Particle swarm and simulated annealing for multi-global optimization

A. Ismael F. Vaz and Edite M.G.P. Fernandes

Production and Systems Department - Engineering School

Minho University - Braga - Portugal

Ana I.P.N. Pereira

Mathematics Department - Polytechnic Institute of Braganca

Bragança - Portugal

`{aivaz,emgpf}@dps.uminho.pt, apereira@ipb.pt`

WSEAS 2005 - Lisbon - Portugal

June 16-18, 2005

Work partially supported by FCT grant POCI/MAT/58957/2004 and Algoritmi research center

Outline

- Multi-global optimization
- The Particle Swarm Paradigm
- The multi-local particle swarm optimization algorithm
- The Simulated Annealing
- The stretched simulated annealing algorithm
- Numerical results
- Conclusions

Multi-global optimization

We address the following optimization problem

$$\min_{x \in R^n} f(x)$$
$$s.t. \;\; a \leq x \leq b$$

where $f : R^n \longrightarrow R$ is the objective function and a, b are the simple bounds on the variables x.

We try to obtain all the feasible global optima for function $f(x)$.

The Particle Swarm Paradigm (PSP)

The PSP is a population (swarm) based algorithm that mimics the social behavior of a set of individuals (particles).

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

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

The new travel 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 $j = 1, \ldots, n$, where $\iota(t)$ is a weighting factor (inertial), μ is the *cognition* parameter and ν is the *social* parameter. $\omega_{1j}(t)$ and $\omega_{2j}(t)$ are random numbers drawn from the uniform $(0, 1)$ distribution.

The best ever particle

$\hat{y}(t)$ is a particle position with global best function value so far, *i.e.*,

$$\hat{y}(t) = \arg\min_{a \in \mathcal{A}} f(a)$$
$$\mathcal{A} = \left\{ y^1(t), \ldots, y^s(t) \right\} .$$

In an algorithmic point of view we just have to keep track of the particle with the best ever function value.

Features

Population based algorithm.

1. Good
   (a) Easy to implement.
   (b) Easy to parallelize.
   (c) Easy to handle discrete variables.
   (d) Only uses objective function evaluations.
2. Not so good
   (a) Slow rate of convergence near an optimum.
   (b) Quite large number of function evaluations.
   (c) In the presence of several global optima the algorithm may not converge.

The multi-local particle swarm optimization (MLPSO) algorithm

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

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

for $j = 1, \ldots, n$, where $\nabla f(x)$ is the gradient of the objective function.

Each particle uses the steepest descent direction.

The simulated annealing (SA)

The SA method can be characterized by four main phases:

- generation of a new candidate point;
- acceptance criterion;
- reduction of the control parameters;
- stopping rule.

Generating a new candidate

In the adaptived simulated annealing (ASA) algorithm, a new feasible candidate point is generated by

$$y_j = x_j(t) + \lambda_j(b_j - a_j)$$

for $j = 1, \ldots, n$, with $\lambda_j \in (-1, 1)$ given by

$$\lambda_j = sgn\left(u - \frac{1}{2}\right)\left(\left(1 + \frac{1}{c_{F_j}(t)}\right)^{|2u-1|} - 1\right) c_{F_j}(t)$$

where u is drawn from the uniform $(0, 1)$ distribution.

$c_{F_j}(t)$ parameters are updated in all iterations, and redefined when appropriate.

Acceptance criterion - Metropolis

$$x(t+1) = \begin{cases} y & \text{if} \qquad \tau \leq \min\left\{1, e^{\frac{f(x(t))-f(y)}{c_A(t)}}\right\} \\ x(t) & \text{otherwise} \end{cases}$$

where τ is drawn from the uniform $(0, 1)$ distribution.

If $f(y) \leq f(x(t))$ then y is always accepted. If $f(y) > f(x(t))$ then y is accepted with some probability.

The control parameter $c_A(t)$ (cooling schedule) must be updated so that

$$\lim_{t\to\infty} c_A(t) = 0.$$

$c_A(t)$ is also redefined when appropriate.

Features

Not population based (one point in each iteration).

1. Good
   (a) Can be applied to discrete and continuous optimization problems.
   (b) Asymptotically converges to a global solution.
   (c) Only uses objective function evaluations.

2. Not so good
   (a) Not so easy to implement.
   (b) Depends highly on several parameters.
   (c) Requires good choices of initial parameters.
   (d) Large number of function evaluations.

The stretched simulated annealing (SSA) algorithm

Assumption: All global solutions are isolated points.

The SSA algorithm generates a sequence of optimization problems defined as

$$\min_{a \le x \le b} \Phi(x) = \begin{cases} f(x) & \text{if} \quad t = 1 \\ h(x) & \text{if} \quad t > 1 \end{cases}$$

where

$$h(x) = \begin{cases} \tilde{f}(x) & \text{if} \;\; x \in \mathcal{V}_\epsilon(\bar{x}) \\ f(x) & \text{otherwise} \end{cases}$$

$\bar{x}$ represents an already detected global minimizer, $\mathcal{V}_\epsilon(\bar{x})$ denotes a neighborhood ϵ of $\bar{x}$, and $\tilde{f}(x)$ is the stretched function. Each global minimizer is found by ASA algorithm.

Stretched function $\tilde{f}(x)$

Two stage transformation of f:

Elevation

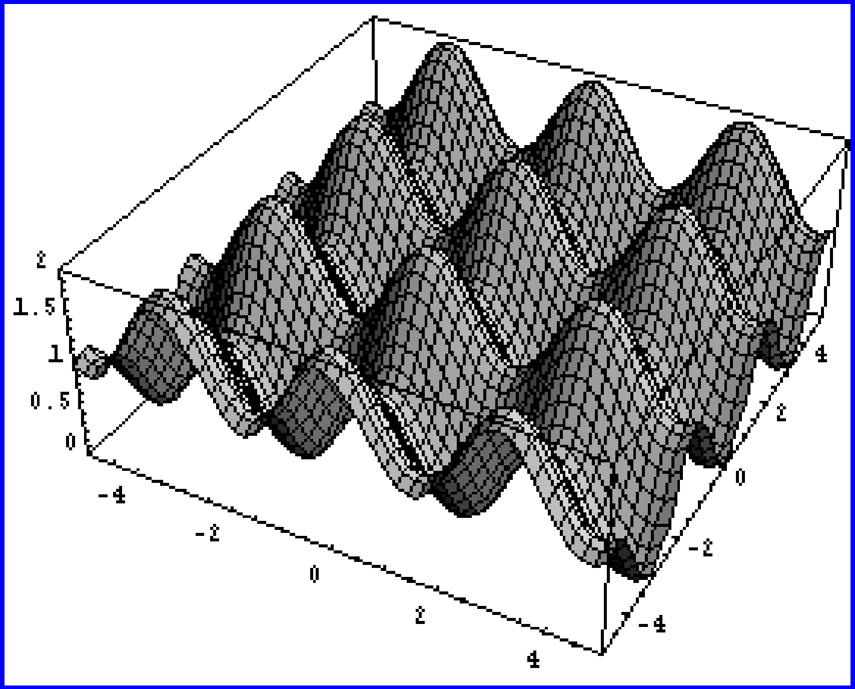
$$\bar{f}(x) = f(x) + \frac{\gamma_1}{2}\|x - \bar{x}\|(sgn(f(x) - f(\bar{x})) + 1)$$

Stretching

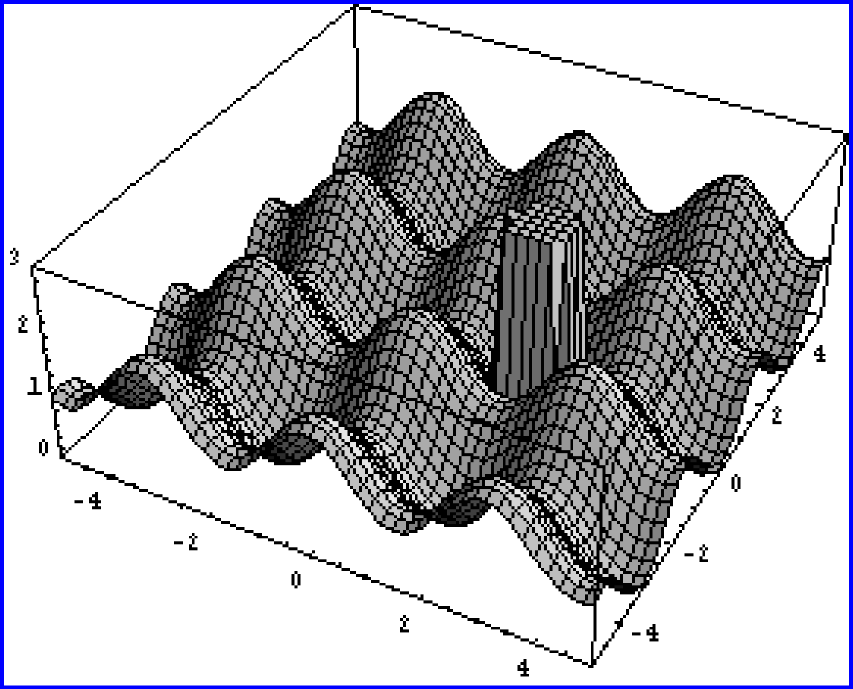
$$\tilde{f}(x) = \bar{f}(x) + \gamma_2 \frac{sgn(f(x) - f(\bar{x})) + 1}{2\tanh(\xi(\bar{f}(x) - \bar{f}(\bar{x})))}$$

The function stretching effect

In $[-5, 5]^2$, the function has 12 global minimizers:



Original function



Stretched function

After finding $\left(\frac{\pi}{2}, 0\right)$, SSA algorithm searches for another global in the stretched function.

NUMERICAL RESULTS

Test functions	n	N_{global}	$f(x^*)$
b2	2	1	0
bohachevsky	2	1	0
branin	2	3	3.98E-01
dejoung	3	1	0
easom	2	1	-1
f1	30	1	-1.26E+04
goldprice	2	1	3
griewank	6	1	0
hartmann3	3	1	-3.86E+00
hartmann6	6	1	-3.32E+00
hump	2	2	0
hump_camel	2	2	-1.03E+00
levy3	2	18	-1.77E+02
parsopoulos	2	12	0
rosenbrock10	10	1	0
rosenbrock2	2	1	0

NUMERICAL RESULTS

Test functions	n	N_{global}	$f(x^*)$
rosenbrock5	5	1	0
shekel10	4	1	-1.05E+01
shekel5	4	1	-1.02E+01
shekel7	4	1	-1.04E+01
shubert	2	18	-1.87E+02
storn1	2	2	-4.08E-01
storn2	2	2	-1.81E+01
storn3	2	2	-2.28E+02
storn4	2	2	-2.43E+03
storn5	2	2	-2.48E+04
storn6	2	2	-2.49E+05
zakharov10	10	1	0
zakharov2	2	1	0
zakharov20	20	1	0
zakharov4	4	1	0
zakharov5	5	1	0

MLPSO

	freq	N_{it}	N_{ge}	f^*_{avg}	f^*_{min}
b2	100%	68851	1124	1.84E-10	3.14E-11
bohachevsky	100%	26811	1546	3.81E-11	1.39E-14
branin	100%	16386	2425	3.98E-01	3.98E-01
dejoung	100%	14187	45659	5.67E-14	2.60E-16
easom	0%	Flat problem			
f1	0%	Non differentiable			
goldprice	0%	100000	56	1.21E+02	2.34E+01
griewank	67%	29873	1217700	5.01E-03	9.75E-09
hartmann3	80%	100000	913	-3.79E+00	-3.85E+00
hartmann6	0%	100000	3530	-2.90E+00	-3.09E+00
hump	100%	24600	996	4.65E-08	4.65E-08
hump_camel	100%	22548	944	-1.03E+00	-1.03E+00
levy3	1%	100000	565	-1.47E+02	-1.72E+02
parsopoulos	85%	46086	1520	5.14E-17	9.38E-21
rosenbrock10	0%	100000	2126	1.18E+04	7.74E+03
rosenbrock2	0%	100000	46	1.08E+01	1.59E+00

MLPSO

	freq	N_{it}	N_{ge}	f^*_{avg}	f^*_{min}
rosenbrock5	0%	100000	3178	3.03E+02	8.52E+01
shekel10	100%	100000	14977	-8.28E+00	-1.01E+01
shekel5	100%	100000	19100	-7.63E+00	-1.00E+01
shekel7	100%	100000	16596	-8.42E+00	-1.00E+01
shubert	7%	100000	253	-1.42E+02	-1.80E+02
storn1	100%	24297	3148	-4.08E-01	-4.08E-01
storn2	90%	68360	451	-1.81E+01	-1.81E+01
storn3	60%	84587	218	-2.00E+02	-2.28E+02
storn4	60%	100000	161	-2.28E+03	-2.43E+03
storn5	40%	100000	4222	-2.39E+04	-2.48E+04
storn6	10%	100000	46	-1.18E+05	-2.36E+05
zakharov10	0%	100000	1829	6.25E+01	4.63E+01
zakharov2	100%	21401	3820	1.39E-11	2.95E-14
zakharov20	0%	100000	1901	2.02E+02	1.33E+02
zakharov4	0%	100000	2362	4.70E+00	2.40E+00
zakharov5	0%	100000	1454	9.39E+00	3.13E+00

SSA

	freq	N_{ASA}	N_{fe}	f^*_{avg}	f^*_{min}
b2	100%	5	24066	2.05E-06	3.86E-11
bohachevsky	100%	6	34411	5.46E-08	1.82E-09
branin	100%	6	10529	3.98E-01	3.98E-01
dejoung	100%	4	10606	9.59E-07	9.37E-08
easom	100%	4	17422	-1.00E+00	-1.00E+00
f1	0%	4	100000	-1.31E+04	-1.34E+04
goldprice	100%	4	26197	3.00E+00	3.00E+00
griewank	0%	16	100000	1.18E-02	9.86E-03
hartmann3	100%	4	13379	-3.86E+00	-3.86E+00
hartmann6	100%	5	78301	-3.32E+00	-3.32E+00
hump	100%	5	20200	1.35E-07	4.66E-08
hump_camel	100%	5	17531	-1.03E+00	-1.03E+00
levy3	37%	11	18217	-1.77E+02	-1.77E+02
parsopoulos	100%	15	16542	3.21E-09	2.68E-10
rosenbrock10	0%	4	100000	6.98E-01	7.23E-02
rosenbrock2	80%	10	66902	1.90E-02	1.17E-03

SSA

	freq	N_{ASA}	N_{fe}	f^*_{avg}	f^*_{min}
rosenbrock5	60%	6	111073	1.02E-02	6.43E-03
shekel10	80%	7	32961	-1.05E+01	-1.05E+01
shekel5	80%	6	29745	-1.02E+01	-1.02E+01
shekel7	80%	5	22206	-1.04E+01	-1.04E+01
shubert	99%	32	51684	-1.87E+02	-1.87E+02
storn1	100%	5	5850	-4.08E-01	-4.08E-01
storn2	100%	5	39877	-1.81E+01	-1.81E+01
storn3	100%	5	63510	-2.28E+02	-2.28E+02
storn4	100%	5	59841	-2.43E+03	-2.43E+03
storn5	100%	5	101864	-2.48E+04	-2.48E+04
storn6	100%	5	103191	-2.49E+05	-2.49E+05
zakharov10	100%	4	80004	5.77E-03	3.90E-04
zakharov2	100%	4	3775	3.37E-07	1.25E-10
zakharov20	0%	5	100000	2.57E+00	2.22E+00
zakharov4	100%	4	24747	1.81E-06	2.34E-07
zakharov5	100%	4	44203	6.72E-06	2.29E-06

Conclusions

- Two approaches to multi-global optimization problems;
- The MLPSO algorithm is able to identify most of the global optima as well as several local optima;
- The SSA algorithm is able to identity most of the global optima and some local optima in particular problems.

The End

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

First Page