

The raven roosting optimisation algorithm

Anthony Brabazon · Wei Cui · Michael O’Neill

Published online: 10 January 2015
© Springer-Verlag Berlin Heidelberg 2015

Abstract A significant stream of literature which draws inspiration from the foraging activities of various organisms to design optimisation algorithms has emerged over the past decade. The success of these algorithms across a wide variety of application domains has spurred interest in the examination of the foraging behaviours of other organisms to develop novel and powerful, 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 claimed that these roosts can serve as *information centres* to spread knowledge concerning the location of food resources in the environment. In this paper we look at the social roosting and foraging behaviour of one species of bird, the common raven, and take inspiration from this to design a novel optimisation algorithm which we call the *raven roosting optimisation algorithm*. The utility of the algorithm is assessed on a series of benchmark problems and the results are found to be competitive. We also provide a novel taxonomy which classifies foraging-inspired optimisation algorithms based on the underlying social communication mechanism embedded in the algorithms.

Keywords Social foraging · Social roosting · Raven roosting · Information centre · Optimisation

Communicated by V. Loia.

A. Brabazon (✉) · W. Cui · M. O’Neill
Complex Adaptive Systems Laboratory and School of Business,
University College Dublin, Dublin, Ireland
e-mail: anthony.brabazon@ucd.ie

W. Cui
e-mail: wei.cui.c@gmail.com

M. O’Neill
e-mail: m.oneill@ucd.ie

1 Introduction

When the location of food resources are not known with certainty, organisms need to develop a search strategy to find these resources as efficiently and effectively as possible. 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 to design powerful search algorithms. Examples of these algorithms include ant colony optimisation (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; Le Dinh et al. 2013; Nakrani and Tovey 2004; Pham et al. 2006; Yang 2005) to name but a few.

1.1 The context of foraging

Real-world foraging behaviours are context-sensitive and depend on the nature of exploited resources as characterised by their location, size and quality, 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 (Anderson 1991; Benoit-Bird and Au 2009; Deygout et al. 2010; Emlen 1966; Serfass 1995). As would be expected, there has been heavy interaction of these factors in evolutionary time and it is 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 location, and quality, changes over time as a result of factors such as consumption and degradation due to environmental influences. This suggests that higher-quality food foraging strategies need to be adaptive to changing conditions and to feedback based on their degree of past success. This aspect of foraging makes it particularly interesting as a source of inspiration for the design of optimisation algorithms. Many real-world problems are ‘hard’ precisely because they occur in a dynamic environment.

1.2 Social foraging

Foraging for food can be an individual activity where each individual in a species forages on its own (solitary foraging), or a social activity where foraging is a group behaviour. Social foraging was long thought to be confined to higher-level animals such as primates; however, it is now known that co-operative foraging behaviours exist in many species of mammals, fish, birds, and insects (Lonnstedt et al. 2014).

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, and
3. how are food finds divided up between members of the group.

The first two of these topics have particular relevance for the design of foraging inspired optimisation algorithms.

1.3 Communicating information about food

A key aspect of social foraging is the transmission of information between conspecifics about food finds using ‘food calls’ or other behavioural cues and signals. Although a huge array of specific natural mechanisms for social communication of information about food finds exist, these have been broadly categorised into four main groupings in the social foraging literature. The forager who finds a food resource may (Bradbury and Vehrencamp 2011) (p. 591),

1. take up position at the food and broadcast an easily localisable signal to fellow foragers,
2. generate a chemical or visible trail between the food and a central location and then induce fellow foragers to follow this trail to the food,
3. return to a central location and provide directions to fellow foragers on how to find the food, or

4. return to a central location, recruit fellow foragers, and lead them back to the food site.

The economic balance of benefits and costs of the four options vary. Broadcasting an advertisement for a food resource from its location is relatively easy, and the related signal can be optimised for maximum range and localisation. The major cost (risk) is that of eavesdropping as both intraspecific and interspecific competitors could use the signals to locate the advertised food resource. Predators could also use the signal to find the animal advertising the food resource.

The use of chemically mediated trails are common in species of ants, termites, and stingless bees. These trails can create large aggregations of foragers at a food find within a short period. One risk of such trails is their interception by eavesdroppers and predators. Some insects such as stingless bees reduce this risk by creating a broken (as distinct from a continuous) trail wherein pheromone is only laid down every 5–10 m.

Perhaps the best-known example of recruitment at a central location (such as a hive) and providing directions to fellow foragers is provided by the honeybee dance (Seeley 1995; von Frisch 1967). The private nature of this dance (it is performed within the hive) minimises the risk of eavesdropping by non-hive members. The system is costly in that it imposes a significant cognitive burden on both senders and receivers (to produce and to process the information in the dance) and the waggle dances are also an energetically expensive display.

The final mechanism of ‘communication, recruitment at a central location and subsequently leading followers back to the food site’, creates fewer eavesdropper risks than broadcast signals from the food site. However, a potential leader requires a mechanism for locating likely recruits, the ability to find the food again, and sufficient compensation for the extra time and energy that leading imposes on it.

1.3.1 Why communicate information about food?

Although the transmission of information which advertises the existence of and in some cases, the location and quality of food discoveries, is common, an interesting question is why this is so. The benefit to the recipient of the information is evident but it is less clear why an animal would wish to transmit this information. The sender of a food call bears costs, including the risk that a predator will use the signal to locate and attack the sender, and of course, the sender incurs a cost in that the food resource will be shared.

A number of potential benefits to the sender of the signal have been suggested (Bradbury and Vehrencamp 2011). Having more conspecifics at a feeding location means that vigilance and predator defense duties can be shared, and it

also dilutes the risk to an individual if a predator attacks. There may also be other payoffs in that food sharing behaviour may attract potential mates to the food site. In cases where prey items are large in size or dangerous, animals may send food calls to recruit a group of conspecifics to attack the prey item.

The topological distribution of food resources also plays an important role in determining whether a species will tend to group together and forage socially. Species that feed on large, ephemeral, clumps of food such as seeds or fruit often live in groups, as in this case, the key limitation in foraging tends to be finding the location of good food sites (Davies et al. 2012).

Hence, the decision as to whether or not it makes sense to forage socially and/or to alert conspecifics to food resources when found, can be viewed through an economic lens of costs and benefits and many species vary their propensity to give food calls depending on the amount and visibility of food found (Bradbury and Vehrencamp 2011).

1.4 A taxonomy of foraging algorithms

The classification of four methods of communicating information about food finds from the behavioural ecology literature (see Sect. 1.3) provides a useful and novel taxonomy which we can use to distinguish between the best-known families of foraging inspired optimisation algorithms.

In bacteria foraging optimisation an important mechanism in the algorithm is the attraction of other conspecifics to good resource locations via emission of a simulated ‘attractant’ chemical, akin to broadcasting a food signal (communication mechanism 1). In ant-colony optimisation, the communication is via a chemical trail (mechanism 2), whereas in honey bee optimisation algorithms, the emphasis is placed on provision of detailed information as to location via a simulation of the dance process (mechanism 3).

The fourth communication mechanism, namely recruitment at a central site with subsequent ‘leader’ behaviour to bring the conspecifics to the food resource, has not attracted much attention in the design of foraging-inspired optimisation algorithms to date.

Several examples of this foraging behaviour exist in nature including *tandem running* by ants. For example, when an ant of the species *Temnothorax albipennis* finds food, it leads another (naïve) ant from the nest to the food source to ‘teach’ it the route to the food resource (Franklin and Franks 2012).

1.5 Focus of this study

In this study we take an example of the fourth communication mechanism above, drawing on an example of a *social roosting* behaviour. Several species of animals engage in social roosting including some species of birds and bats. These

roosts can potentially serve as *information centres* for the exchange of information concerning the location of food and other resources. We examine one instance of social roosting and associated foraging behaviour, namely that of juvenile common ravens, where information is exchanged at a nocturnal roost, and foragers with knowledge of the location of good food resources recruit for those resources and subsequently lead conspecifics back to them. We design a series of optimisation algorithms drawing inspiration from these behaviours and test them on a series of benchmark functions.

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

2 Roosts as information centres

A social behaviour which is exhibited by many animals is ‘roosting’ where multiple conspecifics come together to rest. This naturally leads to the question as to what evolutionary advantages this behaviour produces. Initial explanations centred on 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) and Ward and Zahavi (1973). This hypothesis suggested that birds join colonies and roosts to increase their foraging efficiency by means of the exchange of information regarding the location of food. The author(s) made a strong claim that enhanced food foraging success was the primary reason for the evolution of gregariousness in birds.

The core tenet 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 they have found).

Interest in the ICH is not restricted to bird roosting behaviours. The possibility that information transfer could also occur in communally roosting bats was first suggested by Fleming (1982). Later work by Wikinson (1992) examined the foraging behaviour of the bat species *Nycticeius humeralis* and found that poorly performing foragers tend to subsequently 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.

2.1 Raven roosts

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, sometimes scavenging the kills of large carnivores such as wolves ([Stahler et al. 2002](#)). 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. Control experiments, in which naïve birds (birds introduced into the roost with no knowledge of the location of food locations) were released at roosts demonstrated that these 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 knowledgeable, 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’ ([Wright et al. 2003](#)).

Considering the reason as to why ravens may share information on food location, one suggestion is that many carcasses will be located within the territory of a mated pair of ravens (the birds become territorial on mating). If a juvenile raven recruits other birds to the carcass in sufficient numbers, the non-territorial group may be able to take the carcass away from the usually dominant territorial pair ([Bradbury and Vehrencamp 2011](#)).

Of course, the foraging process of ravens has a number of additional features. Just like other organisms ([Grüter et al. 2013](#); [Grüter and Leadbeater 2014](#); [Leadbeater and Florent 2014](#); [Wray et al. 2012](#)), 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. In addition, birds can visually survey a wide terrain while flying ([Stahler et al. 2002](#)) and may decide to deviate if an alternative food source is seen whilst in flight. In common with many animals, ravens are known to suffer from ‘neophobia’ (a fear of new things) and have initial reluctance to forage at new carcasses which they have not seen before, particularly if no other birds are feeding there already ([Davies et al. 2012](#); [Stahler et al. 2002](#)). This suggests that ravens will have some reluctance to abandon existing feeding locations for new ones, even if the new locations offer good resources.

In the next section we develop a high-level model of foraging, inspired by the activities of ravens, which is then used to formulate a variety of specific optimisation algorithms.

3 Algorithm development

As discussed in the introduction to this paper, there are a number of elements which contribute to the foraging behaviour of animals ([Giraldeau and Caraco 2000](#); [Stephens and Krebs 1986](#); [Viswanathan et al. 2011](#)), namely

1. an individual’s perception capability, depending on the acuity of its senses such as vision, touch, smell, etc.,
2. an individual’s memory of previously successful / unsuccessful food foraging locations,

3. the capability of an animal to broadcast or receive information about food locations from conspecifics, and
4. a stochastic element to animal movement when searching for new food resources.

All foraging-inspired algorithms embed these components in differing degrees, typically linking them together in a ‘foraging strategy’ and, therefore, will have some high-level similarities. For example, canonical ant colony optimisation algorithms emphasise socially transmitted information via a pheromone trail combined with a random element in that an individual ant is not restricted to follow the strongest pheromone trail when it leaves the ant nest. In most honey bee optimisation implementations, the emphasis is on stochastic recruitment of foraging bees by social transmission of information via the dance language.

Below we propose an algorithm, named the *raven roost optimisation* (RRO) algorithm, which includes all four of the above components. The algorithm draws metaphorical inspiration from the process of raven foraging in that in each iteration of the algorithm, a percentage of the simulated flock of birds (denoted as $\text{Perc}_{\text{follow}}$) are recruited (mimicking social transmission of information at a roost) and follow the bird which found the best location in the last iteration of the algorithm back to that location. The remaining birds return to the best foraging location that they have found so far (simulating a memory). In both cases, if a bird perceives (sees) a ‘better location’ whilst en route to the relevant location they may stochastically decide to stop there instead of continuing with their flight. The details of the operationalisation of the algorithm are provided in the next subsection.

3.1 Description of algorithm

Initially, a randomly located (in the input space) *roosting site* is chosen and this roost location is then fixed for the remainder of the algorithm. Each of the population of N ‘ravens’ is placed at a random initial location (a potential food location) in the search space. The fitness values of the N locations are assessed, and the bird at the location of the best solution is denoted as “LEADER”.

A roosting process is then simulated by mimicking an information-sharing step. In the next iteration of the algorithm, a proportion of the ravens ($\text{Perc}_{\text{follow}}$) are recruited to leave the roost site and follow the LEADER to the best-so-far location. The follower bird then selects a random point within a hypersphere of radius (R_{leader}) around the location previously found by LEADER at which to forage.

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. On leaving the roost, unrecruited birds travel to the best location that they have personally found to date (p_{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 their intended destination, all 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:

$$x_{i,t} = x_{i,t-1} + d_{i,t}, \quad (1)$$

where $x_{i,t}$ is the current position of the i th raven, $x_{i,t-1}$ is its previous position, and $d_{i,t}$ is a random step size. At each step, a raven senses (perceives) the quality of its stopping point and also of its surrounding environment in the range of radius R_{pcept} . In the latter case it samples N_{pcept} perceptions randomly within this hypersphere. If a better location is perceived amongst these locations than the bird’s personal best, there is a $\text{Prob}_{\text{stop}}$ percent chance that the raven stops its flight at that point and forages at the newly found location; otherwise, it takes another flight step and continues to move to its destination. At the conclusion of the algorithm, the highest quality location found is returned. Pseudocode for the algorithm is provided in Algorithm 1, supplemented by Fig. 1 which illustrates the step process.

3.2 Comparison of algorithm with other optimisation algorithms

At a surface level the RRO algorithm bears some similarities with other optimisation algorithms. It is useful to compare its workings with those of some well-known optimisation algorithms (here we consider ACO and PSO algorithms) to clarify what features it shares with existing algorithms and what features are distinct.

To provide a framework for this comparison we employ the four elements of a foraging behaviour described in Sect. 3 above. Table 1 provides a synopsis of how the four elements can be mapped to ACO, PSO, and RRO, respectively. While PSO is not typically considered as a foraging algorithm, it is included in this table for discussion purposes.

3.2.1 Perception

In ACO and PSO, the insects/particles do not have an explicit perception mechanism for sampling (perceiving) the quality of multiple locations on the landscape as they traverse it. They only assess the worth of a single location (or ‘solution’ in ACO) in each iteration, in essence obtaining feedback information to a location decision rather than feedforward information which they can use before selecting a location.

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 their 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 than the bird's personal best
    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 at their destinations (the
  LEADER's vicinity or their personal best), update their
  personal best locations if necessary;
  to the fitness of the location) and evaluate the fitness for each
  forager;
  Update location of the best solution found so far if necessary;
until terminating condition;
    
```

In contrast, a simple perception mechanism is embedded in RRO whereby followers, or birds returning to a ‘private’ location, can sample other locations en route and stochastically stop at these if they find a better location than their p_{best} .

Table 1 Mapping of four foraging behaviours to the ACO, PSO, and RRO algorithms

Foraging behaviour	ACO	PSO	RRO
Individual perception of environment	×	×	✓
Personal memory	×	✓	✓
Social transmission of information	✓	✓	✓
Stochastic search component	✓	✓	✓

3.2.2 Memory

Memory is implemented in distinct ways in each algorithm. In the canonical ACO algorithm, individual ants do not maintain a personal memory as to good solutions they have found; instead, the memory of good solutions found by all ants is maintained via pheromone trails in the simulated search environment. PSO embeds a concept of personal memory in which each particle ‘remembers’ the location of the best solution it has found to date (p_{best}). RRO embeds a similar concept as each bird remembers the best resource location that it has found to date.

3.2.3 Social transmission of information

A core component of all three algorithms is the social transmission of information. From the discussion in Sect. 1.4 it can be seen that the mechanisms of transmission in each algorithm are quite distinct. In ACO, feedback as to the quality of solutions is transmitted socially via pheromone deposits in the environment. In PSO, all particles are aware of the best location found to date by any member of the swarm (g_{best}) and this information is blended with personal information and a momentum factor to produce the position update for each particle. Hence, it can be considered that g_{best} is ‘publicly

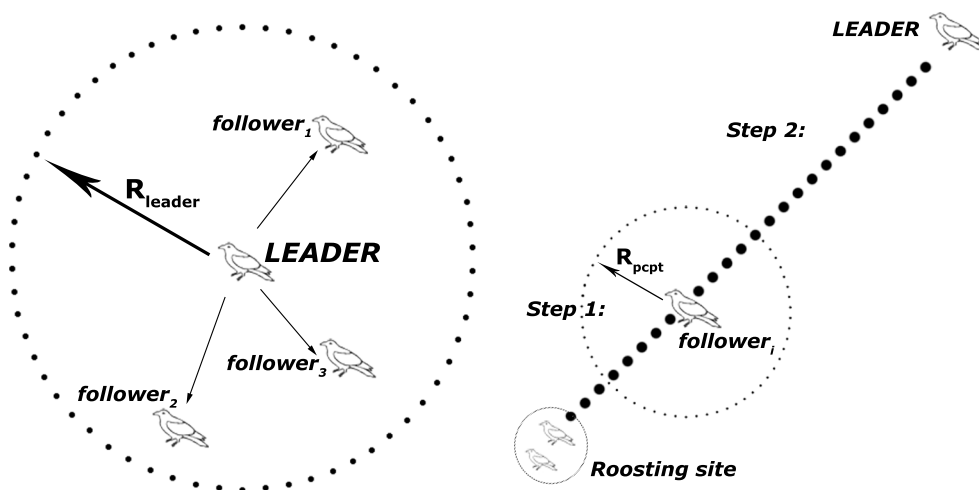


Fig. 1 Illustration of the following and the step process in the RRO algorithm

broadcast’ to all members of the swarm and this information always impacts on the position update of particles. In RRO, a distinct ‘follower’ mechanism is implemented, whereby the bird who has found the ‘best’ location to date recruits a number of followers and leads them back to this site. However, not all birds in the roost are recruited. Therefore, unlike PSO, the best-so-far location does not influence the search process of all birds in each iteration of the algorithm.

3.2.4 Stochastic search

All three algorithms embed stochastic search using different mechanisms. In canonical ACO, the decision as to which arc to follow when leaving a node in the construction graph is typically determined by two pieces of information, the relative amount of pheromone deposited on each arc and information from a problem-specific heuristic which attempts to assess which outgoing arc may be ‘better’. The ant stochastically selects its next arc based on a blending of these pieces of information, tending to favour arcs which are heavily reinforced with pheromone and arcs which are considered good using the heuristic guide to arc quality. During position updates for particles in the PSO algorithm, the relative weight accorded to p_{best} and g_{best} is also varied stochastically. In RRO, a stochastic element is embedded in the algorithm as the length of each step taken is randomly chosen as each bird flies, and the perception samplings at each step are drawn randomly from within a hypersphere at each of these locations.

4 Experiments and results

In this section we describe the experiments undertaken and present the results from these experiments. Four standard benchmark problems (see 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. Griewank and Rastrigin are more complex functions which contain multiple local optima. The aim in all the experiments was to find the vector of values which minimise the value of the test functions. In total, 192 separate experiments are undertaken. The

results of each experiment are averaged over 30 independent runs.

No single study can simultaneously investigate, benchmark, and concisely report results from experiments covering a multiplicity of test functions of varying dimensionality and differing parameter settings. Hence we have had to make choices in deciding where to focus our work. In this initial exploration of the RRO algorithm we have placed most focus on gaining insight into the relative importance of each of the components of the RRO algorithm. Accordingly, we design and test a canonical version of the algorithm (denoted as RRO0) and 13 variants of this (the thirteen variants are denoted as RRO1–RRO12 and RROv, respectively) to undertake an analysis of the relative importance of different parameter settings and of each of the key mechanisms in the algorithm. Details of these variants are set out in Table 3 and are discussed in detail below.

4.1 Overview of experiments

In the first set of experiments we undertake an initial assessment of the performance of the canonical version (or ‘full’ version) of the RRO algorithm (RRO0) on the four test problems. We benchmark these results against those of a canonical version of the PSO algorithm (Ganesan et al. 2012; Kennedy and Eberhart 1995; Kennedy et al. 2001), and against random search (RS). Our rationale for selecting the PSO algorithm is that, like the RRO algorithm, it explicitly incorporates a concept of social and private information and blends this information in the search process. In our implementation we use the canonical version of the PSO algorithm, with the following update steps:

$$v_i(t) = \omega v_i(t - 1) + c_1 \text{rand}(0, 1)(p_{best}(t - 1) - x_i(t - 1)) + c_2 \text{rand}(0, 1)(g_{best}(t - 1) - x_i(t - 1)) \tag{2}$$

$$x_i(t) = x_i(t - 1) + v_i(t), \tag{3}$$

where $x_i(t)$ and $v_i(t)$ are the position and velocity of particle i at time t , ω is the inertia weight (initially set to 1), and c_1 and c_2 are set to 2. Many variants of the PSO algorithm exist in the literature and could have been chosen for implementation in our study. The focus of these experiments was to obtain initial insight into the performance of RRO0 and assess whether it appears reasonably competitive against other heuristics. No

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

Table 3 Parameter setting of algorithms

Algorithm	Radius of perception (R_{pcpt})	Radius of leader (R_{leader})	Perceptions (N_{pcpt})	Steps (N_{steps})	Proportion of followers ($\text{Perc}_{\text{follow}}$)	Probability of stopping ($\text{Prob}_{\text{stop}}$)
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

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)

claim is made that the canonical version of PSO used in this study produces the best possible performance from the entire family of PSO algorithms on the test problems.

These experiments are 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 RRO0 algorithm which includes the perception mechanism. In RROv, the bird still makes its multi-step flight, however, no perceptions are made in the hypersphere surrounding each stopping location.

The next set of experiments examine the sensitivity of RRO0 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 RRO0 algorithm (denoted as RRO1–RRO12). The specific values of the parameters for each algorithm variant are set out in Table 3. Whilst the values for the parameters are chosen judgementslly, we note that values chosen for the two radii (R_{pcpt} and 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 both R_{pcpt} and R_{leader} for the canonical version of the algorithm (RRO0) were chosen after initial experimentation as $\frac{R}{3.6\sqrt[D]{N}}$. The choice of parameter settings for R_{pcpt} , N_{pcpt} , N_{steps} and $\text{Prob}_{\text{stop}}$ govern the importance of ‘perception’ in the algorithm. If all are set to low values, individual

Table 4 Parameter setting for experiments

Parameters	Values
Trials	30
Size of population	$N = 30$
Dimension of problem	$D = 20, 40, 60$

perception by a bird will play little role in their foraging and reliance will instead be placed on search around the best point found by the LEADER or the bird’s p_{best} . Conversely, if higher values are set for these parameters, the emphasis on perception is increased, and the algorithm will place less reliance on social information and memory.

4.1.1 Stopping criteria

The experimental parameters are shown in Table 4. In each experiment, 30 ravens in the case of RRO, or 30 particles in the case of PSO, are used. When assessing the relative effectiveness of multiple algorithms against one another, particularly when the algorithms have distinctly different internal processes, it is often not a trivial matter to design a perfect basis for comparison. A common approach is to allocate an equivalent number of fitness function evaluations to each algorithm on the basis that each algorithm then has access to the same amount of information from the environment. While this approach appears reasonable,

Table 5 Iteration settings for each trial

RRO0	RRO1	RRO2	RRO3	RRO4	RRO5	RRO6	RRO7	RRO8	RRO9	RRO10	RRO11	RRO12	RROv	PSO	RS
500	500	500	500	1,000	250	1,000	250	500	500	500	500	500	5,000	60,000	60,000

it ignores the issue that a more complex algorithm will have a longer run time than a simpler one, for the same number of fitness function evaluations. This is a particular issue when random search is used as a benchmark as the computational overhead of a random search algorithm (excluding the cost of fitness function evaluation) will be low.

In our experiments we seek to provide a balance between these issues. We adjust the number of iterations for which each algorithm is run to produce a similar number of fitness function evaluations across all experiments. Hence the stopping criterion for each algorithm was set at a maximum number of fitness function evaluations in each case. RRO has some additional computational overhead over and above RS or PSO due to its implementation of a ‘step’ flight process. The computational overhead of this relative to the computational cost of fitness function evaluation varies depending on the complexity of the fitness calculation, which in turn is impacted by the complexity and dimensionality of the test problems examined. In order to present a conservative assessment of the performance of RRO and its variants, we, therefore, allow RS and PSO to undertake a slightly greater number of fitness function evaluations.

Table 5 sets out the number of iterations used for each algorithm. As a baseline, we run RRO0–RRO4 for 500 iterations as (with 30 ravens, this produces a maximum of approximately 3,300 fitness evaluations per iteration, leading to a maximum of some 1,650,000 fitness function evaluations over 500 iterations of the entire algorithm). In the case of RRO5, the number of perception samples taken at each step is halved (5 samples vs the 10 samples taken in RRO0–RRO4) and hence we run the algorithm variant for 1,000 iterations (rather than 500 iterations). In the case of RS and PSO, there are 30 fitness function evaluations per iteration of the algorithm, and the maximum number of iterations for each algorithm is set at 60,000 (giving a maximum of 1,800,000 fitness function evaluations). All reported results are averaged over 30 independent experiments for each problem and algorithm pairing and we test the statistical significance of all differences in performance at a 95 % level using a *t* test.

4.1.2 Type of computer system used

The experiments were undertaken on a PC Intel Core i7 (2.93 GHz) system with 12 GB RAM.

4.2 Hypotheses tested

In our tables of results, we provide a statistical analysis of the difference in the mean performance of each algorithmic variant against the canonical version of RRO0 and also of the performance of each version of RRO against PSO and RS. In all cases, the null hypothesis is that there is no difference in performance. The notation used is as follows:

In assessing the performance of the 13 variants of RRO with different parameter settings (RRO1–RRO12 and RROv) against the performance of the canonical algorithm (RRO0):

- H_0 : the RRO $_i$ ($i:1-12$) algorithm performs as well as the RRO0 algorithm.

In considering the performance of the various versions of RRO against PSO:

- H_{PSO} : the RRO algorithm performs as well as the PSO algorithm.

In considering the performance of the various versions of RRO performs against RS:

- H_{RS} : the RRO algorithm performs as well as the RS algorithm.

4.3 Analysis of raven roost algorithm

Initially, we focus attention on comparing the results obtained by RRO0 against those of PSO and RS. Figures 4 and 5 provide boxplots of the best-of-run results for each algorithm over the 30 trials. Figures 2 and 3 provide a graphic showing the evolution of the average best for each algorithm for the 40 and 60D cases (the illustrations for the 20D case are qualitatively similar and are, therefore, omitted to save space).

A practical issue in preparing Figs. 2 and 3 is that the number of fitness function evaluations in each iteration varies from one algorithmic variant to another. In order to produce these figures we standardise the comparisons based on the 250 iterations of RRO5 and RRO7 and plot the best fitness value obtained at the end of each of these 250 iterations (averaged over all 30 runs) for RRO5 and RRO7. In the case of RRO0–RRO3 and RRO8–RRO12 (which run for 500 iterations), the best fitness is plotted every second iteration, and for RRO4 and RRO6 the best fitness from each fourth iter-

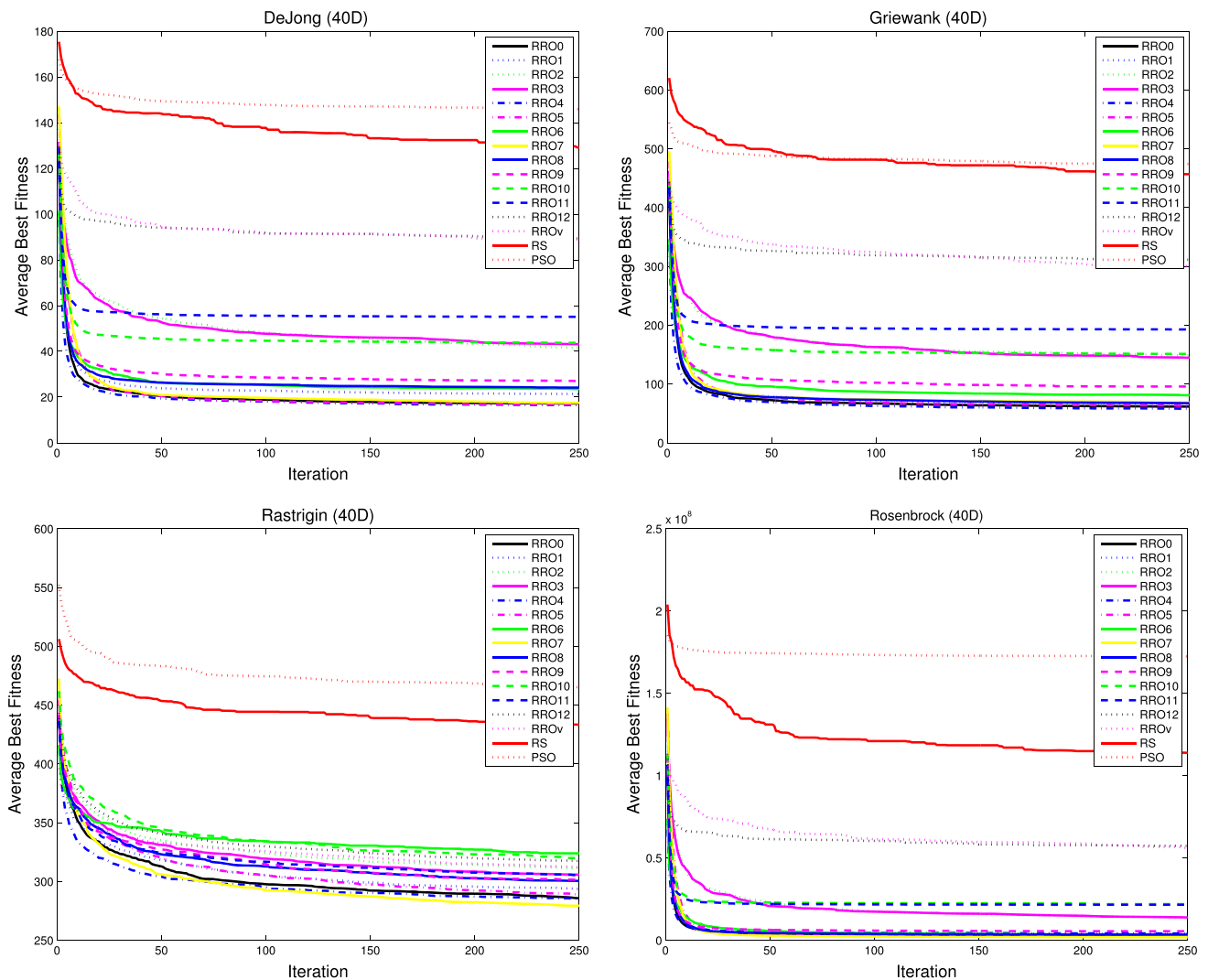


Fig. 2 Average best performance (averaged over 30 trials) of each algorithm variant on test problems (40D) case

ation is plotted. Hence, (for example) iteration 1 in Figs. 2 and 3 plots the average best performance after the first iteration of algorithms RRO5/RRO7, after the second iteration of algorithms RRO0–RRO3 and RRO8–RRO12, and after the fourth iteration for algorithm RRO 4. Similarly, for RROv, PSO, and RS, the plotted fitnesses are sampled and plotted from an interval determined by dividing the total number of iterations for that algorithm by 250.

4.3.1 Effectiveness and efficiency of canonical RRO

From the end-of-run boxplots (Figs. 4, 5) we can observe that the RRO0 algorithm generally outperforms both random search and PSO, and in eleven out of twelve cases the difference is statistically significant for PSO (and in 12 out of 12 cases for RS). The relevant p values are shown in the two right-most columns in Tables 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 and 17.

Examining the evolution of the performance of each algorithm in Figs. 2 and 3, it can be noted that RRO0 quickly outperforms both PSO and RS and hence appears to make more efficient use of information on the search environment gathered from fitness function evaluations.

4.3.2 Convergence and stability of canonical RRO

The boxplots (Figs. 4, 5) indicate that the results for the RRO0 algorithm show reasonable convergence to similar values by the end of the run (across all 30 trials) suggesting robustness to the (random) initial raven positions for each trial and to (random) choice of roosting site location. It is also noted that the results from the RRO0 algorithm (see Figs. 2, 3) have converged reasonably quickly across each problem examined. From the same figures it can also be observed that the results for PSO and RS show a high degree of convergence

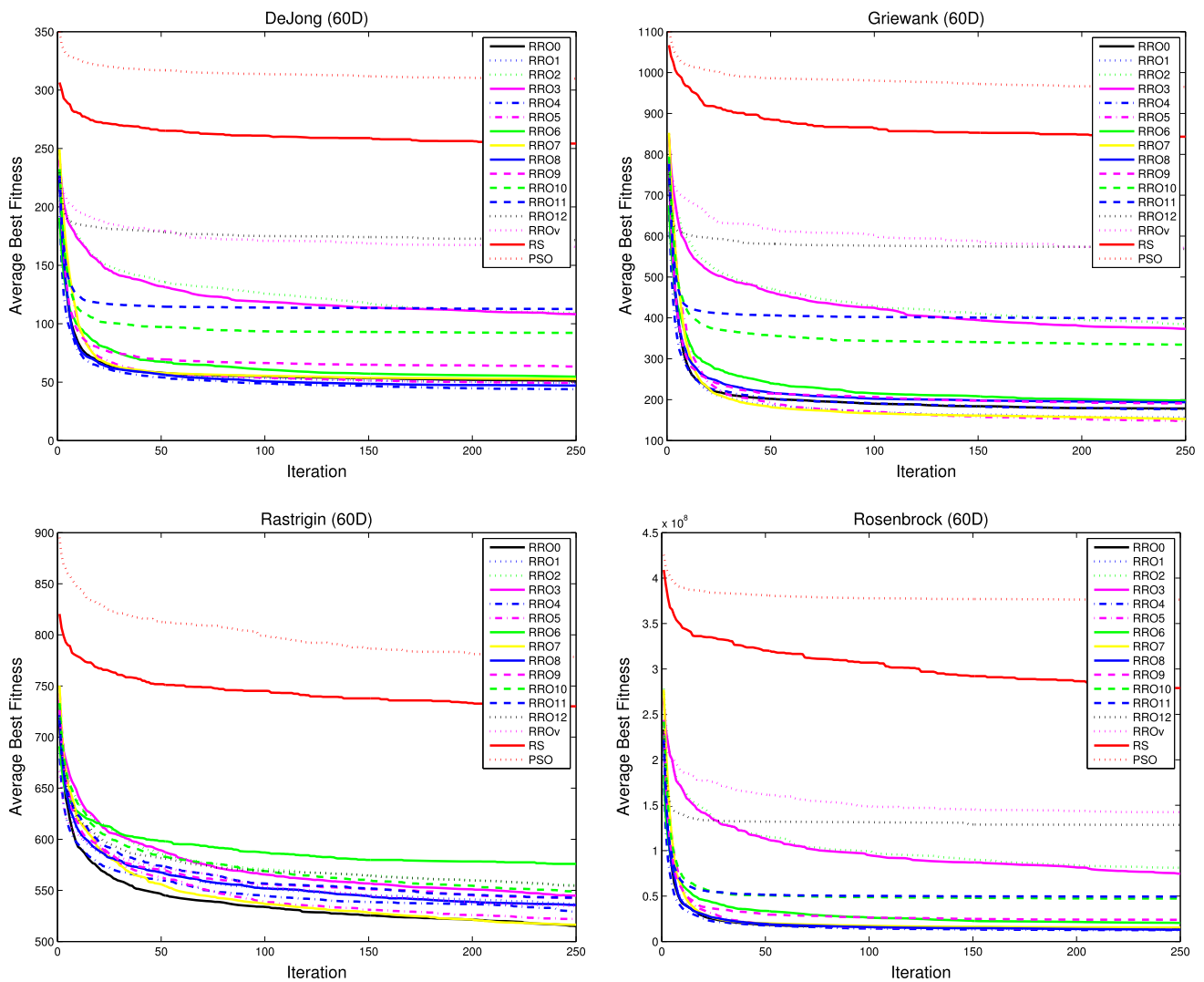


Fig. 3 Average best performance (averaged over 30 trials) of each algorithm variant on test problems (60D) case

from quite early in their runs and hence, the results obtained from these algorithms are not particularly sensitive to the precise choice of number of iterations.

4.4 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 (identical, except for the removal of the perception mechanism), are examined.

Figures 2 and 3 compare the evolution of the average best fitness of the three algorithms for the tested problems, and Tables 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 and 17 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 RRO0 out-

performs the RROv algorithm on all problems and that in eleven out of twelve instances the difference is statistically significant (the relevant *p* values are shown in the third from right-most column in each Table). Hence, we conclude that the perception mechanism is an important component of the RRO algorithm.

4.5 Parameter sensitivity analysis

Figures 2 and 3 also enable us to visually assess the results of sensitivity analysis of the six parameters of the RRO. Column H_0 in Tables 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 and 17 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 $H_{PSO}-H_{RS}$ in Tables 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 and 17 show the *p* values of the statistical tests used to

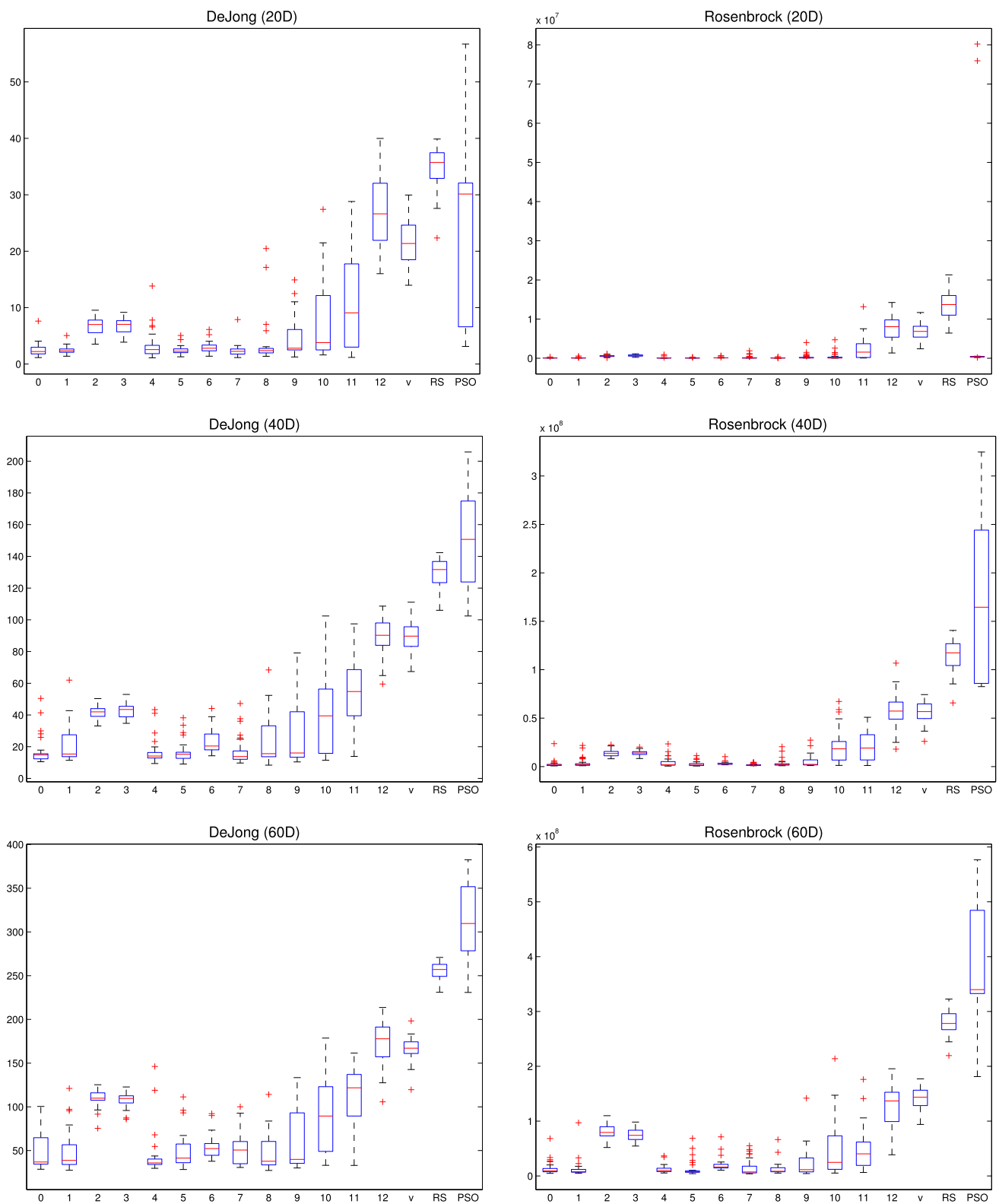


Fig. 4 Boxplot of best results obtained by each algorithm variant (plotted over 30 runs) on DeJong and Rosenbrock Functions. The x axis shows the algorithm variant identifier RRO0–RR012, RROv, RS and PSO

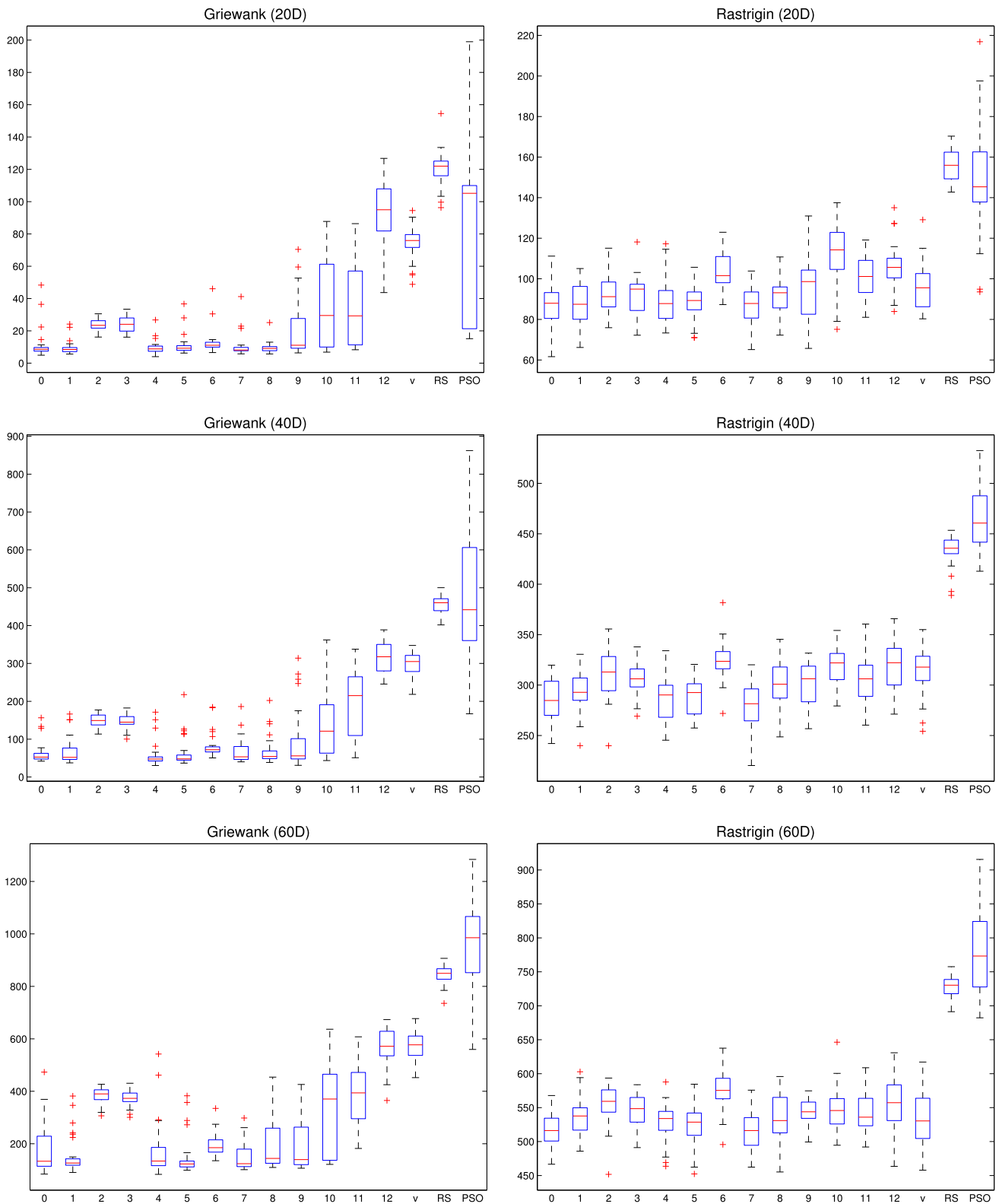


Fig. 5 Boxplot of best results obtained by each algorithm variant (plotted over 30 runs) on Griewank and Rastrigin Functions. The x axis shows the algorithm variant identifier RRO0–RR012, RROv, RS, and PSO

Table 6 Results of algorithm comparison (DeJong 20D)

Algorithms	Best	Mean	Std.	H_0	H_{PSO}	H_{RS}
RRO0	1.1433	2.5694	1.1965		0.0000	0.0000
RRO1	1.3931	2.4239	0.7036	0.5683	0.0000	0.0000
RRO2	3.5274	6.8762	1.5374	0.0000	0.0000	0.0000
RRO3	3.8778	6.7870	1.2901	0.0000	0.0000	0.0000
RRO4	1.0977	3.2996	2.5732	0.1640	0.0000	0.0000
RRO5	1.2759	2.3970	0.7734	0.5101	0.0000	0.0000
RRO6	1.3882	2.9684	0.9849	0.1638	0.0000	0.0000
RRO7	1.1473	2.4024	1.1746	0.5876	0.0000	0.0000
RRO8	1.3734	3.6288	4.3107	0.1997	0.0000	0.0000
RRO9	1.2557	4.6890	3.5094	0.0027	0.0000	0.0000
RRO10	1.6098	7.7694	7.2951	0.0003	0.0000	0.0000
RRO11	1.1705	10.9890	8.6096	0.0000	0.0000	0.0000
RRO12	16.0196	26.3861	6.4034	0.0000	0.6162	0.0000
RROv	13.9676	21.5745	4.0803	0.0000	0.0679	0.0000
RS	22.3475	34.7632	4.0542			
PSO	3.1221	28.2508	19.2281			

Table 7 Results of algorithm comparison (DeJong 40D)

Algorithms	Best	Mean	Std.	H_0	H_{PSO}	H_{RS}
RRO0	10.5405	17.2223	9.2001		0.0000	0.0000
RRO1	11.4513	21.3832	11.8610	0.1344	0.0000	0.0000
RRO2	33.0629	41.6081	4.3000	0.0000	0.0000	0.0000
RRO3	34.7454	43.1591	4.4858	0.0000	0.0000	0.0000
RRO4	9.3906	16.4889	8.0312	0.7434	0.0000	0.0000
RRO5	9.0687	16.7350	6.7431	0.8158	0.0000	0.0000
RRO6	14.2764	23.5263	7.5739	0.0053	0.0000	0.0000
RRO7	9.7388	17.4518	9.0761	0.9229	0.0000	0.0000
RRO8	8.3576	24.1669	14.9873	0.0347	0.0000	0.0000
RRO9	10.3897	27.0794	18.6460	0.0119	0.0000	0.0000
RRO10	11.4622	43.7419	28.1358	0.0000	0.0000	0.0000
RRO11	13.9066	55.1106	23.1648	0.0000	0.0000	0.0000
RRO12	59.5071	89.3896	12.5065	0.0000	0.0000	0.0000
RROv	67.3813	88.8919	10.3705	0.0000	0.0000	0.0000
RS	106.0110	129.2142	10.0996			
PSO	102.4266	145.9071	33.2386			

test whether there is any difference between the performance of the RRO algorithm variants and PSO and RS, respectively.

RRO0, 1 > RRO2, 3

4.5.1 Impact of varying perception radius

Initially we compare the performance of RRO0 with three variants (RRO1–RRO3) which have larger perception radii for R_{pcpt} and/or R_{leader} . In general, across most problem instances, the performance ranking on the four algorithm variants is as follows:

Relatively 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, and RRO3 against that of PSO, all are found to out-

Table 8 Results of algorithm comparison (DeJong 60D)

Algorithms	Best	Mean	Std.	H_0	H_{PSO}	H_{RS}
RRO0	28.6655	51.3196	22.4168		0.0000	0.0000
RRO1	27.5026	49.3129	22.3908	0.7299	0.0000	0.0000
RRO2	75.4347	109.3080	9.8436	0.0000	0.0000	0.0000
RRO3	85.7726	108.1016	8.2982	0.0000	0.0000	0.0000
RRO4	29.7697	43.8796	25.4664	0.2346	0.0000	0.0000
RRO5	28.3357	49.2651	20.5080	0.7124	0.0000	0.0000
RRO6	37.7723	54.6793	13.6009	0.4856	0.0000	0.0000
RRO7	30.7351	52.6931	20.1475	0.8038	0.0000	0.0000
RRO8	27.3253	46.9927	20.8157	0.4416	0.0000	0.0000
RRO9	30.0976	63.3947	34.1911	0.1112	0.0000	0.0000
RRO10	33.1730	92.1664	45.9878	0.0001	0.0000	0.0000
RRO11	33.0007	112.6194	33.6566	0.0000	0.0000	0.0000
RRO12	105.6885	171.7069	24.3181	0.0000	0.0000	0.0000
RROv	119.6281	165.6292	15.2706	0.0000	0.0000	0.0000
RS	231.0031	254.2297	11.3885			
PSO	230.8468	309.7660	42.5911			

Table 9 Results of algorithm comparison (Griewank 20D)

Algorithms	Best	Mean	Std.	H_0	H_{PSO}	H_{RS}
RRO0	4.9201	11.1692	9.1849		0.0000	0.0000
RRO1	5.6865	9.4731	4.1299	0.3601	0.0000	0.0000
RRO2	16.1899	24.0840	3.6261	0.0000	0.0000	0.0000
RRO3	16.1379	23.8251	4.6714	0.0000	0.0000	0.0000
RRO4	4.0471	9.6061	4.1517	0.3992	0.0000	0.0000
RRO5	6.2308	10.9994	6.3156	0.9338	0.0000	0.0000
RRO6	6.5796	12.7668	7.4445	0.4622	0.0000	0.0000
RRO7	5.7474	10.2424	6.9608	0.6612	0.0000	0.0000
RRO8	5.7403	9.4787	3.3913	0.3482	0.0000	0.0000
RRO9	6.3332	20.1232	17.4931	0.0160	0.0000	0.0000
RRO10	6.8043	37.2083	28.4341	0.0000	0.0015	0.0000
RRO11	8.3011	37.9738	25.9648	0.0000	0.0016	0.0000
RRO12	43.6980	93.1524	19.2567	0.0000	0.2374	0.0000
RROv	48.8332	74.4491	11.0190	0.0000	0.7022	0.0000
RS	96.2830	119.9137	11.4135			
PSO	15.1391	78.9007	62.4815			

perform PSO, and the difference in mean performance is statistically significant in virtually all problem instances.

4.5.2 Impact of varying the 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. Tables 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 and 17 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

Table 10 Results of algorithm comparison (Griewank 40D)

Algorithms	Best	Mean	Std.	H_0	H_{PSO}	H_{RS}
RRO0	42.0067	61.6318	28.2116		0.0000	0.0000
RRO1	37.4196	66.6595	34.8635	0.5416	0.0000	0.0000
RRO2	113.5695	148.7497	16.7754	0.0000	0.0000	0.0000
RRO3	100.1396	144.8812	17.9930	0.0000	0.0000	0.0000
RRO4	30.4829	58.0296	33.1604	0.6521	0.0000	0.0000
RRO5	36.7204	63.0155	38.9499	0.8753	0.0000	0.0000
RRO6	50.3735	81.1179	32.7351	0.0165	0.0000	0.0000
RRO7	40.0386	68.7545	34.0277	0.3811	0.0000	0.0000
RRO8	38.4638	67.4838	37.1559	0.4948	0.0000	0.0000
RRO9	30.8743	95.7281	78.8665	0.0297	0.0000	0.0000
RRO10	43.5050	151.2932	102.3744	0.0000	0.0000	0.0000
RRO11	51.0081	192.8115	82.9302	0.0000	0.0000	0.0000
RRO12	245.5225	311.5603	42.4035	0.0000	0.0000	0.0000
RROv	218.4021	300.7273	31.8063	0.0000	0.0000	0.0000
RS	401.7900	456.9386	23.8332			
PSO	167.0205	474.4007	151.7499			

Table 11 Results of algorithm comparison (Griewank 60D)

Algorithms	Best	Mean	Std.	H_0	H_{PSO}	H_{RS}
RRO0	84.3365	178.4021	93.9705		0.0000	0.0000
RRO1	90.5712	156.2390	71.6040	0.3084	0.0000	0.0000
RRO2	306.5267	384.9607	29.7312	0.0000	0.0000	0.0000
RRO3	300.6642	373.2262	28.7739	0.0000	0.0000	0.0000
RRO4	83.4968	176.1740	104.9454	0.9313	0.0000	0.0000
RRO5	99.1501	147.9523	74.0430	0.1686	0.0000	0.0000
RRO6	135.4182	198.1581	44.3852	0.3021	0.0000	0.0000
RRO7	101.1797	152.7378	55.9668	0.2038	0.0000	0.0000
RRO8	109.6892	193.6979	97.1950	0.5379	0.0000	0.0000
RRO9	106.7870	190.9501	94.7350	0.6085	0.0000	0.0000
RRO10	121.4726	334.6285	173.1223	0.0001	0.0000	0.0000
RRO11	182.1786	399.0440	111.2215	0.0000	0.0000	0.0000
RRO12	365.1553	568.6470	76.4417	0.0000	0.0000	0.0000
RROv	452.0903	571.3513	49.1254	0.0000	0.0000	0.0000
RS	735.2996	843.2480	37.9307			
PSO	559.5375	965.1286	170.0045			

the parameter N_{pcpt} . Both RRO4 and RRO5 outperform PSO and RS on all problem instances, significantly so on eleven of the twelve instances in the case of PSO (in all twelve in the case of RS).

4.5.3 Impact of varying the 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 (10 steps) with RRO6 (5 steps) and RRO7 (20 steps). Again we control for the number of total fitness function evaluations of all algorithm variants.

Tables 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 and 17 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

Table 12 Results of algorithm comparison (Rastrigin 20D)

Algorithms	Best	Mean	Std.	H_0	H_{PSO}	H_{RS}
RRO0	61.6345	86.6571	10.9716		0.0000	0.0000
RRO1	66.1654	87.7620	10.0601	0.6858	0.0000	0.0000
RRO2	75.9237	91.9517	8.5941	0.0419	0.0000	0.0000
RRO3	72.2118	92.3292	9.4472	0.0361	0.0000	0.0000
RRO4	73.3173	89.6263	11.2907	0.3059	0.0000	0.0000
RRO5	70.9641	88.1040	9.1517	0.5812	0.0000	0.0000
RRO6	87.2707	104.2442	9.5035	0.0000	0.0000	0.0000
RRO7	65.1208	87.3773	8.8159	0.7803	0.0000	0.0000
RRO8	72.2482	91.4735	9.4268	0.0733	0.0000	0.0000
RRO9	65.6834	95.4169	15.5322	0.0144	0.0000	0.0000
RRO10	75.1167	112.9087	15.2423	0.0000	0.0000	0.0000
RRO11	81.0313	100.7231	10.4160	0.0000	0.0000	0.0000
RRO12	83.9045	106.0616	10.9475	0.0000	0.0000	0.0000
RROv	80.2567	96.1776	10.8675	0.0013	0.0000	0.0000
RS	142.7369	156.5669	8.5785			
PSO	93.5700	149.1411	27.0457			

Table 13 Results of algorithm comparison (Rastrigin 40D)

Algorithms	Best	Mean	Std.	H_0	H_{PSO}	H_{RS}
RRO0	242.0983	285.8088	20.8212		0.0000	0.0000
RRO1	239.9658	293.7275	21.0714	0.1486	0.0000	0.0000
RRO2	239.8915	311.3059	24.2634	0.0001	0.0000	0.0000
RRO3	269.2177	305.7641	14.8344	0.0001	0.0000	0.0000
RRO4	245.3586	285.5230	23.1231	0.9601	0.0000	0.0000
RRO5	257.4362	289.2121	16.3326	0.4840	0.0000	0.0000
RRO6	271.9300	323.9496	19.2015	0.0000	0.0000	0.0000
RRO7	220.3040	279.3972	22.2758	0.2542	0.0000	0.0000
RRO8	248.6233	300.5300	22.0478	0.0101	0.0000	0.0000
RRO9	256.8026	301.6179	21.3625	0.0052	0.0000	0.0000
RRO10	279.2941	319.9033	22.1380	0.0000	0.0000	0.0000
RRO11	260.2963	305.4708	21.9666	0.0008	0.0000	0.0000
RRO12	271.2305	317.5597	22.7900	0.0000	0.0000	0.0000
RROv	254.2242	312.2624	24.3679	0.0000	0.0000	0.0000
RS	388.8564	433.4854	15.2177			
PSO	412.8650	465.2801	31.8425			

significant. Both RRO6 and RRO7 are found to outperform PSO and RS on all problem instances, with the difference in mean performance being significant on 11 of the 12 instances in the case of PSO (in all 12 in the case of RS).

4.5.4 Impact of varying the proportion of followers

The parameter $\text{Perc}_{\text{follower}}$ 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. Tables 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 and 17 show that the performance ranking across the algorithm variants is generally

Table 14 Results of algorithm comparison (Rastrigin 60D)

Algorithms	Best	Mean	Std.	H_0	H_{PSO}	H_{RS}
RRO0	466.6402	514.9922	26.4103		0.0000	0.0000
RRO1	485.8427	536.7983	28.3844	0.0032	0.0000	0.0000
RRO2	451.7679	555.0060	28.8960	0.0000	0.0000	0.0000
RRO3	491.0995	544.8866	25.6993	0.0000	0.0000	0.0000
RRO4	463.6742	528.7501	27.9410	0.0548	0.0000	0.0000
RRO5	452.3285	522.0707	30.3623	0.3393	0.0000	0.0000
RRO6	495.7037	575.9629	28.0688	0.0000	0.0000	0.0000
RRO7	462.4604	516.3762	31.8793	0.8554	0.0000	0.0000
RRO8	455.1942	535.7413	33.9887	0.0106	0.0000	0.0000
RRO9	499.4521	542.3014	21.1436	0.0000	0.0000	0.0000
RRO10	494.6903	548.9187	32.2119	0.0000	0.0000	0.0000
RRO11	492.0794	542.9837	27.7475	0.0002	0.0000	0.0000
RRO12	463.5904	554.3203	36.0934	0.0000	0.0000	0.0000
RROv	457.8350	534.6582	40.4950	0.0298	0.0000	0.0000
RS	691.2642	730.1369	15.1548			
PSO	682.3185	778.4898	59.7009			

Table 15 Results of algorithm comparison (Rosenbrock 20D)

Algorithms	Best	Mean	Std.	H_0	H_{PSO}	H_{RS}
RRO0	34,063	89,710	52,383		0.1334	0.0000
RRO1	28,854	99,855	95,093	0.6107	0.1341	0.0000
RRO2	87,388	575,969	228,115	0.0000	0.1707	0.0000
RRO3	250,556	697,429	227,874	0.0000	0.1812	0.0000
RRO4	13,946	123,723	184,815	0.3362	0.1357	0.0000
RRO5	25,506	88,922	54,746	0.9547	0.1333	0.0000
RRO6	49,990	143,591	138,093	0.0504	0.1371	0.0000
RRO7	42,697	239,210	421,153	0.0586	0.1442	0.0000
RRO8	15,194	90,971	64,958	0.9343	0.1334	0.0000
RRO9	43,930	383,991	765,698	0.0401	0.1554	0.0000
RRO10	39,889	466,966	971,925	0.0380	0.1622	0.0000
RRO11	67,693	2,553,544	3,014,762	0.0000	0.4109	0.0000
RRO12	1,345,975	7,354,348	3,271,392	0.0000	0.6265	0.0000
RROv	2,412,261	6,764,762	2,233,833	0.0000	0.7426	0.0000
RS	6,424,540	13,631,012	3,668,334			
PSO	186,917	5,569,390	19,715,022			

RRO0 > RRO8 > RRO9 > RRO10

The results show that algorithmic performance is sensitive to the setting of the parameter $\text{Perc}_{\text{follow}}$, with increasing reliance on social information leading to a degradation in the performance of the algorithm. Plausibly 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 and RS on all problem instances, with the difference in mean performance being significant on

11 of the 12 instances in the case of PSO (in all twelve in the case of RS).

4.5.5 Impact of varying the probability of stopping

The parameter $\text{Prob}_{\text{stop}}$ 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

Table 16 Results of algorithm comparison (Rosenbrock 40D)

Algorithms	Best	Mean	Std.	H_0	H_{PSO}	H_{RS}
RRO0	716,077	2,662,908	4,136,681		0.0000	0.0000
RRO1	889,920	3,999,693	5,162,500	0.2730	0.0000	0.0000
RRO2	8,088,236	13,717,609	3,369,159	0.0000	0.0000	0.0000
RRO3	8,352,652	13,949,865	2,878,015	0.0000	0.0000	0.0000
RRO4	549,298	4,182,098	5,139,341	0.2123	0.0000	0.0000
RRO5	652,118	2,602,896	2,458,967	0.9458	0.0000	0.0000
RRO6	1,818,842	3,232,560	1,631,106	0.4857	0.0000	0.0000
RRO7	912,597	1,852,210	1,002,168	0.3012	0.0000	0.0000
RRO8	856,104	3,511,087	4,386,071	0.4441	0.0000	0.0000
RRO9	1,024,249	5,489,751	6,404,683	0.0469	0.0000	0.0000
RRO10	1,189,280	21,976,222	18,885,477	0.0000	0.0000	0.0000
RRO11	1,152,772	21,593,279	15,408,112	0.0000	0.0000	0.0000
RRO12	18,022,327	57,361,336	20,024,847	0.0000	0.0000	0.0000
RROv	26,100,671	56,150,451	10,905,740	0.0000	0.0000	0.0000
RS	65,705,121	113,919,865	16,077,614			
PSO	82,573,731	172,425,977	78,992,915			

Table 17 Results of algorithm comparison (Rosenbrock 60D)

Algorithms	Best	Mean	Std.	H_0	H_{PSO}	H_{RS}
RRO0	4,763,486	14,053,562	12,753,026		0.0000	0.0000
RRO1	4,920,542	12,818,565	16,973,557	0.7512	0.0000	0.0000
RRO2	51,911,755	81,113,385	12,867,503	0.0000	0.0000	0.0000
RRO3	54,567,270	74,667,130	11,167,395	0.0000	0.0000	0.0000
RRO4	5,459,199	12,468,188	7,375,277	0.5579	0.0000	0.0000
RRO5	4,204,857	13,518,573	14,855,674	0.8815	0.0000	0.0000
RRO6	10,553,423	20,528,952	12,288,728	0.0499	0.0000	0.0000
RRO7	3,820,881	15,349,822	15,390,135	0.7237	0.0000	0.0000
RRO8	5,251,040	13,079,222	1,242,0126	0.7654	0.0000	0.0000
RRO9	4,341,940	23,887,419	28,304,495	0.0881	0.0000	0.0000
RRO10	5,280,445	47,269,522	51,538,364	0.0011	0.0000	0.0000
RRO11	6,264,836	49,389,896	39,757,828	0.0000	0.0000	0.0000
RRO12	38,512,404	128,354,601	37,235,512	0.0000	0.0000	0.0000
RROv	93,952,140	142,348,286	20,047,512	0.0000	0.0000	0.0000
RS	219,505,672	278,966,021	22,691,611			
PSO	181,252,077	376,232,029	110,060,401			

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 (effectively) 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.

Tables 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 and 17 show that the performance ranking across the algorithm variants is

RRO0 > RRO11 > RRO12

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 $\text{Prob}_{\text{stop}}$. 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 and RS, the two variants

of RRO generally outperform PSO and RS (except for two instances of RRO12) with the differences being statistically significant in most problem instances.

4.5.6 Effectiveness and efficiency of RRO variants

From the end-of-run boxplots in Figs. 4 and 5 we can see that RRO0 typically performed as well or better than most of the variants (RRO1–RRO12) tested, although the differences in mean performance were usually only significant when compared with the performance of RRO2–RRO3 and RRO10–RRO12, as discussed in the subsections above. The differences in performance stem from the way that the fitness function evaluation budget is shared between the components of each algorithmic variant, and in the case of RRO2 and RRO3 the radius of perception. The general conclusion that can be drawn from the results is that the canonical version of the algorithm is not outperformed by any of the variants of it that were tested. Assessing the performance of the variants (RRO1–RRO12) against that of PSO and RS, while there is some variability in the results produced by each of the variants, most perform quite competitively against PSO and RS. This suggests that the performance of RRO is reasonably robust with respect to choice of parameter settings, at least for the range of test problems and settings examined.

4.5.7 Convergence and stability of RRO variants

Figures 4 and 5 also allow us to assess the end-of-run convergence of each of the RRO variants (RRO1–RRO12). In general, most variants apart from RRO9–RRO12 show a good degree of convergence. The greater dispersion in results produced by RRO9 and RRO10 is not unexpected, as these variants allocate high portions of the raven population to act as followers (60 and 80 % respectively). As this proportion increases, the balance of the algorithm shifts towards exploitation around the location found by the LEADER and this increases the risk of the algorithm becoming trapped in a local optimum. In RRO11 and RRO12, the propensity of a raven to stop during their flight if they perceive a better location than their own p_{best} is increased. While on the one hand this increases the chance that a new good location will be harvested by a bird, it has the side-effect of reducing the degree of exploration of the terrain. The results suggest that the latter factor acts to increase the variability of the performance of the algorithm.

5 Conclusions

A significant stream of literature which draws inspiration from the foraging activities of various organisms to design optimisation algorithms has emerged over the past decade. As with many natural computing algorithms, these algorithms

are populational in nature and a key component of them is a social communication mechanism which transfers information about good locations on the search landscape between members of the population. To date, relatively little attempt has been made to compare and contrast the communication mechanisms embedded in the various foraging-inspired algorithms.

In this study, drawing on the social foraging literature, we outline a taxonomy to help categorise existing optimisation algorithms which have been inspired by foraging metaphors using the nature of the underlying communication mechanism. In doing so, it is apparent that one mechanism, namely ‘recruitment at a central location and subsequently leading followers back to the food site’ has not attracted much attention thus far for the design of optimisation algorithms. We take inspiration from one example of the use of this communication mechanism in nature, the social roosting and foraging behavior of ravens, to develop a novel algorithm.

The performance of the resulting RRO algorithm is tested on a number of standard benchmark optimisation problems and is found to be competitive. A series of analyses were also undertaken on the RRO algorithm to gain insight as to the importance of its component elements. These experiments indicate the importance of the perception mechanism. An exploration of the sensitivity of the algorithm’s performance to various parameter settings was also undertaken.

The study opens up a door for follow-on work in a number of areas. This study presents an initial examination of the RRO algorithm and as already noted, no single study can exhaustively examine and benchmark all aspects of a new algorithm. Further work remains to be undertaken to expand the range of target problems/performance benchmarks examined, to further explore the sensitivity of the results of the RRO algorithm to the choice of parameter settings, and to assess scalability. As described in Sect. 4.1, certain of these such as perception range need to be scaled for the size of the search space, and hence, we would expect the results to be impacted by choice of parameter. With reference to the mechanisms in the algorithm, it is evident that elements such as perception could be operationalised in a variety of ways and it would be interesting to examine the impact on the algorithm’s performance of alternative implementations of these mechanisms.

Acknowledgments The authors would like to acknowledge the contribution of the anonymous reviewers to the improvement of this paper.

References

- Anderson J (1991) Foraging behavior of the American white pelican (*Pelecanus erythrorhynchos*) in western Nevada. *Colonial Waterbirds* 14:166–172
- Benoit-Bird K, Au W (2009) Cooperative prey herding by the pelagic dolphin *Stenella longirostris*. *J Acoust Soc Am* 125(1):125–137

- Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: from natural to artificial systems. Oxford University Press, Oxford
- Bradbury J, Vehrencamp S (2011) Principles of animal communication, 2nd edn. Sinauer Associates, Sunderland
- 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? *Behav Ecol* 13(1):42–51
- Davies N, Krebs J, West S (2012) An introduction to behavioural ecology, 4th edn. Wiley-Blackwell, Chichester
- Deygout C, Gault A, Duriez O, Sarrazin F, Bessa-Gomes C (2010) Impact of food predictability on social facilitation by foraging scavengers. *Behav Ecol* 21(6):1131–1139
- Dorigo M (1992) Optimization, learning and natural algorithms. Ph.D. thesis, Politecnico di Milano
- Dorigo M, DiCaro G (1999) Ant colony optimization: a new metaheuristic. In: Proceedings of IEEE congress on evolutionary computation (CEC 1999). IEEE Press, pp 1470–1477
- Dorigo M, Maniezzo V, Colomi A (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern Part B: Cybern* 26(1):29–41
- Dorigo M, Stützle T (2004) Ant colony optimization. MIT Press, Cambridge
- Emlen J (1966) The role of time and energy in food preference. *Am Nat* 100(916):611–617
- Fleming T (1982) Foraging strategies of plant-visiting bats. In: Kunz T (ed) Ecology of bats. Plenum Press, New York, pp 287–325
- Franklin E, Franks N (2012) Individual and social learning in tandem-running recruitment by ants. *Anim Behav* 84:361–368
- Ganesan T, Vasant P, Elamvazuthy I (2012) A hybrid PSO approach for solving non-convex optimization problems. *Arch Control Sci* 22(1):87–105
- Giraldeau LA, Caraco T (2000) Social foraging theory. Princeton University Press, New Jersey
- Grüter C, Segers F, Ratnieks F (2013) Social learning strategies in honeybee foragers: do the costs of using private information affect the use of social information? *Anim Behav*. doi:10.1016/j.anbehav.2013.03.041
- Grüter C, Leadbeater E (2014) Insights from insects about adaptive social information use. *Trends Ecol Evol* 29(3):177–184
- Leadbeater E, Florent C (2014) Foraging bumblebees do not rate social information above personal experience. *Behav Ecol Sociobiol* 68:1145–1150
- Le Dinh L, Ngoc V, Vasant P (2013) Artificial bee colony algorithm for solving optimal power flow problem. *Sci World J* 2013:159040
- Lomstedt O, Ferrari M, Chivers D (2014) Lionfish predators use flared fin displays to initiate cooperative hunting. *Biol Lett* 10:20140281
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks. IEEE Press, pp 1942–1948
- Kennedy J, Eberhart R, Shi T (2001) Swarm intelligence. Morgan Kaufman, San Mateo
- Marzluff J, Heinrich B, Marzluff C (1996) Raven roosts are mobile information centres. *Anim Behav* 51:89–103
- Marzluff J, Heinrich B (2001) Raven roosts are still information centres. *Anim Behav* 61:F14–F15
- Nakrani S, Tovey C (2004) On honey bees and dynamic server allocation in internet hosting centres. *Adaptive Behav* 12(3–4):223–240
- Passino K (2000) Distributed Optimization and Control Using Only a Germ of Intelligence. In: Proceedings of the 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 Syst Mag* 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
- Seeley T (1995) The wisdom of the hive. Harvard University Press, Cambridge
- Serfass T (1995) Cooperative foraging by North American river otters *Lutra canadensis*. *Can Field Nat* 4:458–459
- Stahler D, Heinrich B, Smith D (2002) Common ravens, *Corvus corax*, preferentially associate with grey wolves, *Canis lupus*, as a foraging strategy in winter. *Anim Behav* 64:283–290
- Stephens D, Krebs J (1986) Foraging theory. Princeton University Press, New Jersey
- von Frisch K (1967) The dance language and orientation of bees. Harvard University Press, Cambridge
- Viswanathan G, da Luz M, Rapposo E, Stanley E (2011) The physics of foraging: an introduction to random searches and biological encounters. Cambridge University Press, Cambridge
- 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. *Anim Behav* 44(3):501–518
- Wray M, Klein B, Seeley T (2012) Honey bees use social information in waggle dances more fully when foraging errors are more costly. *Behav Ecol* 23(1):125–131
- Wright J, Stone R, Brown N (2003) Communal roosts as structured information centres in the raven, *Corvus corax*. *J Anim Ecol* 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