

## TP : ParadiseO

### 1 La plateforme ParadiseO

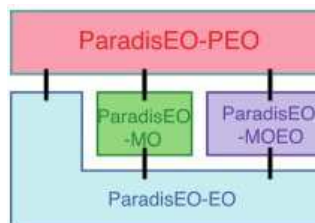
ParadisEO<sup>1</sup> (PARallel and DIStributed Evolving Objects) est une plateforme dédiée à la conception de métaheuristiques. Cette plateforme est développée par l'équipe Dolphin<sup>2</sup> (Optimisation multi-critère parallèle coopérative) de l'INRIA Futurs.

*ParadisEO is a white-box object-oriented framework dedicated to the flexible design of metaheuristics. Based on EO<sup>3</sup> (Evolutionary Computation Framework), this template-based, ANSI-C++ compliant computation library is portable across both Windows system and sequential platforms (Unix, Linux, MacOS, etc.).*

*ParadisEO can be used under several environments as the build process (CMake) is able to generate :*

- Visual Studio 8 2005 (Win32 + Win64) projects
- Visual Studio 7 .NET 2003 projects
- Visual Studio 7 and 6 projects
- NMake Makefiles
- MinGW Makefiles
- Borland Makefiles
- KDevelop projects
- Unix Makefiles
- Xcode Projects
- MSYS Makefiles
- WMake Makefiles

*ParadisEO is composed of several packages that constitute a global framework.*



- *ParadisEO-EO provides tools for the development of population-based metaheuristic :*
  - \* Genetic algorithm
  - \* Genetic programming
  - \* Particle Swarm Optimization
  - \* ...
- *ParadisEO-MO provides tools for the development of single solution-based metaheuristics :*
  - \* Hill-Climbing
  - \* Tabu Search
  - \* Simulated annealing
  - \* Incremental evaluation, partial neighbourhood
  - \* ...
- *ParadisEO-MOEO provides tools for the design of Multi-objective metaheuristics :*
  - \* MO fitness assignment schemes (the ones used in NSGA-II, IBEA ...)
  - \* MO diversity assignment schemes (sharing, crowding)

<sup>1</sup><http://paradisEO.gforge.inria.fr/>

<sup>2</sup>[http://www.inria.fr/futurs/recherche/les-equipes-de-recherche/DOLPHIN\\_page](http://www.inria.fr/futurs/recherche/les-equipes-de-recherche/DOLPHIN_page)

<sup>3</sup><http://eodev.sourceforge.net/>

- \* Elitism
- \* Performance metrics (contribution, entropy ...)
- \* Easy-to-use standard evolutionary algorithms (NSGA-II, IBEA ...)
- \* ...
- *ParadisEO-PEO provides tools for the design of parallel and distributed metaheuristics :*
  - \* Parallel evaluation
  - \* Parallel evaluation function
  - \* Island model
  - \* Cellular model

*Furthermore, ParadiseO also introduces tools for the design of distributed, hybrid and cooperative models :*

- *High level hybrid metaheuristics : coevolutionary and relay model*
- *Low level hybrid metaheuristics : coevolutionary and relay model*

### 2 Installation

#### 2.1 ParadiseO

L'archive d'installation de ParadiseO est disponible à l'adresse <http://paradisEO.gforge.inria.fr/> mais compte tenu de l'espace mémoire important nécessaire à son installation, vous n'avez pas à l'installer sur votre compte.

ParadisEO est installé dans le répertoire suivant : `/home/gisEns/fseynhae/paradisEO-1.0.1-beta/`

#### 2.2 Archive pour les TP

Pour découvrir ParadiseO, vous devez installer l'archive `tp_opti_gis_28092007.tgz` disponible dans le répertoire `/home/gisEns/fseynhae/public`.

Une fois cette archive décompressée, placez-vous dans le répertoire `TP_OPTI` créé et éditez le fichier `install.cmake` comme suit :

- Indiquez pour la variable `EO_SRC_DIR` le chemin :  
`/home/gisEns/fseynhae/paradisEO-1.0.1-beta/paradisEO-EO`
- Indiquez pour la variable `EO_BIN_DIR` le chemin :  
`/home/gisEns/fseynhae/paradisEO-1.0.1-beta/paradisEO-EO/build`
- Indiquez pour la variable `MO_SRC_DIR` le chemin :  
`/home/gisEns/fseynhae/paradisEO-1.0.1-beta/paradisEO-MO`
- Indiquez pour la variable `MO_BIN_DIR` le chemin :  
`/home/gisEns/fseynhae/paradisEO-1.0.1-beta/paradisEO-MO/build`

Créez ensuite un répertoire de nom `built`, placez-vous dans ce répertoire et tapez la commande :  
`/home/gisEns/fseynhae/cmake-2.4.7/bin/cmake ../`

Enfin exécutez la commande : `make`

### 3 Contenu de l'archive installée

A la racine du répertoire `TP_OPTI` :

**tsp** : Répertoire contenant les fichiers sources pour la résolution du TSP par ParadiseO (répertoire **src**) et des benchmarks pour le TSP (répertoire **benchs**).

**lesson1** : Répertoire contenant une implémentation d'une recherche locale `hill_climbing.cpp` et un tutoriel associé `lesson1.pdf`.

**lesson2** : Répertoire contenant une implémentation d'une recherche tabou `tabu_search.cpp` et un tutoriel associé `lesson2.pdf`.

**lesson3** : Répertoire contenant une implémentation d'un recuit simulé `simulated_annealing.cpp` et un tutoriel associé `lesson3.pdf`.

**lesson4** : Répertoire contenant une implémentation d'un algorithme génétique `gen_algo.cpp`.

**built** : Répertoire contenant les exécutables des implémentations précédentes (sous-répertoires **lesson1**, **lesson2**, **lesson3**, et **lesson4**).