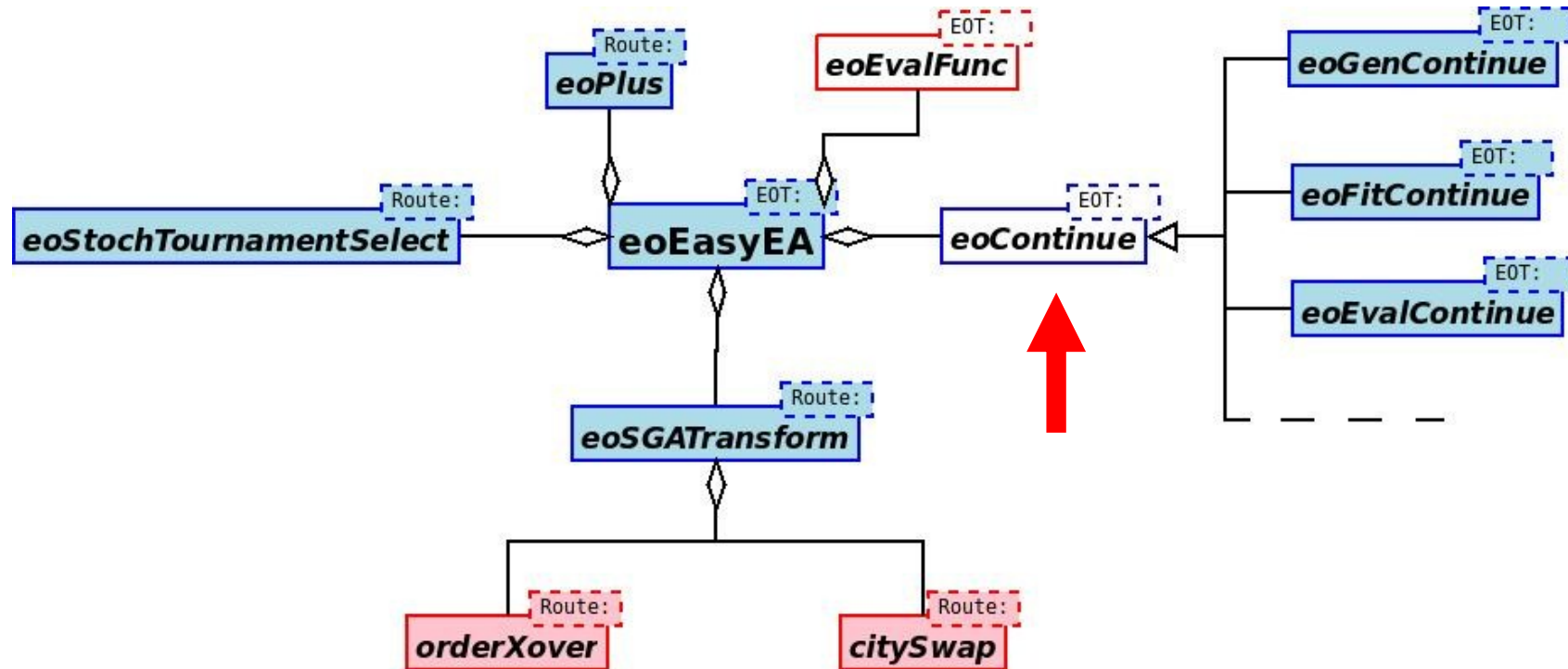
Construction d'un AG sous ParadisEO-EO



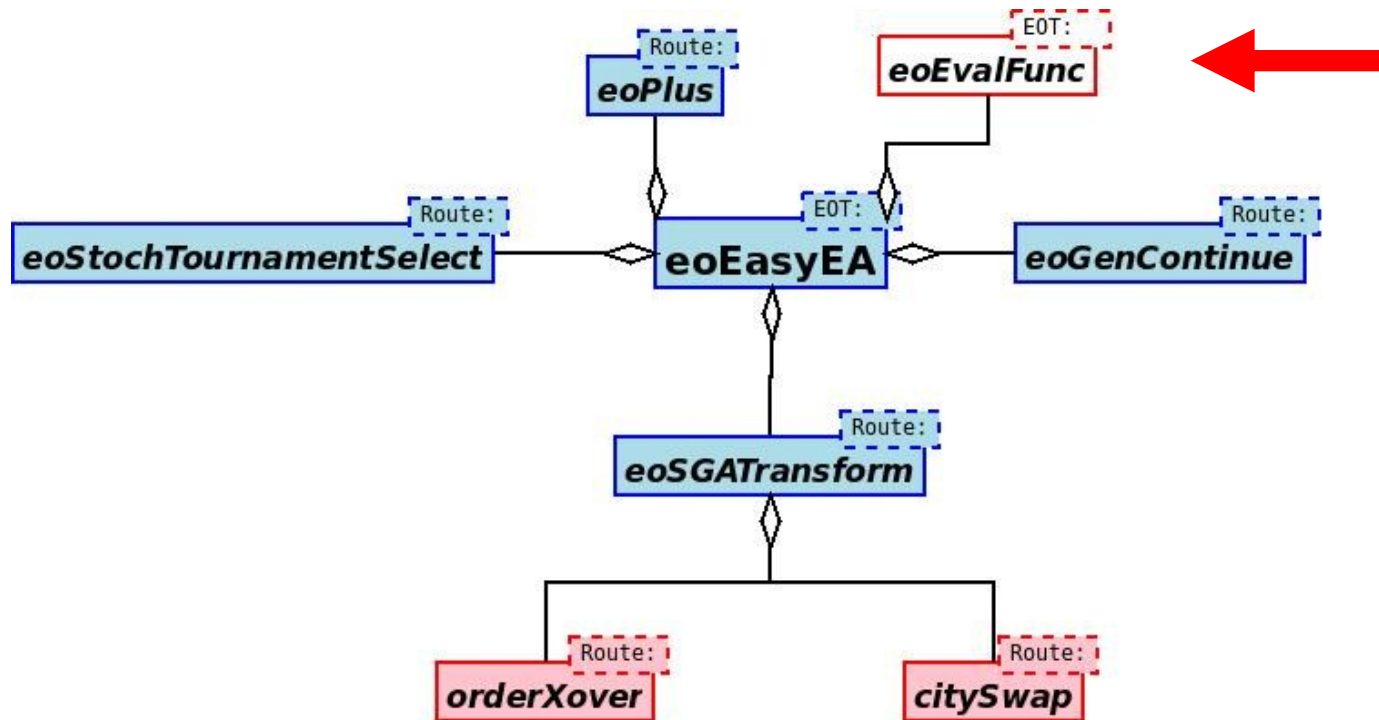
Construction d'un AG sous ParadisEO-EO



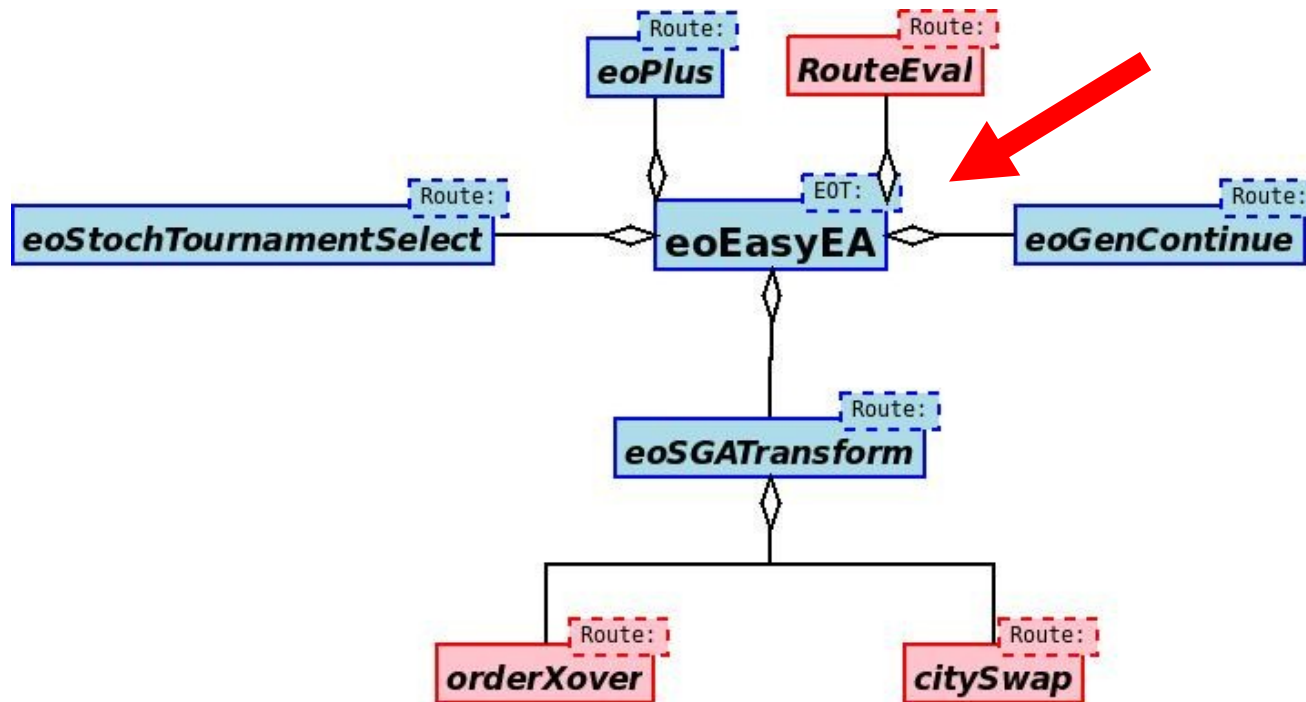
Construction d'un AG sous ParadisEO-EO



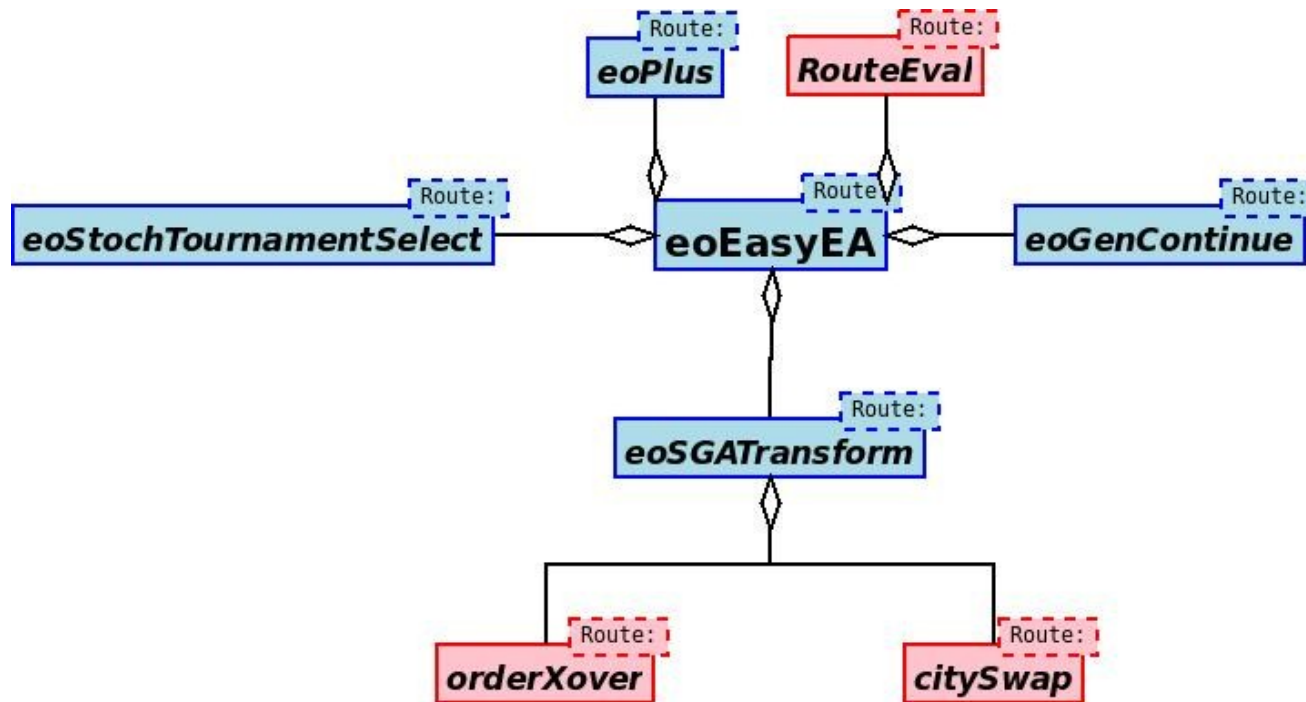
Construction d'un AG sous ParadisEO-EO



Construction d'un AG sous ParadisEO-EO



Construction d'un AG sous ParadisEO-EO



RouteInit route_init;

RouteEval full_route_eval;

eoPop <Route> pop (POP_SIZE, route_init);

eoGenContinue <Route> continue (NUM_GEN);

OrderXover crossover;

CitySwap mutation;

eoStochTournamentSelect <Route> select_one;

eoSelectNumber <Route> select (select_one, POP_SIZE);

eoSGATransform <Route> transform (cross, CROSS_RATE, mutation, MUT_RATE);

eoPlusReplacement <Route> replace;

eoEasyEA <Route> ea (continue, full_route_eval, select, transform, replace);

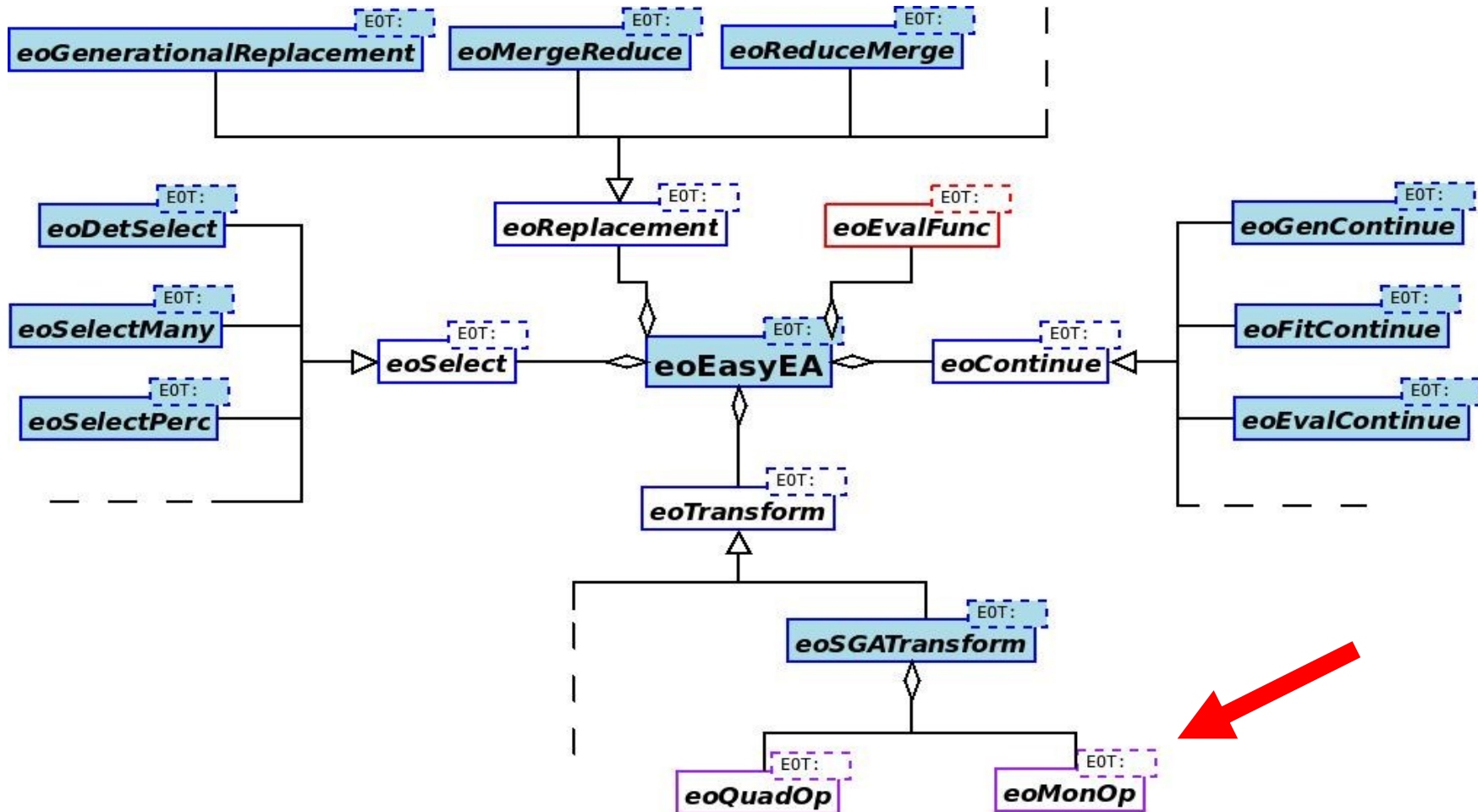
ea (pop);

Comment faire un AG sous ParadisEO

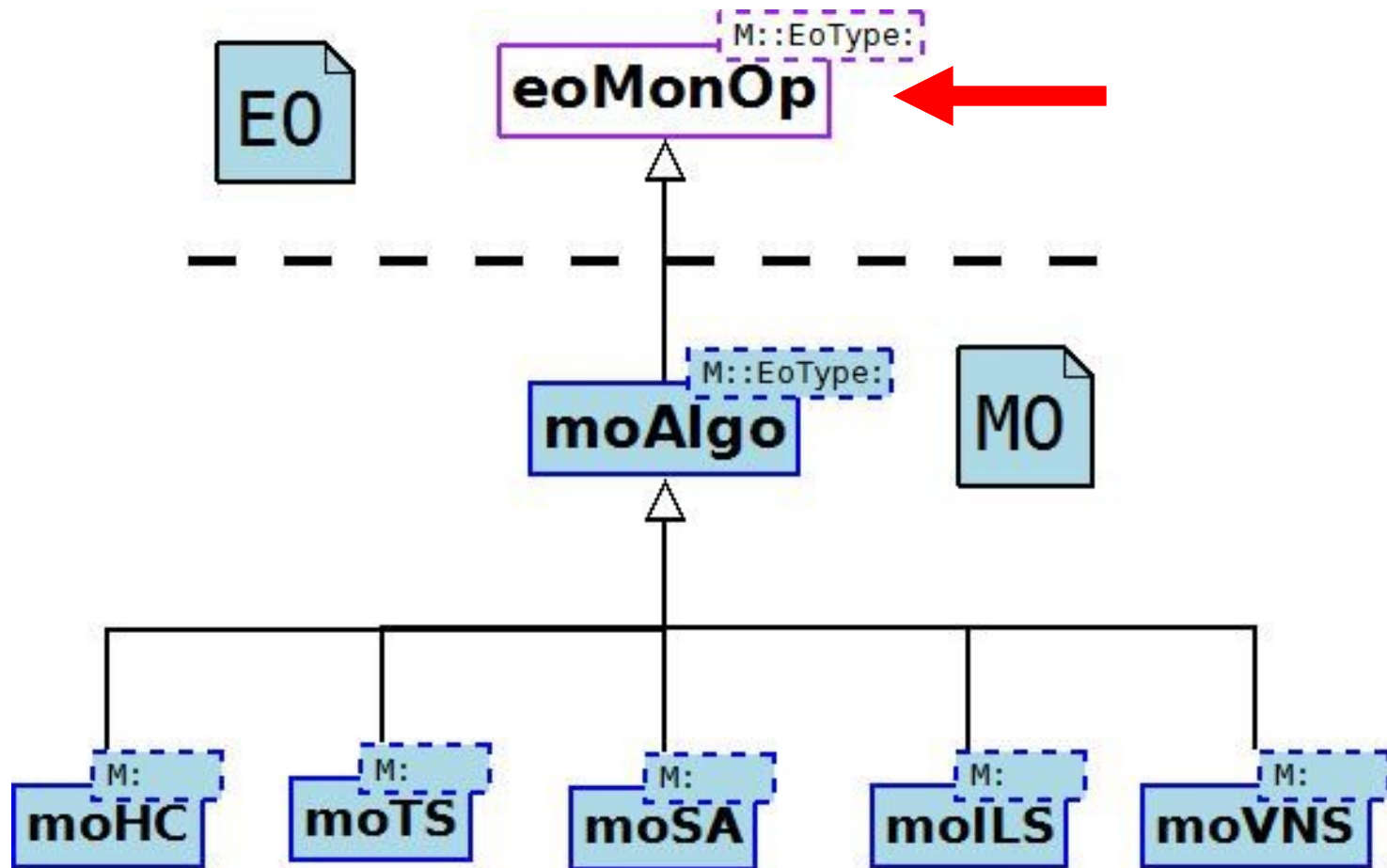
- Pour ajouter de l'intensification :
 - Couplage avec des recherches locales
→ ParadisEO-MO

 - Pour paralléliser le tout et/ou faire des exécutions simultanées :
 - Parallélisation de toutes les étapes d'évolution
 - Modèle maître/esclave
 - Exécution de plusieurs AGs et échange d'informations
 - Modèle insulaire
- ParadisEO-PEO

Détail sur la coopération EO/MO



Détail sur la coopération EO/MO



- Les AGs sont :
 - De très bons outils en recherche opérationnelle
 - Bien définis actuellement pour le cas multi-objectif
 - Capables d'affronter des problèmes réalistes :
 - taille des instances
 - paysage chaotique
 - Aisément parallélisables
 - Hautement paramétrables

- Les AGs ne sont pas :
 - facile à paramétrer
 - si simple à optimiser :
 - Jeu sur les paramètres
 - Coopération avec d'autres méthodes
 - les meilleurs approches pour les problèmes continus
 - forcément nécessaire en premier choix :
 - Recherches locales multi-start

Merci de votre attention