Modeling Fast and Robust Ant Nest Relocation using Particle Swarm Optimization

Hideyasu Sasaki

National Institute of Information and Communications Technology, Tokyo 184-8795, Japan
hsasaki@nict.go.jp

Abstract

Ant nest relocation is smoother and swifter than the same process undertaken by any other animal. Within the population of ants, the ratio that participates in nest relocation is only 58.0% at best and 31.0% at worst. Does such a low active ratio improve or deteriorate ant nest relocation? In this study, we use a particle swarm optimization (PSO) algorithm to simulate real-world ant nest relocation. Our PSO-based algorithm duplicates the velocity and position of an inactive particle (representing an inactive ant) with the velocity and position of an active particle (representing an active ant). The number of particles that the algorithm computes is dramatically reduced, and the global best position can be identified at an early stage. In a series of simulations, our algorithm performs significantly better and faster with active ratios of 15%, 30%, 35%, 45%, 55%, 60%, and 75%–95% than with the full 100% active ratio. We confirm the robust and stable performance of our algorithm at active ratios of 60%, 80%, and 85%. Clustering of the simulation results shows that low active ratios improve ant nest relocation. Furthermore, three field studies carried out by biology experts empirically demonstrate that we have successfully modeled and simulated real-world ant nest relocation using our PSO-based algorithm.

Introduction

Ants have a reputation for being hardworking animals. For instance, worker ants can relocate an entire nest within a short period of time (Burd et al., 2002). To the best of our knowledge, no other animals can compete with ants in terms of the speed and smoothness of the nest relocation process.

Real-world problems involve multiple objectives that are often in conflict with one another, yet should be optimized simultaneously (Ali et al., 2016a,b; Awad et al., 2013, 2016, 2017). The impressive speed and precise movement of ant nest relocation provides an intuitive solution for multi-objective optimization problems.

The swarm behavior of ant nest relocation has attracted researchers in the computational intelligence community. Various models and simulations have been proposed for real-world ant nest relocation (Chowdhury et al., 2002; John et al., 2008, 2009; Sasaki and Leung, 2013; Sasaki, 2017). However, ant nest relocation has not yet been studied as a multi-objective optimization problem in terms of the precision and speed of the swarm behavior.

Over the last decade, biologists have reported that many workers of the *Temnothorax* ant species remain inactive and do not work at all (Dornhaus et al., 2008; Charbonneau et al., 2017). Inactive worker ants lick their own bodies, and active worker ants feed and relocate them. During ant nest relocation, active workers shoulder these inactive workers on their backs and move them toward the new nest (Dornhaus et al., 2008). Among *Temnothorax* ant species, behavioral biologists have rigorously studied *Temnothorax albipennis*, also known as the rock ant, for its "scale-free" swarm behavior of nest relocation. Regarding the *Temnothorax albipennis* ant, the active ratio within the populations of worker ants in a colony (hereafter referred to as the *active ratio*) is 58.0% at best and 31.0% at worst. Variations in the active ratio do not depend on the size of the population of worker ants in the colony (Dornhaus et al., 2008). Inactive workers act as a reserve labor force for replacing active workers, however, active workers do not replace inactive workers that have been removed from the nest Charbonneau et al. (2017).

An open problem on the scale-free swarm behavior of the *Temnothorax albipennis* ant is whether such a low active ratio improves or deteriorates ant nest relocation compared with the full 100% active ratio performance (Sasaki, 2018a,b, 2019). Although researchers have proposed models and simulations that accurately describe ant nest relocation, they have not solved this problem. A positive answer to this problem would provide technological inspiration for promising swarm-based algorithms in the context of computational intelligence and in specific aspects of swarm intelligence regarding the active ratio within the populations of computational agents.

In previous work, we have shown that the frequency of mutual contact among worker ants determines the active ratio in ant nest relocation (Sasaki and Leung, 2013; Sasaki, 2017). The frequency of mutual contact rises from 1 to 40 times per minute, and the active ratio gradually rises to the maximum. In this study, we use a particle swarm optimization (PSO) algorithm to model and simulate real-world ant

nest relocation. In particular, we investigate the relation between the active ratio and the performance of worker ants in nest relocation. The simulation results show that our PSO-based algorithm performs significantly better and faster at active ratios that are lower than the full 100% active ratio. We confirm the robust and stable performance of our algorithm at certain low active ratios. The simulation results are supported by three field studies that were carried out by expert ant biologists.

Model

To model ant nest relocation while focusing on the active ratio, we employ a population-based optimization method, namely the original or canonical PSO algorithm developed by Kennedy and Eberhart (1995), and refer to several popular PSO variants (Clerc and Kennedy, 2002; Cleghorn and Engelbrecht, 2018).

The swarm behavior of ant nest relocation can be described using the velocity and position of an ant (John et al., 2008, 2009). Worker ants find the shortest path toward a new nest. The shortest path is found when ants move from an old nest to a new nest and when the distance of nest relocation is minimized. The PSO algorithm describes the velocity and position of particles. There are various optimization methods, and the PSO algorithm and its variants are some of the best techniques for modeling the swarm behavior of ant nest relocation, as they can easily focus on the velocity and position of an ant.

We model the swarm behavior of ant nest relocation using the velocity and position of a particle with two types of "best" positions. Our PSO-based algorithm is initialized with a group of random particles and searches for optima by updating the velocity and position of a particle in each iteration of the algorithm. At every iteration, the velocity and position of a particle are updated in comparison with the two types of best positions, which are evaluated with a fitness function called a "cost function". One best position is the "personal best position" of a particle, and the other is the "global best position" within the swarm of particles. After finding the two best positions, the velocity and position of a particle are updated at every iteration (Pereira, 2010).

First, our PSO-based algorithm minimizes the output of the cost function with the velocity and position of an "active" particle. The output of the cost function is equivalent to the distance that an "active" worker ant moves from an old nest to a new nest. An active particle that represents an active worker ant is defined and updated in the same way as in the original PSO algorithm developed by Kennedy and Eberhart (1995). The velocity $v \in R$ and position $x \in R$ of active particle $i \in N$ in a population of size S at a discrete time step $t \in N$, which is the t-th number of the iteration process, are defined and updated, respectively, as follows:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1(t)\{p_i(t) - x_i(t)\} + c_2 r_2(t)\{g(t) - x_i(t)\}, \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (2)$$

where the random components $r_{1,k}(t)$ and $r_{2,k}(t) \sim U(0,1)$, k is the vector component of v_i and x_i, and the coefficients c_1, c_2, and ω are the personal (cognitive), social, and inertia weights, respectively. The position p represents the personal "best" position that particle i has visited, where "best" means the location where the particle had obtained the lowest cost function evaluation. The position g represents the global "best" position that the particles in the neighborhood of the i-th particle have visited.

The vector component k of the velocity v and position x of particle i represents the direction of movement of the particle. A particle represents a worker ant that moves on the ground. The direction of movement of a particle is defined in a discrete form that consists of eight two-dimensional directions, i.e., $K = 8$, as shown in Fig. 1. An ant not only moves forward and backward, but also shifts in the transverse direction and moves and shifts at the same time. Therefore, we define the direction of movement of a particle that represents a worker ant in a discrete form that consists of eight two-dimensional directions.

The personal best position p of active particle i and the global best position g at time t (or the t-th iteration) are defined and updated with the cost function f as follows:

$$p_i(t+1) = \begin{cases} x_i(t+1) & \textbf{if } f(x_i(t+1)) < f(p_i(t)) \\ p_i(t) & \textbf{else}, \end{cases} \quad (3)$$

$$g(t+1) = \underset{p_i(t) \in P(t)}{\arg\min} f(p_i(t+1)), \quad (4)$$

where the cost function f takes the form

$$f(x_i^k(t)) = \sum_{i \in S} \sum_{k \in K} ||x_i^k(t)||^2, \quad (5)$$

where $||\cdot||$ denotes the Euclidean norm.

In the modeling process, we choose the sphere function for the cost function f. This simple function provides a straightforward model that allows us to assess the benefits and limitations of our PSO-based algorithm in simulating ant nest relocation. An ant moves around the two-dimensional ground. There are many functions that could

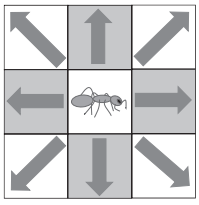

Figure 1: Particle representing a worker ant has eight two-dimensional directions of movement on the ground.

calculate such two-dimensional distances over which the particles move. Almost all field studies carried out by experts in ant biology simply use the Euclidean distance that the ants relocated a nest (Burd et al., 2002; Pratt, 2005b; Franks et al., 2006; Dornhaus et al., 2008, 2009; Charbonneau et al., 2017). Researchers in the computational intelligence community tend to have followed the biological practice of field researchers (Chowdhury et al., 2002; John et al., 2008, 2009; Sasaki and Leung, 2013; Sasaki, 2017, 2018a,b, 2019). We use the sphere function given by (5) through the index i and vector component k for the cost function f, though our PSO-based algorithm could easily use other functions (as will be considered in future work). The cost function f represents the distance between the position x of particle i (representing a worker ant) and the global best position g that leads to the shortest path toward point zero (representing a new nest as the target for movement). The distance that a particle moves can take a positive value or a negative value from point zero, so the square of the difference is computed. The accumulated square of the difference at the position of a particle should be minimized through vector component k.

Next, we define the swarm behavior of "inactive" worker ants in modeling ant nest relocation. An inactive worker ant does not relocate the nest by itself. Instead, active worker ants carry the inactive ants toward the new nest. In our PSO-based algorithm, the velocity v and position x of "inactive" particle i (representing an inactive worker ant) at time t (or the t-th iteration) are defined and updated, respectively, as follows:

$$v_i(t+1) = v_{i-1}(t+1), \tag{6}$$

$$x_i(t+1) = x_{i-1}(t+1) = x_{i-1}(t) + v_i(t+1). \tag{7}$$

The personal best position p of inactive particle i and the global best position g at time t (or the t-th iteration) are defined and updated with the cost function f given by (5), respectively, as follows:

$$p_i(t+1) = p_{i-1}(t+1), \tag{8}$$

$$g(t+1) \; = \; p_i(t+1). \tag{9}$$

Fig. 2 shows that our PSO-based algorithm duplicates the velocity and position of inactive particle i with the velocity and position of active particle i-1 that is closest to inactive particle i. This duplication of the velocity and position of inactive particles reduces the number of particles that the algorithm computes, allowing the global best position to be identified at an early stage of the iteration.

Simulation

Setting

Our PSO-based algorithm comprises the coefficients c_1, c_2, and ω that are the personal, social, and inertia weights. The parameter configurations follow the earlier work of Clerc and Kennedy (2002) and refer to the Yarpiz Project (2015). Cleghorn and Engelbrecht (2018, 2016) have developed a sophisticated theory for determining parameter configurations and enhanced the stability criteria of particles. However, we use the conservative approach of the original PSO algorithm in our simulations. The memory of the *Temnothorax albipennis* ant lasts for only 1–2 min (Pratt, 2005a). It is reasonable to assume that the parameter configurations for nest relocation are genetically coded in the brains of the ants in a static, rather than dynamic, form. The sophisticated parameter configurations developed by Cleghorn and Engelbrecht (2018, 2016) might lead to overfitting in modeling the swarm behavior of ant nest relocation. Therefore, we follow the parameter configurations of the original PSO algorithm given by:

$$c_1 = \chi\varphi_1, c_2 = \chi\varphi_2,$$

$$\omega = \chi = 2\kappa/|2 - \varphi - \sqrt{(\varphi^2 - 4\varphi)}| \,, \tag{10}$$

where $\kappa = 1$, $\varphi = \varphi_1 + \varphi_2$, and $\varphi_1 = \varphi_2 = 2.05$.

A lower limit v_{min} and an upper limit v_{max} of the velocity v are assigned through index i by:

$$v_{min} = -v_{max} \;\text{ and }\; v_{max} = 0.2(x_{max} - x_{min}) \,. \tag{11}$$

Similarly, a lower bound x_{min} and an upper bound x_{max} of the position x are assigned through index i by:

$$[x_{min}, x_{max}] = [-10, 10] \,. \tag{12}$$

The lower and upper bounds are applied to the personal best position p through index i and to the global best position g.

The damping ratio ω_{damp} of the inertia weight is set to 1.

Active and inactive particles are assigned according to active ratios of 5%, 10%, ..., 95%, and 100%, as listed in Table 1. During ant nest relocation, the worker ants gather and form a basic unit of a swarm. Behavioral biologists report that the minimal size of this basic unit consists of 20 individuals, and so this is the smallest population size of a swarm of

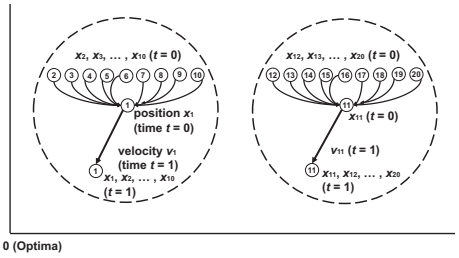

Figure 2: Inactive particles $i = 2, ..., 10/12, ..., 20$ duplicate the velocities $v_{i=1/11}(t = 1)$ and positions $x_{i=1/11}(t = 0)$ of active particles $i = 1/11$ that are closest to the inactive particles at the active ratio 10%.

Active Ratio	Active Particle "ⓝ" & Inactive Particle "◯"
5%	① ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯
	◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯
10%	① ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯
	⑪ ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯
15%	① ◯ ◯ ◯ ◯ ◯ ◯ ⑧ ◯ ◯
	◯ ◯ ◯ ◯ ⑮ ◯ ◯ ◯ ◯ ◯
20%	① ◯ ◯ ◯ ◯ ⑥ ◯ ◯ ◯ ◯
	⑪ ◯ ◯ ◯ ◯ ⑯ ◯ ◯ ◯ ◯
25%	① ◯ ◯ ◯ ⑤ ◯ ◯ ◯ ⑨ ◯
	◯ ◯ ⑬ ◯ ◯ ◯ ⑰ ◯ ◯ ◯
30%	① ◯ ◯ ◯ ⑤ ◯ ◯ ⑧ ◯ ◯
	⑪ ◯ ◯ ◯ ⑮ ◯ ◯ ⑱ ◯ ◯
35%	① ◯ ◯ ④ ◯ ◯ ⑦ ◯ ◯ ⑩
	◯ ◯ ⑬ ◯ ◯ ⑯ ◯ ◯ ⑲ ◯
40%	① ◯ ◯ ④ ◯ ⑥ ◯ ◯ ⑨ ◯
	⑪ ◯ ◯ ⑭ ◯ ⑯ ◯ ◯ ⑲ ◯
45%	① ◯ ◯ ④ ◯ ◯ ⑦ ◯ ⑨ ◯
	⑪ ◯ ⑬ ◯ ⑮ ◯ ⑰ ◯ ⑲ ◯
50%	① ◯ ③ ◯ ⑤ ◯ ⑦ ◯ ⑨ ◯
	⑪ ◯ ⑬ ◯ ⑮ ◯ ⑰ ◯ ⑲ ◯
55%	① ② ◯ ④ ⑤ ◯ ⑦ ◯ ⑨ ◯
	⑪ ◯ ⑬ ◯ ⑮ ◯ ⑰ ◯ ⑲ ◯
60%	① ◯ ③ ◯ ⑤ ⑥ ◯ ⑧ ◯ ⑩
	⑪ ◯ ⑬ ◯ ⑮ ⑯ ◯ ⑱ ◯ ⑳
65%	① ② ◯ ④ ⑤ ◯ ⑦ ⑧ ◯ ⑩
	⑪ ◯ ⑬ ⑭ ◯ ⑯ ⑰ ◯ ⑲ ◯
70%	① ② ③ ◯ ⑤ ⑥ ◯ ⑧ ⑨ ◯
	⑪ ⑫ ⑬ ◯ ⑮ ⑯ ◯ ⑱ ⑲ ◯
75%	① ② ③ ◯ ⑤ ⑥ ⑦ ◯ ⑨ ⑩
	⑪ ◯ ⑬ ⑭ ⑮ ◯ ⑰ ⑱ ⑲ ◯
80%	① ② ③ ④ ◯ ⑥ ⑦ ⑧ ⑨ ◯
	⑪ ⑫ ⑬ ⑭ ◯ ⑯ ⑰ ⑱ ⑲ ◯
85%	① ② ③ ④ ⑤ ⑥ ◯ ⑧ ⑨ ⑩
	⑪ ⑫ ⑬ ◯ ⑮ ⑯ ⑰ ⑱ ⑲ ◯
90%	① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ◯
	⑪ ⑫ ⑬ ⑭ ⑮ ⑯ ⑰ ⑱ ⑲ ◯
95%	① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩
	⑪ ⑫ ⑬ ⑭ ⑮ ⑯ ⑰ ⑱ ⑲ ◯
100%	① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩
	⑪ ⑫ ⑬ ⑭ ⑮ ⑯ ⑰ ⑱ ⑲ ⑳

Table 1: Active and inactive particles at the respective active ratios.

worker ants (Geraghty et al., 2007). Following this finding, we divided the population of particles into groups of 20 particles consisting of active and inactive particles. In every 20 particles, an active particle brings an almost equal number of inactive particles that are closest to the active particle toward the target of zero for the cost function. This simple simulation setting is straightforward and rational for task sharing.

We considered population sizes S ranging from 20 to 400, because swarms of workers from the *Temnothorax albipennis* ant species have been observed to range from a minimum of 20 ants to a maximum of 400 (Geraghty et al., 2007).

The maximum number of iterations was set to 40, 80, 120, 160, 200, 240, 400, or 1000, equivalent to 1, 2, 3, 4, 5, 6, 10, or 25 cycles of mutual contact among ants during nest relocation at a frequency of 40 contacts per minute, respectively.

Our simulations consisted of 50 runs of the PSO-based algorithm for each population size.

Results

The performance of our PSO-based algorithm was measured by the average global best position (hereafter referred to as the *global best*) and the average execution time (hereafter referred to as the *execution time*) of the algorithm. These metrics evaluate the precision of convergence and the speed of convergence toward the target point of zero for the cost function, respectively. Smaller values of the global best indicate better performance, and a smaller execution time denotes faster performance. The simulations were performed on a Windows 10 Pro (64-bit) laptop PC equipped with a Core i7-7700 3.6 GHz quad-core processor with 16 GB of RAM.

We identified the active ratios at which our PSO-based algorithm performed better and faster than the full 100% active ratio for all population sizes.

We measured the global best and execution time for all population sizes at active ratios of 5%, 10%, ..., 95%, and 100%. Statistical testing was employed to determine whether our PSO-based algorithm performed better and faster than with the full 100% active ratio. The null hypothesis assumed that the global best and execution time were equal to those given by the full 100% active ratio for all population sizes. We used a two-tailed F-test to measure the two-sample variances and a two-tailed t-test for two-sample equality assuming equal/unequal variances.

Behavioral biologists divide the entire population into small populations of 20–100 and large populations of 100–400 (Franks et al., 2006; Dornhaus et al., 2008, 2009). We examined the active ratios at which our PSO-based algorithm performed better and faster than the full 100% active ratio in the entire population sizes, the small population sizes, large population sizes, and population sizes of 57 and 165, which are the medians of the small population sizes and large population sizes, respectively (Dornhaus et al., 2008). The simulation results of our PSO-based algorithm for these population sizes are omitted from this study due to page limitation.

Discussion

We now discuss the simulation results and show that our PSO-based algorithm performed significantly better and faster than the full 100% active ratio at low active ratios. We also confirm the robust and stable performance of our algorithm at certain low active ratios. Three field studies on ant nest relocation that were conducted by ant biologists are then consulted. Their records provide sufficient support to demonstrate that our PSO-based algorithm performed like real-world ant swarm behavior at low active ratios.

First, we identify the active ratios at which our PSO-based algorithm performed significantly better and faster than the full 100% active ratio. Our PSO-based algorithm performed significantly better and faster than the full 100% active ratio at active ratios of 15%, 30%, 35%, 45%, 55%, 60%, and 75%–95% for certain maximum numbers of iterations.

Over the entire range of population sizes, our PSO-based algorithm performed significantly better and faster than the full 100% active ratio at active ratios denoted by a red check

mark (✓), namely 15%, 30%, 75%, and 80% at certain maximum numbers of iterations (*see* Table 2). With the small population sizes, our PSO-based algorithm performed significantly better and faster than the full 100% active ratio at active ratios of 15%, 30%, 45%, 75%, and 80% for certain maximum numbers of iterations (*see* Table 3). With the large population sizes, our PSO-based algorithm only performed significantly better and faster than the full 100% active ratio at an active ratio of 75% for certain maximum numbers of iterations (*see* Table 4). With a population size of 57, our PSO-based algorithm performed significantly better and faster than the full 100% active ratio at active ratios of 30%, 35%, 60%, 75%, and 85%–95% for certain maximum numbers of iterations. Finally, with a population size of 165, our PSO-based algorithm performed significantly better and faster than the full 100% active ratio at active ratios of 15%, 30%, 45%, 55%, and 80%–95% for certain maximum numbers of iterations. Tables with the population sizes of 57 and 165 are omitted from this study due to page limitation.

We performed the two statistical tests (F-test and t-test) using the simulation results for the various active ratios and maximum numbers of iterations. The results are in conflict in terms of the global best and execution time. Instead of relying on the statistical tests, we used clustering to identify the active ratios at which our PSO-based algorithm performed significantly better and faster than the full 100% active ratio.

We placed a check mark against the active ratios at which our PSO-based algorithm performed better and faster than the full 100% active ratio in the simulation results. We then applied standard k-means clustering to the simulation results. The results of clustering indicate that our PSO-based algorithm achieved equivalent performance to the full 100% active ratio for a number of active ratios (denoted by the square symbol, ☐). We excluded the active ratios with a check mark within a square symbol (☑) from the simulation results. Our PSO-based algorithm performed significantly better and faster than the full 100% active ratio at those active ratios indicated by a red check mark (✓).

In the standard k-means clustering, we classified the global best and execution time at active ratios of 5%, ..., 100% for the entire set of population sizes as well as the small population sizes, large population sizes, and population sizes of 57 and 165. Each matrix of the data object for clustering has twenty rows, corresponding to active ratios from 5% to 100%, and two columns for the global best and execution time at each maximum number of iterations. We iteratively calculated the distance between each data object (here, the global best and execution time) and all cluster centers, and assigned the data object to the closest cluster (in terms of Euclidean distance) at every iteration. We iteratively updated the calculation process and assigned the new means as the centroids of the data objects in the new clusters until the assignments no longer changed.

Active Ratio	Maximum Number of Iterations							
	40	80	120	160	200	240	400	1000
5%-10%								
15%	☐	☐	☐	☐	☐	☐	☐	✓
20%-25%								
30%		☐	☐	☑	☐	☑	☐	✓
35%	☐	☐	☐	☐	☐	☐	☑	☑
40%	☐	☑	☐	☑	☐	☐	☐	☐
45%	☐	☐	☐	☐	☐		☐	☐
50%								
55%	☐	☐	☐	☐	☐	☐	☐	☐
60%	☐	☑	☐	☑	☐	☑	☐	☑
65%	☐	☑	☐	☐	☐	☑	☐	☐
70%	☐	☐	☐	☐	☐	☑	☐	☐
75%	☐	☐	☐	☐	☐	☐	✓	☐
80%	☐	☑	☐	☐	☐	☐	☑	✓
85%	☐	☐	☐	☑	☐	☐	☐	
90%-95%								
100%	☐	☐	☐	☐	☐	☐	☐	☐

Table 2: Active ratios that performed better and faster than the full 100% active ratio across all population sizes (✓).

Active Ratio	Maximum Number of Iterations							
	40	80	120	160	200	240	400	1000
5%-10%								
15%	☐	☐	☐	☐		☐	☐	✓
20%-25%								
30%	☐	☐	☐	☑		☑	☐	✓
35%	☐	☐	☐	☐		☐	☐	☐
40%	☐	☑	☐	☑	☐	☐	☐	☐
45%	☐	✓	☐	☐		☐	☐	☐
50%								
55%	☐	☐	☐	☐		☑	☐	☐
60%	☐	☑	☐	☑	☐	☑	☐	☑
65%	☐	☑	☐	☐	☐	☑	☐	☐
70%	☐	☐	☐	☐		☑	☐	☐
75%	☐	☐	☐	☐		☐	✓	☐
80%	☐	☑	☐	☐		☐	☑	✓
85%	☐	☐	☐	☑	☐	☐	☐	
90%	☐							
95%	☐							
100%	☐	☐	☐	☐	☐	☐	☐	☐

Table 3: Active ratios that performed better and faster than the full 100% active ratio with small population sizes (✓).

Active Ratio	Maximum Number of Iterations							
	40	80	120	160	200	240	400	1000
5%-10%								
15%	☐	☐	☑	☑	☐	☑	☐	
20%-25%								
30%		☐	☐	☑	☐	☑	☐	
35%	☐	☐	☐	☑	☐	☐	☑	☑
40%	☐	☐	☐	☑	☐	☑	☐	☐
45%	☐	☐	☑	☐	☑	☑	☐	☐
50%								
55%	☐	☐	☐	☑	☐	☐	☐	☐
60%	☐	☐	☐	☑	☐	☐	☐	☐
65%	☐	☐	☑	☑	☐	☑	☐	☐
70%	☐	☐	☑	☑	☐	☑	☐	☐
75%	☐	☐	☐	☐	☐	☐	✓	☐
80%	☐	☐	☑	☑	☐	☑	☑	
85%	☐	☐	☑	☑	☐	☐	☐	
90%						☐		
95%						☐		
100%	☐	☐	☐	☐	☐	☐	☐	☐

Table 4: Active ratios that performed better and faster than the full 100% active ratio with large population sizes (✓).

At the active ratios denoted by square symbols, the simulation results of our PSO-based algorithm were ultimately assigned to the cluster that included the simulation results for the full 100% active ratio. We repeated the clustering process using 2–11 clusters, and found that the assignments remained unchanged when six clusters were used. Thus, the number of clusters was set to six.

Next, we identify the active ratios that remained robust and stable over various ranges of maximum numbers of iterations (e.g., 40–80, ..., and 40–1000). Our PSO-based algorithm was robust and stable at the three active ratios of 60%, 80%, and 85%, over certain ranges of maximum numbers of iterations.

For the entire set of population sizes and the small population sizes, the performance of our PSO-based algorithm was neither robust nor stable for any active ratios (*see* Tables 5 and 6). With the large population sizes, our PSO-based algorithm was robust and stable at an active ratio denoted by a red check mark (✓), namely 80% for maximum numbers of iterations from 40 to 1000 (*see* Table 7). With a population size of 57, our PSO-based algorithm was robust and stable at two active ratios, 60% and 85%, over maximum numbers of iterations from 40 to 240 and from 40 to 1000, respectively. With a population size of 165, our PSO-based algorithm was robust and stable at an active ratio of 60% over maximum numbers of iterations from 40 to 80. Tables with the population sizes of 57 and 165 are omitted from this study due to page limitation.

We averaged the rank of the simulation results in terms of the global best and execution time for active ratios of 5%, ..., 100% over ranges of maximum numbers of iterations 40–80, 40–120, 40–160, 40–200, 40–240, 40–400, and 40–1000 for the entire set of population sizes, as well as the small population sizes, large population sizes, and population sizes of 57 and 165. Check marks indicate the active ratios at which our PSO-based algorithm performed better and faster than the full 100% active ratio in the averaged ranking of the simulation results over the various ranges of maximum numbers of iterations. We applied standard k-means clustering to the averaged ranking results. The clustering results show that our PSO-based algorithm was equivalent to the full 100% active ratio at a number of active ratios (denoted by the square symbol, ☐). We excluded active ratios with a check mark within a square symbol (☑) from the averaged ranking results. The performance of our PSO-based algorithm was robust and stable at the active ratios indicated by a red check mark (✓).

In the standard k-means clustering, we classified the global best and execution time at active ratios of 5%, ..., 100% for the entire set of population sizes, as well as the small population sizes, large population sizes, and population sizes of 57 and 165. Each matrix of the data object for clustering has twenty rows, corresponding to active ratios 5%–100%, and 4–16 columns consisting of the global best and execution time at maximum numbers of iterations 40–80, ..., 40–1000. We iteratively calculated the distance between each data object (here, the global best and execution time) and all cluster centers, and assigned the data object to the closest cluster (in terms of the Euclidean distance) at every iteration. We iteratively updated the calculation process and assigned the new means as the centroids of the data objects in the new clusters until the assignments no longer changed.

Active ratios with square symbol indicate that the aver-

Active Ratio	Range of Maximum Numbers of Iterations						
	40-80	-120	-160	-200	-240	-400	-1000
5%-10%							
15%	☐	☐	☐	☐	☐	☐	
20%-25%							
30%	☐	☐	☐	☐	☐	☐	
35%-45%	☐	☐	☐	☐	☐	☐	☐
50%							
55%	☐	☐	☐	☐	☐	☐	☐
60%	☐	☐	☑	☐	☐	☐	☑
65%-75%	☐	☐	☐	☐	☐	☐	☐
80%-85%	☐	☐	☐	☐	☐	☐	
90%-95%							
100%	☐	☐	☐	☐	☐	☐	☐

Table 5: Robustness and stability of active ratios for all population sizes (✓).

Active Ratio	Range of Maximum Numbers of Iterations						
	40-80	-120	-160	-200	-240	-400	-1000
5%-10%							
15%	☐	☐	☐	☐	☐	☐	
20%-25%							
30%	☐	☐	☐	☐	☐	☐	
35%	☐	☐	☐	☐	☐	☐	☐
40%	☑	☐	☐	☐	☐	☐	☐
45%	☑	☐	☐	☐	☐	☐	☐
50%							
55%	☐	☐	☐	☐	☐	☐	☐
60%	☑	☑	☑	☐	☐	☑	☑
65%-75%	☐	☐	☐	☐	☐	☐	☐
80%-85%	☐	☐	☐	☐	☐	☐	
90%-95%	☐	☐	☐	☐	☐		
100%	☐	☐	☐	☐	☐	☐	☐

Table 6: Robustness and stability of active ratios with small population sizes (✓).

Active Ratio	Range of Maximum Numbers of Iterations						
	40-80	-120	-160	-200	-240	-400	-1000
5%-10%							
15%	☐	☐	☐	☐	☐	☐	
20%-25%							
30%	☐	☐	☐	☐	☐	☐	
35%-40%	☐	☐	☐	☐	☐	☐	☐
45%	☐	☐	☐	☐	☐	☑	☐
50%							
55%	☐	☐	☐	☐	☐	☐	☐
60%	☐	☐	☑	☐	☐	☐	☐
65%-75%	☐	☐	☐	☐	☐	☐	☐
80%	☐	☐	☑	☑	☑	☑	✓
85%	☐	☐	☑	☑	☑	☐	
90%-95%							
100%	☐	☐	☐	☐	☐	☐	☐

Table 7: Robustness and stability of active ratios with large population sizes (✓).

aged ranking results were ultimately assigned to the cluster that included the averaged ranking results at the full 100% active ratio (again, the number of clusters is six).

The results presented above are supported by three field studies on ant nest relocation that were carried out by expert ant biologists (Pratt, 2005b; Dornhaus et al., 2009, 2008). These field studies show that the maximum active ratios range 31.0%–58.0% across all population sizes, 31.0%–56.0% for small population sizes, and 52.0%–58.0% for large population sizes.

Active ratios 30%–60% in our simulation results completely cover the range of maximum active ratios observed in the field studies. The maximum of 58.0% is between the active ratios of 55% and 60% in our simulation results, and is slightly closer to the latter value. The performance of our PSO-based algorithm was not only significantly better and faster than the full 100% active ratio, but was also robust and stable at an active ratio of 60%. Thus, based on the empirical support of these three field studies, we have successfully simulated ant nest relocation at low active ratios using our PSO-based algorithm.

We chose the sphere function as the objective in our PSO-based algorithm. With this simple function, we generated various sets of simulation results and identified the benefits and limitations of the algorithm very clearly. Although we have only presented simulation results obtained with the sphere function, we could apply various cost functions to our PSO-based algorithm. We will conduct simulations using other functions in future studies.

Our modeling has been limited to the relation between the active ratio and ant nest relocation, because real-world ant nest relocation is a scale-free swarm behavior that depends not on the population size, but on the frequency of mutual contact among ants, which determines the active ratio.

Conclusion

Ant nest relocation is smoother and swifter than the same process performed by other animals. However, many worker ants are never involved in nest relocation. An open problem concerns whether this low active ratio improves ant nest relocation. In this study, we used an algorithm based on the original PSO and simulated real-world ant nest relocation. Our PSO-based algorithm mimics the real-world ant swarm behavior of active and inactive workers in ant nest relocation. Our algorithm duplicates the velocity and position of an inactive particle, which represents an inactive worker ant, with the velocity and position of an active particle, representing an active worker ant. Clustering of the simulation results of our PSO-based algorithm provides sufficient support to demonstrate that the simulation results are reliable. Our PSO-based algorithm performs significantly better and faster than the full 100% active ratio at active ratios of 15%, 30%, 35%, 45%, 55%, 60%, and 75%–95%. In particular, we have confirmed the robust and stable performance of our PSO-based algorithm at active ratios of 60%, 80%, and 85%. The active ratio of 60% is very close to the maximum active ratio of 58.0% reported in three field studies conducted by expert ant biologists. We have shown that low active ratios significantly improve ant nest relocation in terms of speed and the precision of movement. Our model is limited to the relation between the active ratio and ant nest relocation, because the population size has no impact on the relocation process. The simulations used a simple cost function, though we could apply other functions to the algorithm. In future work, we will conduct simulations using various cost functions.

References

Ali, M. Z., Awad, N. H., and Duwairi, R. M. (2016a). Multi-objective differential evolution algorithm with a new improved mutation strategy. *International Journal of Artificial Intelligence*, 14(2):23–41.

Ali, M. Z., Awad, N. H., Suganthan, P. N., and Reynolds, R. G. (2016b). A modified cultural algorithm with a balanced performance for the differential evolution frameworks. *Knowledge-Based Systems*, 111(1):73–86.

Awad, N. H., Ali, M. Z., and Duwairi, R. M. (2013). Cultural algorithm with improved local search for optimization problems. In *Proceedings of the 15th IEEE Congress on Evolutionary Computation*, pages 284–291. IEEE Press, Piscataway, NJ.

Awad, N. H., Ali, M. Z., and Duwairi, R. M. (2017). Multi-objective differential evolution based on normalization and improved mutation strategy. *Natural Computing*, 16(4):661–675.

Awad, N. H., Ali, M. Z., Suganthan, P. N., and Jaser, E. (2016). A decremental stochastic fractal differential evolution for global numerical optimization. *Information Sciences*, 372(1):470–491.

Burd, M., Archer, D., Aranwela, N., and Stradling, D. J. (2002). Traffic dynamics of the leaf-cutting ant, *Atta cephalotes*. *The American Naturalist*, 159(3):283–293.

Charbonneau, D., Sasaki, T., and Dornhaus, A. (2017). Who needs 'lazy' workers? inactive workers act as a 'reserve' labor force replacing active workers, but inactive workers are not replaced when they are removed. *PLoS ONE*, 12(9):e0184074.

Chowdhury, D., Guttal, V., Nishinari, K., and Schadschneider, A. (2002). A cellular-automata model of flow in ant trails: non-monotonic variation of speed with density. *Journal of Physics A: Mathematical and General*, 35(41):L573–L577.

Cleghorn, C. W. and Engelbrecht, A. P. (2016). Particle swarm optimizer: the impact of unstable particles on performance. In *Proceedings of the 7th IEEE Symposium Series on Computational Intelligence*, pages 1–7. IEEE Press, Piscataway, NJ.

Cleghorn, C. W. and Engelbrecht, A. P. (2018). Particle swarm stability: a theoretical extension using the non-stagnate distribution assumption. *Swarm Intelligence*, 12(1):1–22.

Clerc, M. and Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73.

Dornhaus, A., Holley, J.-A., Pook, V. G., Worswick, G., and Franks, N. R. (2008). Why do not all workers work? colony size and workload during emigrations in the ant *Temnothorax albipennis*. *Behavioral Ecology and Sociobiology*, 63(1):43–51.

Dornhaus, A., Holley, J.-A., and Franks, N. R. (2009). Larger colonies do not have more specialized workers in the ant *Temnothorax albipennis*. *Behavioral Ecology*, 20(5):922–929.

Franks, N. R., Dornhaus, A., Best, C. S., and Jones, E. L. (2006). Decision making by small and large house-hunting ant colonies: one size fits all. *Animal Behaviour*, 72(3):611–616.

Geraghty, M. J., Dunn, R. R., and Sanders, N. J. (2007). Body size, colony size, and range size in ants (*Hymenoptera*: *Formicidae*): are patterns along elevational and latitudinal gradients consistent with Bergmann's rule? *Myrmecological News*, 10:51–58.

John, A., Schadschneider, A., Chowdhury, D., and Nishinari, K. (2008). Characteristics of ant-inspired traffic flow: applying the social insect metaphor to traffic models. *Swarm Intelligence*, 2(1):25–41.

John, A., Schadschneider, A., Chowdhury, D., and Nishinari, K. (2009). Traffic like collective movement of ants on trails: absence of jammed phase. *Physical Review Letters*, 102(10):e108001.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. IN *Proceedings of the 1995 IEEE International Conference on Neural Networks*, pages 1942–1948. IEEE Press, Piscataway, NJ.

Pereira, G. (2010). *Particle Swarm Optimization in Artificial Life by Example*. The codes are publicly available at http://web.ist.utl.pt/gdgp/VA/pso.htm.

Pratt, S. C. (2005a). Quorum sensing by encounter rates in the ant *Temnothorax albipennis*. *Behavioral Ecology*, 16(2):488–496.

Pratt, S. C. (2005b). Behavioral mechanisms of collective nest-site choice by the ant *Temnothorax curvispinosus*. *Insectes Sociaux*, 52(4):383–392.

Sasaki, H. (2017). Modeling time-sensitive swarm dynamics. In *Proceedings of the 8th IEEE Symposium Series on Computational Intelligence*, pages 2489–2496. IEEE Press, Piscataway, NJ.

Sasaki, H. (2018a). A PSO-based algorithm describes ant nest move. In *Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics*, pages 243–248. IEEE Press, Piscataway, NJ.

Sasaki, H. (2018b). Particle swarm optimization describes ant nest move at low participation rate. In *Proceedings of the 9th IEEE Symposium Series on Computational Intelligence*, pages 989–996. IEEE Press, Piscataway, NJ.

Sasaki, H. (2019). Modeling ant nest relocation at low active ratio by particle swarm optimization. In *Proceedings of the 21st IEEE Congress on Evolutionary Computation*. IEEE Press, Piscataway, NJ. (to appear).

Sasaki, H. and Leung, H. (2013). Trail traffic flow prediction by contact frequency among individual ants. *Swarm Intelligence*, 7(4):307–326.

Yarpiz Team (2015). *Particle Swarm Optimization in MATLAB*. In Yarpiz - Academic Source Codes and Tutorials. The codes are publicly available at http://yarpiz.com/50/ypea102-particle-swarm-optimization.