بسم الله الرحمن الرحیم

مقدمات درس روشهای شبیه سازی در فیزیک (نظریه و محاسبات)

Preliminaries for Advanced topics in computational Physics and Optimization

سیدمحمدصادق موحد

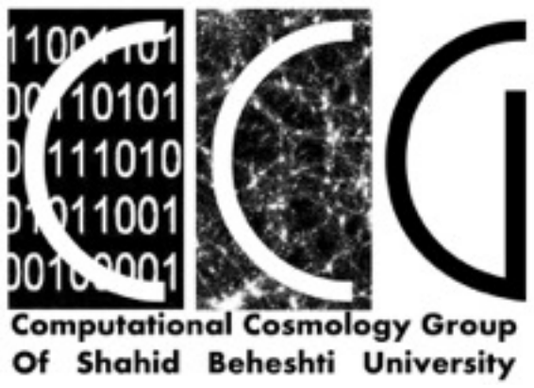
دانشکده فیزیک دانشگاه شهید بهشتی

گروه کیهانشناسی محاسباتی و آزمایشگاه ابن سینا

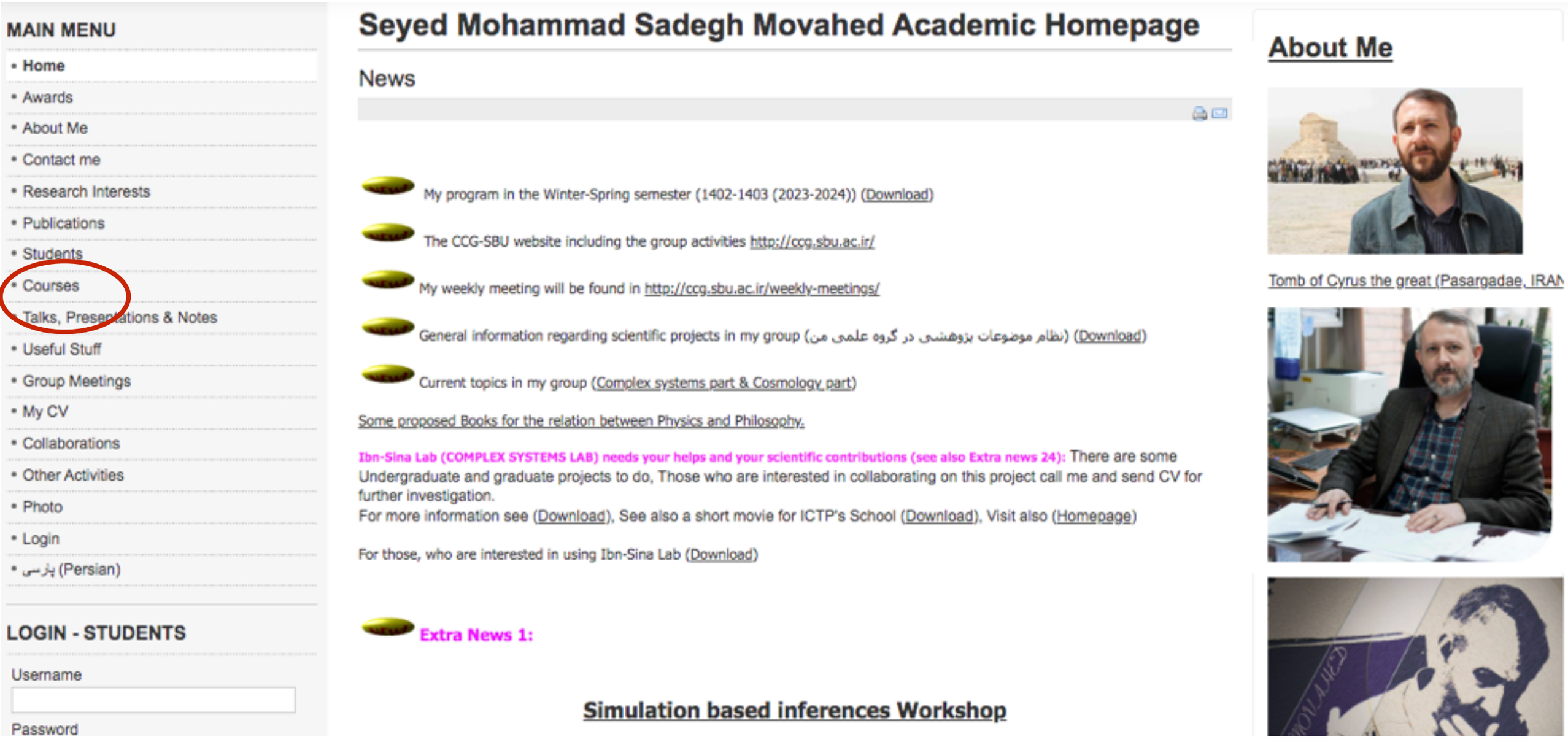
نیم سال دوم، سال تحصیلی ۱۴۰۳-۱۴۰۲

ccg.sbu.ac.ir

smovahed.ir



وبسایت درس



http://facultymembers.sbu.ac.ir/movahed/

وبسایت درس

SHAHID BEHESHTI UNIVERSITY
Department of Physics

Department of Physics
Shahid Beheshti University

MAIN MENU

- Home
- Awards
- About Me
- Contact me
- Research Interests
- Publications
- Students
- **Courses**
- Talks, Presentations & Notes
- Useful Stuff
- Group Meetings
- My CV
- Collaborations
- Other Activities
- Photo
- Login
- پارسی (Persian)

LOGIN - STUDENTS

Username

Password

Remember Me

LOGIN

Advanced course on Computational Physics and Optimization (Winter-Spring 2024)

Thursday, 08 February 2024 00:00

Advanced course on Computational Physics and Optimization for Ph.D. and MS students (Winter-Spring 2024)

This course is devoted to advanced and more recent topics in computational methods for physics and including some topics for Optimization.

Link for class (TBA)

Link for my previous lectures on Computational Physics (SBU-VPN needed)

Link for my previous lectures on Computational Physics

Link for my lectures on Optimization (Khajeh Nasir Digital Library, SBU VPN needed)

Some topics to teach are as follows:

- **Solving coupled Differential Equations and Boundary Value Problems**
- **Chaotic phenomena**
- **Probability Distribution functions and transformations**
- **Correlation functions, Two-point correlation function**
- **Spectral analysis**
- **Monte Carlo simulation**
- **Basic topics for Molecular dynamics simulations**
- **Simulation by VPython**
- **Machine learning in Physics**
- **Topological Based Data Analysis**

- Course subjects and program (Download)
- A good movie presented by Pooyan Goodarzi to connect the server, remotely (Link)

About Me

Tomb of Cyrus the great (Pasargadae, IRAN)

smovahed.ir

The timetable of Course
طرح درس و برنامه زمانبندی

smovahed.ir

Some relevant references in my webpage
برخی از منابع مندرج در وبسایت درس

smovahed.ir

Simulation and Data Sciences
شبیه سازی و علم داده

smovahed.ir

Optimization: General view
بهینه سازی: نگاه کلی

Generic examples

1) Common notion in everyday life
2) Shortest path
3) Euler-Lagrange differential equation
4) Variational approach to compute the upper limit of ground state of a typical system
5) Many physical systems are governed by minimization principle (Gravity, Thermodynamics, …)

Transformation into the optimization problems

1) Determination of the self affine properties of polymers in random media
2) Study of interfaces and elastic manifolds in disordered environments
3) Investigation of the low-temperature behavior of disordered magnets
4) Investigation of morphology of fox line in superconductors
5) Solution of Protein Folding
6) Calculation of ground state of electronic systems
7) Optimization of laser fibers
8) .
9) .
10) .

Canonical definition of Linear optimization

$X = (x_1, x_2, ..., x_N)$ a row vector

$X \in R$

$\mathcal{H} \subset R$ (cost function)

Find $X \in R$ which minimizes or maximizes $\mathcal{H}$

Canonical definition of Linear optimization

$$
\begin{aligned}
X \quad &= \quad (x_1, x_2, ..., x_N) \qquad \text{a row vector} \\
&\quad C^T X \qquad \text{To be minimized (cost function)} \\
&\quad AX < B \\
&\quad X \geq 0 \qquad \text{Constraints}
\end{aligned}
$$

مفهوم و جایگاه روشهای بهینه سازی

Some keywords:

- Feasible region: A set of value of X which fulfills or satisfies all conditions;
- Robustness: Resilience against perturbation;
- Complexity: Time and algorithms

Optimization (General)

Posterior Analysis and Data modeling

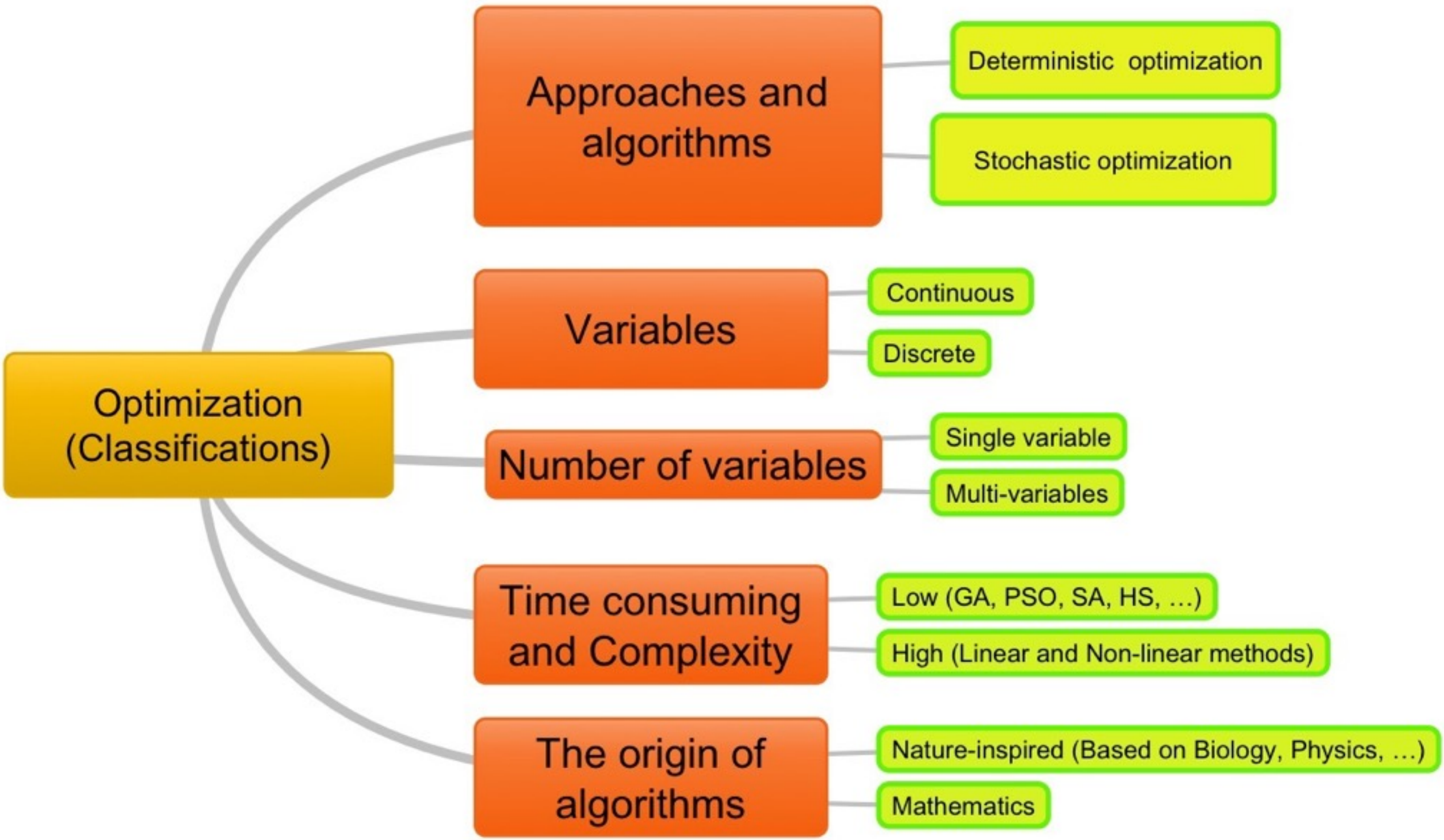
- Fisher forecast
- Paramegter estimation

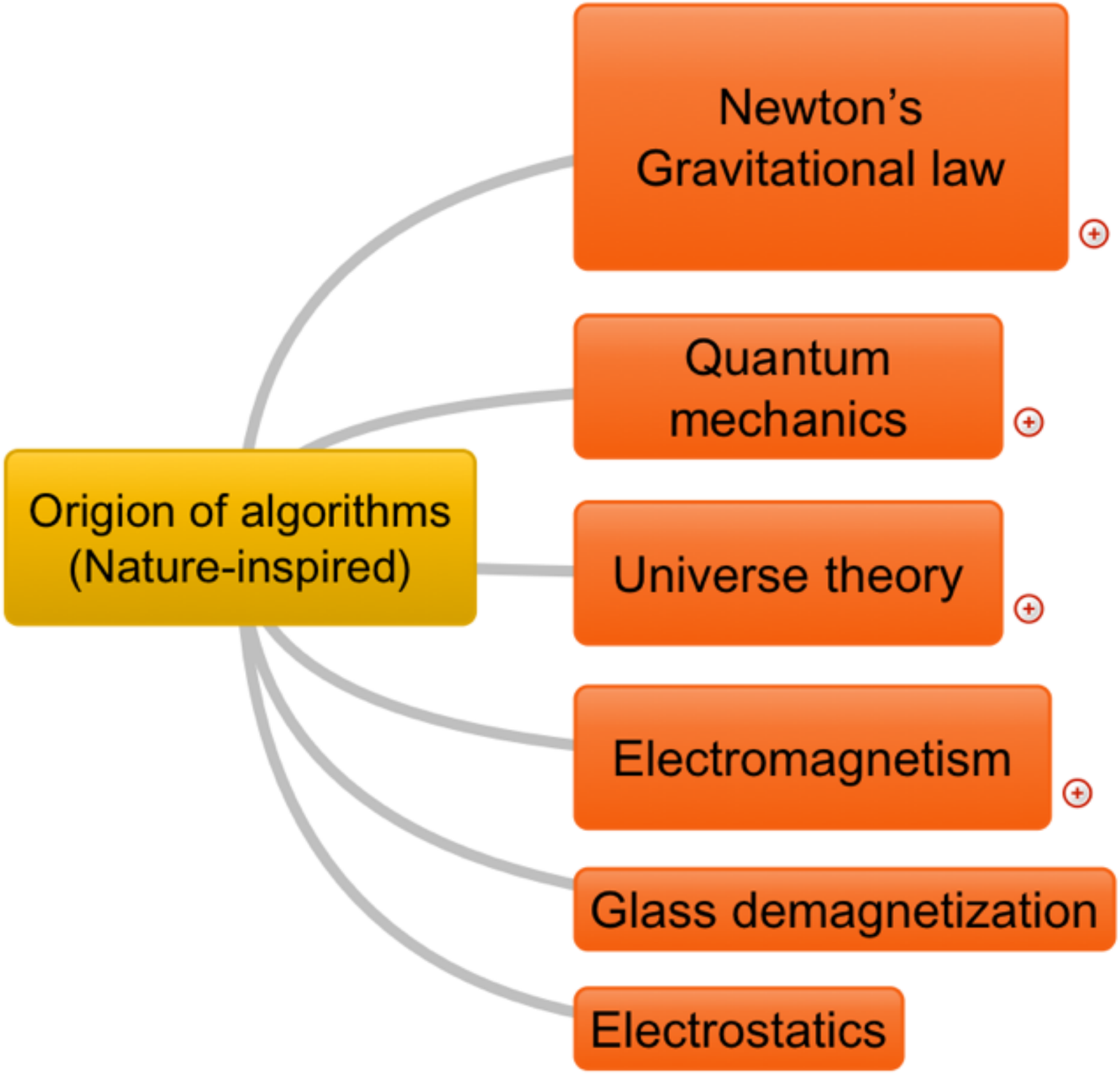
Minimization Principles

- Classical mechanics (Ferma, Euler-Lagrange, ..)
- Optics
- Thermodynamics
- Critical phenomena and Phase transitions
- Quantum mechanics

Algorithms

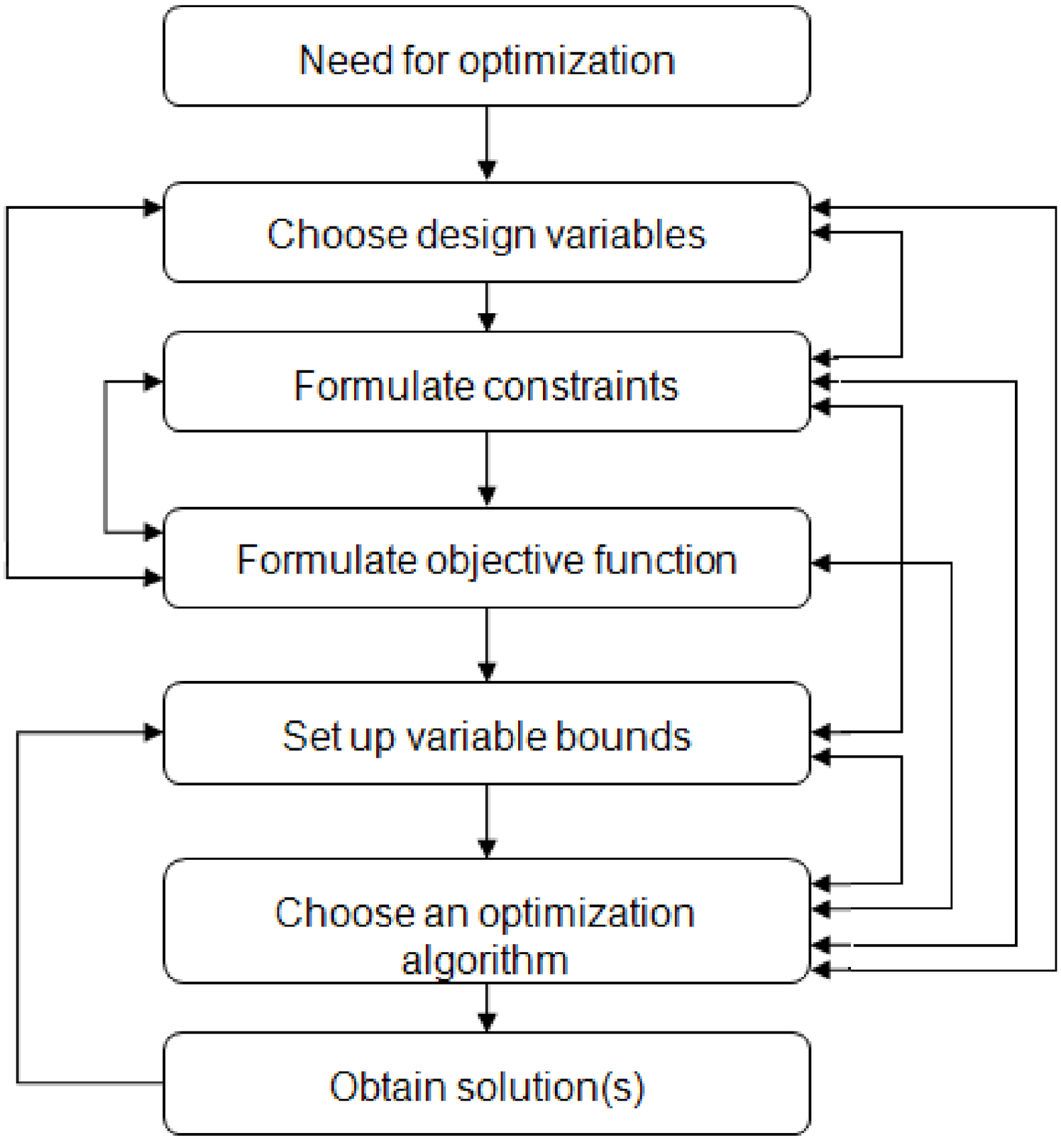
- FFT
- Genetic algorithm
- Machine learning
- Linear/Non-liniear Programing (Linear/Non-linear optimization): a method to achieve best outcome
- Stochastic programing: Looking for the solution by using random generator
- Particle swarm optimization (PSO): 1995 based on fishes and birds swarms
- Simulated annealing (SA): 1983 based on annealing of metal
- Harmony search (HS): 2001 based on adjusting the instrument to obtain good harmony


Optimization (Classifications)
Approaches and algorithms
Deterministic optimization
Stochastic optimization
Variables
Continuous
Discrete
Number of variables
Single variable
Multi-variables
Time consuming and Complexity
Low (GA, PSO, SA, HS, …)
High (Linear and Non-linear methods)
The origin of algorithms
Nature-inspired (Based on Biology, Physics, …)
Mathematics



Biswas, Anupam, et al. "Physics-inspired optimization algorithms: a survey." Journal of Optimization 2013 (2013).

Optimization Flowchart



https://mech.iitm.ac.in/meiitm/

Optimization Flowchart

A) Design variables
- Model building
- Observable quantities
- Prior informations

Optimization Flowchart

B) Constraints

- Geometry and topology
- Boundary conditions (periodic boundary, ….)

Optimization Flowchart

C) Objective Function (cost function)

- Posterior and Likelihood
- Hamiltonian
- Entropy
- Thermodynamic Potential
- Nature-inspired functions

Optimization Flowchart

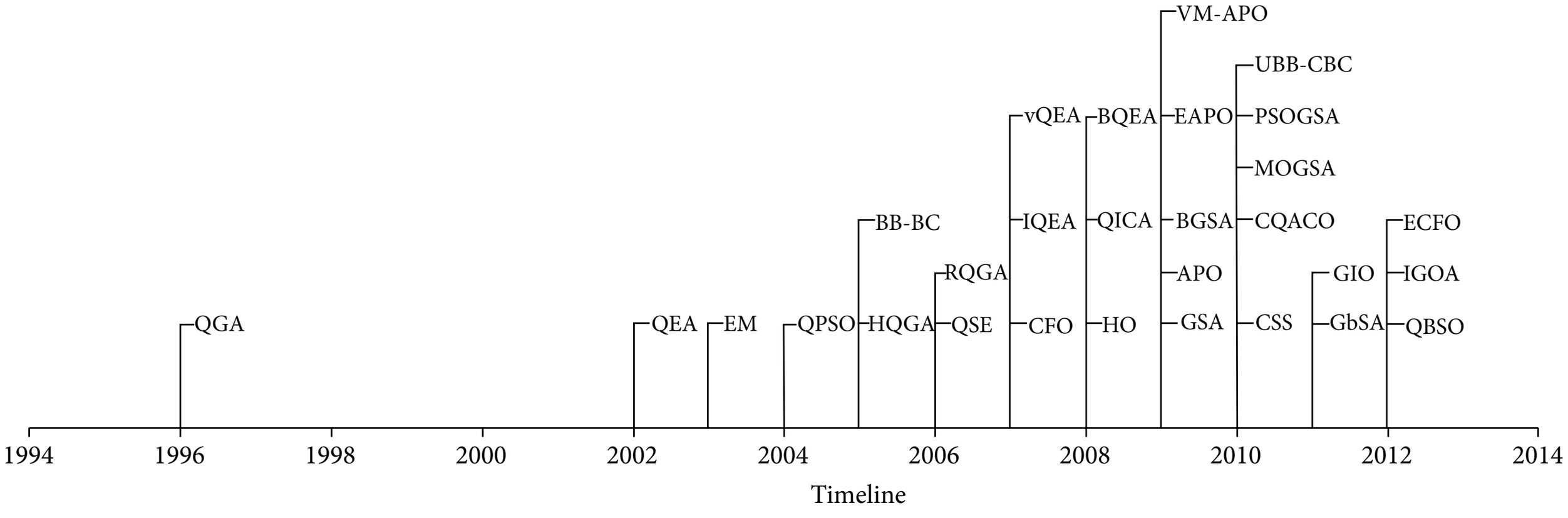
D) Variable bounds

- Variable domains coming from theories or experiments

Optimization Flowchart

E) Optimization Algorithms

Physics-inspired algorithms



Biswas, Anupam, et al. "Physics-inspired optimization algorithms: a survey." Journal of Optimization 2013 (2013).

Physics-inspired algorithms

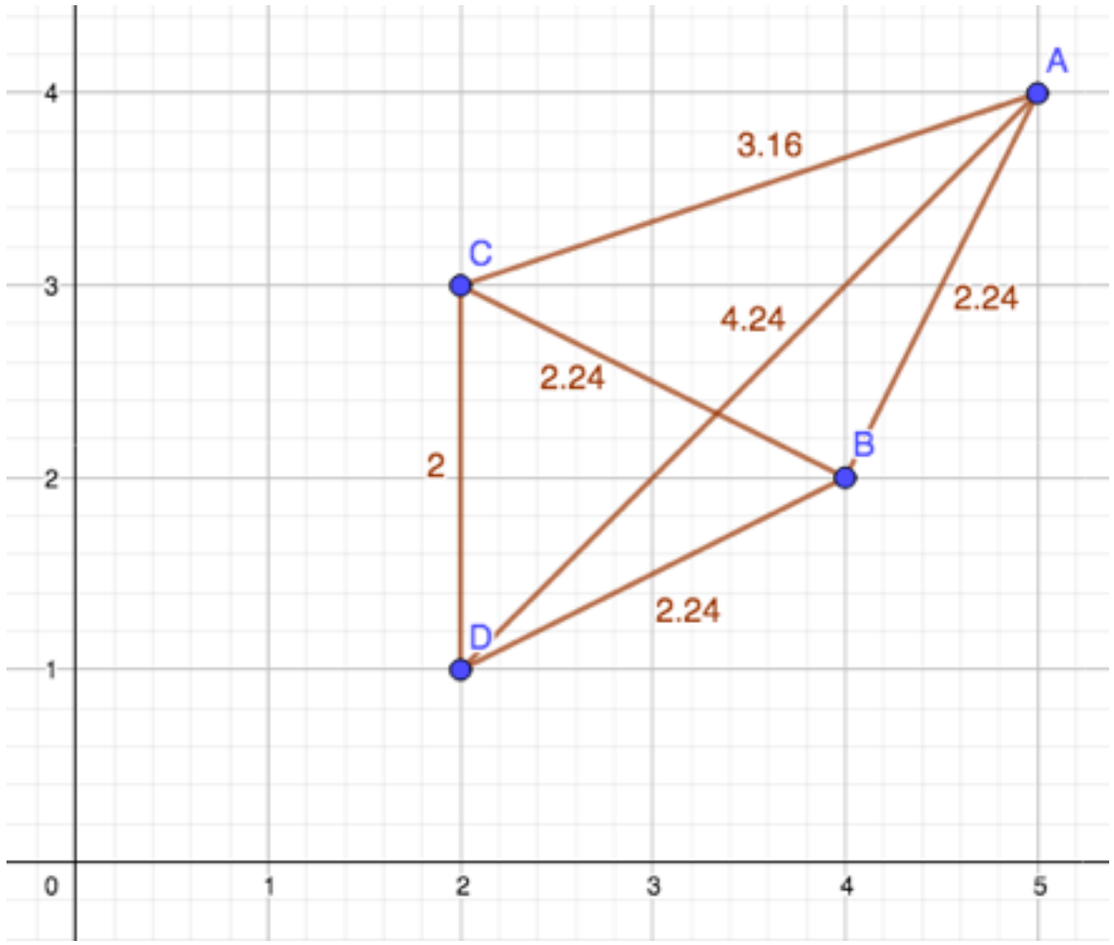
ACO:	Ant colony optimization
APO:	Artificial physics optimization
BB-BC:	Big bang-big crunch
BFO:	Bacterial forging optimization
BGSA:	Binary gravitational search algorithm
BIS:	Biological immune system
BQEA:	Binary Quantum-inspired evolutionary algorithm
CFO:	Central force optimization
CQACO:	Continuous quantum ant colony optimization
CSS:	Charged system search
EAPO:	Extended artificial physics optimization
ECFO:	Extended central force optimization
EM:	Electromagnetism-like heuristic
GA:	Genetic Algorithm
GbSA:	Galaxy-based search algorithm
GIO:	Gravitational interaction optimization
GSA:	Gravitational search algorithm
HO:	Hysteretic optimization
HQGA:	Hybrid quantum-inspired genetic algorithm
HS:	Harmony search
IGOA:	Immune gravitation inspired optimization algorithm
IQEA:	Improved quantum evolutionary algorithm
LP:	Linear programming
MOGSA:	Multiobjective gravitational search algorithm
NLP:	Nonlinear programming
PSO:	Particle swarm optimization
PSOGSA:	PSO gravitational search algorithm
QBSO:	Quantum-inspired bacterial swarming optimization
QEA:	Quantum-inspired evolutionary algorithm
QGA:	Quantum-inspired genetic algorithm
QGO:	Quantum genetic optimization
QICA:	Quantum-inspired immune clonal algorithm
QPSO:	Quantum-behaved particle swarm optimization
QSE:	Quantum swarm evolutionary algorithm
RQGA:	Reduced quantum genetic algorithm
SA:	Simulated annealing
TSP:	Travelling salesman problem
UBB-CBC:	Unified big bang-chaotic big crunch
VM-APO:	Vector model of artificial physics optimization
vQEA:	Versatile quantum-inspired evolutionary algorithm.

Biswas, Anupam, et al. "Physics-inspired optimization algorithms: a survey." Journal of Optimization 2013 (2013).

Examples

1) Traveling Salesman Problem (TSP)

$$
\begin{aligned}
X &= (x_1, x_2, ..., x_N) \\
&= \{1, 2, 3, ..., N\} \\
\mathcal{H}(X) &= \sum_{i=1}^{N} d(x_i, x_{i+1}) \\
x_{N+1} &= x_1 \\
X &\rightarrow \hat{P}[1, 2, 3, ..., N]
\end{aligned}
$$

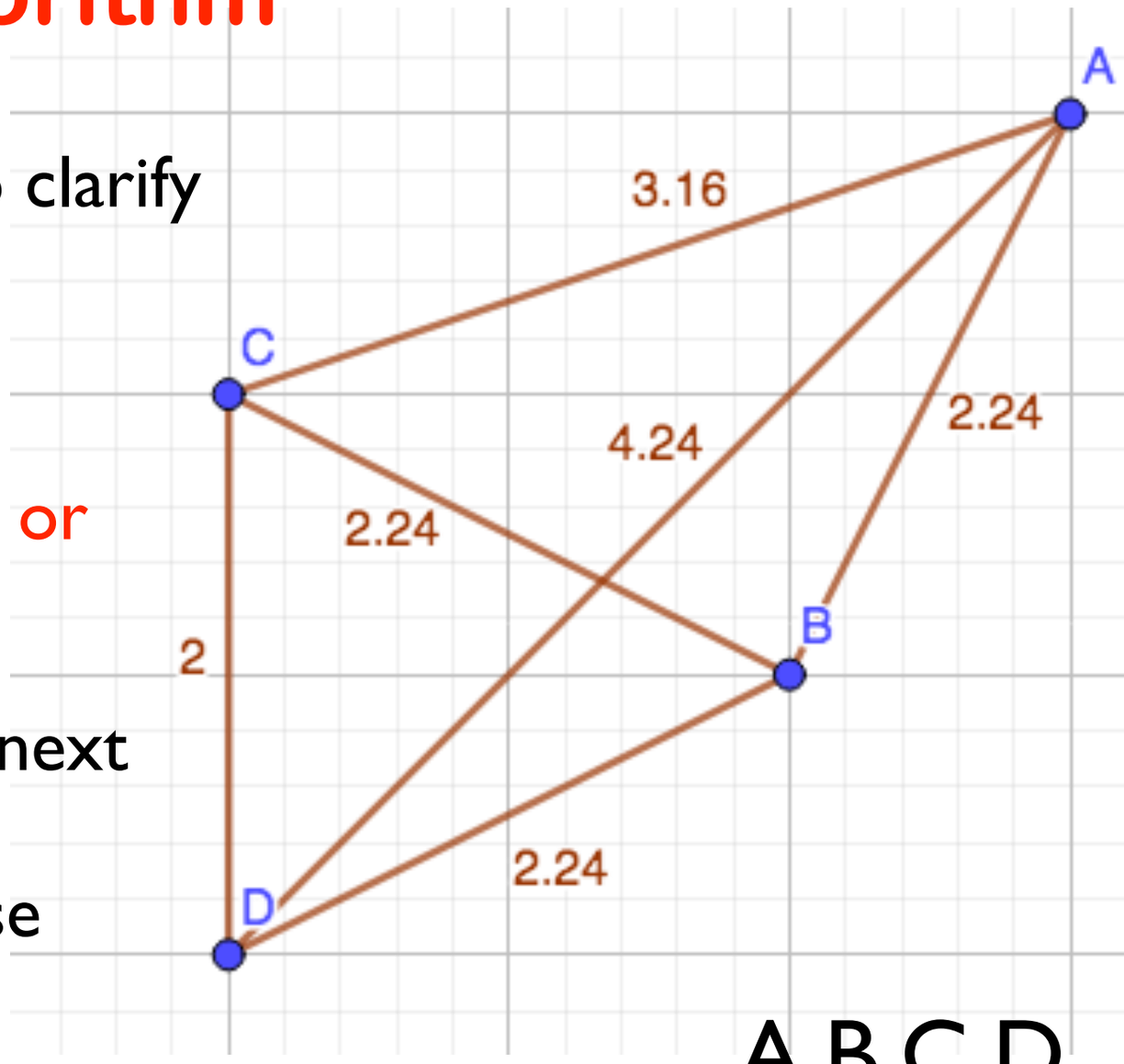


TSP Algorithm

1) Set the labels of each city to zero to clarify the times of visit
2) Starting from an arbitrary city
3) Traveling to another unvisited city

This can be done either in deterministic or stochastic approaches

1) For each given starting point select next unvisited destination randomly
2) Check the conditions of our purpose



A,B,C,D
A,B,D,C
C,D,B,A
D,C,B,A

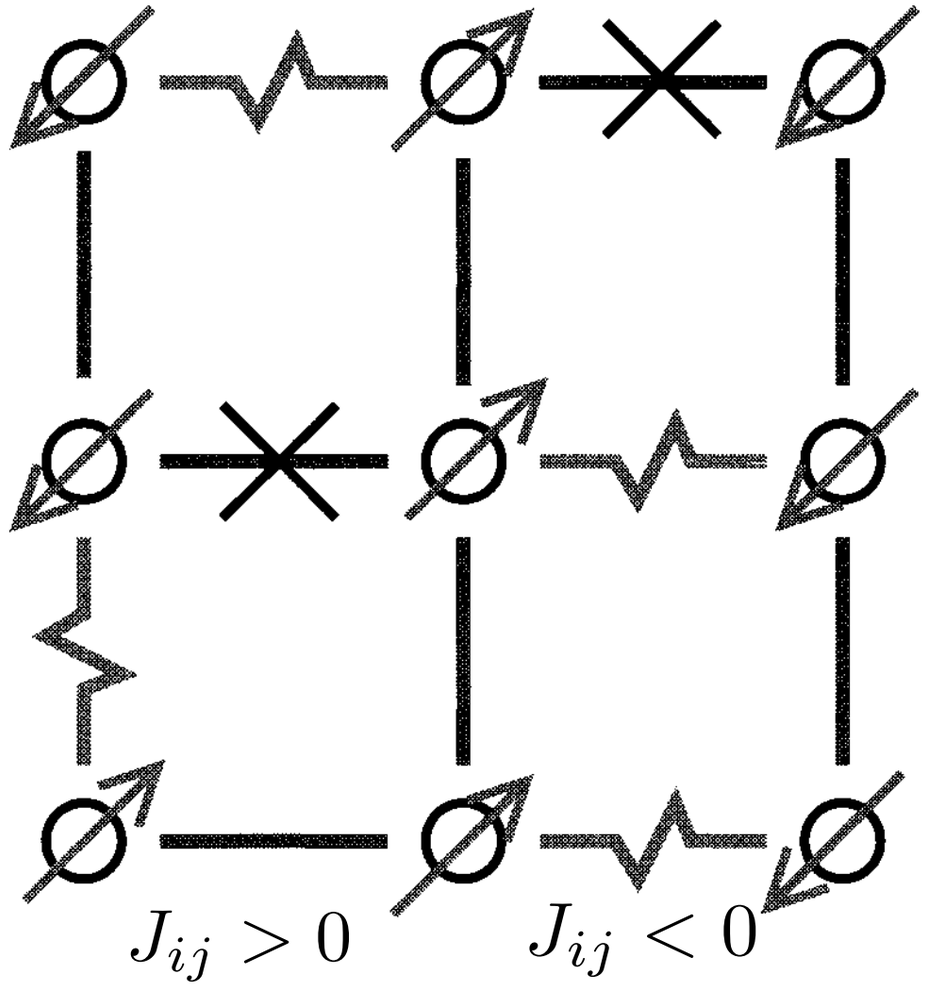
Exercise: Try to solve TSP according to following conditions:

- Visit twice C-City
- Visit necessarily C before D

Examples

2) Ising Spin Glasses

$$
\begin{aligned}
X &= (\sigma_1, \sigma_2, ..., \sigma_N) \\
&= \{1, 2, 3, ..., N\} \\
X &= \{-1, +1\} \\
\mathcal{H}(X) &= -\sum_{\langle i,j \rangle = 1}^{N} J_{ij}\sigma_i\sigma_j \\
\sigma_i &= \pm 1, \quad \sigma_i \,/\!/\, \sigma_{i+1} \quad \text{for} J_{ij} > 0 \\
\sigma_i &= \pm 1, \quad \sigma_i \,/\!\!/\, \sigma_{i+1} \quad \text{for} J_{ij} < 0
\end{aligned}
$$



Ferromagnetic and anti-Ferromagnetic

frustrated states

Terminal: General properties

Look at the

http://facultymembers.sbu.ac.ir/movahed/attachments/Introduction%20to%20command%20Linux.pdf

http://facultymembers.sbu.ac.ir/movahed/attachments/computational_all.pdf

- Making alias and unalias (Local capability):

Example 1: ls -> show the list of content in the current location;

alias "list" instead of "ls"

```
Seyeds-MacBook-Pro-1047:~ sadegh$ alias list="ls"

Seyeds-MacBook-Pro-1047:~ sadegh$ unalias list
```

Example 2: making an alias to open a typical program

```
Seyeds-MacBook-Pro-1047:Desktop sadegh$ alias math="open
-a Mathematica"
```

Terminal: General properties

- Making alias and unalias (Global capability):
  nano (emacs) ~/.bashrc

```
alias texedit='open -a TextEdit'
```

to active new alias: source ~/.bashrc

some useful commands

http://facultymembers.sbu.ac.ir/movahed/attachments/computational_all.pdf

rm -r —> delete a folder

cp -r —> copy folder

mkdir -p —> create a folder (enforcement)

rm -r ./*/ —> remove the folders inside the folder

Bash script

Some main questions:

1) What is the Bash script good for?
2) What is the Bash script itself?
3) How can make a Bash script?

What is the Bash script good for?

1) Making recipe;
2) Including different commends ranging from making a folder to call the compiler to compile and then run an executive program and so on;

Bash script: Structure

#! (shebang (hashbang) character):
Number sign+ exclamation sign

```
#!/usr/bin/env bash
i=0
num=100
for((i=1; i<=num; i++)); do
mkdir -p sadegh.${i}
name=sadegh.${i}
cp danial_story.jpg ${name}
echo ${name}
done
```

usr: Universal System Resources

Bash script: Structure

#! (shebang (hashbang) character):
Number sign+ exclamation sign

```
#!/usr/bin/env bash
i=0
num=100
for((i=1; i<=num; i++)); do
mkdir -p sadegh.${i}
name=sadegh.${i}
cp danial_story.jpg ${name}
echo ${name}
done
```

usr: Universal System Resources

To make an executive file: change the mode via

chmod u+x file.sh

Bash script: Example 2

Example 2: Make a bash script to do following tasks:
reading from a file and make associated folders and plot input data

Bash script: Example 2

Example 2: Make a bash script to do following tasks:
reading from a file and make associated folders and move a typical file to each created folder

```
#!/usr/bin/env bash
i=0
for name in `cat input` ; do
let "i=i+1"
C[i]=$name
echo $name
mkdir -p $name
cp danial_story.jpg ${name}
done
```

Bash script: Example 3

Example 3: Make a bash script to do following tasks:

1) We have 48 text file entitled 1.txt to 48.txt;

2) We have a file including the name of countries and we would like to assign each text file to the corresponding country's name in separated folders. Also we are going to select all available pairs (all combinations) $\frac{48!}{(2)!(48-2)!} = 1128$

3) Move each two corresponding data to associated folder and plot the data in that folder

Bash script: Example 3

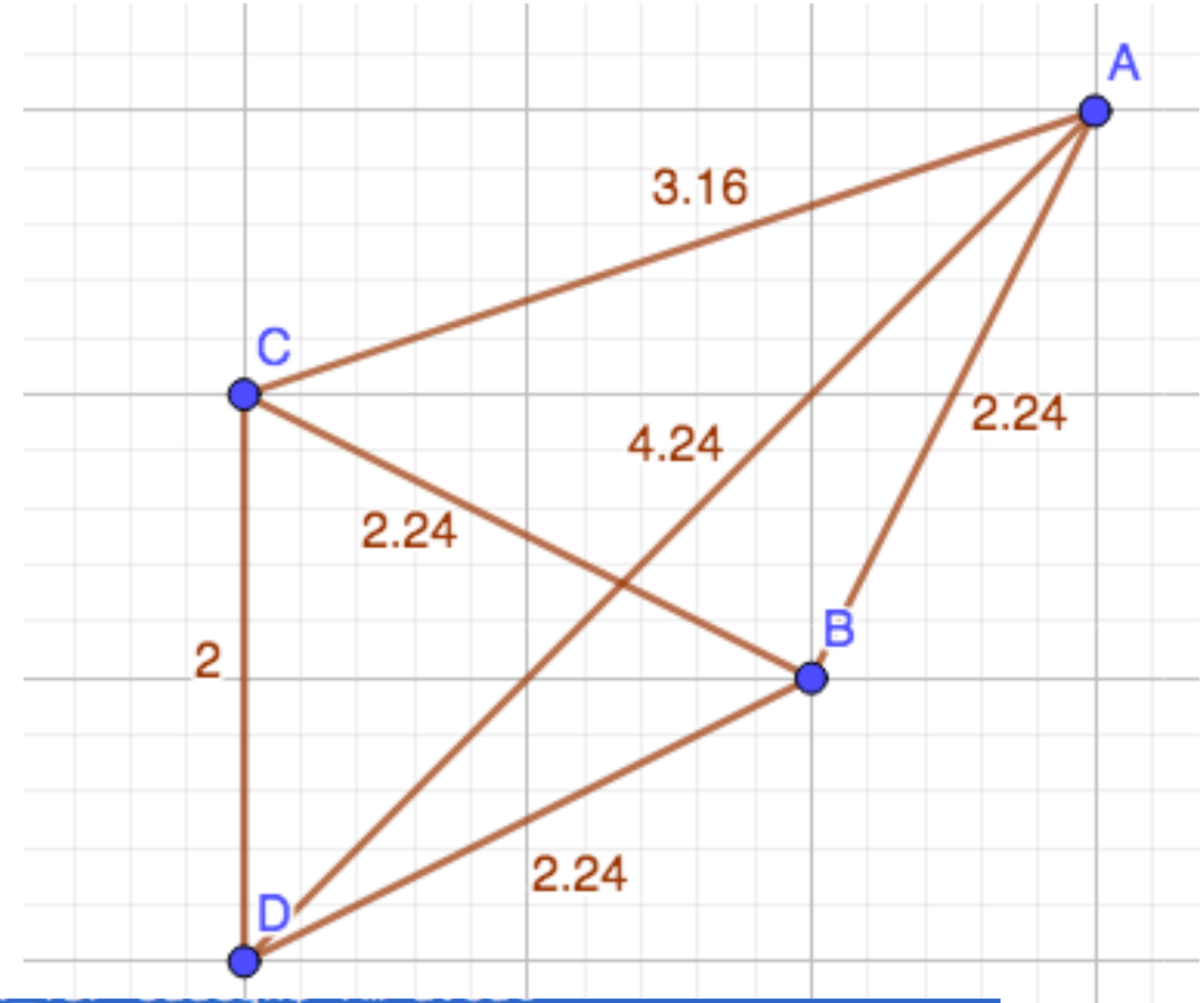
```
#!/usr/bin/env bash

i=0
for name in $(cat list_arrange); do
    let i=$i+1
    c[i]=$name
    #echo $name
done
let num=$i
for ((i=1; i<=$num; i++)); do
    let k=$i+1
    for ((j=$k; j<=$num; j++)); do
        mkdir -p ${c[i]}_${c[j]}
        cp $i.txt ${c[i]}_${c[j]}/${c[i]}.txt
        cp $j.txt ${c[i]}_${c[j]}/${c[j]}.txt
        echo ${c[i]}
        echo ${c[j]}
        cd ${c[i]}_${c[j]}
        python3.6 ../plot.py ${c[i]} ${c[j]}
        cd ..
    done
done
```

Bash script: Example 4

Example 4: Traveling Salesman Problem (TSP)

A,B,C,D
A,B,D,C
C,D,B,A
D,C,B,A



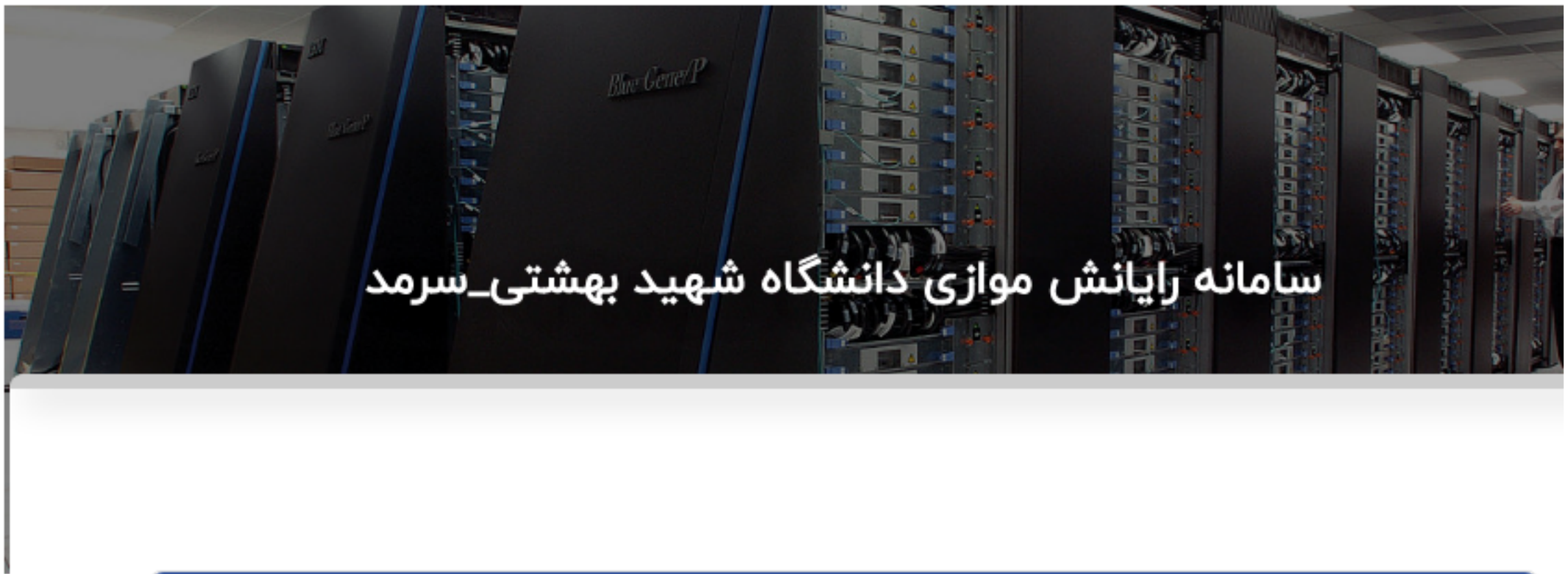

```
(base) Seyeds-MacBook-Pro-1047:example4_TSP sadegh$ gfortran TSP_random.f90
(base) Seyeds-MacBook-Pro-1047:example4_TSP sadegh$ ./a.out
6.48 1 2 4 3
6.48 1 2 3 4
6.48 3 4 2 1
6.48 4 3 2 1
```

Bash script: Example 5

Bash script: Example 6

Others left as exercises for you

Sarmad



https://resevp.sbu.ac.ir/sarmad

Some useful commands

Example 1: we are interested in copying a file from our machine to cluster

scp ./plot.py m_movahed@192.168.220.100:/share/users/m_movahed/TDA

Example 2: after finishing our program in the cluster, we want to move the results from cluster to our local machine

scp m_movahed@192.168.220.100:/share/users/m_movahed/TDA/plot1.py .

Notice: Use “tmux” when you are connected to cluster

Some useful commands:
tmux command

1) Connecting to the cluster
2) In corresponding terminal type: `tmux` (pre-installed) to create a session
3) `tmux LS` (shows a list of sessions)
4) `CL+b %` (splitting vertically the terminal)
5) `CL+b "` (splitting horizontally the terminal)
6) moving between different sessions
   `CL+b arrows (top, down, left, right)`
7) Submitting a job and running a program
8) `CL+b d` ——> to Detach from session
9) `tmux a -t <session-ID>`
10) Exit (disconnecting from cluster)
11) To check our job connect to cluster, `tmux LS`, `tmux a -t <session-ID>`
12) To kill the session, `tmux a -t <session-ID>`, `CL+b:` `type kill-session`

see the Pooyan's lectures for more details via
http://ccg.sbu.ac.ir/resources/computers/

Some useful commands

Notice: Use “tmux” when you are connected to cluster
after reconnecting use “tmux attach”

Example:

```
while true
do
sleep 1
echo “Hello Dear”
done
```

see the Pooyan’s lectures for more details via
http://ccg.sbu.ac.ir/resources/computers/

run your job on a cluster

1) It essentially needs to make the Bash scrip; (see the example)
2) Shell managing (terminal managing)

see the end of this file:
http://facultymembers.sbu.ac.ir/movahed/attachments/computational_all.pdf

Number Representation

Error estimation and propagation