Signal processing and identification–AI module, Lecture 1

Introduction

Ziemowit Dworakowski
AGH University of Science and Technology,
Department of Robotics and Mechatronics

1

Introduction

dr hab. inż. Ziemowit Dworakowski, prof. AGH
Department of Robotics and Mechatronics
D-1, r. 407 (IV floor)

Office hours: Thursday, 10:00 – 11:00 (officially)
(but better to just come and try whenever or write an e-mail a day before asking whether I'll be available)

e-mail: zdw@agh.edu.pl
www (course data): galaxy.agh.edu.pl/~zdw

Scope of expertise:
Artificial intelligence and soft computing applied in:
- Structural health monitoring, vision systems, vibration suppression, vibrodiagnostics, decision systems etc.
- ***Artificial neural networks (and other classification methods)***
- *Decision fusion*
- *Evolutionary computation*
- *Fuzzy systems*
- *Control*

2

Introduction

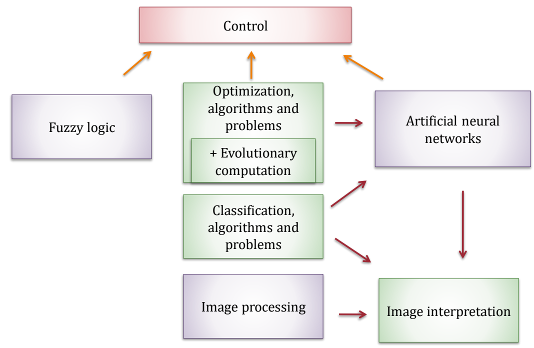
Supplementary material:

1) **Lecture notes (main source of knowledge)**

2) J. Arabas, Wykłady z algorytmów ewolucyjnych, *WNT Warszawa (2004)*
3) R.L. Haupt, S.E. Haupt, Practical Genetic Algorithms, *Wiley Interscience*
4) K. Worden, W. J. Staszewski, and J. J. Hensman, "Natural computing for mechanical systems research: A tutorial overview," *Mech. Syst. Signal Process.*, vol. 25, no. 1, pp. 4–111, Jan. 2011.
5) Bishop, C.M. „*Pattern Recognition and Machine Learning*" Springer, 2009
6) S. Haykin, *Neural Networks A Comprehensive Foundation*. Pearson Prentice Hall, 2001.
7) Winston, P.H. *Lectures on:* Artificial Intelligence, *Open Course Ware, MIT*
https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/lecture-videos/

3

Introduction – aim and scope of AI module



4

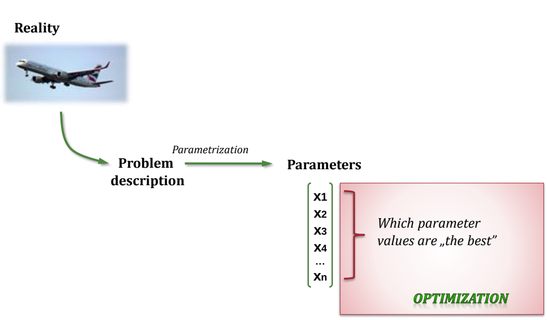
Signal processing and identification–AI module, Lecture 1

Optimization

dr inż. Ziemowit Dworakowski
AGH University of Science and Technology,
Department of Robotics and Mechatronics

5

Optimization

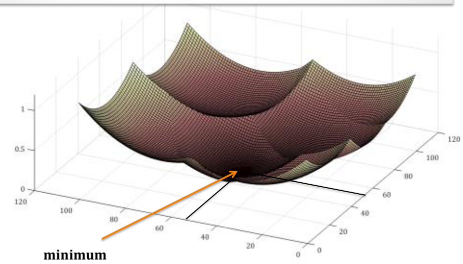


6

Optimization

Optimization task:
- Find a global **minimum** (or maximum) of the **objective function**

(Function defined in a **parameter space**)



7

Optymalizacja

Optimization task:
- Find a global **minimum** (or maximum) of the **objective function**

(Function defined in a **parameter space**)

One criterion *or* Multi-criterion

Continuous function *or* Non-continuous function

With constraints *or* Without constraints

8

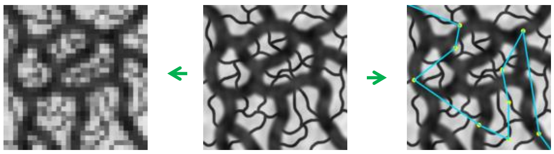
Optimization: examples

- **Path planning**
- **Scheduling**
(e.g. aircraft management)
- **Production or resources optimization**
- **Modeling**
(Finding best set of model parameters etc.)
- **And many more …**

9

Step 1 - parametrization

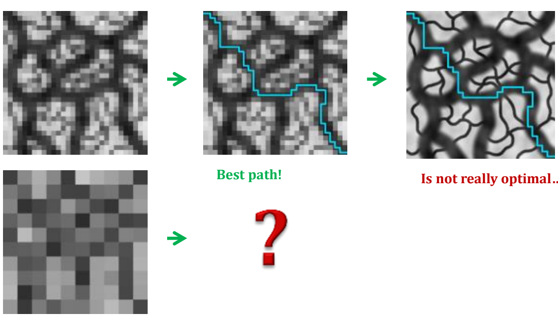
Step 1: Choose a representation
Step 2: Find the best set of parameters



10

Step 1 - parametrization

Step 1: Choose a representation
Step 2: Find the best set of parameters



Best path!

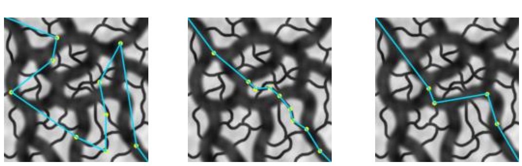
Is not really optimal...

?

11

Step 1 - parametrization

Step 1: Choose a representation
Step 2: Find the best set of parameters



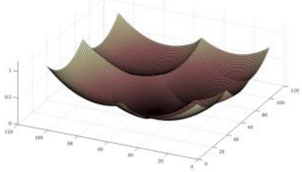
Best solution!
Computationally demanding...

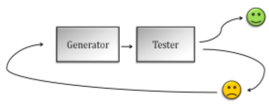
Best solution impossible to find

12

Step 2: Looking for a minimum

Step 1: Choose a representation
Step 2: Find the best set of parameters





In order to be clear here, we'll use a continuous function of two parameters (in practical applications we rarely have that simple representation)

Assumptions:

- *We have two parameters that can be changed in particular range*
- *We don't know function equation, but we can easily compute its value for any given set of parameters*

13

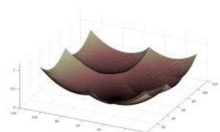
Algorithms

Algorithm 0: **Brute force**

Check all the possibilities…

Algorithm 1: **Grid search**

Pick a finite number of possible values for each parameter and check function value in this grid



14

Algorithms

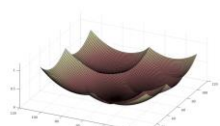
Algorithm 2: Random method

Pick a point in parameter space at random

↓

Check function value in that point

Repeat until result that is good enough or until no time is left



15

Algorithms

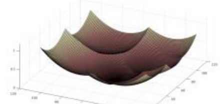
Algorithm 3: random walk („A 1+1 method")

Pick a point in parameter space at random

Check function value in that point. If its the best so far, save it

Make a small step in random direction from a saved point, pick a new point there

Repeat until result is good enough
or until no time is left



16

Algorithms

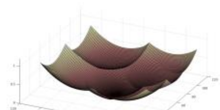
Algorithm 4: Steepest gradient descent

Pick a point in parameter space at random

Calculate a gradient of the objective function in a salected point

Make a step towards steepest gradient descent, pick a new point there

Repeat until result is no longer improved
or until no time is left



17

Algorithms

Algorithm 5: Multistart gradient descent

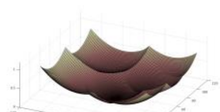
Pick a point in parameter space at random

Calculate a gradient of the objective function in a salected point

Make a step towards steepest gradient descent, pick a new point there

Repeat ~~until result is no longer improved~~
~~or until no time is left~~
For a number of iterations

Repeat until no time is left



18

Algorithms

Algorithm 6: Steepest gradient descent with momentum

Pick a point in parameter space at random

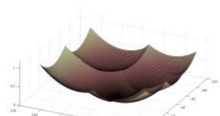
↓

Calculate a gradient of the objective function in a salected point

↓

Make a step towards **weighted sum** of the steepest gradient descent **and previous direction of movement**

Repeat until result is no longer improved or until no time is left



19

Algorithms

Algorithm 7: Newton's method

Pick a point in parameter space at random

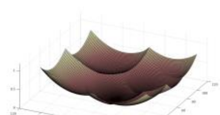
↓

Calculate a **Taylor series** of objective function in a salected point

↓

Make a step **to the point that minimizes square approximation of the objective function**

Repeat until result is no longer improved or until no time is left



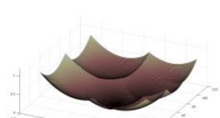
20

Local and global minima

Objective function has at least one **global minimum**

Objective function can have also few or many **local minima** – the points in which the function value is minimal only for a particular neighborhood

Most of the optimization algorithms is **attracted by the local minima** meaning, that if the algorithm encounters starts to descent towards local minimum, getting out of that area can be hard (the algorihtm might not be able to propose a new search point that is outside of the local minimum)



21

Optimization algorithms - comparison

Algorithm	Implementation, configuration	Sensitivity to local minima	Sensitivity to dimensionality	Non-continuous OF	„flat" OF	Convergence speed	Overall efficiency
Grid search	😐	🙂	🙁	🙂	🙂	🙁	🙂
Random	🙂	🙂	🙁	🙂	🙂	🙁	🙁
1+1	🙂	😐	🙂	🙂	🙂	🙂	🙂
Gradient descent	🙂	🙁	🙂	🙁	🙁	😐	😐
Multistart gradient descent	😐	😐	🙂	🙁	🙁	😐	😐
Gradient with momentum	😐	😐	🙂	🙁	😐	😐	🙂
Newton	😐	🙁	😐	🙁	😐	🙂	🙂

22

Signal processing and identification–AI module, Lecture 1

Optimization

1) What is the optimization task?
2) How representation can affect the optimization task?
3) What is an objective function?
4) How brute force algorithm works?
5) How optimization algorithms (all described!) work?
6) What is a local minimum of the objective function?

23