Extension of PSwarm to Linearly Constrained Derivative-Free Global Optimization

A. Ismael F. Vaz[1] and Luís Nunes Vicente[2]

[1]University of Minho - Portugal
aivaz@dps.uminho.pt

[2]University of Coimbra - Portugal
lnv@mat.uc.pt

SIAM Conference on Optimization

May 10-13, 2008

Outline

Outline

Outline

Outline

Problem formulation

The problem we are addressing is:

Problem definition - bound constraints

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

where $\ell \leq z \leq u$ are understood componentwise.

Smoothness – Assumption

To apply particle swarm or coordinate search, smoothness of the objective function $f(z)$ is not required.
For the convergence analysis of coordinate search, and therefore of the hybrid algorithm, some smoothness of the objective function $f(z)$ is imposed.

Problem formulation

The problem we are addressing is:

Problem definition - bound constraints

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

where $\ell \leq z \leq u$ are understood componentwise.

Smoothness – Assumption

To apply particle swarm or coordinate search, smoothness of the objective function $f(z)$ is not required.
For the convergence analysis of coordinate search, and therefore of the hybrid algorithm, some smoothness of the objective function $f(z)$ is imposed.

Particle Swarm (new position and velocity)

The new particle position is updated by

Update particle

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

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

Update velocity

$$v^p(t+1) = \iota(t)v^p(t) + \mu\omega_1(t)\left(y^p(t) - x^p(t)\right) + \nu\omega_2(t)\left(\hat{y}(t) - x^p(t)\right),$$

where $\iota(t)$, μ and ν are parameters and $\omega_1(t)$ and $\omega_2(t)$ are random vectors drawn from the uniform $(0, 1)$ distribution.
$y^p(t)$ is the best particle p position and $\hat{y}(t)$ is the best population position.

Particle Swarm (new position and velocity)

The new particle position is updated by

Update particle

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

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

Update velocity

$$v^p(t+1) = \iota(t)v^p(t) + \mu\omega_1(t)\left(y^p(t) - x^p(t)\right) + \nu\omega_2(t)\left(\hat{y}(t) - x^p(t)\right),$$

where $\iota(t)$, μ and ν are parameters and $\omega_1(t)$ and $\omega_2(t)$ are random vectors drawn from the uniform $(0, 1)$ distribution.
$y^p(t)$ is the best particle p position and $\hat{y}(t)$ is the best population position.

Some properties

- Easy to implement.
- Easy to deal with discrete variables.
- Easy to parallelize.
- For a correct choice of parameters the algorithm terminates ($\lim_{t \to +\infty} v(t) = 0$).
- Uses only objective function values.
- Convergence for a global optimum under strong assumptions (unpractical).
- High number of function evaluations.

Some properties

- Easy to implement.
- Easy to deal with discrete variables.
- Easy to parallelize.
- For a correct choice of parameters the algorithm terminates ($\lim_{t \to +\infty} v(t) = 0$).
- Uses only objective function values.
- Convergence for a global optimum under strong assumptions (unpractical).
- High number of function evaluations.

Some properties

- Easy to implement.
- Easy to deal with discrete variables.
- Easy to parallelize.
- For a correct choice of parameters the algorithm terminates ($\lim_{t \to +\infty} v(t) = 0$).
- Uses only objective function values.
- Convergence for a global optimum under strong assumptions (unpractical).
- High number of function evaluations.

Some properties

- Easy to implement.
- Easy to deal with discrete variables.
- Easy to parallelize.
- For a correct choice of parameters the algorithm terminates ($\lim_{t \to +\infty} v(t) = 0$).
- Uses only objective function values.
- Convergence for a global optimum under strong assumptions (unpractical).
- High number of function evaluations.

Some properties

- Easy to implement.
- Easy to deal with discrete variables.
- Easy to parallelize.
- For a correct choice of parameters the algorithm terminates ($\lim_{t \to +\infty} v(t) = 0$).
- Uses only objective function values.
- Convergence for a global optimum under strong assumptions (unpractical).
- High number of function evaluations.

Some properties

- Easy to implement.
- Easy to deal with discrete variables.
- Easy to parallelize.
- For a correct choice of parameters the algorithm terminates ($\lim_{t\to+\infty} v(t) = 0$).
- Uses only objective function values.
- Convergence for a global optimum under strong assumptions (unpractical).
- High number of function evaluations.

Some properties

- Easy to implement.
- Easy to deal with discrete variables.
- Easy to parallelize.
- For a correct choice of parameters the algorithm terminates ($\lim_{t \to +\infty} v(t) = 0$).
- Uses only objective function values.
- Convergence for a global optimum under strong assumptions (unpractical).
- High number of function evaluations.

Introduction to direct search 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.
- Coordinate search is a simple direct search method.

Introduction to direct search 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.

- Coordinate search is a simple direct search method.

Introduction to direct search 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.

- Coordinate search is a simple direct search method.

Some definitions

Maximal positive basis

Formed by the coordinate vectors and their negative counterparts:

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

$D_{\oplus}$ spans $\mathbb{R}^n$ with nonnegative coefficients.

Coordinate search

The direct search method based on $D_{\oplus}$ is known as coordinate or compass search.

Some definitions

Maximal positive basis

Formed by the coordinate vectors and their negative counterparts:

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

$D_{\oplus}$ spans $\mathbb{R}^n$ with nonnegative coefficients.

Coordinate search

The direct search method based on $D_{\oplus}$ is known as coordinate or compass search.

Some definitions

Given $D_\oplus$ and the current point $y(t)$, two sets of points are defined: a grid M_t and the poll set P_t.

Sets

The grid M_t is given by

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

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

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

Some definitions

Given $D_\oplus$ and the current point $y(t)$, two sets of points are defined: a grid M_t and the poll set P_t.

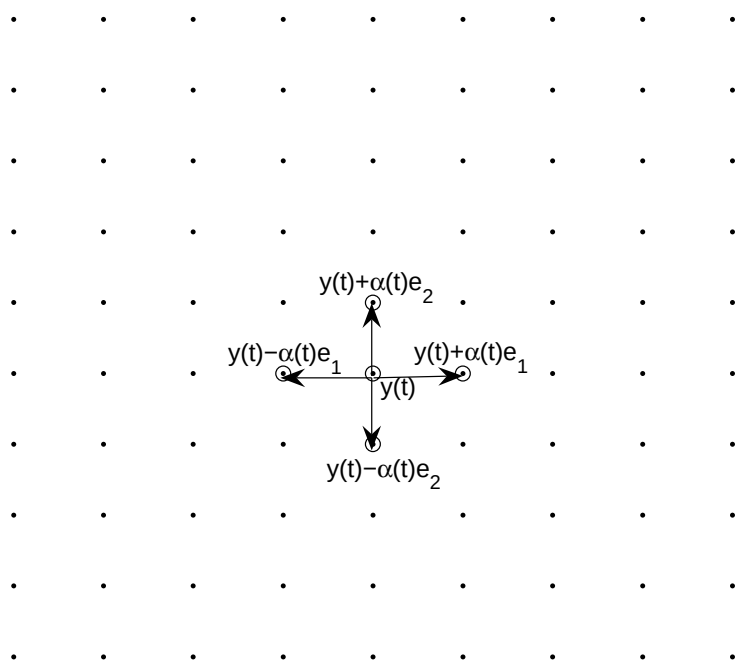
Sets

The grid M_t is given by

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

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

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



The grid M_t and the set P_t when $D_\oplus = \{e_1, e_2, -e_1, -e_2\}$.

Coordinate search

- The search step conducts a finite search on the grid M_t.
- If no success is obtained in the search step then a poll step follows.
- The poll step evaluates the objective function at the elements of P_t, searching for points which have a lower objective function value.
- If success is attained, the value of $\alpha(t)$ may be increased, otherwise it is reduced.

Coordinate search

- The search step conducts a finite search on the grid M_t.
- If no success is obtained in the search step then a poll step follows.
- The poll step evaluates the objective function at the elements of P_t, searching for points which have a lower objective function value.
- If success is attained, the value of $\alpha(t)$ may be increased, otherwise it is reduced.

Coordinate search

- The search step conducts a finite search on the grid M_t.
- If no success is obtained in the search step then a poll step follows.
- The poll step evaluates the objective function at the elements of P_t, searching for points which have a lower objective function value.
- If success is attained, the value of $\alpha(t)$ may be increased, otherwise it is reduced.

Coordinate search

- The search step conducts a finite search on the grid M_t.
- If no success is obtained in the search step then a poll step follows.
- The poll step evaluates the objective function at the elements of P_t, searching for points which have a lower objective function value.
- If success is attained, the value of $\alpha(t)$ may be increased, otherwise it is reduced.

Motivation for PSwarm

Central idea

A particle swarm iteration is performed in the search step and if no progress is attained a poll step is taken.

Key points

- In the first iterations the algorithm takes advantage of the particle swarm ability to find a global optimum (exploiting the search space), while in the last iterations the algorithm takes advantage of the pattern search robustness to find a stationary point.
- The number of particles in the swarm search can be decreased along the iterations (no need to have a large number of particles around a local optimum).

Motivation for PSwarm

Central idea

A particle swarm iteration is performed in the search step and if no progress is attained a poll step is taken.

Key points

- In the first iterations the algorithm takes advantage of the particle swarm ability to find a global optimum (exploiting the search space), while in the last iterations the algorithm takes advantage of the pattern search robustness to find a stationary point.
- The number of particles in the swarm search can be decreased along the iterations (no need to have a large number of particles around a local optimum).

Motivation for PSwarm

Central idea

A particle swarm iteration is performed in the search step and if no progress is attained a poll step is taken.

Key points

- In the first iterations the algorithm takes advantage of the particle swarm ability to find a global optimum (exploiting the search space), while in the last iterations the algorithm takes advantage of the pattern search robustness to find a stationary point.
- The number of particles in the swarm search can be decreased along the iterations (no need to have a large number of particles around a local optimum).

Motivation for PSwarm

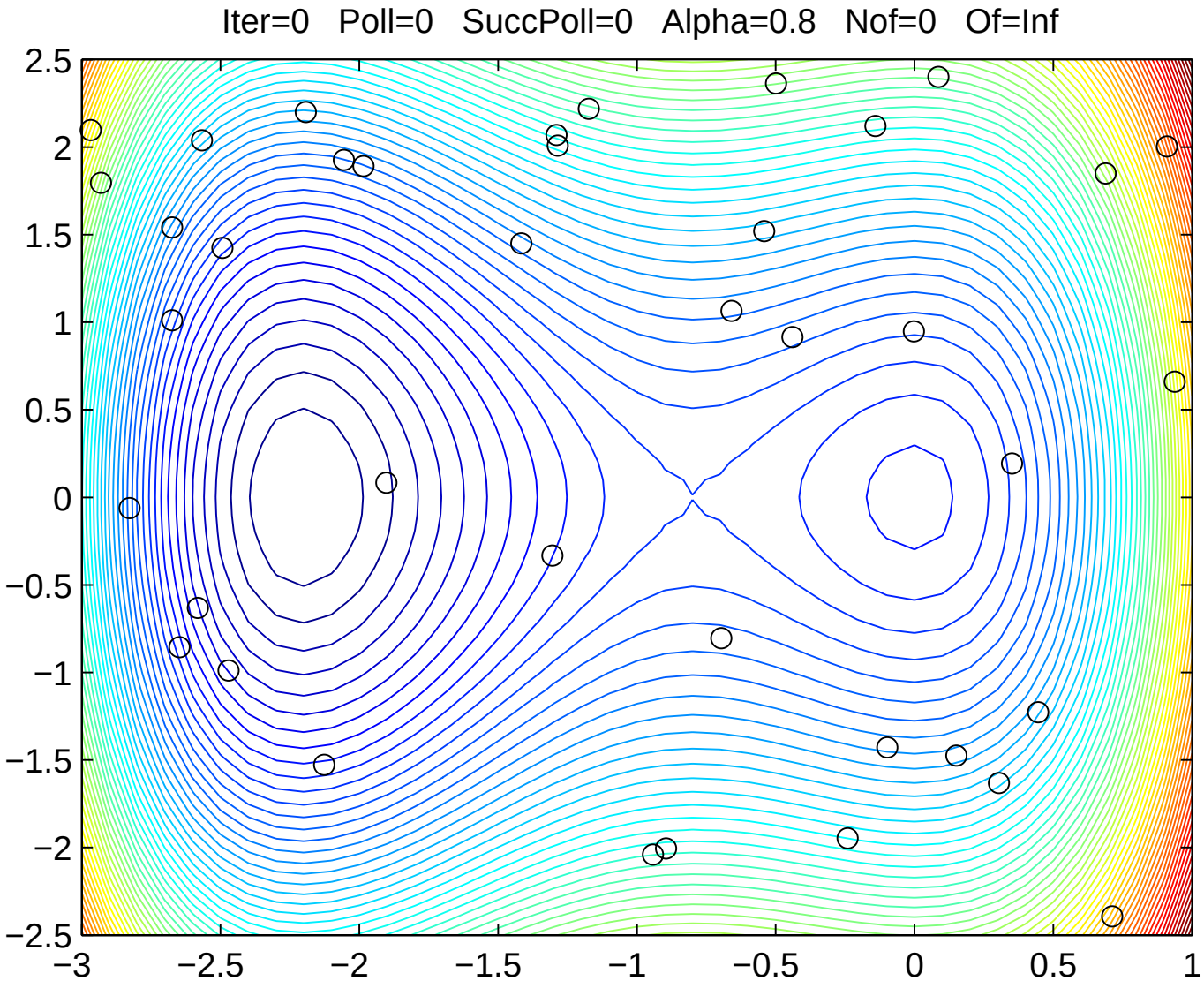
Central idea

A particle swarm iteration is performed in the search step and if no progress is attained a poll step is taken.

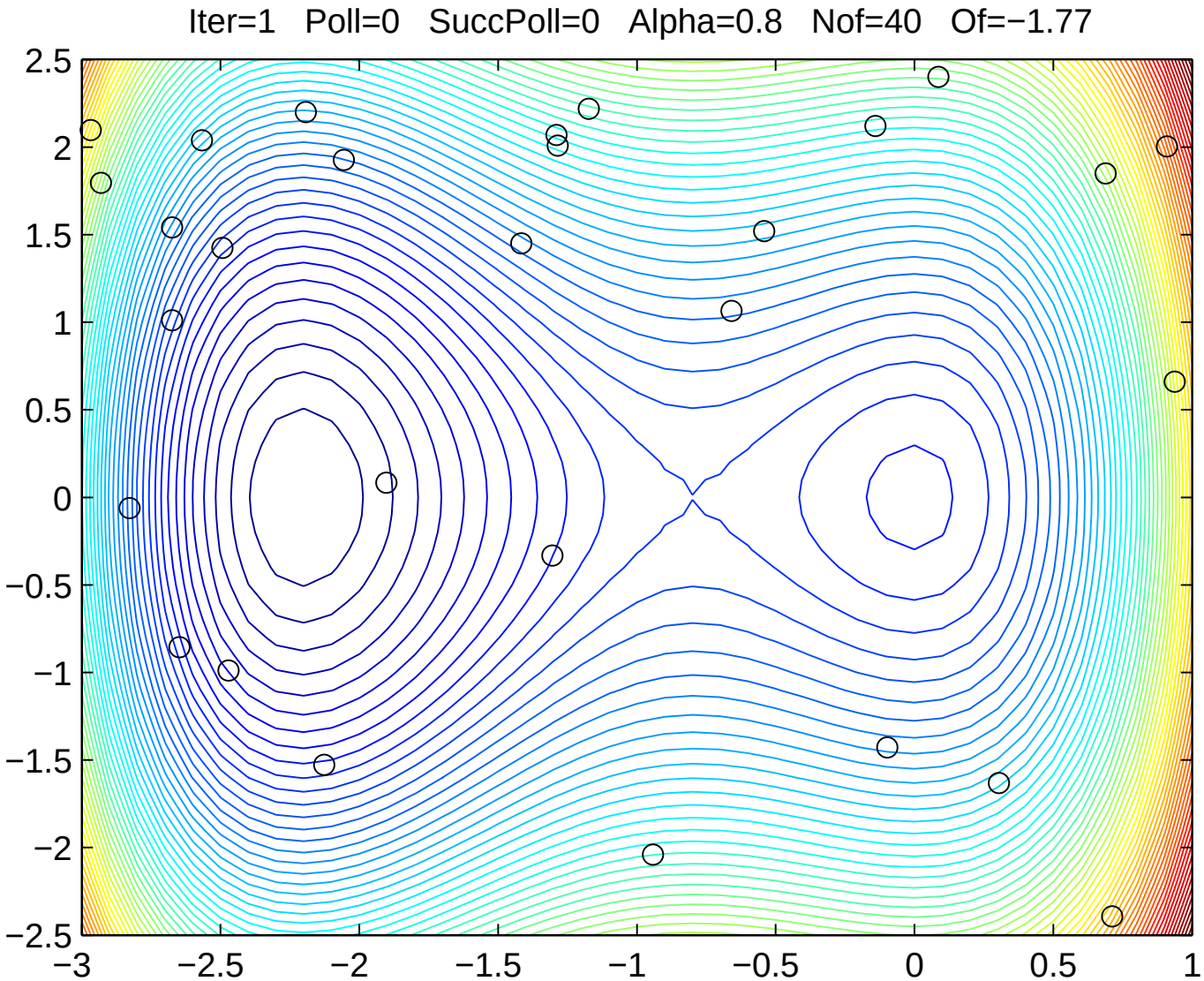
Key points

- In the first iterations the algorithm takes advantage of the particle swarm ability to find a global optimum (exploiting the search space), while in the last iterations the algorithm takes advantage of the pattern search robustness to find a stationary point.
- The number of particles in the swarm search can be decreased along the iterations (no need to have a large number of particles around a local optimum).

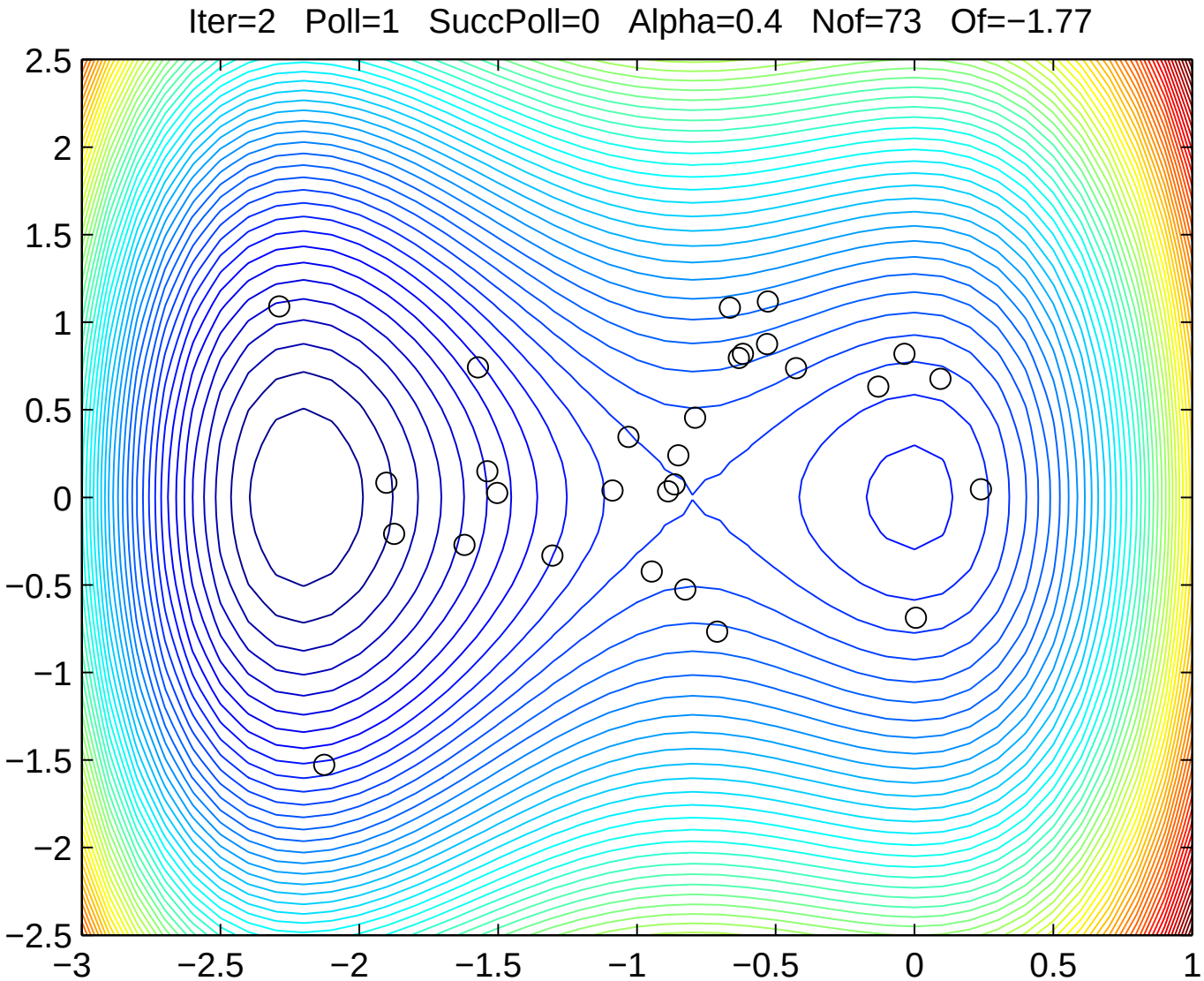
An example - Treccani function



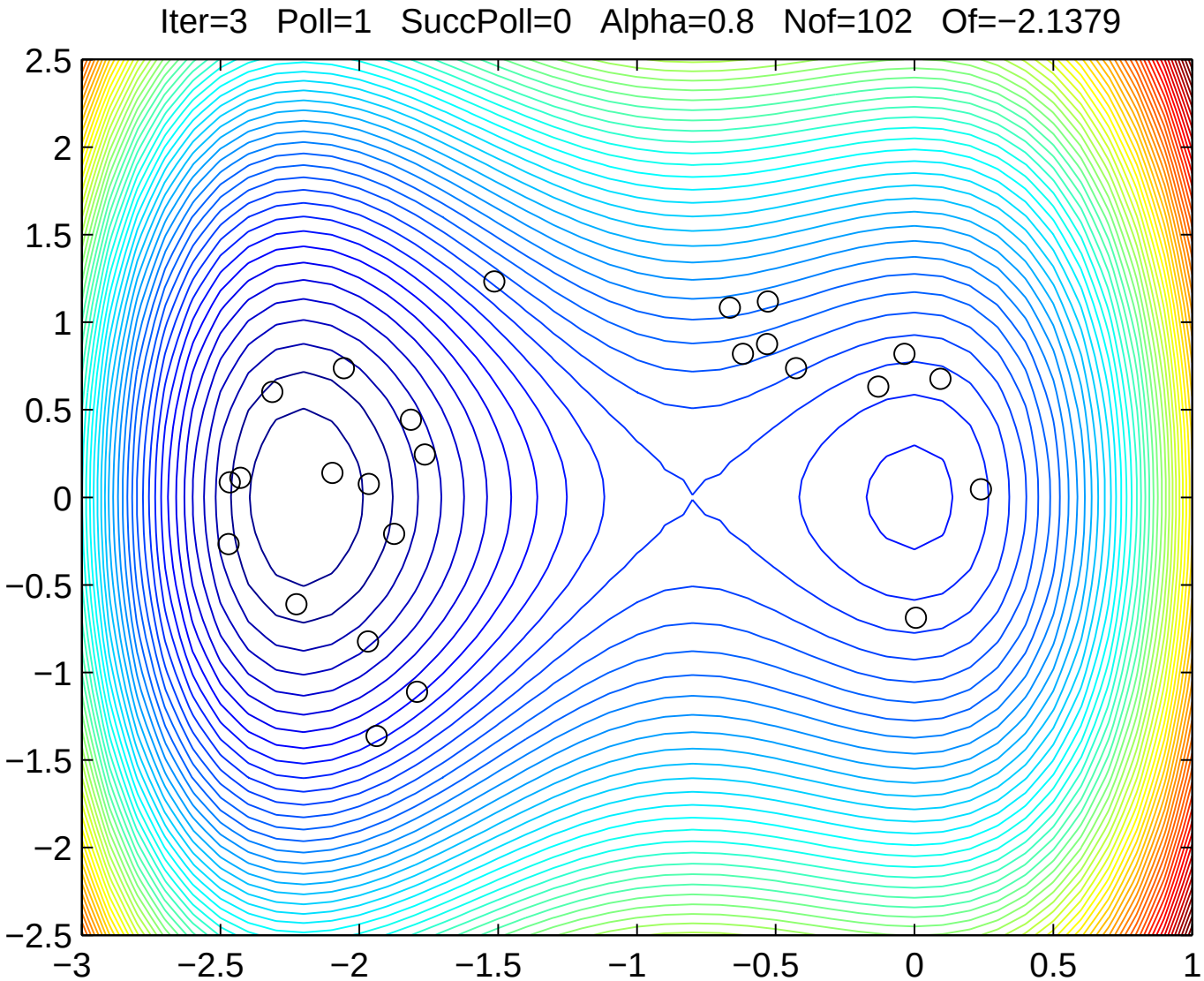
An example - Treccani function



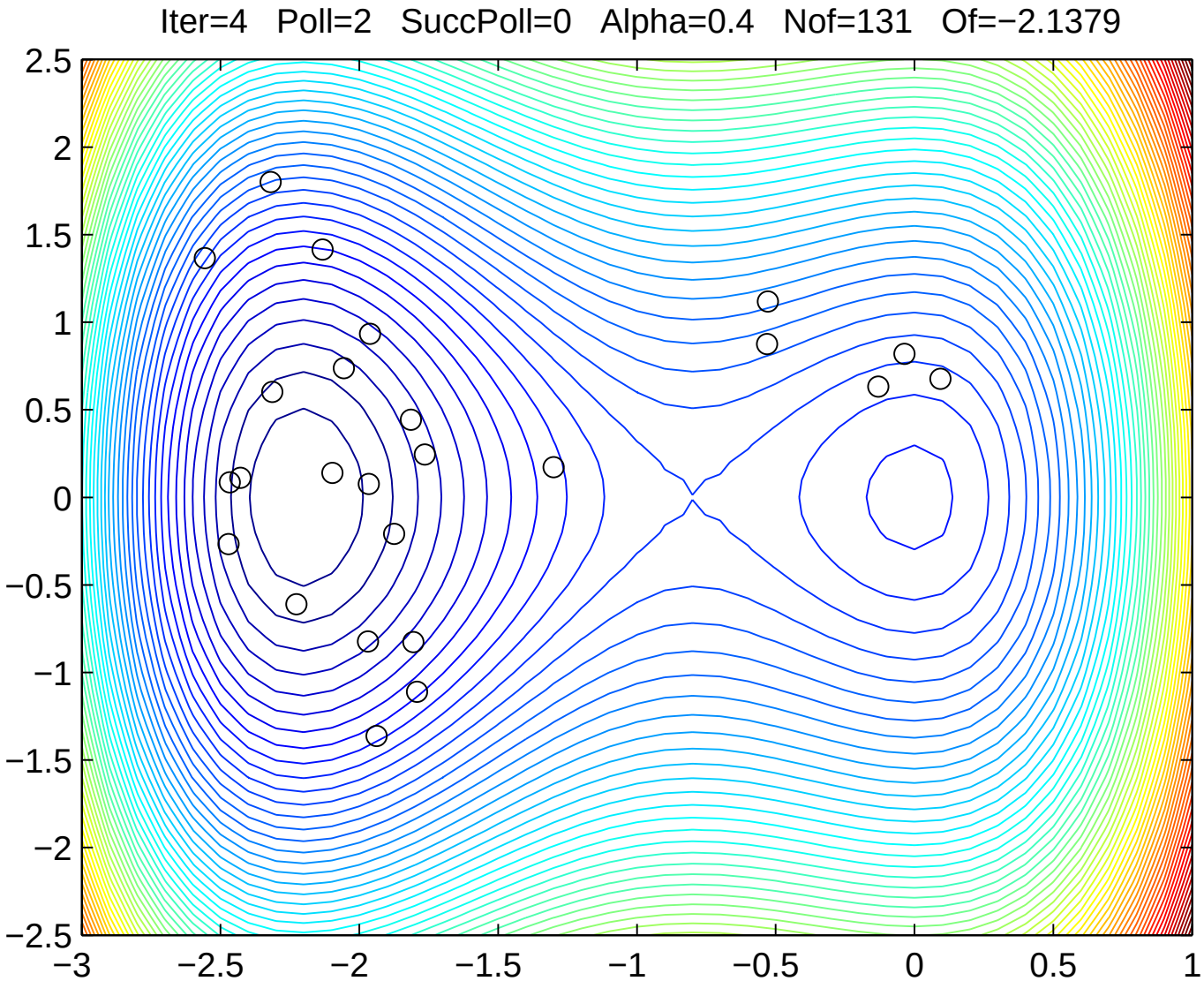
An example - Treccani function



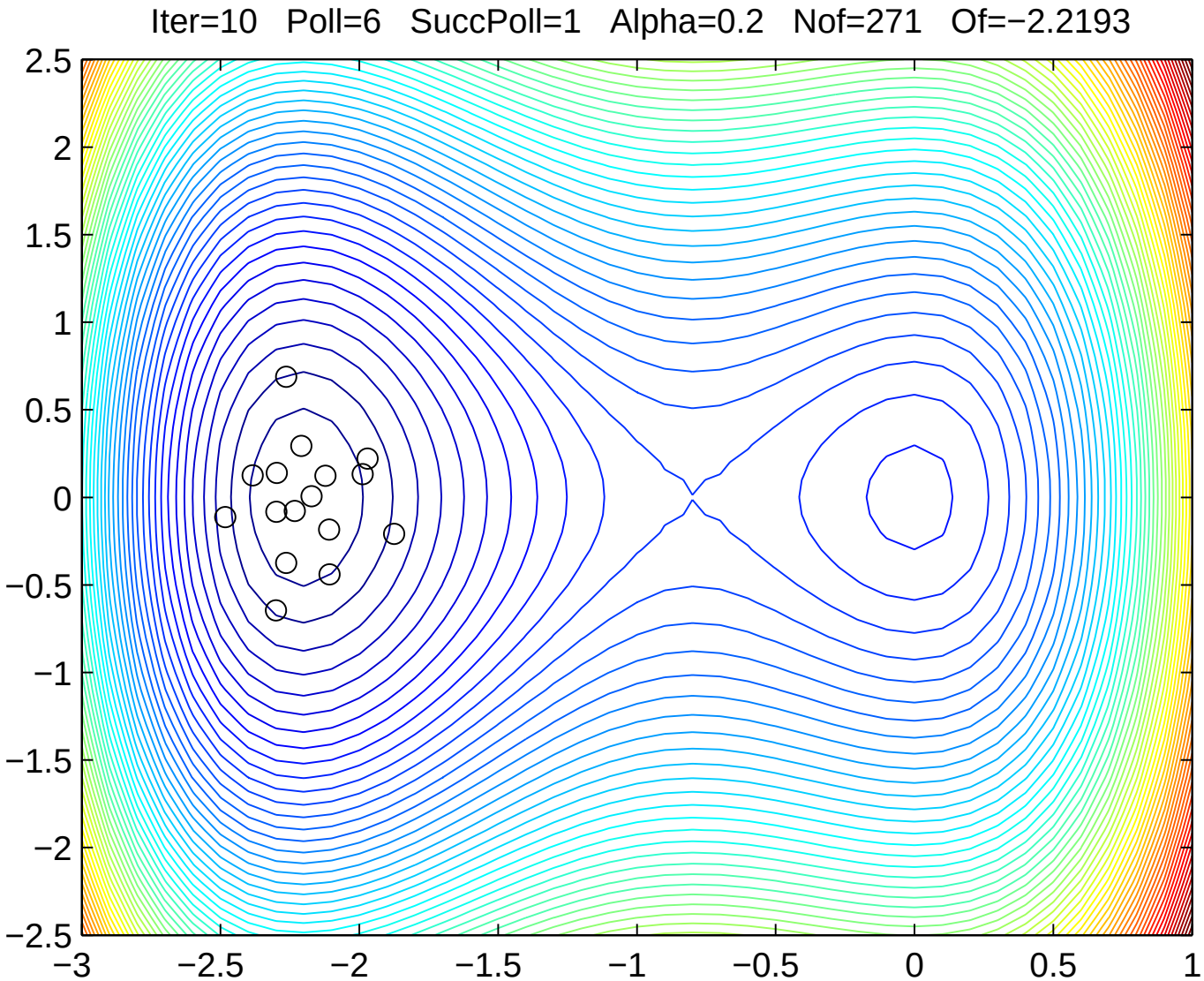
An example - Treccani function



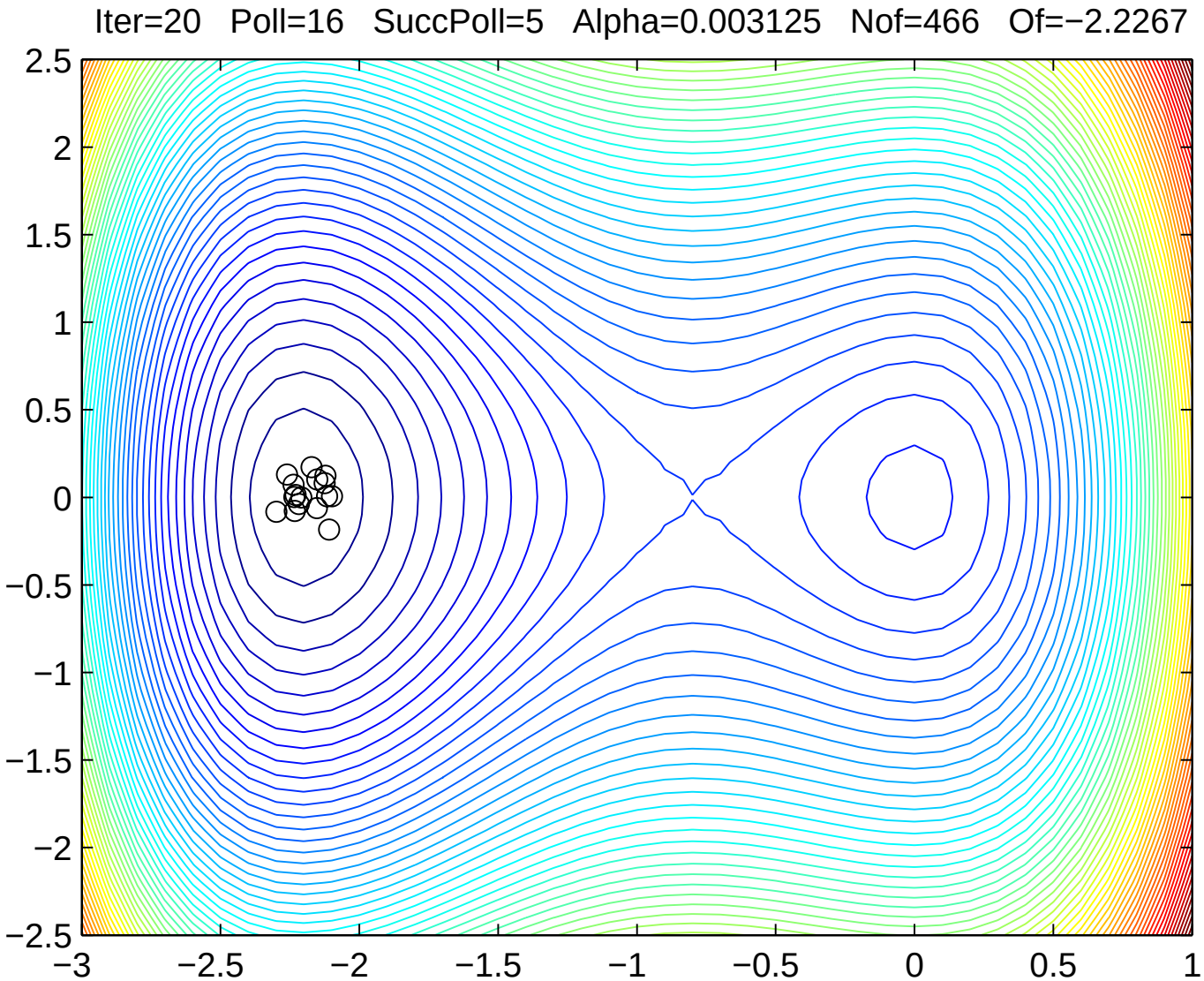
An example - Treccani function



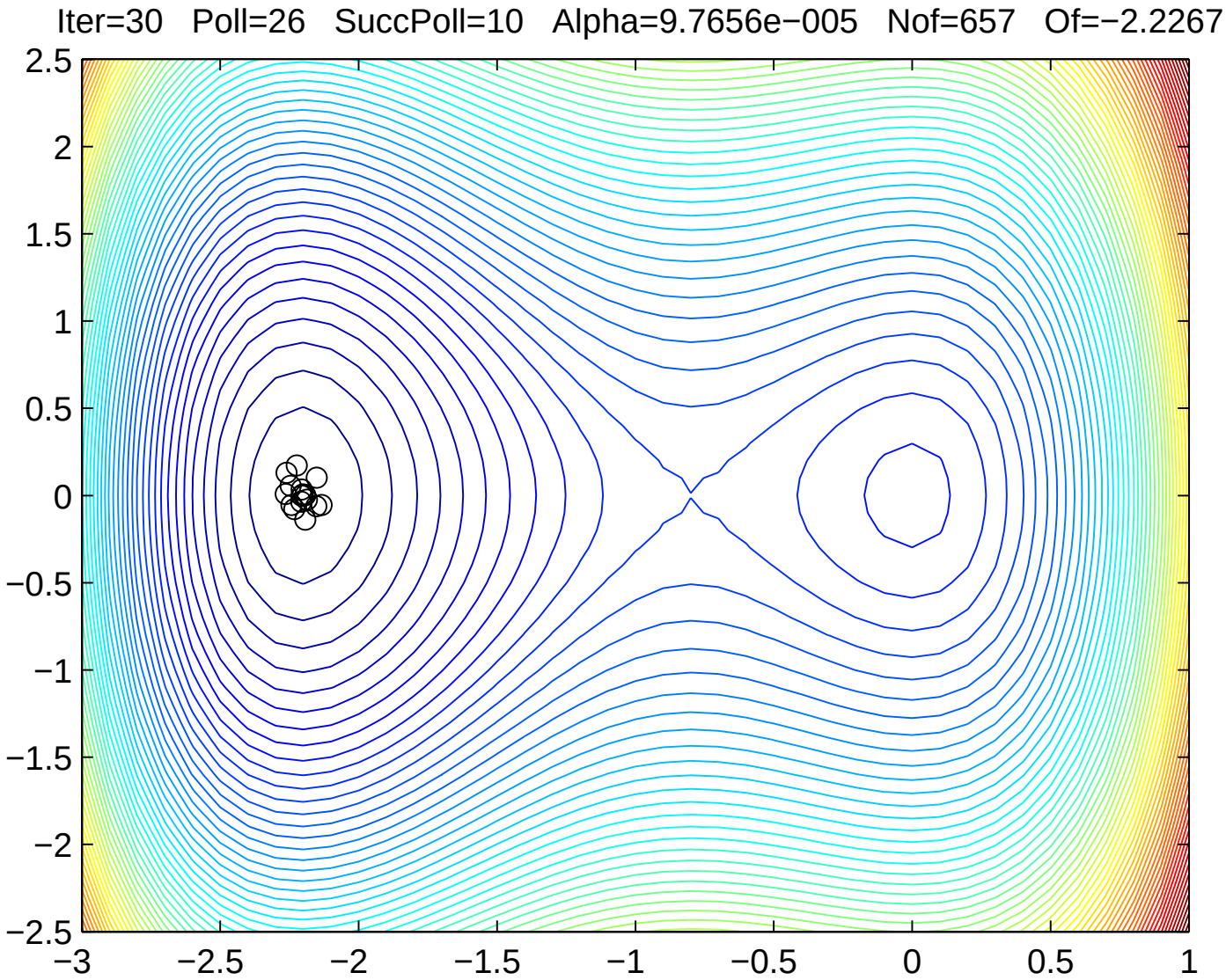
An example - Treccani function



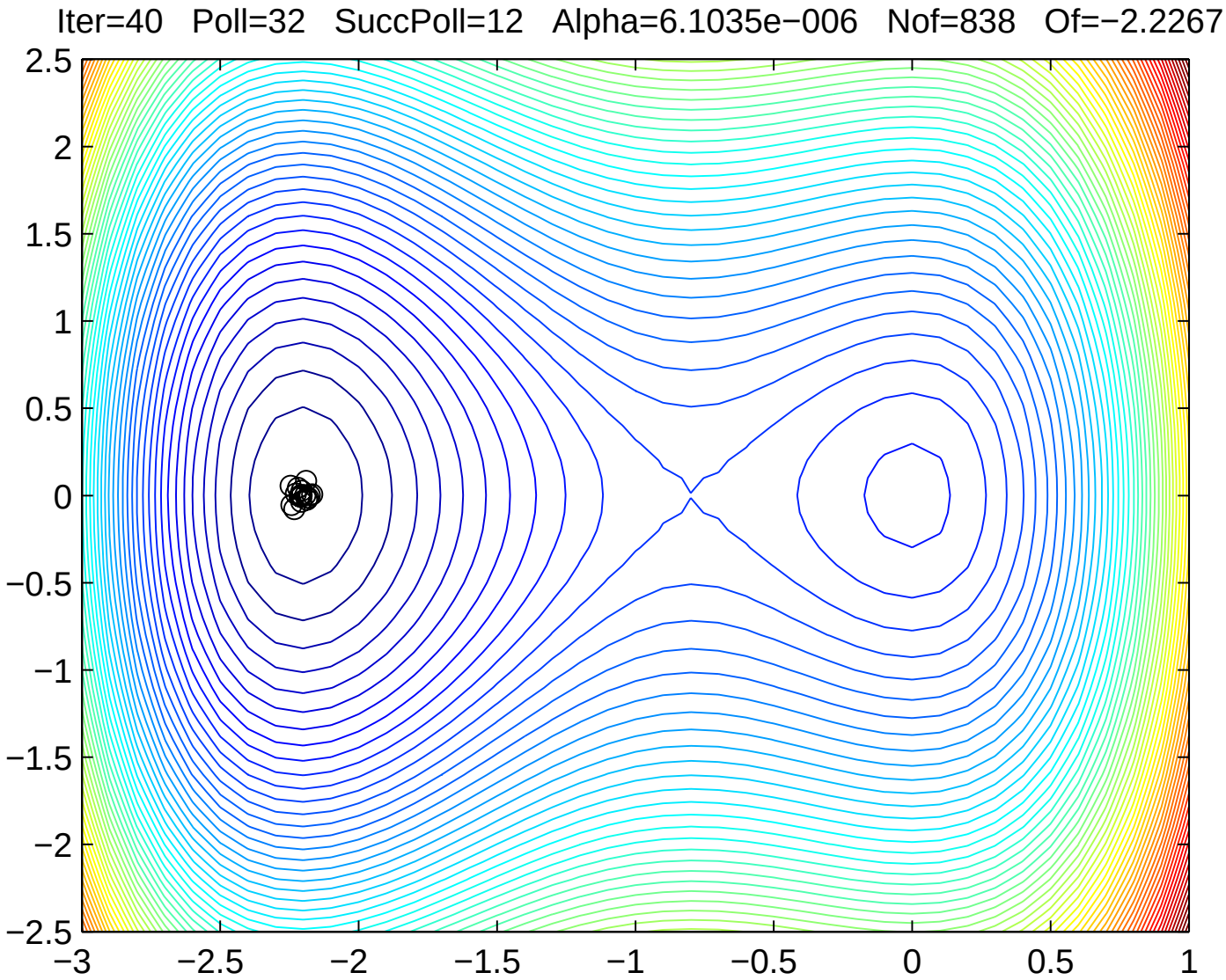
An example - Treccani function



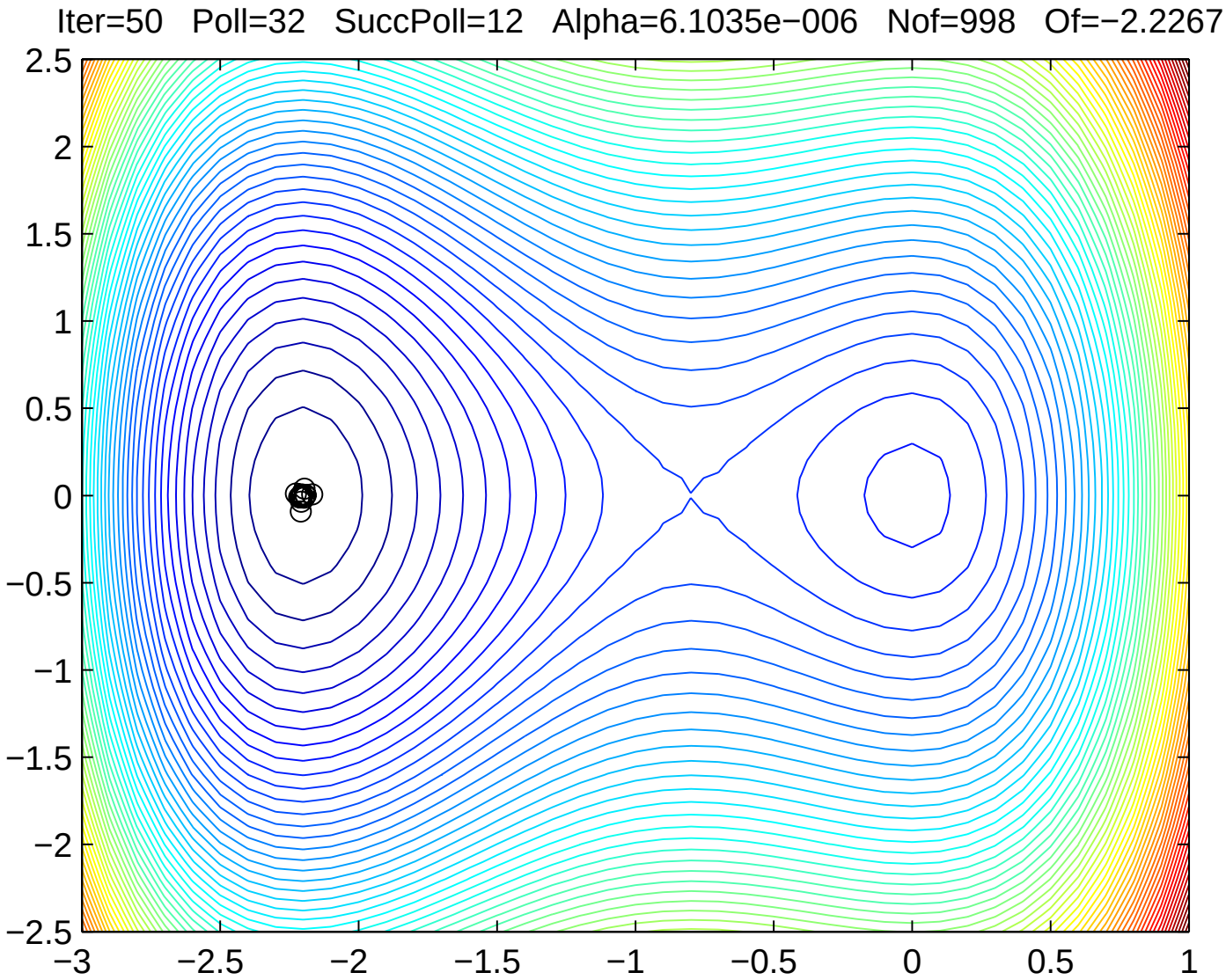
An example - Treccani function



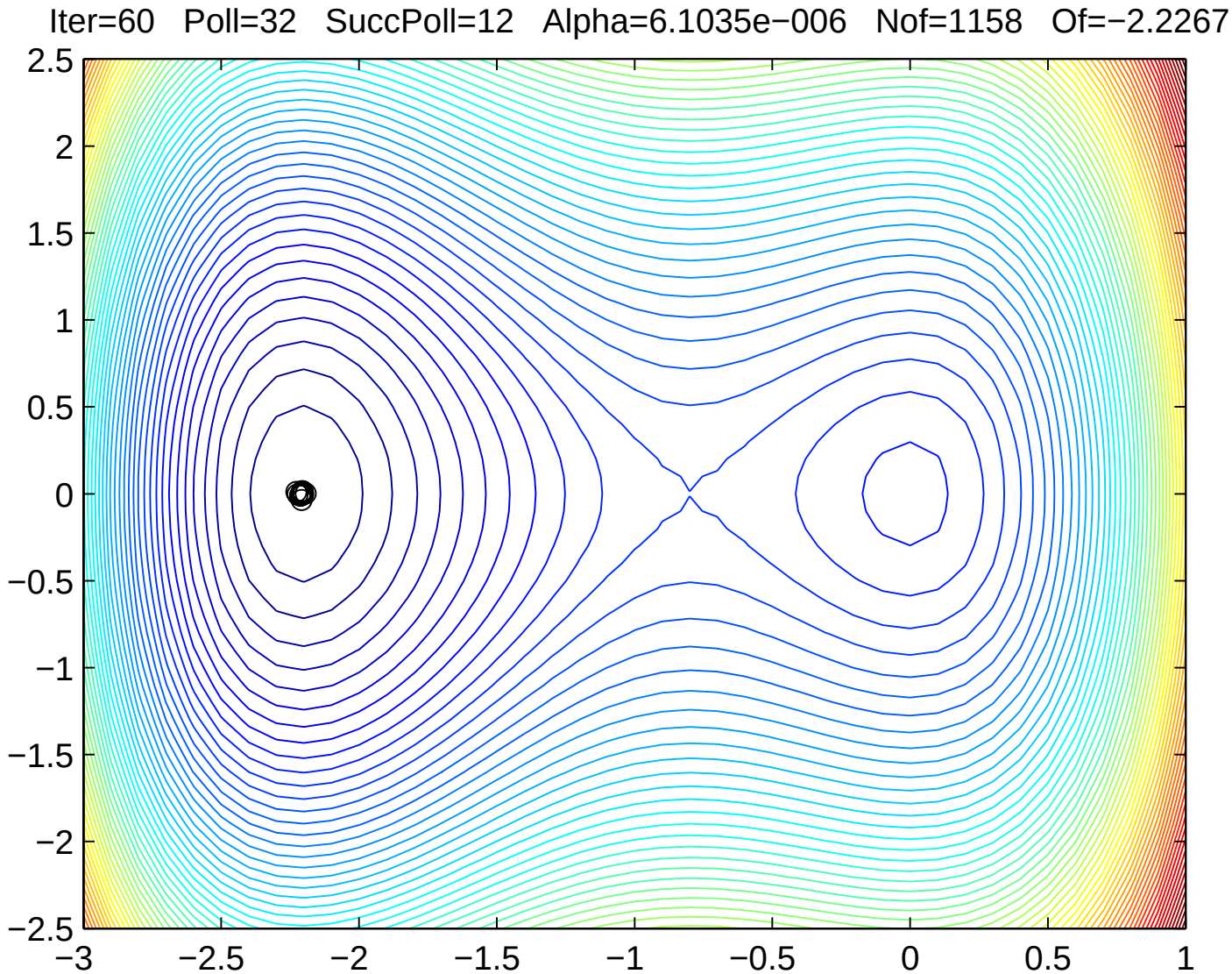
An example - Treccani function



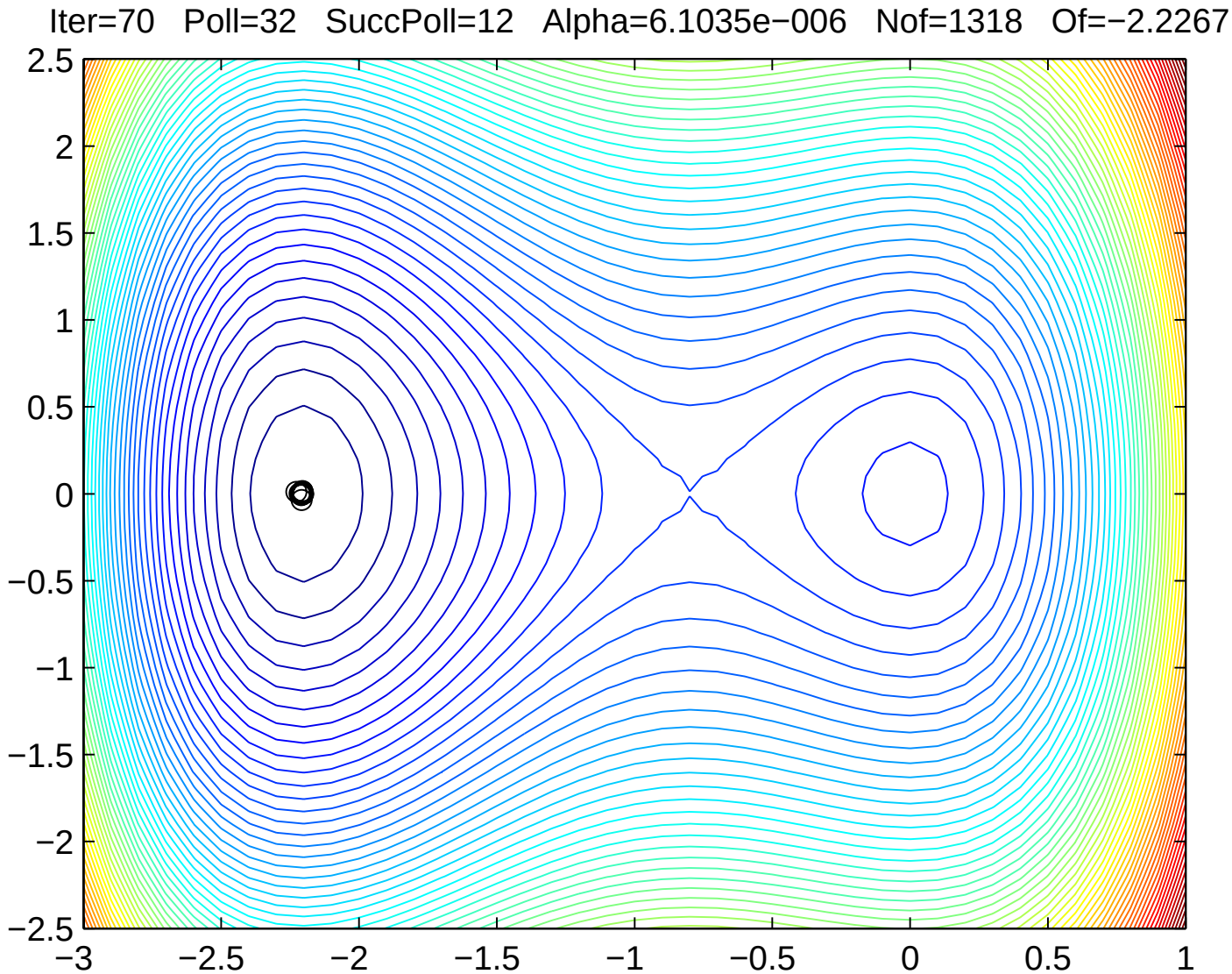
An example - Treccani function



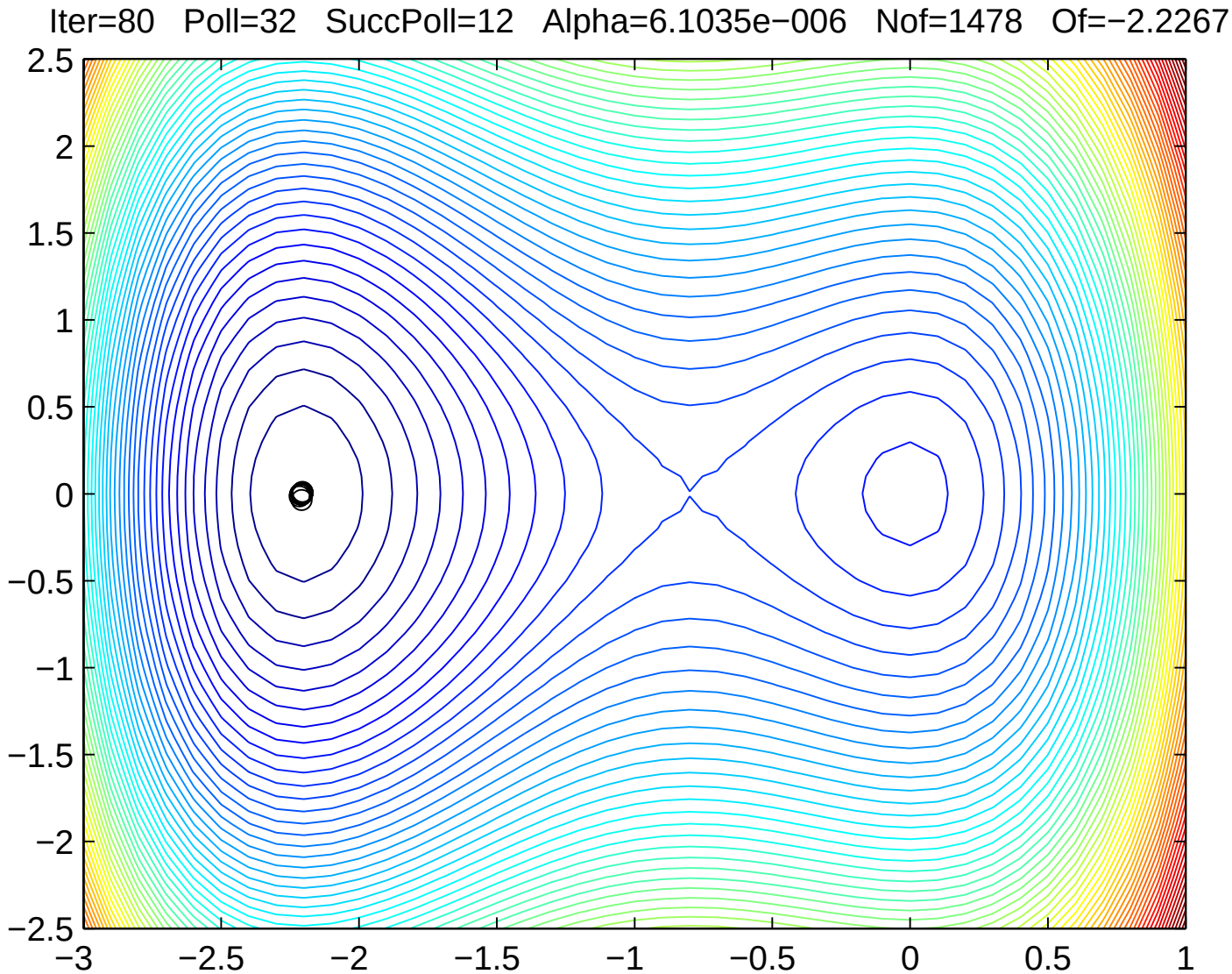
An example - Treccani function



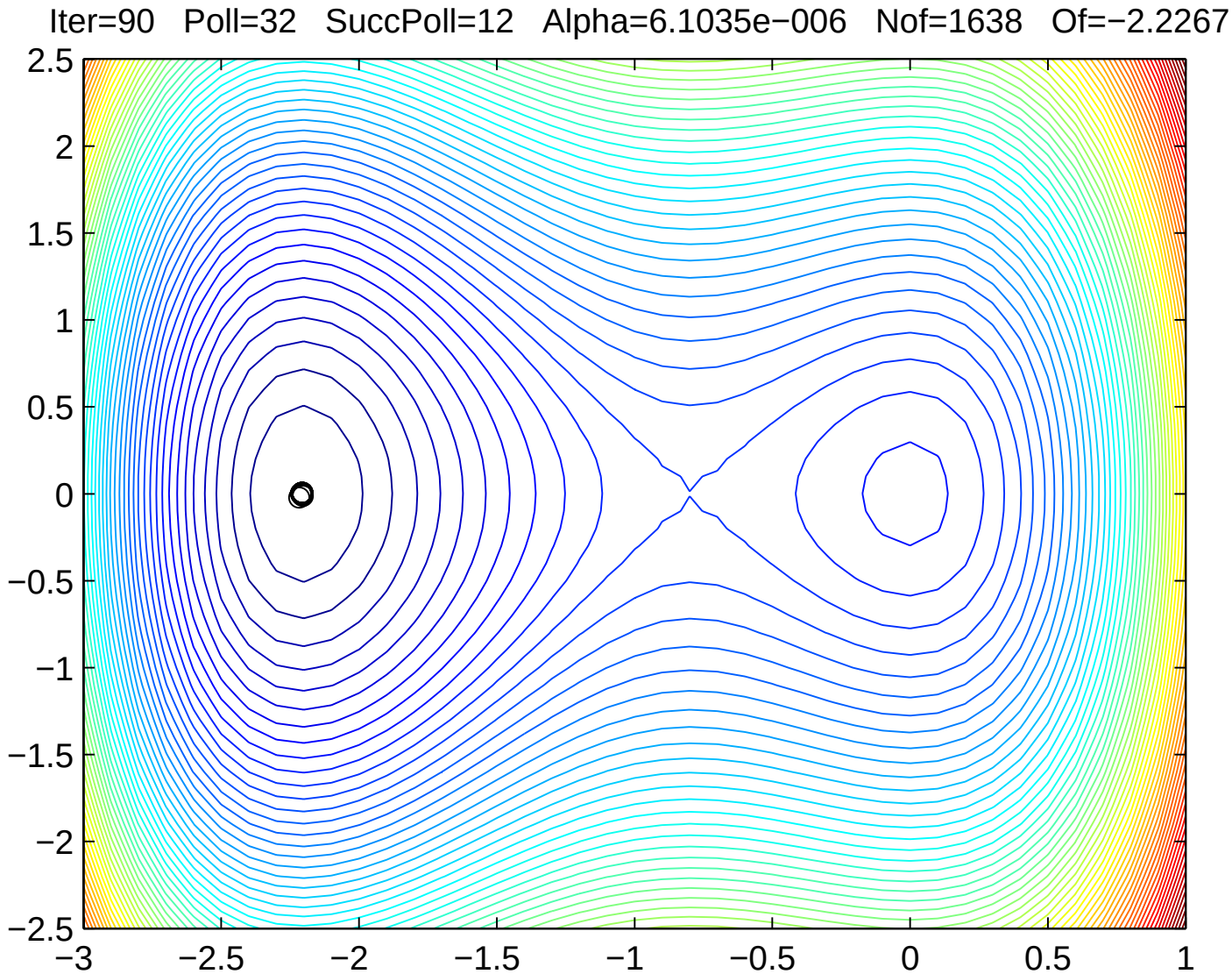
An example - Treccani function



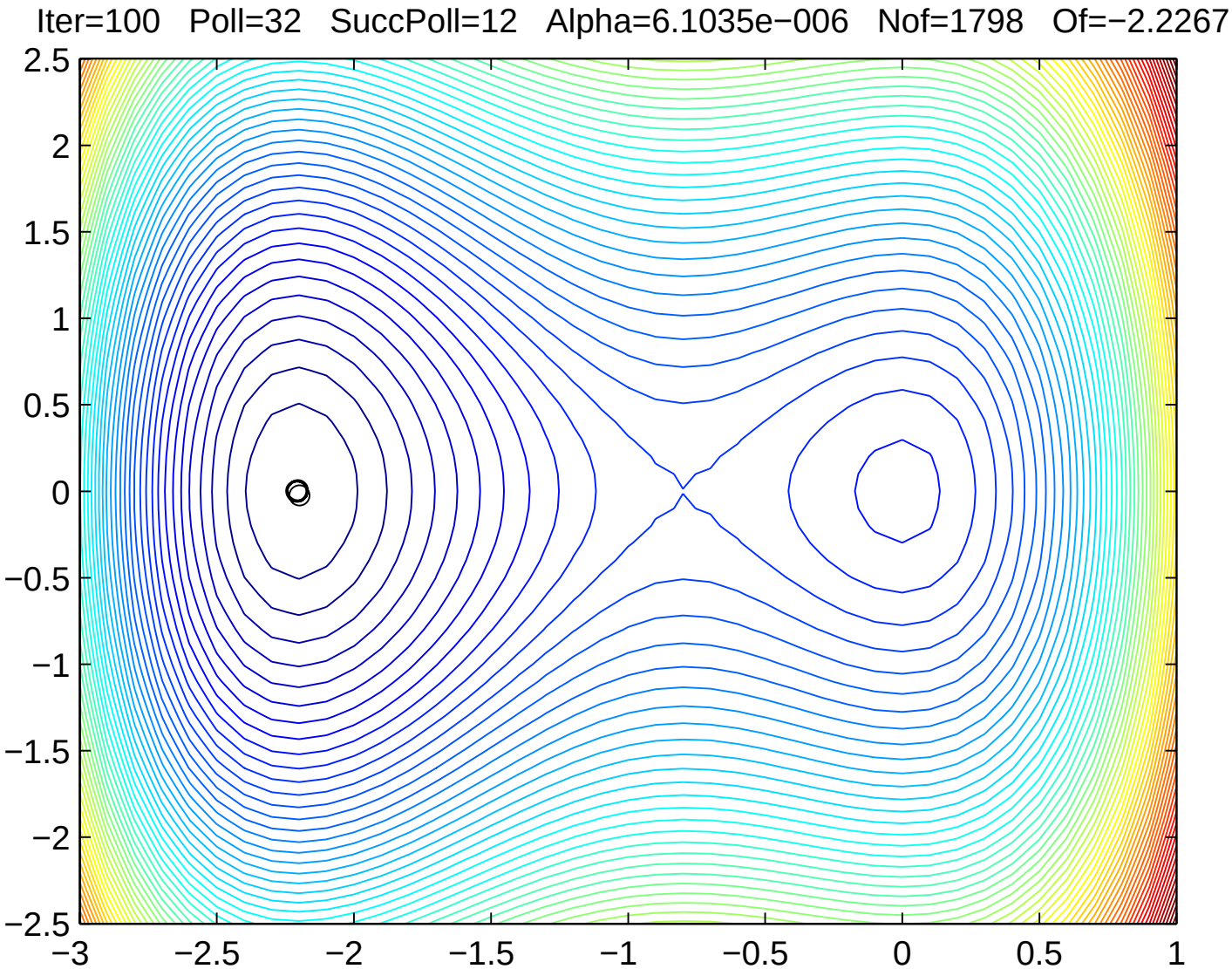
An example - Treccani function



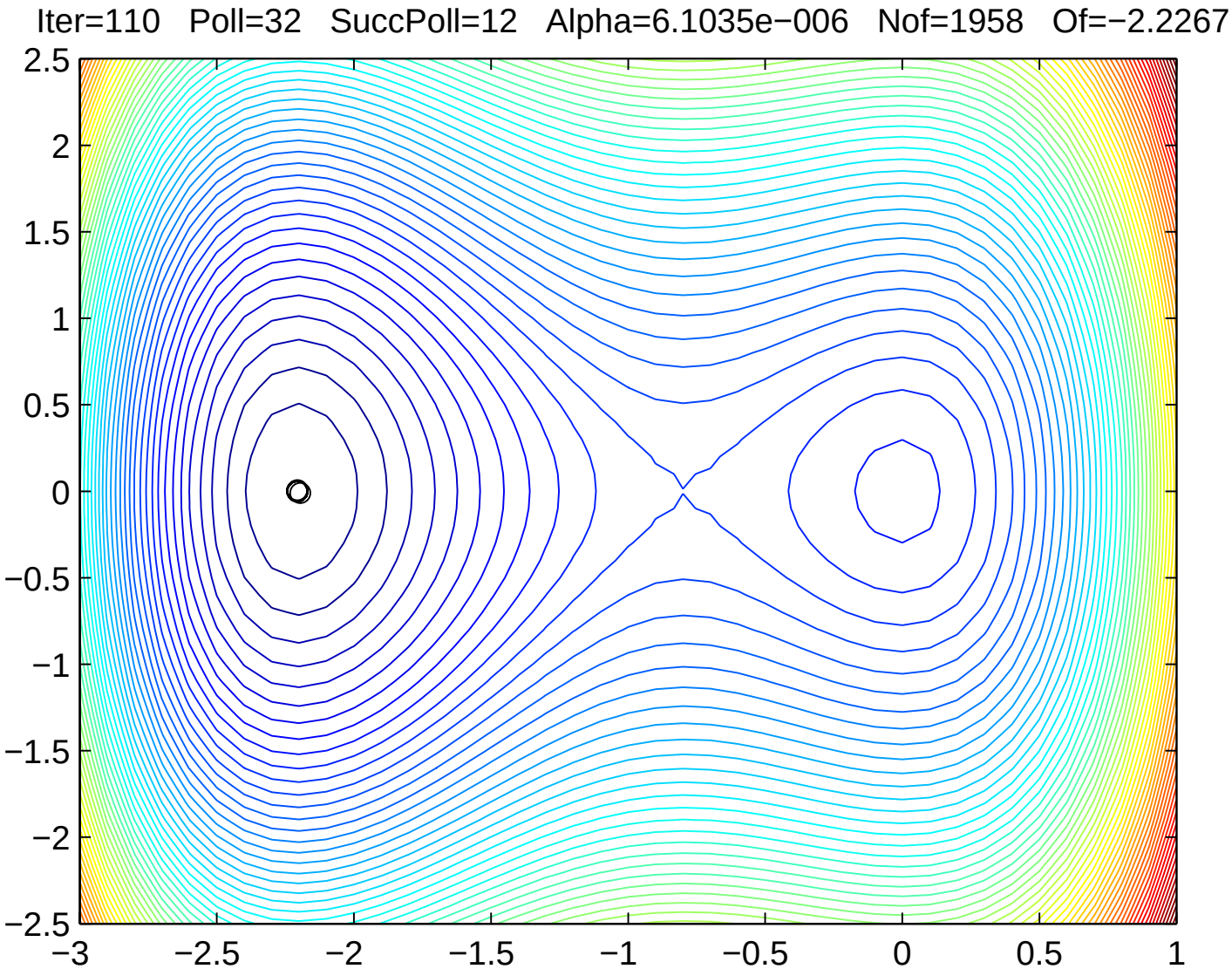
An example - Treccani function



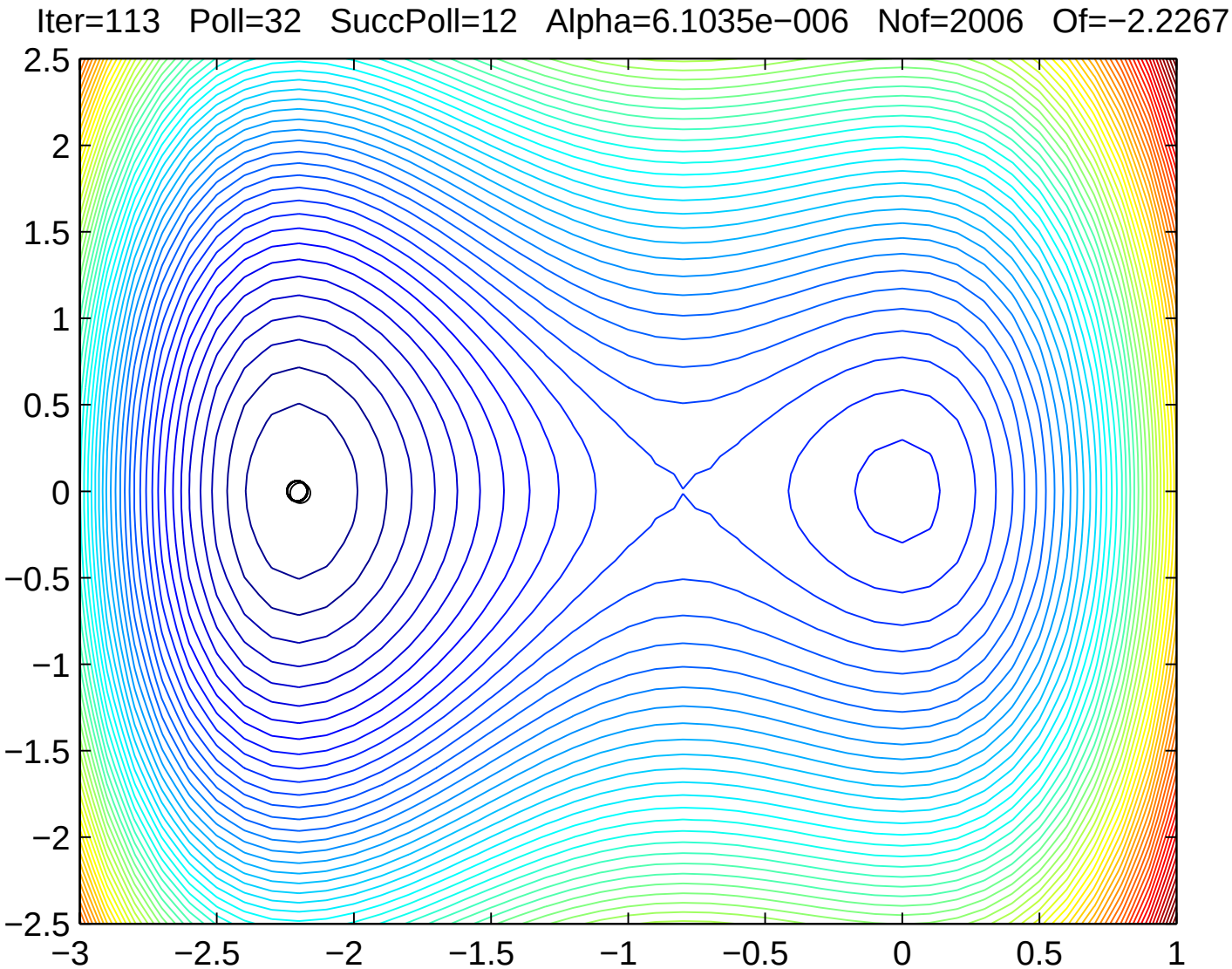
An example - Treccani function



An example - Treccani function

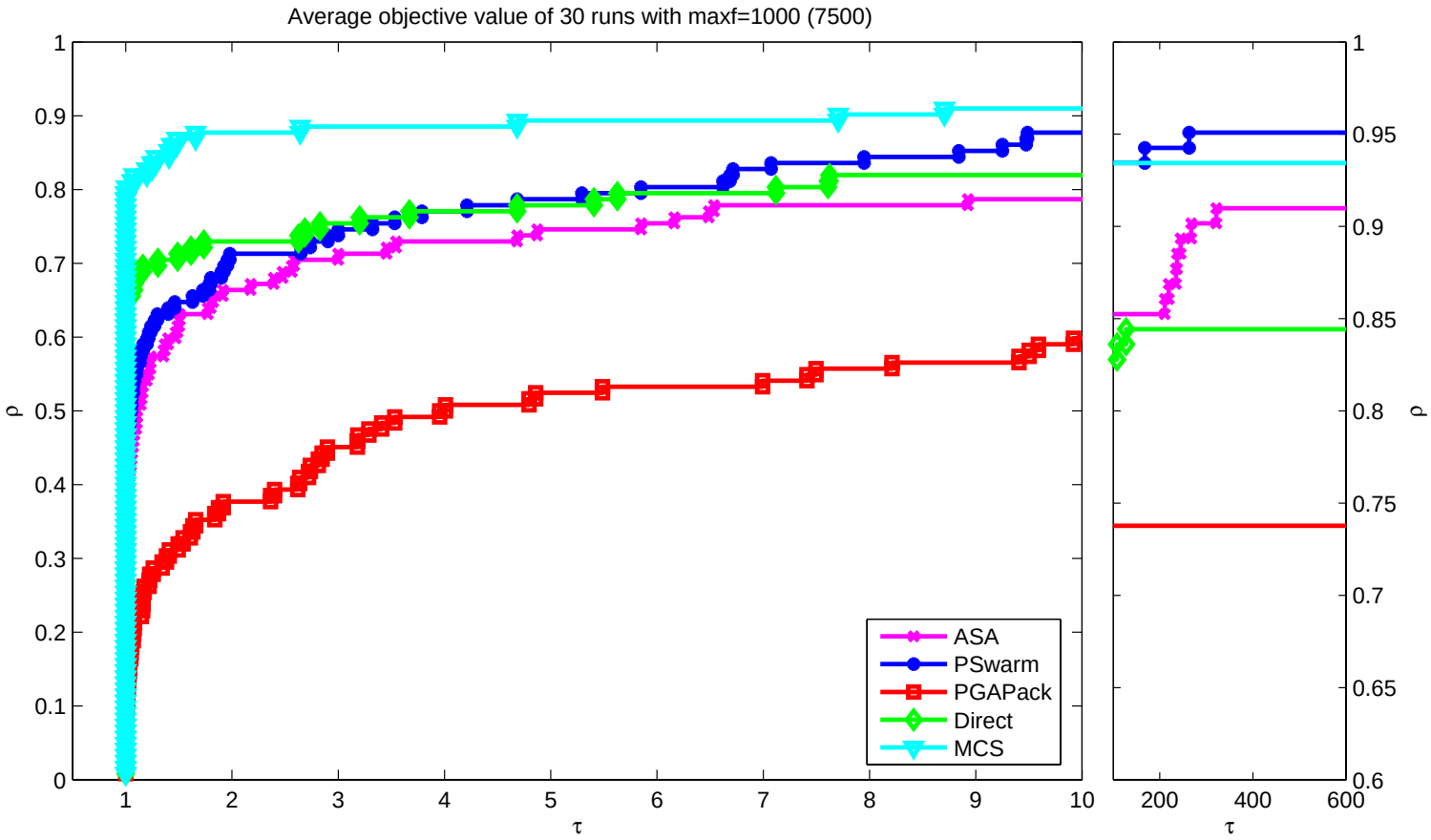


An example - Treccani function



Numerical results (final value for f)

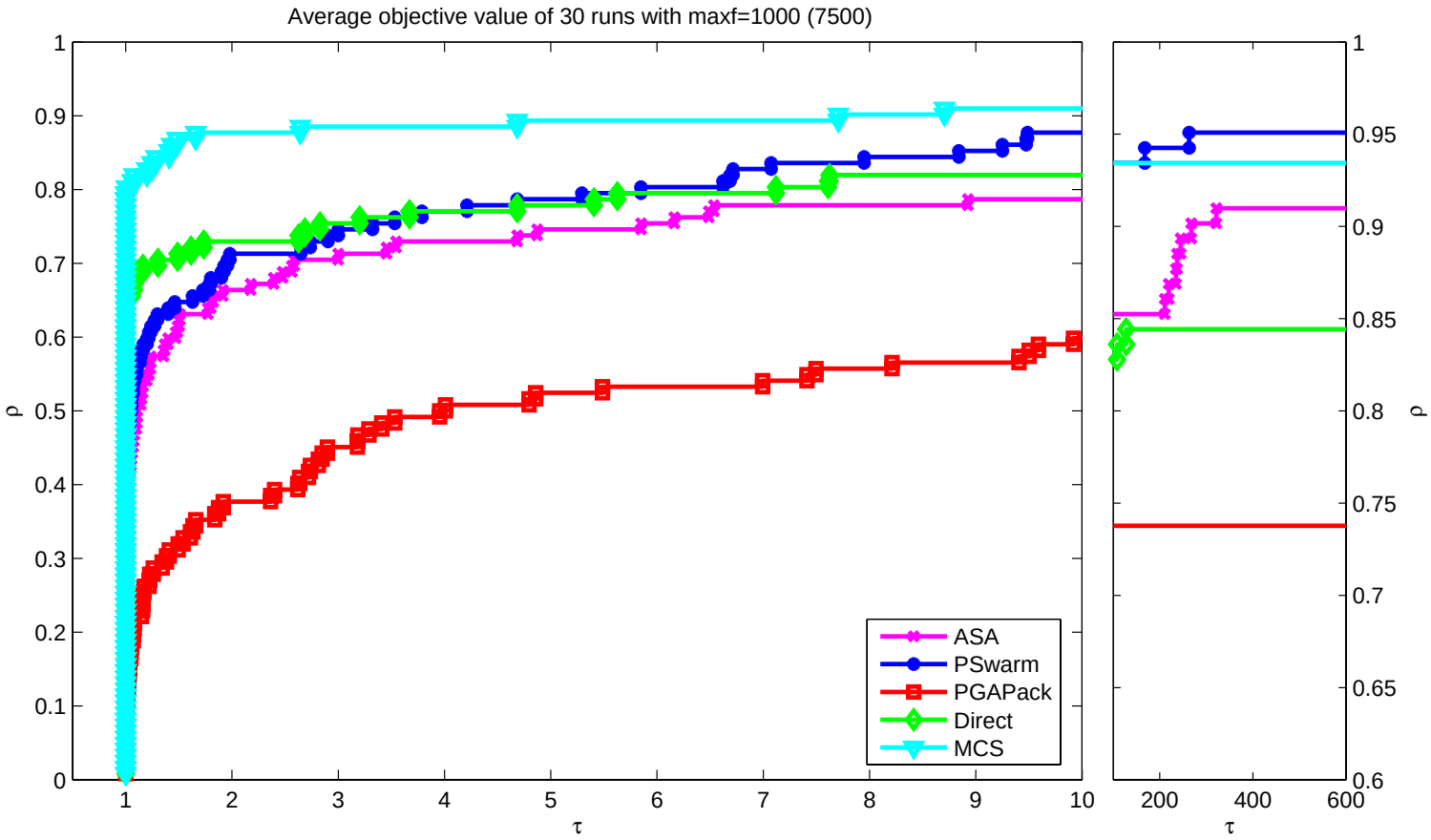
- 122 problems where 12 are of large dimension (100-300 variables).



For further details see Vaz and Vicente, JOGO, 2007

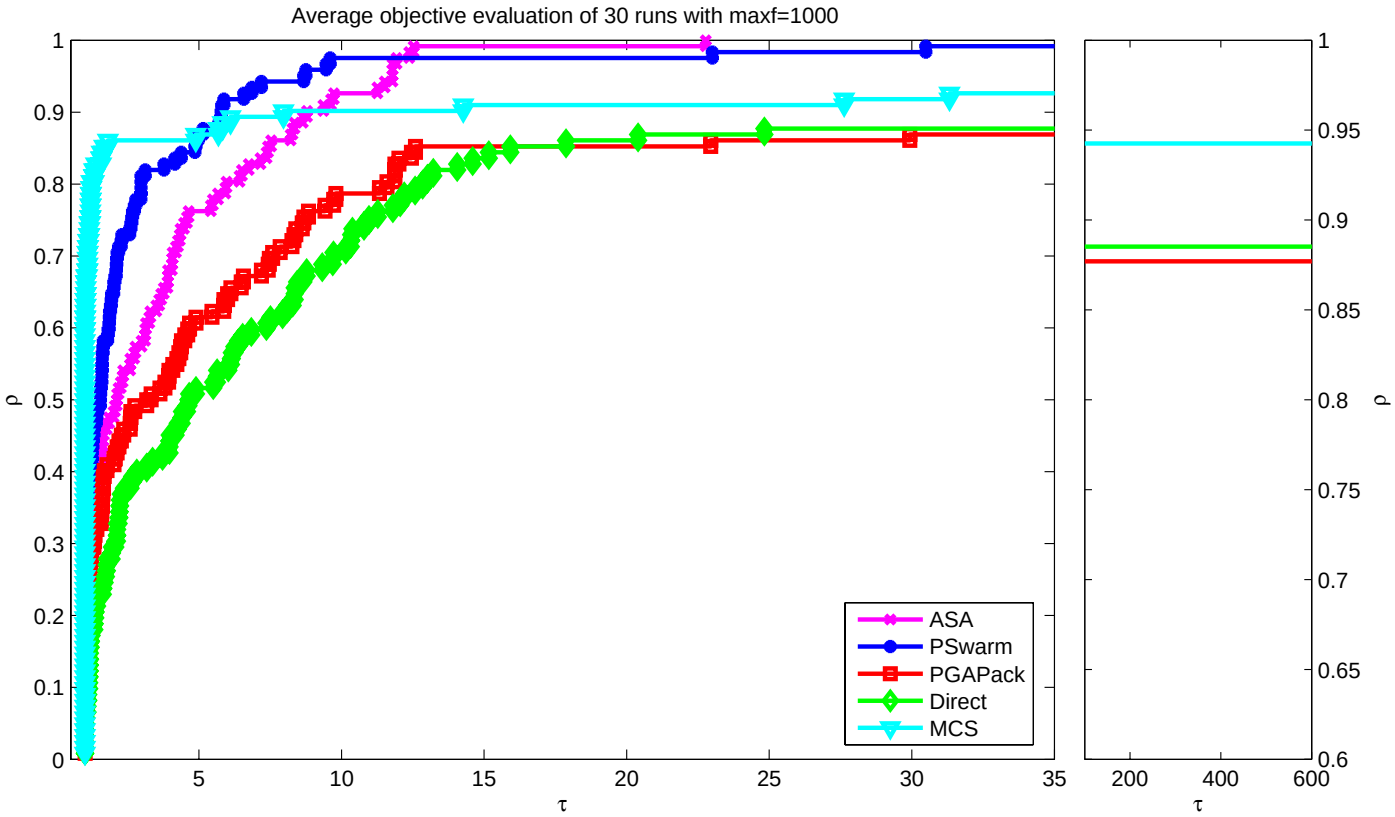
Numerical results (final value for f)

- 122 problems where 12 are of large dimension (100-300 variables).



For further details see Vaz and Vicente, JOGO, 2007

Numerical results (number of evaluations)

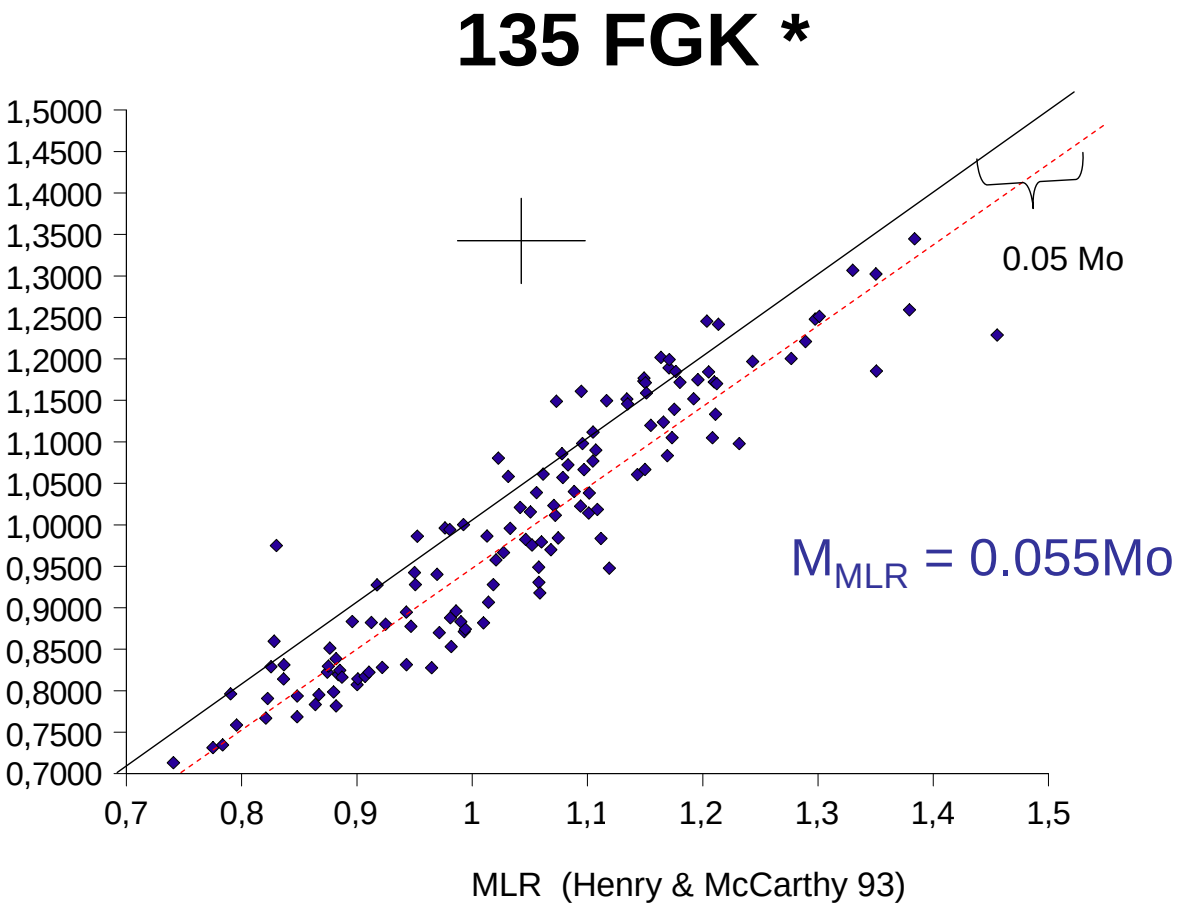


Average number of objective function evaluations.

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

Parameter estimation in Astrophysics

The goal is to determine a set of six stellar parameters from observable data. The objective function requires simulation (CESAM code). PSwarm was very successful on a set of 135 stars ($193 \times 2000 \times 25 = 18.40$ years computational time in parallel).



* Stars belonging to spectral types F, G, or K.

Mass-luminosity relation (MLR)

Joint work with J.M. Fernandes - University of Coimbra.

Outline

Problem formulation

The problem we are now addressing is:

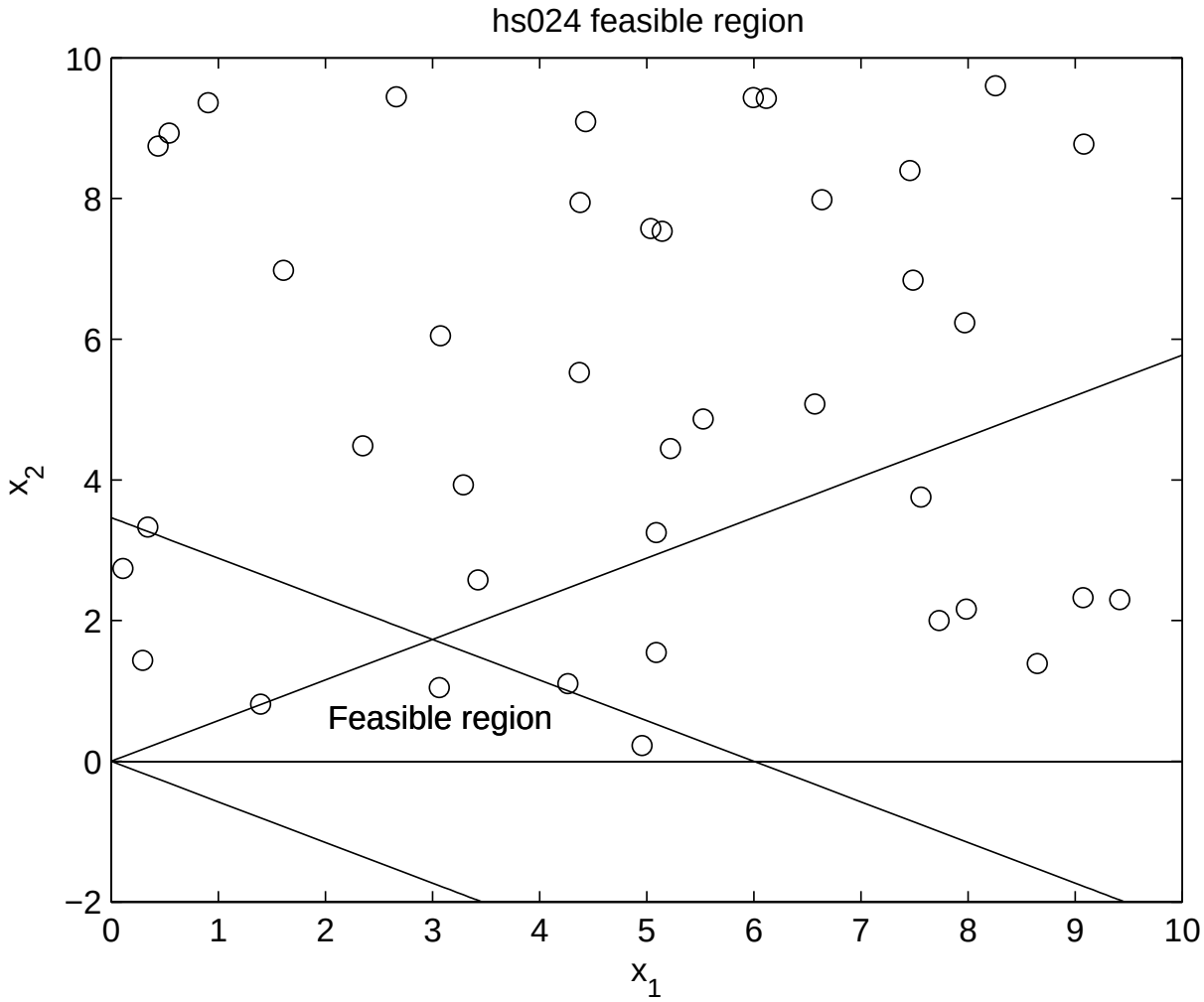
Problem definition - bound and linear constraints

$$\begin{aligned} \min_{z \in \mathbb{R}^n} \quad & f(z) \\ \text{s.t.} \quad & Az \leq b, \\ & \ell \leq z \leq u, \end{aligned}$$

where A is a $m \times n$ matrix, b is a m column vector and $\ell \leq z \leq u$ are understood componentwise.

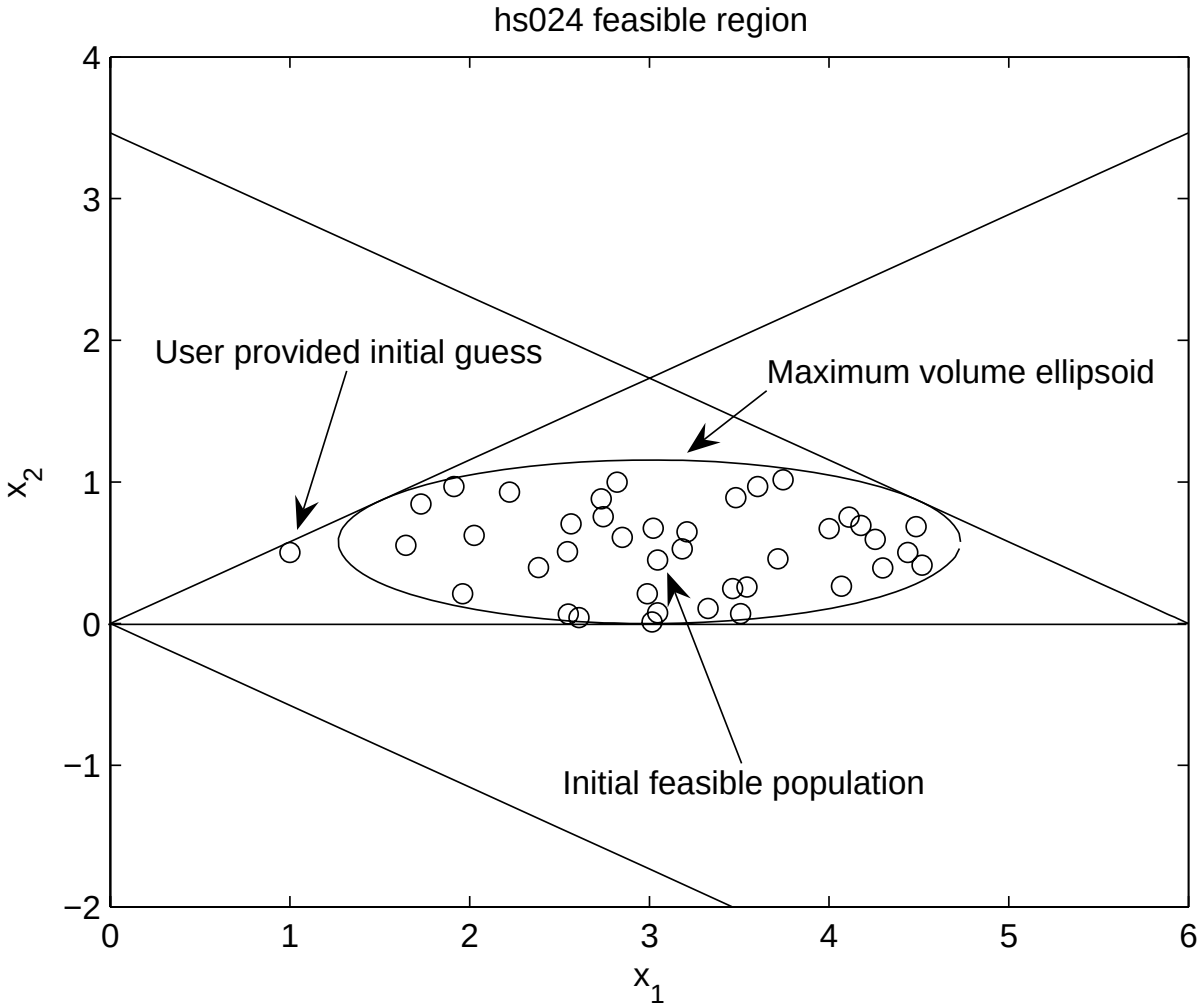
Feasible initial population

Obtaining an initial feasible population and controlling feasibility in the linear constrained case is critical.



Feasible initial population

Getting an initial feasible population allows a more efficient search for the global optimum.



Zhang and Gao interior-point code is being used to compute the maximum volume ellipsoid.

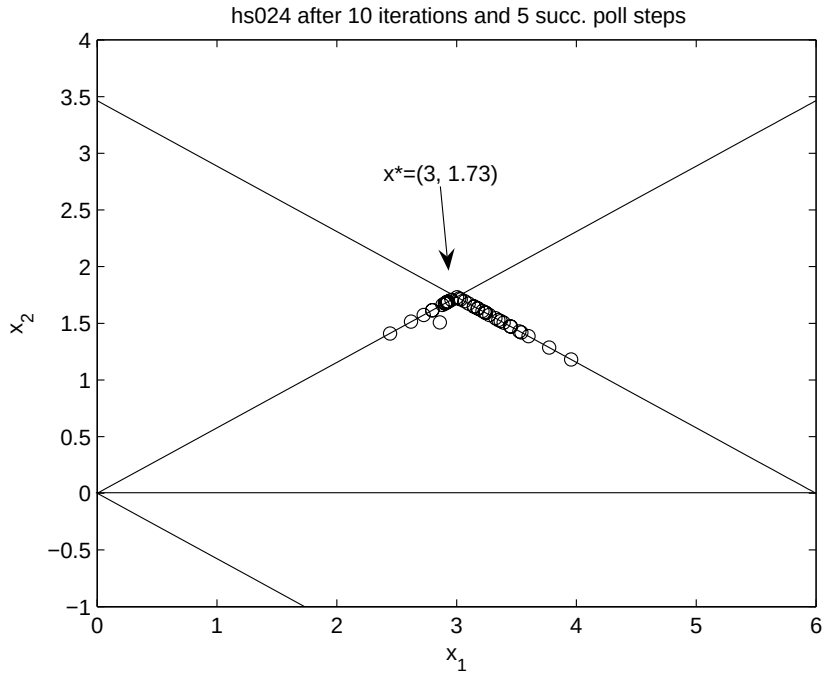
Search step (Particle Swarm)

Feasibility is kept during the optimization process for all particles. This is achieved by introducing a maximum allowed step in the "search" direction.

Maximum allowed step

$$x^p(t+1) = x^p(t) + \alpha_{max} v^p(t+1),$$

where α_{max} is the maximum step allowed to keep $x^p(t+1)$ inside the feasible region.



Poll step

For the coordinate search method applied to bound constrained problems 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.

Penalty/Barrier function

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

Linear constraints

For the case of linear constraints this is no longer true.

Poll step

For the coordinate search method applied to bound constrained problems 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.

Penalty/Barrier function

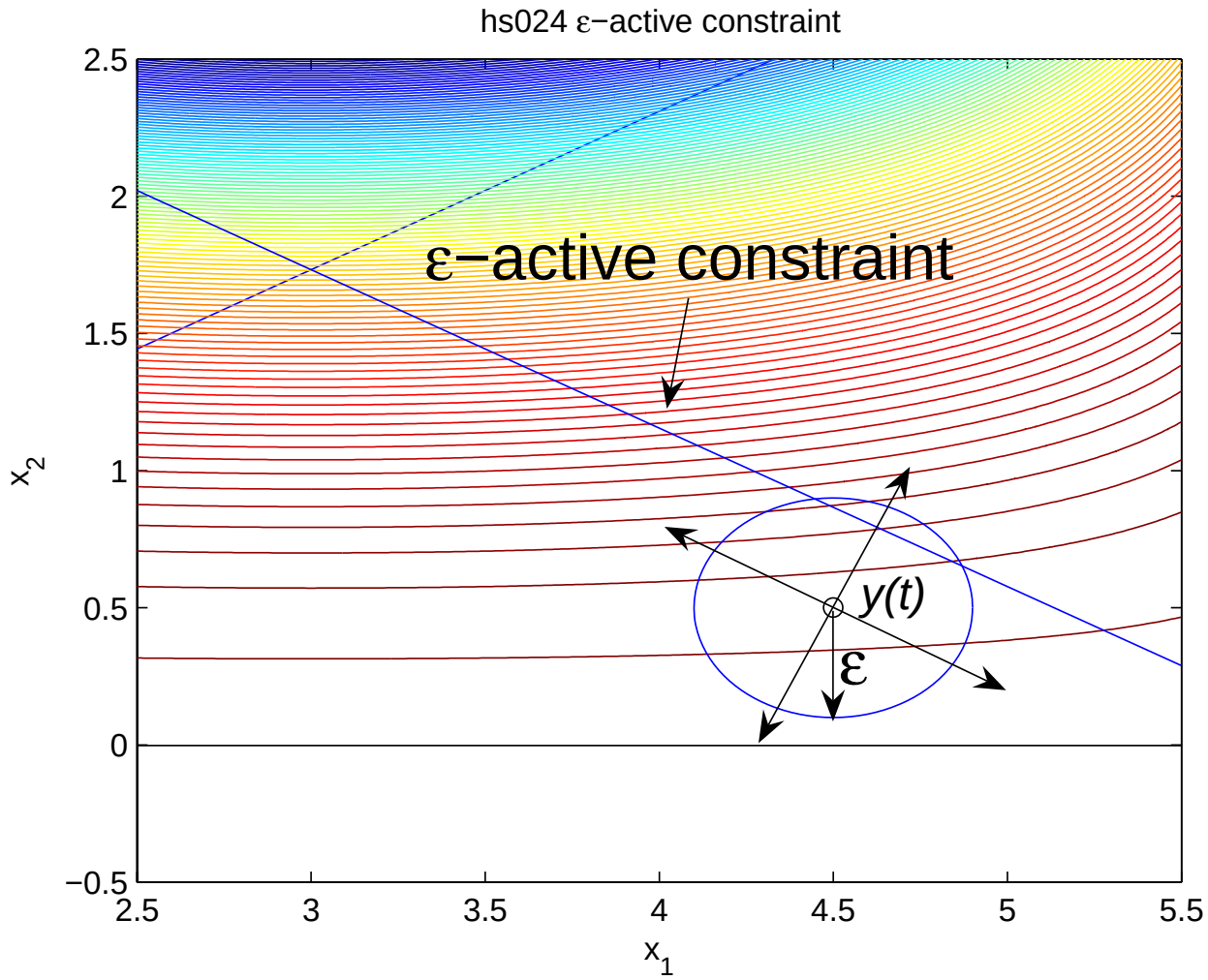
$$\hat{f}(z) = \begin{cases} f(z) & \text{if } z \in \Omega, \\ +\infty & \text{otherwise.} \end{cases}$$

Linear constraints

For the case of linear constraints this is no longer true.

Positive generators for the tangent cone

The set of polling directions needs to conform with the geometry of the feasible set.



Positive generators for the tangent cone

No ϵ-active constraints

The positive spanning set is the maximal positive basis $D_{\oplus}$.

For ϵ-active constraint(s)

The polling directions are the positive generators for the tangent cone of the ϵ-active constraints (obtained by QR factorization)

Degeneracy

The ϵ parameter is dynamically adapted when degeneracy in the ϵ-active constraints is detected. If no success is attained the maximal positive basis is used.

Positive generators for the tangent cone

No ϵ-active constraints

The positive spanning set is the maximal positive basis $D_{\oplus}$.

For ϵ-active constraint(s)

The polling directions are the positive generators for the tangent cone of the ϵ-active constraints (obtained by QR factorization)

Degeneracy

The ϵ parameter is dynamically adapted when degeneracy in the ϵ-active constraints is detected. If no success is attained the maximal positive basis is used.

Positive generators for the tangent cone

No ϵ-active constraints

The positive spanning set is the maximal positive basis $D_{\oplus}$.

For ϵ-active constraint(s)

The polling directions are the positive generators for the tangent cone of the ϵ-active constraints (obtained by QR factorization)

Degeneracy

The ϵ parameter is dynamically adapted when degeneracy in the ϵ-active constraints is detected. If no success is attained the maximal positive basis is used.

Test problems

- 120 problems with linear constraints were collected from 1564 optimization problems (AMPL, CUTE, GAMS, NETLIB, etc.).
- 23 linear, 55 quadratic and 32 general nonlinear.
- 10 highly non-convex objective functions with random generated linear constraints (Pinter).
- The test problems are coded in AMPL (*A Modeling Language for Mathematical Programming*).
- Test problems available at `http://www.norg.uminho.pt/aivaz` (under *software*).

Test problems

- 120 problems with linear constraints were collected from 1564 optimization problems (AMPL, CUTE, GAMS, NETLIB, etc.).
- 23 linear, 55 quadratic and 32 general nonlinear.
- 10 highly non-convex objective functions with random generated linear constraints (Pinter).
- The test problems are coded in AMPL (*A Modeling Language for Mathematical Programming*).
- Test problems available at `http://www.norg.uminho.pt/aivaz` (under *software*).

Test problems

- 120 problems with linear constraints were collected from 1564 optimization problems (AMPL, CUTE, GAMS, NETLIB, etc.).
- 23 linear, 55 quadratic and 32 general nonlinear.
- 10 highly non-convex objective functions with random generated linear constraints (Pinter).
- The test problems are coded in AMPL (*A Modeling Language for Mathematical Programming*).
- Test problems available at `http://www.norg.uminho.pt/aivaz` (under *software*).

Test problems

- 120 problems with linear constraints were collected from 1564 optimization problems (AMPL, CUTE, GAMS, NETLIB, etc.).
- 23 linear, 55 quadratic and 32 general nonlinear.
- 10 highly non-convex objective functions with random generated linear constraints (Pinter).

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

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

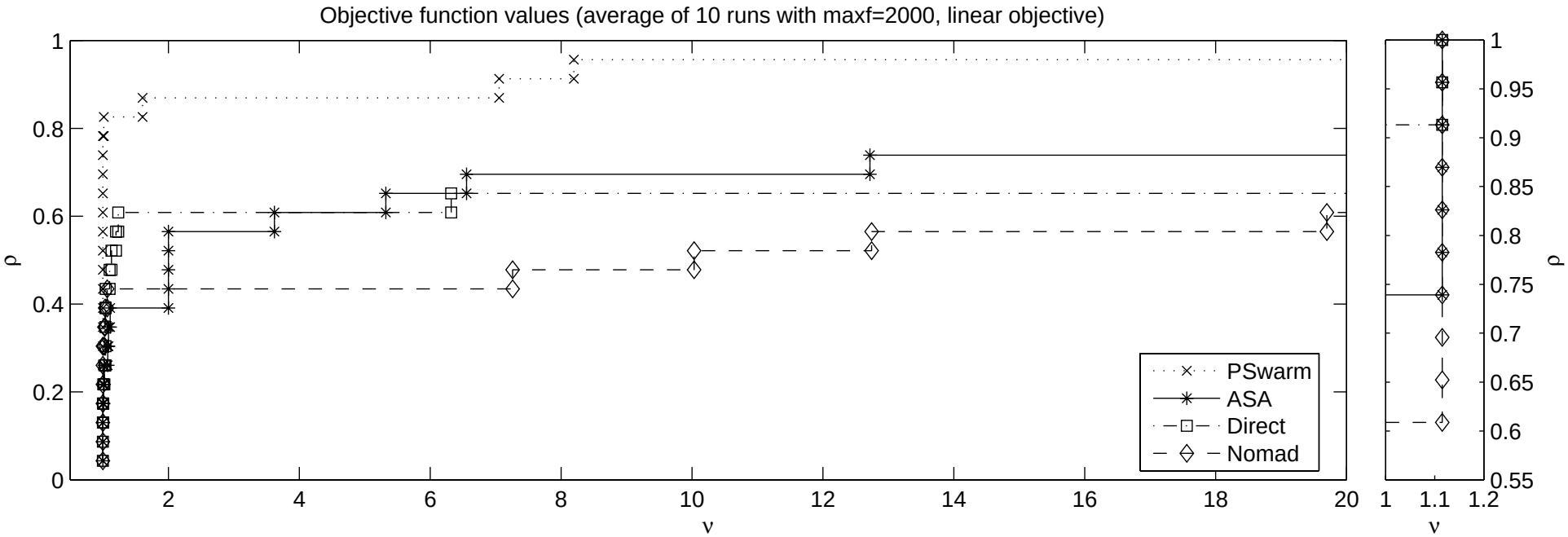
Test problems

- 120 problems with linear constraints were collected from 1564 optimization problems (AMPL, CUTE, GAMS, NETLIB, etc.).
- 23 linear, 55 quadratic and 32 general nonlinear.
- 10 highly non-convex objective functions with random generated linear constraints (Pinter).

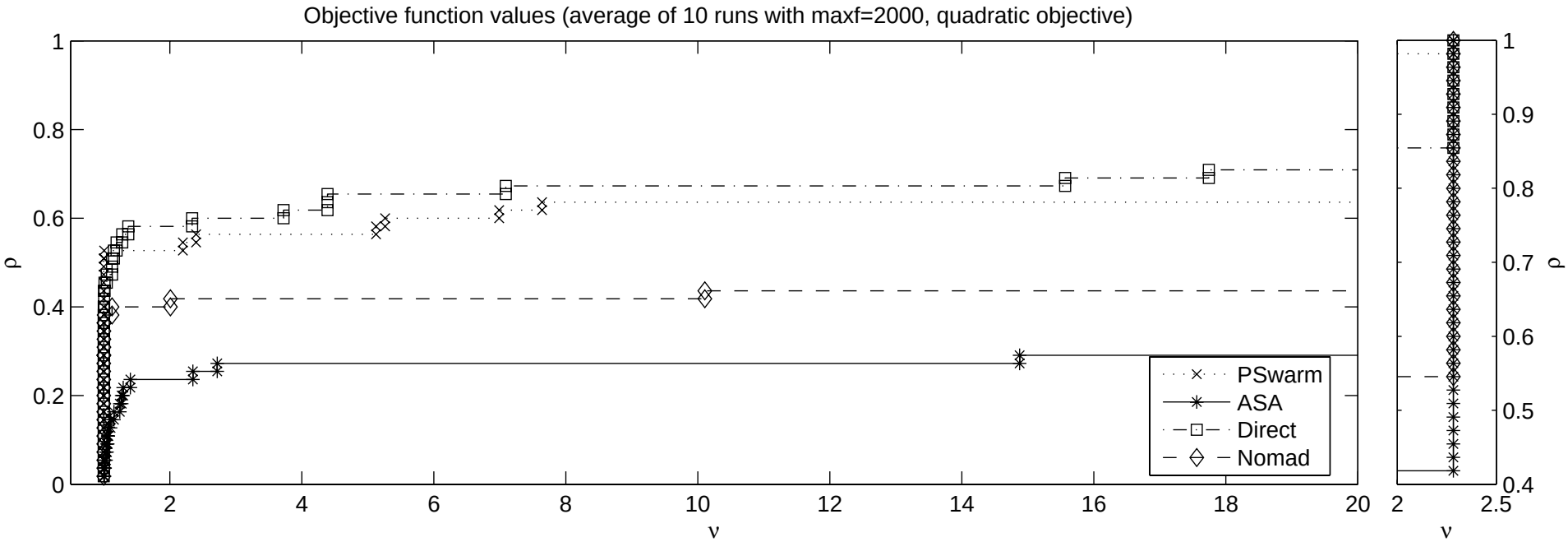
- The test problems are coded in AMPL (*A Modeling Language for Mathematical Programming*).

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

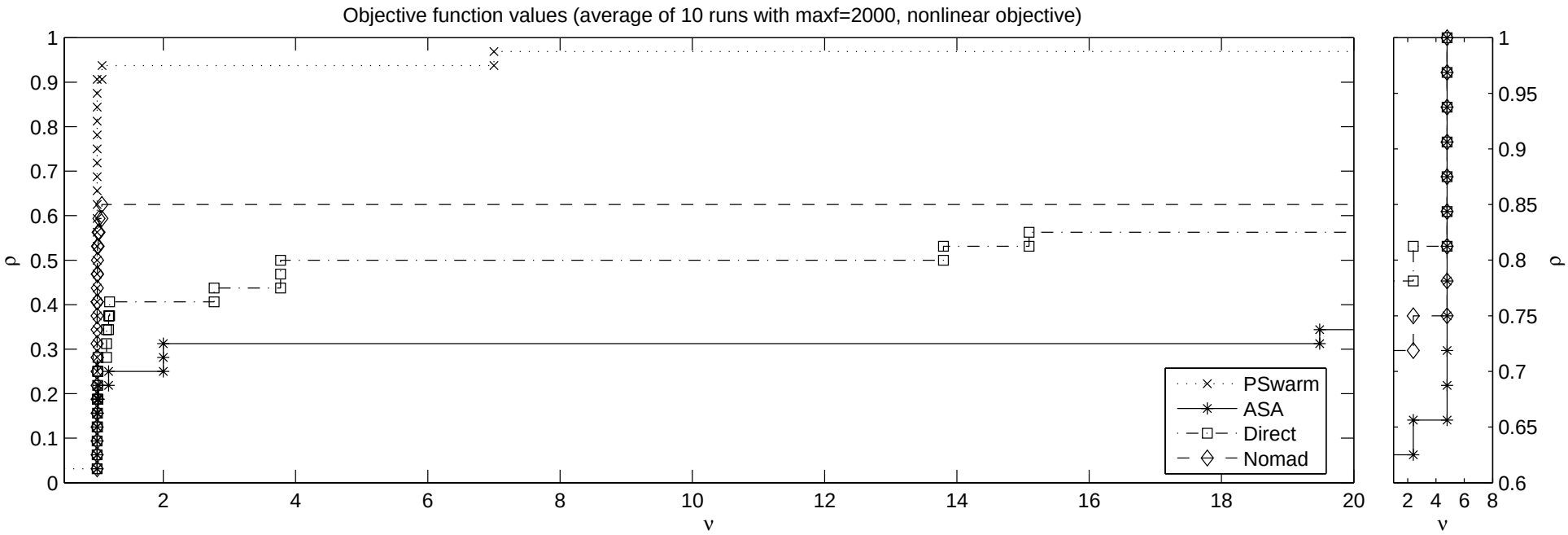
Linear objective functions



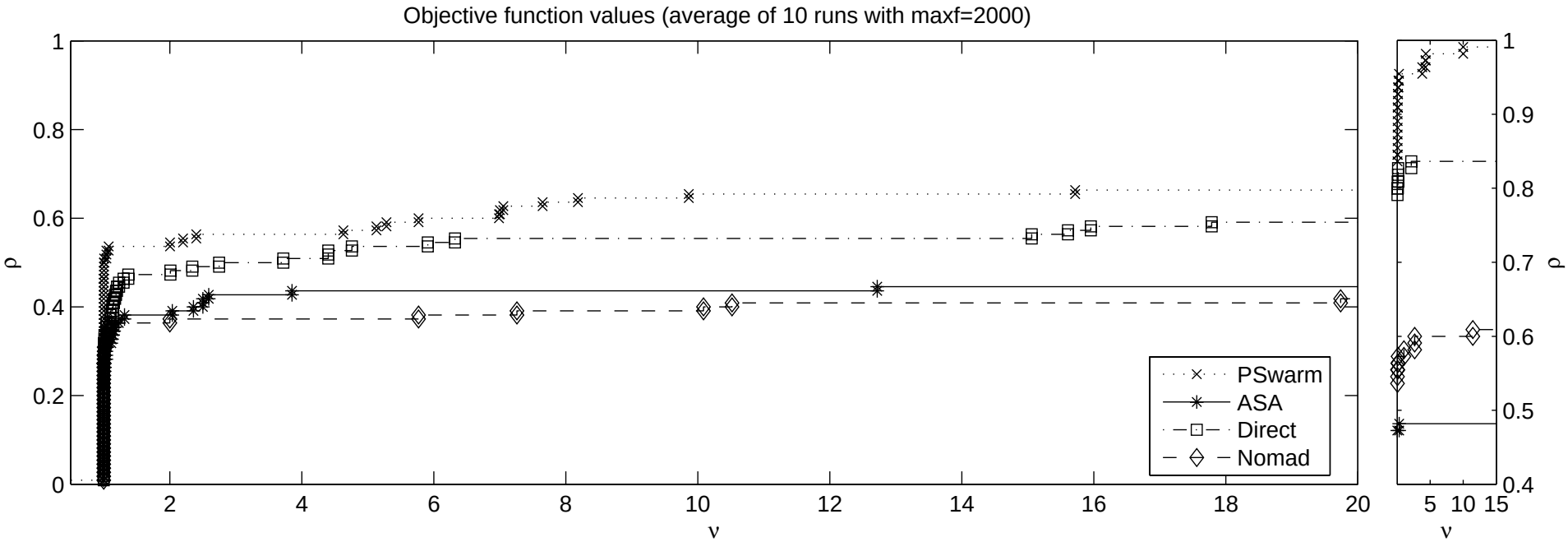
Quadratic objective functions



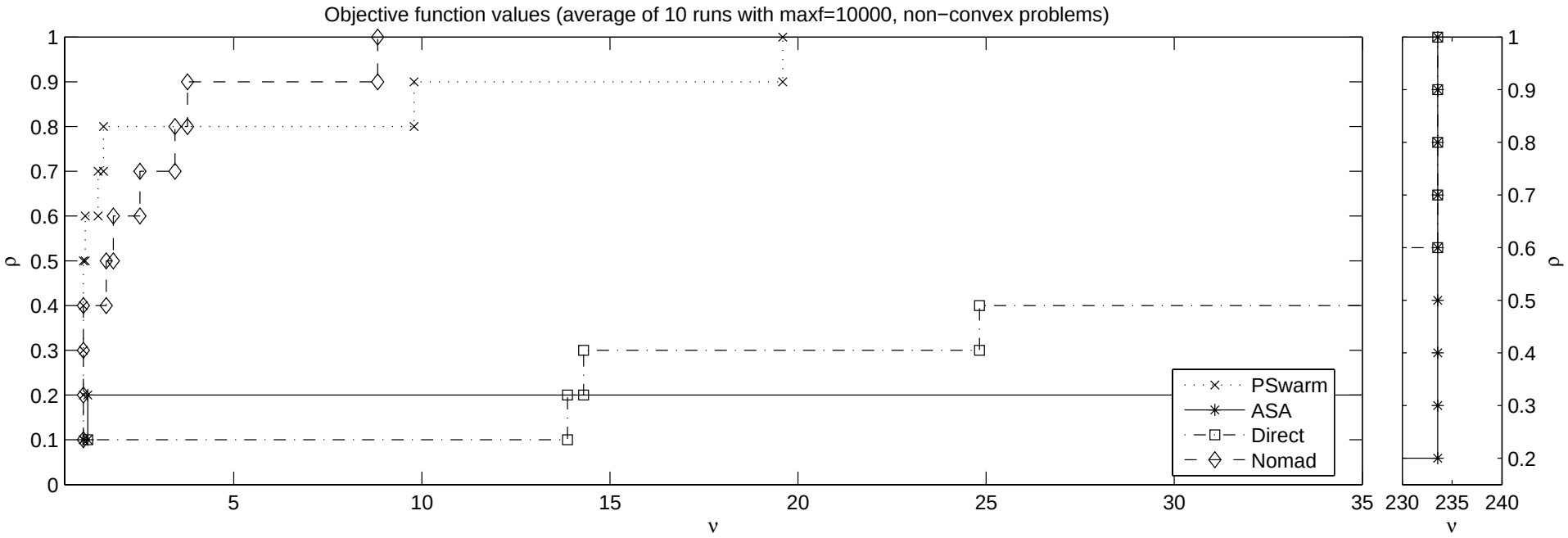
General nonlinear objective functions



All objective functions



Highly non-convex objective functions



Outline

Conclusions

Conclusions

- Development of a hybrid algorithm for derivative-free global optimization with bound and/or linear constraints.
- PSwarm shown to be a robust and competitive solver.

Availability

Only version 0.1 is publicly available at:

Version 1.1 available soon.

Conclusions

Conclusions

- Development of a hybrid algorithm for derivative-free global optimization with bound and/or linear constraints.
- `PSwarm` shown to be a robust and competitive solver.

Availability

Only version 0.1 is publicly available at:

- www.norg.uminho.pt/aivaz/pswarm
- the NEOS server

Version 1.1 available soon.

Conclusions

Conclusions

- Development of a hybrid algorithm for derivative-free global optimization with bound and/or linear constraints.
- `PSwarm` shown to be a robust and competitive solver.

Availability

Only version 0.1 is publicly available at:

- www.norg.uminho.pt/aivaz/pswarm
- the NEOS server

Version 1.1 available soon.

Conclusions

Conclusions

- Development of a hybrid algorithm for derivative-free global optimization with bound and/or linear constraints.
- `PSwarm` shown to be a robust and competitive solver.

Availability

Only version 0.1 is publicly available at:

- www.norg.uminho.pt/aivaz/pswarm
- the NEOS server

Version 1.1 available soon.

Conclusions

Conclusions

- Development of a hybrid algorithm for derivative-free global optimization with bound and/or linear constraints.
- `PSwarm` shown to be a robust and competitive solver.

Availability

Only version 0.1 is publicly available at:

- www.norg.uminho.pt/aivaz/pswarm
- the NEOS server

Version 1.1 available soon.

References


A.I.F. Vaz and L.N. Vicente.
A particle swarm pattern search method for bound constrained global optimization.
Journal of Global Optimization, 39:197–219, 2007.


Yin Zhang and Liyan Gao.
On numerical solution of the maximum volume ellipsoid problem.
SIAM Journal on Optimization, 14:53–76, 2003.

The end

email: aivaz@dps.uminho.pt
Web http://www.norg.uminho.pt/aivaz

email: lnv@mat.uc.pt
Web http://www.mat.uc.pt/∼lnv