Evolutionnary algorithm control
in distributed computing environnement

Sébastien Verel

LISIC

Université du Littoral Côte d’Opale
Equipe OSMOSE (ex. CAMOME)

verel@lisic.univ-littoral.fr
http://www.lisic.univ-littoral.fr/~verel

Séminaire du LISIC,

October, 17, 2013,

Calais
**Optimization**

**Inputs**
- **Search space**: Set of all feasible solutions, $\mathcal{X}$
- **Objective function**: Quality criterium $f : \mathcal{X} \rightarrow \mathbb{R}$

**Goal**
Find the best solution according to the criterium

$$x^* = \text{argmax } f$$

*But, sometime, the set of all best solutions, good approximation of the best solution, good 'robust' solution...*
Black box Scenario

We have only \{(x_0, f(x_0)), (x_1, f(x_1)), ...\} given by an "oracle"
No information is either not available or needed on the definition of objective function

- Objective function given by a computation, or a simulation
- Objective function can be irregular, non differentiable, non continuous, etc.
- (Very) large search space for discrete case (combinatorial optimization), i.e. NP-complete problems
Typical applications

- Large combinatorial problems:
  Scheduling problems, logistic problems, DOE, problems from combinatorics (Schur numbers), etc.

- Calibration of models:
  Physic world $\Rightarrow$ Model(params) $\Rightarrow$ Simulator(params)
  $\text{Model}(\text{Params}) = \arg\min_M \text{Error}(\text{Data}, M)$

- Shape optimization:
  Design (shape, parameters of design)
  using a model and a numerical simulator
Search algorithms

**Principle**

**Enumeration of the search space**

- A lot of ways to enumerate the search space
- Using random sampling: Monte Carlo technics
- Local search technics:
Search algorithms

- **Single solution-based**: Hill-climbing technics, Simulated-annealing, tabu search, Iterative Local Search, etc.

- **Population solution-based**: Genetic algorithm, Genetic programming, ant colony algorithm, etc.

**Design components are well-known**

- Probability to decrease,
- Memory of path, of sub-space
- Diversity of population, etc.
Position of the work

- One Evolutionary Algorithm key point: Exploitation / Exploration tradeoff
- One main practical difficulty:
  Choose operators, design components, value of parameters, representation of solutions
- Parameters setting (Lobo et al. 2007):
  - Off-line before the run: parameter tuning,
  - On-line during the run: parameter control.
Position of the work

- One Evolutionary Algorithm key point: Exploitation / Exploration tradeoff
- One main practical difficulty: Choose operators, design components, value of parameters, representation of solutions
- Parameters setting (Lobo et al. 2007):
  - Off-line before the run: parameter tuning,
  - On-line during the run: parameter control.

Main question

How to combine correctly the design components according to the problem?
Fitness landscapes

Definition
Fitness landscape $(\mathcal{X}, \mathcal{N}, f)$
- $\mathcal{X}$ is the search space
- $\mathcal{N} : \mathcal{X} \rightarrow 2^{\mathcal{X}}$ is a neighborhood relation
- $f : \mathcal{X} \rightarrow \mathbb{R}$ is a objective function
Multimodal Fitness landscapes

Local optima $x^*$

no neighbor solution with higher fitness value

$$\forall x \in \mathcal{N}(x^*), f(x) \leq f(x^*)$$
Local Optima Network (LON)

Join work

Gabriela Ochoa (univ. Stirling),
Marco Tomassini (univ. Lausanne),
Fabio Daolio (univ. Lausanne)
Energy surface and inherent networks (Doye, 2002)

- Model of 2D energy surface
- Contour plot, partition of the configuration space into basins of attraction surrounding minima
- Landscape as a network

Inherent network:
- **Nodes**: energy minima
- **Edges**: two nodes are connected if the energy barrier separating them is sufficiently low (transition state)
Basins of attraction
Example of small $N^K$ landscape with $\mathcal{X} = \{0, 1\}^6$ and $K = 2$

- Basins of attraction are interlinked and overlapped!
- Most neighbours of a given solution are outside its basin
Local optima network
[GECCO’08 - IEEE TEC’10]

- **Nodes**: local optima
- **Edges**: transition probabilities
Structure of Local Optima Network
Quadratic Assignment Problem [CEC’10 - Phys A’11]
Network Metrics

- $nv$: #vertices
- $lo$: path to best
- $lv$: avg path
- $fnn$: fitness corr.
- $wii$: self loops
- $cc$: clust. coef.
- $zout$: out degree
- $y2$: disparity
- $knn$: degree corr.
ILS Performance vs LON Metrics
[GECCO’12]

1. Multiple linear regression on all possible predictors:
   \[
   \text{perf} = \beta_0 + \beta_1 k + \beta_2 \log(nv) + \beta_2 lo + \cdots + \beta_{10} knn + \epsilon
   \]

2. Step-wise backward elimination of each predictor in turn.

Table: Summary statistics of the final linear regression model. Multiple R-squared: 0.8494, Adjusted R-squared: 0.8471.

<table>
<thead>
<tr>
<th>Predictor</th>
<th>$\hat{\beta}_i$</th>
<th>Std. Error</th>
<th>$p$-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>10.3838</td>
<td>0.58512</td>
<td>$9.24 \cdot 10^{-47}$</td>
</tr>
<tr>
<td>lo</td>
<td>0.0439</td>
<td>0.00434</td>
<td>$1.67 \cdot 10^{-20}$</td>
</tr>
<tr>
<td>zout</td>
<td>$-0.0306$</td>
<td>0.00831</td>
<td>$2.81 \cdot 10^{-04}$</td>
</tr>
<tr>
<td>y2</td>
<td>$-7.2831$</td>
<td>1.63038</td>
<td>$1.18 \cdot 10^{-05}$</td>
</tr>
<tr>
<td>knn</td>
<td>$-0.7457$</td>
<td>0.40501</td>
<td>$6.67 \cdot 10^{-02}$</td>
</tr>
</tbody>
</table>
Tuning and Local Optima Network

- Relevant features using "complex system" technics
- Search difficulty is related to the features of LON:
  - Classification of instances
  - Understanding the search difficulty
  - Guideline of design
- Time complexity can be predicted:
  - Tuning with Portefolio technics
Multiobjective optimization

**Multiobjective optimization problem**

- $\mathcal{X}$: set of feasible solutions in the decision space
- $M \geq 2$ objective functions $f = (f_1, f_2, \ldots, f_M)$ (to maximize)
- $\mathcal{Z} = f(\mathcal{X}) \subseteq \mathbb{R}^M$: set of feasible outcome vectors in the objective space
Definition

**Pareto dominance relation**

A solution \( x \in \mathcal{X} \) dominates a solution \( x' \in \mathcal{X} \) \((x' \prec x)\) iff

- \( \forall i \in \{1, 2, \ldots, M\}, \ f_i(x') \leq f_i(x) \)
- \( \exists j \in \{1, 2, \ldots, M\} \) such that \( f_j(x') < f_j(x) \)
Goal

Find the **Pareto Optimal Set**, or a **good approximation** of the Pareto Optimal Set
Motivations

Join work

Arnaud Liefooghe, Clarisse Dhaenens, Laetitia Jourdan, Jérémie Humeau,
DOLPHIN Team, INRIA Lille - Nord Europe

Goal

- Understand the dynamics of search algorithms and evaluate the time complexity
- Guideline to design efficient local search algorithms
- Portefolio: between of set of algorithms, select the best one
Local search algorithms

Two main classes:

- **Scalar approaches**: multiple scalarized aggregations of the objective functions, only able to find a subset of Pareto solutions (supported solutions)
- **Pareto-based approaches**: directly or indirectly focus the search on the Pareto dominance relation
Scalar approaches

- multiple scalarized aggregations of the objective functions

**Different aggregations**

- **Weighted sum:**
  
  \[ f(x) = \sum_{i=1}^{m} \lambda_i f_i(x) \]

- **Weighted Tchebycheff:**
  
  \[ f(x) = \max_{i=1..m} \{ \lambda_i (r_i^* - f_i(x)) \} \]

- ...
A Pareto-based approach: Pareto Local Search

- Archive solutions using **Dominance relation**
- Iteratively improve this archive by exploring the neighborhood
MNK-landscapes with tunable objective correlation

Conflicting objectives
\[ \rho = -0.9 \]
- Large Pareto set
- Few Pareto sol.

Independent objectives
\[ \rho = 0.0 \]

Correlated objectives
\[ \rho = 0.9 \]
- Small Pareto set
- All supported sol.
Summary of results and guidelines for local search design

<table>
<thead>
<tr>
<th>Landscape features</th>
<th>Problem properties</th>
<th>Suggestion for the design of local search</th>
</tr>
</thead>
</table>
| Cardinality of the Pareto optimal set     | $N$ $K$ $M$ $\rho$ | + limited-size archive
|                                           |                    | − unbounded archive                      |
| Proportion of supported solutions         |                    | + scalar efficient
|                                           |                    | − scalar not efficient                   |
| Connectedness of the Pareto optimal set   | $\rho \geq 0$      | + two-phase efficient                    |
|                                           |                    | − two-phase not efficient                |
|                                           | $\rho < 0$         |                                          |
| Number of Pareto local optima             |                    | + Pareto not efficient                   |
|                                           |                    | − Pareto efficient                       |

Following project...
One EA key point:
   Exploitation / Exploration tradeoff

One main practical difficulty:
   Choose operators, design components, value of parameters, representation of solutions

Parameters setting (Lobo et al. 2007):
   - Off-line before the run: *parameter tuning*,
   - On-line during the run: *parameter control*.

Increasing number of computation resources (GPU, ...)
   but, add parameters

Main question

**Control** the execution of different metaheuristics
   in a **distributed environment**
Distributed Adaptive Metaheuristic Selection (DAMS) [GECCO, 2011]

Join work

Bilel Derbel,
Université Lille 1, INRIA Lille Nord Europe
Execution of metaheuristics in a distributed environment

- 3 metaheuristics \( \{M_1, M_2, M_3\} \)
- 4 nodes of computation

Which algorithm can we design?
Execution of metaheuristics in a distributed environment

- 3 metaheuristics \( \{M_1, M_2, M_3\} \)
- 4 nodes of computation

Which algorithm can we design?
Execution of metaheuristics in a distributed environment

- 3 metaheuristics \( \{M_1, M_2, M_3\} \)
- 4 nodes of computation

Which algorithm can we design?
Execution of metaheuristics in a distributed environment

- 3 metaheuristics \{M_1, M_2, M_3\}
- 4 nodes of computation
Execution of metaheuristics in a distributed environment

- 3 metaheuristics \{M_1, M_2, M_3\}
- 4 nodes of computation

Goal of control: Optimal strategy

The strategy that gives the best overall performances:

- At first, execution of \(M_3\)
- then, \(M_1\) and \(M_2\), and then, \(M_1\).
Control: Local strategy

$m^n$ attachements with $n$ nodes and $m$ metaheuristics

Local distributed strategy:

For each computational node:
- Measure the efficiency of neighboring metaheuristics
- Select a metaheuristic according to the local information

Trade-off between:

*Exploitation* of the most promising metaheuristics

*Exploration* of new ones
Select Best and Mutate strategy (SBM)

For each nodes:

\[ k \leftarrow \text{initMeta()} \]
\[ P \leftarrow \text{initPop()} \]
\[ \text{reward} \leftarrow 0 \]

while Not stop do
    Migration of populations
    Send \((k, \text{reward})\)
    Receive \(\forall i \ (k_i, \text{reward}_i)\)

    \[ k \leftarrow \text{best meta from } k_i \text{'s} \]
    if \(\text{rnd}(0, 1) < p_{mut}\) then
        \[ k \leftarrow \text{random meta} \]
    end if

    \[ P_{\text{new}} \leftarrow \text{meta}_k(P) \]
    \[ \text{reward} \leftarrow \text{Reward}(P, P_{\text{new}}) \]
    \[ P \leftarrow P_{\text{new}} \]
end while
Select Best and Mutate strategy (SBM)

For each nodes:

\[ k \leftarrow \text{initMeta}() \]
\[ P \leftarrow \text{initPop}() \]
\[ \text{reward} \leftarrow 0 \]

\textbf{while} Not stop \textbf{do}

Migration of populations
Send \((k, \text{reward})\)
Receive \(\forall i (k_i, \text{reward}_i)\)

\[ k \leftarrow \text{best meta from } k_i\text{'s} \]
\textbf{if} \( \text{rnd}(0,1) < p_{mut} \) \textbf{then}
\[ k \leftarrow \text{random meta} \]
\textbf{end if}

\[ P_{\text{new}} \leftarrow \text{meta}_k(P) \]
\[ \text{reward} \leftarrow \text{Reward}(P, P_{\text{new}}) \]
\[ P \leftarrow P_{\text{new}} \]

\textbf{end while}
Dynamics of SBM-DAMS

Average frequency and fitness

- First, \(1/l\) bit-flip mutation (opposite of sequential oracle)
- Then 5-bits, 3-bits, 1-bit, \(1/l\) bit-flip

See sbmDams.mov
Conclusion
Conception des algorithmes d’optimisation : réglage et contrôle des paramètres

- One EA key point: Exploitation / Exploration tradeoff
- One main practice difficulty:
  Choose operators, design components, value of parameters

<table>
<thead>
<tr>
<th>Fitness Landscapes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features, description of the search space</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Off-line tuning:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model to predict the performances,</td>
</tr>
<tr>
<td>Design guideline, portefolio</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>On-line control:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine learning technics</td>
</tr>
</tbody>
</table>

⇒ Toward (more) automatic design