

# ORDER-VALUE OPTIMIZATION AND NEW APPLICATIONS

José Mario Martínez

[www.ime.unicamp.br/~martinez](http://www.ime.unicamp.br/~martinez)

UNICAMP, Brazil

August 2, 2011

## Collaborators

Roberto Andreani (Applied Math - UNICAMP)

Leandro Martínez (Chemistry - UNICAMP)

Flávio Yano (Itaú Bank)

Mário Salvatierra (Fed. Univ. Amazonas)

Giovane César (Applied Math - UNICAMP)

Roberto Marcondes (Computer Science - USP)

Paulo J. Silva (Computer Science - USP)

Cibele Dunder (Itaú Bank)

Luís Felipe Bueno (Applied Math - UNICAMP)

Lucas Garcia Pedroso (Applied Math - UNICAMP)

Maria Aparecida Diniz (Applied Math - UNICAMP)

Ernesto Birgin (Computer Science - USP)

# Outline

- Introduce Order-Value Optimization problems
- Review of Algorithms and Convergence results
- Applications

# Order-Value Optimization (OVO) Problems

Let  $f_i : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $i = 1, \dots, m$   
 $J \subset \{1, \dots, m\}$

For all  $x \in \Omega$ , we define  $i_1(x), i_2(x), \dots, i_m(x)$  by

$$f_{i_1(x)}(x) \leq f_{i_2(x)}(x) \leq \dots \leq f_{i_m(x)}(x)$$

The OVO problem is:

$$\underset{x \in \Omega}{\text{minimize}} \quad \sum_{j \in J} f_{i_j(x)}(x)$$

## Examples

$$J = \{m\} \longrightarrow \min_{x \in \Omega} \max\{f_1(x), \dots, f_m(x)\}$$

$$J = \{1\} \longrightarrow \min_{x \in \Omega} \min\{f_1(x), \dots, f_m(x)\}$$

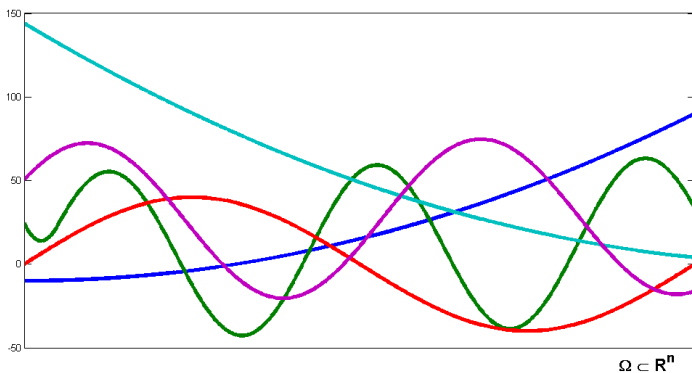
$$J = \{p\} \longrightarrow \min_{x \in \Omega} f_{i_p(x)}(x) \quad (\text{VaR-Like})$$

$$J = \{1, \dots, p\} \longrightarrow \min_{x \in \Omega} \sum_{j=1}^p f_{i_j(x)}(x) \quad (\text{LOVO})$$

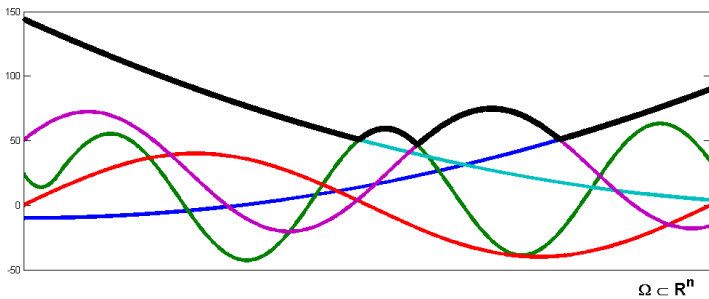
$$J = \{p+1, \dots, m\} \longrightarrow \min_{x \in \Omega} \sum_{j=p+1}^m f_{i_j(x)}(x) \quad (\text{CVaR-Like})$$

$$J = \{q+1, \dots, p\} \longrightarrow \min_{x \in \Omega} \sum_{j=q+1}^p f_{i_j(x)}(x)$$

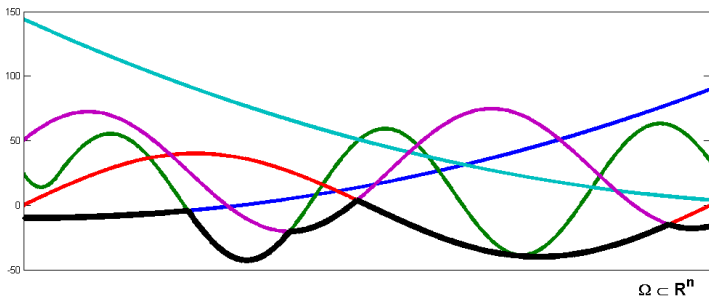
# Non-smoothness and Many local minimizers



$$J = \{m\} \text{ (Minimax)}$$

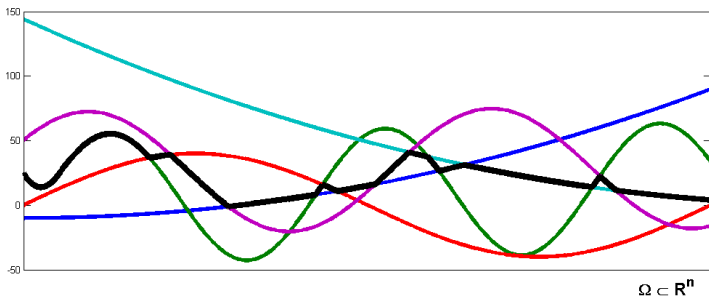


$$J = \{1\} \text{ (Minimin)}$$





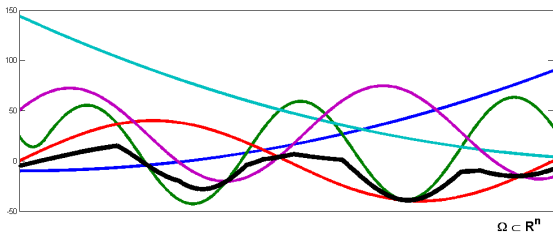
$$J = \{p\}, \text{ (VaR-like)}$$



In this example,  $p = 3$

$$J = \{1, \dots, p\}, \text{ (LOVO)}$$

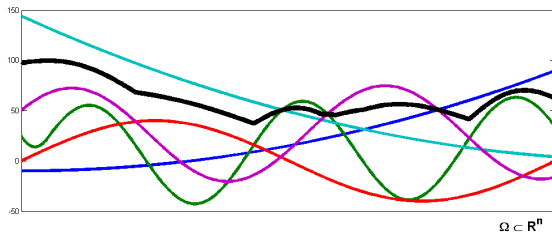
$$\min_{x \in \Omega} \frac{1}{p} \sum_{j=1}^p f_{i_j(x)}(x)$$



In this example,  $p = 2$

$$J = \{p + 1, \dots, m\}, \text{ (CVaR-like)}$$

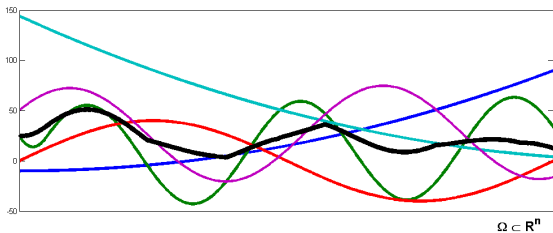
$$\min_{x \in \Omega} \frac{1}{m-p} \sum_{j=p+1}^m f_{j_j(x)}(x)$$



In this example,  $p = 3$

$$J = \{q + 1, \dots, p\}$$

$$\min_{x \in \Omega} \frac{1}{p - q} \sum_{j=q+1}^p f_{ij(x)}(x)$$



In this example,  $q = 1, p = 4$

# Reformulation of CVaR-like

Fact

$$b_1 \leq b_2 \leq \dots \leq b_m, \quad p \leq m - 1$$

$$\Rightarrow b_{p+1} + \dots + b_m = \underset{\xi \in \mathbb{R}}{\text{Minimum}} \quad (m - p)\xi + \sum_{i=1}^m \max\{0, b_i - \xi\}$$

$$\text{Minimizers} = \{\xi \in [b_p, b_{p+1}]\}$$

↓

$$\underset{x \in \Omega}{\text{Minimize}} \quad f_{i_{p+1}(x)}(x) + \dots + f_{i_m(x)}(x)$$

is equivalent to:

$$\underset{x \in \Omega, \xi \in \mathbb{R}}{\text{Minimize}} \quad (m - p)\xi + \sum_{i=1}^m \max\{0, f_i(x) - \xi\}$$

# Reformulation of VaR-like

From the same “fact”

$$\text{Minimize}_{x \in \Omega} f_{i_p}(x)$$

is equivalent to:

Minimize  $\xi$  (with respect to  $x \in \Omega$  and  $\xi \in \mathbb{R}$ )

subject to

$$\xi \text{ minimizes } (m - p)\xi + \sum_{i=1}^m \max\{0, f_i(x) - \xi\} \quad (\text{wrt } \xi)$$

# Consequence for the Reformulations of CVaR-like and VaR-like

CVaR-like is a Nonlinear Programming problem  
Convex if the  $f_i$  are convex  
Linear-Programming if the  $f_i$  are linear  
with many inequality constraints

VaR-like is a Bilevel Programming problem  
with many Complementarity Constraints  
that come from the KKT conditions of the Lower-Level problem

# Primal method for minimizing CVaR-like

Consider

$$\text{Minimize}_{x \in \Omega, \xi \in \mathbb{R}} (m - \rho)\xi + \sum_{i=1}^m \max\{0, f_i(x) - \xi\}$$

Use smoothing to deal with of max and ordinary NLP for minimizing on  $\Omega \times \mathbb{R}$ .



# Primal Method for Minimizing VaR-like

Given the current point  $x^k \in \Omega$  (convex) take a sufficient descent direction  $d^k$  for all  $j$  such that

$$f_{i_p(x^k)}(x^k) - \epsilon \leq f_j(x^k) \leq f_{i_p(x^k)}(x^k) + \epsilon$$

Line-search along  $d^k \rightarrow x^{k+1} = x^k + \alpha_k d^k$

Global convergence to ( $\epsilon$ )stationary points

Local

Superlinear

Quadratic

Convex subproblems (linear or quadratic constraints)

# Risk Minimization

- $m$  scenarios
- $f_i(x)$  = predicted loss caused by decision  $x \in \Omega$  under scenario  $i$
- $f_{i_p(x)}(x)$  = VaR associated with  $x$
- $\frac{1}{m-p} \sum_{j=p+1}^m f_{j(x)}(x)$  = CVaR associated with  $x$
- $\underset{x \in \Omega}{\text{minimize}} f_{i_p(x)}(x) \equiv \text{minimize VaR}$
- $\underset{x \in \Omega}{\text{minimize}} \sum_{j=p+1}^m f_{j(x)}(x) \equiv \text{minimize CVaR}$

# Low Order-Value Optimization (LOVO)

Define, as always,  $i_1(x), \dots, i_m(x)$  by:

$$f_{i_1(x)}(x) \leq \dots \leq f_{i_m(x)}(x); \quad p \leq m$$

Then, the LOVO problem is:

$$\underset{x \in \Omega}{\text{minimize}} \quad \sum_{j=1}^p f_{i_j(x)}(x)$$

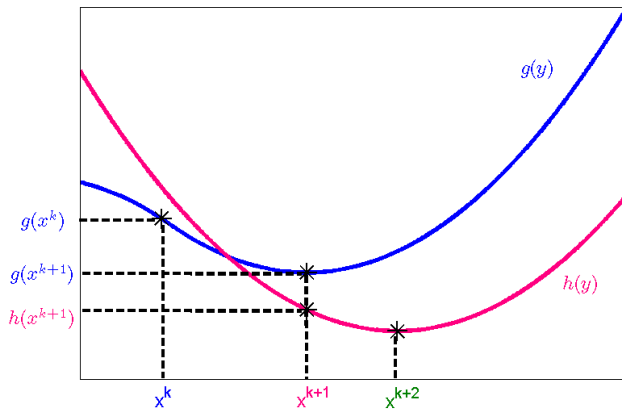
## Fact

$$\sum_{j=1}^p f_{i_j(x)}(y) \leq \sum_{j=1}^p f_{i_j(x)}(x) \Rightarrow \sum_{j=1}^p f_{i_j(y)}(y) \leq \sum_{j=1}^p f_{i_j(x)}(x)$$

$$\Downarrow$$

In order to decrease the LOVO function we may “fix”

$(i_1(x), \dots, i_p(x))$  and “minimize”  $\sum_{j=1}^p f_{i_j(x)}(y)$  with respect to  $y$ .



$$g(y) = \sum_{j=1}^p f_{ij}(x^k)(y)$$

$$h(y) = \sum_{j=1}^p f_{ij}(x^{k+1})(y)$$

# Methods for Unconstrained LOVO problems

## Line-Search

At iteration  $k$

Find a sufficient descent direction for  $\sum_{j=1}^p f_{i_j(x^k)}(x)$

take

$$\sum_{j=1}^p f_{i_j(x^k + \alpha_k d)}(x^k + \alpha_k d) <_{\text{suf}} \sum_{j=1}^p f_{i_j(x^k)}(x^k)$$

Global Convergence to points  $x^*$  such that

$$\nabla \sum_{j=1}^p f_{i_j(x^*)}(x^*) = 0$$

# Trust-Region methods for LOVO

Typical iteration

Given  $x^k$ , the trust region defined by  $\Delta$  and a quadratic

approximation of  $\sum_{j=1}^p f_{ij}(x^k)(x)$ :

Minimize the quadratic approximation on the trust region  $\Delta$

- If the reduction of  $\sum_{j=1}^p f_{ij}(x^k)(x)$  is sufficiently large with respect to the reduction of the quadratic approximation (Ared  $\geq 0.1$  Pred) accept the solution of the trust region subproblem as  $x^{k+1}$ .
- Otherwise, reduce  $\Delta$ .

# Convergence of trust-region methods for LOVO

At every limit point  $x^*$ ,

$$\nabla \sum_{j=1}^p f_{i_j(x^*)}(x^*) = 0.$$

Using the true Hessian to define the quadratic approximation:

$$\nabla^2 \sum_{j=1}^p f_{i_j(x^*)}(x^*) \geq 0$$

Local convergence: quadratic



# Fitting with LOVO

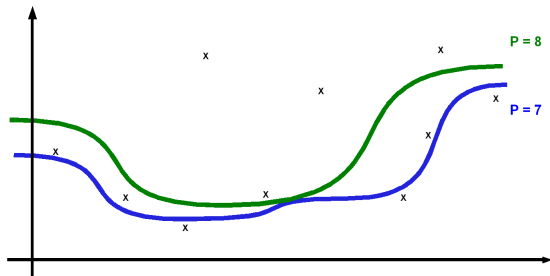
Observations

$t$	$y$
$t_1$	$y_1$
$\vdots$	$\vdots$
$t_m$	$y_m$

Model

$$y_j \approx M(x, t_j)$$

$$f_j(x) = [y_j - M(x, t_j)]^2$$



## Constrained LOVO problems

$$\text{Minimize } \sum_{j=1}^p f_{ij(x)}(x)$$

subject to

$$h(x) = 0, g(x) \leq 0.$$

Augmented Lagrangian (PHR-Like) (Code Algencan in  
[www.ime.usp.br/~egbirgin/tango](http://www.ime.usp.br/~egbirgin/tango))

$$\text{Minimize approx } \sum_{j=1}^p f_{ij(x)}(x) + \frac{\rho}{2} \left[ \left\| h(x) + \frac{\lambda}{\rho} \right\|^2 + \left\| \left( g(x) + \frac{\mu}{\rho} \right)_+ \right\|^2 \right]$$

Update  $\lambda, \mu \geq 0, \rho$ . ( $a_+ = \max\{0, a\}$ )

# Convergence of Algencan-LOVO

Global minimization of subproblems



Global Minimization

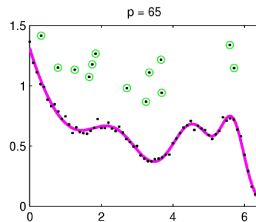
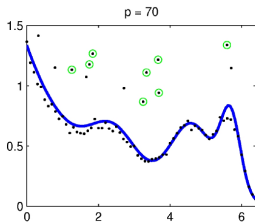
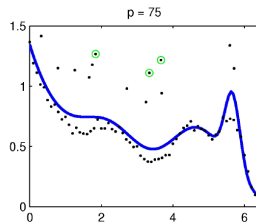
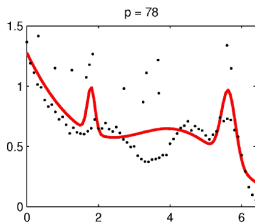
Limit points are either feasible or stationary points of Infeasibility

Feasible limit points that satisfy the CPLD constraint qualification are “KKT”

Boundedness of penalty parameter

# Model fitting with Algencon-LOVO

Find the parameters of a Boundary Value Problem fitting a set of data that contains outliers



## Fitting Nash-Equilibrium Models

### Nash-Equilibrium Model:

Given the parameters  $x \in \Omega$ , the players  $1, 2, \dots, m$  take, simultaneously, decisions  $y_1, \dots, y_m$ .

Player  $j$  takes his/her decision minimizing

$f_j(x, y_1, \dots, y_{j-1}, z, y_{j+1}, \dots, y_m)$  with respect to  $z$ .

## Fitting Nash-Equilibrium Models

### Nash-Equilibrium Model:

Given the parameters  $x \in \Omega$ , the players  $1, 2, \dots, m$  take, simultaneously, decisions  $y_1, \dots, y_m$ .

Player  $j$  takes his/her decision minimizing

$f_j(x, y_1, \dots, y_{j-1}, z, y_{j+1}, \dots, y_m)$  with respect to  $z$ .

### Inverse Nash-Equilibrium:

$\bar{y}_1, \dots, \bar{y}_m$  are known

Discover the parameters  $x$ .

## Fitting Nash-Equilibrium Models

### Nash-Equilibrium Model:

Given the parameters  $x \in \Omega$ , the players  $1, 2, \dots, m$  take, simultaneously, decisions  $y_1, \dots, y_m$ .

Player  $j$  takes his/her decision minimizing

$f_j(x, y_1, \dots, y_{j-1}, z, y_{j+1}, \dots, y_m)$  with respect to  $z$ .

### Inverse Nash-Equilibrium:

$\bar{y}_1, \dots, \bar{y}_m$  are known

Discover the parameters  $x$ .

### LOVO-Inverse-Nash-Equilibrium:

$\bar{y}_1, \dots, \bar{y}_m$  are known but only 90% of these observations are reliable.

# Fitting Nash-Equilibrium Models

$$\text{Minimize } \sum_{j=1}^p (y_{ij}(x,y) - \bar{y}_{ij}(x,y))^2$$

subject to

Player  $j$  minimizes  $f_j(x, y_1, \dots, y_{j-1}, z, y_{j+1}, \dots, y_m)$

with respect to  $z$ , for all  $j = 1, \dots, m$



## Dealing with LOVO constraints

Minimize  $f(x)$

s/t  $x$  satisfies at least  $p$  constraints of the subset  $\begin{cases} g_1(x) \leq 0 \\ \vdots \\ g_m(x) \leq 0 \end{cases}$

We define  $L(x, \mu, \rho) = f(x) + \rho \sum_{j=1}^p \left( g_{i_j(x)}(x) + \frac{\mu_{i_j}}{\rho} \right)_+^2$

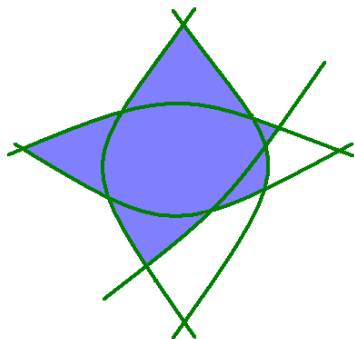
where  $\left( g_{i_1(x)}(x) + \frac{\mu_{i_1}}{\rho} \right)_+^2 \leq \dots \leq \left( g_{i_m(x)}(x) + \frac{\mu_{i_m}}{\rho} \right)_+^2$

minimize  $L(x, \mu, \rho)$

update  $\mu, \rho$

# LOVO constraints

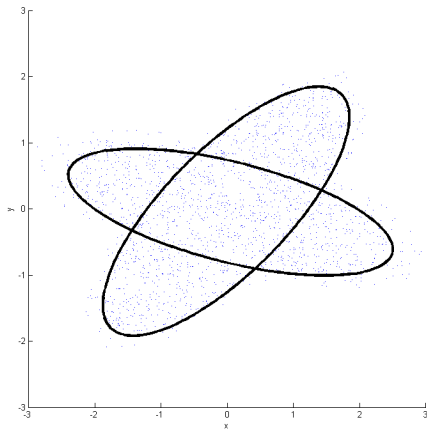
Example of Feasible Region for  $p = m - 1$



## Example of Minimization with LOVO constraints

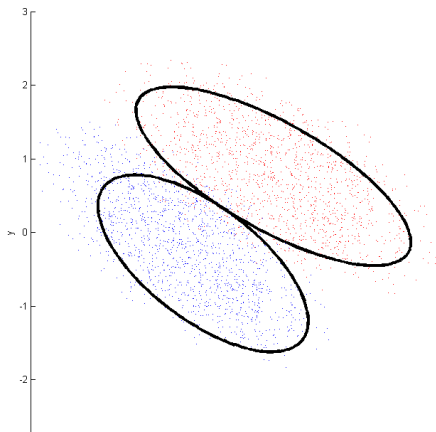
Find the union of 2 ellipses with smallest area that contain 90% of the points  $\{P_1, \dots, P_m\}$

We have one constraint  $P_i \in E_1 \cup E_2$  for each point  $P_i$ .



## Example of Minimization with LOVO constraints

Given two sets of points A and B, find two ellipses with no intersection area, that contain, respectively, 80% of the points of A and 80% of the points of B.



## Topics in computer vision

Curve detection {  
detection of lines  
detection of circles  
detection of ellipses  
others

[Comparisons with HT, Ransac, QMDPE, LKS in  
Cesar-Andreani-Marcondes-JMM & Silva 2007]

# Lines

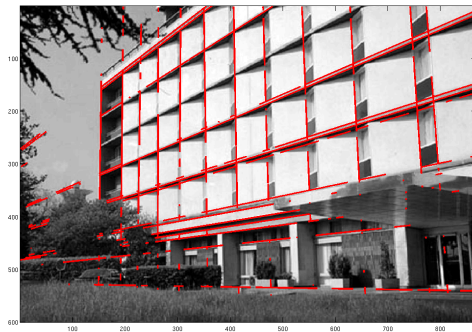
Given  $\{(x_1, y_1), \dots, (x_m, y_m)\}$  find  $\theta$  and  $\rho$  such that

$$\sum_{j=1}^p f_{i_j(\theta, \rho)}(\theta, \rho) \text{ is minimal,}$$

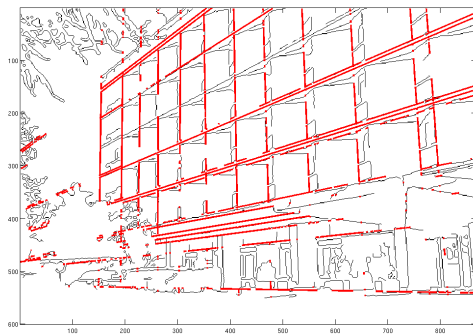
where  $f_i(\theta, \rho) = [x_i \cos \theta + y_i \sin \theta - \rho]^2$   $i = 1, \dots, m$ ,

and  $f_{i_1(\theta, \rho)}(\theta, \rho) \leq \dots \leq f_{i_m(\theta, \rho)}(\theta, \rho) \quad \forall \theta \text{ and } \rho > 0$

# Detection of lines

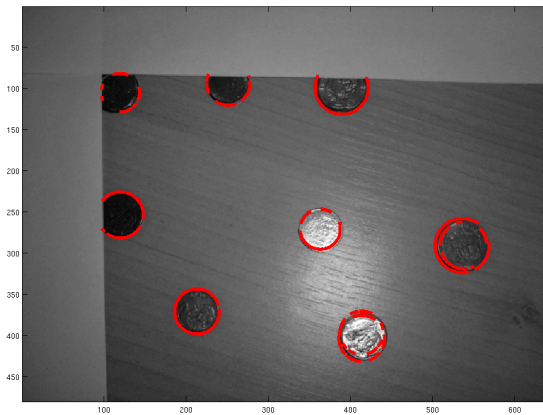


# Detection of lines

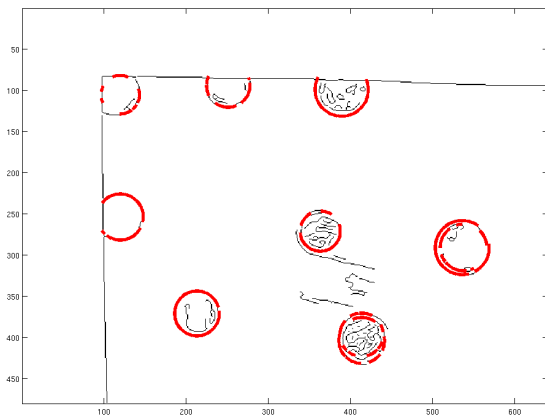




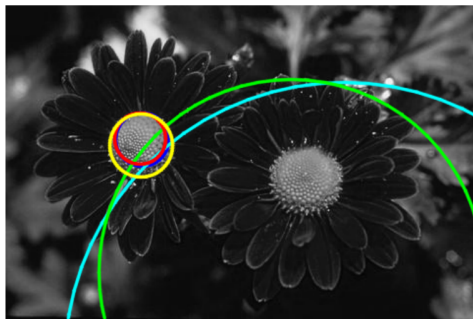
# Detection of circles



# Detection of circles



# Detection of circles



- LOVO
- Hough Transform
- Least k-square
- Ransac
- QMDPE

## Current work

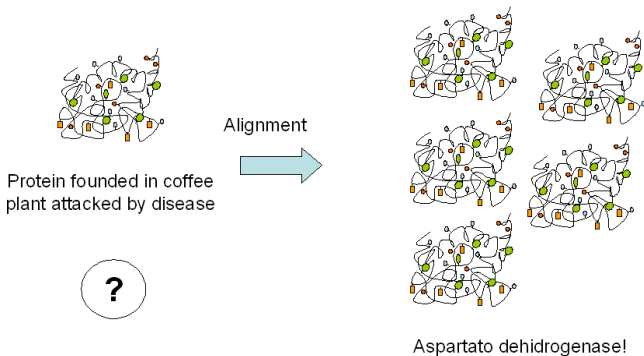
Tracking Parametric Curves in video sequences

The tracked curve in frame  $t-1$  should be used as initial solution for the LOVO problem to be solved in frame  $t$ .

# Protein Alignment

Finding common 3D structures of two given proteins.  
Evaluating similarity.

Identification of the function of new proteins



# Protein Alignment

Data: 3D coordinates of  $C - \alpha$  atoms of both proteins.

(Available, for example, in Protein Data Bank PDB)

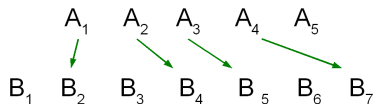
Protein A:  $(A_1, \dots, A_{n_A})$

Protein B:  $(B_1, \dots, B_{n_B})$

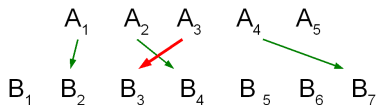
$A_i, B_j \in \mathbb{R}^3$

To each **admissible bijection** between (a subset of)  $A$  and (a subset of)  $B$  corresponds a **score**.

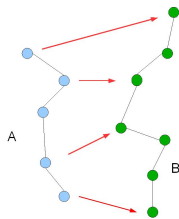
Example admissible bijection  $\Phi$



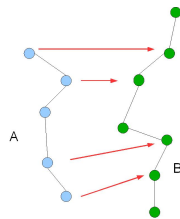
Non-admissible bijection



Given fixed relative positions of A and B, the bijection with best score may be obtained using dynamic programming



Score = 1.31



Score = 13.42

Changing the relative positions, both “best bijection” and score change.



$-f_i(x)$  = score associated with  
“bijection  $i$ ” under the  
movement defined by  $x$ .

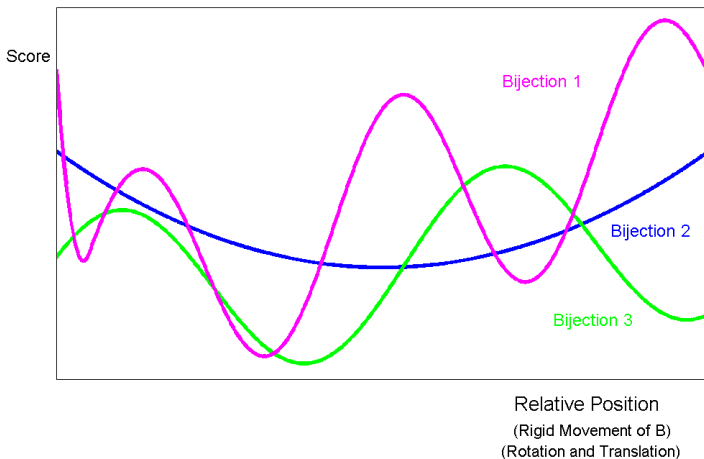


Minimize  $f_{i_1(x)}(x)$  (LOVO,  $p = 1$ )

Objective of the alignment:

maximize the score, with respect to bijection and relative position.

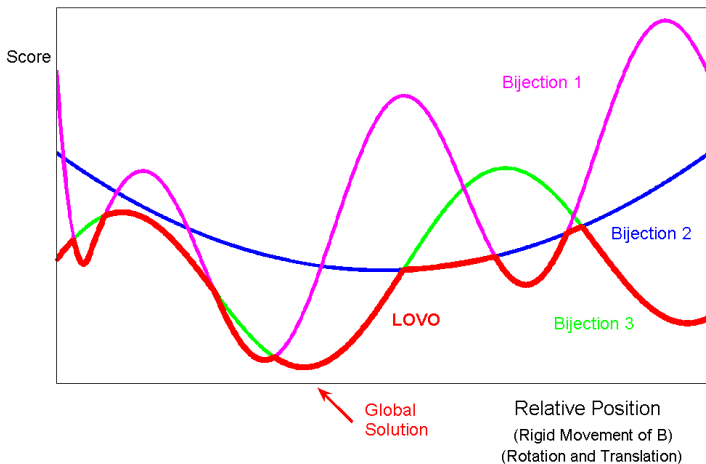
Modelling as LOVO:



Objective of the alignment:

maximize the score, with respect to bijection and relative position.

Modelling as LOVO:




# LOVOALIGN PACKAGE

[www.ime.unicamp.br/~martinez/lovoalign](http://www.ime.unicamp.br/~martinez/lovoalign)

On-line alignment  
of proteins

# www.ime.unicamp.br/~martinez/lovoalign

**LovoAlign** 

**Home**

**Server**

**Download**

**Userguide**

**References**

**Contact/Help**

**Home**

LovoAlign is a new protein structural alignment package. The methods used for structural alignment are based on Low Order Value Optimization (LOVO) theory. The use of LOVO theory led to the development of fast convergent algorithms that provide very robust optimization of scoring functions.

Numerical experiments show that the LOVO algorithms implemented here provide the most reliable optimization of the STRUCTAL alignment while being very fast.

Simple input parameters can be used to align two structures, a single structure to a whole database, or to perform an all-on-all database structural alignment.

The current version of the LovoAlign software can be downloaded, with source codes, at no cost.

Concluído Internet 15:04

# www.ime.unicamp.br/~martinez/lovoalign

The screenshot shows a web browser window titled "LOVOALIGN: Protein Structural Alignment - Microsoft Internet Explorer". The address bar displays the URL "http://www.ime.unicamp.br/~martinez/lovoalign/align/index.html". The page content includes a navigation menu on the left with buttons for Home, Server, Download, Userguide, References, and Contact/Help. The main content area is titled "Online server" and contains the following text:

The first protein uploaded will be aligned to the second protein. The method used is the Newton method with dynamic programming, which provides the best alignments. No information is saved in this site.

Below the text are two input fields for "First protein:" and "Second protein:", each with a "Procurar..." button and a "Chain:" dropdown menu. An "Align" button is positioned to the right of the second protein's dropdown.

A "Notes" section follows, containing three numbered points:

1. If there is more than one structure in the PDB file, separated by the 'END' or 'ENDMDL' keywords, only the first structure will be considered.
2. If there is more than one chain in the PDB file, no chain is specified, and the chains are not separated by a 'END' keyword, they will be treated as a single molecule and will be considered for the alignment.
3. The output contains all atoms of the original pdb file rotated and translated according to the best alignment obtained for the chain or molecule considered.

The browser's status bar at the bottom shows "Concluido" and "Internet".

# Conclusion

We presented:

- The general form of Order-Value functions and OVO problems
- Particular cases: VaR-like, CVaR-like, LOVO, ...
- Discussion of nonsmoothness and many local minimizers
- Nonlinear Programming Reformulations
- Primal (trust-region and line-search) methods for unconstrained OVO problems with and without smoothing
- Using LOVO for Nonlinear Regression with Outliers
- Constrained LOVO problems
- Convergence of Algorithms for Constrained LOVO
- Application to Nash-Equilibrium fitting
- LOVO constraints
- Computer Vision
- Protein Alignment

## Some references

- 1 R. Andreani, J. M. Martínez, L. Martínez and F. Yano. Continuous Optimization Methods for Structure Alignments. To appear in *Mathematical Programming*.
- 2 R. Andreani, J. M. Martínez, M. Salvatierra and F. Yano. Quasi-Newton methods for order-value optimization and value-at-risk calculations. *Pacific Journal of Optimization* 2, pp. 11-33 (2006).
- 3 R. Andreani, C. Dunder and J. M. Martínez. Nonlinear-Programming Reformulation of the Order-Value Optimization problem. *Mathematical Methods of Operations Research* 61, pp. 365-384 (2005).
- 4 R. Andreani, C. Dunder and J. M. Martínez. Order-Value Optimization: formulation and solution by means of a primal Cauchy method. *Mathematical Methods of Operations Research* 58, pp. 387-399 (2003).