2010-04-15
Maurice.Clerc@WriteMe.com

A mini-benchmark

You have designed a nice brand new algorithm. Before to run it on a large (unbiased) benchmark, you may try it on this mini-benchmark of four problems. It has been carefully chosen to be deceptive, *in average*, if the algorithm is not well "balanced" or is biased in favour of a diagonal of the search space.

Tripod

The function to be minimised is ([2])

$$\begin{aligned} f = & \frac{1-sign(x_2)}{2}\left(|x_1| + |x_2+50|\right) \\ & + \frac{1+sign(x_2)}{2}\frac{1-sign(x_1)}{2}\left(1+|x_1+50| + |x_2-50|\right) \\ & + \frac{1+sign(x_1)}{2}\left(2+|x_1-50| + |x_2-50|\right) \end{aligned} \quad (1)$$

with

$$\begin{aligned} sign(x) &= -1 \quad \text{if} \quad x \leq 0 \\ &= 1 \quad \text{else} \end{aligned}$$

The search space is $[-100, 100]^2$. The solution point is $(0, -50)$, where $f = 0$. Here, we allow 10^4 fitness evaluations and a run is said to be successful if it finds a fitness less than 0.0001.

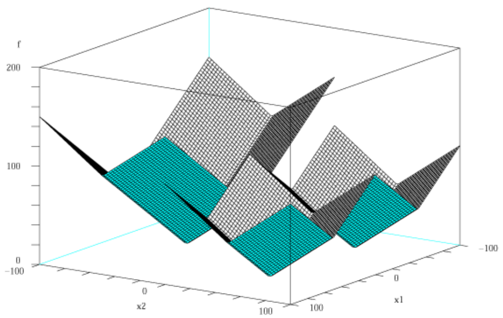

Figure 1: Tripod function. Not that difficult, but may be deceptive for algorithms that are easily trapped into a local minimum

Rosenbrock F6

$$f = 390 + \sum_{d=2}^{10} \left(100\left(z_{d-1}^2 - z_d\right)^2 + \left(z_{d-1} - 1\right)^2\right) \quad (2)$$

with $z_d = x_d - o_d$. The search space is $[-100, 100]^{10}$. The offset vector $O = (o_1, \cdots, o_{10})$ is defined by its C code below. The solution point is $O + (1, \ldots, 1)$ where $f = 390$. There is also a local minimum at $(o_1 - 2, \cdots, o_{10})$, where $f = 394$[1]. Here, we allow 10^5 fitness evaluations and a run is said to be successful if it finds a fitness less than 0.01. This problem is coming from the CEC 2005 benchmark and is difficult for algorithms that reduce too quickly the searched area space, and particularly discriminant in terms of success rate.

Offset (C code)

static double offset_2[10] = { 81.0232, -48.395, 19.2316, -2.5231, 70.4338, 47.1774, -7.8358, -86.6693, 57.8532, -9.9533}

Compression spring

This is a simplified version of a more difficult problem (see[4, 1, 3]). There are three variables

$$\begin{array}{lclll} x_1 & \in & \{1, \ldots, 70\} & \text{granularity} & 1 \\ x_2 & \in & [0.6, 3] & & \\ x_3 & \in & [0.207, 0.5] & \text{granularity} & 0.001 \end{array}$$

and four constraints

$$\begin{array}{lcl} g_1 & := & \frac{8 C_f F_{max} x_2}{\pi x_3^3} - S \leq 0 \\ g_2 & := & l_f - l_{max} \leq 0 \\ g_3 & := & \sigma_p - \sigma_{pm} \leq 0 \\ g_4 & := & \sigma_w - \frac{F_{max} - F_p}{K} \leq 0 \end{array}$$

with

[1] The Rosenbrock function is indeed multimodal as soon the dimension is greater than three [5].

$$
\begin{aligned}
C_f &= 1 + 0.75\frac{x_3}{x_2 - x_3} + 0.615\frac{x_3}{x_2} \\
F_{max} &= 1000 \\
S &= 189000 \\
l_f &= \frac{F_{max}}{K} + 1.05\left(x_1 + 2\right)x_3 \\
l_{max} &= 14 \\
\sigma_p &= \frac{F_p}{K} \\
\sigma_{pm} &= 6 \\
F_p &= 300 \\
K &= 11.5 \times 10^6 \frac{x_3^4}{8x_1 x_2^3} \\
\sigma_w &= 1.25
\end{aligned}
$$

and the function to be minimised is

$$f = \pi^2 \frac{x_2 x_3^2 \left(x_1 + 2\right)}{4} \tag{3}$$

The best known solution is $(7, 1.386599591, 0.292)$, which gives the fitness value $f^* = 2.6254214578$. For the results given on the table 1, a penalty method has been used to take the constraints into account, but any method is of course acceptable. Here, we allow 2×10^4 fitness evaluations and a run is said to be successful if it finds a fitness f so that $|f - f^*| \leq 10^{-10}$. Because of the granularities this problem may be deceptive for some algorithms.

Gear train

For more details, see [4, 3]. The function to be minimised is

$$f\left(x\right) = \left(\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4}\right)^2 \tag{4}$$

The search space is $\{12, 13, \ldots, 60\}^4$. There are several solutions, depending on the required precision. Here, we used 10^{-13}. So, a possible solution is $f^* = f\left(19, 16, 43, 49\right) = 2.7 \times 10^{-12}$. We allow 2×10^4 fitness evaluations, and a run is then said successful if it finds a fitness f so that $|f - f^*| \leq 10^{-13}$. In this problem, only "integer" positions are acceptable. A lot of algorithms are not comfortable with such constraints.

Reasonable results

Some results with two PSO variants are given in the table 1. We say that an algorithm A "beats" an algorithm B it the mean success rate of A is greater than the one of B. Two cases:

- your algorithm does not beat the algorithm 1, i.e. SPSO-2007 (Standard PSO 2007). Forget it.

Table 1: Success rates over 100 runs

Algorithm / Function	1 SPSO-2007	2 Variable PSO (similar to SPSO-2007 but with bi-directional variable ring topology and variable swarm)
Tripod	50%	94%
Rosenbrock F6	82%	75%
Compression spring	35%	72%
Gear train	6%	22%
Mean	*43.25%*	*65.5%*

- it beats Algorithm 2. It is really promising. It is worth running it on a more complete benchmark. Be sure this benchmark is non biased (the solution point must not be on a diagonal of the search space)

References

[1] Maurice Clerc. *Particle Swarm Optimization.* ISTE (International Scientific and Technical Encyclopedia), 2006.

[2] Louis Gacôgne. Steady state evolutionary algorithm with an operator family. In *EISCI*, pages 373–379, Kosice, Slovaquie, 2002.

[3] Godfrey C. Onwubolu and B. V. Babu. *New Optimization Techniques in Engineering.* Springer, Berlin, Germany, 2004.

[4] E. Sandgren. Non linear integer and discrete programming in mechanical design optimization, 1990. ISSN 0305-2154.

[5] Yun-Wei Shang and Yu-Huang Qiu. A note on the extended rosenbrock function. *Evolutionary Computation*, 14(1):119–126, 2006.