Universidade do Minho

A particle swarm pattern search method for bound constrained nonlinear optimization

A. Ismael F. Vaz

Departamento de Produção e Sistemas

Escola de Engenharia, Universidade do Minho

aivaz@dps.uminho.pt

Luís Nunes Vicente

Departamento de Matemática,

Universidade de Coimbra

lnv@mat.uc.pt

Work partially financed by FCT (grants POCI/MAT/58957/2004, POCI/MAT/59442/2004)

Universidade do Minho

Outline

- Introduction.
- Particle Swarm paradigm.
- Pattern search.
- Hybrid algorithm (`PSwarm`).
- Convergence results.
- Numerical results and conclusions.

Outline

- Introduction.
- Particle Swarm paradigm.
- Pattern search.
- Hybrid algorithm (`PSwarm`).
- Convergence results.
- Numerical results and conclusions.

Outline

- Introduction.
- Particle Swarm paradigm.
- Pattern search.
- Hybrid algorithm (`PSwarm`).
- Convergence results.
- Numerical results and conclusions.

Universidade do Minho

Outline

❖ Outline

Outline

- Introduction.
- Particle Swarm paradigm.
- Pattern search.
- Hybrid algorithm (`PSwarm`).
- Convergence results.
- Numerical results and conclusions.

Universidade do Minho

Outline

- Introduction.
- Particle Swarm paradigm.
- Pattern search.
- Hybrid algorithm (`PSwarm`).
- Convergence results.
- Numerical results and conclusions.

Outline

- Introduction.
- Particle Swarm paradigm.
- Pattern search.
- Hybrid algorithm (`PSwarm`).
- Convergence results.
- Numerical results and conclusions.

Universidade do Minho

Introduction

Problem formulation

Universidade do Minho

We are addressing the problem in the following mathematical form

$$\min_{z \in \mathbb{R}^n} \; f(z)$$
$$\text{s.t.} \quad \ell \leq z \leq u,$$

where $\ell \leq z \leq u$ are to be understood has componentwise inequalities.

To apply the particle swarm paradigm or the pattern search smoothness of the objective function $f(z)$ is not requested.

For the theoretical results of the pattern search and therefore of the hybrid algorithm some smoothness of the objective function $f(z)$ is imposed.

Universidade do Minho

Particle Swarm

Universidade do Minho

Particle Swarm paradigm (PS)

Population based algorithms tries to mimic the social behavior of a population (swarm) of individuals (particles).

An individual behavior is a combination of its past experience (cognitive influence) and from the society experience (social influence).

In the optimization context, one particle p, at time instance t, is represented by its current position ($x^p(t)$), its best ever position ($y^p(t)$) and a *traveling* velocity ($v^p(t)$).

Let $\hat{y}(t)$ represent the best particle position of the population.

New position and velocity

The new particle position is updated by

$$x^p(t+1) = x^p(t) + v^p(t+1),$$

where $v^p(t+1)$ is the new velocity given by

$$v_j^p(t+1) = \iota(t)v_j^p(t) + \mu\omega_{1j}(t)\left(y_j^p(t) - x_j^p(t)\right) + \nu\omega_{2j}(t)\left(\hat{y}_j(t) - x_j^p(t)\right),$$

for $j = 1, \ldots, n$.

- $\iota(t)$ is the inertial factor
- μ is the *cognitive* parameter and ν is the *social* parameter
- $\omega_{1j}(t)$ and $\omega_{2j}(t)$ are random numbers drawn from the uniform $(0, 1)$ distribution.

New position and velocity

The new particle position is updated by

$$x^p(t+1) = x^p(t) + v^p(t+1),$$

where $v^p(t+1)$ is the new velocity given by

$$v_j^p(t+1) = \iota(t)v_j^p(t) + \mu\omega_{1j}(t)\left(y_j^p(t) - x_j^p(t)\right) + \nu\omega_{2j}(t)\left(\hat{y}_j(t) - x_j^p(t)\right),$$

for $j = 1, \ldots, n$.

- $\iota(t)$ is the inertial factor
- μ is the *cognitive* parameter and ν is the *social* parameter
- $\omega_{1j}(t)$ and $\omega_{2j}(t)$ are random numbers drawn from the uniform $(0,1)$ distribution.

Universidade do Minho

New position and velocity

The new particle position is updated by

$$x^p(t+1) = x^p(t) + v^p(t+1),$$

where $v^p(t+1)$ is the new velocity given by

$$v_j^p(t+1) = \iota(t)v_j^p(t) + \mu\omega_{1j}(t)\left(y_j^p(t) - x_j^p(t)\right) + \nu\omega_{2j}(t)\left(\hat{y}_j(t) - x_j^p(t)\right),$$

for $j = 1, \ldots, n$.

- $\iota(t)$ is the inertial factor
- μ is the *cognitive* parameter and ν is the *social* parameter
- $\omega_{1j}(t)$ and $\omega_{2j}(t)$ are random numbers drawn from the uniform $(0,1)$ distribution.

New position and velocity

Universidade do Minho

The new particle position is updated by

$$x^p(t+1) = x^p(t) + v^p(t+1),$$

where $v^p(t+1)$ is the new velocity given by

$$v_j^p(t+1) = \iota(t)v_j^p(t) + \mu\omega_{1j}(t)\left(y_j^p(t) - x_j^p(t)\right) + \nu\omega_{2j}(t)\left(\hat{y}_j(t) - x_j^p(t)\right),$$

for $j = 1, \ldots, n$.

- $\iota(t)$ is the inertial factor
- μ is the *cognitive* parameter and ν is the *social* parameter
- $\omega_{1j}(t)$ and $\omega_{2j}(t)$ are random numbers drawn from the uniform $(0,1)$ distribution.

Handling constraints

Simple bound constraints are handled by a projection onto $\Omega = \{x \in \mathbb{R}^n : \ \ell \leq x \leq u\}$, for all particles $i = 1, \ldots, s$.

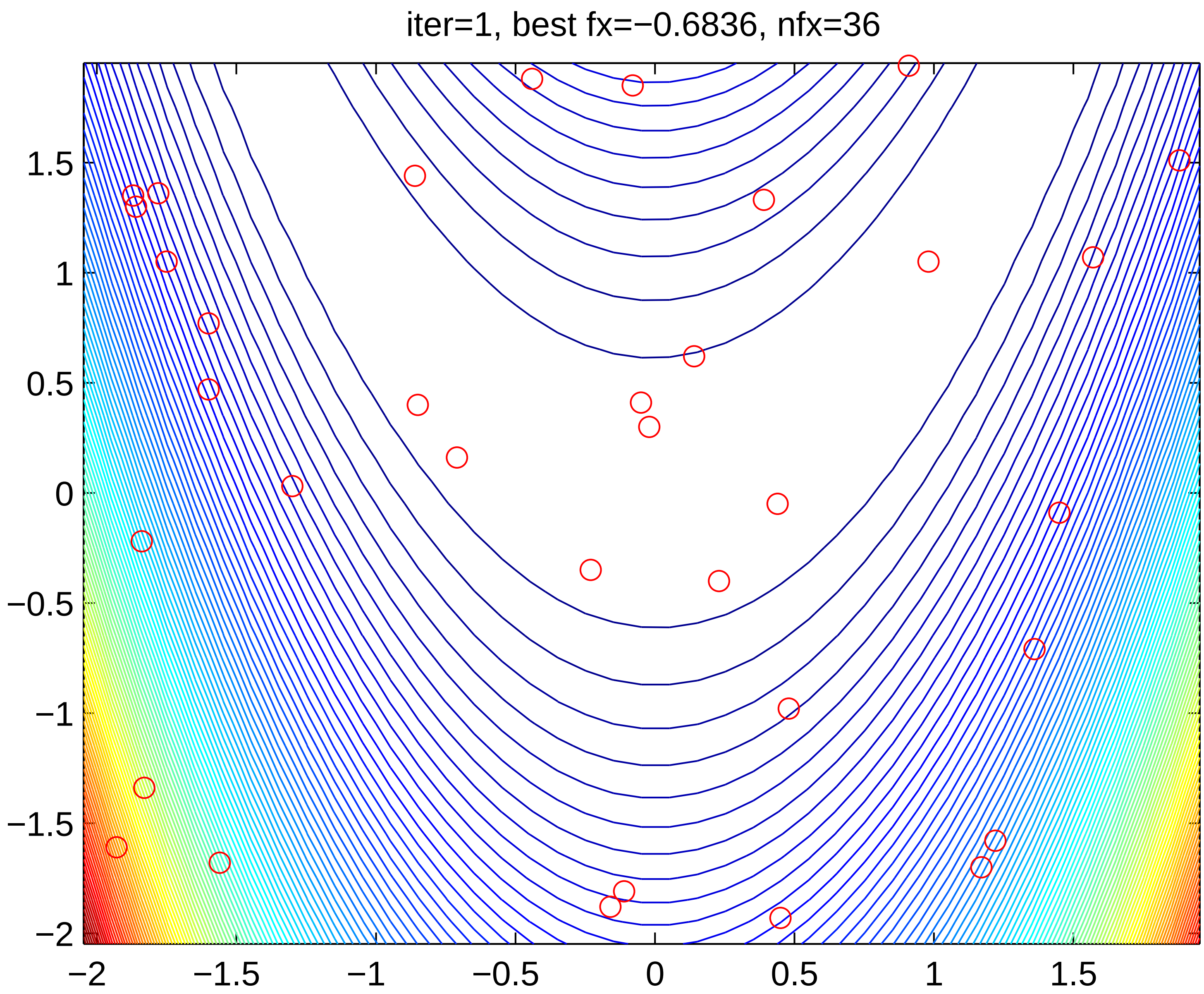
$$proj_\Omega(x_j^i(t)) = \begin{cases} \ell_j & \text{if } \ x_j^i(t) < \ell_j, \\ u_j & \text{if } \ x_j^i(t) > u_j, \\ x_j^i(t) & \text{otherwise,} \end{cases}$$

for $j = 1, \ldots, n$.

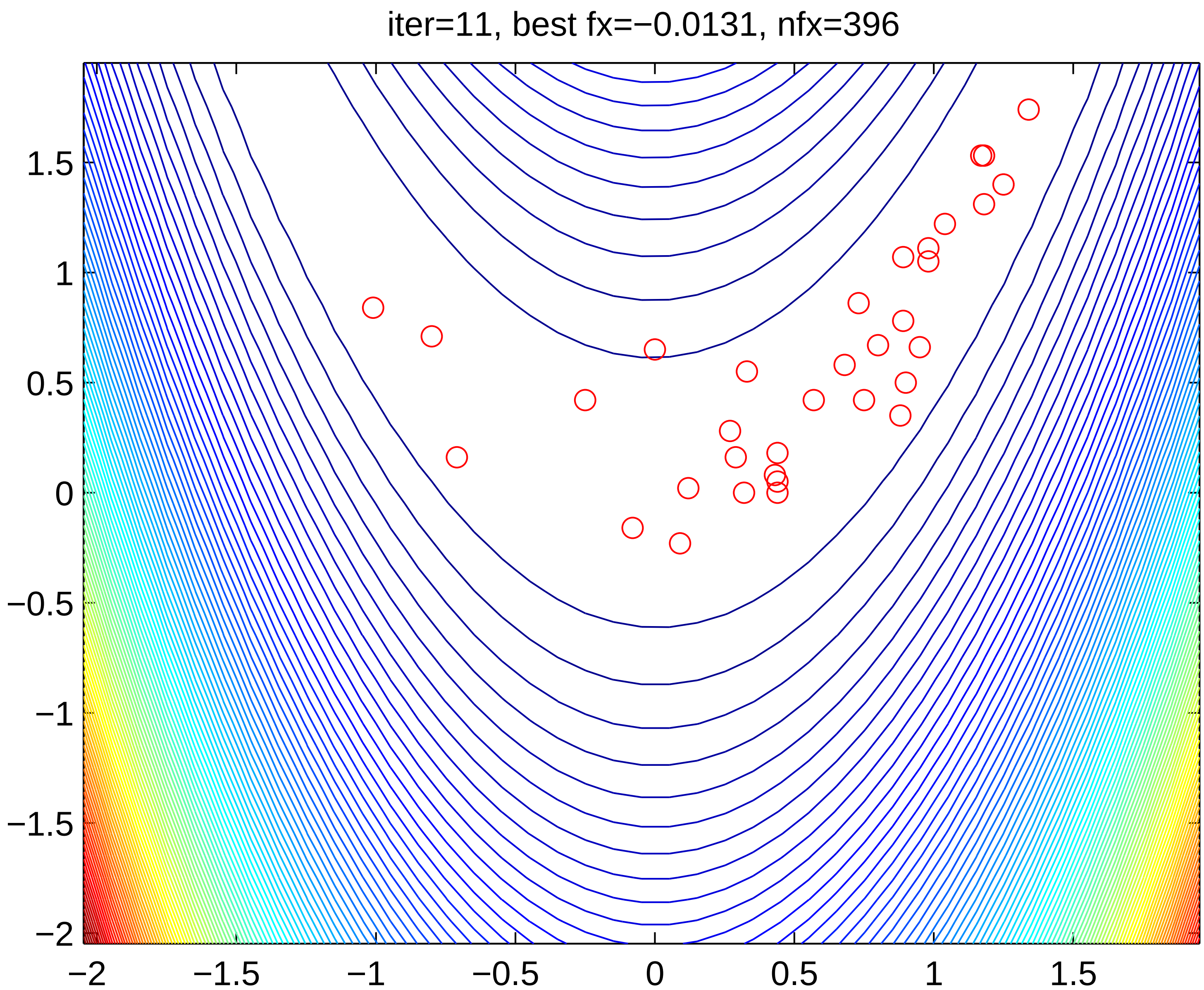
The projection is applied to the new particles position.

Example with `ir2.mod` *problem*

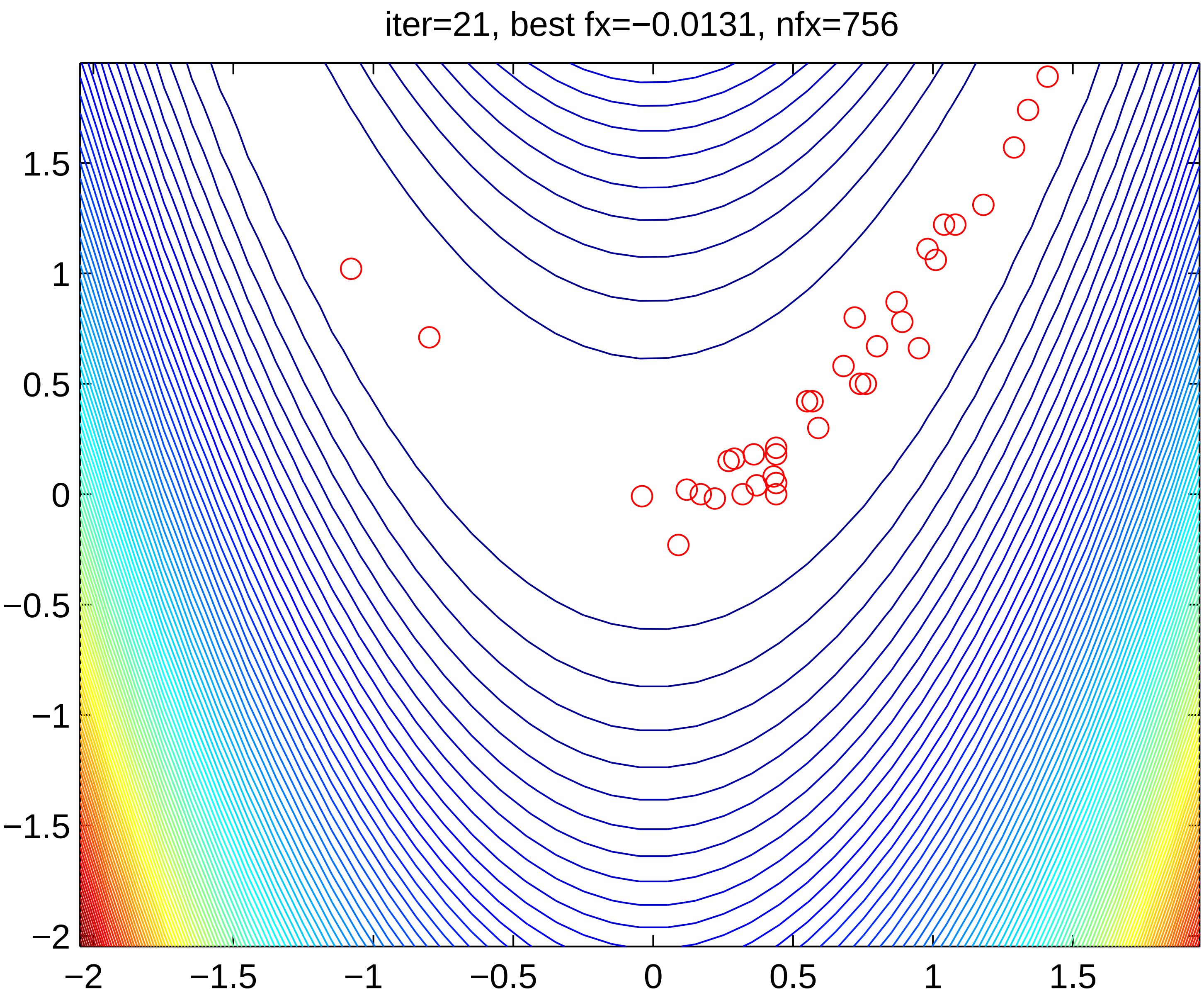
Universidade do Minho



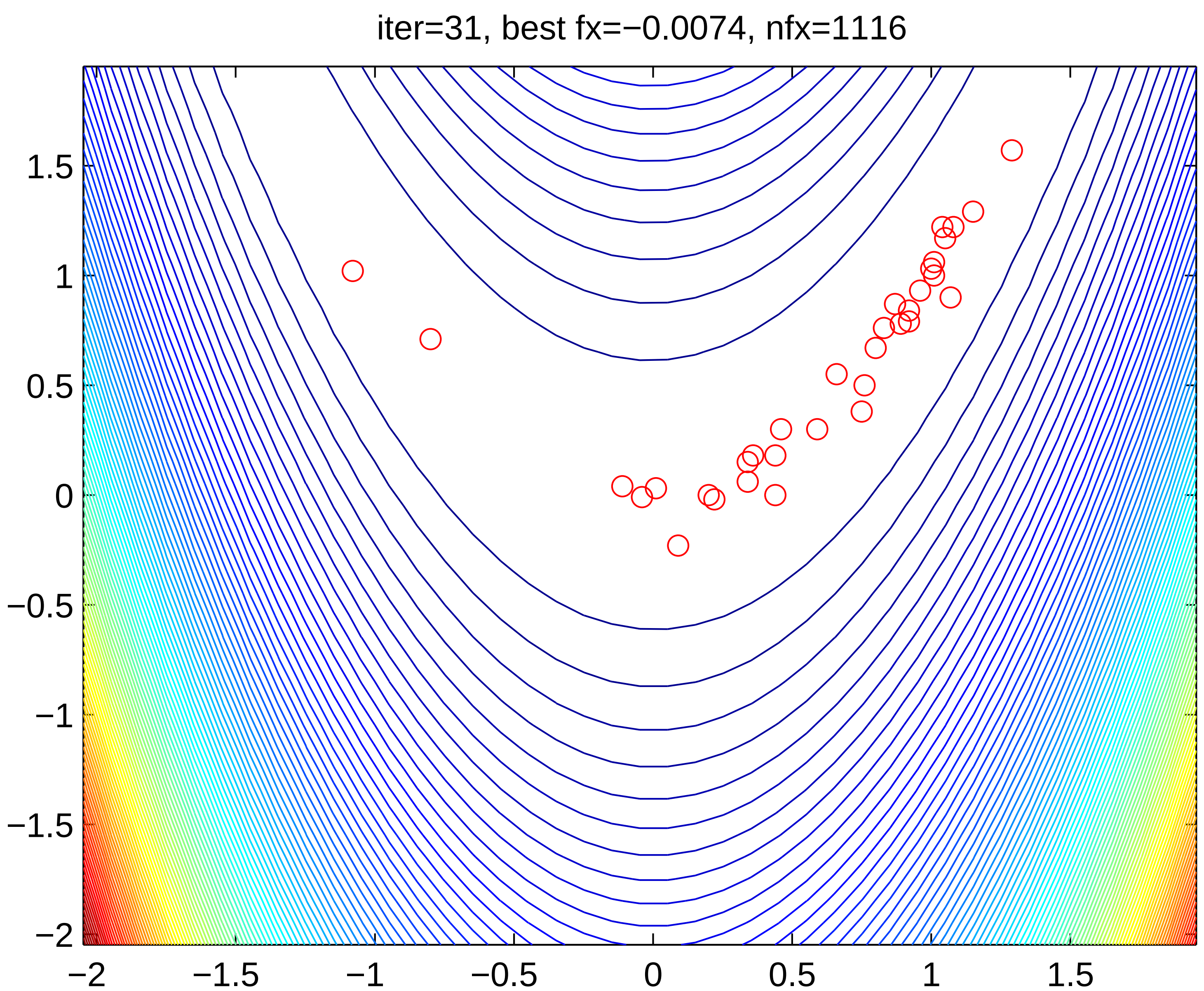
Example



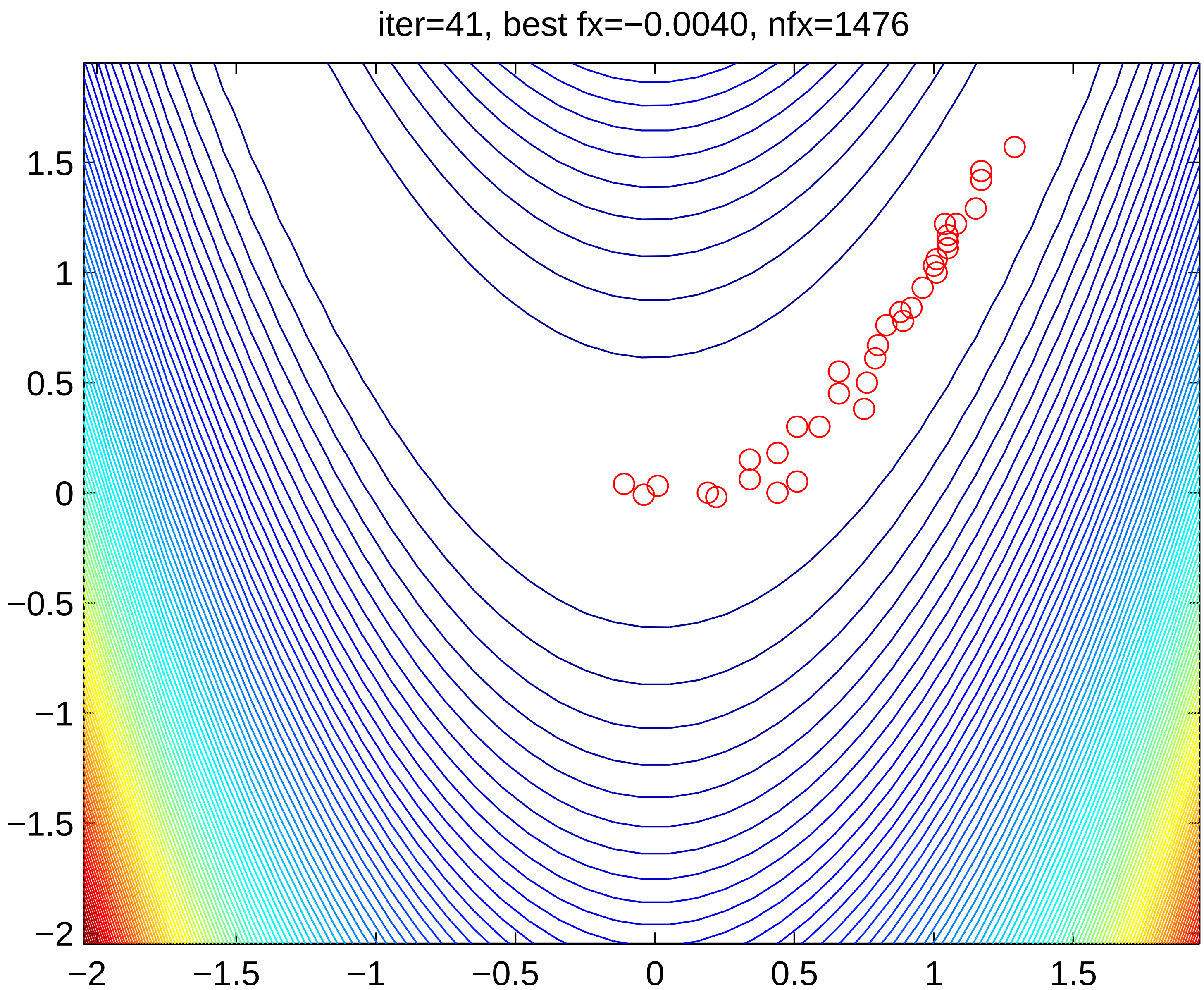
Example



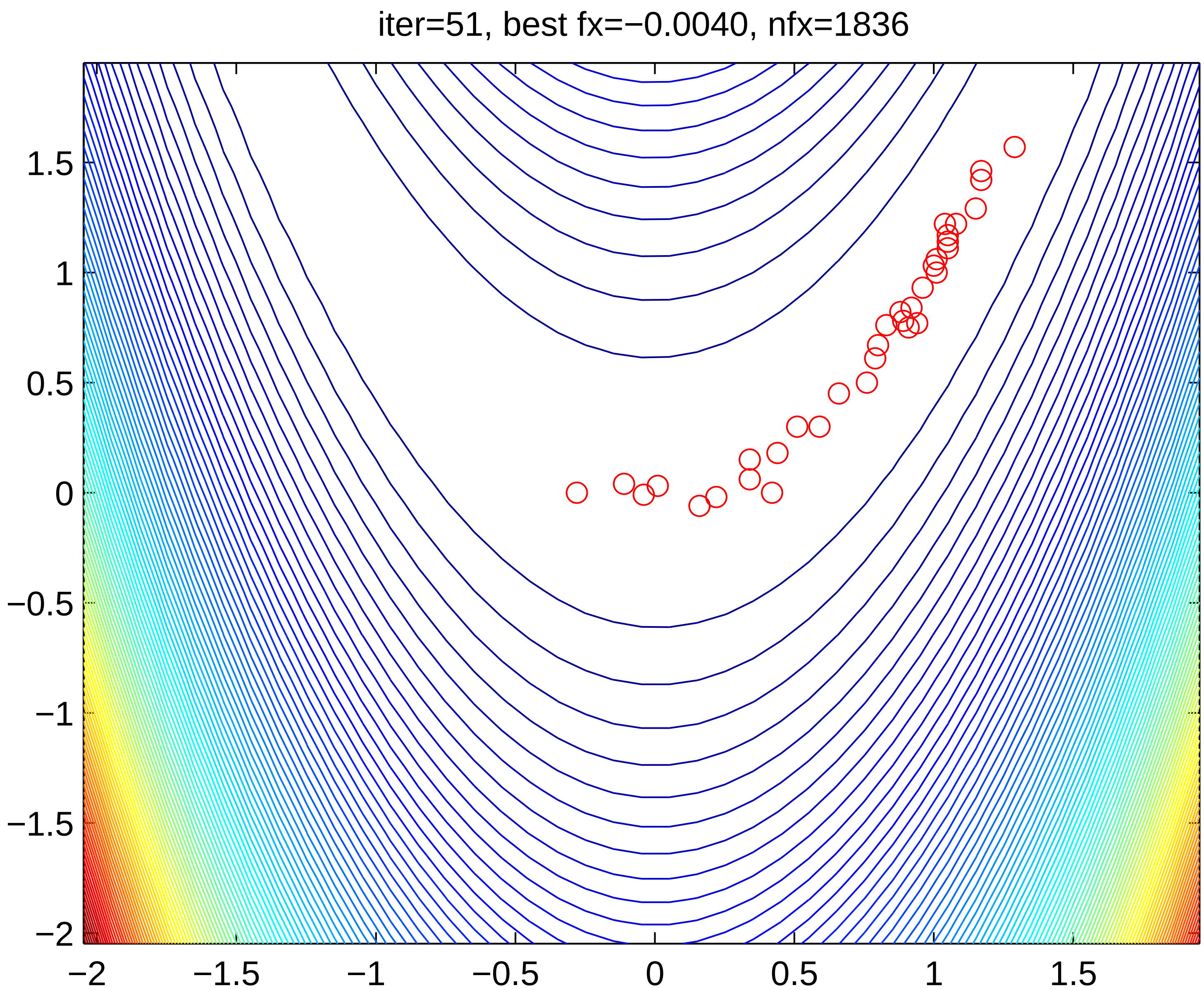
Example



Example



Example



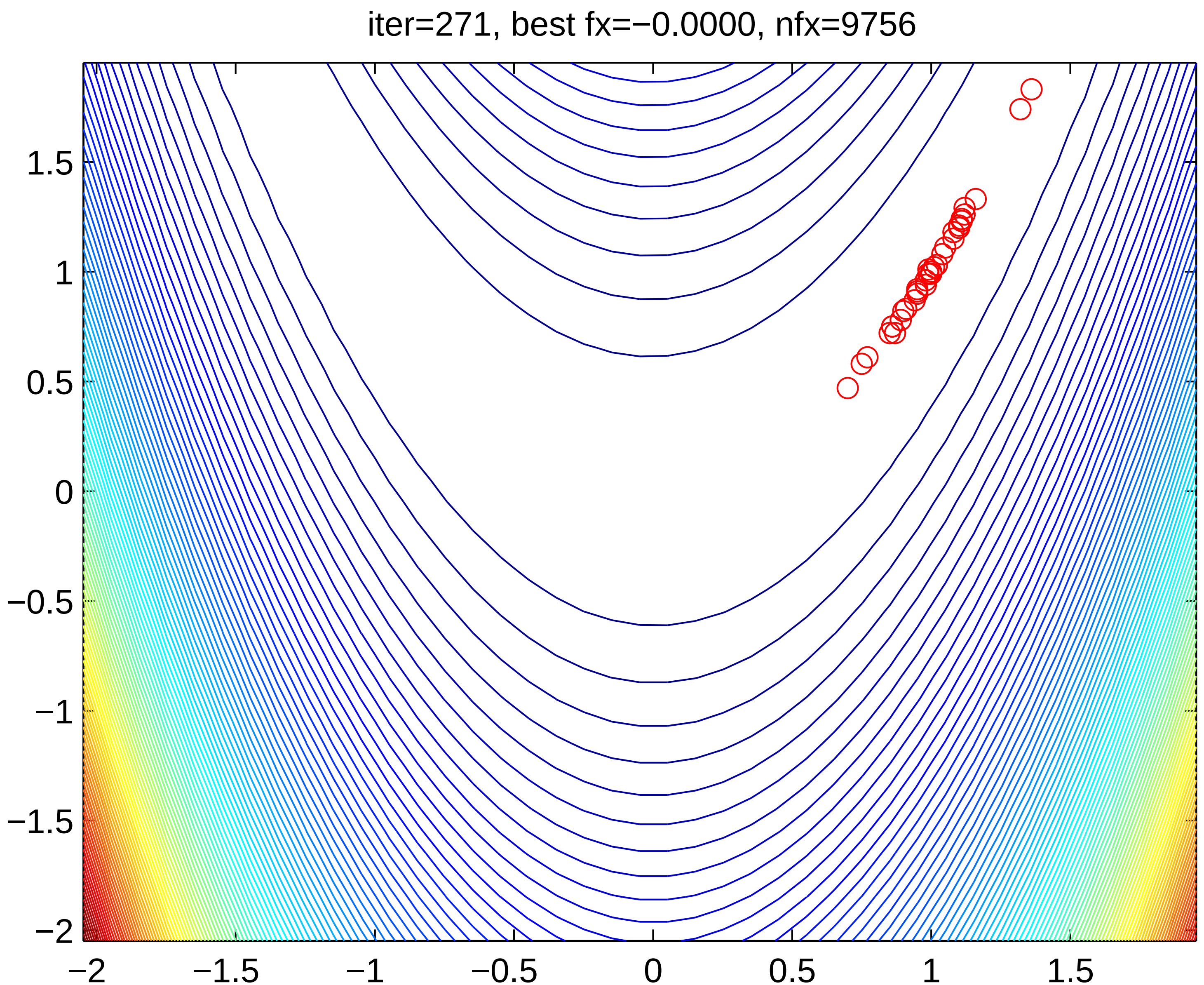
Universidade do Minho

Particle Swarm

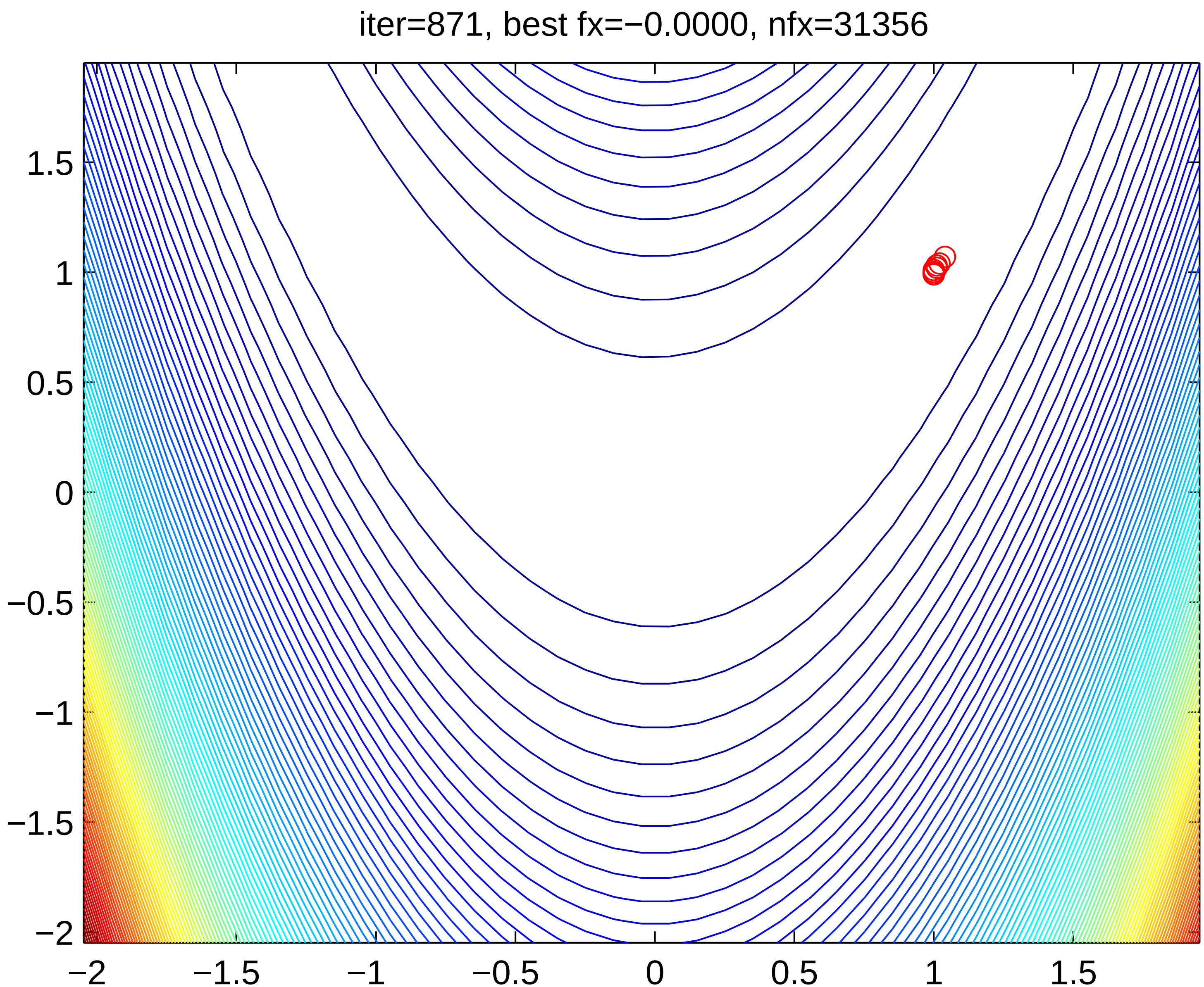
- Particle Swarm paradigm (PS)
- New position and velocity
- Constraints
- Example
- Properties

University do Minho

Example

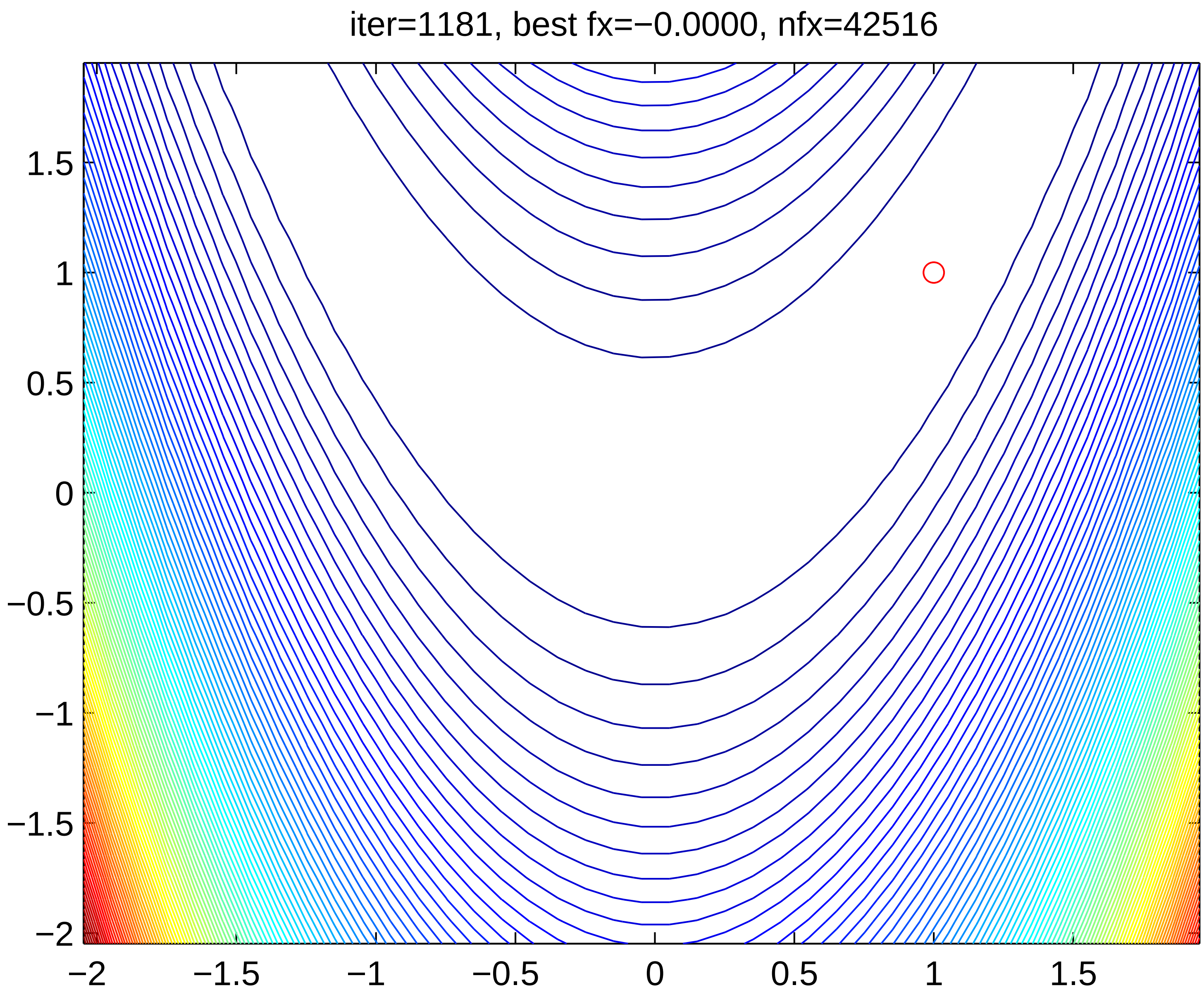


Example



iter=871, best fx=−0.0000, nfx=31356
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5

Example



iter=1181, best fx=−0.0000, nfx=42516
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5

Universidade do Minho

Some properties

- Easy to implement.
- Easy to deal with discrete variables.
- Easy to paralelize.
- For a correct choice of parameters the algorithm terminates ($\lim_{t \to +\infty} v(t) = 0$).
- Only objective function evaluations (without derivatives or approximation to derivatives).
- Convergence for a global optimum, but with a slow rate of convergence near an optimum.
- High number of function evaluations.

Some properties

- Easy to implement.
- Easy to deal with discrete variables.
- Easy to paralelize.
- For a correct choice of parameters the algorithm terminates ($\lim_{t \to +\infty} v(t) = 0$).
- Only objective function evaluations (without derivatives or approximation to derivatives).
- Convergence for a global optimum, but with a slow rate of convergence near an optimum.
- High number of function evaluations.

Some properties

- Easy to implement.
- Easy to deal with discrete variables.
- Easy to paralelize.
- For a correct choice of parameters the algorithm terminates ($\lim_{t \to +\infty} v(t) = 0$).
- Only objective function evaluations (without derivatives or approximation to derivatives).
- Convergence for a global optimum, but with a slow rate of convergence near an optimum.
- High number of function evaluations.

Universidade do Minho

Some properties

- Easy to implement.
- Easy to deal with discrete variables.
- Easy to paralelize.
- For a correct choice of parameters the algorithm terminates ($\lim_{t\to+\infty} v(t) = 0$).
- Only objective function evaluations (without derivatives or approximation to derivatives).
- Convergence for a global optimum, but with a slow rate of convergence near an optimum.
- High number of function evaluations.

Some properties

- Easy to implement.
- Easy to deal with discrete variables.
- Easy to paralelize.
- For a correct choice of parameters the algorithm terminates ($\lim_{t \to +\infty} v(t) = 0$).
- Only objective function evaluations (without derivatives or approximation to derivatives).
- Convergence for a global optimum, but with a slow rate of convergence near an optimum.
- High number of function evaluations.

Some properties

Universidade do Minho

- Easy to implement.
- Easy to deal with discrete variables.
- Easy to paralelize.
- For a correct choice of parameters the algorithm terminates ($\lim_{t \to +\infty} v(t) = 0$).
- Only objective function evaluations (without derivatives or approximation to derivatives).
- Convergence for a global optimum, but with a slow rate of convergence near an optimum.
- High number of function evaluations.

Some properties

Universidade do Minho

- Easy to implement.
- Easy to deal with discrete variables.
- Easy to paralelize.
- For a correct choice of parameters the algorithm terminates ($\lim_{t \to +\infty} v(t) = 0$).
- Only objective function evaluations (without derivatives or approximation to derivatives).
- Convergence for a global optimum, but with a slow rate of convergence near an optimum.
- High number of function evaluations.

Universidade do Minho

Pattern search

Introduction to direct methods

Direct search methods are an important class of optimization methods that try to minimize a function by comparing objective function values at a finite number of points.

Universidade do Minho

Introduction to direct methods

Direct search methods are an important class of optimization methods that try to minimize a function by comparing objective function values at a finite number of points.

Direct search methods do not use derivative information of the objective function nor try to approximate it.

Introduction to direct methods

Direct search methods are an important class of optimization methods that try to minimize a function by comparing objective function values at a finite number of points.

Direct search methods do not use derivative information of the objective function nor try to approximate it.

Pattern search method belongs to the class of direct search methods where its structure is more rigid.

Some definitions

Let

$$D_{\oplus} = \{e_1, \ldots, e_n, -e_1, \ldots, -e_n\}$$

be a positive maximal basis.

Universidade do Minho

Some definitions

Let

$$D_{\oplus} = \{e_1, \ldots, e_n, -e_1, \ldots, -e_n\}$$

be a positive maximal basis.

The direct method base on this set is known as coordinate or compass search.

Some definitions

Let

$$D_{\oplus} = \{e_1, \ldots, e_n, -e_1, \ldots, -e_n\}$$

be a positive maximal basis.

The direct method base on this set is known as coordinate or compass search.

Given a generating set D and the current point $y(t)$ two sets of points are defined: a grid M_t and the poll set P_t. The grid M_t is given by

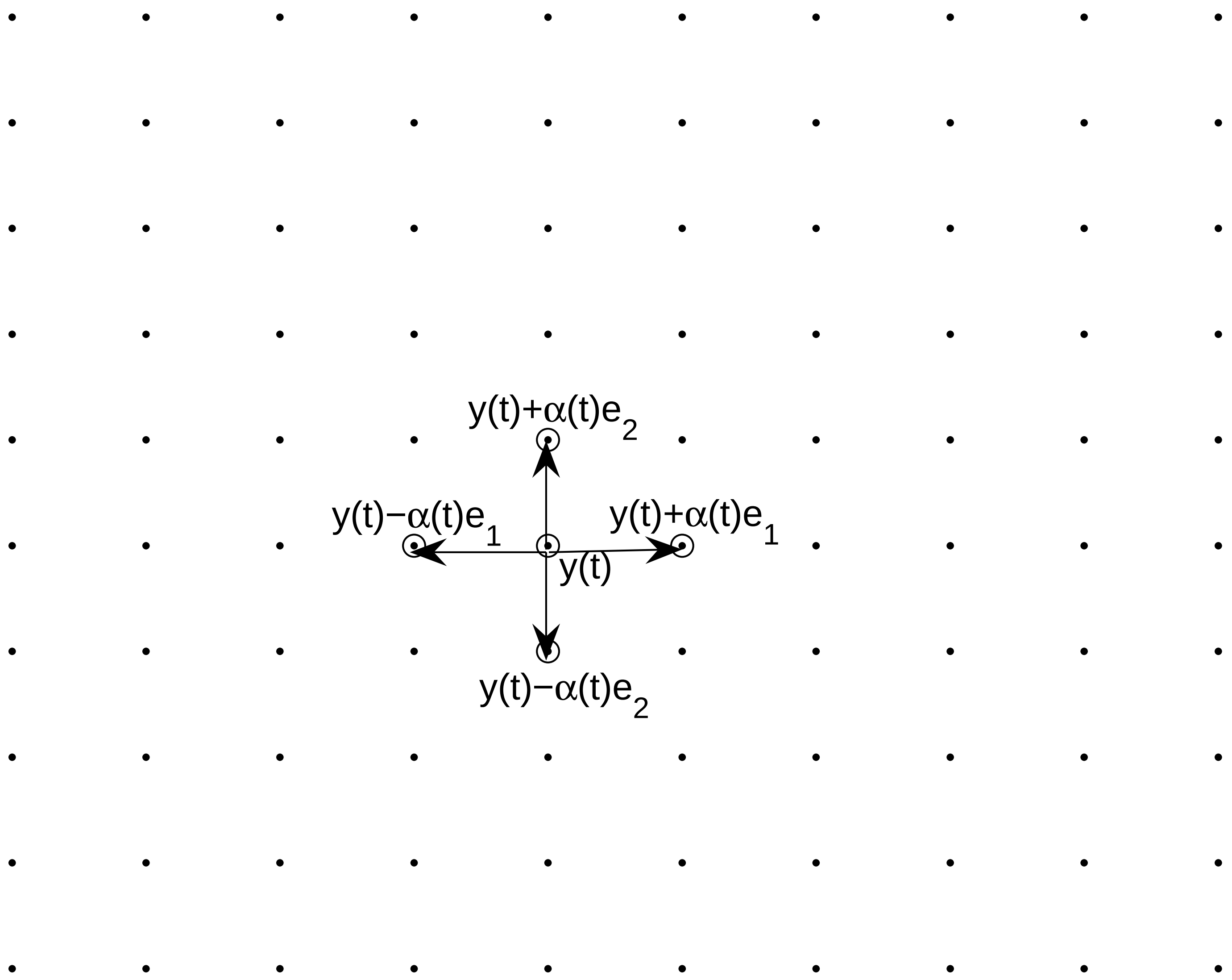
$$M_t \;=\; \left\{ y(t) + \alpha(t) D z, \; z \in \mathbb{N}_0^{|D|} \right\},$$

where $\alpha(t) > 0$ is the grid size parameter. The poll set is given by

$$P_t \;=\; \{ y(t) + \alpha(t) d, \; d \in D \} .$$

M_t and P_t sets example

Universidade do Minho



The set M_t and the set P_t when $D = \{e_1, e_2, -e_1, -e_2\}$

Pattern search

The search step conducts a finite search on the M_t grid.

Universidade do Minho

Universidade do Minho

Pattern search

The search step conducts a finite search on the M_t grid.

If no success is obtained in the search step then a poll step follows.

Pattern search

The search step conducts a finite search on the M_t grid.

If no success is obtained in the search step then a poll step follows.

The poll step evaluates the objective function in the elements of P_t in searching for points that has a lower objective function value.

If a success it attained the value of $\alpha(t)$ may be expended, otherwise it is reduced.

Universidade do Minho

Handling bound constraints

For the coordinate search method it is sufficient to initialize the algorithm with a feasible initial guess ($y(0) \in \Omega$) and to use $\hat{f}$ as the objective function.

$$\hat{f}(z) = \begin{cases} f(z) & \text{if } \; z \in \Omega, \\ +\infty & \text{otherwise.} \end{cases}$$

Universidade do Minho

The hybrid algorithm

Motivation

The hybrid algorithm tries to combine the best of each algorithms.

The particle swarm ability of searching for the global optimum.

The guaranty to obtain at least a stationary point in the pattern search.

Central idea: To apply the particle swarm algorithm in the search step and when no further success is possible to apply the poll step.

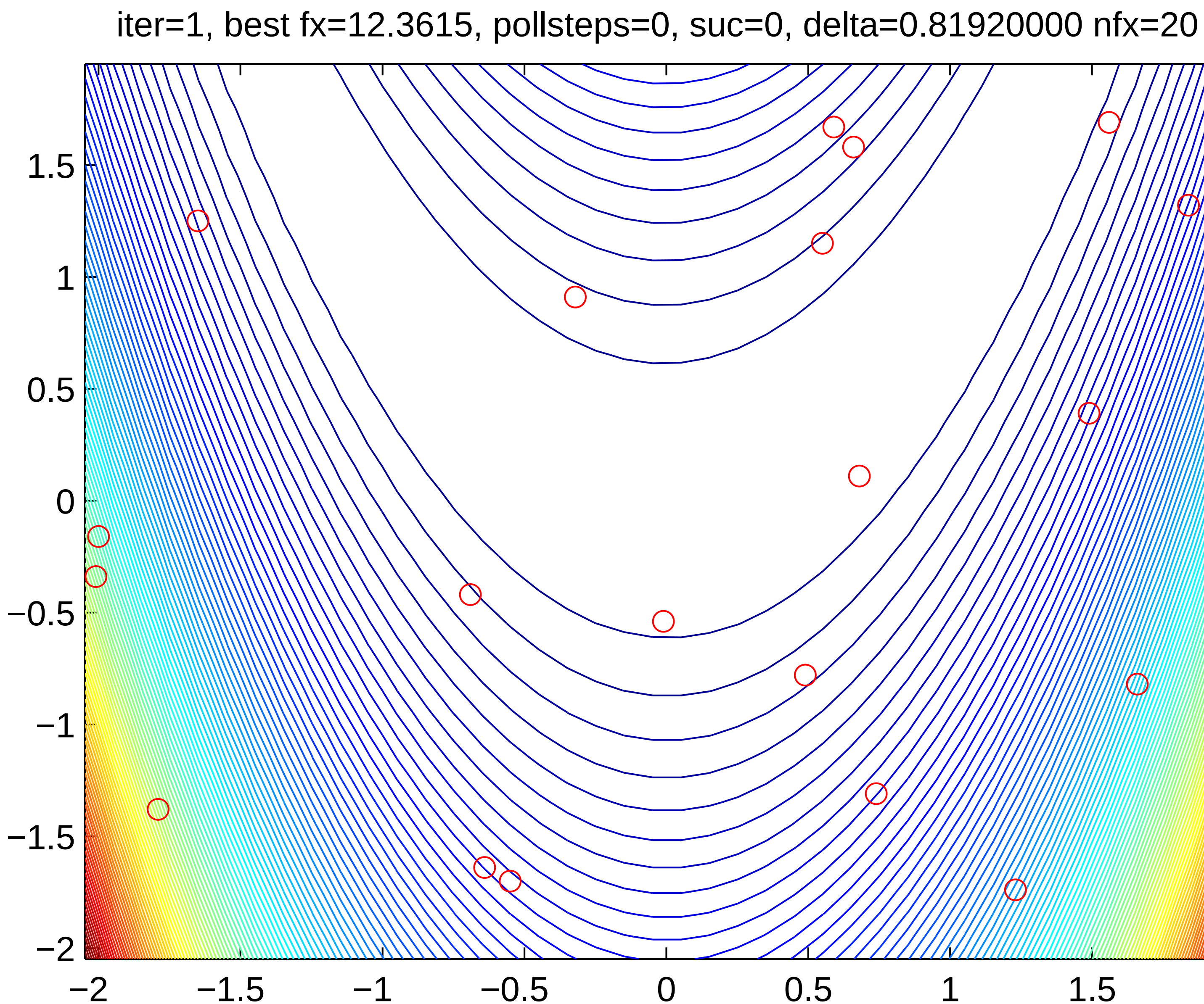
Universidade do Minho

Example with `ir2.mod` *problem*

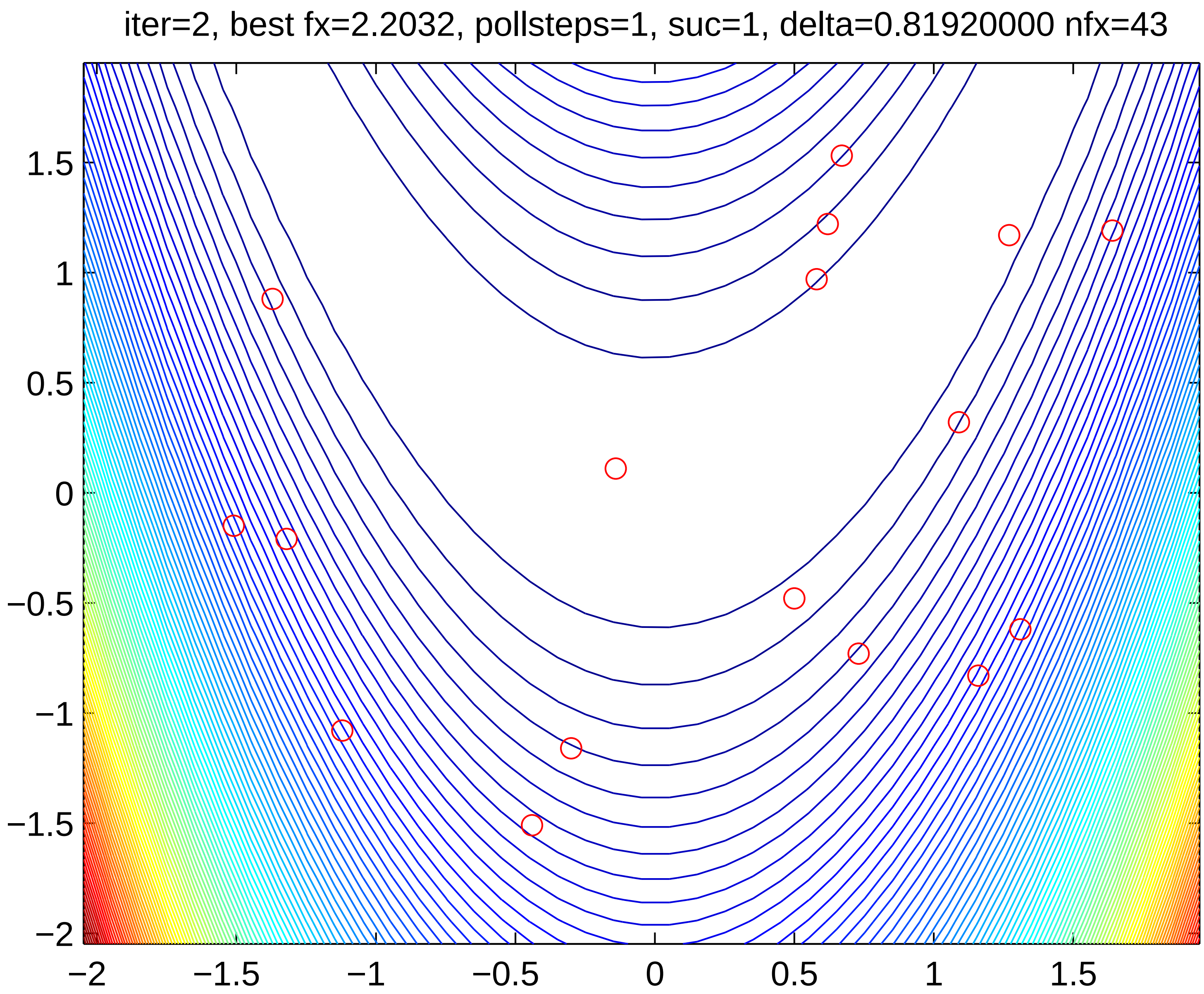
Universidade do Minho

The hybrid algorithm

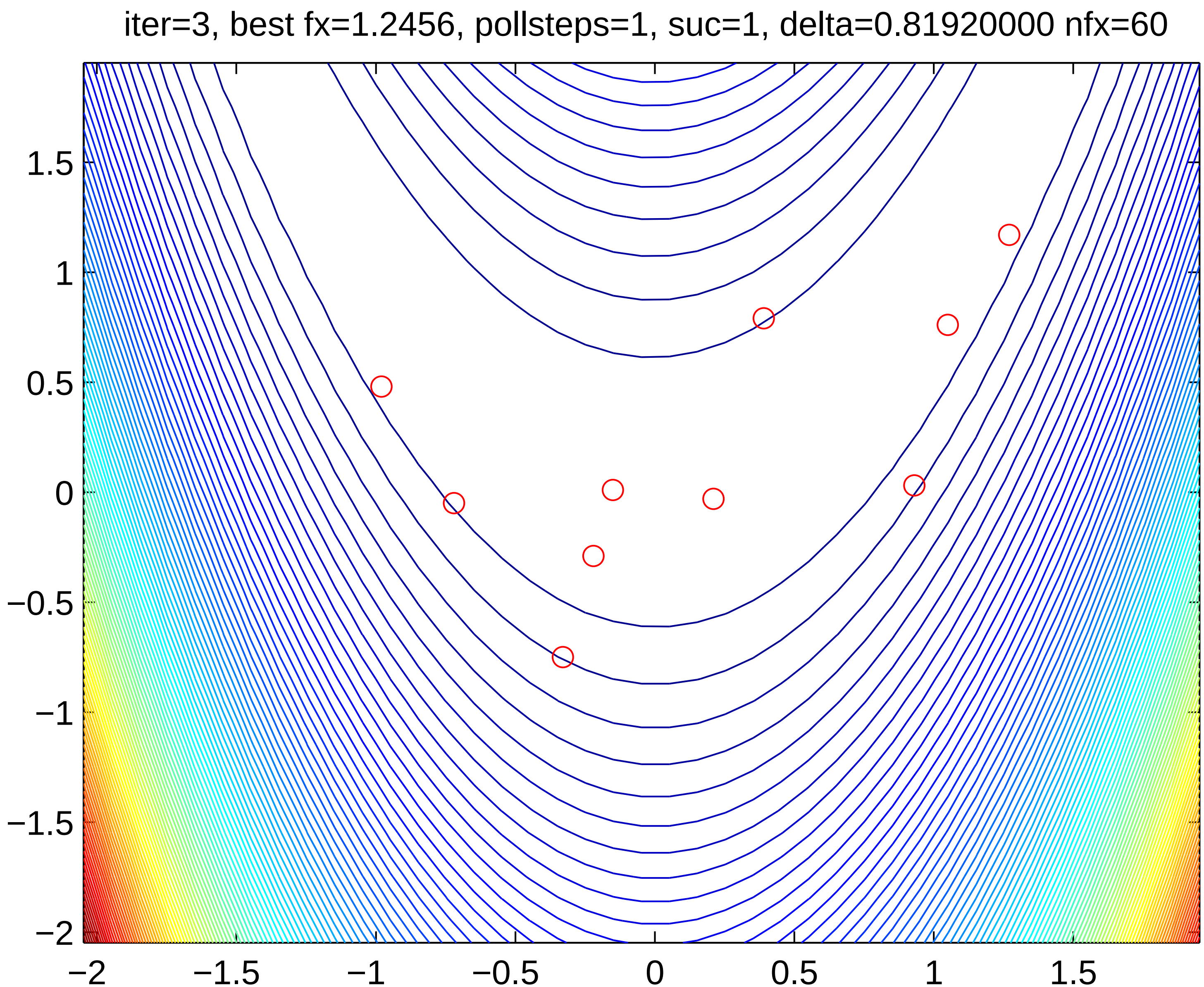
- Motivation
- Example



Example with `ir2.mod` *problem*



Example with `ir2.mod` *problem*



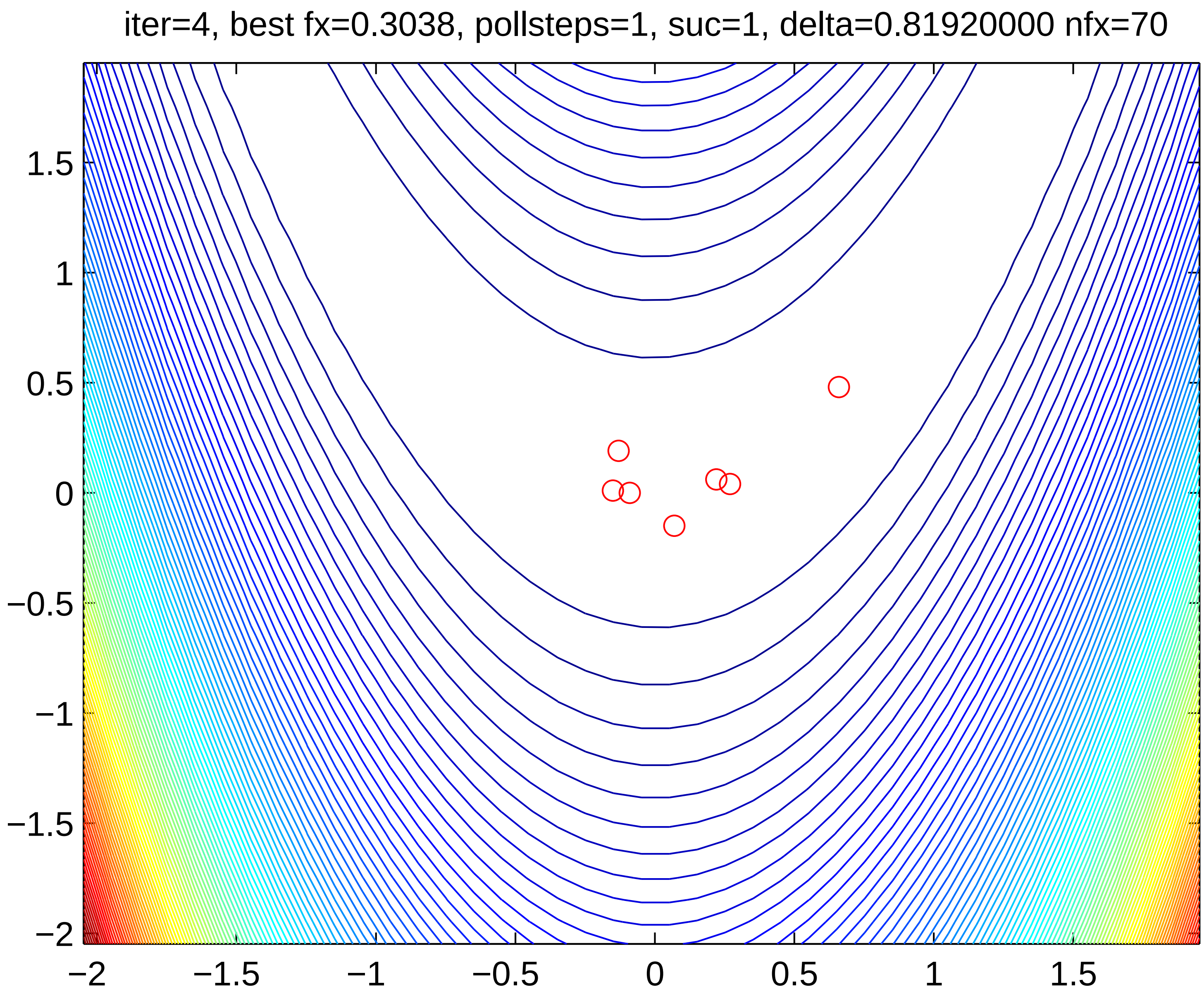
Example with `ir2.mod` *problem*

Universidade do Minho

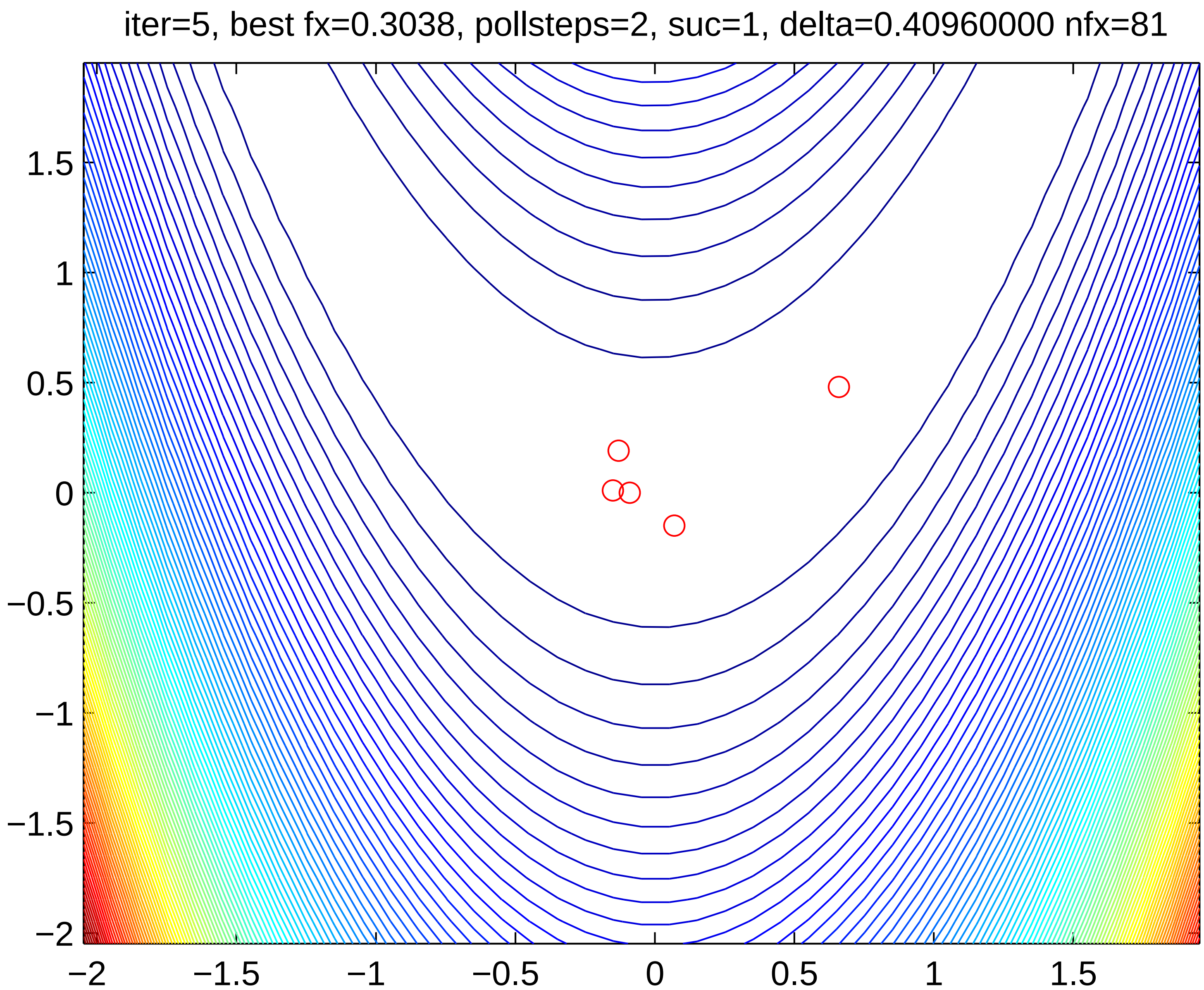
The hybrid algorithm

❖ Motivation

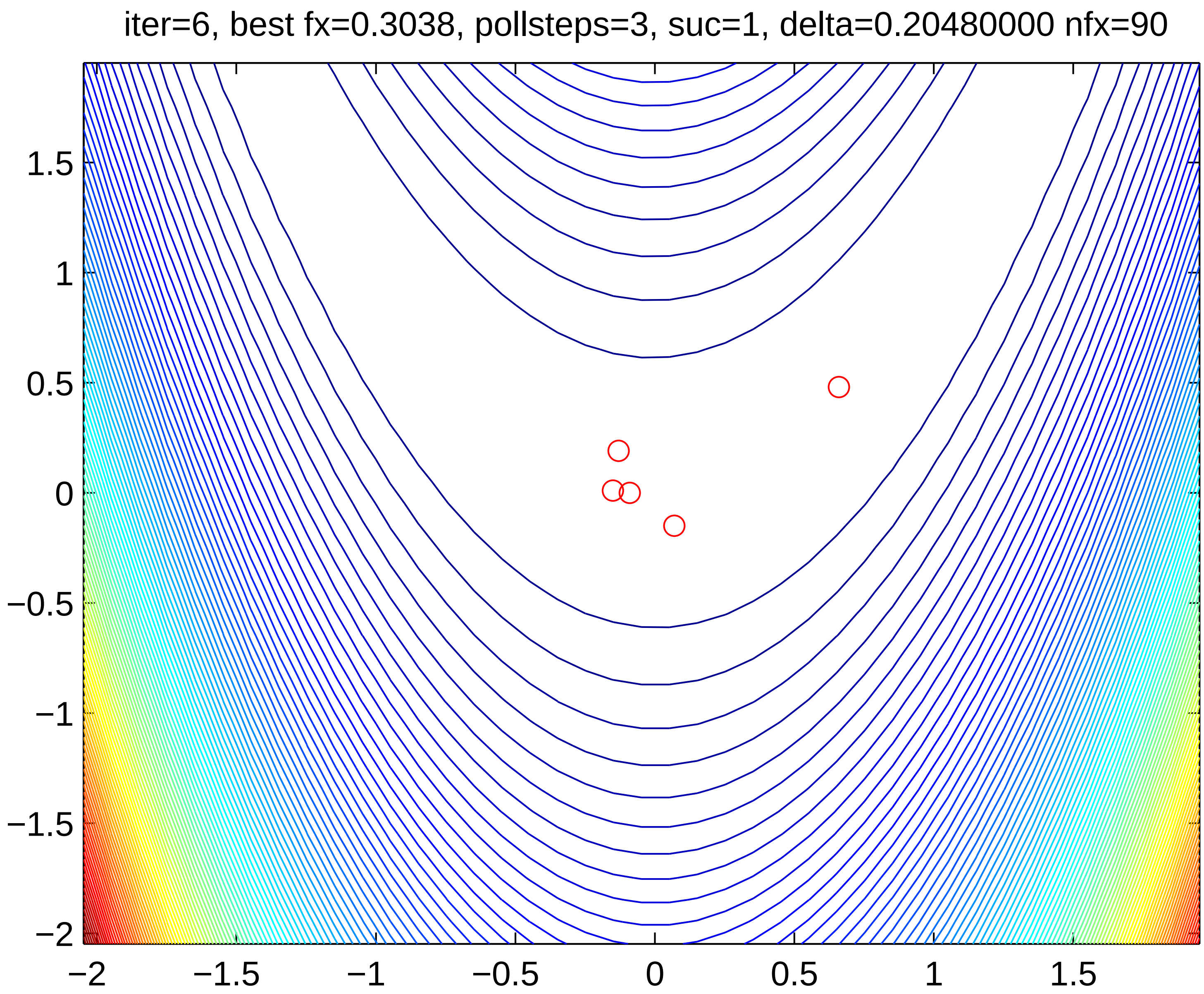
❖ Example



Example with `ir2.mod` *problem*



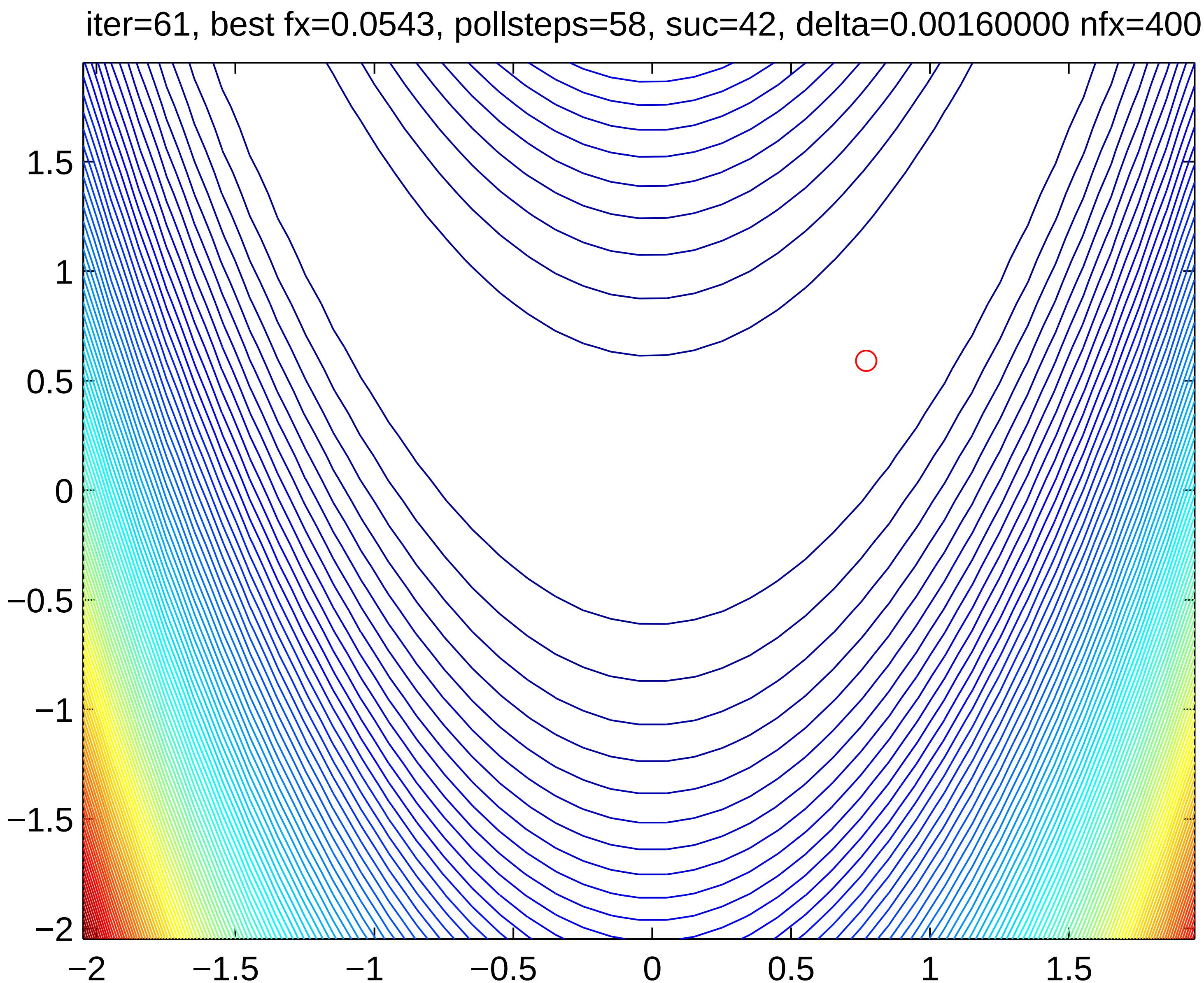
Example with `ir2.mod` *problem*



Example with `ir2.mod` *problem*

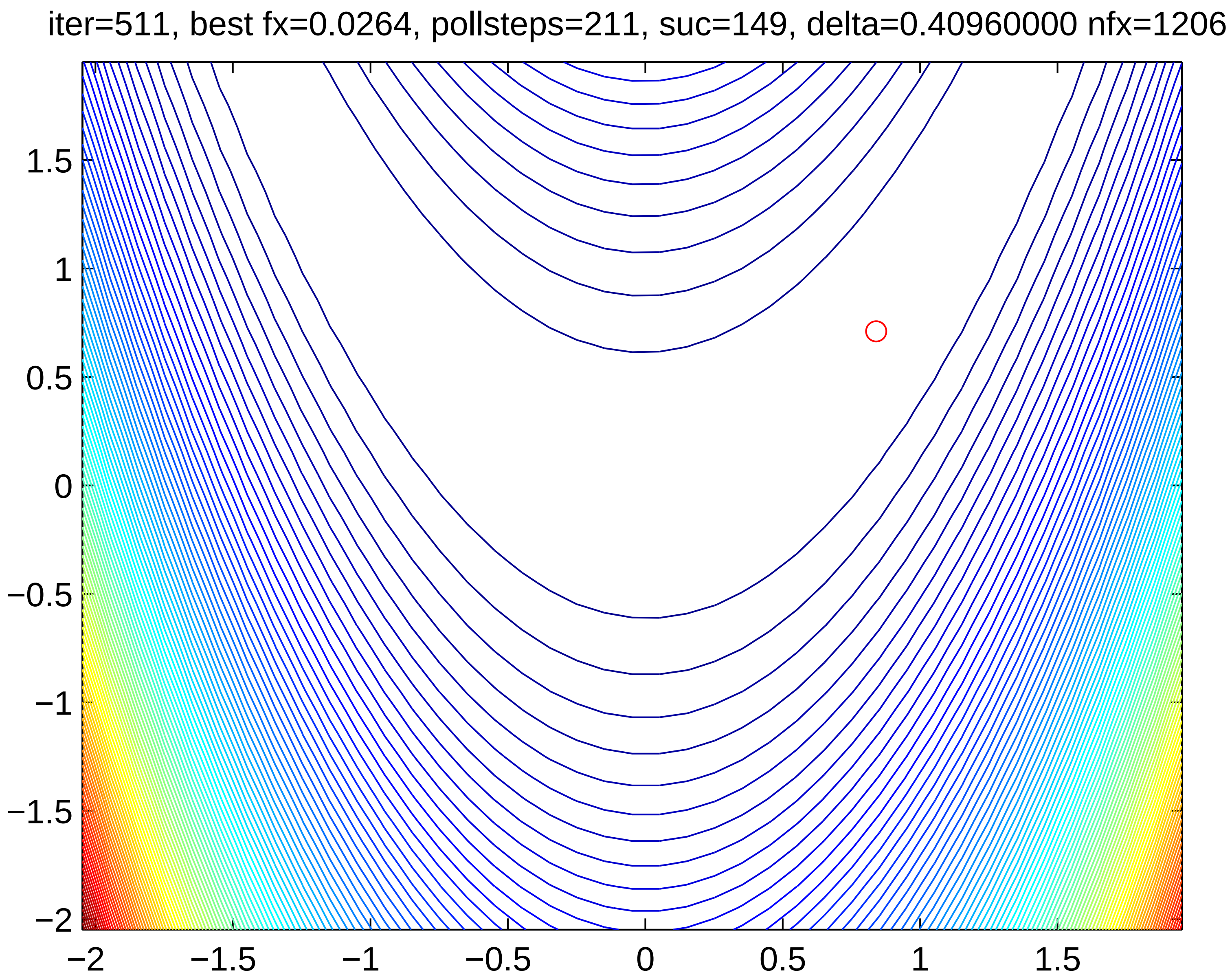
Universidade do Minho

- The hybrid algorithm
  - ❖ Motivation
  - ❖ Example



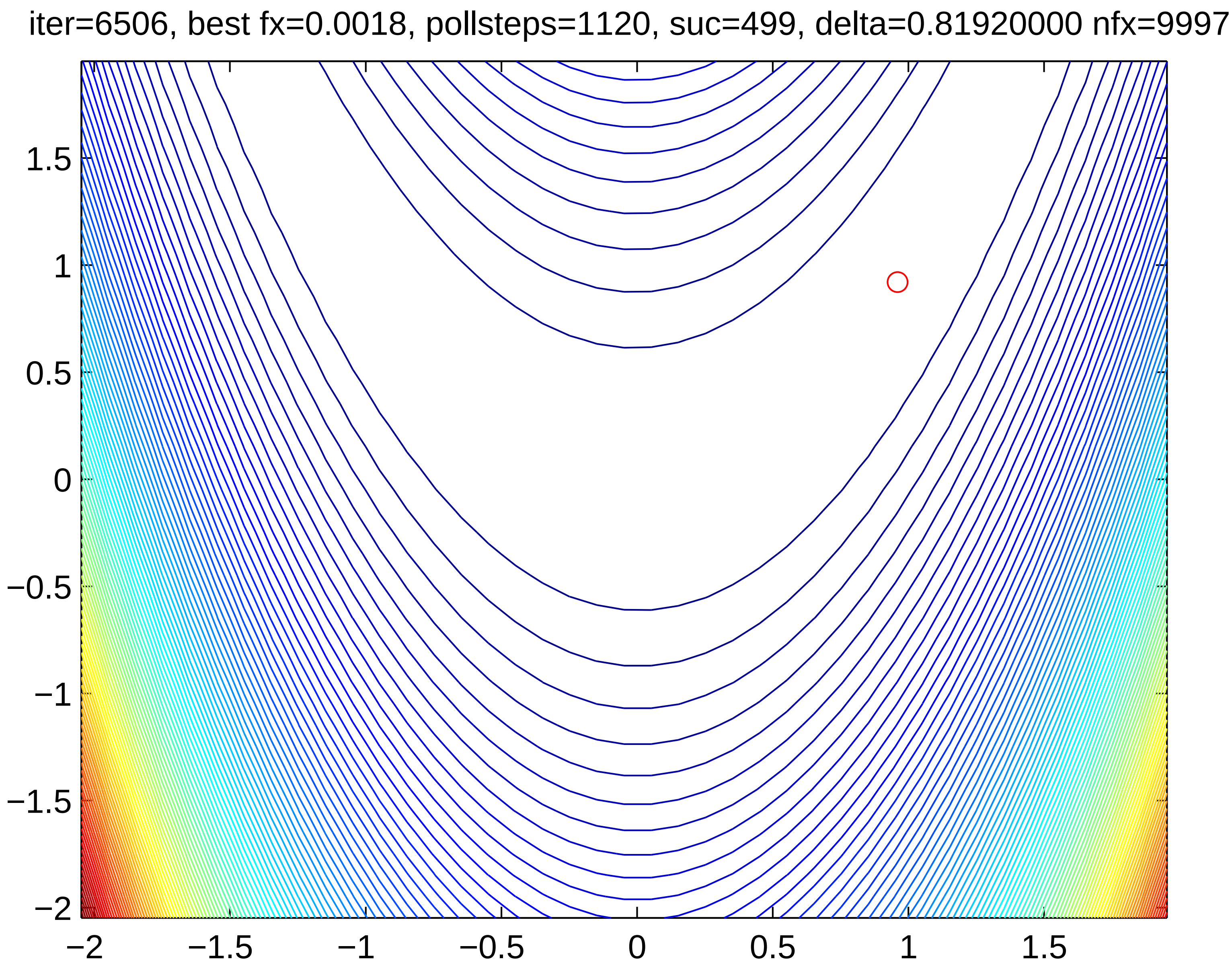
Example with `ir2.mod` *problem*

Universidade do Minho



Universidade do Minho

Example with `ir2.mod` *problem*



Universidade do Minho

Convergence

Global convergence

Universidade do Minho

Teorema 1 *Let* $L(\hat{y}(0)) = \{z \in \mathbb{R}^n : f(z) \leq f(\hat{y}(0))\}$ *be a bounded set. Then, there exists a subsequence* $\{\hat{y}(t_k)\}$ *of the iterates produced by the hybrid algorithm (with* $\alpha_{tol} = \upsilon_{tol} = 0$*) such that*

$$\lim_{k \longrightarrow +\infty} \hat{y}(t_k) \; = \; \hat{y}_* \quad \textit{and} \quad \lim_{k \longrightarrow +\infty} \alpha(t_k) \; = \; 0,$$

for some $\hat{y}_* \in \Omega$ *and such that the subsequence* $\{t_k\}$ *consists of unsuccessful iterations.*

Finite termination

Teorema 2 *Suppose that for t sufficiently large one has that $\iota(t)$, $E(y^i(t))$, $i = 1, \ldots, s$, and $E(\hat{y}(t))$ are constant and that $E(proj_{M_t}(x^i(t-1) + v^i(t))) = E(x^i(t-1) + v^i(t))$, $i = 1, \ldots, s$. Then, if the control parameters for particle swarm, $\bar{\iota}$, $\bar{\omega}_1$, $\bar{\omega}_2$, μ, and ν, are chosen so that $\max\{|a|, |b|\} < 1$, where $\bar{\omega}_1 = E(\omega_1(t))$, $\bar{\omega}_2 = E(\omega_2(t))$, $\bar{\iota} = \iota(t)$ for all t, and a and b are defined respectively by (1) and (2), then*

$$\lim_{t \longrightarrow +\infty} E(v^i_j(t)) \;=\; 0, \quad i = 1, \ldots, s, \; j = 1, \ldots, n.$$

and the hybrid algorithm will stop almost surely in a finite number of iterations.

$$a \;=\; \frac{(1 + \bar{\iota} - \mu\bar{\omega}_1 - \nu\bar{\omega}_2) + \sqrt{(1 + \bar{\iota} - \mu\bar{\omega}_1 - \nu\bar{\omega}_2)^2 - 4\bar{\iota}}}{2}, \qquad (1)$$

$$b \;=\; \frac{(1 + \bar{\iota} - \mu\bar{\omega}_1 - \nu\bar{\omega}_2) - \sqrt{(1 + \bar{\iota} - \mu\bar{\omega}_1 - \nu\bar{\omega}_2)^2 - 4\bar{\iota}}}{2}. \qquad (2)$$

Universidade do Minho

Numerical results

Test problems

Universidade do Minho

122 problems were collected from the global optimization literature.

12 problems of high dimension (between 100 and 300 variables). The others are small (< 10) and medium size (< 30).

Majority of objective functions are differentiable, but multimodal.

All problems have simple bounds on the variables (needed for the search step — particle swarm).

The test problems were coded in AMPL (*A Modeling Language for Mathematical Programming*).

Test problems available on `http://www.norg.uminho.pt/aivaz` (under *software*).

How to compare the solvers performance?

Universidade do Minho

Performance profiles – Dolan and Moré, 2003.

One advantage of the performance profiles is that it can be represented in one figure, drawing for each solver a cumulative distribution function $\rho(\tau)$ representing the performance ratio. $\rho_s(1)$ is the probability of solver s winning over the remaining ones. Bigger $\rho_s(1)$ values means higher probability of winning (be the best).

On the other hand solvers with higher $\rho_s(\tau)$, $\tau \to \infty$, are the most robust. If $\rho_s(\tau) = 1$ then solver s solved all the problems.

Used parameters

Universidade do Minho

`PSwarm`
$\alpha_{tol} = 10^{-5}$, $\nu = \mu = 0.5$, $\phi(t) = 2$, $\theta(t) = 0.5$,
$\alpha(0) = \max_{j=1,..,n}(u_j - \ell_j)/c$ with $c = 5$ and $s = 20$.

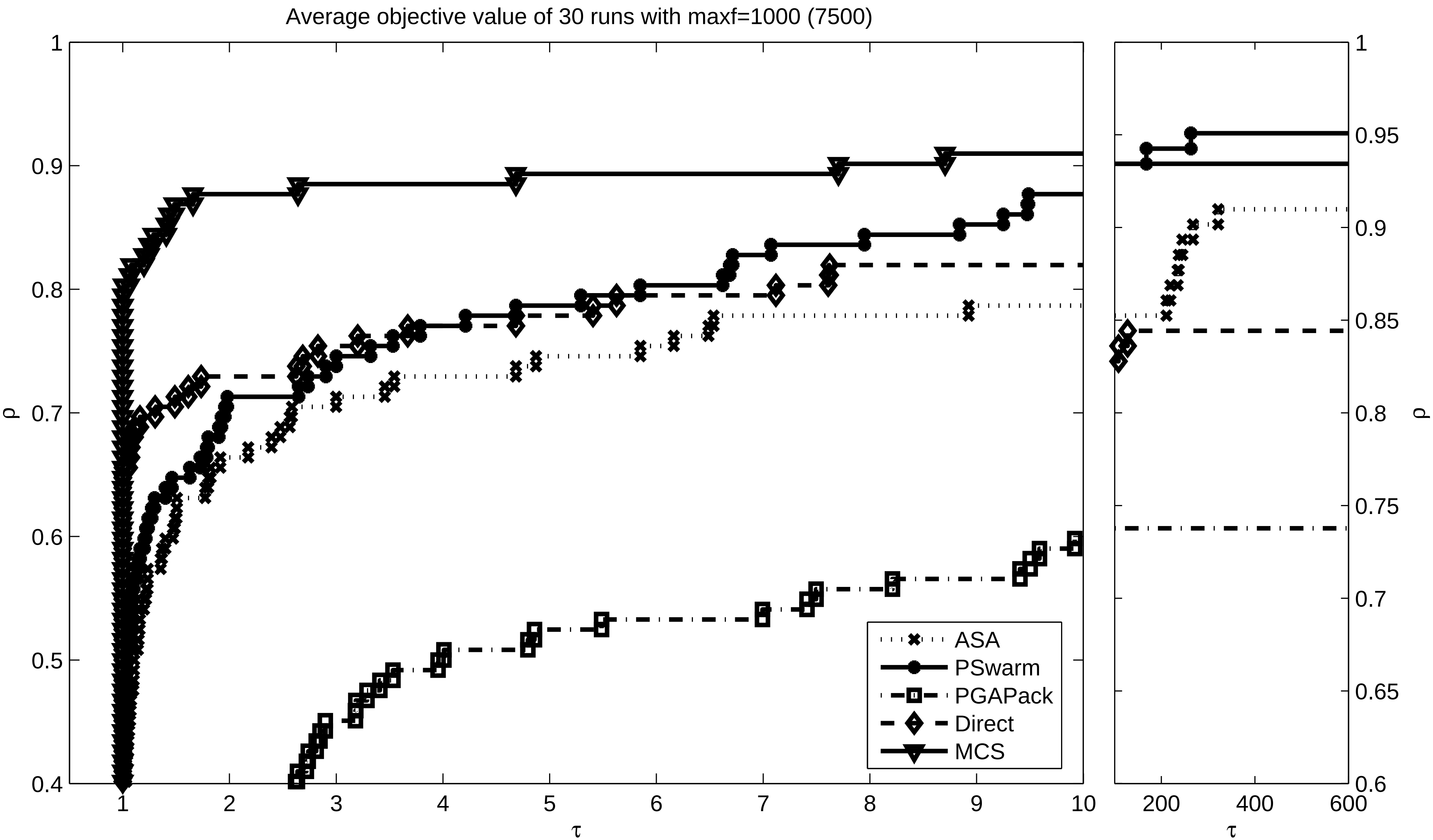
The inertial parameter ι was linearly interpolated between 0.9 and 0.4, *i.e.*, $\iota(t) = 0.9 - (0.5/t_{max})t$, where t_{max} is the maximum number of iterations allowed.

The initial population is obtained by generating s random points drawn from the uniform distribution $U(\ell, u)$, *i.e.*, $x_j^i(0) \sim U(\ell_j, u_j)$, $j = 1, \ldots, n$, for all particles $i = 1, \ldots, s$ (initial feasible approximations).

`PGAPack`
Genetic algorithm population of 200.

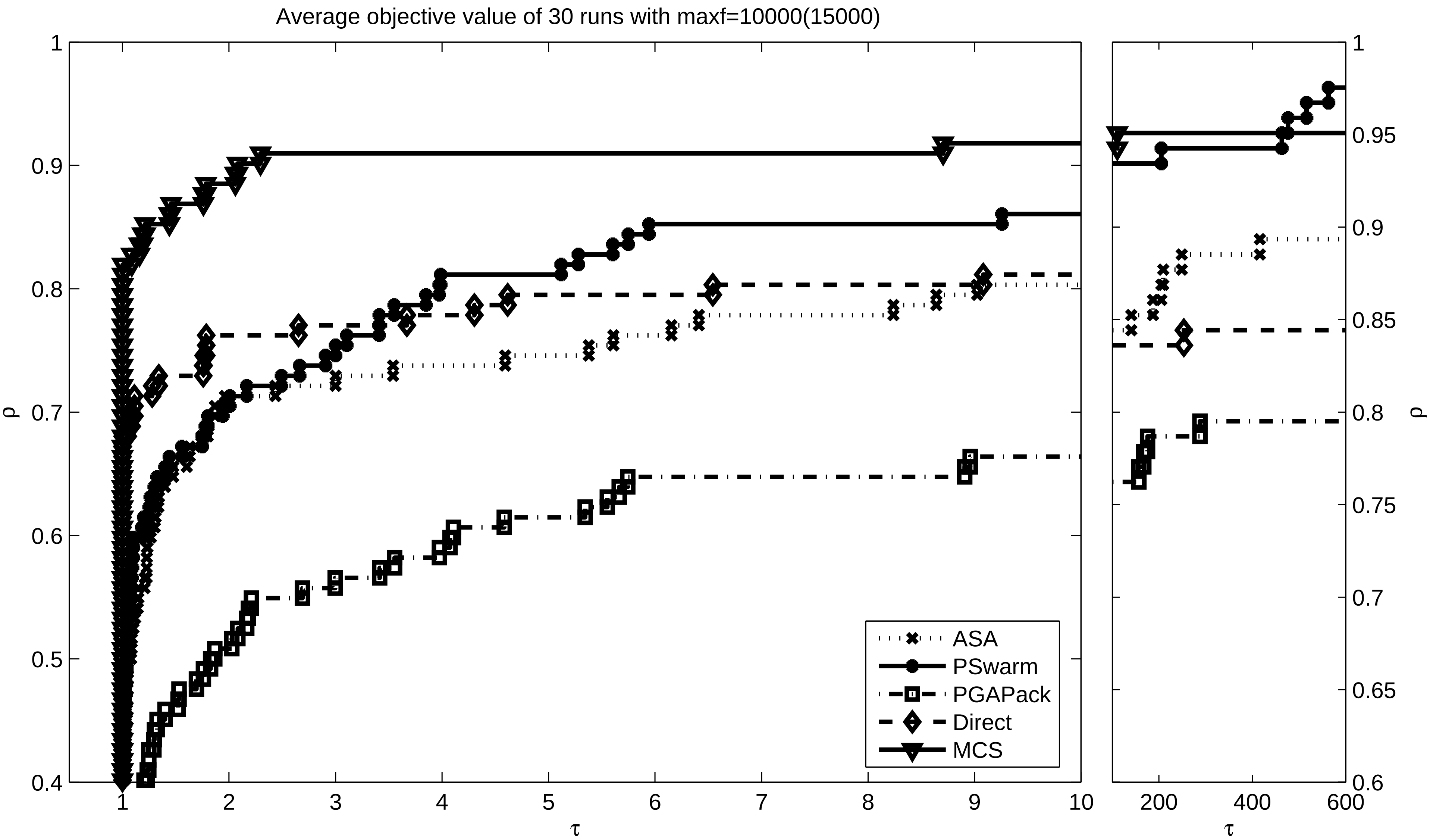
Average objective value

Universidade do Minho

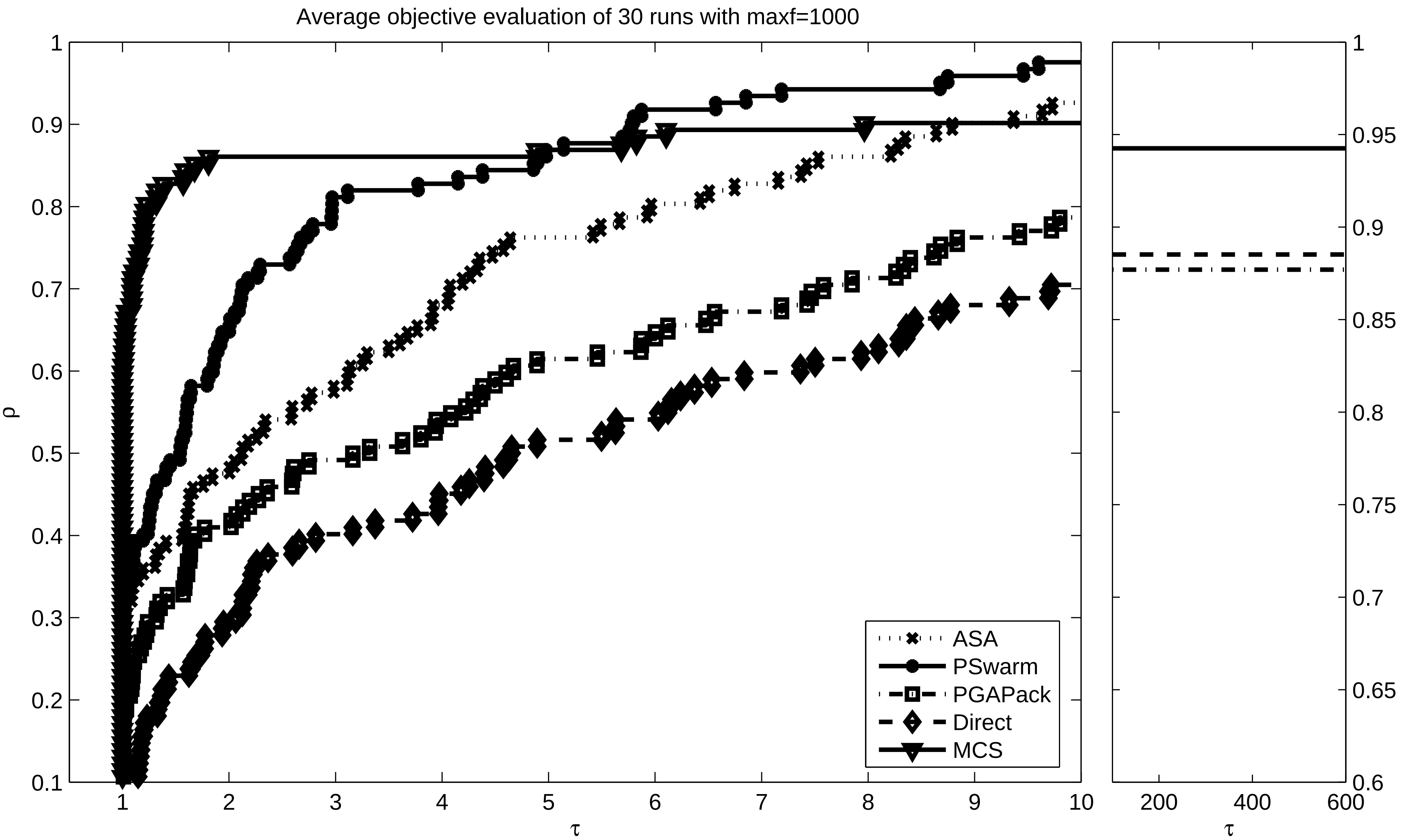


Average objective value

Universidade do Minho

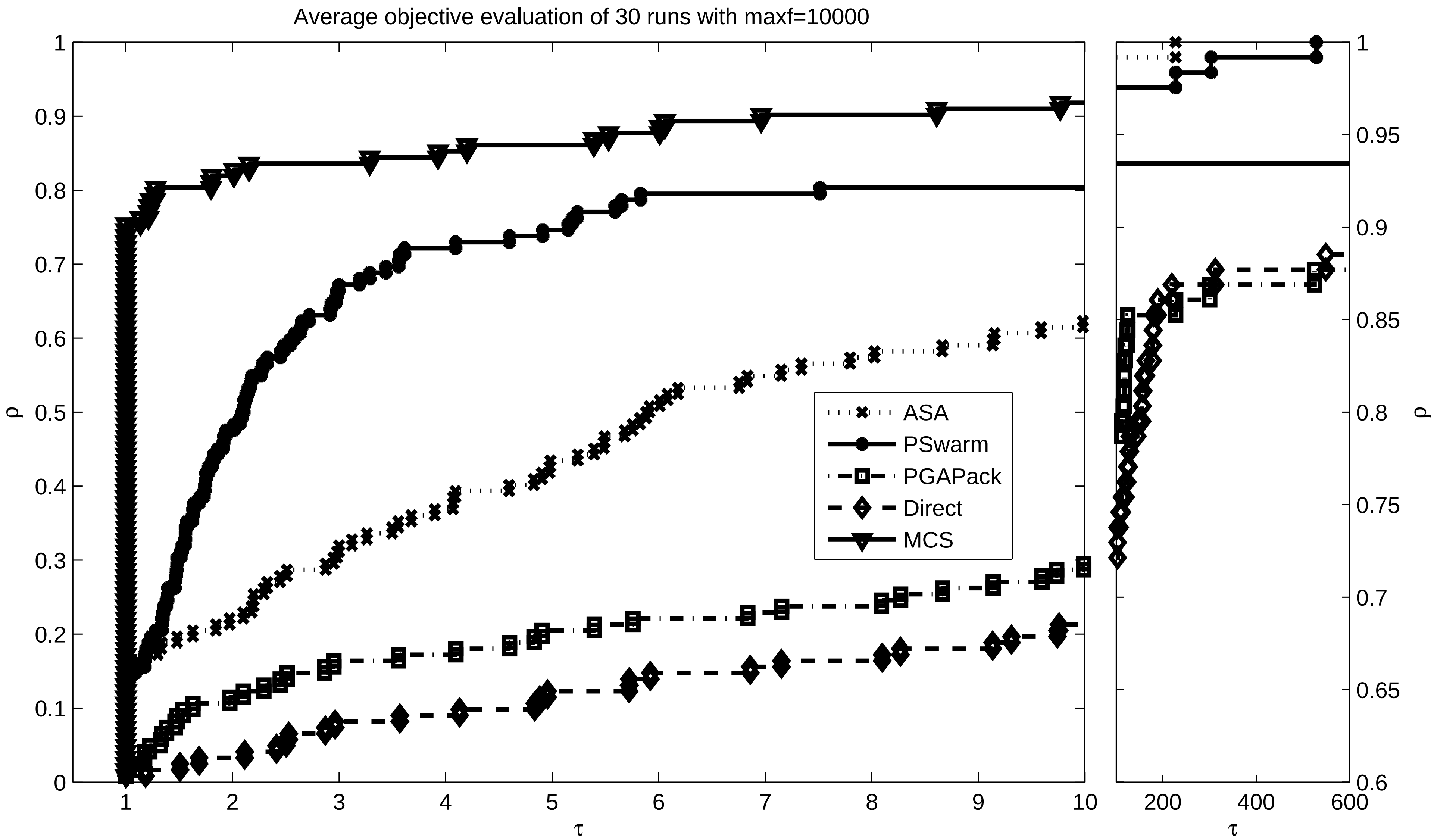


Average of objective function evaluations



Average of objective function evaluations

Universidade do Minho



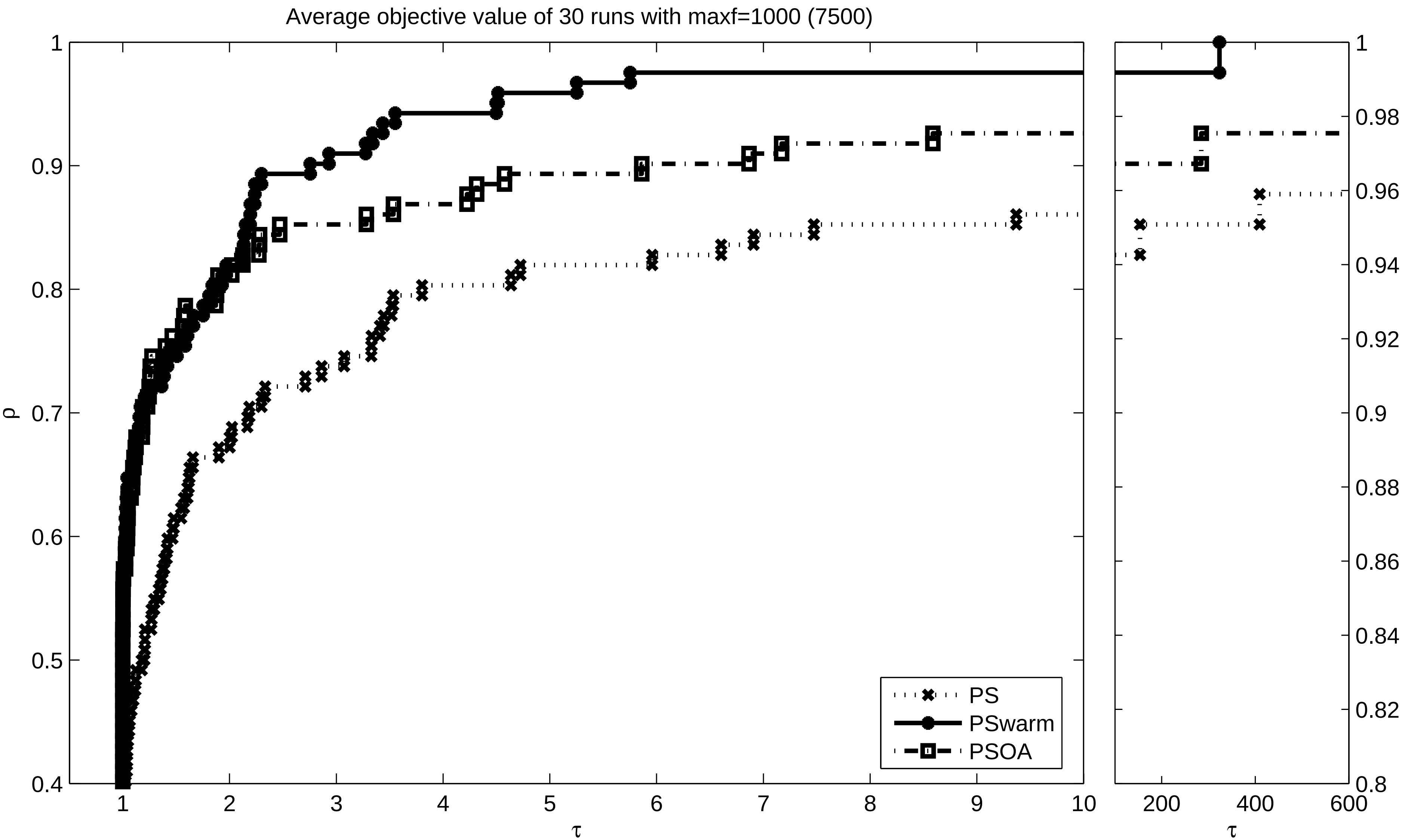
Average number of o.f. evaluation

$max f$	ASA	PGAPack	PSwarm	Direct	MCS
1000	857	1009	686	1107	1837
10000	5047	10009	3603	11517	4469

Universidade do Minho

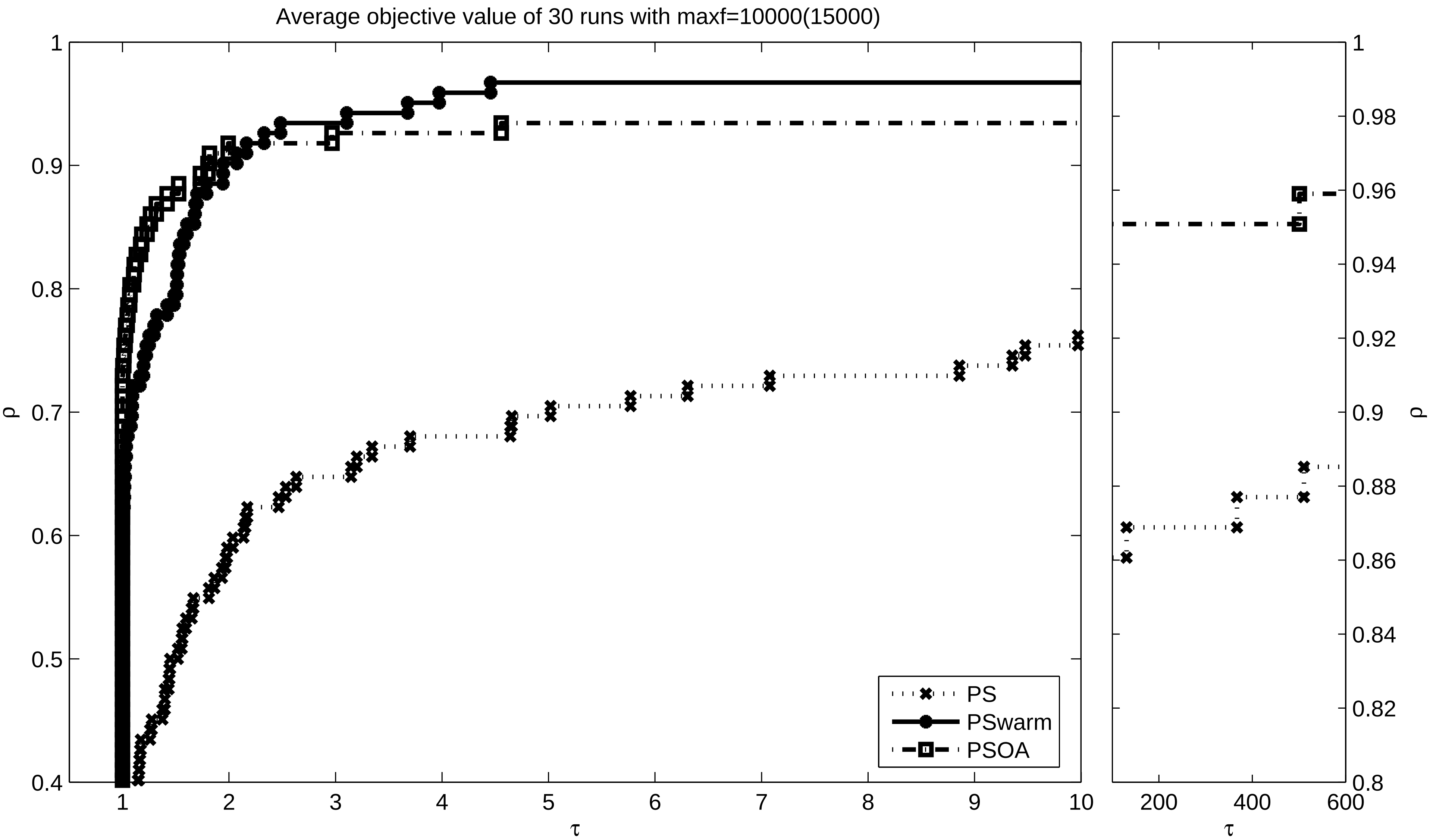
Pattern search vs Particle swarm vs PSwarm

Universidade do Minho



Pattern search vs Particle swarm vs PSwarm

Universidade do Minho



Conclusions and future work

Universidade do Minho

Conclusions and future work

Conclusions

- Development of an hybrid algorithm for global optimization.
- Convergence and termination properties of the algorithm.
- `PSwarm` shown to be an robust and competitive algorithm.

Future work

- Parallel version since both particle swarm and pattern search are easy to parallelize.
- More general constraints handling (linear and nonlinear).

Conclusions and future work

Conclusions

- Development of an hybrid algorithm for global optimization.
- Convergence and termination properties of the algorithm.
- `PSwarm` shown to be an robust and competitive algorithm.

Future work

- Parallel version since both particle swarm and pattern search are easy to parallelize.
- More general constraints handling (linear and nonlinear).

Universidade do Minho

Conclusions and future work

❖ Conclusions and future work

Conclusions and future work

Conclusions

- Development of an hybrid algorithm for global optimization.
- Convergence and termination properties of the algorithm.
- `PSwarm` shown to be an robust and competitive algorithm.

Future work

- Parallel version since both particle swarm and pattern search are easy to parallelize.
- More general constraints handling (linear and nonlinear).

Conclusions and future work

Conclusions

- Development of an hybrid algorithm for global optimization.
- Convergence and termination properties of the algorithm.
- `PSwarm` shown to be an robust and competitive algorithm.

Future work

- Parallel version since both particle swarm and pattern search are easy to parallelize.
- More general constraints handling (linear and nonlinear).

Universidade do Minho

Conclusions and future work

Conclusions

- Development of an hybrid algorithm for global optimization.
- Convergence and termination properties of the algorithm.
- `PSwarm` shown to be an robust and competitive algorithm.

Future work

- Parallel version since both particle swarm and pattern search are easy to parallelize.
- More general constraints handling (linear and nonlinear).

Universidade do Minho

The End

email: aivaz@dps.uminho.pt

Web http://www.norg.uminho.pt/

email: lnv@mat.uc.pt

Web http://www.mat.uc.pt/∼lnv