The LifeCycle Model

By Nathan Weir

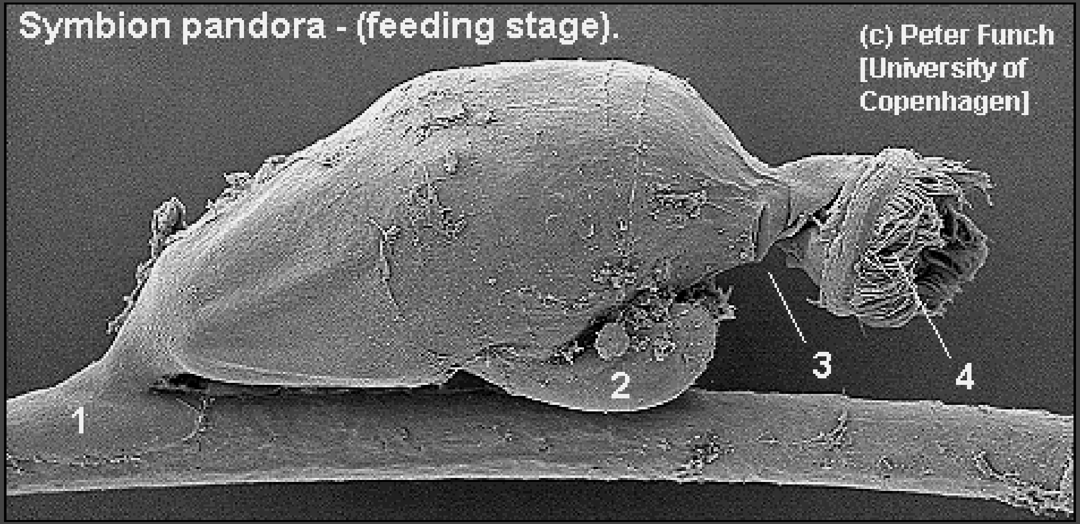
Choosing a Search Heuristic

- Genetic Algorithms, Particle Swarm Optimization, and Hill Climbing each have different strengths & weaknesses
  - GA - Widely applicable, powerful with domain info
  - PSO - Often achieves superior results to GAs on numerical analysis problems
  - HC - Good for local search, bad for functions with many local maxima/minima that are close to the global max/min
- Choosing just one algorithm can be difficult, risk of missing a better choice.

Combining GAs, PSOs, and HCs

- What if we combined the major types of adaptive search heuristics?
  - Best of all worlds - None of the drawbacks and all of the strengths of each algorithm.
- Inspired by the life cycle adaptations of an organism
  - Puberty, mating seasons, hair density by the season
  - *Symbion pandora* - Lobster parasite, changes behavior depending on activities of host

Life cycle Symbion pandora



High Level Overview

Searching for the minimum

```
program LifeCycle Model
begin
   initialize
   while (not terminate-condition) do
      for (all individuals)
         evaluate fitness
         switch LifeCycle stage if no recent improvement
      for (PSO particles)
         calculate new velocity vectors
         move
      for (GA individuals)
         select new population
         recombine population
         mutate population
      for (HillClimbers)
         find possible new neighbouring solution
         evaluate fitness for the new solution
         shift to new solution with probability p
end
```

PSO Setup

At each iteration, the velocity of each particle is updated by

$$v_i = \chi(wv_i + \varphi_{1i} (p_i - x_i) + \varphi_{2i} (p_g - x_i))$$

Where

χ = constriction coefficient
w = inertia weight
φ_1, φ_2 = random vals, diff for each particle in each dim
v_i = velocity of the ith particle
p_i = position of best candidate sol. found by ith particle
p_g = best global position found so far

GA Setup

- "Standard" GA
- Uses arithmetic crossover
  - Each parent is replaced by a child that is comprised of a random amount of the genes from each parent
- Mutation determined per dimension, constructed so that it's effect decreases over time until it is negligible.

Hill Climber Setup

- "stochastic" hill climber
  - Standard hill climbing, but instead of examining all surrounding points, simply pick one at random and advance if it's a better enough point. Otherwise, try again.

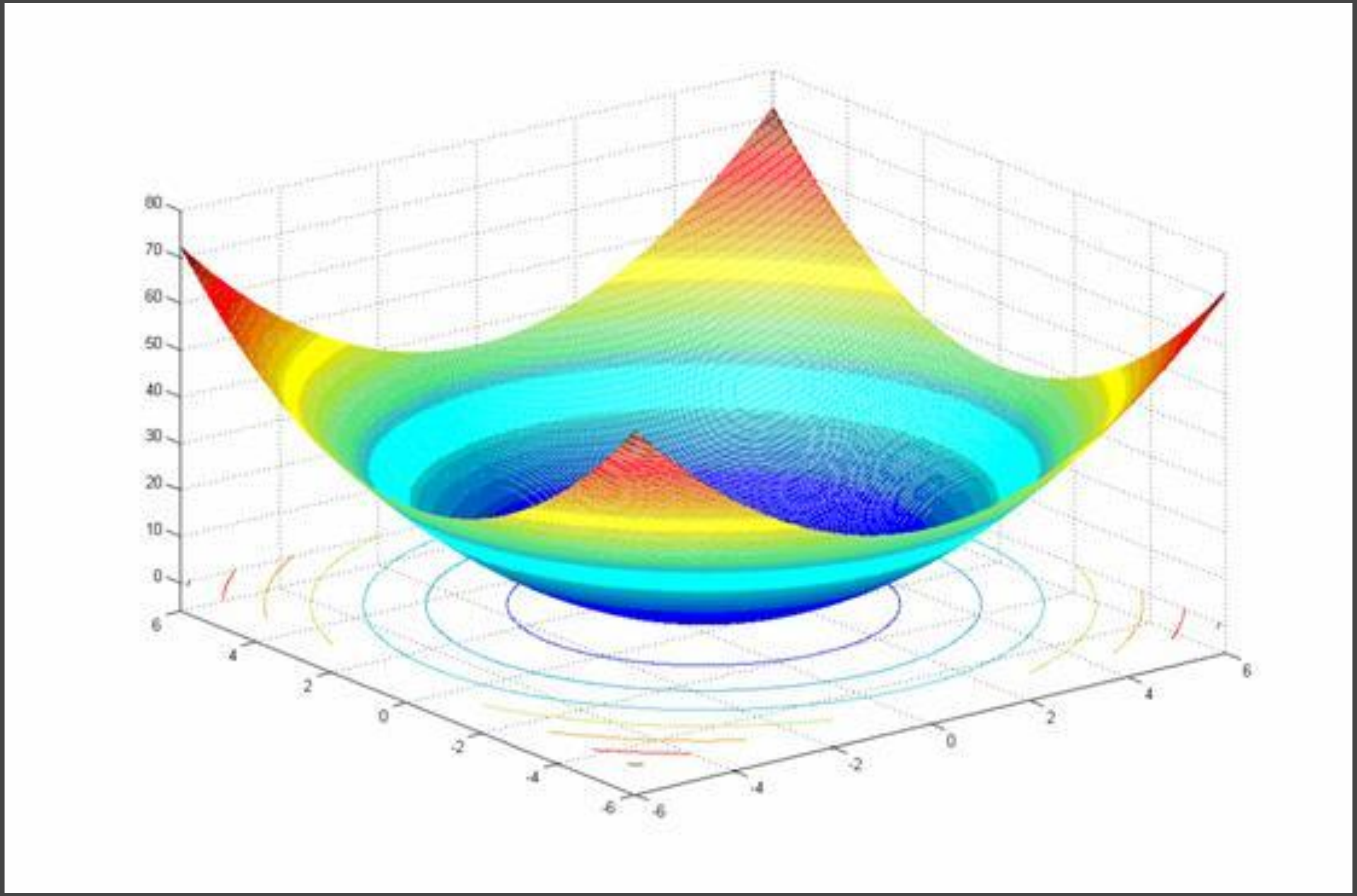
The Test

- Test performance of LifeCycle model versus straight GA, PSO, and HC algorithms.
- 5 benchmark functions: Sphere, Rosenbrock, Griewank, Rastrigan, and Ackley.

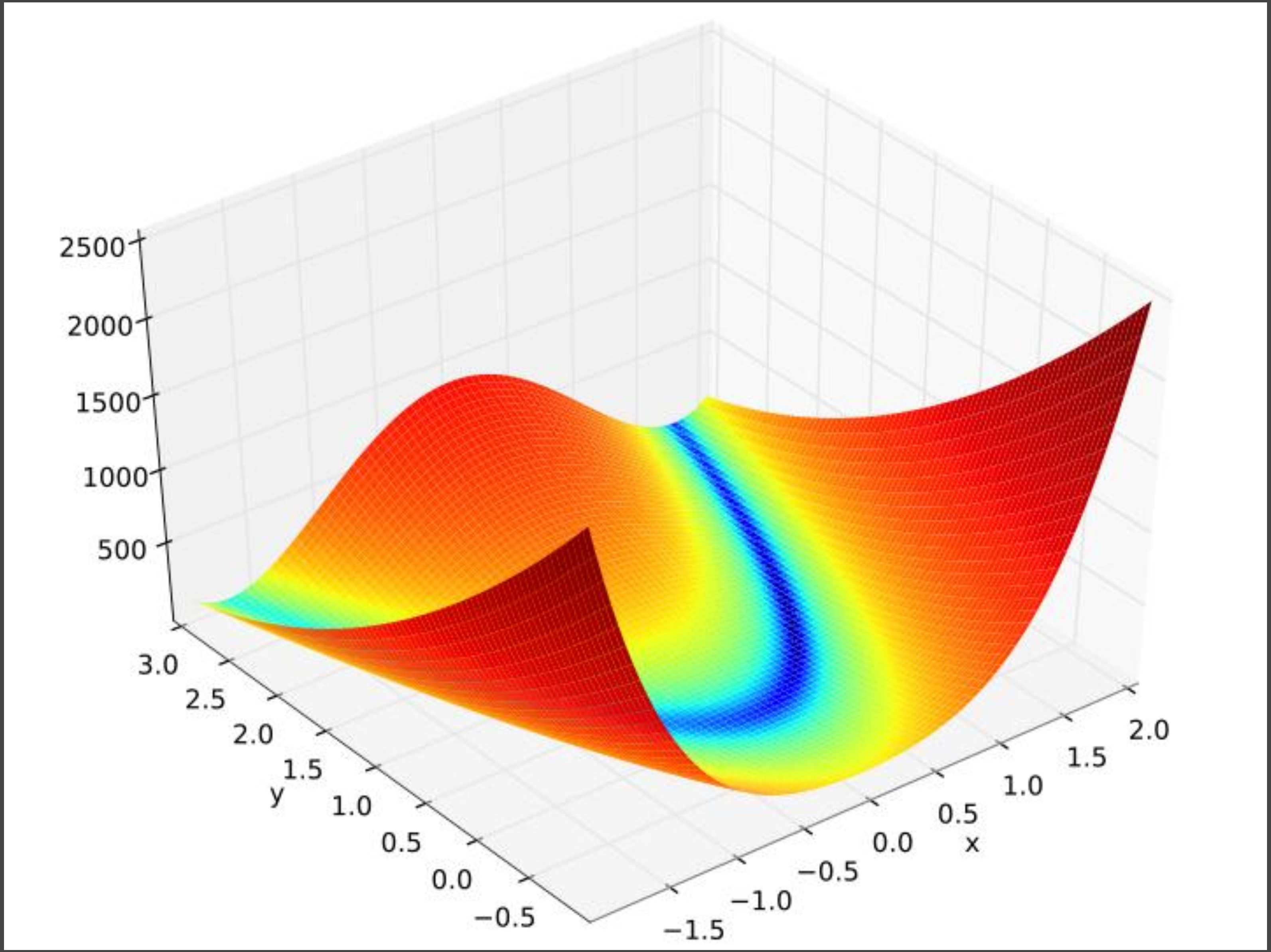
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$
Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$
Griewank	$f_3(x) = \frac{1}{4000} \sum_{i=1}^{n} (x_i - 100)^2 - \prod_{i=1}^{n} cos(\frac{x_i - 100}{\sqrt{i}}) + 1$
Rastrigin	$f_4(x) = \sum_{i=1}^{n} (x_i^2 - 10cos(2\pi x_i) + 10)$
Ackley	$f_5(x) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)}$

Function	Search space	Initialisation range
f_1 *Sphere*	$-100 \leq x_i \leq 100$	$50 \leq x_i \leq 100$
f_2 *Rosenbrock*	$-100 \leq x_i \leq 100$	$15 \leq x_i \leq 30$
f_3 *Griewank*	$-600 \leq x_i \leq 600$	$300 \leq x_i \leq 600$
f_4 *Rastrigin*	$-10 \leq x_i \leq 10$	$2.56 \leq x_i \leq 5.12$
f_5 *Ackley*	$-32.768 \leq x_i \leq 32.768$	$16.384 \leq x_i \leq 32.768$

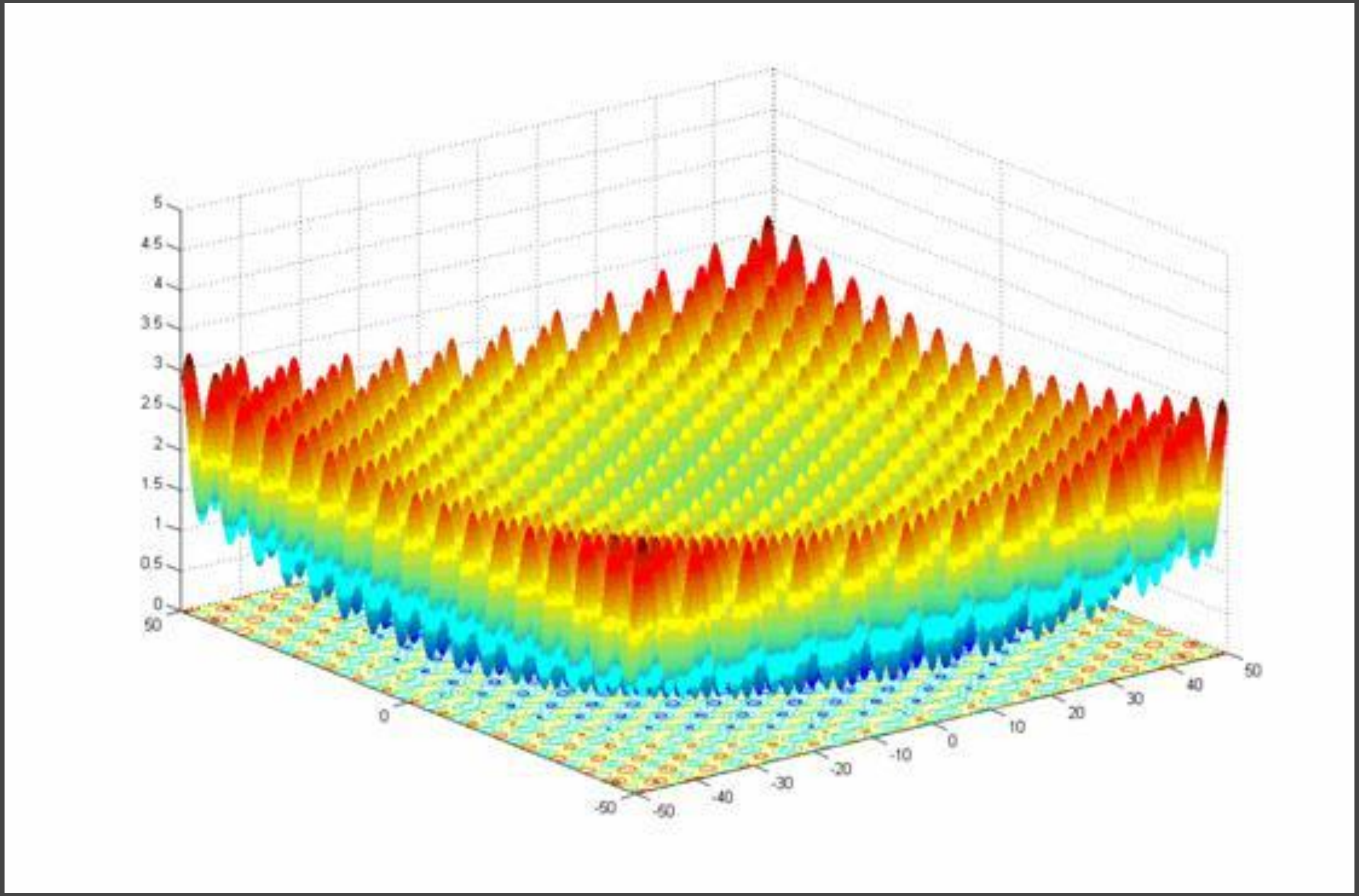
Sphere



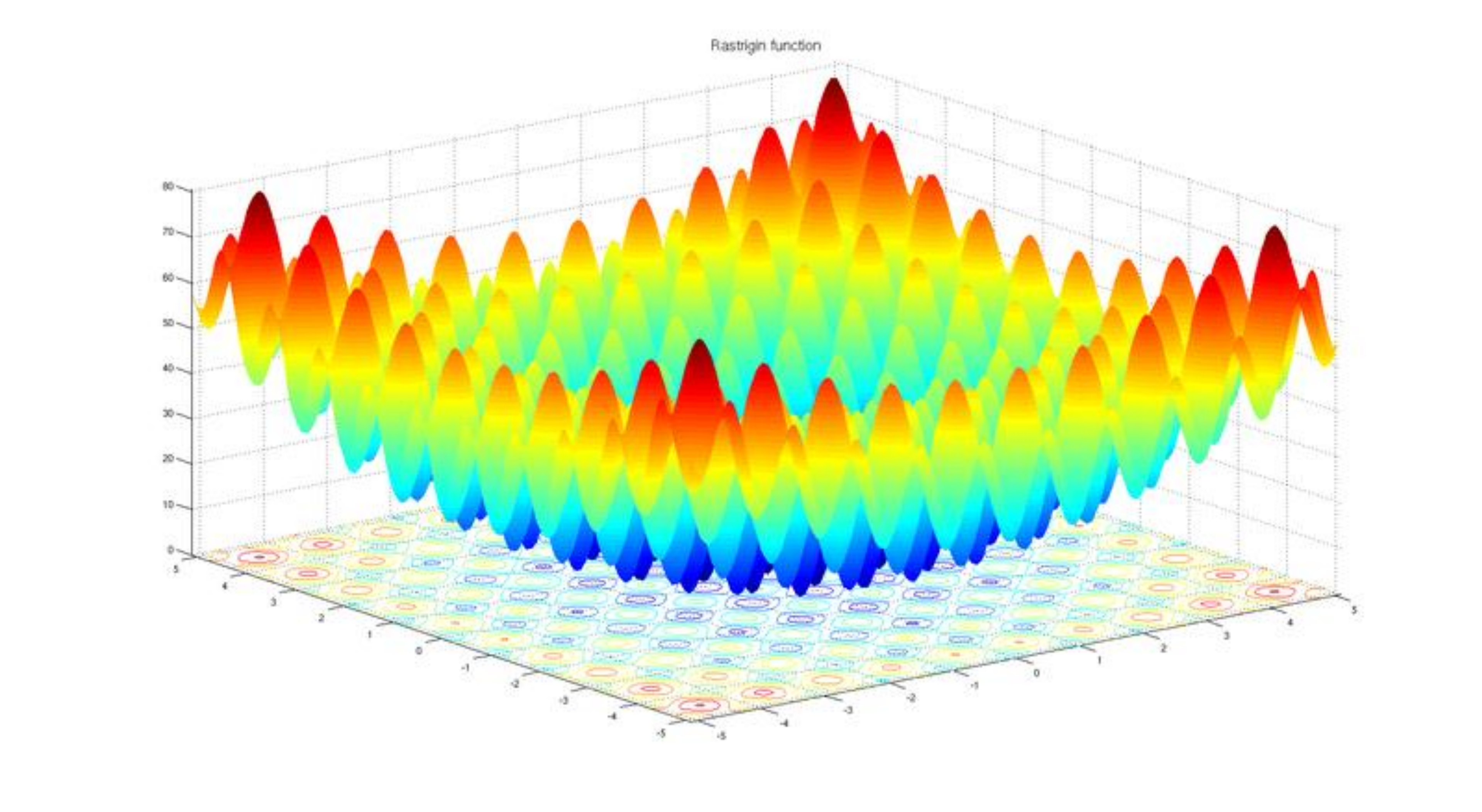
Rosenbrock


2500
2000
1500
1000
500
3.0
2.5
2.0
1.5
y
1.0
0.5
0.0
−0.5
−1.5
−1.0
−0.5
0.0
0.5
x
1.0
1.5
2.0

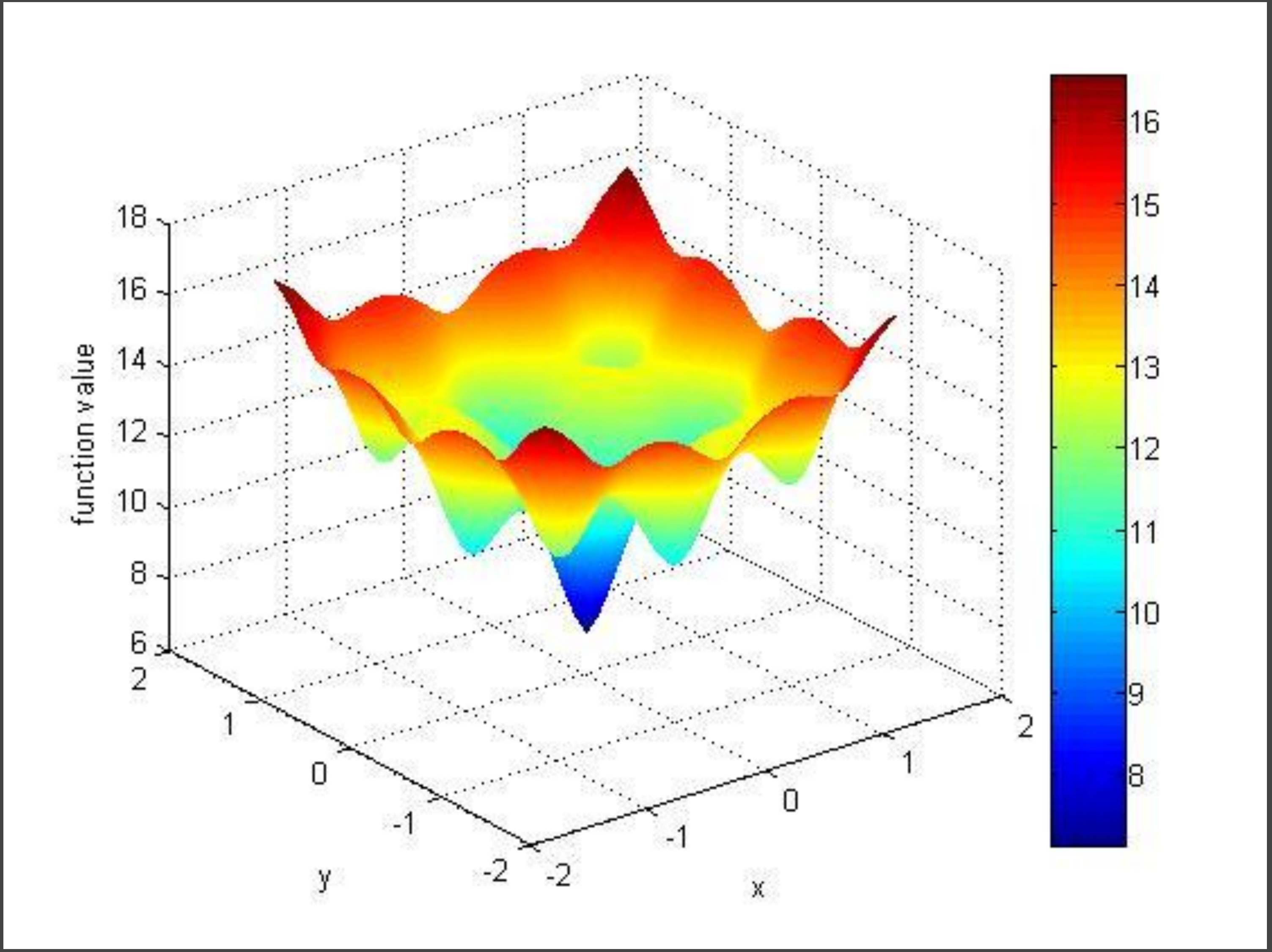
Griewank



Rastrigan



Ackley



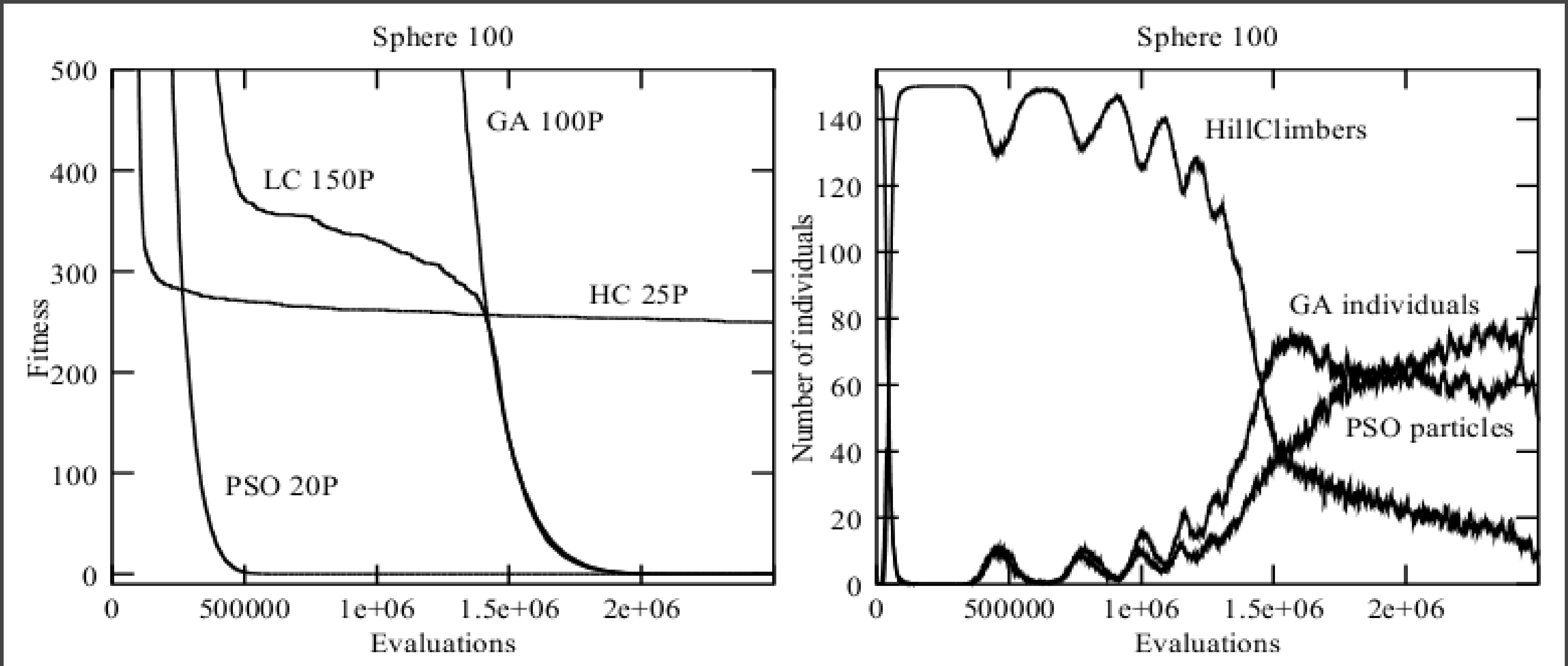
function value
18
16
14
12
10
8
6
2
1
0
-1
-2
y
-2
-1
0
1
2
x
16
15
14
13
12
11
10
9
8

Results

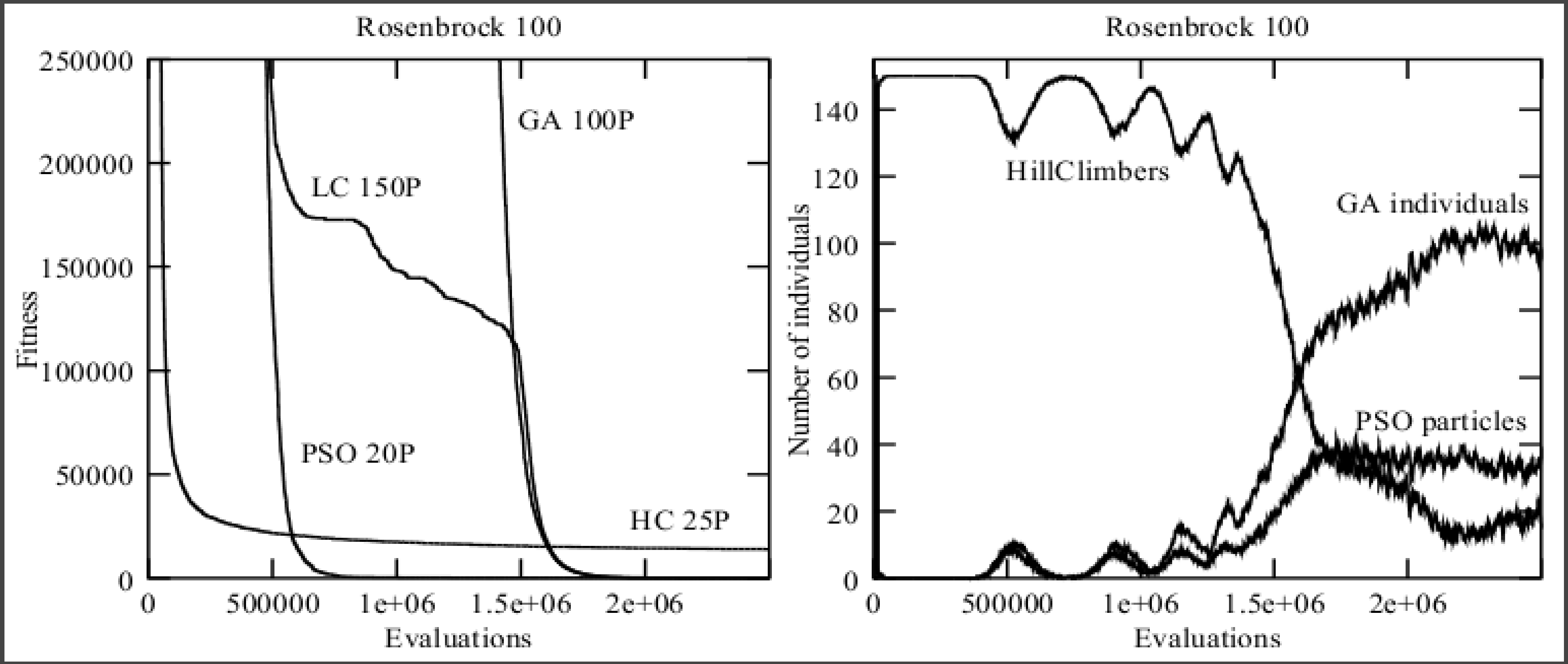
Not a complete waste of time!

- The LifeCycle model was at least nearly the winner in every test, and for the Ackley function completely outperformed the competition.
- Found to be in general a nice middle ground.

Sphere - Results

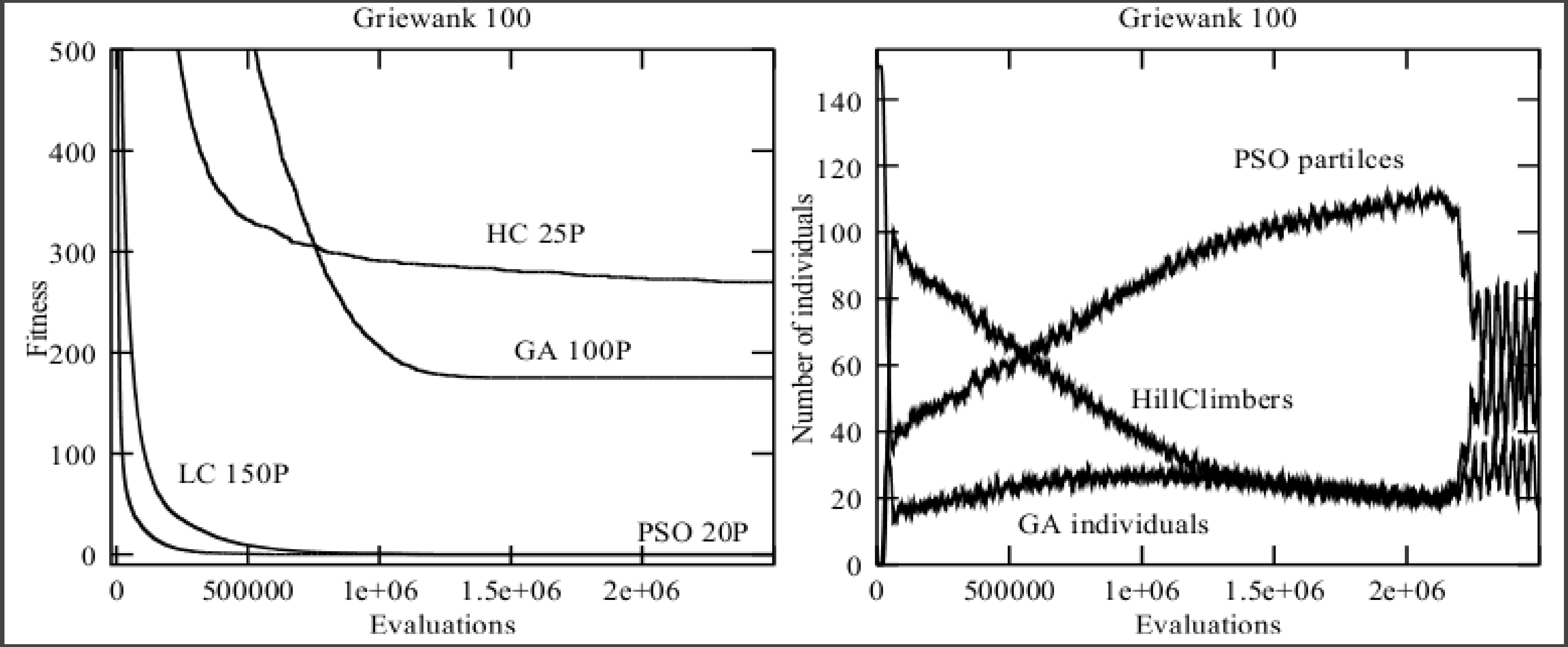


Rosenbrock



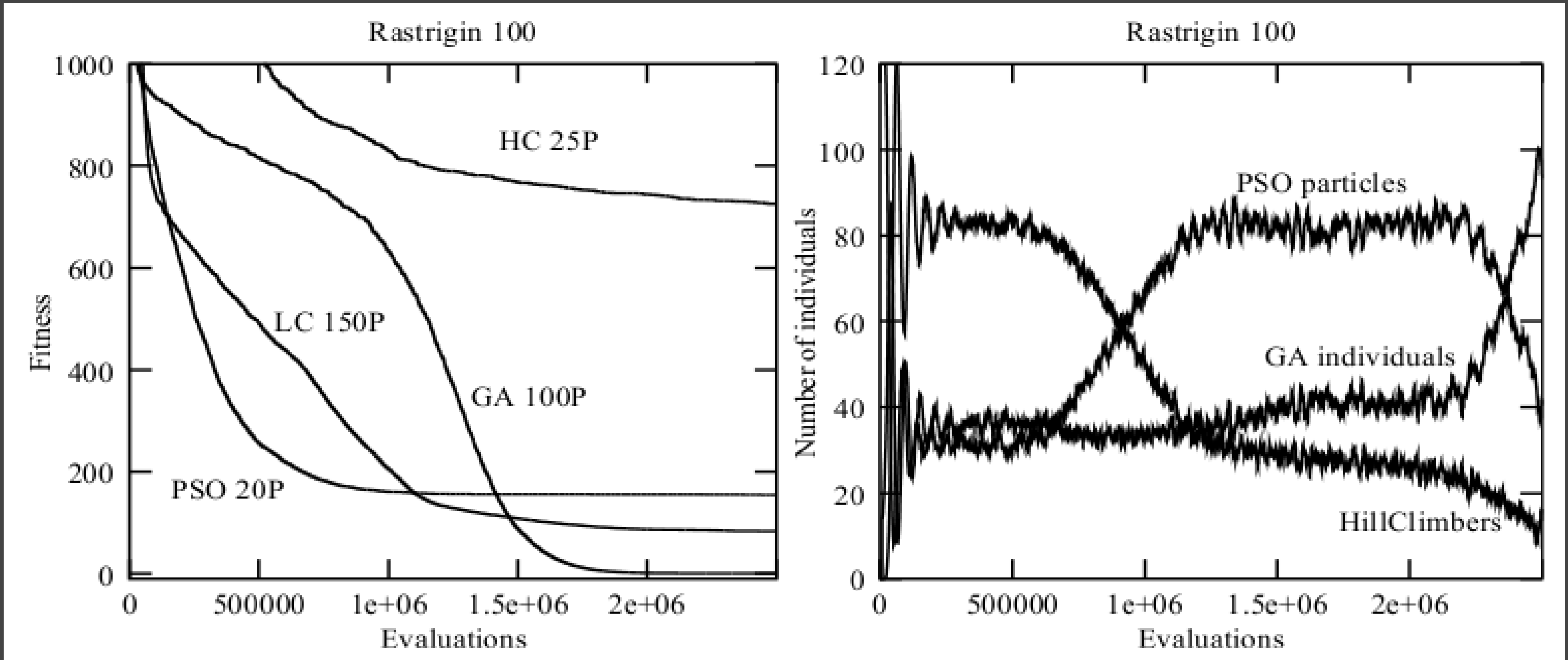
Rosenbrock 100
250000
200000
150000
100000
50000
0
Fitness
GA 100P
LC 150P
PSO 20P
HC 25P
0
500000
1e+06
1.5e+06
2e+06
Evaluations
Rosenbrock 100
140
120
100
80
60
40
20
0
Number of individuals
HillClimbers
GA individuals
PSO particles
0
500000
1e+06
1.5e+06
2e+06
Evaluations

Griewank



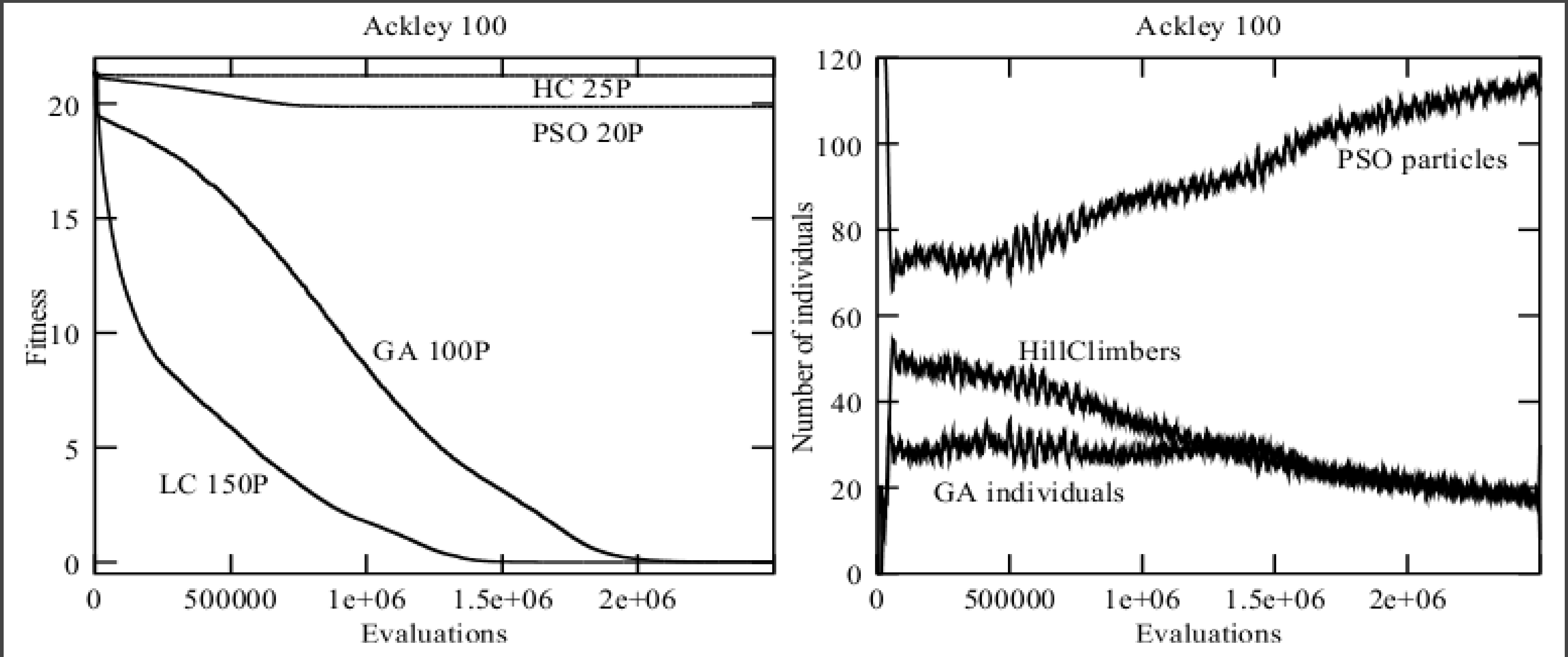
Griewank 100
Fitness
500
400
300
200
100
0
0
500000
1e+06
1.5e+06
2e+06
Evaluations
HC 25P
GA 100P
LC 150P
PSO 20P
Griewank 100
Number of individuals
140
120
100
80
60
40
20
0
0
500000
1e+06
1.5e+06
2e+06
Evaluations
PSO partilces
HillClimbers
GA individuals

Rastrigan



Rastrigin 100
Fitness
1000
800
600
400
200
0
0
500000
1e+06
1.5e+06
2e+06
Evaluations
HC 25P
LC 150P
GA 100P
PSO 20P
Rastrigin 100
Number of individuals
120
100
80
60
40
20
0
0
500000
1e+06
1.5e+06
2e+06
Evaluations
PSO particles
GA individuals
HillClimbers

Ackley


Ackley 100
HC 25P
PSO 20P
GA 100P
LC 150P
Fitness
Evaluations
0
500000
1e+06
1.5e+06
2e+06
20
15
10
5
0
Ackley 100
PSO particles
HillClimbers
GA individuals
Number of individuals
Evaluations
120
100
80
60
40
20
0
500000
1e+06
1.5e+06
2e+06

Future Work

- Try variants of the search algorithms used.
- Different search algorithms: SOC EAs (?), simulated annealing, etc.
- Taboo search - prevent repeating possible solutions once they are marked as "taboo"
- Exchanging PSO/HC particles with multiple GA individuals
  - GAs work better with large populations, creating groups of them may yield better results.