Project Proposals

Optimization of Complex Systems

Andrea Brilli
Dipartimento di Ingegneria Informatica, Automatica e Gestionale
Sapienza University of Rome

Academic Year 2025–2026

Structure of each project. Every proposal consists of two components: an **Implementation (I)** part, where the student codes the algorithm(s) and runs numerical experiments on test problems, and a **Theory (T)** part, where the student studies convergence properties, complexity bounds, and/or theoretical guarantees of the proposed method. Difficulty is rated as: *low* < *medium/low* < *medium* < *medium/high* < *high*.

The proposals below are *reference points* and can be freely combined. Upon request, and if consistent with the course, a project on a topic of personal interest is also possible.

Contents

1. Derivative-Free Optimization

1.1. DFO 1 — Line-search derivative-free algorithms

Idea. Standard derivative-free optimization relies on direct-search strategies based on polling directions. An alternative paradigm couples derivative-free function evaluations with a **line-search** framework: rather than sampling on a mesh, the algorithm selects a direction and performs an inexact line search along it without ever computing $\nabla f(x)$.

Project tasks:

- Implement a line-search DFO algorithm.
- Study its theoretical properties (stationarity, convergence).
- Compare numerically with the direct-search methods seen in class.

Difficulty Implementation (I): **medium** Theory (T): **medium**

1.2. DFO 2 — Mesh-based derivative-free algorithms

Idea. Mesh-based DFO methods (e.g. MADS, GPS) enforce that iterates always lie on a structured grid whose resolution decreases over iterations. This **mesh constraint** provides a powerful theoretical tool to establish convergence to Clarke stationary points even for non-smooth objectives, at the cost of a more constrained search.

Project tasks:

- Implement a mesh-enforcing DFO algorithm.
- Carry out the theoretical convergence analysis.
- Test on standard DFO benchmarks.

Difficulty Implementation (I): **medium/low** Theory (T): **high**

1.3. DFO 3 — Linear model construction in DFO

Idea. Model-based DFO methods build a **local linear (or quadratic) approximation** of f from function evaluations, then use it as a surrogate to drive the search. This project focuses on the construction and maintenance of good linear models.

Project tasks:

- Implement a linear-model-based DFO algorithm.
- Study the theory of model accuracy and its impact on convergence.
- Test on standard DFO benchmarks.

Difficulty Implementation (I): **medium/low** Theory (T): **medium/high**

2. Multiobjective Optimization

2.1. MO 1 — Scalarization methods

> **Idea.** Scalarization converts the multiobjective problem into a family of **single-objective problems** parametrized by weights or reference points (e.g. weighted sum, ε-constraint). By solving this family one recovers an approximation of the entire Pareto front Ω_P.

Project tasks:

- Implement multiple scalarization schemes.
- Study theoretical properties.
- Compare the quality of the recovered Pareto front across methods.

Difficulty Implementation (I): **medium/low** Theory (T): **medium/low**

2.2. MO 2 — Single-sequence steepest-descent

> **Idea.** Unlike scalarization, **single-sequence** methods maintain one iterate $x_k \in \Re^n$ and move it towards the Pareto front by computing, at each step, a *common descent direction* for all objectives simultaneously, the multiobjective analogue of the steepest-descent direction, obtained via the stationarity measure $\theta(x)$ seen in class.

Project tasks:

- Implement the multiobjective steepest-descent algorithm.
- Study convergence to Pareto-stationary points.
- Test on benchmark problems.

Difficulty Implementation (I): **medium** Theory (T): **medium/low**

2.3. MO 3 — Gradient-based direct Pareto front approximation

> **Idea.** Rather than finding a single Pareto-optimal point, these methods maintain a **lists of iterates** and use gradient information to move them towards an approximation of the Pareto front Ω_P.

Project tasks:

- Implement a gradient-based front-approximation algorithm.
- Study theoretical guarantees.

Difficulty Implementation (I): **high** Theory (T): **medium/high**

> **Recommended for groups of 2.**

2.4. MO 4 — Derivative-free direct Pareto front approximation

> **Idea.** This project combines the challenges of DFO and multiobjective optimization: approximate the Pareto front Ω_P of a multiobjective problem when **no gradient information is available**. Direct-search strategies must be adapted to handle multiple conflicting objectives and maintain a diverse approximation set.

Project tasks:

- Implement a DFO algorithm targeting direct Pareto front approximation.
- Develop or adapt the convergence theory to the derivative-free setting.

Difficulty Implementation (I): **high** Theory (T): **high**

> **Recommended for groups of 2.**

3. Constraint-Handling Techniques

3.1. CH 1 — Interior gradient-based methods

> **Idea.** Interior methods handle constraints by keeping all iterates **strictly feasible**. Barrier functions (e.g. logarithmic barriers) add a penalty term that blows up near the boundary of the feasible set Ω, naturally preventing constraint violations. Gradient-based steps are then taken on the modified objective.

Project tasks:

- Implement an interior-point gradient-based method.
- Study the theoretical guarantees.

Difficulty Implementation (I): **medium** Theory (T): **medium/high**

3.2. CH 2 — Exterior gradient-based methods (sequential penalty)

> **Idea.** Exterior (or *sequential penalty*) methods allow iterates to be **infeasible**, but penalize constraint violations by adding an infeasibility term to the objective, and a penalty parameter $\rho > 0$ is progressively increased to drive iterates towards the feasible region. Gradient information is exploited to minimize the penalized objective at each outer iteration.

Project tasks:

- Implement a sequential penalty gradient-based algorithm.
- Study theoretical convergence.

Difficulty Implementation (I): **medium** Theory (T): **medium/high**

3.3. CH 3 — Exterior derivative-free methods (sequential penalty)

> **Idea.** This project extends the sequential penalty approach to the **derivative-free** setting: gradient information is unavailable, so derivative-free strategies must be used to minimize the penalized objective at each outer iteration. The interplay between penalty growth and step-size reduction makes this problem particularly delicate.

Project tasks:

- Implement a sequential penalty DFO algorithm.
- Study how penalty growth and step-size schedules must be coordinated for convergence.

Difficulty Implementation (I): **high** Theory (T): **medium/high**

3.4. CH 4 — Interior derivative-free methods

> **Idea.** Interior DFO methods maintain **strict feasibility** at every iterate while relying solely on function evaluations, no gradient of either f or the constraint functions is available. Adapting barrier to the derivative-free setting requires careful management of the interplay between barrier parameter reduction and the polling step size.

Project tasks:

- Implement an interior DFO method.
- Study the convergence theory in the derivative-free setting.

Difficulty Implementation (I): **high** Theory (T): **high**

Project overview

Code	Title	I	T
DFO 1	Line-search DFO algorithms	medium	medium
DFO 2	Mesh-based DFO algorithms	medium/low	high
DFO 3	Linear model construction	medium/low	medium/high
MO 1	Scalarization methods	medium/low	medium/low
MO 2	Single-sequence steepest-descent	medium	medium/low
MO 3	Gradient-based front approx.*	high	medium/high
MO 4	DFO front approximation*	high	high
CH 1	Interior gradient-based	medium	medium/high
CH 2	Exterior gradient-based (penalty)	medium	medium/high
CH 3	Exterior DFO (penalty)	high	medium/high
CH 4	Interior DFO	high	high

* recommended for groups of 2