Emergence of an optimal search strategy from a simple random walk

Tomoko Sakiyama and Yukio-Pegio Gunji

Faculty of Science, Department of Earth and Planetary Science, Kobe University, Nada, Kobe 657-8501, Japan

In reports addressing animal foraging strategies, it has been stated that Lévy-like algorithms represent an optimal search strategy in an unknown environment, because of their super-diffusion properties and power-law-distributed step lengths. Here, starting with a simple random walk algorithm, which offers the agent a randomly determined direction at each time step with a fixed move length, we investigated how flexible exploration is achieved if an agent alters its randomly determined next step forward and the rule that controls its random movement based on its own directional moving experiences. We showed that our algorithm led to an effective food-searching performance compared with a simple random walk algorithm and exhibited super-diffusion properties, despite the uniform step lengths. Moreover, our algorithm exhibited a power-law distribution independent of uniform step lengths.

1. Introduction

When agents can exploit previously acquired knowledge, such as profitable food locations or the cost required to reach those locations, selecting an effective route entails solving a so-called deterministic walk problem, such as the ‘travelling tourist problem’ [1,2]. In contrast, in the absence of prior knowledge, ‘chance’ or ‘randomness’ represents a more appropriate scenario for exploring an unknown environment, and many random search algorithms, such as the Lévy walk or the Brownian walk algorithm, are effective for random exploration and have been very well studied [3–7].

The Lévy walk is defined as a process whereby an agent takes steps of length $l$ at each time, and $l$ shows a probability density distribution with a power-law tail

$$\text{P}(l) \sim l^{-\mu},$$

where $1 < \mu \leq 3$. A Lévy walk with $\mu \approx 2$ results in optimal random searches in environments with randomly distributed food sites [8]. For $\mu > 3$, the Lévy walk converges to Brownian motion. In environments with extremely abundant resources, the Lévy and Brownian walk food-searching algorithms result in similar exploration efficiencies. However, the Lévy walk is more suitable than the Brownian walk for searching in low-density environments, and Lévy foragers can survive for longer periods of time without extinction [9,10].

To determine which algorithm is more suitable for modelling animal movement behaviour, recent analyses have made comparisons between an animal’s actual movements and model simulations and have attempted to determine whether the movements correspond to a Lévy walk [8,11] or to a Brownian walk [12]. However, it appears that, in searching for food, animals do not depend on a particular type of movement modelled by a certain algorithm, but, rather, the rules guiding their movement change depending on the context [13,14].

Therefore, in determining an animal’s foraging behaviour, its interactions with the environment and/or its effects on the surrounding world, the issue that arises is not only the determination of the optimal algorithm for a random walk, but also the estimation of how these flexible behaviours arise from what that animal is doing [15].

When an agent’s movement obeys a certain distribution, that agent behaves exclusively according to the probabilistic rule determining the distribution.
From this perspective, the direction and step length are completely determined by a roll of the dice, which is independent of whether the dice are biased.

If an agent behaving according to a random movement rule sometimes alters the next step forward or the rule itself, then the resulting movement does not follow the distribution resulting from a random roll of the dice and the application of the above-mentioned rule. Thus, a balancing of chance with necessity or of a passive with an active movement mode must be achieved by the agent’s interaction with its surroundings.

Here, starting with a simple random walk with uniform step length and discrete angle distributions, we consider an agent who alters the rule that controls him based on his own experiences, and we propose an effective exploration algorithm (re-valued, REV, a random walk algorithm) despite the fixed step lengths. In this study, the agent experiences random directional biases; based on these biases, he disregards the rule that he had been obeying until that time and changes the rule itself using his memory, which was obtained by chance, to implement an indeterminate forward step that is independent of the rule. There are few studies detailing agent movements for agents, who proceed with unknown forward steps based on limited memories of their own experiences [16]. We also report the emergence of a power-law distribution that is independent of step length.

2. Material and methods

2.1. Two-dimensional simple random walk

Each trial is run for a maximum of 10 000 time steps. First, we use a simple two-dimensional random walk to set the simulation stage for each trial and set the agent at the origin. In this algorithm, a random number from 0.00 to 1.00 is selected at each time step. We assume that the agent moves in two-dimensional square lattices. The agent selects one direction among four discrete directions, $-x$, $+x$, $-y$, or $+y$, depending on a random number classified to one of the following four intervals, [0.0, 0.25], [0.25, 0.5], [0.5, 0.75] or [0.75, 1.0]. For example, if the random number is 0.23 at the $t$th step (i.e. $R(t) = 0.23$), the agent steps forward $-1$ along the $x$-axis. We create an agent that, depending on its experiences, alters the rule that controls that agent’s movement directions.

2.2. Two-dimensional re-valued random walk

In our REV random walk model, at the $t$th step, the agent has a memory that gives it access to four random numbers, from $R(t)$ to $R(t - 3)$. This model implies that the agent has a limited memory. While the agent in the simple random walk has access to only one random number, $R(t)$, and the interval controlling the agent’s move is fixed, the interval in the REV random walk, $I(t)$, $s = 0, 1, 2, 3$, is changed using the four random numbers when the successor of the agent’s directional moves exceeds a threshold number. The four intervals are initially set to $I_0(t) = [0.0, 0.25], I_1(t) = [0.25, 0.5], I_2(t) = [0.5, 0.75]$ and $I_3(t) = [0.75, 1.0]$. The next walk is defined by the following equations:

$$ (x(t + 1), y(t + 1)) = (x(t), y(t)) + (u(t), v(t)), \quad \text{if } R(t) \in I_0(t) $$

$$ = (x(t) + 1, y(t)), \quad \text{if } R(t) \in I_1(t) $$

$$ = (x(t), y(t) + 1), \quad \text{if } R(t) \in I_2(t) $$

$$ = (x(t) + 1, y(t) + 1), \quad \text{if } R(t) \in I_3(t). $$

Thus, every trial starts as a simple random walk.

A directional move that consists of a series of the same move, such as $+x$, $-x$, $+y$, $-y$, etc., is counted as follows:

$$ E(x(t + 1)) = E(x(t)) + 1, \quad \text{if } R(t) \in I_0(t), $$

$$ R(t - 2) \in I_1(t), \quad R(t - 3) \in I_2(t), \quad \text{for some } s \in \{0, 1, 2, 3\}; $$

$$ E(x(t + 1)) = E(x(t)), \quad \text{otherwise.} $$

For example, if the agent’s directional move from $t - 3$ to $t$ corresponds to a $-y, -y, -y, -y$ sequence, it implies

$$ R(t) \in I_3(t), \quad R(t - 1) \in I_2(t), \quad R(t - 2) \in I_2(t), \quad R(t - 3) \in I_2(t). $$

Therefore,

$$ E(x(t + 1)) = E(x(t)) + 1. $$

However, if the agent’s directional move from $t - 3$ to $t$ corresponds to $-y, -y, -y, -y$, then it implies

$$ R(t) \in I_3(t), \quad R(t - 1) \in I_3(t), \quad R(t - 2) \in I_2(t), \quad R(t - 3) \in I_2(t). $$

Therefore,

$$ E(x(t + 1)) = E(x(t)). $$

If $E(x(t))$ exceeds a threshold number, $\theta$, the four intervals are changed as follows:

$$ I_0(t) = [0.0, NR(t)], $$

$$ I_1(t) = [NR(t) + NR(t) + NR(t - 1)], $$

$$ I_2(t) = [NR(t) + NR(t - 1) + NR(t - 2)], $$

and

$$ I_3(t) = [NR(t) + NR(t - 1) + NR(t - 2), 1]. $$

where $NR(t - k), k = 0, 1, 2, 3$, is defined by

$$ NR(t - k) = \frac{R(t - k)}{\sum I_i R(t - i)}. $$

When $E(x(t))$ exceeds $\theta$, $E(x(t))$ is reset to 0.

For example, if $E(x(t))$ exceeds $\theta$ and $R(t - 3) = 0.20, R(t - 2) = 0.15, R(t - 1) = 0.10$ and $R(t - 3) = 0.20$, then $NR(t - k)$ for $k = 0, 1, 2, 3$ is given as follows:

$$ NR(t - 3) = \frac{0.20}{(0.20 + 0.15 + 0.10 + 0.20)} = 0.31, $$

$$ NR(t - 2) = \frac{0.15}{(0.20 + 0.15 + 0.10 + 0.20)} = 0.23, $$

$$ NR(t - 1) = \frac{0.10}{(0.20 + 0.15 + 0.10 + 0.20)} = 0.15 $$

and

$$ NR(t) = \frac{0.20}{(0.20 + 0.15 + 0.10 + 0.20)} = 0.31. $$

Thus, $I_0(t)$ for $s = 0, 1, 2, 3$ is given as follows:

$$ I_0(t) = [0.0, 0.31], $$

$$ I_1(t) = [0.31, 0.31 + 0.15] = [0.31, 0.46], $$

$$ I_2(t) = [0.31 + 0.15, 0.31 + 0.15 + 0.23] = [0.46, 0.69] $$

and

$$ I_3(t) = [0.31 + 0.15 + 0.23, 0.31 + 0.15 + 0.23 + 0.31] = [0.69, 1.00]. $$

Finally, $E(x(t))$ is reset to 0.

The flow chart of the REV random walk algorithm is presented in figure 1.

In our simulations, $\theta$ is set to 4, and we discuss the agent’s behavioural differences between $\theta = 4$ and $\theta = 8, 64$ and 128.

In the simple random walk algorithm, the probability of a random number falling into a particular interval is 0.25, as the intervals are divided equally. Thus, the agent moves north, east, west or south with equal probability. In contrast, the intervals in the REV random walk algorithm are divided unequally, depending on the agent’s history.
3. Results

3.1. Threshold $\theta = 4$

Figure 2 shows an example of an agent trajectory for each algorithm. Compared with a simple random walk, the agent can reach farther areas by following the REV random algorithm while spending longer amounts of time in certain areas. In fact, the root-squared distance between the start and endpoints was significantly larger in the REV random algorithm compared with the simple random walk algorithm (REV random versus random: $N_{\text{REV-R}} = 100$, $N_B = 100$, $\text{MEAN}_{\text{REV-R}} = 246.0 \pm 149.5$, $\text{MEAN}_R = 84.4 \pm 52.1$, Mann–Whitney $U$-test, $U = 1446.0$, $p < 1.0 \times 10^{-15}$).

In the simple random walk analysis, it is known that the mean-squared distance and time step are related by the following relation [5,17]:

$$\langle R^2 \rangle \sim t^2H.$$

Parameter $H$ is determined depending on the algorithm ($H > 1/2$ for a Lévy walk (super-diffusion), $H = 1/2$ for a correlated random walk with more than $10^2$ time steps or for a Brownian walk (diffusion)). Figure 3 shows the mean-squared distance and the time step obtained from our algorithm (the averaged $R^2$ was obtained from 1000 trials at each discrete time step). The fit for parameter $H$ according to the above model was $H \sim 0.61$, indicating that super-diffusion was achieved.
(N = 19, R^2 = 0.99, F = 2,109.00, p < 1.0 × 10^{-15}). We also checked the degree of diffusion by restricting the fit to the largest five values of the time step because it appeared that H converged to H ∼ 1/2. However, we found that the fit was H ∼ 0.57 (N = 5, R^2 = 0.99, F = 1,108.00, p < 1.0 × 10^{-15}).

To evaluate the exploration efficiency, we established four feeder sites consisting of 10 × 10 squares distributed as shown in figure 4b. The agent is initially set at the centre of the foraging region (starting point). If the agent reaches one of the four fixed sites, the time step required to reach the site is the output. We conducted 100 trials based on each algorithm and estimated how frequently the agent could reach an individual feeder in a limited period of 10,000 time steps. Figure 4 presents a comparison of the performance of the REV random and simple random walk algorithms. If the feeder sites are closely located, within 10–20 distance units from the starting point (figure 4b), an agent can reach a feeder site in 10,000 steps under both algorithms. However, if the feeder sites are located farther away from the starting point, at 110–120 distance units, the performance of the REV random algorithm is better than that of the simple random walk algorithm (number of successes in 10,000 steps, REV random versus random, 10–20: 96 versus 98, Fisher’s test, p = 0.36, n.s.; 60–70: 64 versus 62, χ^2-test, χ^2 = 0.022, p = 0.88, n.s.; 110–120: 32 versus 13, χ^2-test, χ^2 = 9.29, p < 0.001).

We estimated not only whether an agent reaches the target but also the time required to reach the feeder (figure 4c). There is no significant difference between the two algorithms when the feeder is 10–20 distance units away from the centre (REV random versus random, N_{REV-R} = 96, N_R = 98, MEAN_{REV-R} = 303.7 ± 753.2, MEAN_R = 235.4 ± 302.8, Mann–Whitney U-test, U = 4592.0, p = 0.69, n.s.; figure 4c). However, in the other cases, the agent following the REV random algorithm can reach the feeder site significantly faster than that following the random walk algorithm, even when the feeder is placed at a middle-range distance away from center (REV random versus random, 60–70: N_{REV-R} = 64, N_R = 62, MEAN_{REV-R} = 2589.0 ± 2239.4, MEAN_R = 5003.0 ± 2480.1, Mann–Whitney U-test, U = 761.00, p < 0.001, 110–120: N_{REV-R} = 32, N_R = 13, MEAN_{REV-R} = 3596.0 ± 2467.6, MEAN_R = 6670.0 ± 1657.9, Mann–Whitney U-test, U = 761.00, p < 0.001). Thus, the REV random algorithm is more efficient at exploring both nearby (but not very close) and distant areas within a certain limited time.

When considering the final location of the agent after 10,000 time steps for 100 trials using each algorithm, we found that there was no relationship between the x and y coordinates for either algorithm (REV random, R^2 = −0.01, F = 0.0041, p = 0.95, n.s.; random, R^2 = −0.0057, F = 0.44, p = 0.51, n.s.; electronic supplementary material, figure S1).

Figure 5 provides a histogram of the time duration (in units of time steps) between rule change events. Interestingly, we found that the histogram shows a power-law distribution. (power-law versus exponential, N = 57, μ = 1.85, λ = 0.0063, χ^2 = 0.97, ω_{exp} = 0.03, goodness of fit test: G = 41.39, d.f. = 36, p = 0.25) [18]. We discuss the significance of this finding in §3.2.

3.2. Difference between θ = 4 and other threshold values (8, 64 and 128)

We further investigated how θ threshold values affect the exploration process. Here, we compare a threshold of 4 with a threshold of 8 using the same analytical process described earlier (i.e. based on the ability to reach a target within a certain distance or within a certain time limit). According to figure 6a, the threshold 8 REV random algorithm is not as efficient as the threshold 4 version with respect to the exploration efficiency (number of successes in 10,000 steps for threshold 4 versus threshold 8, 10–20: 96 versus 92, χ^2-test, χ^2 = 0.80, p = 0.37, n.s.; 60–70: 64 versus 36, χ^2-test, χ^2 = 14.6, p < 0.001; 110–120: 32 versus 27, χ^2-test, χ^2 = 0.38 p = 0.54, n.s.). We also evaluated the distance between the start and endpoints after 10,000 steps for four different threshold values (4, 8, 64 and 128). Figure 6b shows that the distance increases depending on the threshold values, suggesting that changing the current rule affects the agent’s bias towards movement in a straight line (threshold 4 versus threshold 8: N4 = 100, N8 = 100, MEAN4 = 246.0 ± 149.5, MEAN8 = 295.6 ± 174.5, Mann–Whitney U-test, U = 4122.5, p < 0.05; threshold 4 versus threshold 64: N4 = 100, N64 = 100, MEAN4 = 246.149.5 ± 149.5, MEAN64 = 526.2 ± 318.5, Mann–Whitney U-test, U = 2172.0, p < 1.0 × 10^{-15}). However, we observed no significant difference between a threshold of 4 and a threshold of 128 (threshold 4 versus threshold 128: N4 = 100, N128 = 100, MEAN4 = 246.0 ± 149.5, MEAN128 = 323.4 ± 259.2, Mann–Whitney U-test, U = 4436.0, p = 0.17, n.s.). As can be seen from electronic supplementary material, figure S2, the changing of the current rule occurs less often as threshold values increase, and the areas with a concentrated presence become larger until the first rule change.

4. Discussion

Our results indicate that a flexible random walk similar to a Lévy walk will be achieved using probability intervals that change based on the agent’s history and that have a uniform step length distribution. As mentioned earlier, this algorithm results in super-diffusion. Therefore, it will be possible to attain long-range exploratory efficiency [5,17]. Although we assumed a very low food density in our simulations, the important result is that efficiency was achieved through a balance between near and distant areas. Effective long-range models,
In the algorithm, the current probability intervals change depending on an agent’s own experiences. The agent only knows its most recent four random numbers concretely. Based on this event history, the agent experiences directionally biased movement. If the number of consecutive steps in the same direction reaches the threshold value, the current probability intervals are changed. At this moment (the $i$th step), the values of $R(t - 3)$ to $R(t - 1)$ are based on the probability intervals defined by the previous rule, and only $R(t)$ is free from these intervals (e.g. $R(t - 3) \in I_0(t - 1)$, $R(t - 2) \in I_0(t - 1)$, $R(t - 1) \in I_0(t - 1)$ and $R(t) \in I_1(t - 1)$). Therefore, using these four numbers provided by chance, the agent takes a new step associated with new probability intervals. Thus, the agent’s move forward is biased. However, the accumulating directional bias causes the probability intervals to change at a higher frequency. When the direction of movement is biased, it implies that one of the four probability interval regions is larger than the others. Therefore, the range of random numbers that fall within that region is large. Thus, the avoidance of dominance of one region over the others is achieved upon the next interval change, which occasionally produces local areas of concentrated presence. We obtain a power-law distribution based on the distribution of time intervals between interval change events. By altering the rule and taking the next step with ambiguity, the agent can involve the world actively. However, the agent cannot change the rule with complete freedom. Instead, he produces directional inequality by choosing a direction using limited information provided by chance. Thus, having to take such an action as a result of chance might be associated with achieving effective exploration.

In our algorithm, the agents only have knowledge of the most recent four random numbers determining their movement. Thus, the algorithm does not require agents to necessarily have high-performance memory abilities. By deviating from one-to-one behaviour with respect to a certain fixed rule, i.e. a simple diffusion-predictive moving frame, it is possible for the agent to display behaviour beyond a predictive frame, i.e. to achieve super-diffusion. Therefore, our algorithm could represent an abstract explanation for biological phenomena focusing on an agent’s degree of activity.

It was recently reported that the nest-searching behaviour of foraging ants corresponds to a Brownian walk [19]. However, in some species of insects and fishes, searching behaviour changes in response to certain environmental alterations linked to extinction, e.g. the amount of available food [20,21]. In fact, ants show different searching behaviours depending on the relevance of the visual information provided to them [22]. Dramatic behavioural changes caused by the agent’s involvement and interactions with the environment and other agents, such as nest-mates, might be possible [13,22,23]. Balancing an agent’s own degree of activity and passivity
based on limited available information will achieve such flexible behaviours [24].

Acknowledgements. The authors thank the Japan Society for the Promotion of Science for financial support (B2130093).