# Complexity and Simplicity of Optimization Problems

Yurii Nesterov, CORE/INMA (UCL)

September 4, 2012 (Moscow)

# Developments in Computer Sciences

## Age of Revolutions:

- Revolution of Personal Computers: 1980 – 2000.
- Revolution of Internet an Telecommunications: 1990 – 2010.
- Algorithmic Revolution: 2000 - now.

**NB** Advances of the last years are based on algorithmic Know How

## Examples

- Numerical TV, ADSL
- Google (search, maps, video maps) , Netflix (E-Shops), etc.
- GPS navigators (intelligent routes, positioning, data), etc.

**Main design tool:** Optimization Methods.

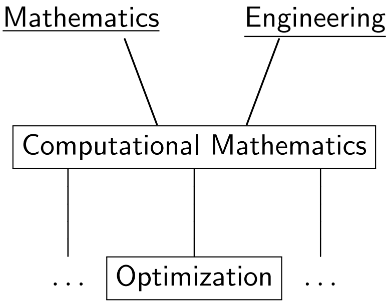
# Introductory Lecture (September 4, 2012)

## Algorithmic Challenges in Optimization: Mathematical Point of View

**Main topics:**

- What can be interesting in Optimization for mathematicians?
- Main directions for the research.
- Advertising of the main course.

# Genealogy


Mathematics
Engineering
Computational Mathematics
. . .
Optimization
. . .

# Mathematics

**Objects:** Abstract notions, axioms and theorems.
**Methods:** Logical proofs.
**Results:** Perfectly correct statements.

(Monopoly for the absolute truth.)

## Definition

MATHEMATICS IS AN ART OF DISCOVERING THE REAL FACTS ABOUT IMAGINARY OBJECTS.

## Behavioral rules

- Any question has a right to be answered.
- The older is the question, the more prestigious is finding the answer (e.g. Great Fermat Theorem; Jackpot principle?).
- Many important problems remain unsolved.

# Engineering

**Objects:** Exist in the real nature.
**Methods:** Experience, modeling, physical sciences.
**Results:** Reliable constructions (under normal conditions).

## Definition

ENGINEERING IS AN ART OF CONSTRUCTING THE REAL OBJECTS BASED ON IMAGINARY FACTS.

## Behavioral rules

- Open questions: importance is measured by practical consequences.
- Old problems quickly loose the relevance (Philosopher's stone, Perpetual motion). Alexandrian solution for Gordian knot?
- All really important problems are solvable. (Life still goes on!)

# Computational Mathematics: A Child of Two Extremes?

**Objects:** Mathematical models.
**Methods:** Iterative procedures implemented on computers.
**Results:** Numbers.

Too much of ambiguity in the input and output?

### Definition (?)

Computational mathematics is an art of producing imaginary facts about imaginary objects.

Other suggestions? Difficult to find ...

# Computational Mathematics: Hope for true respect?

(Do not mix with *mathematical computations*!)

## Observations

- Position of the International Union of Mathematicians.
- Very often, engineers prefer their homebred algorithms.
- Books on Computational Mathematics (fuzzy questions, many assumptions, fuzzy answers). Usually very thick!
- In view of the fast progress in computers, the computational experience becomes obsolete very quickly.
- Accumulation of knowledge?

# Optimization Fields

## Mathematical Optimization

- Optimality conditions and Nonlinear Analysis.
- Optimal Control.
- Semi-infinite optimization.
- Optimization in Banach spaces.
- Quantum Computing (???)

## Engineering Optimization

- Genetic algorithms, ants, etc.
- Surrogate Optimization, Tabu Search.
- Neural Networks, Simulated Annealing, etc.

Time to introduce *Algorithmic Optimization*?

# Comparing theoretical goals ...

## Mathematics

- The more general is the statement, the more powerful it is.
- Problem classes should be as abstract as possible.

## Algorithmic Optimization

- Statements proved for all numerical schemes are usually silly.
- We have already enough troubles with problems formed by the simplest functions.
- The main goal is the selection of the best scheme applicable to a particular problem.
- All possible efforts should be spent for exploiting the structure of a particular problem instance in the numerical scheme.

# Main declarations

**Our claim:**

- In Computational Mathematics, there exist research directions interesting both for mathematicians and engineers.
- For these developments, we need new mathematical tools.
- The new schemes have good chances to become the most efficient in practice.

**Our field:** NONLINEAR OPTIMIZATION

**Our goals:**

- Optimization Methods with full Complexity Analysis.
- No gap between Theory and Practice.

# Underwater rocks

- Data size.
- Dimension.
- Accuracy.
- Discreteness.

**Main goal:** Cut off unsolvable problem keeping a significant number of real-life applications.

# Complexity issues

**Example:**

**Goal:** Solve equation $x^2 + 2ax + b = 0$ with integer $a, b$.
**Answer:** $x = -a \pm \sqrt{a^2 - b}$.

What is the complexity of this problem?

**Naive answer:** 4 a.o. + 1 sqrt. Works well when $a^2 - b = \frac{m^2}{n^2}$.
**If not,** we need to introduce a lot of details:

- Representation of input, output and intermediate results.
- Computational tools.
- Required accuracy, etc.

**Note:** for some variants, the problem is *unsolvable*.

# Algorithmic complexity

**Meta-Theorem.** Assume that in our problem class $\mathcal{P}$:

- Complexity of the problems is an increasing unbounded function of the data size.
- Speed of computers is finite.

Then there exists a problem in $\mathcal{P}$, which cannot be solved during the time of existence of Universe.

**Corollary:** The majority of problem classes, solvable from mathematical point of view, contain numerically unsolvable instances.

How to distinguish solvable and unsolvable problems?

# Scale for complexity measures

**Engineering scale:** TIME OF HUMAN LIFE.

**Observation:** Before solving the problem, we need to *pose* it. (collecting the data, coding it, etc.)

**Fair goal:** SOLVE ANY PROBLEM, WHICH WE CAN POSE.

### Example

Pose the problem $\equiv$ write down its formulation by hand.

**Complexity measure:** Number of digits in the data set.

**Polynomial-time methods:** performance is proportional to the data length.

# Small and big numbers (by Engineering Scale)

## Small numbers

- Number of production items for a time period.
- Total length of highways in Europe (in km).

## Big number

**Orders in a pack of 52 cards:** $52! \approx 8.05 \cdot 10^{67}$ variants.
**Compare:**

- 65 years $= 2 \cdot 10^9$ sec.
- Cumulative Human Population of Earth: $10^{11}$.

**Mathematician:** Practical experience is too limited.
**Engineer:** Practical experience is extraordinary selective.

# NP-hard problems: price for universality?

**Example: find Boolean solution $x_i = \pm 1$ to the following equation:**

$$(*) \qquad \sum_{i=1}^{n} a_i x_i = 0,$$

where all $a_i > 0$ are integer. Full search: $2^n$ variants (exponential in the dimension $n$). For $n = 100$, we have $2^n \approx 10^{30}$.

**Closed form solution:**

$$2^n \cdot \int_0^{2\pi} \left[ f(t) \stackrel{\text{def}}{=} \prod_{i=1}^{n} \cos(a_i t) \right] \cdot dt \;=\; 2\pi \cdot (\#\text{ of solutions to } (*))$$

**Can we compute this integral? Yes!** Since $f(t)$ is a trigonometric polynomial of degree $N = \sum_{i=1}^{n} a_i$, we need $O(nN)$ a.o.

If all $a_i$ have a "real-life origin", then $N$ is reasonably small.

# Artificial coefficients

**Problem:** Find a Boolean solution of the system

$$(**):\quad \sum_{i=1}^{n} a_i^j x_i \;=\; 0, \quad j = 1, \ldots, m,$$

where all $a_i^j$ are integer. Denote $M = \max_{1 \le j \le m} \sum_{i=1}^{n} |a_i^j|$.

Define $b_i = \sum_{j=1}^{m} (M+1)^{j-1} a_i^j$, $i = 1, \ldots, n$.

**Lemma:** Boolean $x$ satisfies (**) if and only if $\sum_{i=1}^{n} b_i x_i = 0$.

**Note:** Physical sense of the residuals is lost. (Same for accuracy.)

Extreme NP-hard problem instance: for given $\alpha, \beta \in Z$ find

$$x, y \in Z: \quad x^2 = \alpha y + \beta.$$

# Reducibility of the problems

**NP-hard problems:** are mutually reducible with *polynomial growth* of coefficients.

**Old Mathematical Principle**

The problem is solved if it can be reduced to another problem with known solution. (Or, with known method for finding its solution.)

**Combinatorial Optimization:** this works (?) since we are looking for *exact* solutions.

**Nonlinear Optimization:**

- We are able to compute only *approximate* solutions.
- Transformation of problems changes the quality of approximations and the residuals. Be careful!

# Continuity and Discreteness

**Main principle:** Avoid discrete variables by all possible means.

**Example**

"To be or not to be?" (*Hamlet*, Shakespeare, 1601)

- Discrete choice is difficult for human beings.
- It is also difficult for numerical methods.

**Any compromise solution must be feasible:** $\{x, y\} \Rightarrow [x, y]$.

Thus, we always work with *convex* objects (sets, functions, etc.).

# Golden Rules

1. Try to find an unsolved and easy problem.

2. Try to keep the physical sense of the components (hoping to avoid big numbers).

3. New optimization scheme must be supported by complexity analysis.

4. The first encouraging numerical experiments must be performed by the author.

# Lecture 1: Intrinsic complexity of Black-Box Optimization

**Negative results.** All problems below are NP-hard

- Finding a descent direction for nonsmooth nonconvex function.
- Optimal control problems with nonlinear ODE.
- Minimization of cubic polynomial on a Euclidean sphere.

**Lower bounds: analytic complexity for finding $\epsilon$-solution**

- Nonconvex optimization: $O(\frac{1}{\epsilon^n})$.
- Nonsmooth convex optimization: $O(\frac{1}{\epsilon^2})$.
- Nonsmooth strongly convex optimization: $O(\frac{1}{\mu\epsilon})$.
- System of linear equations: $O(\frac{1}{\epsilon})$.
- Smooth convex optimization: $O(\frac{1}{\epsilon^{1/2}})$.

**Tools:** Resisting oracle, worst functions in the world.

# Lecture 2: Looking into the Black Box (Structural Optim.)

## Smoothing technique

- Minimax representation of nonsmooth function.
- Smoothing by prox functions.
- Solving the smoothed problems by Fast Gradient Methods.

## Self-concordant barriers

- Polar set. Geometric origin of self-concordant barriers.
- Barrier parameter. How to construct self-concordant barriers.
- Polynomial-time path-following method.

# Lecture 3: Huge-scale optimization problems

**Main features**

- Even the simplest vector operations are difficult.
- Acceptable cost of one iteration: logarithmic in dimension.

**Available technique:**

1. Coordinate descent methods for smooth functions.
2. Subgradient methods for nonsmooth functions with sparse subgradients.

**Applications:**

- Finite-element schemes.
- Problems with PDE-constraints.
- Google problem.

# Lecture 4: Nonlinear analysis of combinatorial problems

## Boolean Quadratic Optimization

- Simple polyhedral and eigenvalue bounds.
- Semidefinite relaxation. $\frac{2}{\pi}$-approximation.

## Counting technique for NP-hard problems

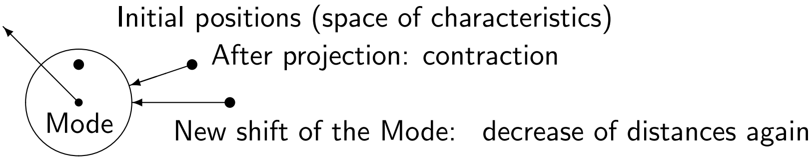
- Generating functions. Polynomials on unit circle.
- Counting problems.
- Fast computations by FFT.

# Lecture 5: Algorithmic models of human behavior

**Main obstacles for the rational choice:**

- For nonsmooth functions, marginal utilities do not work.
- Dimension. Impossibility of massive computations.
- Conscious/Subconscious behavioral patters.

**Example of subconscious adjustment**




**Limiting pattern:** points stick all together.

**Topics:** 1. Random intuitive search as a basis of rational behavior.

2. Algorithms of rational consumption.

Thank you for your attention!