A modeling-language based approach to automatically recommend optimization methods

Sofiane Tanji

ICTEAM/INMA
Université catholique de Louvain

Joint work with François Glineur (UCLouvain)

ISMP 2024 | Tuesday, July 23, 2024

Talk Outline

1. Motivation and context
2. Step 1 : Modeling language
3. Step 2 : Literature representation
4. Step 3 : Problem reducibility
5. Step 4 : Ranking of methods
6. Conclusions

Motivating example: LASSO regression

Consider $X \in \mathbb{R}^{n\times d}$, $y \in \mathbb{R}^n$, $\lambda_1, \lambda_2, \lambda_3 \in \mathbb{R}_+^*$. LASSO regression solves:

$$\hat{\beta} \in \arg\min_{\beta\in\mathbb{R}^d} \left\{ \frac{1}{2}\|y - X\beta\|^2 + \lambda_1\|\beta\|_1 \right\}, \qquad \text{(nonsmooth convex)}$$

or equivalently,

$$\hat{\beta} \in \arg\min_{\beta\in\mathbb{R}^d} \left\{ \frac{1}{2}\|y - X\beta\|^2 \text{ subject to } \|\beta\|_1 \le \lambda_2 \right\} \qquad \text{(QP)}$$

or equivalently,

$$\hat{\beta} \in \arg\min_{\beta\in\mathbb{R}^d} \left\{ \|\beta\|_1 \text{ subject to } \|y - X\beta\|^2 \le \lambda_3 \right\} \qquad \text{(SOCP)}$$

For each formulation, you can use many different methods:

Formulation	Some applicable methods
nonsmooth convex	coordinate descent, subgradient method, proximal gradient ...
convex QPs	augmented Lagrangian, interior-point methods
SOCPs	interior-point methods

Research Questions

Q1. Where to find all methods applicable to a given formulation ?

Q2. How to find equivalent reformulations ?

Q3. How to find most efficient (formulation, method) combinations ?

What optimization problems usually look like

Consider any optimization problem in the black-box form:

$$\min_x f(x) \tag{1}$$

where f is decomposed into simpler components f_i, for example:

$$f(x) = f_1(x) + f_2 \circ f_3(x) + \sum_{j=1}^{n} f_4^j(x) - \max_{j=1,\dots,n} f_5^j(x) \tag{2}$$

with

- possible assumptions on the f_i's (convexity , Lipschitz continuity, etc.)
- access to certain oracles for each f_i (subgradient, proximal operator, etc.)

What you can find in the literature

Theorem: Worst-case convergence rate of Algorithm 1

Suppose assumptions $\{(A1), (A2), (A3)\} = \mathcal{A}$ hold.
Consider some initial conditions $\mathcal{I}$.
Then, Algorithm 1 with parameters $\mathcal{P}$ applied to Problem (1) satisfies for all $k \geq 1$

$$F(x_k) - F(x^*) \leq \varphi(k, \mathcal{A}, \mathcal{I}, \mathcal{P}) \tag{3}$$

Existing results (always) have

- Some template optimization problem (in red)
- The considered optimization method (in purple)
- A rate of convergence (in green)

In this talk

We present Optimization Methods Ranking Assistant (OMRA)

- A modeling language to describe optimization problems
- An encoding of existing results in the literature
- Detection of methods applicable to user-provided problems
- Ranking of applicable methods by worst-case performance

Pipeline

User problem $\longrightarrow$ Set of templates $\longrightarrow$ Applicable methods $\longrightarrow$ Associated rates $\longrightarrow$ Ranking of methods

2. Modeling language

3. Literature representation

5. Main output

4. Problem reduction

Related work

Encode mathematical knowledge

- Lean proof assistant[1], Linnaeus[2]

Representing optimization in standard form for computer analysis

- Modeling languages: AMPL, GAMS, YALMIP, JuMP, CVX etc.
- IQC[3] and PEP analysis[4]

Choosing the best algorithm to solve an optimization problem

- Benchopt[5]: benchmarking of optimization algorithms

---

[1] De Moura et al., "The Lean theorem prover".

[2] Zhao, Lessard, and Udell, "An automatic system to detect equivalence between iterative algorithms".

[3] Lessard, Recht, and Packard, "Analysis and design of optimization algorithms via integral quadratic constraints".

[4] Taylor, Hendrickx, and Glineur, "Smooth strongly convex interpolation and exact worst-case performance of first-order methods".

[5] Moreau et al., "Benchopt: Reproducible, efficient and collaborative optimization benchmarks".

Talk Outline

User problem $\longrightarrow$ Set of templates $\longrightarrow$ Applicable methods $\longrightarrow$ Associated rates $\longrightarrow$ Ranking of methods

2. Modeling language

3. Literature representation

5. Main output

4. Problem reduction

Optimization problem as a labeled tree

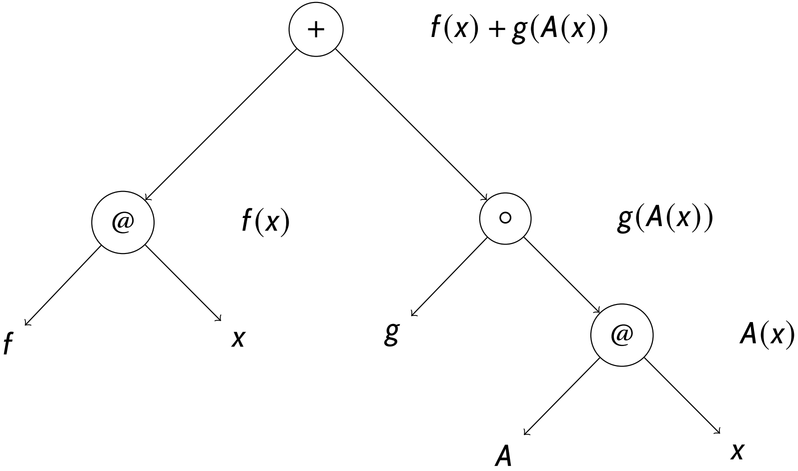
Example

$$\min_{x \in \mathbb{R}^n} f(x) + g(A(x)) \tag{4}$$

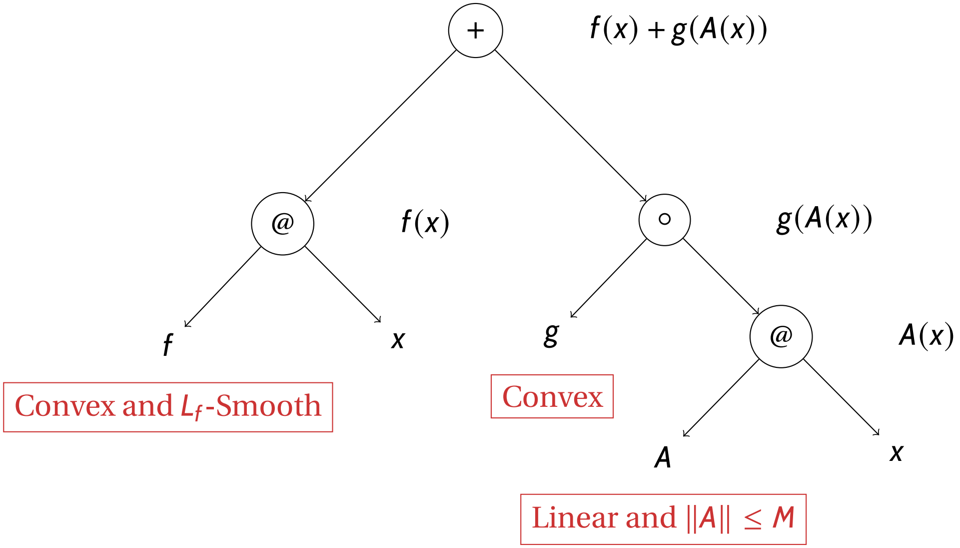
where

- $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is convex, L_f-smooth. Access to ∇f.
- $g : \mathbb{R}^m \longrightarrow \mathbb{R}$ is convex. Access to $prox_g$.
- $A : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ is a linear mapping with $||A|| \leq M$. Access to $x \longrightarrow Ax$.

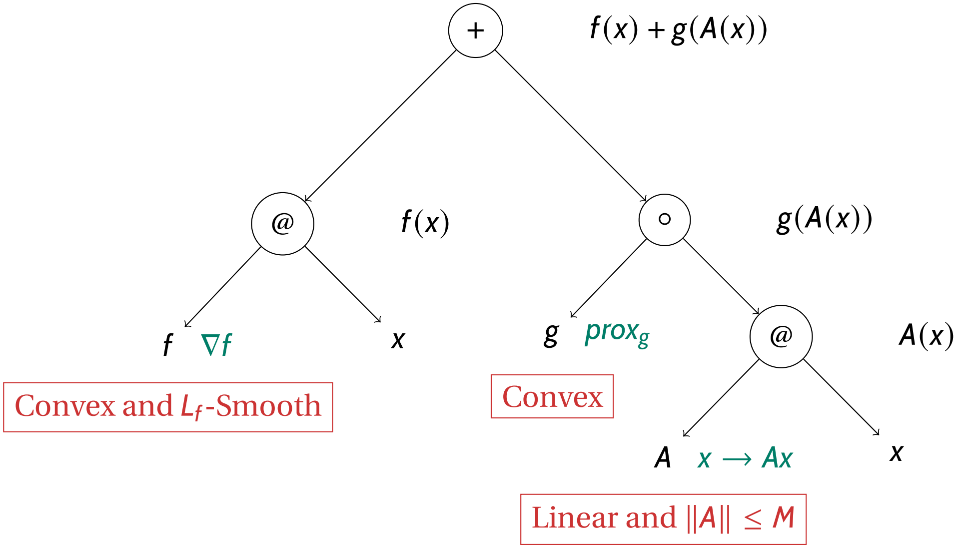
Optimization problem as a labeled tree



Optimization problem as a labeled tree



Optimization problem as a labeled tree



How to write it in OMRA

Problem

$$\min_{x \in \mathbb{R}^n} f(x) \tag{5}$$

where

- $f : \mathbb{R}^n \to \mathbb{R}$ is convex,
- f is L_f-smooth with $L_f \in]0, 10[$,
- f is available through a first-order oracle $x \to \nabla f(x)$.

```
pb = Problem()
Rn = pb.declare_space("Rn")
R = pb.declare_space("R", 1)
f = pb.declare_function("f", Rn, R)
x = pb.declare_variable("x", Rn)
f.add_property(Convex())
f.add_property(Smooth(0, 10.))
pb.set_objective(f(x))
pb.declare_oracle(Derivative(f))
```

Talk Outline

User problem $\longrightarrow$ Set of templates $\longrightarrow$ Applicable methods $\longrightarrow$ Associated rates $\longrightarrow$ Ranking of methods

2. Modeling language

3. Literature representation

5. Main output

4. Problem reduction

ABCs of encoding the literature

Claim: All convergence theorems have the following form

Theorem: Worst-case convergence rate of Algorithm 1

Suppose assumptions $\mathcal{A} = \{A1, A2, A3\}$ hold. Consider initial conditions $\mathcal{I}$. Then, Algorithm 1 with parameters $\mathcal{P}$ applied to Problem (1) satisfies for all $k \geq 1$

$$F(x_k) - F(x^*) \leq \varphi(k, \mathcal{A}, \mathcal{I}, \mathcal{P}) \tag{6}$$

meaning we have to encode three elements

- Template problem (in red)
- Method parameters (in purple)
- Convergence rate (in green)

Elements to encode the literature

Template problem

✔ Just an optimization problem !

Convergence rate

Example: Convergence rate of fixed-step (γ) GD for f convex and L-smooth

$$\underbrace{f(x_N) - f_*}_{\text{performance measure}} \leq \frac{L}{2} \frac{\overbrace{\|x_0 - x^*\|^2}^{\text{initial conditions}}}{1 + \gamma L \min\left\{2N, \frac{-1+(1-\gamma L)^{-2N}}{\gamma L}\right\}}$$

and method parameter γ and template parameter L.

Optimization method

- Enough to encode the parameters !
- Dependence of parameters to unknown quantities

First answer !

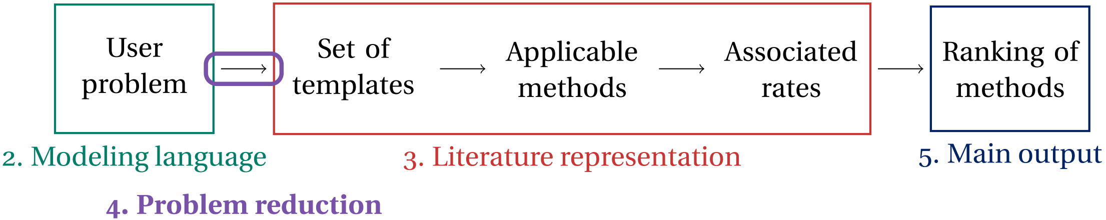
Research Questions

- ✓ Where to find all methods applicable to a given formulation ?
- Q2. How to find equivalent reformulations ?
- Q3. How to find most efficient (formulation, method) combinations ?

Answer to Q1

- Results are scattered in the literature
- We aggregated many existing results in OMRA *(and we'll keep doing so)*

Talk Outline



Scenario 1 : immediate matching

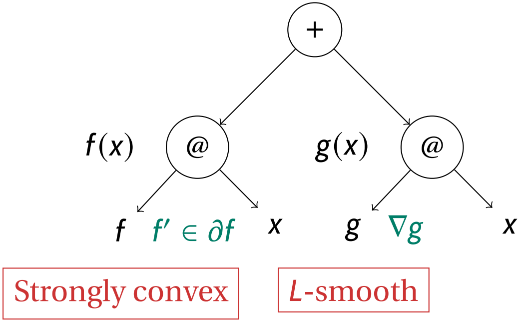


Figure: Optimization Problem P

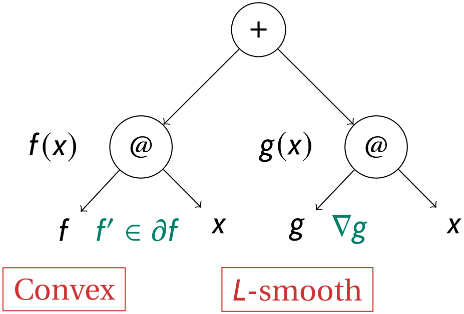


Figure: Optimization Template T

Scenario I : immediate matching

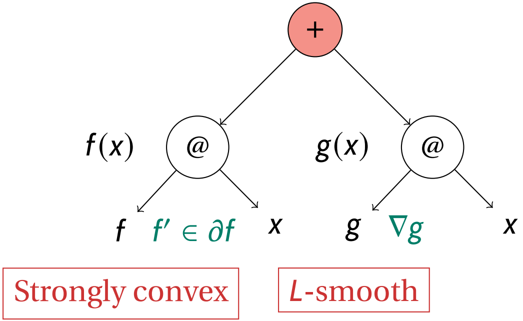


Figure: Optimization Problem P

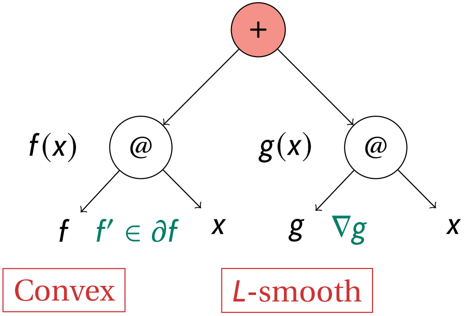


Figure: Optimization Template T

Scenario 1 : immediate matching

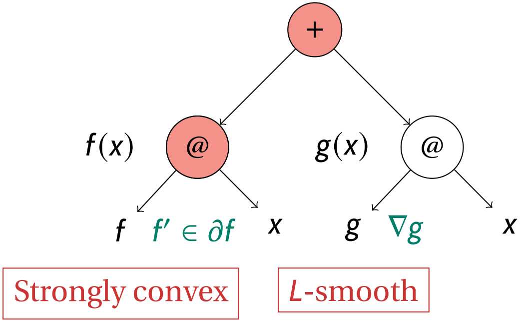


Figure: Optimization Problem P

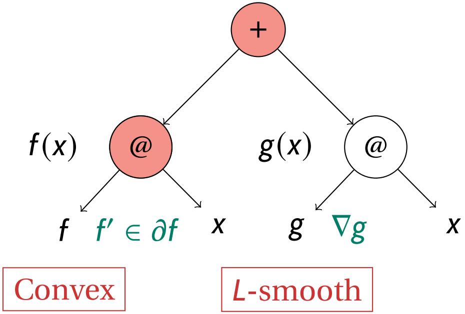


Figure: Optimization Template T

Scenario 1 : immediate matching

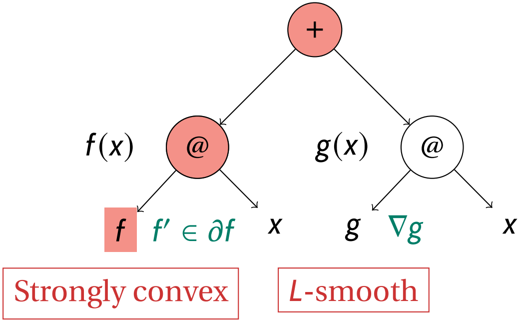


Figure: Optimization Problem P

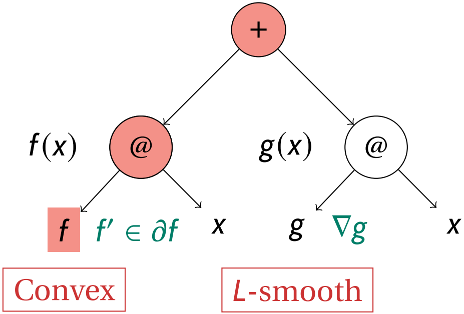


Figure: Optimization Template T

Scenario 1 : immediate matching

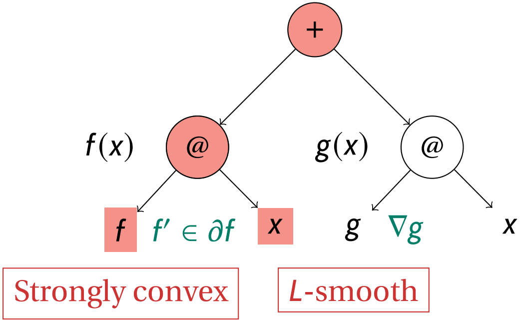


Figure: Optimization Problem P

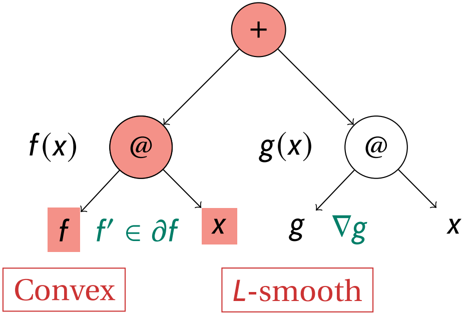


Figure: Optimization Template T

Scenario 1 : immediate matching

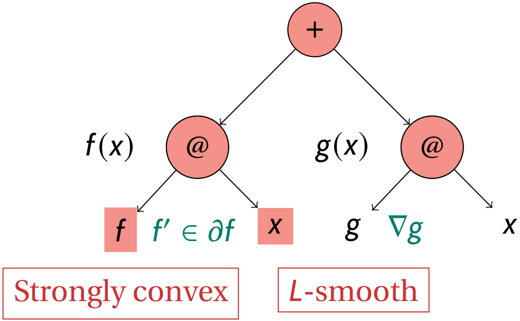


Figure: Optimization Problem P

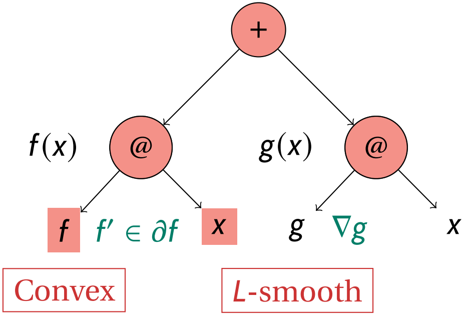


Figure: Optimization Template T

Scenario I : immediate matching

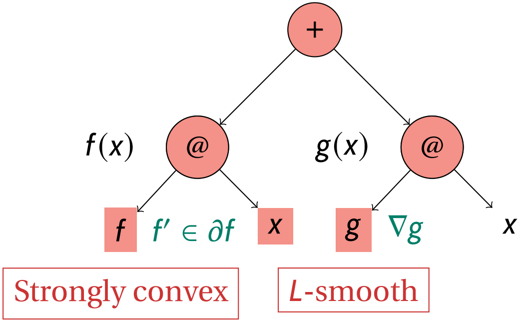


Figure: Optimization Problem P

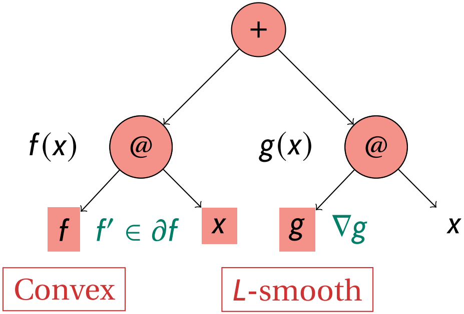


Figure: Optimization Template T

Scenario I : immediate matching

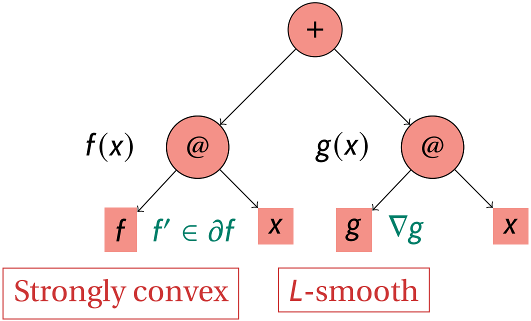


Figure: Optimization Problem P

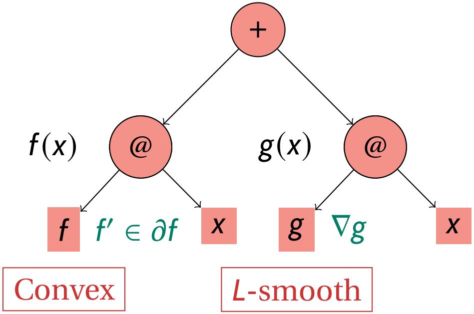


Figure: Optimization Template T

Scenario 2 : reformulations

We need to match user-provided problems to templates.

- ✔ Ideal scenario is immediate match of user-provided problem to template.
- Solution otherwise: use *mathematical results and reformulation tricks !*

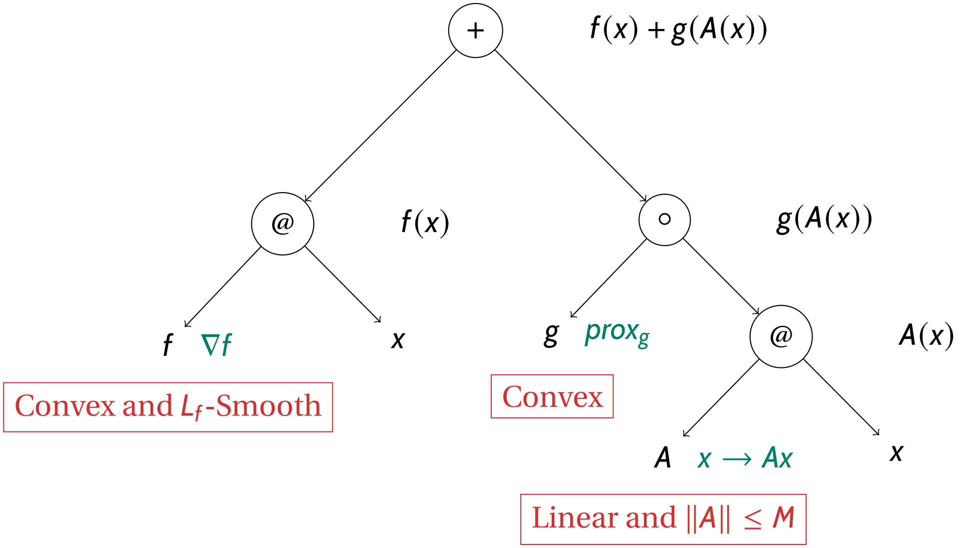
Mathematical results:

- Sum of smooth (resp. convex) functions is smooth (resp. convex),
- Proximal operator of $f \circ A$ is computable given $prox_f$, A and A^* ...

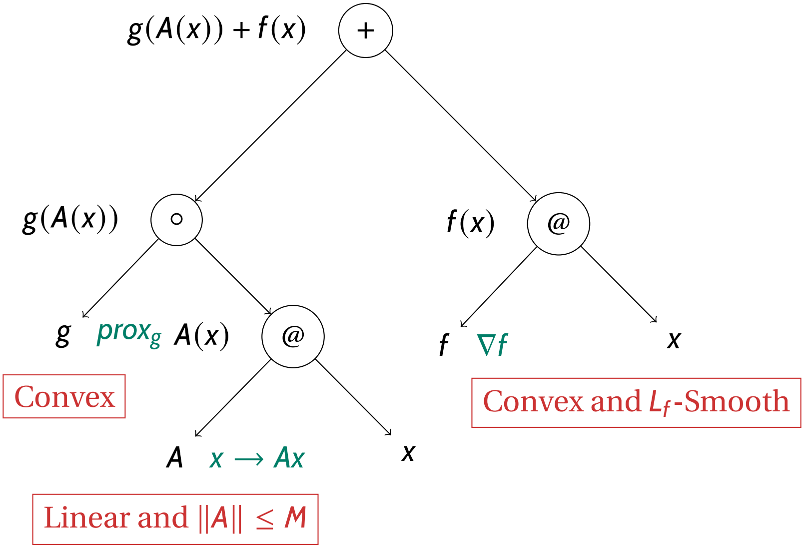
Reformulation tricks:

- Commutativity of operators,
- losing structure,
- regrouping terms,
- transfer of conditioning ...

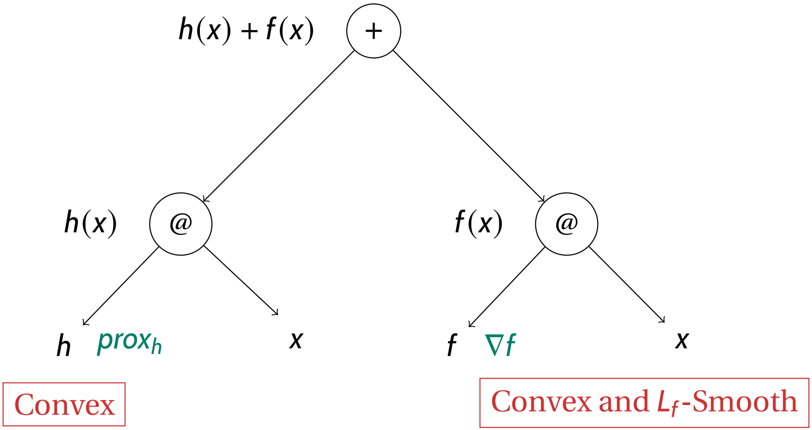
Reformulation example



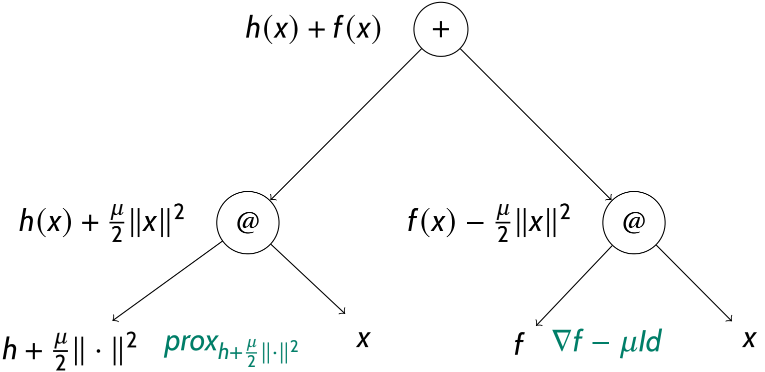
Commutativity of the sum *operator*



Losing structure



Transfer of conditioning



$\boxed{\text{Strongly convex}}$

$\boxed{\text{Convex and } (L_f - \mu)\text{-Smooth}}$

Second answer !

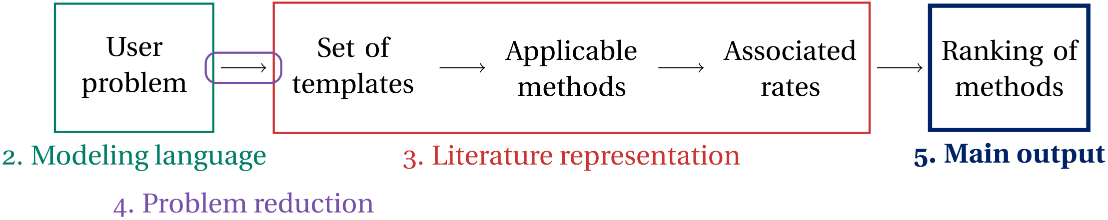
Research Questions

- ✓ Where to find all methods applicable to a given formulation ?
- ✓ How to find equivalent reformulations ?
- Q3. How to find most efficient (formulation, method) combinations ?

Answer to Q2

- ▶ Apply reformulation tricks on the original problem

Talk Outline



So far: from a user-provided problem, we get a list of (Template, Method, Rate)

Goal:

Q3. How to find most efficient (template, method) combinations ?

Our ranking criterion:

The convergence rate associated to each (template, method) combination

How to deal with sophisticated rate functions

Example: Convergence rate of fixed-step (γ) GD for f convex and L-smooth

$$f(x_N) - f_* \leq \frac{L}{2} \frac{\|x_0 - x^*\|^2}{1 + \gamma L \min\left\{2N, \frac{-1+(1-\gamma L)^{-2N}}{\gamma L}\right\}}$$

1. Compute rates numerically whenever possible
2. Drop asymptotically worse methods if high iteration budget
3. Compare leading coefficients whenever possible
4. *Sampling method*

Research questions

Remember the research questions

- ✓ Where to find all methods applicable to a given formulation ?
- ✓ How to find equivalent reformulations ?
- ✓ How to find most efficient (formulation, method) combinations ?

Here are some answers

A1. Query the knowledge database of OMRA.

A2. Apply standard reformulation to tree representation of a problem

A3. Rank the rates associated to matched (template, method) pairs

Conclusions

Contribution: a principled approach and its Python implementation, OMRA

- Modeling language to describe your optimization problem
- Automatic computation of equivalent reformulations
- Matching reformulations to known templates
- Retrieve methods applicable to above templates
- Retrieve known worst-case convergence rates
- Guess best performing method using retrieved rates

Next steps

- *Make this encyclopedia available through a website **(very soon)***

Enrich the toolbox

- Aggregate more results in the database
- Add reformulation techniques (η-trick, duality (for the LASSO example))
- Include "numerical" PEP results in the database

User features

- Identification of settings for which a method performs above average
- Code generation for recommended methods

Thank you again for your attention!

Questions ?

Do not hesitate to contact me:

→ sofiane.tanji@uclouvain.be

→ https://sofianetanji.com