Soft Computing manuscript No.
(will be inserted by the editor)

The Raven Roost Algorithm: A social foraging-inspired algorithm for optimisation

Anthony Brabazon · Wei Cui · Michael O'Neill

Received: date / Accepted: date

Abstract The foraging strategies of various organisms, including amongst others, ants, bacteria and honey bees, have served as the design inspiration for several families of optimisation algorithms. The success of these algorithms in a wide variety of appplication domains has spurred interest in the examination of the foraging behaviours of other organisms in order to further extend the range of naturally-inspired optimisation algorithms. A variety of animals, including some species of birds and bats, engage in *social roosting* whereby large numbers of conspecifics gather together to roost, either overnight or for longer periods. It has been posited that these roosts can serve as an 'information centre' to spread information concerning the location of food resources in the environment. In this paper we look at one example of social roosting, that of the common raven, and draw inspiration from this to design an optimisation algorithm. We also consider the role of individual perception in the foraging process. The utility of the resulting algorithms are assessed on a series of benchmark problems and the results are found to be very competitive. Potential for future work using the social roosting metaphor is also indicated.

A. Brabazon
Complex Adaptive Systems Laboratory and School of Business, University College Dublin, Dublin, Ireland
E-mail: anthony.brabazon@ucd.ie

W. Cui
Complex Adaptive Systems Laboratory and School of Business, University College Dublin, Dublin, Ireland
E-mail: wei.cui.c@gmail.com

M. O'Neill
Complex Adaptive Systems Laboratory and School of Computer Science and Informatics, University College Dublin, Dublin, Ireland
E-mail: m.oneill@ucd.ie

Keywords Social foraging · Social roosting · Perception · Optimisation

1 Introduction

When the location and quality of food resources are not known with certainty, organisms need to develop a search strategy in order to find them. Foraging success is vital for survival, and organisms with better-quality foraging strategies will be preferentially selected in the process of evolution. This observation has led to the development of a significant literature in computer science which takes inspiration from the foraging strategies of various organisms in order to design powerful search algorithms. Examples include ant colony optimisation algorithms (Bonabeau et al., 1999; Dorigo, 1992; Dorigo and DiCaro, 1999; Dorigo et al., 1996; Dorigo and Stützle, 2004), bacteria foraging algorithms (Passino, 2000, 2002), and honey bee algorithms (Chong et al., 2006; Nakrani and Tovey, 2004; Pham et al., 2006; Yang, 2005) to name but a few.

Real-world foraging behaviours are context-sensitive and depend on the nature of exploited resources as characterised by their location, size and quality (Deygout et al., 2010), the degree of competition for these resources, the predation risk faced whilst foraging, the locomotion capability of an organism, its sensory and cognitive capabilities, its daily energy requirements, the energy 'cost' of finding, subduing and digesting food resources, and the ability of the organism to store energy. As would be expected, there has been heavy interaction of these factors in evolutionary time. It is also plausible that advances in sensory capabilities and mobility have been driven, at least in part, by their adaptive impact on resource capture capability.

Another aspect of foraging is that it takes place in a dynamic environment as food source location, and quality, changes over time as a result of factors such as consumption and degradation as a result of environmental influences. This suggests that higher-quality food foraging strategies will need to be adaptive to changing conditions and to feedback based on the degree of past success. This underscores the importance of both lifetime and social learning based on feedback or via social influences such as the passing on of strategies to younger animals from their elders through observation and imitation.

The activity of food foraging can be individual where each individual in a species forages on its own (solitary foraging) or social, where foraging is a group behaviour. Social foraging, a subfield of behavioural ecology, has attracted substantial research interest (Giraldeau and Caraco, 2000; Stephens and Krebs, 1986; Viswanathan et al., 2011) and topics of interest include:

1. how do members of the group search for food,

2. how are food finds communicated to other members of the group,
3. and how are food finds divided up between members of the group.

The essence of social foraging is that there must be some communication between organisms resulting in recruitment to new food sources. Recruitment may occur at a communal nest or den (Sumpter and Brannstrom, 2008). Taking the two best-known families of foraging algorithms (ant and honey bee), in both cases the organisms are *central place foragers* in that they return to a colony or hive in order to store food and therefore they can interact with 'tribe' members and potentially pass on information about food finds. A key aspect of their interaction is that successful foragers seek to recruit other conspecifics to food resources that they have found. This 'interaction' may be direct or indirect, an example of the latter being provided by pheromone trail-marking by some species of ants.

Many other organisms also engage in social gathering and an as yet neglected area of the foraging literature in computer science is that related to *social roosting*. Several animals engage in social roosting including bats and birds and these roosts can potentially serve as an information centre leading to the exchange of information concerning the location of food resources. In this study, we examine one instance of social roosting and associated foraging behaviour, namely the nocturnal roosting behaviour of juvenile, common ravens, and design a series of 'raven-roost' optimisation algorithms drawing inspiration from this process.

1.1 Structure

The remainder of this contribution is organised as follows. Section 2 provides some background on raven roosting and foraging behaviours, and the associated *information centre hypothesis*. Section 3 outlines the design of the raven roost algorithm. In Section 4 we outline the experimental design, and present the results for each algorithm variant. Finally, conclusions and opportunities for future work are discussed in Section 5.

2 Raven Roosting

A social behaviour which is exhibited by some animals, including some bird and bat species, is 'roosting' where multiple animals come together to rest. This naturally leads to the question as to what are the advantages of this behaviour. Initial explanations centred on the possible anti-predatory benefits, increased opportunities for mate choice, enhanced care of young, increased opportunity for status display and thermal benefits during overnight roosting (Dall, 2002; Marzluff and Heinrich, 2001).

An alternative explanation, the *Information Center Hypothesis* (ICH), was proposed by Zahavi (1972); Ward and Zahavi (1973) and this suggested that birds join colonies and roosts in order to increase their foraging efficiency by means of the exchange of information regarding the location of food. The author(s) claimed that enhanced food foraging success was the primary reason for the evolution of gregariousness in birds.

The core tenent of the ICH is that birds which successfully find food advertise this fact at the roost site and are subsequently followed by several conspecifics to the food resource (i.e. they 'recruit' for the food resource). Interest in the ICH is not restricted to bird roosting behaviours. The possibility that information transfer could also occur in communally roosting bats was initially suggested by Fleming (1982). Later work by Wikinson (1992) examined the foraging behaviour of the bat species *Nycticeius humeralis* and found that unsuccessful foragers tend to follow previously successful foragers and that the foraging success of putative followers is greater than that of unsuccessful bats which depart solitarily. The author concluded that information transfer concerning good foraging sites was taking place, potentially via echolocation pulses, although the exact mechanism of information transfer was not isolated in the study.

Raven roosts consist of juvenile, non-breeding, unrelated common ravens. Ravens normally arrive at roosts shortly before sunset and typically leave the roost in highly synchronised groups at dawn the next day. The first comprehensive study of information transfer in raven roosts was undertaken by Marzluff et al. (1996) who examined roosting behaviours of the common raven (*Corvus corax*) in the forested mountains of Maine (USA). Ravens in this region are specialist feeders on the carcasses of large mammals in winter. These food sources are ephemeral as they degrade or are consumed quickly, and the location of carcasses is unpredictable. Hence, the search for food resources is continuous. The typical food discovery process observed commenced with a small number of birds feeding at a carcass site, followed by a rapid (overnight) doubling in numbers with most of these birds arriving simultaneously at dawn. The carcass would be consumed over several days and at the final stage of carcass depletion, feeding group size declined rapidly as many birds left the carcass in the afternoon (prior to sunset) and did not return to it the next day.

Marzluff et al. (1996) undertook careful observation at both roosting and foraging sites, and monitored the change in the number of ravens at a carcass from one day to the next. The number of birds at a carcass at different times of the day was also monitored in order to control for local enhancement effects during a day. Control experiments, wherein naive birds (birds with no knowledge of the location of food locations) were released at roosts demonstrated that naive birds found feeding locations by following their roost mates, providing evidence for the existence of information sharing. Observations by the authors also indicated that the same individuals in a roost are not always knowledgable, suggesting that information sharing rather than mere parasitism (wherein 'excellent' foragers were simply followed by less-skilled conspecifics) was taking place. The study concluded that information sharing did take place at roosts and that ravens which successfully found a new food resource recruited other members of the roost to that resource.

These findings were extended in a study by Wright et al. (2003) which examined the behaviour of ravens in a large roost in North Wales in the United Kingdom. In contrast to North America, raven roosts in Europe are typically larger and have more stable membership. The researchers deposited baited carcasses at various locations around the roost and found that most carcasses were consumed within five days. Observational evidence suggested that recruitment started for each carcass via a single bird on day zero with subsequent recruitment of about six to seven birds per day. Birds that first discovered the baited carcasses recruited conspecifics using pre-roost (evening) acrobatic flight displays and vocalisations. The 'discoverer' birds spent the night surrounded in the roost by the group that would follow them out the next morning to the food source. Recruitment appeared to be a competitive activity which was more successful for geographically closer carcasses, consistent with the idea that the pre-roost displays accurately reflected the energetic state of the displaying bird and therefore the relative distance and profitability of the carcass discovered. The authors considered that the findings 'provide strong circumstantial evidence for raven roosts as structured information centres'.

In a similar fashion to the foraging activities of ants and bees, socially roosting birds can be influenced by means of social learning through a recruitment mechanism. Of course, the foraging process of ravens (and indeed of social insects) has a number of additional complexities. A bird may have private knowledge of food resource locations and hence may place different weight on socially-acquired information depending on its private knowledge. This is also true of honey bees (Wray et al., 2011), although this issue has attracted little attention thus far in the honey bee optimisation literature. In addition, birds can survey a wide terrain whilst in flight and may decide to deviate from a 'follower' behaviour if an alternative food source is seen whilst in flight.

In the following sections we undertake a number of experiments which examine the utility of a series of optimisation algorithms which are inspired by a social roosting process, which incorporates both social and private information. We also consider the implications of embedding an individual 'perception' process in the algorithms. Considering these issues could potentially provide some interesting insights into high-level foraging behaviours as well as assisting in the design of efficient optimisation algorithms.

3 Model Development

In the proposed raven roosting optimisation (RRO) algorithm, a randomly-located (in the input space) roosting site is chosen. The roost location is then fixed for the remainder of the algorithm. Initially, each of the population of N 'ravens' are placed at random locations in the search space. Each of these locations corresponds to a potential food resource location. Next, the fitness values of the N locations is assessed, and the location of the best solution is denoted as "LEADER".

The roosting process is then simulated by mimicking an information-sharing step. As in real-world raven roosting, only a portion of the roost members will be recruited to a new food source and other roost members will continue to return to a 'private' food location or to continue with solitary trial and error search for food. A proportion of the ravens ($Perc_{follow}$) are recruited to leave the roost and follow the LEADER. Mimicking noisy recruitment and personal perception, a recruited follower is able to 'perceive' a hypersphere of radius (R_{leader}) around the LEADER and may therefore forage at another location in this region, if a better location is seen by them. On leaving the roost, unrecruited birds travel to the best location that they have found to date (their personal best) and continue to forage there. The inclusion of a personal best 'memory' for each bird embeds a concept of 'private information' as unrecruited birds in essence are choosing to rely on private information rather than piggy-back on socially-broadcast information from the LEADER.

Whilst in flight to the intended destination, individual birds maintain a search for new food sources en-route. We simulate this process by dividing their flight into N_{steps}. The length of each step is chosen randomly, and the bird's position in flight is updated using the following equation:

$$p_{i,t} = p_{i,t-1} + d_{i,t} \tag{1}$$

where $p_{i,t}$ is the current position of the i^{th} raven, $p_{i,t-1}$ is its previous position, and $d_{i,t}$ is a random step size. At each step, a raven senses its surrounding environment in the range of radius R_{pcpt}, and makes N_{pcpt} random perceptions within this hypersphere. If a better location is perceived than the bird's personal best, there is $Prob_{stop}$ chance that the raven stops its flight at that point and forages at the newly-found location; otherwise, it takes another step and continues to fly to its destination. At the conclusion of the algorithm, the fittest location found is returned. Pseudocode for the algorithm is provided in Algorithm 1.

In our experiments we design and test 13 variants of the raven roosting optimisation algorithm in order to undertake a comprehensive analysis of the relative importance of different parameter settings and of each of the key mechanisms

Algorithm 1: Raven Roosting Optimisation Algorithm

```
Randomly select a roosting site;
repeat
    The $N$ foraging ravens are assigned to $N$ random locations in the search space;
    Evaluate the fitness of each raven location;
    Update the personal best location of each raven;
    The location of the best solution is denoted as LEADER;
    Recruit $Perc_{follow}$ percentage of the $N$ foragers from the roosting site which will search in the vicinity of the LEADER (within the range of radius $R_{leader}$) and the rest of the ravens will seek to travel to their personal best locations;
    Set $step = 0$;
    while $step < N_{steps}$ do
        On the way to its destination (whether the destination is the LEADER's vicinity or the personal best location), each raven flies for a while and searches in the vicinity of its current position (within the range with radius $R_{pcpt}$);
        if a better solution is found then
            There is a $Prob_{stop}$ chance the raven will stop;
            Update the personal best location;
        else
            It continues to fly;
        end
        $step = step + 1$;
    end
    For the ravens which finally arrive their destinations (the LEADER's vicinity or the personal best), update their personal best locations if necessary;
    Update location of the best solution found so far if necessary;
until terminating condition;
```

Table 1 Parameter Setting of Algorithms

Algorithm	**Radius of Perception** (R_{pcpt})	**Radius of Leader** (R_{leader})	**Perceptions** (N_{pcpt})	**Steps** (N_{steps})	**Proportion of Followers** ($Perc_{follow}$)	**Probability of Stopping** ($Prob_{stop}$)
RRO0	$\frac{R}{3.6\sqrt[D]{N}}$	$\frac{R}{3.6\sqrt[D]{N}}$	10	10	0.2	0.1
RRO1	$\frac{R}{3.6\sqrt[D]{N}}$	$\frac{R}{1.8\sqrt[D]{N}}$	10	10	0.2	0.1
RRO2	$\frac{R}{1.8\sqrt[D]{N}}$	$\frac{R}{3.6\sqrt[D]{N}}$	10	10	0.2	0.1
RRO3	$\frac{R}{1.8\sqrt[D]{N}}$	$\frac{R}{1.8\sqrt[D]{N}}$	10	10	0.2	0.1
RRO4	$\frac{R}{3.6\sqrt[D]{N}}$	$\frac{R}{3.6\sqrt[D]{N}}$	5	10	0.2	0.1
RRO5	$\frac{R}{3.6\sqrt[D]{N}}$	$\frac{R}{3.6\sqrt[D]{N}}$	20	10	0.2	0.1
RRO6	$\frac{R}{3.6\sqrt[D]{N}}$	$\frac{R}{3.6\sqrt[D]{N}}$	10	5	0.2	0.1
RRO7	$\frac{R}{3.6\sqrt[D]{N}}$	$\frac{R}{3.6\sqrt[D]{N}}$	10	20	0.2	0.1
RRO8	$\frac{R}{3.6\sqrt[D]{N}}$	$\frac{R}{3.6\sqrt[D]{N}}$	10	10	0.4	0.1
RRO9	$\frac{R}{3.6\sqrt[D]{N}}$	$\frac{R}{3.6\sqrt[D]{N}}$	10	10	0.6	0.1
RRO10	$\frac{R}{3.6\sqrt[D]{N}}$	$\frac{R}{3.6\sqrt[D]{N}}$	10	10	0.8	0.1
RRO11	$\frac{R}{3.6\sqrt[D]{N}}$	$\frac{R}{3.6\sqrt[D]{N}}$	10	10	0.2	0.2
RRO12	$\frac{R}{3.6\sqrt[D]{N}}$	$\frac{R}{3.6\sqrt[D]{N}}$	10	10	0.2	0.4

Note: R is the radius of the search space.
D is the dimensionality of the test problem.
N is the number of ravens (in RRO) or particles (in PSO).

in the algorithm. Details of these variants are set out in Table 1 and are discussed in detail in the next section.

4 Results and Discussion

In this section we describe the experiments undertaken and present the results from these experiments. Four standard

Table 2 Optimisation Problems

Name	Function	Search Space	Optima
DeJong	$F(x) = \sum_{i=1}^{n} x_i^2$	$[-5.12\ 5.12]^n$	0.0^n
Griewank	$F(x) = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}})$	$[-600\ 600]^n$	0.0^n
Rastrigin	$F(x) = 10n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)]$	$[-5.12\ 5.12]^n$	0.0^n
Rosenbrock	$F(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	$[-30\ 30]^n$	1.0^n

benchmark problems (Table 2), at three levels of dimensionality (20, 40 and 60), were used to test the developed algorithms. Two of these functions namely, DeJong and Rosenbrock, represent unimodal problems; and the other two, Griewank and Rastrigin, are more complex multimodal functions which contain multiple local optima. The aim in all the experiments is to find the vector of values which minimise the value of the test functions.

4.1 Overview of Experiments

In the first set of experiments we undertake a proof of concept study to assess the performance of the canonical version of the algorithm (denoted as RRO0) on the four benchmark problems. This is then followed by an investigation of the importance of the perception mechanism in the RRO algorithm. Accordingly, we test a variant of the RRO algorithm in which the perception mechanism is switched off (this variant of the RRO is denoted as RROv) and compare its performance against that of the canonical RRO algorithm which has the perception mechanism.

The second set of experiments examine the sensitivity of the canonical RRO to changes in the values of six of its parameters. In each case, we select two or three values for each of these parameters, producing 12 variants of the RRO algorithm (denoted as RRO1-RRO12). The specific values of the parameters for each algorithm variant are set out in Table 1. Whilst these are chosen judgementally, the values chosen for the two radii (R_{pcpt} & R_{leader}) are problem-specific, as they are influenced by the choice of the number of ravens (N), the radius (size) of the search space (R), and the dimensionality of this space (D). In this study, the values of R_{pcpt} and R_{leader} were chosen after initial experimentation as $\frac{R}{3.6\sqrt[D]{N}}$ and $\frac{R}{3.6\sqrt[D]{N}}$.

Finally, the results from RRO are compared against those of canonical Particle Swarm Optimisation (PSO) a powerful and well-known optimisation heuristic.

In each experiment, 30 ravens in the case of RRO, or 30 particles in the case of PSO, are used. In order to allow a reasonably fair comparison, we adjust the the number of algorithm iterations as necessary in order to equalise the number of fitness function evaluations across all experiments. The experimental parameters are shown in Tables 3&4. All reported results are averaged over 30 experiments for each problem and algorithm, and we test the statistical significance of all differences in performance at a conservative 99% level using a t-test. The experiments were undertaken on an Intel Core i7 (2.93 GHz) system with 12 GB RAM.

Table 4 Parameter Setting of Experiments

Parameters	Values
Trials	30
Size of Population	$N = 30$
Dimension of Problem	$D = 20, 40, 60$

4.2 Hypotheses

The first set of experiments concern the testing of the importance of the perception mechanism of the RRO. The null hypothesis is that the algorithm with perception mechanism turned off (RROv) performs better than the canonical RRO (RRO0). Therefore the following hypothesis is tested.

- H_{v_0}: the RROv algorithm outperforms the RRO0 algorithm;

The next set of hypotheses concern the analysis of the performance of the 12 variants of RRO with different parameter settings (RRO1-RRO12) against the performance of the canonical algorithm (RRO0). In this case, the null hypotheses are that the variant algorithm outperform the canonical algorithm.

- H_{i_0}: the RROi ($i : 1 - 12$) algorithm outperforms the RRO0 algorithm;

The final set of hypotheses concern the analysis of the performance of the various versions of RRO against the performance of PSO. The null in each case is that PSO performs better.

Table 3 Iteration Settings for Each Trial

RRO0	RRO1	RRO2	RRO3	RRO4	RRO5	RRO6	RRO7	RRO8	RRO9	RRO10	RRO11	RRO12	RROv	PSO
500	500	500	500	1000	250	1000	250	500	500	500	500	500	5000	60000

- H_0: the PSO algorithm outperforms the RRO0 algorithm;
- H_v: the PSO algorithm outperforms the RROv algorithm;
- H_i: the PSO algorithm outperforms the RROi ($i : 1-12$) algorithm.

4.3 Analysis of Perception Mechanism in RRO

In order to assess the effect of the perception mechanism on the performance of the RRO algorithm, two algorithms, RRO0 with a perception mechanism and RROv without a perception mechanism, were tested on the four optimisation problems (shown in Table 2) using 20, 40, and 60 dimensions respectively. The performance of both algorithms were also compared with that of the canonical PSO.

Figures 1&2 compare the average fitness of the three algorithms for the tested problems, and Tables 5-7 show the best fitness value obtained from all 30 runs (‘Best’), and the average of the best fitnesses (‘Mean’) and its standard deviation over all 30 runs. The results show that the standard RRO algorithm (RRO0) outperforms the RROv algorithm on all problems, and that in ten out of twelve of the test instances (four test problems at three levels of dimensionality) the difference is statistically significant (the relevant p-values are shown in Table 7). Comparing the RRO0 algorithm against PSO, the results show that the RRO0 outperforms PSO on all problems, and that in eleven out of twelve cases the difference is statistically significant. In the case of RROv, it outperforms PSO on ten of the twelve problems and the difference is statistically significant in nine out of the ten.

Hence, we conclude that the perception mechanism is an important component of the RRO algorithm, and that the canonical RRO algorithm is highly-competitive against PSO, outperforming it on all cases.

4.4 Parameter Sensitivity Analysis

Figures 3-12 show the results of sensitivity analysis of the six parameters of RRO respectively, and Tables 5-7 show the best fitness value obtained from all 30 runs (‘Best’), and the average of the best fitnesses (‘Mean’) and their standard deviation over all 30 runs.

Columns H_{i_0} (i:1-12) in Tables 5-7 show the p-values of the statistical tests used to determine whether there are any differences in mean performance between the RRO0 algorithm and the other variants of RRO algorithm.

Columns H_0-H_{12} in Tables 5-7 show the p-values of the statistical tests used to test whether there is any difference between the performance of PSO algorithm and the RRO algorithms.

4.4.1 Impact of Varying Perception Radius

Figures 3 & 4 compare the performance of the canonical version of RRO (RRO0) with three variants (RRO1-3) which have larger perception radii for R_{pcpt} and/or R_{leader}. In general, across most problem instances, the performance ranking across the four algorithm variants is as follows:

$$RRO0, 1 > RRO2, 3 > PSO$$

Little difference is noted between the performance of RRO0 and RRO1, or between RRO2 and RRO3 on the various problem instances. This indicates that the performance of the RRO algorithm is sensitive to the changes on parameter R_{pcpt}, and not as sensitive to the changes in the parameter R_{leader}. Comparing the performances of RRO1, RRO2, RRO3 against that of PSO, all are found to outperform PSO, and the difference in mean performance are statistically significant in virtually all problem instances.

4.4.2 Impact of Varying Number of Perception Samples

Next we consider the parameter which governs the number of perception samples that the ravens can utilise. In essence, this proxies elements of the animal’s cognitive processing ability, as in addition to the radius of sensory perception being finite, assessments of resource quality within the range of sensory perception are likely to be imperfect due to time and cognitive limitations. In RRO0, ten samplings are made in each ‘perception’, and in algorithm variants RRO4 and RRO5 this number is altered to 5 and 20 respectively (note, total number of fitness function evaluations is held constant across all experiments). Figures 5&6 show that the performances of the three algorithms, RRO0, RRO4 and RRO5, are similar over all problem instances with no clear evidence that increasing or decreasing the number of samplings (within the range tested) makes a notable difference. This suggests that the RRO algorithm is not highly sensitive to the changes in the parameter N_{pcpt}. Both RRO4 and RRO5 outperform PSO on all problem instances, significantly so on eleven of the twelve instances.

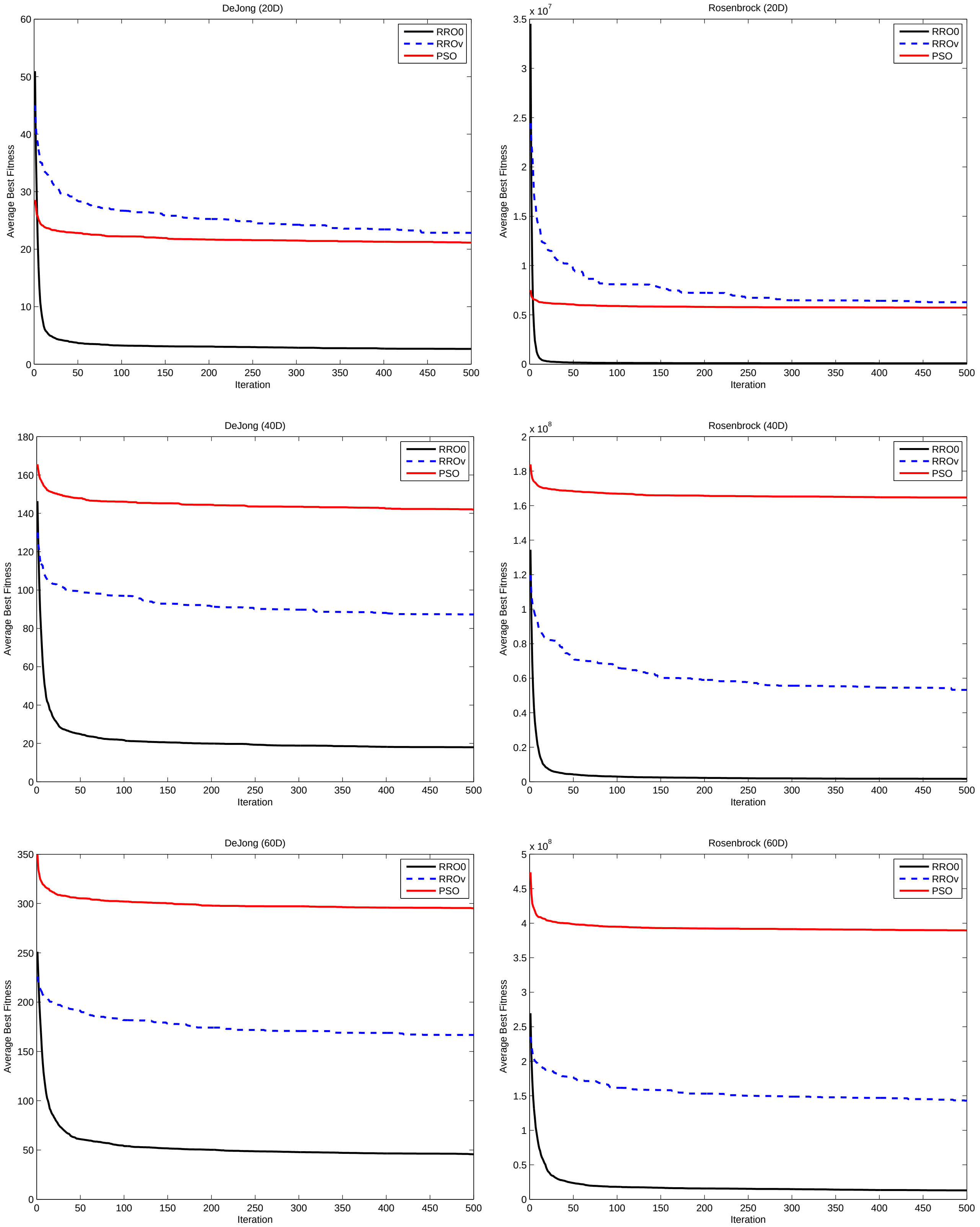

Fig. 1 Component Analysis I (DeJong & Rosenbrock Functions). The x-axis only shows the iterations for testing the RRO0 algorithm.

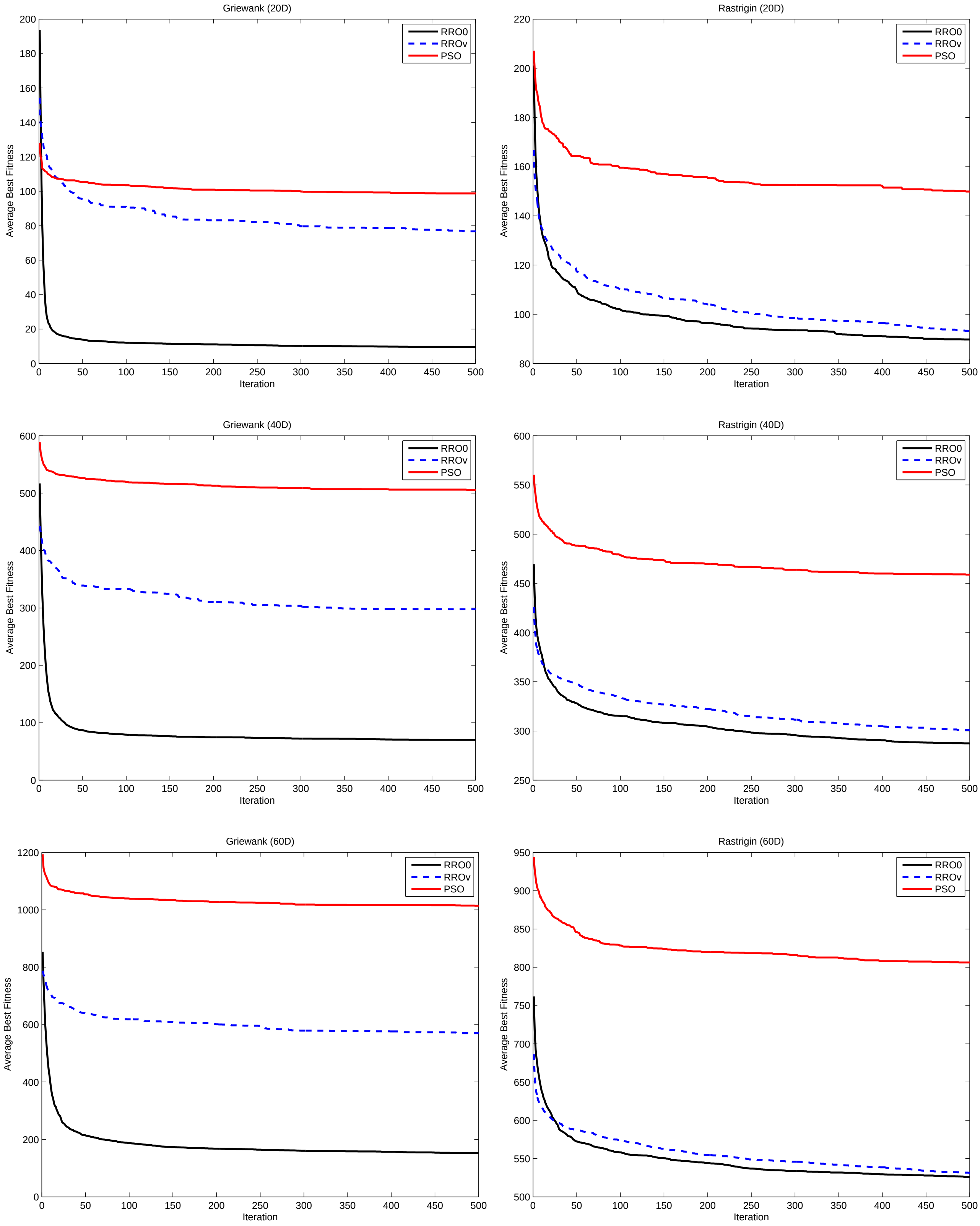

Fig. 2 Component Analysis II (Griewank & Rastrigin Functions). The x-axis only shows the iterations for testing the RRO0 algorithm.

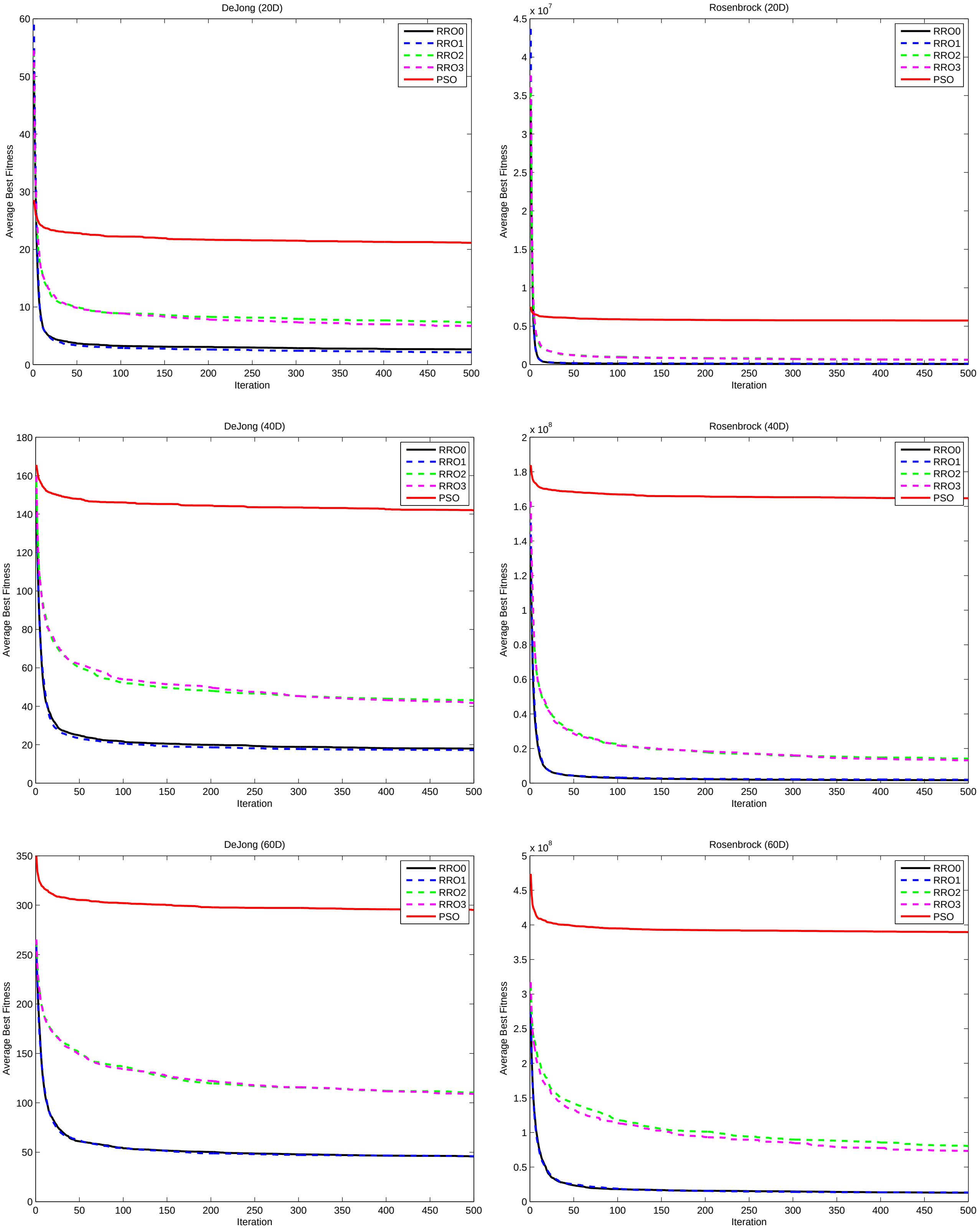

Fig. 3 Sensitivity Analysis I of Parameters R_{pcpt} & R_{leader} (DeJong & Rosenbrock Functions). The x-axis only shows the iterations for testing the RRO algorithms.

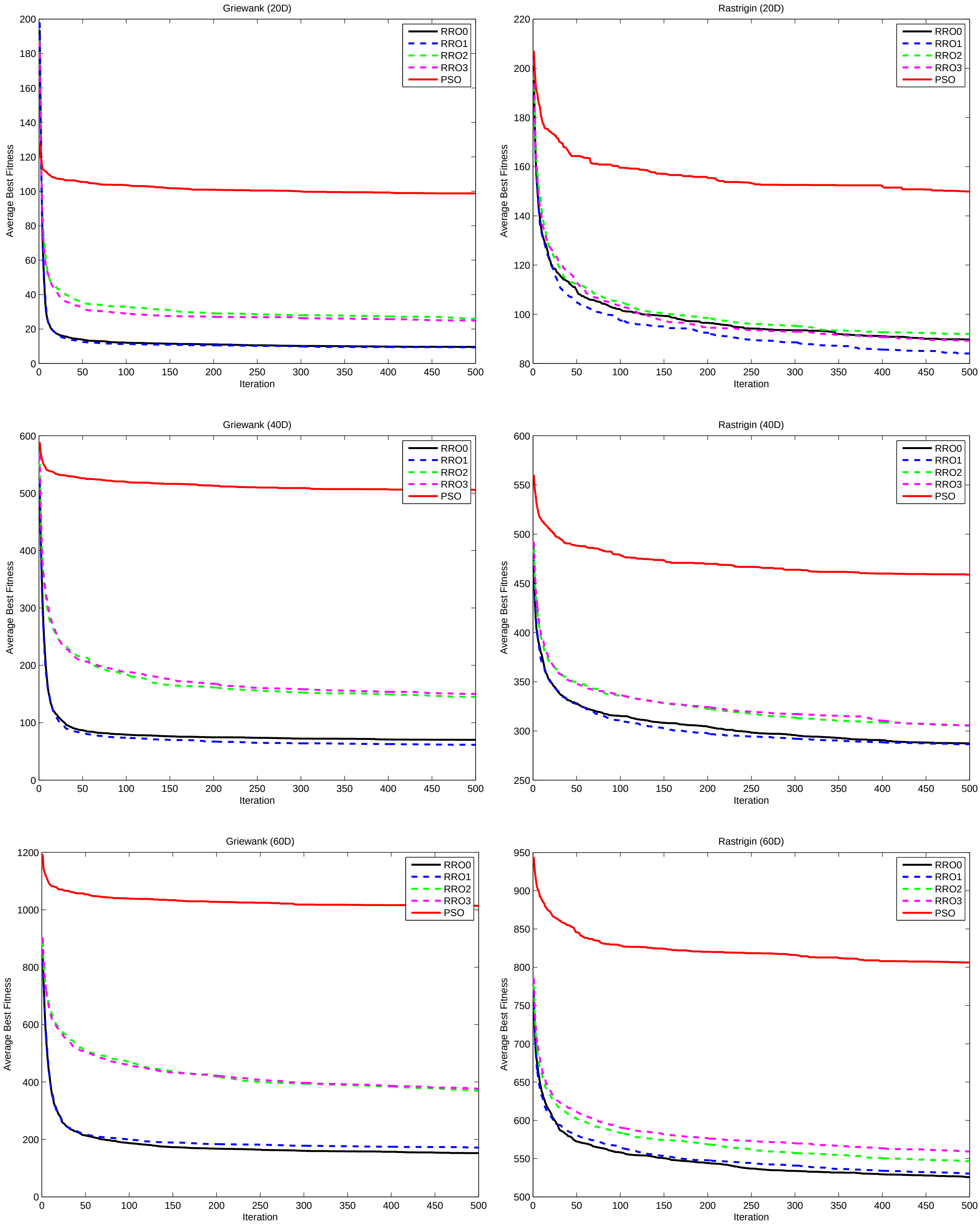

Fig. 4 Sensitivity Analysis II of Parameters R_{pcpt} & R_{leader} (Griewank & Rastrigin Functions). The x-axis only shows the iterations for testing the RRO algorithms.

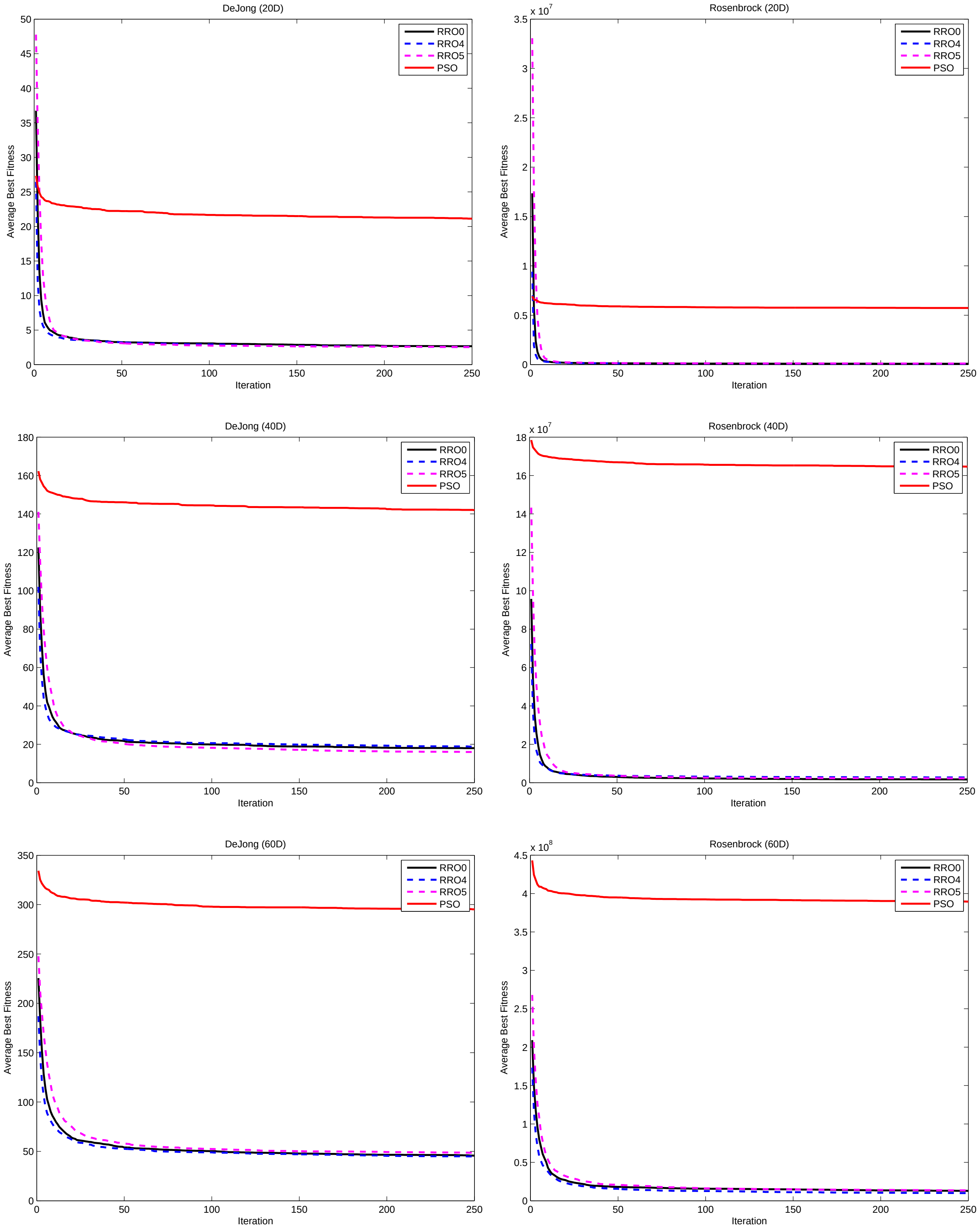

Fig. 5 Sensitivity Analysis I of Parameter N_{pcpt} (DeJong & Rosenbrock Functions). The x-axis only shows the iterations for testing the RRO5 algorithm.

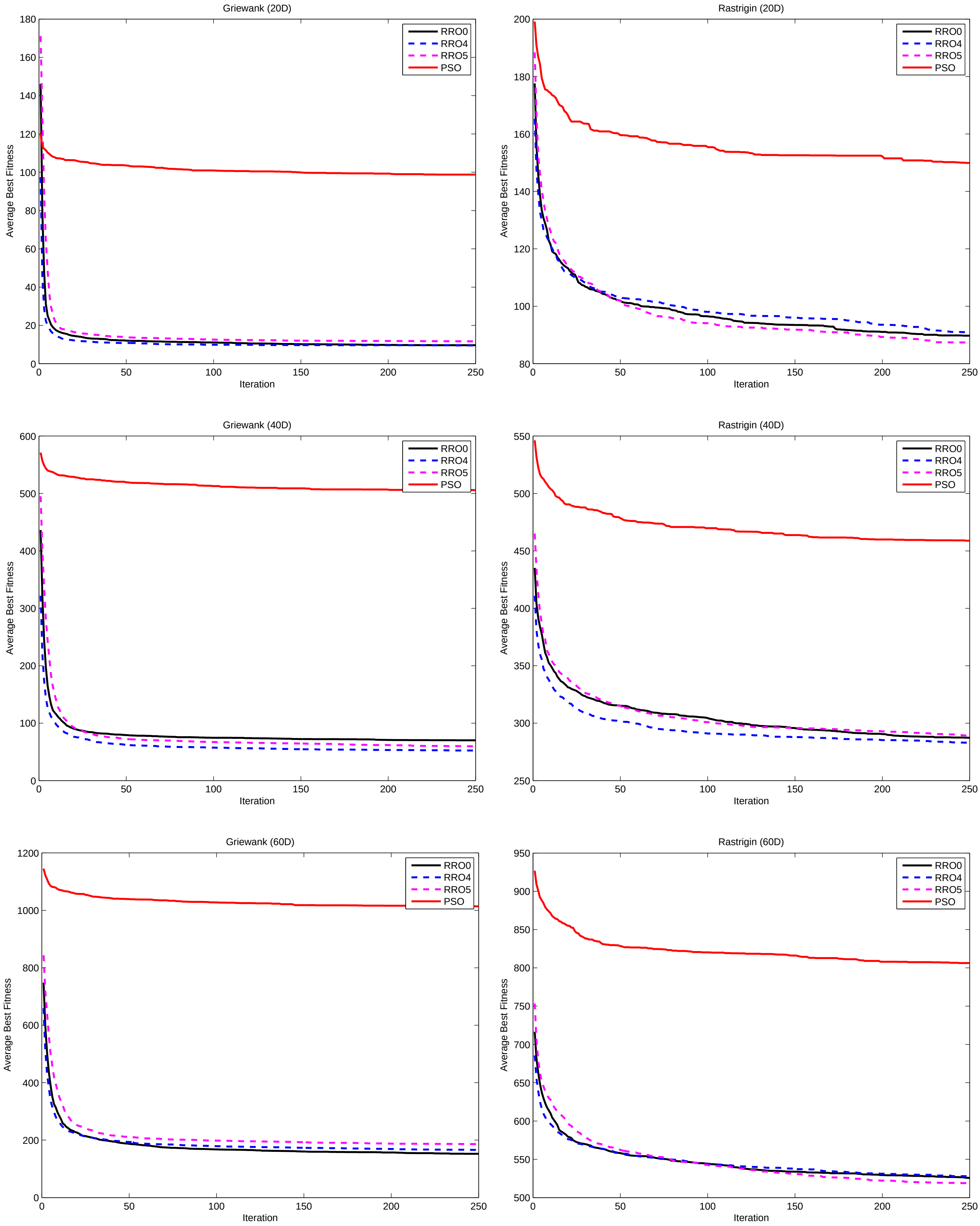

Fig. 6 Sensitivity Analysis II of Parameter N_{pcpt} (Griewank & Rastrigin Functions). The x-axis only shows the iterations for testing the RRO5 algorithm.

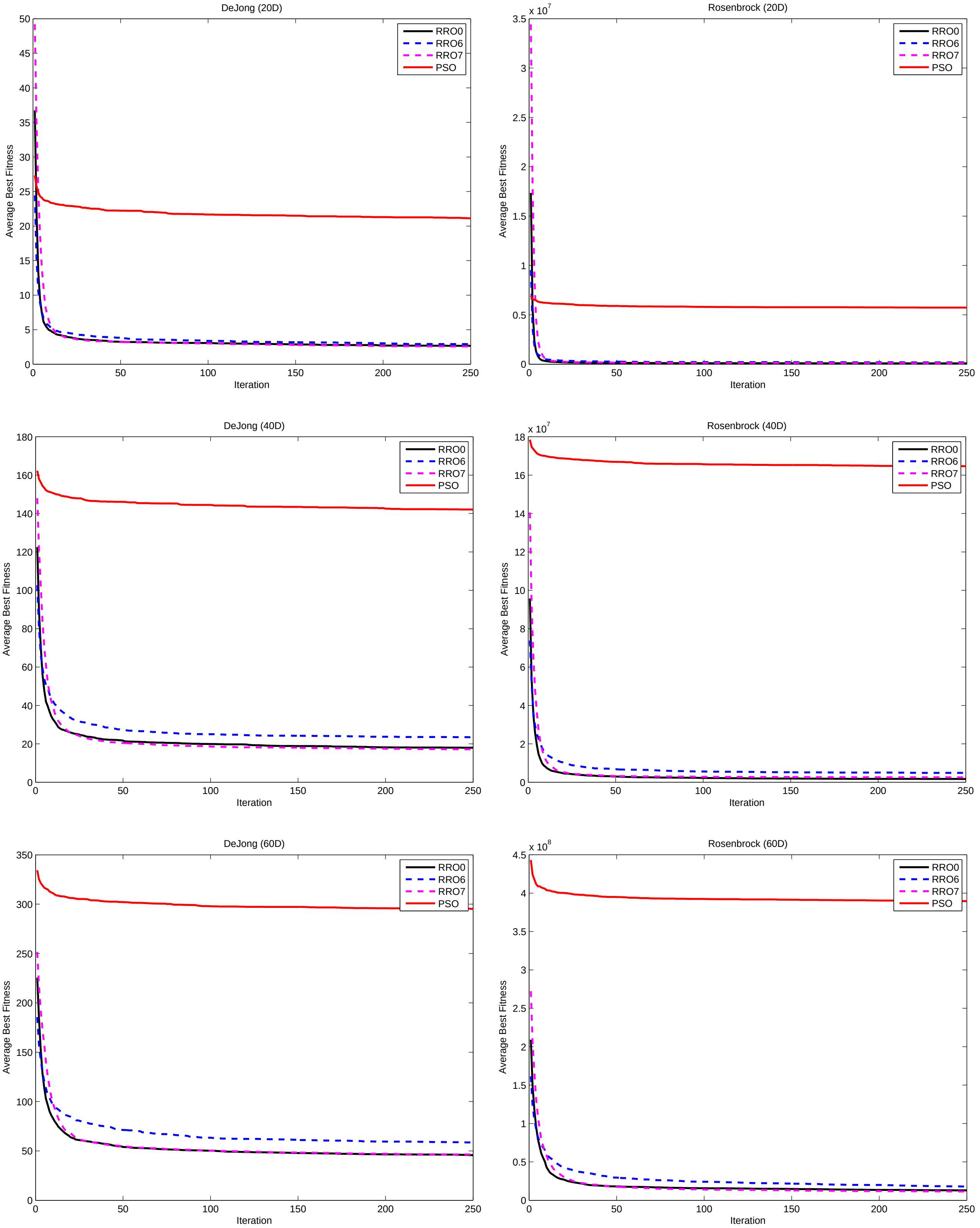

Fig. 7 Sensitivity Analysis I of Parameter N_{steps} (DeJong & Rosenbrock Functions). The x-axis only shows the iterations for testing the RRO7 algorith.

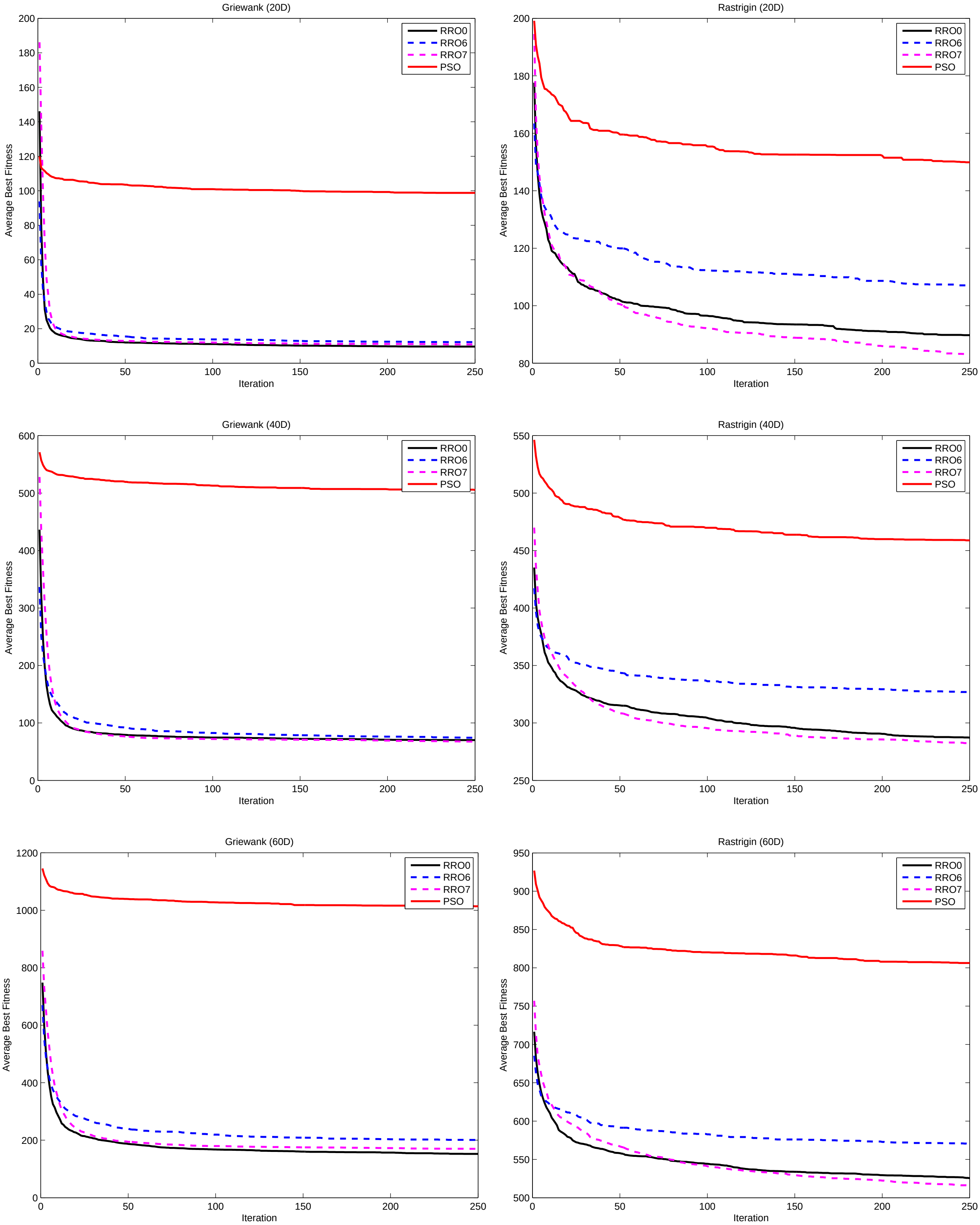

Fig. 8 Sensitivity Analysis II of Parameter N_{steps} (Griewank & Rastrigin Functions). The x-axis only shows the iterations for testing the RRO7 algorithm.

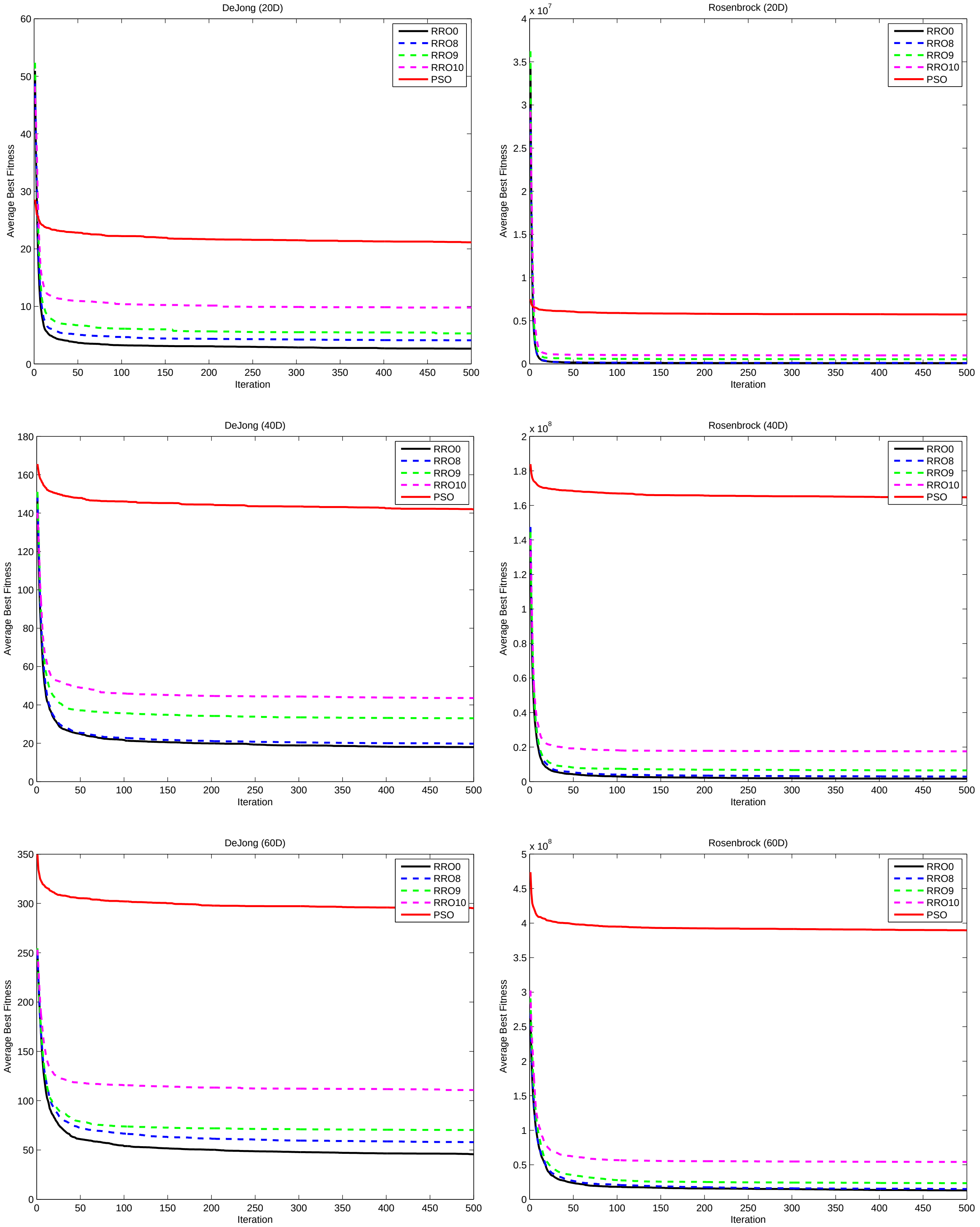

Fig. 9 Sensitivity Analysis I of Parameter $Perc_{follow}$ (DeJong & Rosenbrock Functions). The x-axis only shows the iterations for testing the RRO algorithms.

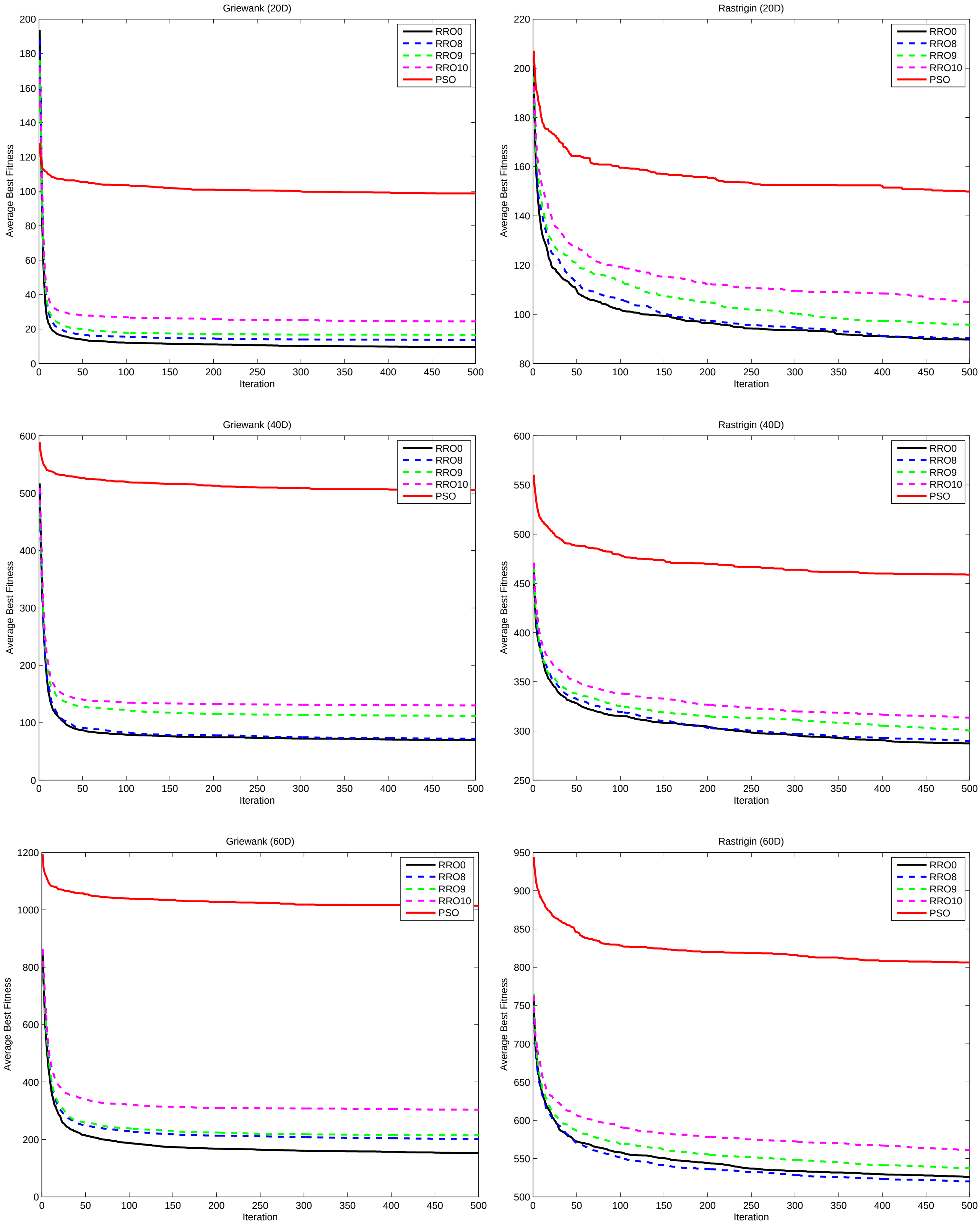

Fig. 10 Sensitivity Analysis II of Parameter $N_{Perc_{follow}}$ (Griewank & Rastrigin Functions). The x-axis only shows the iterations for testing the RRO algorithms.

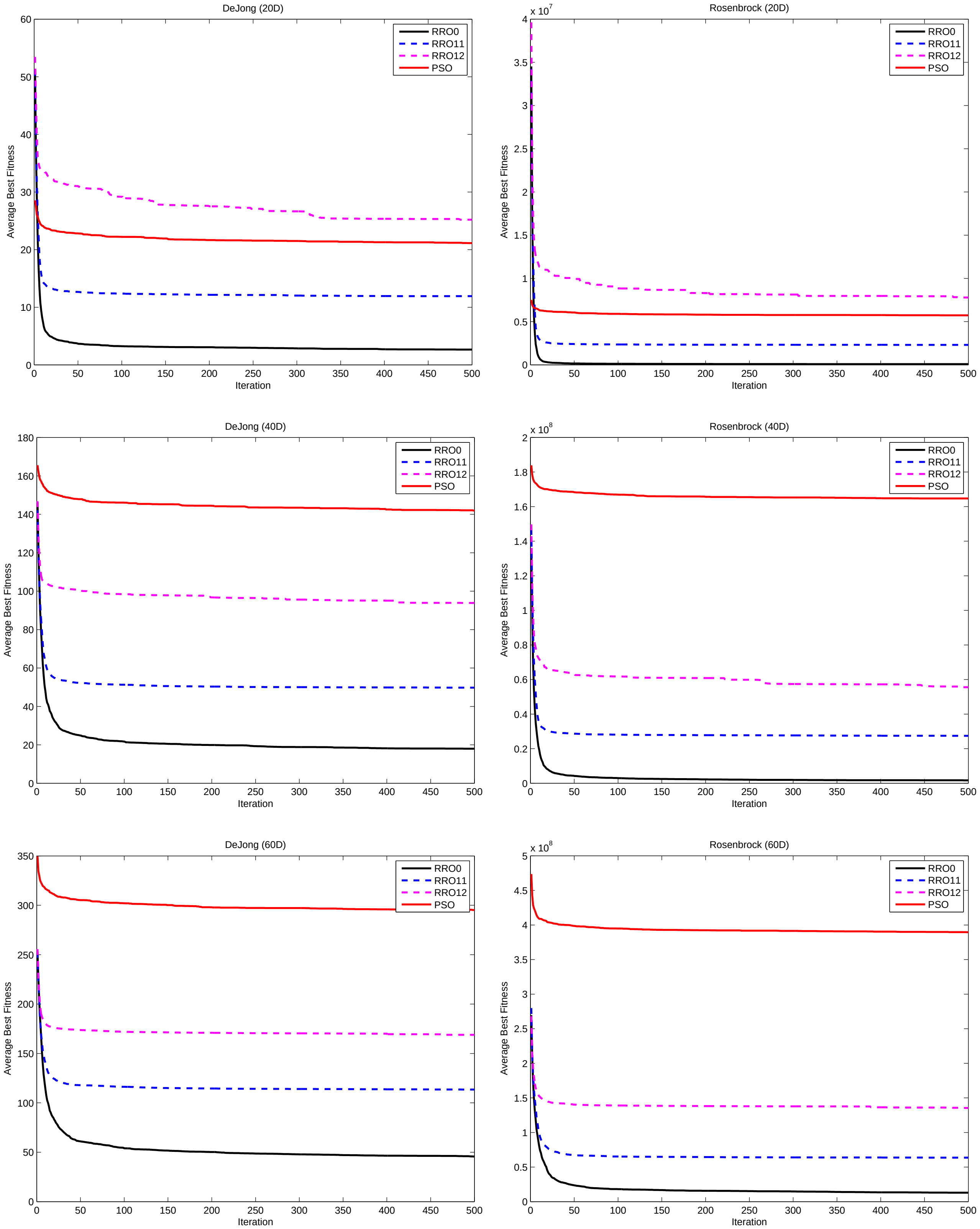

Fig. 11 Sensitivity Analysis I of Parameter $Prob_{stop}$ (DeJong & Rosenbrock Functions). The x-axis only shows the iterations for testing the RRO algorithms.

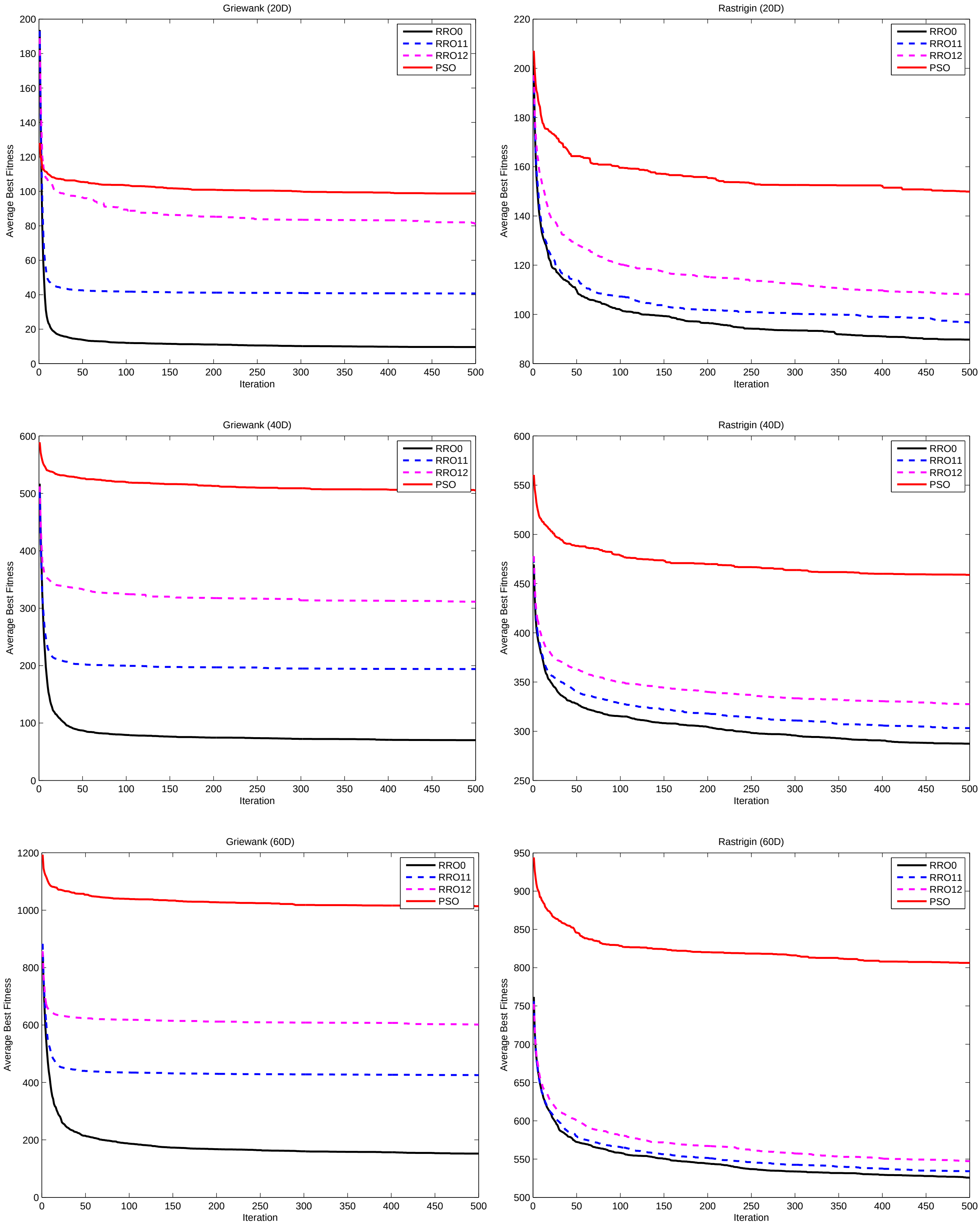

Fig. 12 Sensitivity Analysis II of Parameter $Prob_{stop}$ (Griewank & Rastrigin Functions). The x-axis only shows the iterations for testing the RRO algorithms.

Table 5 Results of Algorithm Comparison (I)

Function		**PSO**		**RRO0**			**RRO1**				**RRO2**				**RRO3**			
		Best	**Mean (Std.)**	**Best**	**Mean (Std.)**	H_0	**Best**	**Mean (Std.)**	H_1	H_{1_0}	**Best**	**Mean (Std.)**	H_2	H_{2_0}	**Best**	**Mean (Std.)**	H_3	H_{3_0}
DeJong	**20D**	4.2752	21.1347 (20.7797)	1.6065	2.6663 (1.4325)	**0.00**	**1.1565**	**2.1496** (**0.5512**)	**0.00**	0.07	5.2718	7.3157 (1.2992)	**0.00**	**0.00**	4.6700	6.7133 (1.3111)	**0.00**	**0.00**
	40D	75.4903	141.9867 (41.1804)	9.5379	17.9804 (11.0417)	**0.00**	10.1402	17.1491 (7.4220)	**0.00**	0.73	26.9314	43.2206 (5.5639)	**0.00**	**0.00**	29.0524	41.7310 (**5.1826**)	**0.00**	**0.00**
	60D	211.9589	295.2266 (43.8444)	30.2088	45.7234 (16.6323)	**0.00**	30.4605	45.9524 (17.8679)	**0.00**	0.95	88.6984	110.2546 (8.8859)	**0.00**	**0.00**	94.4696	108.9824 (**7.84617**)	**0.00**	**0.00**
Griewank	**20D**	15.9867	98.7569 (53.9338)	**5.7483**	9.6480 (4.9005)	**0.00**	6.1322	**9.3319** (3.4726)	**0.00**	0.77	17.7677	26.0183 (4.2394)	**0.00**	**0.00**	12.7696	24.8437 (5.4425)	**0.00**	**0.00**
	40D	187.5288	505.3347 (139.6617)	38.3486	70.2330 (32.0062)	**0.00**	37.8556	61.5970 (25.1599)	**0.00**	0.25	111.4742	145.0010 (17.2393)	**0.00**	**0.00**	123.1153	150.0776 (**13.6288**)	**0.00**	**0.00**
	60D	697.1503	1013.9464 (159.2638)	**90.9474**	**152.2529** (60.6620)	**0.00**	103.4983	171.2382 (86.6312)	**0.00**	0.32	303.8749	369.3153 (**25.1409**)	**0.00**	**0.00**	302.2994	375.2835 (30.9475)	**0.00**	**0.00**
Rastrigin	**20D**	111.4707	149.9047 (23.3029)	60.3031	89.7571 (9.1124)	**0.00**	64.5716	84.0321 (12.6712)	**0.00**	0.04	63.9981	91.9586 (10.8918)	**0.00**	0.39	72.6854	89.2459 (**8.1506**)	**0.00**	0.81
	40D	348.0436	458.8609 (49.0941)	239.5747	287.2116 (20.0619)	**0.00**	241.3647	286.5220 (21.0780)	**0.00**	0.89	267.2119	305.2914 (19.7228)	**0.00**	**0.00**	244.7283	305.7325 (23.4687)	**0.00**	**0.00**
	60D	687.6921	806.2230 (48.6207)	474.5812	525.4923 (28.8488)	**0.00**	482.7234	530.4105 (**23.3308**)	**0.00**	0.47	480.8923	546.6167 (28.4400)	**0.00**	**0.00**	506.3305	559.2304 (27.7060)	**0.00**	**0.00**
Rosenbrock	**20D**	0.0022E+8	0.0573E+8 (0.2028E+8)	**0.0002E+8**	**0.0008E+8** (0.0004E+8)	0.13	0.0003E+8	0.0012E+8 (0.0015E+8)	0.13	0.16	0.0029E+8	0.0064E+8 (0.0023E+8)	0.17	**0.00**	0.0031E+8	0.0062E+8 (0.0022E+8)	0.17	**0.00**
	40D	0.0410E+8	1.6465E+8 (1.0408E+8)	0.0079E+8	**0.0169E+8** (**0.0092E+8**)	**0.00**	0.0066E+8	0.0195E+8 (0.0145E+8)	**0.00**	0.41	0.0790E+8	0.1409E+8 (0.0288E+8)	**0.00**	**0.00**	0.0641E+8	0.1316E+8 (0.0233E+8)	**0.00**	**0.00**
	60D	1.7760E+8	3.8955E+8 (1.2999E+8)	0.0421E+8	0.1288E+8 (0.1136E+8)	**0.00**	0.0399E+8	0.1323E+8 (0.0956E+8)	**0.00**	0.89	0.5535E+8	0.8037E+8 (0.1121E+8)	**0.00**	**0.00**	0.4396E+8	0.7330E+8 (0.1410E+8)	**0.00**	**0.00**

Table 6 Results of Algorithm Comparison (II)

Function		**RRO4**				**RRO5**				**RRO6**				**RRO7**				**RRO8**			
		Best	**Mean (Std.)**	H_4	H_{4_0}	**Best**	**Mean (Std.)**	H_5	H_{5_0}	**Best**	**Mean (Std.)**	H_6	H_{6_0}	**Best**	**Mean (Std.)**	H_7	H_{7_0}	**Best**	**Mean (Std.)**	H_8	H_{8_0}
DeJong	**20D**	1.4991	2.5669 (1.1619)	**0.00**	0.76	1.5891	2.5454 (0.8687)	**0.00**	0.69	1.5088	2.9136 (1.0286)	**0.00**	0.44	1.2010	2.5728 (1.4167)	**0.00**	0.80	1.4606	4.1146 (3.9197)	**0.00**	0.06
	40D	**9.0582**	18.8028 (10.8095)	**0.00**	0.77	10.8285	**15.9968** (6.3378)	**0.00**	0.39	16.2316	23.4681 (9.0448)	**0.00**	0.03	10.8245	17.1935 (8.6807)	**0.00**	0.76	11.1705	19.7799 (13.4182)	**0.00**	0.57
	60D	28.8283	**44.9532** (17.2981)	**0.00**	0.86	30.3544	48.6439 (23.0164)	**0.00**	0.57	36.3137	58.5603 (16.7095)	**0.00**	**0.00**	**25.0978**	46.4181 (20.5039)	**0.00**	0.88	29.6150	57.9549 (35.2765)	**0.00**	0.09
Griewank	**20D**	6.0347	9.5260 (**2.5569**)	**0.00**	0.90	6.6022	11.6896 (7.7297)	**0.00**	0.22	7.3184	12.2352 (8.7066)	**0.00**	0.16	6.1172	10.9125 (6.7669)	**0.00**	0.41	6.8504	13.6950 (14.6094)	**0.00**	0.15
	40D	36.8656	**52.4067** (18.4740)	**0.00**	0.01	**32.3914**	59.6499 (26.2060)	**0.00**	0.16	42.0510	74.0755 (31.9177)	**0.00**	0.64	37.4639	67.5224 (34.1407)	**0.00**	0.75	36.4429	72.1251 (53.0083)	**0.00**	0.86
	60D	98.5873	166.0234 (76.6849)	**0.00**	0.44	106.8965	186.0886 (85.9802)	**0.00**	0.08	142.4185	200.8603 (69.4473)	**0.00**	**0.00**	100.3306	169.7919 (80.0074)	**0.00**	0.34	101.0157	201.3229 (104.6275)	**0.00**	0.03
Rastrigin	**20D**	71.3454	90.9226 (9.6148)	**0.00**	0.63	65.3628	87.3840 (12.2795)	**0.00**	0.39	85.8851	107.0209 (10.4252)	**0.00**	**0.00**	**38.4059**	**83.1847** (13.5187)	**0.00**	0.03	72.5187	90.3095 (9.1947)	**0.00**	0.81
	40D	248.0362	283.0796 (20.0364)	**0.00**	0.42	253.9719	289.3054 (19.6366)	**0.00**	0.68	271.3238	326.9089 (**18.4973**)	**0.00**	**0.00**	**232.0383**	**282.4597** (23.7787)	**0.00**	0.40	239.8390	289.9280 (20.6859)	**0.00**	0.60
	60D	460.6616	527.8386 (26.1039)	**0.00**	0.74	479.2569	518.8316 (24.6766)	**0.00**	0.34	499.4723	570.2138 (30.0119)	**0.00**	**0.00**	**421.6484**	**516.0484** (34.2679)	**0.00**	0.25	482.2438	520.0339 (30.4802)	**0.00**	0.47
Rosenbrock	**20D**	0.0001E+8	0.0008E+8 (**0.0004E+8**)	0.13	0.80	0.0003E+8	0.0011E+8 (0.0011E+8)	0.13	0.29	0.0003E+8	0.0018E+8 (0.0033E+8)	0.13	0.11	0.0004E+8	0.0011E+8 (0.0019E+8)	0.13	0.45	0.0003E+8	0.0011E+8 (0.0013E+8)	0.13	0.30
	40D	0.0068E+8	0.0273E+8 (0.0343E+8)	**0.00**	0.11	0.0119E+8	0.0228E+8 (0.0184E+8)	**0.00**	0.12	0.0183E+8	0.0485E+8 (0.0441E+8)	**0.00**	**0.00**	0.0075E+8	0.0254E+8 (0.0255E+8)	**0.00**	0.09	0.0075E+8	0.0289E+8 (0.0392E+8)	**0.00**	0.10
	60D	0.0407E+8	**0.0995E+8** (**0.0547E+8**)	**0.00**	0.20	0.0500E+8	0.1354E+8 (0.1447E+8)	**0.00**	0.84	0.1070E+8	0.1799E+8 (0.0678E+8)	**0.00**	0.03	**0.0351E+8**	0.1148E+8 (0.1342E+8)	**0.00**	0.66	0.0435E+8	0.1481E+8 (0.1347E+8)	**0.00**	0.55

Table 7 Results of Algorithm Comparison (III)

Function		**RRO9**				**RRO10**				**RRO11**				**RRO12**				**RROv**			
		Best	**Mean (Std.)**	H_9	H_{9_0}	**Best**	**Mean (Std.)**	H_{10}	H_{10_0}	**Best**	**Mean (Std.)**	H_{11}	H_{11_0}	**Best**	**Mean (Std.)**	H_{12}	H_{12_0}	**Best**	**Mean (Std.)**	H_v	H_{v_0}
DeJong	**20D**	1.5511	5.3019 (6.0197)	**0.00**	0.02	1.7374	9.8000 (8.2305)	**0.00**	**0.00**	2.2931	11.9306 (7.9883)	0.02	**0.00**	15.5601	25.1969 (5.6315)	0.30	**0.00**	15.8007	22.8450 (4.1234)	0.66	**0.00**
	40D	12.4404	33.0257 (23.1231)	**0.00**	**0.00**	9.8889	43.5791 (30.2329)	**0.00**	**0.00**	12.9182	49.7813 (23.1734)	**0.00**	**0.00**	51.6317	93.8596 (16.6496)	**0.00**	**0.00**	62.7160	87.2870 (10.3682)	**0.00**	**0.00**
	60D	33.5133	70.2673 (31.7188)	**0.00**	**0.00**	33.2854	110.7640 (45.8308)	**0.00**	**0.00**	35.7375	113.4807 (39.2146)	**0.00**	**0.00**	98.1337	168.8542 (25.8505)	**0.00**	**0.00**	147.0169	166.5786 (12.5331)	**0.00**	**0.00**
Griewank	**20D**	7.4927	16.4869 (14.0096)	**0.00**	0.01	6.5483	24.4491 (22.0757)	**0.00**	**0.00**	8.0266	40.7251 (22.8007)	**0.00**	**0.00**	36.3134	81.1147 (19.9827)	0.09	**0.00**	56.4041	76.7073 (9.7682)	0.03	**0.00**
	40D	33.8026	111.8769 (82.9200)	**0.00**	0.01	42.9797	130.1342 (85.6608)	**0.00**	**0.00**	52.5280	193.9789 (70.9648)	**0.00**	**0.00**	206.7434	311.2986 (51.1770)	**0.00**	**0.00**	227.4857	297.6577 (39.3877)	**0.00**	**0.00**
	60D	92.1387	214.1813 (108.4552)	**0.00**	**0.00**	111.1807	303.8207 (160.3469)	**0.00**	**0.00**	96.4510	425.7359 (129.0514)	**0.00**	**0.00**	355.9943	601.4963 (78.6815)	**0.00**	**0.00**	468.2077	569.7915 (53.8802)	**0.00**	**0.00**
Rastrigin	**20D**	60.9972	95.5107 (14.0398)	**0.00**	0.06	78.2426	104.8989 (13.8468)	**0.00**	**0.00**	67.2630	96.7490 (11.4940)	**0.00**	0.01	76.8831	108.1296 (13.4959)	**0.00**	**0.00**	69.1588	93.1236 (11.4649)	**0.00**	0.21
	40D	251.4815	300.4172 (23.5029)	**0.00**	0.02	257.0324	313.3773 (22.0377)	**0.00**	**0.00**	242.8399	303.2224 (24.1622)	**0.00**	**0.00**	300.5967	327.5516 (18.7778)	**0.00**	**0.00**	259.8361	300.7493 (18.5818)	**0.00**	**0.00**
	60D	475.7276	537.4513 (29.9695)	**0.00**	0.12	492.9566	561.0391 (32.8767)	**0.00**	**0.00**	452.1456	534.1298 (33.8848)	**0.00**	0.29	478.7062	547.3517 (28.3334)	**0.00**	**0.00**	433.2352	531.5328 (42.4731)	**0.00**	0.52
Rosenbrock	**20D**	0.0003E+8	0.0054E+8 (0.0078E+8)	0.16	**0.00**	0.0004E+8	0.0099E+8 (0.0138E+8)	0.20	**0.00**	0.0004E+8	0.0231E+8 (0.0301E+8)	0.36	**0.00**	0.0030E+8	0.0779E+8 (0.0415E+8)	0.58	**0.00**	0.0068E+8	0.0628E+8 (0.0260E+8)	0.88	**0.00**
	40D	**0.0062E+8**	0.0651E+8 (0.0897E+8)	**0.00**	**0.00**	0.0125E+8	0.1751E+8 (0.2062E+8)	**0.00**	**0.00**	0.0169E+8	0.2743E+8 (0.1891E+8)	**0.00**	**0.00**	0.1384E+8	0.5553E+8 (0.1981E+8)	**0.00**	**0.00**	0.3425E+8	0.5320E+8 (0.0991E+8)	**0.00**	**0.00**
	60D	0.0450E+8	0.2332E+8 (0.2434E+8)	**0.00**	0.03	0.0525E+8	0.5410E+8 (0.4634E+8)	**0.00**	**0.00**	0.0612E+8	0.6355E+8 (0.4136E+8)	**0.00**	**0.00**	0.2949E+8	1.3542E+8 (0.4067E+8)	**0.00**	**0.00**	1.0703E+8	1.4290E+8 (0.1983E+8)	**0.00**	**0.00**

4.4.3 Impact of Varying Number of Flight Steps

The parameter N_{steps} governs the number of flight steps taken by a raven. At the end of each step, a 'perception' is made of the landscape by the bird. In RRO0 we set the value of this parameter at ten. In order to examine the sensitivity to this setting we compare the results of RRO0 with RRO6 (five steps) and RRO7 (twenty steps). Again we control for the number of total fitness function evaluations of all algorithm variants.

Figures 7&8 show that algorithm performance is slightly sensitive to changes in the parameter N_{steps}, as increasing the value of this parameter tends to improve performance. However, the differences between RRO0 and RRO7 are not found to be statistically significant. Both RRO6 and RRO7 are found to outperform PSO with the difference in mean performances being statistically significant in eleven of the twelve problem instances.

4.4.4 Impact of Varying Proportion of Followers

The parameter $Perc_{follower}$ determines the proportion of the population that follow the LEADER from the roost to its food find, and serves as a tunable 'recruitment' propensity parameter. The parameter setting also governs how intensively the roost population 'exploits' the food find of the LEADER, or in other words, the level of reliance of the roost on social as distinct from private information. In RRO0 the value of this parameter is set to 0.2, compared with values of 0.4, 0.6 and 0.8 in RRO8, RRO9 and RRO10 respectively. Figures 9&10 show that the performance ranking across the algorithm variants is:

$$RRO0 > RRO8 > RRO9 > RRO10 > PSO$$

The results show that algorithmic performance is sensitive to the setting of the parameter $Perc_{follow}$, with increasing reliance on social information leading to a degradation in the performance of the algorithm. Most likely this occurs as high values of this parameter will encourage heavy exploitation of LEADER information, thereby reducing the diversity of the search process. All three variants (RRO8, RRO9 and RRO10) outperform PSO with the differences being statistically significant in eleven of the twelve problem instances.

4.4.5 Impact of Varying Probability of Stopping

The parameter $Prob_{stop}$ governs the probability that a raven will stop at a location that it 'sees' during flight if it has better food resources than the bird's personal best location. In essence, this parameter governs the propensity of a bird to change feeding location. It also proxies a 'noisy' assessment of resource quality by a bird, as it allows for the case that a good food source is found by a bird but is incorrectly assessed as to its quality. Obviously the value of this parameter can vary between 0 and 1, the former case corresponding to the situation where in-flight perception is turned off, the latter to a 'greedy' search under perfect assessment of resource quality. In RRO0, the probability of stopping is set at 0.1. Two variants on this are examined, RRO11 and RRO12 where the value is 0.2 and 0.4 respectively.

Figures 11&12 show that the performance ranking across the algorithm variants is:

$$RRO0 > RRO11 > RRO12 > PSO$$

These results indicate that performance is enhanced when the probability of stopping is low, and that the performance of the algorithm is sensitive to the parameter value for $Prob_{s\_top}$. Whilst this may appear a counter-intuitive result, in that good feeding sites are bypassed, a lower stopping probability will encourage longer flights from the roost, and therefore greater traversal of the search space. Comparing the performance of RRO11 and RRO12 with PSO, the two variants of RRO generally outperform PSO (except for two instances of RRO12) with the differences being statistically significant in most problem instances.

Figures 3-12 also compare the performances of the PSO and the 13 RRO algorithms. As discussed above, virtually all of the RRO algorithms outperform the PSO algorithm across the problem instances with the degree of out-performance tending to increase as the dimensionality of the problem instance increases, indicating good scalability of the RRO.

5 Conclusions

Social-roosting behaviours are common in nature and provide an opportunity for information-sharing between conspecifics. In this study we draw inspiration from the social roosting and foraging behavior of ravens in order to develop a novel optimisation algorithm. The performance of the algorithm is tested on a number of standard benchmark optimisation problems and is found to be very competitive against the well-known PSO heuristic. A series of analyses are undertaken on the canonical raven roost algorithm and these indicate the importance of the perception mechanism in the algorithm, and highlight the degree of sensitivity of the algorithm's performance to various parameter settings.

The study opens up a door for follow-on work in a number of areas. Perception can be operationalised in the algorithm in a variety of ways and it would be interesting to examine the impact on the algorithm's performance of alternative implementations of this mechanism. As would be expected with any new algorithm there is also scope to further test its utility on a wider range of problems and to undertake further parameter sensitivity analysis. More generally, the domain of social roosting can be extended to include the roosting and foraging behaviour of other bird species and other animals such as bats. In the latter case, while the echolocation capability of bats is well-known, we still have a poor understanding of its role in peer-to-peer communication. It remains an open research question as to what other forms of optimisation algorithms can be inspired by the activities of a wider range of socially roosting animals.

Another interesting area of future work would be to compare the communication mechanisms, and perceptive capabilities, of socially-roosting animals with those of social insects such as ants and honey bees. Whilst a notable literature has emerged concerning the application of ant and honey bee foraging metaphors for the construction of optimisation algorithms, these have placed most focus on social communication and recruitment. In contrast, the results from the RRO algorithm indicates that the inclusion of specific perception and memory mechanisms can assist in the design of quality optimisation algorithms.

References

Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, Oxford

Chong C, Low M, Sivakumar A, Gay K (2006) A Bee Colony Optimization Algorithm to Job Shop Scheduling. In: Proceedings of the 2006 Winter Simulation Conference (WinterSim 2006). IEEE Press, New Jersey, pp 1954-1961

Dall S (2002) Can information sharing explain recruitment to food from communal roosts? Behavioral Ecology, 13(1):42-51

Deygout C, Gault A, Duriez O, Sarrazin F, Bessa-Gomes, C (2010) Impact of food predictability on social facilitation by foraging scavengers. Behavioral Ecology, 21(6):1131-1139

Dorigo M. (1992). Optimization, Learning and Natural Algorithms. PhD Thesis, Politecnico di Milano

Dorigo M, DiCaro G (1999) Ant colony optimization: a new meta-heuristic. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC 1999). IEEE Press, pp 1470-1477

Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, And Cybernetics - Part B: Cybernetics, 26(1):29-41

Dorigo M, Stützle T (2004) Ant Colony Optimization. MIT Press, Cambridge, Massachusetts

Fleming T (1982) Foraging strategies of plant-visiting bats. In: Kunz T (ed) Ecology of Bats. Plenum Press, New York, pp 287-325

Giraldeau LA, Caraco T (2000) Social Foraging Theory. Princeton University Press, New Jersey

Marzluff J, Heinrich B, Marzluff C (1996) Raven roosts are mobile information centres. Animal Behaviour, 51:89-103

Marzluff J, Heinrich B (2001) Raven roosts are still information centres. Animal Behaviour, 61:F14-F15

Nakrani S, Tovey C (2004) On Honey Bees and Dynamic Server Allocation in Internet Hosting Centres. Adaptive Behavior, 12(3-4):223-240

Passino K (2000) Distributed Optimization and Control Using Only a Germ of Intelligence. In: Proceedings of the 2000 IEEE International Symposium on Intelligent Control. IEEE Press, pp 5-13

Passino K (2002) Biomimicry of Bacterial Foraging for Distributed Optimization and Control. IEEE Control Systems Magazine, 22(3):52-67

Pham D, Ghanbarzadeh A, Koc E, Otri S, Rahim S, Zaidi M (2006) The Bees Algorithm - A novel tool for complex optimisation problems. In: Proceedings of International Production Machines and Systems (IPROMS 2006). Elsevier, UK, pp 454-459

Stephens D, Krebs J (1986) Foraging Theory. Princeton University Press, New Jersey

Sumpter D, Brannstrom A (2008) Synergy in social communication. In: D'Ettorre P, Hughes DP (eds) Social Communication. Oxford University Press

Viswanathan G, da Luz M, Raposo E, Stanley E (2011) The Physics of Foraging: An Introduction to Random Searches and Biological Encounters. Cambridge University Press

Ward P, Zahavi A (1973) The importance of certain assemblages of birds as 'information centres' for food finding. Ibis, 115:517-534

Wilkinson G (1992) Information transfer at evening bat colonies. Animal Behaviour, 44(3):501-518

Wray M, Klein B, Seeley T (2011) Honey bees use social information in waggle dances more fully when foraging errors are more costly. Behavioral Ecology. `http://beheco.oxfordjournals.org/content/early/2011/09/30/beheco.arr165.full.pdf+html`. Accessed 1 October 2011

Wright J, Stone R, Brown N (2003) Communal roosts as structured information centres in the raven, Corvus corax. Journal of Animal Ecology, 72:1003-1014

Yang XS (2005) Engineering Optimization via Nature-Inspired Virtual Bee Algorithms. In: Mira J, Álvarez J (eds) Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach. Springer, Berlin, pp 317-323

Zahavi A (1971) The function of pre-roost gatherings and communal roosts. Ibis, 113:106-109