Computational Intelligence Optimization

Ferrante Neri

Department of Mathematical Information Technology, University of Jyväskylä

12.09.2011
1. What is Optimization?

2. What is a fitness landscape?

3. Features of landscape vs algorithmic solutions?

4. Metaheuristics

5. Memetic Computing
Who am I and where do I come from?

- I am Italian born researcher adopted by Finland
- Adjunct Professor at the University of Jyväskylä
- Leader of the Computational Intelligence Optimization group
Where is located Jyväskylä?

- In the middle of Finland (Keski Suomi)
Jyväskylä, Summer and Winter

- Summer

- Winter
My Home
How to get back home from work?
What is Optimization?

- To go back home in the shortest possible time is an **Optimization Problem**
- If I have to visit a shop the problem is **Constrained**
- If I want to take some physical exercise the problem is **Multi-objective**
- If the amount of people in the street can affect my walking speed, the problem is **Time-dependant** (affected by uncertainties)
Maximize

Minimize

subject to

\[ \begin{align*}
  f_m & \quad m = 1, 2, \ldots, M \\
  g_j (x) & \leq 0 \quad j = 1, 2, \ldots, J \\
  h_k (x) & = 0 \quad k = 1, 2, \ldots, K \\
  x_i^L & \leq x_i \leq x_i^U \quad i = 1, 2, \ldots, n
\end{align*} \]  

(1)

where \( g_j \) and \( h_k \) are inequality and equality constraints, respectively.
Under the statement that the linear distance is what actually matters in terms of time-loss, the minimization of the time is simply the search of the shortest path.

This problem is trivial as the objective function (fitness) $f$ would be well-known (and linear).

This would allow the application of an exact method, e.g. gradient based.

In real-world problems, the analytical expression of the objective function is usually not available.

As a black box we must find the optimum anyway.
Metaheuristics and CIO

- **Blunt Definition (1):** Metaheuristics are those algorithms which do not require hypotheses on the objective function.

- **Blunt Definition (2):** Computational Intelligence Optimization is a subject which integrates artificial intelligence into algorithms for solving optimization problems.
OK....but....What is the best optimizer?

- In a nutshell: **There is no best optimizer!**

- No Free Lunch Theorem(s) (1997): for a given pair of algorithms $A$ and $B$:

$$\sum_f P(x_m|f, A) = \sum_f P(x_m|f, B) \quad (2)$$

where $P(x_m|f, A)$ is the probability that algorithm $A$ detects the optimal solution for a generic objective function $f$ and $P(x_m|f, B)$ is the analogue probability for algorithm $B$.

- Ad-hoc algorithmic design is fundamental!
Wise Ignorance

- Black Box= We do not know anything about the objective function: TRUE
- Black Box= We do not know anything about the optimization problem: FALSE
- Even though the objective function can be unknown, we can still analyse the fitness landscape prior to design an algorithm
- An efficient design takes into account the features of the fitness landscape/optimization problem
Problem Analysis (Some ideas)

What is the **dimensionality** of the problem? Note: The complexity of a problem does NOT grow linearly with its dimensionality.

What’s the **multimodality** degree?

Does the landscape contain **plateaus**?

How much is **ill-conditioned**? (Importance of the variables)

Is the function **separable**? partially separable? (connection amongst the variables)
Computational Intelligence Optimization: A brute taxonomy

Single-solution Algorithms

Population-based Algorithms
  ■ Evolutionary Algorithms. Evolutionary metaphor. GAs, ES, GP, etc.
  ■ Swarm Intelligence. Groups of animals, fish, birds, bacteria, bees, monkeys, etc.
An example of single-solution algorithm: Simulated Annealing

one solution is progressively perturbed so we have a current best $x_{cb}$ and a trial $x_{tr}$

$x_{tr}$ replaces $x_{cb}$ if it is “better” ....

....OR if is worse with an exponentially decreasing probability over time: I am ready to accept a certain worsening at the beginning of the optimization process but I would rather keep my solution at the end of the optimization
Evolutionary Algorithm: A general framework

INITIALIZE population with random individuals;
EVALUATE each individual;
While TERMINATION CONDITION is not satisfied
SELECT parents;
RECOMBINE pairs of parents;
MUTATE the resulting offspring;
EVALUATE new individuals;
SELECT individuals for the next generation;
Swarm Intelligence: A general framework

INITIALIZE population with random individuals;
EVALUATE each individual;
While TERMINATION CONDITION is not satisfied
For EACH parent;
PERTURB an individual;
EVALUATE the individual and compare it with that prior the perturbation;
SELECT the winning individual;
DIFFERENTIAL EVOLUTION: SOMETHING IN THE BETWEEN

INITIALIZE population with random individuals;
EVALUATE each individual;
While TERMINATION CONDITION is not satisfied
For EACH parent;
PERTURB an individual;
RECOMBINE pairs of parents; EVALUATE the individual and compare it with that prior the perturbation;
SELECT the winning individual;
A unifying concept

There is a plenty of algorithms inspired by the most diverse phenomena....

At the end of the day, all the algorithms have the same structure, i.e. they are a combination of two classes of operations:

- GENERATION of a trial
- SELECTION of the new current best

Further metaphor: if designing algorithms is like cooking, we need to select proper operators and be able to combine them efficiently

To know a plenty of algorithms is like to know by hearth a recipe book.....it does not mean to be able to cook
Memetic Computing

MC is a subject which studies complex and dynamic computing structures composed of interacting modules (memes) whose evolution dynamics is inspired by the diffusion of ideas.

- All the algorithms can be seen as a set of operators which interact while solving an optimization problem.
- A proper selection of the combination of these operators is an alternative perspective to state an optimization problem.
- This structure suggests the idea that algorithms can be designed automatically by machines.
- This will be the future step in Computational Intelligence Optimization.

NOTE: Software platform for Memetic Computing design.
Memetic Algorithms (MAs) are computational intelligence structures combining multiple and various operators in order to address optimization problems. The combination and interaction amongst operators evolves and promotes the diffusion of the most successful units and generates an algorithmic behavior which can handle complex objective functions and hard fitness landscapes. "Handbook of Memetic Algorithms" organizes, in a structured way, all the the most important results in the field of MAs since their earliest definition until now. A broad review including various algorithmic solutions as well as successful applications is included in this book. Each class of optimization problems, such as constrained optimization, multi-objective optimization, continuous vs combinatorial problems, uncertainties, are analysed separately and, for each problem, memetic recipes for tackling the difficulties are given with some successful examples. Although this book contains chapters written by multiple authors, a great attention has been given by the editors to make it a compact and smooth work which covers all the main areas of computational intelligence optimization. It is not only a necessary read for researchers working in the research area, but also a useful handbook for practitioners and engineers who need to address real-world optimization problems. In addition, the book structure makes it an interesting work also for graduate students and researchers in related fields of mathematics and computer science.
Thanks for your attention.
Questions?