A Novel Bio-inspired Approach Based on the Behavior of Mosquitoes

Xiang Feng [a)], *Francis C.M. Lau* [b)], *Huiqun Yu* [a)]
a) Department of Computer Science and Engineering, East China University of Science and Technology
b) Department of Computer Science, The University of Hong Kong, Hong Kong
Corresponding author. E-mail address: yhq@ecust.edu.cn (Huiqun Yu), xfeng@ecust.edu.cn (Xiang Feng).
Address: 130 Meilong Road, Shanghai 200237, PR China.

Abstract

This paper proposes a new nature-inspired algorithm (NA)—mosquito host-seeking algorithm (MHSA)—the inspiration for which comes from the host-seeking behavior of mosquitoes. Applying the algorithm to the traveling salesman problem (TSP), every city pair is treated as an artificial mosquito, and the TSP solving process is transformed into the host-seeking behavior of a swarm of artificial mosquitoes. We study the evolution of "swarms", the artificial mosquitoes' microcosmic actions, and macroscopic swarm intelligence, and present efficient solutions to TSP using MHSA. The proposed MHSA is fundamentally different from the other popular NAs in its motivation, principle, the optimization mechanism, its elements and their states, and the biological model, the mathematical model and theoretical foundation on which it is based. We show that (1) MHSA can converge; (2) its parameter setting does not depend on algorithm learning or prior knowledge; and (3) MHSA can describe complex behaviors and dynamics. The properties of MHSA, including correctness, convergence and stability, are discussed in details. Simulation results attest to the effectiveness and suitability of MHSA.

Key words: Traveling salesman problem (TSP), mosquito host-seeking algorithm (MHSA), distributed and parallel algorithm.

1. Introduction

Recently in computer science, there is an increased interest in computational approaches that are inspired by the principles of nature and that can solve difficult problems [1,9]. Whereas physical and biological sciences try to find microscopic laws that extrapolates to the macroscopic realm, computer science is principally synthetic and is concerned with the construction of algorithms that may be inspired by physics or biology [2]. Computing can be seen as a property of nature, and therefore by mimicking natural phenomena, it is possible to construct new intelligent algorithms for solving computationally difficult problems [5]. Successful nature-inspired approaches include genetic algorithm [10], ant colony optimization [3,12], particle swarm optimization [21] and cellular particle swarm optimization [13], molecular algorithm [1], artificial life [4], artificial neural networks [8], cellular automaton[13], simulated annealing algorithm [14], elastic net [8], etc.

The goal of this work is to investigate and develop a biology-inspired approach—mosquito host-seeking algorithm (MHSA)—as a new addition in the category of nature-inspired algorithms (NAs). To demonstrate the problem solving abilities of MHSA, we

have applied the approach to the traveling salesman problem (TSP), where every city pair is treated as an artificial mosquito. The TSP solving process is transformed into the kinematics and dynamics of a swarm of artificial mosquitoes. We study the host-seeking behaviors of mosquitoes, the evolution of their "swarms", the mosquitoes' microcosmic actions, and macroscopic swarm intelligence, and finally present efficient solutions to TSP.

The proposed MHSA is motivated by certain observations on the host-seeking behaviors of mosquitoes: that is, the kinematics and dynamics of mosquitoes exhibit the properties of parallelism, openness, local interactivity, and self-organization. This stimulated us to mimic the host-seeking behaviors of mosquitoes in constructing a model for intelligent computing. This new model can overcome some of the limitations of existing popular NAs. MHSA has these advantages: (1) it has the ability to perform large-scale distributed parallel optimization; (2) it can converge; (3) its parameters do not depend on algorithm learning or any prior knowledge; (4) it can describe complex behaviors and dynamics; (5) it has a comprehensive optimization ability for multiple objectives; (5) it is robust—MHSA is basically independent of the initial conditions, problem size, small-range parameters changes, etc; (7) it has a powerful processing ability in a complex and dynamic real-time changing environment; (8) it is flexible and easy to adapt to a wide range of optimization problems. All in all, it is fundamentally different from the other popular NAs in many regards, including the optimization mechanism, its elements and their states, and the biological model, the mathematical model and the theoretical foundation on which it is based.

We apply MSHA to TSP with the motivation that the algorithm coupled with TSP provides an ideal platform for the study of general methods that can by applied to a wide range optimization problems. Recently, using TSP as a platform, some new methods were presented: (1) the Lin-Kernighan-Heulsgaun (LKH) algorithm [11], which is a well known method that can reach optimal results for the TSPLIB benchmarks [17]; (2) the multiagent optimization system (MAOS) algorithm [20], which is a nature-inspired method which supports cooperative search by the self-organization of a group of compact agents situated in an environment with certain shared public knowledge; (3) immune-inspired self-organizing neural network algorithm [15], which is a variation of the Real-Valued Antibody Network algorithm; (4) ant colony optimization with multiple ant clans (ACOMAC) algorithm [19], where the authors introduced the concept of multiple ant clans coming from parallel genetic algorithm to search the solution space; they utilized various islands to avoid local minima and thus the method can yield global minimum for solving TSP.

The structure of the rest of the paper is as follows. In Section 2, we formalize the problem model of TSP as a platform for the study of MHSA. The parallel computing architecture, biological model and mathematical model of MHSA are proposed in Sections 3, 4 and 5, respectively. In Section 6, we discuss the suitability, correctness, convergence, and stability of the MHS model and algorithm. The MHS algorithm is presented in Section 7. In Section 8, we present simulation results which attest to the effectiveness and suitability of MHSA. Finally, conclusions are drawn in Section 9.

2. The TSP model

In this paper, TSP is regarded as a platform for presenting MHSA, and so the TSP mathematical description given here is somewhat different from many of the other ones in the literature. The main feature of our TSP model is that the $n-$city TSP is decomposed

into $n \times n$ paratactic computing cells. We discuss how MHSA solves our TSP using its particular parallel computing architecture in the next section.

Definition 1. *The n-city TSP is defined as:*

$$\textit{Minimize: } Z = \sum_{i,j} d_{ij} \cdot r_{ij}$$

$$\begin{aligned} \textit{s.t.} \quad & 1.\ r_{ij} \in \{0,1\}; \\ & 2.\ r_{ij} = r_{ji}; \\ & 3.\ \sum_{i} r_{ij} = 2; \\ & 4.\ \sum_{j} r_{ij} = 2. \\ & (\ i = \overline{1,n}\ ) \end{aligned}$$

The main parameters of any TSP instance are as follows.

C_i**:** The i-th city ($i = \overline{1,n}$).

C_j**:** The j-th city ($j = \overline{1,n}$).

(x_i, y_i)**:** The coordinates of city C_i.

d_{ij}**:** The distance between city pair (C_i, C_j), $(i, j = \overline{1,n})$, $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

p_{ij}**:** The path between C_i and C_j.

x_{ij}**:** $x_{ij} = 1$ if p_{ij} can be passed through; otherwise 0.

Z**:** The shortest path through n cities.

r_{ij}**:** The solution of TSP: (1) When $t = end$, $r_{ij} = 0, 1$ ($r_{ij} = 1$ if Z passes p_{ij}; otherwise 0);
(2)When $t < end$, $r_{ij} \in [0, 1]$.

r_{ij} is the main variable in the computing cells. With our MHSA, $n \times n$ r_{ij} will evolve in parallel until the algorithm converges ($t = end$). In order to make the solution (r_{ij}) progress towards the optimal result, we introduce an artificial variable c_{ij} into the computing cells, which is the weight of city pair (C_i, C_j), as follows.

- When $t = 0$, $c_{ij}(t = 0) = \max_{i,j} d_{ij} - d_{ij}$;
- When $t > 0$, $c_{ij}(t > 0) \in [0, 1]$.

In every computing cell, r_{ij} and c_{ij} will evolve at the same time together. In addition, the state of the computing cells depends on logical variable x_{ij}. Computing cell $\mathfrak{C}_{ij}$ is "alive" if $x_{ij} = 1$; otherwise "dead". The $n \times n$ computing cells of n-city TSP are shown in Fig. 1.

r_{11}, c_{11}	...	r_{1j}, c_{1j}	...	r_{1n}, c_{1n}
⋮		⋮		⋮
r_{i1}, c_{i1}	...	r_{ij}, c_{ij}	...	r_{in}, c_{in}
⋮		⋮		⋮
r_{n1}, c_{n1}	...	r_{nj}, c_{nj}	...	r_{nn}, c_{nn}

Fig. 1 The $n \times n$ computing cells of n-city TSP

3. The parallel computing architecture of MHSA

The parallel computing architecture of the MHSA, as shown in Fig. 2, is composed of four computing cell arrays, $\mathfrak{C}, \mathfrak{C}_{row}, \mathfrak{C}_{col}$, and $\mathfrak{C}_{globle}$, whose computing cells are denoted by $\mathfrak{C}_{ij}, \mathfrak{C}_{i*}, \mathfrak{C}_{*j}$, and $\mathfrak{C}_{**}$, respectively.

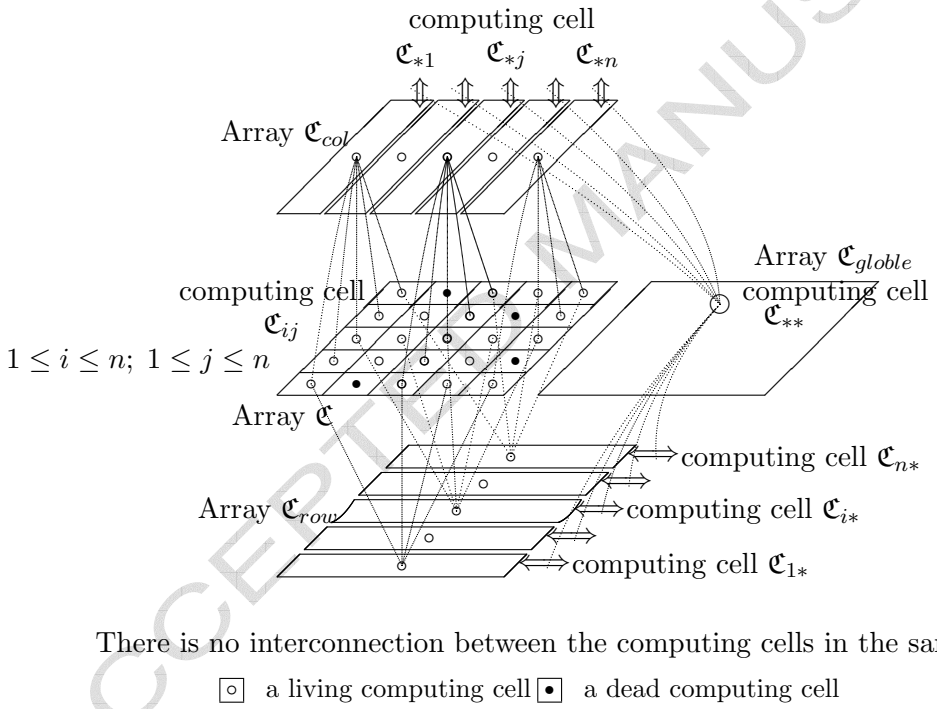

Fig.2 The parallel computing architecture of the MHSA

The number of computing cells in each array is equal to: $n \times n$ for $\mathfrak{C}$, n for $\mathfrak{C}_{row}$, n for $\mathfrak{C}_{col}$, and 1 for $\mathfrak{C}_{globle}$, respectively, and hence the total number of computing cells equals n^2+2n+1. There is no interconnection among computing cells in the same array, whereas there are local interconnections between the following computing cell pairs: $\mathfrak{C}_{ij}$ and $\mathfrak{C}_{i*}$; $\mathfrak{C}_{ij}$ and $\mathfrak{C}_{*j}$; $\mathfrak{C}_{i*}$ and $\mathfrak{C}_{**}$; $\mathfrak{C}_{*j}$ and $\mathfrak{C}_{**}$. It is obvious that the connection degree of each computing cell in the array $\mathfrak{C}$ of $n \times n$ computing cells is equal to at most 2, and the unique computing cell in $\mathfrak{C}_{globle}$ has connection degree $n+n$, with the total number of interconnections being $2n^2+2n$.

At time t in a fixed time slot ϱ , the computing cell $\mathfrak{C}_{ij}$ sends its dynamical state $q_{ij}(t)\langle r_{ij}(t), c_{ij}(t)\rangle$ to computing cells $\mathfrak{C}_{i*}$ and $\mathfrak{C}_{*j}$, and receives the feedback inputs that

are generated by computing cells $\mathfrak{C}_{i*}$ and $\mathfrak{C}_{*j}$ at time $(t-\tau)$. By using the received $q_{ij}(t)$, the computing cell $\mathcal{C}_{i*}$ ($\mathfrak{C}_{*j}$, resp.) obtains its calculation state $r_{i*}(t)$ ($r_{*j}(t)$, resp.) according to the equation $r_{i*}(t) = \sum_j r_{ij}(t)$ ($r_{*j}(t) = \sum_i r_{ij}(t)$, resp.); which yields its current output to be fed back to the computing cell $\mathfrak{C}_{ij}$. Meanwhile, the computing cell $\mathfrak{C}_{i*}$ and $\mathfrak{C}_{*j}$ receive the feedback from computing cell $\mathfrak{C}_{**}$. The computing cells, $\mathfrak{C}_{i*}$, $\mathfrak{C}_{*j}$, and $\mathfrak{C}_{**}$, will change their calculation states respectively. The computing cell $\mathfrak{C}_{ij}$ will change its dynamical state according to Eqs. 6 and 7 (to be given in Section 5), respectively.

The implementation of MHSA can enjoy a high degree of parallelism and good scalability. All the computations of cellular dynamics both in the same array and in different arrays are concurrently carried out. The cellular structure, the cellular dynamics and the algorithm are all independent of the problem scale. Moreover, there is no direct interconnection among computing cells in the same array, and so it is relatively easy to implement the proposed structure in VLSI technology.

4. The biological model of MHSA

In this model, every computing cell is treated as an artificial mosquito m_{ij}, which transforms the $n-$city TSP solving process into the host seeking behavior of a swarm of $n \times n$ artificial mosquitoes. As shown in Fig. 3, all artificial mosquitoes (in a swarm) are evenly distributed at an even radian surrounding a host. Each female artificial mosquito, corresponding to a living computing cell, is attracted to seek towards the host by carbon dioxide, odours, and radiated heat. The radial distance between an artificial mosquito and the host represents the corresponding artificial mosquito's personal utility—that is, an artificial mosquito's success value of host-seeking. The higher the concentration of carbon dioxide and odours, the faster the artificial mosquitoes try to move toward their host. When all the artificial mosquitoes stop moving, being in an equilibrium state, the computational process arrives at an optimum solution of TSP.

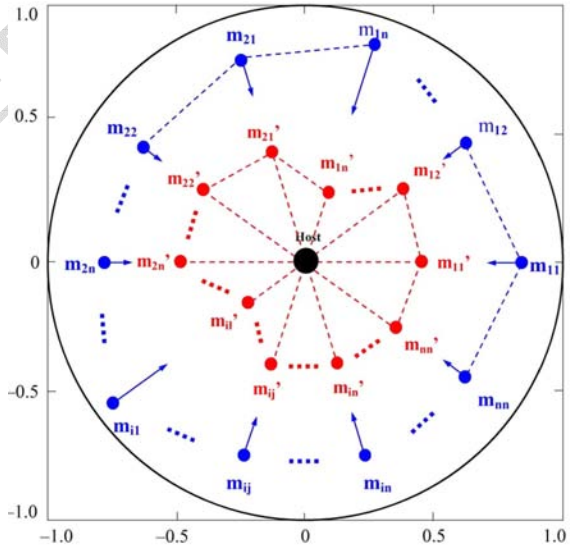

Fig.3 The biological model of MHSA

In more details, each artificial mosquito m_{ij} has a grayscale value r_{ij}, which will constantly change between 0 and 1 as the artificial mosquito moves. When all mosquitoes

are in an equilibrium state, all r_{ij} will be 1 or 0. $r_{ij} = 1$ represents the artificial mosquito m_{ij} attacking the host, as well as represents the path p_{ij} being black (the shortest path Z passes through the path). On the contrary, $r_{ij} = 0$ represents that m_{ij} would not attack the host, or p_{ij} being white (Z does not pass through the path).

In addition, each artificial mosquito m_{ij} has a sex attribute x_{ij}. m_{ij} is female if $x_{ij} = 1$; otherwise, m_{ij} is male if $x_{ij} = 0$. x_{ij} does not evolve with time. The existence of the x variable enables the MHS mathematical model to solve those TSPs where there is no through route between some cities. If the route (p_{ij}) between city i and city j is not a through road, then x_{ij} =0; otherwise x_{ij} =1. Only grayscale values r_{ij} of female artificial mosquitoes will change with the motion of m_{ij}. r_{ij} of male artificial mosquitoes will always be 0—that is, the corresponding computing cell will always be dead and path p_{ij} will always be white.

5. The mathematical model of MHSA

The mathematical model of MHSA for n-city TSP is defined as follows.

Definition 2. *Let $u_{ij}(t)$ be the radial distance between an artificial mosquito m_{ij} and the host at time t, and let $J(t)$ be the utility sum of all artificial mosquitoes. We define $u_{ij}(t)$ and $J(t)$, respectively, by*

$$u_{ij}(t) = \exp\left(-c_{ij}(t) r_{ij}(t) x_{ij}(t) \right); \tag{1}$$

$$J(t) = \sum_{i=1}^{n} \sum_{j=1}^{n} u_{ij}(t) \tag{2}$$

The smaller $u_{ij}(t)$ is, the closer the artificial mosquitoes m_{ij} are to the host, and the larger the utility of m_{ij}.

Definition 3. *At time t, the attraction function, $P(t)$, which is caused by the host is defined by*

$$P(t) = \epsilon^2 \ln \sum_{i=1}^{n} \sum_{j=1}^{n} \exp[-u_{ij}^2(t)/2\epsilon^2] - \epsilon^2 \ln n^2 \tag{3}$$

where $0 < \epsilon < 1$. The smaller $P(t)$ the better. With Eq. (3), we attempt to construct a potential energy function, $P(t)$, such that the decrease of its value would imply the increase of the minimal utility of all the artificial mosquitoes. We prove it in Proposition 3. This way we can solve the optimization problem in the sense that we consider not only the aggregate utility, but also the individual personal utilities, especially the minimum one. In addition, ϵ represents the strength of a host's attraction. The bigger ϵ the better. If we did not get a sufficiently satisfactory result with MHSA, we can make ϵ smaller. The attraction of the host causes the artificial mosquitoes to move to increase their minimal personal utility.

Definition 4. *At time t, the artificial mosquitoes' interaction behavior function, $Q(t)$, is defined by*

$$Q(t) = \sum_{i=1}^{n} \mid \sum_{j=1}^{n} r_{ij}(t) x_{ij}(t) - 2 \mid^2 - \sum_{i,\, j} \int_0^{u_{ij}} \{[1 + \exp(-\zeta_{ij} x)]^{-1} - 0.5\} dx. \tag{4}$$

$\zeta_{ij}(t)$ of artificial mosquito $m_{ij}(t)$ represents the aggregate intention strength at time t. The greater $\zeta_{ij}(t)$ is, the more necessary artificial mosquito $m_{ij}(t)$ has to modify its $r_{ij}(t)$. $\zeta_{ij}(t)$ is usually within [-15,15]. For TSP in particular, $\zeta_{ij}(t)$ can be set to 10.

The first term of $Q(t)$ is related to the constraints on TSP; the second term involves social coordinations among the artificial mosquitoes. The first term of $Q(t)$ corresponds to a penalty function. The second term of $Q(t)$ is chosen as shown because we want $\frac{\partial Q}{\partial u_{ij}}$ to be a monotone decreasing sigmoid function, as shown in Fig. 4. $-\{[1+\exp(-\zeta_{ij}u_{ij})]^{-1}-0.5\}$ is such a function. Therefore we let $\frac{\partial Q}{\partial u_{ij}}$ equal to $-\{[1+\exp(-\zeta_{ij}u_{ij})]^{-1}-0.5\}$. Then $\frac{\partial Q}{\partial u_{ik}}$ is integrated to be Q.

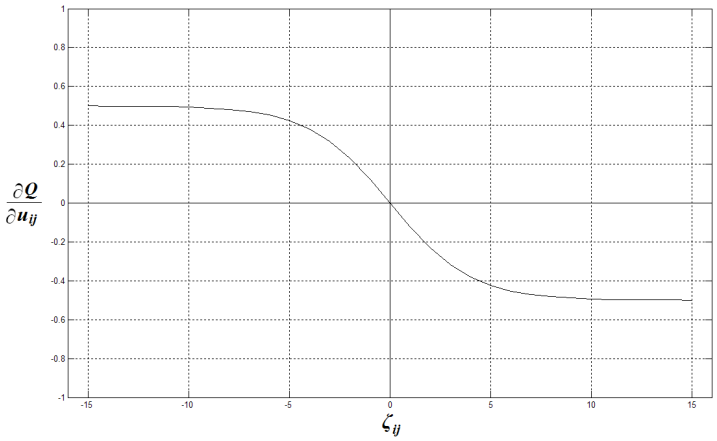

Fig. 4 Graphical presentation of $\frac{\partial Q}{\partial u_{ij}}$.

Artificial mosquitoes can move toward the host along their own radial orbit under the the influence of these factors:

- the personal host-seeking behavior;
- the aggregate host-seeking behavior;
- the attraction of the host;
- the motion that is related to social coordinations in the swarm of artificial mosquitoes.

These four factors can all contribute to the artificial mosquitoes' movements towards the host. Moreover, these factors produce hybrid attraction forces.

Definition 5. *The general hybrid attraction function for artificial mosquito m_{ij}, $E_{ij}(t)$, can be defined by*

$$E_{ij}(t) = -\lambda_1 u_{ij}(t) - \lambda_2 J(t) - \lambda_3 P(t) - \lambda_4 Q(t) \tag{5}$$

where $0 < \lambda_1, \lambda_2, \lambda_3, \lambda_4 < 1$.

Definition 6. *The dynamic equations of computing cell $\mathfrak{C}_{ij}$ for solution variable $r_{ij}(t)$ and weight variable $c_{ij}(t)$ are defined, respectively, as follows.*

$$dr_{ij}(t)/dt = -\lambda_1 \frac{\partial u_{ij}(t)}{\partial r_{ij}(t)} - \lambda_2 \frac{\partial J(t)}{\partial r_{ij}(t)} - \lambda_3 \frac{\partial P(t)}{\partial r_{ij}(t)} - \lambda_4 \frac{\partial Q(t)}{\partial r_{ij}(t)} \tag{6}$$

$$dc_{ij}(t)/dt = -\lambda_1 \frac{\partial u_{ij}(t)}{\partial c_{ij}(t)} - \lambda_2 \frac{\partial J(t)}{\partial c_{ij}(t)} - \lambda_3 \frac{\partial P(t)}{\partial c_{ij}(t)} - \lambda_4 \frac{\partial Q(t)}{\partial c_{ij}(t)} \tag{7}$$

where $\frac{\partial Q(t)}{\partial u_{ij}(t)} = -\{[1+\exp(-10(t)u_{ij}(t))]^{-1} - 0.5\}$.

Definition 7. *Motion equations for artificial mosquito m_{ij} are defined by*

$$\begin{cases} du_{ij}(t)/dt = \Psi_1(t) + \Psi_2(t) \\ \Psi_1(t) = -u_{ij}(t) + \gamma v_{ij}(t) \\ \Psi_2(t) = [-\lambda_1 - \lambda_2\frac{\partial J(t)}{\partial u_{ij}(t)} - \lambda_3\frac{\partial P(t)}{\partial u_{ij}(t)} - \lambda_4\frac{\partial Q(t)}{\partial u_{ij}(t)}]\{[\frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}]^2 + [\frac{\partial u_{ij}(t)}{\partial c_{ij}(t)}]^2\} \end{cases} \tag{8}$$

where $\gamma > 1$. And $v_{ij}(t)$ is a piecewise linear function of $u_{ij}(t)$ defined by

$$v_{ij}(t) = \begin{cases} 0 & if \;\; u_{ij}(t) < 0 \\ u_{ij}(t) & if \;\; 0 \le u_{ij}(t) \le 1 \\ 1 & if \;\; u_{ij}(t) > 1, \end{cases} \tag{9}$$

The definitions of Eqs. (8) and (9) are for the convergence proofs of MHSA (see Section 6.4).

6. The theoretical foundation of MHSA

In this section, we discuss the suitability, correctness, convergence, and stability of the MHS model and algorithm. Subsection A elucidates the correctness of MHSA. Subsections B, C and D show that all the artificial mosquitoes converge to their stable equilibrium states through MHSA.

6.1. Correctness analysis

In the following, we derive some formal properties of the mathematical model presented above.

Proposition 1. *Updating the grayscale values r_{ij} and weights c_{ij} by Eq. (6) and Eq. (7) respectively amounts to changing the speed of artificial mosquito m_{ij} by $\Psi_2(t)$ of Eq. (8).*

Denote the j-th terms of Eq. (6) and Eq. (7) by $\langle\ \frac{dr_{ij}(t)}{dt}\ \rangle_j$ and $\langle\ \frac{dc_{ij}(t)}{dt}\ \rangle_j$, respectively. When r_{ij} is updated according to (6), the first and second terms of (6) will give rise to the following speed increments of the artificial mosquito m_{ij}, respectively:

$$\langle du_{ij}(t)/dt\rangle_1^r = \frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}\langle\frac{dr_{ij}(t)}{dt}\rangle_1 = -\lambda_1[\frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}]^2; \tag{10}$$

$$\langle du_{ij}(t)/dt\rangle_2^r = \frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}\langle\frac{dr_{ij}(t)}{dt}\rangle_2 = -\lambda_2\frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}\frac{\partial J(t)}{\partial r_{ij}(t)} = -\lambda_2\frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}\frac{\partial J(t)}{\partial u_{ij}(t)}\frac{\partial u_{ij}(t)}{\partial r_{ij}(t)} = -\lambda_2\frac{\partial J(t)}{\partial u_{ij}(t)}[\frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}]^2 \tag{11}$$

Similarly, the third and the fourth term of Eq. (6) will give rise to the following speed increments of the artificial mosquito m_{ij}:

$$\langle\ du_{ij}(t)/dt\ \rangle_3^r\ = -\lambda_3\frac{\partial P(t)}{\partial u_{ij}(t)}[\frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}]^2\ ;$$

$$\langle\ du_{ij}(t)/dt\ \rangle_4^r\ = -\lambda_4\frac{\partial Q(t)}{\partial u_{ij}(t)}[\frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}]^2.$$

Similarly, for Eq. (7), we have $\langle du_{ij}(t)/dt\rangle_j^c$, $j = 1, 2, 3, 4$. We thus obtain

$$\sum_{j=1}^{4}\ [\ \langle\ du_{ij}(t)/dt\ \rangle_j^c + \langle\ du_{ij}(t)/dt\ \rangle_j^r\ ]$$

$$= [-\lambda_1 - \lambda_2\frac{\partial J(t)}{\partial u_{ij}(t)} - \lambda_3\frac{\partial Pt)}{\partial u_{ij}(t)} - \lambda_4\frac{\partial Q(t)}{\partial u_{ij}(t)}]\{[\frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}]^2 + [\frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}]^2\} = \Psi_2(t).$$

Therefore, updating $r_{ij}^{(j)}$ and $c_{ij}^{(j)}$ by (6) and (7), respectively, gives rise to the speed increment of artificial mosquito m_{ij} which is exactly equal to $\Psi_2(t)$ of Eq. (8).

Proposition 2. *The first and second term of Eqs.* (6) *and* (7) *will cause the artificial mosquito* m_{ij} *to move towards the host; that is, the personal utility of the artificial mosquito* m_{ij} *increases, in direct proportion to the value of* $(\lambda_1 + \lambda_2)$.

According to Eqs. (10) and (11), the sum of the first and second term of Eqs. (6) and (7) will be

$$\begin{aligned}
&\langle du_{ij}(t)/dt\rangle_1^r + \langle du_{ij}(t)/dt\rangle_2^r + \langle du_{ij}(t)/dt\rangle_1^c + \langle du_{ij}(t)/dt\rangle_2^c \\
&\quad = [-\lambda_1 - \lambda_2 \tfrac{\partial J(t)}{\partial u_{ij}(t)}]\{[\tfrac{\partial u_{ij}(t)}{\partial r_{ij}(t)}]^2 + [\tfrac{\partial u_{ij}(t)}{\partial c_{ij}(t)}]^2\} \\
&\quad = -(\lambda_1 + \lambda_2)x_{ij}^2(t)[r_{ij}^2(t) + c_{ij}^2(t)][u_{ij}(t)]^2 \\
&\quad \leq 0.
\end{aligned}$$

Therefore, the first and second term of (6) and (7) will cause $u_{ij}(t)$ to monotonically decrease.

Proposition 3. *For MHSA, if* ϵ *is very small, then decreasing the attraction forces* $P(t)$ *of the host (Eq.* (3)*) amounts to increasing the minimal utility of the artificial mosquitoes.*

Supposing that $H(t) = \max\limits_{i,j}\{u_{ij}^2(t)\}$, we have

$$[\exp(H(t)/2\epsilon^2)]^{2\epsilon^2} \leq [\sum_{i=1}^{n}\sum_{j=1}^{n}\exp(u_{ij}^2(t)/2\epsilon^2)]^{2\epsilon^2} \leq [nn\exp(H(t)/2\epsilon^2)]^{2\epsilon^2}.$$

Taking the logarithm of both sides of the above inequalities gives

$$H(t) \leq 2\epsilon^2 \ln\sum_{i=1}^{n}\sum_{j=1}^{n}\exp(u_{ij}^2(t)/2\epsilon^2) \leq H(t) + 2\epsilon^2\ln n^2.$$

Since n^2 is constant and ϵ is very small, we have

$$H(t) \approx 2\epsilon^2 \ln\sum_{i=1}^{n}\sum_{j=1}^{n}\exp(u_{ij}^2(t)/2\epsilon^2) - 2\epsilon^2\ln n^2 = 2P(t).$$

It turns out that the attraction forces $P(t)$ at time t represent the maximum of $u_{ij}^2(t)$ among all the artificial mosquitoes m_{ij}, which is the minimal personal utility of the artificial mosquitoes at time t. Hence the decrease of attraction function $P(t)$ will result in the decrease of the maximum of $u_{ij}(t)$.

Proposition 4. *Updating* r_{ij} *and* c_{ij} *according to Eqs.* (6) *and* (7) *amounts to increasing the minimal utility of artificial mosquitoes in direct proportion to the value of* λ_3.

The speed increment of artificial mosquito m_{ij}, which is related to attraction function $P(t)$, is given by

$$\begin{aligned}
\langle\, \tfrac{du_{ij}(t)}{dt}\,\rangle_3 &= \langle\, du_{ij}(t)/dt\,\rangle_3^r \;+\; \langle\, du_{ij}(t)/dt\,\rangle_3^c \\
&= -\lambda_3 \tfrac{\partial P(t)}{\partial u_{ij}(t)}\{[\tfrac{\partial u_{ij}(t)}{\partial r_{ij}(t)}]^2 + [\tfrac{\partial u_{ij}(t)}{\partial c_{ij}(t)}]^2\}.
\end{aligned}$$

Denote by $\langle\frac{dP(t)}{dt}\rangle$ the differentiation of the attraction function $P(t)$ with respect to time t arising from using Eqs. (6), (7). We then have

$$\begin{aligned}
\langle\tfrac{dP(t)}{dt}\rangle &= \tfrac{\partial P(t)}{\partial u_{ij}(t)}\langle\tfrac{du_{ij}(t)}{dt}\rangle_3 \\
&= -\lambda_3[\tfrac{\partial P(t)}{\partial u_{ij}(t)}]^2\{[\tfrac{\partial u_{ij}(t)}{\partial r_{ij}(t)}]^2 + [\tfrac{\partial u_{ij}(t)}{\partial c_{ij}(t)}]^2\} \\
&= -\lambda_3\omega_{ij}^2(t)u_{ij}^2(t)x_{ij}^2(t)[r_{ij}^2(t) + c_{ij}^2(t)][u_{ij}(t)]^2 \\
&\leq 0
\end{aligned}$$

where $\omega_{ij}(t) = \exp[-u_{ij}^2(t)/2\epsilon^2] / \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}\exp[-u_{ij}^2(t)/2\epsilon^2]$.

It can be seen that using Eqs. (6) and (7) gives rise to a monotonic decrease of $P(t)$. Then by Proposition 3, the decrease of $P(t)$ will result in the increase of the minimal utility, in direct proportion to the value of λ_3.

Proposition 5. *Updating r_{ij} and c_{ij} by Eqs.* (6) *and* (7) *gives rise to a monotonic increase of the whole utility of all the artificial mosquitoes, in direct proportion to the value of λ_2.*

Similar to Proposition 2, it follows that when an artificial mosquito m_{ij} modifies its r_{ij} and c_{ij} by Eqs. (6) and (7), differentiation of $J(t)$ with respect to time t will not be positive—i.e., $\langle \frac{dJ_R(t)}{dt} \rangle \leq 0$, and it is directly proportional to the value of λ_2.

Proposition 6. *Updating r_{ij} and c_{ij} by Eqs.* (6) *and* (7) *gives rise to a monotonic decrease of the artificial mosquitoes' behavior interaction function $Q(t)$, in direct proportion to the value of λ_4.*

As in the above, we have

$$\langle \frac{du_{ij}(t)}{dt} \rangle_4 = -\lambda_4 \frac{\partial Q(t)}{\partial u_{ij}(t)} \{ [\frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}]^2 + [\frac{\partial u_{ij}(t)}{\partial c_{ij}(t)}]^2 \}; \text{ and}$$

$$\begin{aligned} \langle \frac{dQ(t)}{dt} \rangle &= \frac{\partial Q(t)}{\partial u_{ij}(t)} \langle \frac{du_{ij}(t)}{dt} \rangle_4 \\ &= -\lambda_4 [\frac{\partial Q(t)}{\partial u_{ij}(t)}]^2 \{ [\frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}]^2 + [\frac{\partial u_{ij}(t)}{\partial c_{ij}(t)}]^2 \} \\ &\leq 0. \end{aligned}$$

6.2. Convergence analysis

In this subsection, we show that all artificial mosquitoes can converge to their stable equilibrium states through MHSA.

In mathematics, *stability theory* deals with the stability of the solutions of differential equations and dynamical systems. Definitions of stability include *Lyapunov stability* and *structural stability*. Lyapunov stability occurs in the study of dynamical systems. *Lyapunov functions* are a family of functions that can be used to demonstrate the stability or instability of some state points of a system. The demonstration of stability or instability requires finding a Lyapunov function for the given dynamical system.

Most of dynamical systems are energy-cost systems. In this kind of systems, the total energy will decrease with the elapsing time until the total energy reaches the least storage. Therefore, the measurement in energy can become the measurement of stability in dynamical systems. In this paper, all related factors that affect the evolution of artificial mosquitoes are treated as corresponding items of a "hybrid attraction function ($E_{ij}(t)$)" which is similar to "energy function". In this subsection, we construct a Lyapunov function based on a hybrid attraction function defined in Eq. (5). This Lyapunov function is an attraction-related positive definite function. We can judge the stability of the given problem by analyzing if the Lyapunov function monotonically decreases with the elapsing time.

Lyapunov second theorem on stability *Consider a function $L(X)$ such that*

- $L(X) > 0$ *(positive definite);*
- $dL(X(t))/dt < 0$ *(negative definite).*

Then $L(X(t))$ is called a Lyapunov function candidate and X is asymptotically stable in the sense of Lyapunov.

It is easier to visualize this method of analysis by imagining a swarm of mosquitoes and a host. If the host disappears along with its attraction and is never restored, eventually the swarm of mosquitoes would come to a stop at some final resting state. This final state is called the stable equilibrium state. For the simple second theorem on stability, a good selection for a Lyapunov function is the hybrid attraction function. Of course, some changes of the attraction function are necessary.

Theorem 1. *If the condition (14) about the parameters remain valid, then MHSA will converge to a stable equilibrium state.*

Proof. Denote $(r_{ij})_{n\times n}(t)$ and $(c_{ij})_{n\times n}(t)$ by $R(t)$ and $C(t)$ respectively. For the biological model of MHSA, we define a Lyapunov function $L(R(t))$

$L(R(t)) \triangleq R(t)$

$R(t)$ is updated according to Eq. (6).

$R(t+1) = R(t) + \Delta R(t+1)$;

$\Delta r_{ij}(t+1) \approx \frac{dr_{ij}(t)}{dt} = -\lambda_1 \frac{\partial u_{ij}(t)}{\partial r_{ij}(t)} - \lambda_2 \frac{\partial J(t)}{\partial r_{ij}(t)} - \lambda_3 \frac{\partial P(t)}{\partial r_{ij}(t)} - \lambda_4 \frac{\partial Q(t)}{\partial r_{ij}(t)}$.

Whatever r_{ij} is initialized or updated to be, the matrix R is dealt with using the following three steps.

- $r_{ij} = 0 \mid \forall x_{ij} = 0$. If $x_{ij} = 0$, then $r_{ij} = 0$, to ensure that a male artificial mosquito m_{ij} (entry (i,j) of R) which is not related is not updated.
- Nonnegativity: $0 \le r_{ij} \le 1$. If $\min\limits_{i,j} r_{ij} < 0$, then let $r_{ij} = r_{ij} - \min\limits_{i,j} r_{ij}$.
- Normalization. Let $r_{ij} = 2\ r_{ij} / \sum\limits_{j=1}^{n} r_{ij}$ and $r_{ij} = 2\ r_{ij} / \sum\limits_{i=1}^{n} r_{ij}$, in order to satisfy the constraint of the problem defined in Definition 1; that is, $\sum\limits_{j=1}^{n} r_{ij} = 2$ and $\sum\limits_{i=1}^{n} r_{ij} = 2$, $i, j = 1, 2, \cdots, n$.

After c_{ij} is updated according to Eq. (7), the matrix C should be dealt with using the three steps just mentioned again.

By using the three steps, the entries of matrices R and C will evolve in the closed unit 1-dimensional space.

Because $0 \le r_{ij} \le 1$, then $R(t) > 0$. So $L(R(t)) > 0$.

$dL(R(t))/dt = \sum\limits_{i,j} dr_{ij}(t)/dt$,

where

$$dr_{ij}(t)/dt = \frac{dr_{ij}(t)}{du_{ij}(t)} \cdot \frac{du_{ij}(t)}{dt} = \frac{1}{\frac{du_{ij}(t)}{dr_{ij}(t)}} \cdot \frac{du_{ij}(t)}{dt}$$
$$= \frac{-1}{c_{ij}(t)x_{ij}(t)\exp[-c_{ij}(t)r_{ij}(t)x_{ij}(t)]} \cdot \frac{du_{ij}(t)}{dt},$$

where

$$du_{ij}(t)/dt = \frac{\partial u_{ij}(t)}{\partial r_{ij}(t)} \frac{dr_{ij}(t)}{dt} + \frac{\partial u_{ij}(t)}{\partial c_{ij}(t)} \frac{dc_{ij}(t)}{dt}$$
$$= [-\lambda_1 - \lambda_2 \frac{\partial J(t)}{\partial u_{ij}(t)} - \lambda_3 \frac{\partial P(t)}{\partial u_{ij}(t)} - \lambda_4 \frac{\partial Q(t)}{\partial u_{ij}(t)}]\{[\frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}]^2 + [\frac{\partial u_{ij}(t)}{\partial c_{ij}(t)}]^2\},$$
$$dr_{ij}(t)/dt = \frac{1}{c_{ij}(t)x_{ij}(t)\exp[-c_{ij}(t)r_{ij}(t)x_{ij}(t)]} \cdot [\lambda_1 + \lambda_2 \frac{\partial J(t)}{\partial u_{ij}(t)} + \lambda_3 \frac{\partial P(t)}{\partial u_{ij}(t)} + \lambda_4 \frac{\partial Q(t)}{\partial u_{ij}(t)}]\{[\frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}]^2 + [\frac{\partial u_{ij}(t)}{\partial c_{ij}(t)}]^2\}.$$

$$\therefore\ dL(R(t))/dt = \sum_{i,j} \frac{\{[\frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}]^2 + [\frac{\partial u_{ij}(t)}{\partial c_{ij}(t)}]^2\}}{c_{ij}(t)x_{ij}(t)\exp[-c_{ij}(t)r_{ij}(t)x_{ij}(t)]} \cdot [\lambda_1 + \lambda_2 \frac{\partial J(t)}{\partial u_{ij}(t)} + \lambda_3 \frac{\partial P(t)}{\partial u_{ij}(t)} - \lambda_4 \frac{\partial Q(t)}{\partial u_{ij}(t)}]. \quad (12)$$

If all the items of the summation in Eq. (12) are less than 0, then $dL(R(t))/dt < 0$; that is,

$$\frac{\{[\frac{\partial u_{ij}(t)}{\partial r_{ij}(t)}]^2 + [\frac{\partial u_{ij}(t)}{\partial c_{ij}(t)}]^2\}}{c_{ij}(t)x_{ij}(t)\exp[-c_{ij}(t)r_{ij}(t)x_{ij}(t)]} \cdot [\lambda_1 + \lambda_2 \frac{\partial J(t)}{\partial u_{ij}(t)} + \lambda_3 \frac{\partial P(t)}{\partial u_{ij}(t)} + \lambda_4 \frac{\partial Q(t)}{\partial u_{ij}(t)}] < 0 \quad (13)$$

Except for the entries that $x_{ij}(t) = 0, r_{ij}(t) = 0, c_{ij}(t) = 0$, we always have $0 < r_{ij}(t), c_{ij}(t) \le 1$. So we have $c_{ij}(t)r_{ij}(t)x_{ij}(t) > 0$ and $c_{ij}(t)x_{ij}(t) > 0$. Then

$\exp[-c_{ij}(t)r_{ij}(t)x_{ij}(t)] > 0$. We obtain $\frac{1}{c_{ij}(t)x_{ij}(t)\exp[-c_{ij}(t)r_{ij}(t)x_{ij}(t)]} > 0$.

Thus to make Eq. (13) exist, the second item $\lambda_1 + \lambda_2\frac{\partial J(t)}{\partial u_{ij}(t)} + \lambda_3\frac{\partial P(t)}{\partial u_{ij}(t)} + \lambda_4\frac{\partial Q(t)}{\partial u_{ij}(t)}$ must be

$$\lambda_1 + \lambda_2\frac{\partial J(t)}{\partial u_{ij}(t)} + \lambda_3\frac{\partial P(t)}{\partial u_{ij}(t)} + \lambda_4\frac{\partial Q(t)}{\partial u_{ij}(t)} < 0 \tag{14}$$

Based on Lyapunov second theorem on stability, as long as we properly select the parameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4$, according to Eq. (14), then convergence and stability can be guaranteed. That is, when $t \to \infty$, $R(t) \to R$. Eq. (14) will be analyzed further in the next subsection. □

6.3. Parameters analysis

There are only five coefficients in MHSA. They are ϵ, λ_1, λ_2, λ_3, and λ_4. We analyze them here and point out what values they should take on.

Firstly, as discussed above, ϵ represents the strength of the attraction forces of the host. The larger ϵ is, the faster the artificial mosquitoes would move towards the host; hence, ϵ influences the convergence speed of the TSP process. ϵ needs to be carefully adjusted in order to obtain the shortest path Z. As required in Proposition 3, ϵ must be small. Usually, $0 < \epsilon < 1$.

Theorem 2. *Regarding the coefficients λ_1, λ_2, λ_3, λ_4, we can draw the following conclusions from Eq. (14).*

1. *If the values of λ_1, λ_2, λ_3, and λ_4 change in direct proportion, the results of the MHS algorithm will hardly be influenced. Thus, we can let $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in (0, 1)$.*
2. *Whatever value λ_3 is chosen to be, convergence of the MHS algorithm will hardly be influenced.*
3. *In order for the MHS algorithm to converge, λ_4 should be much larger than λ_1 and λ_2, based on Eq. (17).*

Proof. 1. It is straightforward from Eq. (14).

2. In Eq. (14), where

$\frac{\partial J(t)}{\partial u_{ij}(t)} = 1$;

$\frac{\partial P(t)}{\partial u_{ij}(t)} = -u_{ij} \cdot \frac{\exp[-(u_{ij})^2/2\epsilon^2]}{\sum_{i=1}^{n}\sum_{j=1}^{n}\exp[-(u_{ij})^2/2\epsilon^2]} < 0$;

$\frac{\partial Q(t)}{\partial u_{ij}(t)} = -[\frac{1}{1+\exp(-10u_{ij})} - \frac{1}{2}] < 0$.

Putting the positive items of Eq. (14) on the left side of "<" and the negative items on the right side, Eq. (14) becomes

$$\lambda_1 + \lambda_2 < -u_{ij} \cdot \frac{\exp[-(u_{ij})^2/2\epsilon^2]}{\sum_{i=1}^{n}\sum_{j=1}^{n}\exp[-(u_{ij})^2/2\epsilon^2]} \cdot \lambda_3 - \frac{\partial Q(t)}{\partial u_{ij}(t)} \cdot \lambda_4 \tag{15}$$

In Eq. (15), since $u_{ij} \in [0, 1]$ and $0 < \epsilon < 1$, then $u_{ij} \cdot \frac{\exp[-(u_{ij})^2/2\epsilon^2]}{\sum_{i=1}^{n}\sum_{j=1}^{n}\exp[-(u_{ij})^2/2\epsilon^2]} \approx \frac{1}{n\times n}$.

Usually n is large because MHSA is good at large scale problems. Thus $\frac{1}{n\times n}$ is very small.

$\frac{1}{n\times n}$ is the coefficient of λ_3 in Eq. (15), and therefore λ_3 will hardly influence the convergence of the MHS algorithm.

3. Based on conclusion 2, Eq. (15) will approximately be

$$\lambda_1 + \lambda_2 < -\frac{\partial Q(t)}{\partial u_{ij}(t)}\lambda_4. \tag{16}$$

Where, $-\frac{\partial Q(t)}{\partial u_{ij}(t)} = \frac{1}{1+\exp(-10u_{ij})} - \frac{1}{2} \approx 0.4933$;
Therefore, Eq. (16) becomes

$$\lambda_1 + \lambda_2 < 0.4933 \cdot \lambda_4. \tag{17}$$

For example, if $\lambda_1 + \lambda_2 \leq 0.45$ and $\lambda_4 \geq 0.9$, then the above condition (Eq. (15)) will be satisfied and MHSA will converge to a stable equilibrium state. □

The larger $\frac{\lambda_4}{\lambda_1+\lambda_2}$, the faster MHSA would converge.

6.4. Convergence proofs

Theorems 3–6 indicate that all the artificial mosquitoes converge to their stable equilibrium points through the algorithm MHSA. For smooth reading, we take out the proofs of all the lemmas and theorems here and assign them to the Appendix.

Lemma 1. *If* $\gamma - 1 > -\Psi_2(t) > 0$, $\frac{\partial \Psi_2(t)}{\partial u_{ij}(t)} < 1$ *for* $u_{ij}(t) > 1$, *and* $\frac{\partial \Psi_2(t)}{\partial u_{ij}(t)} > 1 - \gamma$ *for* $0 < u_{ij}(t) < 1$, *then the artificial mosquito* m_{ij} *will converge to a stable equilibrium point with* $u_{ij}(t) > 1, v_{ij}(t) = 1$.

Lemma 2. *If* $\gamma > 1, -\Psi_2(t) < 0$, *and* $\frac{\partial \Psi_2(t)}{\partial u_{ij}(t)} < 1$ *for* $u_{ij}(t) > 1$, *then the artificial mosquito* m_{ij} *will converge to a stable equilibrium point with* $u_{ij}(t) > 1, v_{ij}(t) = 1$.

Lemma 3. *If* $\gamma > 1$, $\frac{\partial \Psi_2(t)}{\partial u_{ij}(t)} < 1$ *for* $u_{ij}(t) = 1^+$; *and* $\frac{\partial \Psi_2(t)}{\partial u_{ij}(t)} > 1 - \gamma$ *for* $u_{ij}(t) = 0^+$ *and* $u_{ij}(t) = 1^-$, *then the equilibrium points* s_1 *and* s_3 *in Fig. 5 are unstable and a saddle point, respectively.*

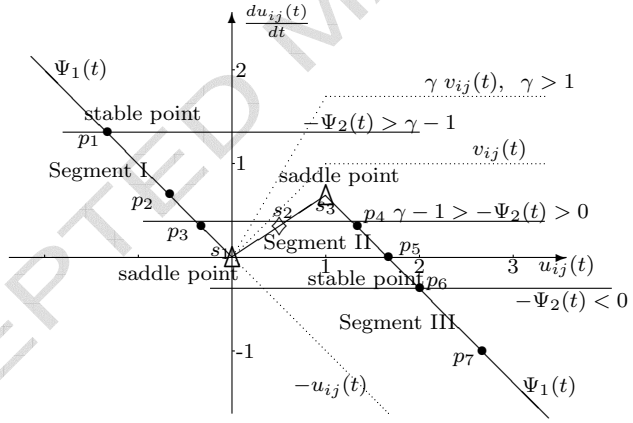

Fig. 5 When $\gamma > 1$, the reachable equilibrium points of the dynamic status $v_{ij}(t)$ of an artificial mosquito m_{ij}. The point where $-\Psi_2(t)$ equals $\Psi_1(t)$ is an equilibrium point. •, △ and ♢ denote a stable equilibrium point, saddle point and unstable equilibrium point, respectively.

Theorem 3. *If* $\gamma > 1$, $\frac{\partial \Psi_2(t)}{\partial u_{ij}(t)} < 1$ *for* $u_{ij}(t) \geq 1^+$, *and* $\frac{\partial \Psi_2(t)}{\partial u_{ij}(t)} > 1 - \gamma$ *for* $0^+ \leq u_{ij}(t) \leq 1^-$, *the dynamical equation* (8) *has a stable equilibrium point on Segment III iff the right side of the equation is larger than 0 for* $u_{ij}(t) = 1$ *and* $v_{ij}(t) = 1$.

Theorem 4. *If* $\gamma > 1$, $\frac{\partial \Psi_2(t)}{\partial u_{ij}(t)} < 1$ *for* $u_{ij}(t) \geq 1^+$, *and* $\frac{\partial \Psi_2(t)}{\partial u_{ij}(t)} > 1 - \gamma$ *for* $0^+ \leq u_{ij}(t) \leq 1^-$, *then the dynamical Eq.* (8) *has a stable equilibrium point iff*

$$\gamma > 1 + 2[\lambda_3 + 0.25\lambda_4]. \tag{18}$$

Lemma 4. *If* $\gamma > 1, \zeta_{ij}(t) \geq 0$, *and the following conditions are valid,*

$$\lambda_1 + \lambda_2 < (1 + 1/32)\lambda_3 + 0.25\lambda_4. \tag{19}$$

$$\gamma > \max\{1 + 4[\lambda_1 + \lambda_2 + 0.25\lambda_3 + \zeta_{ij}(t)\ \lambda_4], 1 + 2[\lambda_3 + 0.25\lambda_4]\}. \tag{20}$$

then the dynamical Eq. (8) *has a stable equilibrium point with* $u_{ij} = 1, v_{ij} = 1$.

Theorem 5. *If* $\gamma > 1$, *and the condition* (19) *are valid, then the dynamical Eq.* (8) *has a stable equilibrium point with* $u_{ij} = 1, v_{ij} = 1$.

Theorem 6. *If the conditions* (19) *and* (20) *remain valid, then MHSA will converge to a stable equilibrium state.*

7. The parallel MHS algorithm

Given the coordinates of n cities and x_{ij} (unavailable paths), our mosquito host-seeking algorithm (MHSA) is as follows.

Step 1. Initialize: Initialize the number of artificial mosquitoes to be $n \times n$. Initialize the sex of all artificial mosquitoes to be x_{ij}. Initialize all grayscale values r_{ij} of artificial mosquitoes m_{ij} to be the average values. Initialize all weight c_{ij} to be d_{ij}. Select the related coefficients $\epsilon, \lambda_1, \lambda_2, \lambda_3, \lambda_4$ to be $0.8, 0.05, 0.05, 0.9, 0.9$, respectively. Section 6.3 (Parameters analysis) gives the theoretical reasons for these values.

Step 2. For each computing cell $\mathfrak{C}_{ij}$ (artificial mosquito m_{ij}):

1. Calculate the $u_{ij}(t)$ of each m_{ij} by Eq. (1), and $du_{ij}(t)/dt$.
2. Calculate $dr_{ij}(t)$ by Eq. (6) and $dc_{ij}(t)$ by Eq. (7).
3. Update the grayscale value $r_{ij}(t)$ by $r_{ij}(t+1) = r_{ij}(t) + dr_{ij}(t)/dt$; update the weight $c_{ij}(t)$ by $c_{ij}(t+1) = c_{ij}(t) + dc_{ij}(t)/dt$.
4. If all $du_{ij}(t)/dt = 0$, then finish successfully.

8. Simulations

We give the experimental results in this section. First, we use a simple example (a small-scale problem) to explain how our MHS algorithm is used. Secondly, we show the actual times and iterations used to solve different TSP instances on a cluster, which should speak for the efficiency and parallelism of our MHSA. Thirdly, we make a performance comparison and a general comparison between MHSA and other benchmark NAs. All the experiments presented in this section were completed on a cluster. Each of the machines of the cluster has a Pentium 4 2.0 GHz CPU with 512 Kbytes of L2 cache and 512 Mbytes of DDR SDRAM, and they are interconnected via Fast Ethernet.

8.1. How to use the MHS algorithm

In order to show how the MHS algorithm can be used to solve a TSP instance, here we give a very simple example of a 10-city TSP for which MHSA has to find the shortest path.

The coordinates of 10 cities are initialized as random numbers between 0 and 1 (see Fig. 6). The X-axis and Y-axis coordinates of the 10 cities are:

$C_1(0.518000, 0.249000)$	$C_2(0.558000, 0.384000)$
$C_3(0.430000, 0.452000)$	$C_4(0.289000, 0.441000)$
$C_5(0.082000, 0.922000)$	$C_6(0.366000, 0.825000)$
$C_7(0.910000, 0.538000)$	$C_8(0.852000, 0.284000)$
$C_9(0.720000, 0.082000)$	$C_{10}(0.591000, 0.097000)$

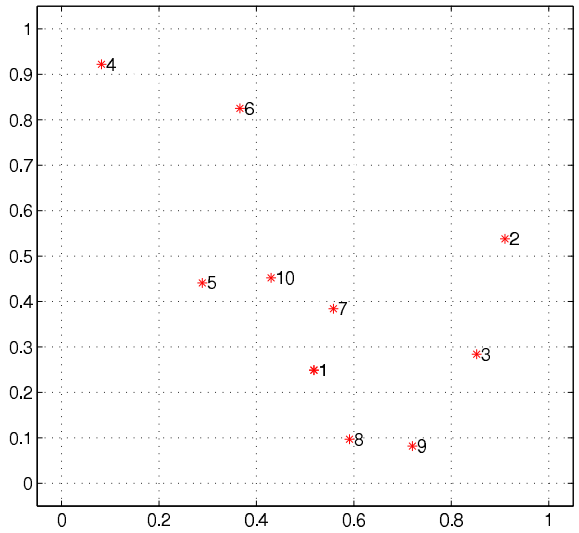

Fig. 6 An example of TSP

Step 1. Initialization: $(t = 0)$

1. x_{ij}:

There are 100 artificial mosquitoes. All $x_{ij} = 1$ $i = \overline{1, 10}, j = \overline{1, 10}$. That is, all paths between any two cities are available, and all artificial mosquitoes are female.

2. c_{ij}:

(1) Initialize the weight c_{ij} of all artificial mosquitoes. Set c_{ij} to be $\max\limits_{i,j} d_{ij} - d_{ij}$. This way the maximization problem is transformed to a minimization problem.

(2) Initialize the 10 artificial mosquitoes m_{ii} to be extremely weak, that is, set $d_{ii} = 10, i = \overline{1, 10}$, so as to make the 10 artificial mosquitoes not able to seek and attack the host. r_{ii} will not be equal to 1.

(3) When $i \neq j$, $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, where (x_i, y_i) is the coordinates of city C_i.

(4) In addition, the initial matrix C should be dealt with using the following three steps.

- $c_{ij} = 0 \mid \forall x_{ij} = 0$.
- Nonnegativity: Let $c_{ij} = c_{ij} - \min\limits_{i,j} c_{ij}$.
- Normalization. Let $c_{ij} = 2\ c_{ij} / \sum\limits_{j=1}^{10} c_{ij}$ and $c_{ij} = 2\ c_{ij} / \sum\limits_{i=1}^{10} c_{ij}$.

Columns 1–5 of the initial matrix $(c_{ij})_{10\times 10}$ are

$c_{ij}(t = o)$	1	2	3	4	5
1	0	0.2210	0.2228	0.2195	0.2226
2	0.2206	0	0.2285	0.2207	0.2188
3	0.2226	0.2287	0	0.2171	0.2183
4	0.2185	0.2201	0.2164	0	0.2267
5	0.2224	0.2191	0.2185	0.2276	0
6	0.2187	0.2225	0.2182	0.2363	0.2250
7	0.2240	0.2228	0.2228	0.2209	0.2225
8	0.2259	0.2215	0.2250	0.2173	0.2208
9	0.2246	0.2236	0.2278	0.2161	0.2192
10	0.2227	0.2209	0.2200	0.2246	0.2261

Columns 6–10 of the initial matrix $(c_{ij})_{10\times 10}$ are

$c_{ij}(t=o)$	6	7	8	9	10
1	0.2191	0.2240	0.2261	0.2249	0.2227
2	0.2225	0.2223	0.2213	0.2235	0.2204
3	0.2184	0.2225	0.2250	0.2279	0.2197
4	0.2357	0.2199	0.2165	0.2155	0.2235
5	0.2253	0.2223	0.2208	0.2194	0.2259
6	0	0.2207	0.2168	0.2164	0.2236
7	0.2212	0	0.2227	0.2225	0.2238
8	0.2171	0.2225	0	0.2299	0.2207
9	0.2165	0.2221	0.2298	0	0.2197
10	0.2241	0.2238	0.2209	0.2200	0

3. r_{ij}:

(1) Initialize all grayscale values r_{ij} of artificial mosquitoes m_{ij} to be the average values. $(r_{ij})_{10\times10} = (2/n)_{10\times10} = (0.2)_{10\times10}$.

Here r_{ij} are initialized by the average values. Alternatively, r_{ij} can also be initialized as random numbers between 0 and 1. In fact, based on the experimental results, we found that the results are not affected by the initialization of r_{ij}.

(2) the initial matrix $(r_{ij})_{10\times10}$ should be dealt with using the following three steps, too.

- $r_{ij} = 0 \mid \forall x_{ij} = 0$. If $x_{ij} = 0$, then $r_{ij} = 0$.
- Nonnegativity: $0 \le r_{ij} \le 1$. If $\min\limits_{i,j} r_{ij} < 0$, then let $r_{ij} = r_{ij} - \min\limits_{i,j} r_{ij}$.
- Normalization. Let $r_{ij} = 2\ r_{ij} / \sum\limits_{j=1}^{10} r_{ij}$ and $r_{ij} = 2\ r_{ij} / \sum\limits_{i=1}^{10} r_{ij}$, in order that TSP can be solved.

4. $Z(R)$:

$Z(R)(t=0) = \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} d_{ij} r_{ij} x_{ij} = 28.5866$.

Step 2. For each computing cell $\mathfrak{C}_{ij}$ (artificial mosquito m_{ij}): (we show only the first evolutionary iteration; t=1)

1. Compute Δr_{ij} and Δc_{ij} in parallel:

(1) Δr_{ij}:

According to Eq. (6), we have

$$\Delta r_{ij} \approx dr_{ij}/dt = -\lambda_1 \frac{\partial u_{ij}}{\partial r_{ij}} - \lambda_2 \frac{\partial J}{\partial r_{ij}} - \lambda_3 \frac{\partial P}{\partial r_{ij}} - \lambda_4 \frac{\partial Q}{\partial r_{ij}}$$

where,

$\frac{\partial u_{ij}}{\partial r_{ij}} = -c_{ij} x_{ij} \exp(-c_{ij} r_{ij} x_{ij})$

$\frac{\partial J}{\partial r_{ij}} = -c_{ij} x_{ij} \exp(-c_{ij} r_{ij} x_{ij})$

$\frac{\partial P}{\partial r_{ij}} = \frac{\partial P}{\partial u_{ij}} \cdot \frac{\partial u_{ij}}{\partial r_{ij}} = -u_{ij} \cdot \frac{exp(-u_{ij}^2/2\epsilon^2)}{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} exp(-u_{ij}^2/2\epsilon^2)} \cdot \frac{\partial u_{ij}}{\partial r_{ij}}$

$\frac{\partial Q}{\partial r_{ij}} = 2\sum\limits_{j=1}^{n} x_{ij} \cdot \sum\limits_{i=1}^{n}(\sum\limits_{j=1}^{n} r_{ij} x_{ij} - 1) - [\frac{1}{1+\exp(-10u_{ij})} - \frac{1}{2}] \cdot \frac{\partial u_{ij}}{\partial r_{ij}}$

(2) Δc_{ij}:

According to Eq. (7), we have

$$\Delta c_{ij} \approx dc_{ij}/dt = \lambda_1 \frac{\partial u_{ij}}{\partial c_{ij}} - \lambda_2 \frac{\partial J}{\partial c_{ij}} - \lambda_3 \frac{\partial P}{\partial c_{ij}} - \lambda_4 \frac{\partial Q}{\partial c_{ij}}$$

where, $\frac{\partial u_{ij}}{\partial c_{ij}} = -r_{ij} x_{ij} \exp(-c_{ij} r_{ij} x_{ij})$

$$\frac{\partial J}{\partial c_{ij}} = -r_{ij}x_{ij}\exp(-c_{ij}r_{ij}x_{ij})$$

$$\frac{\partial P}{\partial c_{ij}} = \frac{\partial P}{\partial u_{ij}} \cdot \frac{\partial u_{ij}}{\partial c_{ij}} = -u_{ij} \cdot \frac{exp(-u_{ij}^2/2\epsilon^2)}{\sum_{i=1}^{n}\sum_{j=1}^{n} exp(-u_{ij}^2/2\epsilon^2)} \cdot \frac{\partial u_{ij}}{\partial c_{ij}}$$

$$\frac{\partial Q}{\partial c_{ij}} = -[\frac{1}{1+\exp(-10u_{ij})} - \frac{1}{2}] \cdot \frac{\partial u_{ij}}{\partial c_{ij}}$$

2. Update $r_{ij}(t=1)$ and $c_{ij}(t=1)$ in parallel:

$$r_{ij}(t=1) = r_{ij}(t=0) + \Delta r_{ij}(t=1)$$

$$c_{ij}(t=1) = c_{ij}(t=0) + \Delta c_{ij}(t=1)$$

$$\lambda_1 = 0.05 \quad \lambda_2 = 0.05 \quad \lambda_3 = 0.9 \quad \lambda_4 = 0.9 \quad \epsilon = 0.8$$

After we get $r_{ij}(t=1)$ and $c_{ij}(t=1)$, matrices R and C should be dealt with using the three steps mentioned in Step 1–3–(2) and the three steps mentioned in Step 1–2–(4), respectively.

Updated for the first time, columns 1–5 of matrix $(r_{ij})_{10\times10}$ are

$r_{ij}(t=1)$	1	2	3	4	5
1	0	0.2201	0.2226	0.2180	0.2223
2	0.2201	0	0.2315	0.2203	0.2176
3	0.2226	0.2315	0	0.2150	0.2167
4	0.2180	0.2203	0.2150	0	0.2297
5	0.2223	0.2176	0.2167	0.2297	0
6	0.2174	0.2229	0.2168	0.2428	0.2264
7	0.2242	0.2225	0.2225	0.2199	0.2221
8	0.2273	0.2211	0.2261	0.2152	0.2200
9	0.2257	0.2242	0.2302	0.2137	0.2180
10	0.2224	0.2198	0.2186	0.2252	0.2272

Columns 6–10 of matrix $(r_{ij})_{10\times10}$ are

$r_{ij}(t=1)$	6	7	8	9	10
1	0.2174	0.2242	0.2273	0.2257	0.2224
2	0.2229	0.2225	0.2211	0.2242	0.2198
3	0.2168	0.2225	0.2261	0.2302	0.2186
4	0.2428	0.2199	0.2152	0.2137	0.2252
5	0.2264	0.2221	0.2200	0.2180	0.2272
6	0	0.2203	0.2148	0.2142	0.2244
7	0.2203	0	0.2224	0.2221	0.2239
8	0.2148	0.2224	0	0.2331	0.2199
9	0.2142	0.2221	0.2331	0	0.2187
10	0.2244	0.2239	0.2199	0.2187	0

Updated for the first time, columns 1–5 of matrix $(c_{ij})_{10\times10}$ are

$c_{ij}(t=1)$	1	2	3	4	5
1	0	0.2205	0.2226	0.2189	0.2223
2	0.2205	0	0.2295	0.2208	0.2185
3	0.2226	0.2295	0	0.2165	0.2178
4	0.2189	0.2207	0.2165	0	0.2281
5	0.2223	0.2185	0.2178	0.2281	0
6	0.2184	0.2228	0.2179	0.2384	0.2255
7	0.2238	0.2224	0.2225	0.2204	0.2221
8	0.2262	0.2213	0.2253	0.2167	0.2205
9	0.2249	0.2238	0.2285	0.2155	0.2189
10	0.2223	0.2203	0.2193	0.2246	0.2261

Columns 6–10 of matrix $(c_{ij})_{10\times10}$ are

$c_{ij}(t=1)$	6	7	8	9	10
1	0.2185	0.2238	0.2262	0.2250	0.2223
2	0.2228	0.2224	0.2213	0.2238	0.2203
3	0.2179	0.2225	0.2253	0.2285	0.2193
4	0.2384	0.2204	0.2167	0.2155	0.2246
5	0.2255	0.2221	0.2205	0.2189	0.2261
6	0	0.2207	0.2164	0.2159	0.2240
7	0.2207	0	0.2224	0.2221	0.2235
8	0.2164	0.2224	0	0.2308	0.2204
9	0.2159	0.2221	0.2308	0	0.2194
10	0.2240	0.2235	0.2204	0.2194	0

When Step 2 is repeated 29 times, at the end of the 29-th iteration, we get the equilibrium state of the TSP example. Matrix $(r_{ij})_{10\times10}(t=29)$ is

$r_{ij}(t=end)$	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	1	1	0	0
2	0	0	1	0	0	1	0	0	0	0
3	0	1	0	0	0	0	0	0	1	0
4	0	0	0	0	1	1	0	0	0	0
5	0	0	0	1	0	0	0	0	0	1
6	0	1	0	1	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	1
8	1	0	0	0	0	0	0	0	1	0
9	0	0	1	0	0	0	0	1	0	0
10	0	0	0	0	1	0	1	0	0	0

When all mosquitoes are in the equilibrium state, all grayscale values r_{ij} converge to 0 or 1 using the MHS algorithm. $r_{ij} = 1$ represents the artificial mosquito m_{ij} having attacked the host, as well as the path p_{ij} being black (the shortest path Z passes through the path). On the contrary, $r_{ij} = 0$ represents m_{ij} did not attack the host, and p_{ij} is white (Z does not pass through the path). In this way, based on $(r_{ij})_{10\times10}(t=29)$, we can draw the shortest path of the example TSP problem in Fig. 6 as Fig. 10.

The experimental results of the TSP example are summarized in Table 1.

Table 1: Experimental results of the TSP example

Parallel processors	1
TSP scale	10
Convergence iterations	29
Convergence time(second)	2
$Z(t=29)$	5.3327
$Z(t\in(29,\infty))$	5.3327

As shown in Table 1, at $t = 29$, the shortest path $Z(= 5.3327)$ is obtained, and stays unchanged in the remainder of the iterations. That is, for $t > 29$, Z and matrix R stay unchanged. Obviously, $Z(t = 0) > Z(t = 29)$, and the TSP example is optimized.

What is more, using the MHS algorithm, only the same optimal solution will be obtained for the same TSP. This verifies the stability of MHSA. Because the scale of the TSP example is very small, we only use one processor in the cluster.

The evolutionary process of the TSP example using MHSA is shown in Fig. 7–10.

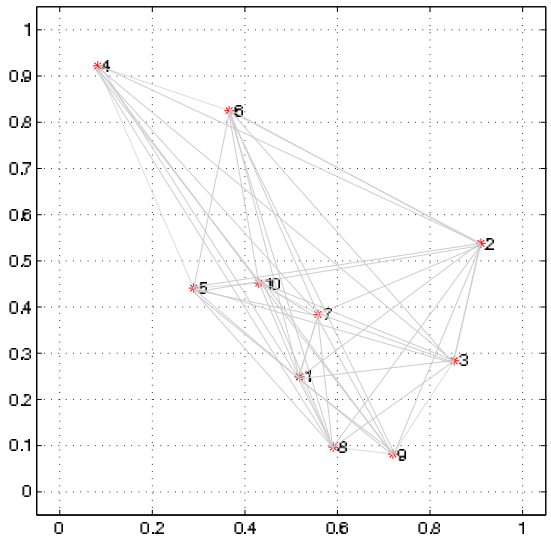

Fig. 7 Grayscale values $r_{ij}(t = 0)$

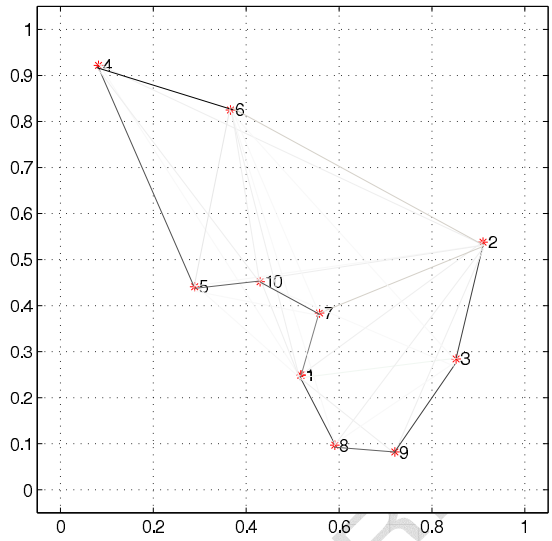

Fig. 8 Grayscale values $r_{ij}(t = 10)$

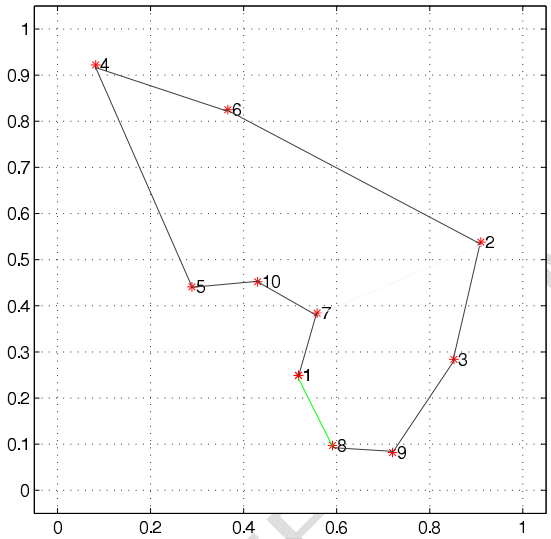

Fig. 9 Grayscale values $r_{ij}(t = 20)$

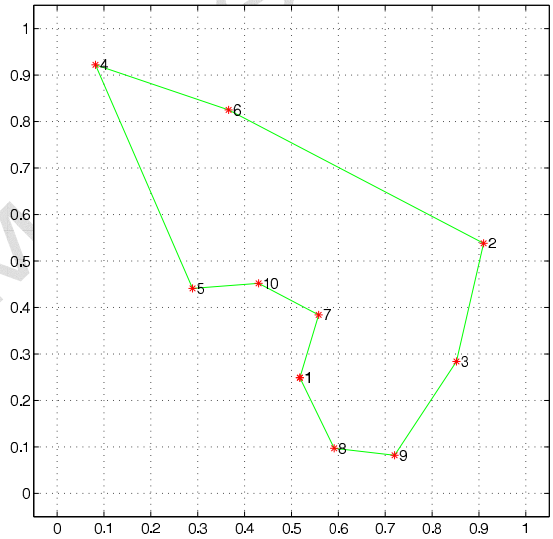

Fig. 10 Grayscale values $r_{ij}(t \geq 29)$

In the four figures, the darker the line between two cities, the larger r_{ij} is. When $r_{ij} = 1$, a stable state is obtained, and path p_{ij} should be drawn in black. We use a green line instead to distinguish the stable state (black line) from other gray lines.

From a large amount of experimental results, we find that all grayscale values r_{ij} will always converge to 0 or 1 regardless of the TSP scale using MHSA. For example, we used 250 parallel processors in the cluster to solve a 5000-city TSP problem by MHSA. 25000000 r_{ij} evolved and updated in parallel. After 28694 seconds (about 8 hours), all 25000000 r_{ij} converged to 0 or 1.

8.2. Efficiency and parallelism of MHS algorithm

The MHS algorithm provides a valuable alternative to traditional methods because of its inherent parallelism. The grayscale values and weights, r_{ij}, c_{ij}, can be computed and updated in parallel without any information exchange, which is the foundation of MHSA's efficiency. The experimental results clearly confirm the outstanding parallel capability of MHSA (see Table 2). We used 1, 8, and 16 computing nodes of the cluster, respectively.

Table 2: Convergence time and speeds of MHSA with scale

Scale	16 parallel nodes		8 parallel nodes		1 parallel node	
cities	time(s)	iterations	time(s)	iterations	time(s)	iterations
110	0.983	242	1.90	166	16.767	176
160	2.583	600	5.74	348	58.827	393
200	5.453	1046	12.62	602	169.793	816
225	10.8	1759	25.87	973	343.651	1312
250	15.868	2227	42.034	1293	569.017	1832
275	29.419	3327	63.617	1722	1044.953	2842
300	38.901	4159	99.039	2276	1571.968	3711
320	63.718	5711	130.258	2679	2113.836	4351
340	84.762	6806	183.987	3410	3358.083	5839
360	118.984	8404	247.028	4122	4276.371	6758
375	155.671	9816	336.842	4988	5839.789	8245
390	262.693	12544	446.898	5834	8130.714	9963
405	307.017	13909	578.865	6845	9729.199	11014
420	374.298	15629	699.978	7718	13528.261	13064
435	425.01	17180	861.57	8609	15294.553	14319
450	594.781	19854	997.091	9528	18108.043	15921
465	698.079	21922	1641.718	11881	25035.577	18459
480	774.948	23927	1418.417	11523	27951.384	20024
495	987.354	26780	2122.651	13738	37178.713	22961
510	1332.217	30418	2196.459	14275	38786.457	24314

Table 2 includes the sequential version which uses one computing node of the cluster. The other parts of the table are for the parallel version, using 4 and 16 computing nodes of the cluster. "Iterations" and "time" are the number of iterations and the time the MHSA algorithm takes to converge, respectively. As shown in Table 2, for a 510-city TSP, the convergence time using 8 computing nodes is about 1/7 the time of the sequential version; and the convergence time using 16 computing nodes is about half the time with 8 computing nodes. The convergence time is almost inversely proportional to the number of computing nodes used by MHSA.

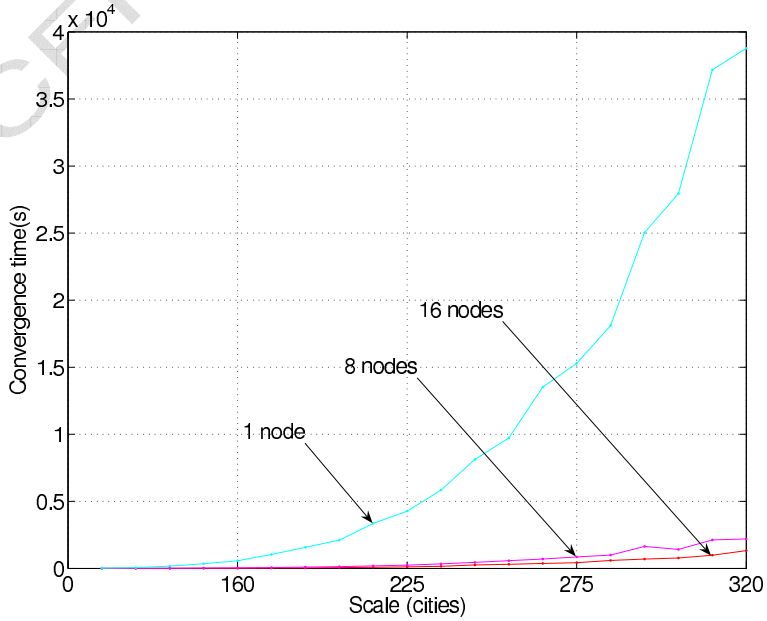

Fig. 11 Convergence time of MHSA with scale

As shown in Fig. 11, the convergence time of the sequential version increases exponentially with the scale, which is similar to all the exact methods. When parallelized, the convergence time drops significantly, which speaks for the high parallelism of MHSA.

8.3. Comparison between MHSA and other benchmark NAs

As mentioned in Section 1, popular nature-inspired approaches (NAs) include genetic algorithm (GA), simulated annealing algorithm (SA), ant colony optimization (ACO), particle swarm optimization (PSO), etc. The proposed MHSA and these benchmark NAs share some common features, which are listed in Table 3.

Table 3: Common features between MHSA and other benchmark NAs

Aspects	Common features
Drawn from	Observations of physical processes that occur in nature
Belong to	The class of meta-heuristics, approximate algorithms
Parallelism	Have inherent parallelism
Performance	Consistently perform well
Fields of application	Artificial intelligence including multi-objective optimization
Solvable problems	All kinds of difficult problems

Next, we compare the proposed algorithm, MHSA, with ant colony optimization(ACO), simulated annealing (SA), elastic net (EN), and self-organizing map (SOM). Results on ACO, SA, EN, and SOM are from [6], [7] and [16]. The MHSA results are averaged over 25 trials. The comparison results on average tour length obtained for five 50-city TSP instances are presented in Table 4.

As shown in Table 4, MHSA has better performance than other benchmark NAs on TSP.

Table 4: Comparison between MHSA and other benchmark NAs on random instances of symmetric TSP

	MHSA	ACO	SA	EN	SOM
City set 1	5.75	5.88	5.88	5.98	6.06
City set 2	6.00	6.05	6.01	6.03	6.25
City set 3	5.49	5.58	5.65	5.70	5.83
City set 4	5.66	5.74	5.81	5.86	5.87
City set 5	6.21	6.18	6.33	6.49	6.70

We also summarize the relative differences between MHSA and the benchmark NAs in Table 5.

Table 5: Relative differences between HMSA and other benchmark NAs

	HMSA	GA	SA	ACO	PSO
Inspired by	Host-seeking behavior of mosquitoes	Natural evolution	Thermodynamics	Behaviors of real ants	Biological swarm (e.g., swarm of bees)
Key components	Hybrid attraction function; differential dynamic equations	Chromosomes	Energy function	Pheromone laid	Velocity-coordinate model
Exploration	Both Macro-evolutionary and Micro-evolutionary processes	Macro-evolutionary processes	Micro-evolutionary processes	Macro-evolutionary processes	Macro-evolutionary processes
Dynamics	Can capture the entire dynamics inherent in the problem	Cannot capture	Can capture partly	Cannot capture	Cannot capture
High-dimensional, highly nonlinear, random behaviors and dynamics	Can describe	Cannot describe	Can describe partly	Cannot describe	Cannot describe
Adaptive to problem changes	Fast	Middle	Fast	Low	Middle
Exchange overhead	Low	Middle	Low	Low	Middle

9. Conclusion

From observations on the host-seeking behaviors of mosquitoes, we find that the kinematics and dynamics of mosquitoes exhibit the properties of parallelism, openness, local interactivity, and self-organization. This stimulates us to develop a new model for intelligent computing by mimicking the host-seeking behaviors of mosquitoes. The mosquito host-seeking (MHS) model can overcome some of the limitations of existing popular NAs. In summary, MHSA has the following advantages:

1. By its inherent parallel structure, the model admits a highly parallel algorithmic implementation and thus possesses the ability to deal with large-scale complex problems.
2. We have proved that the MHS algorithm would converge to a stable equilibrium state, based on Lyapunov second theorem on stability. And since the solution obtained by MHSA can approach the theoretical optimum solution, MHSA offers better performance than other benchmark NAs proposed in recent literature.
3. Parameter values are deduced from the convergence proofs of the MHSA algorithm, and hence their setting does not depend on algorithm learning or any prior knowledge.
4. It is fundamentally different from the other popular NAs in terms of its motivation and basic principle, the optimization mechanism, the elements and their states, and the biological model, the mathematical model and theoretical foundation on which it is based.
5. It is robust in the sense that it is basically independent of the initial conditions, problem size, small-range changes of the parameters, etc.
6. It can be applied to the optimization of multiple objectives which could include aggregate utility, personal utility, minimal personal utility, etc.
7. It has a powerful processing ability in a complex and dynamic real-time changing environment.
8. It has greater flexibility than other methods, making it very easy to adapt to for a wide range of optimization problems.
9. It can describe complex, high-dimensional, highly nolinear, micro-evolutionary and random behaviors and dynamics. This is due to the introduction of the aggregate intention strength factor (ζ_{ij}) in the definition of the artificial mosquitoes' interaction behavior function $Q(t)$. In the future, by studying the complex problems involved in a variety of complicated social interactions and autonomous behaviors, we can try to categorize and model such social behaviors and construct a mathematical model to compute ζ_{ij}.

Appendix

Proof of Lemma 1. For $\gamma > 1$, $\Psi_1(t)$ of artificial mosquito m_{ij} is a piecewise linear function of the stimulus $u_{ij}(t)$, as shown in Fig. 5: Segment I, Segment II, and Segment III. By Eq. (8), a point is an equilibrium point, i.e., $du_{ij}(t)/dt = 0$, iff $-\Psi_2(t)=\Psi_1(t)$ at the point. We see that for the case of $\gamma - 1 > -\Psi_2(t) > 0$, an equilibrium point may be on Segment I, II or III. Note from Eq. (1), $u_{ij}(t) \geq 0$. Thus we need not consider the equilibrium point on Segment I.

Suppose that the artificial mosquito m_{ij} is at an equilibrium point on Segment III at time t_0, and an arbitrarily small perturbation $\triangle u_{ij}$ to the equilibrium point occurs at

time t_1. Since $\frac{\partial \Psi_2(t)}{\partial u_{ij}(t)} < 1$, and $\frac{\partial \Psi_1(t)}{\partial u_{ij}(t)} = -1$ for $u_{ij}(t) > 1$, we have $c = [\frac{\partial \Psi_1(t)}{\partial u_{ij}(t)} + \frac{\partial \Psi_2(t)}{\partial u_{ij}(t)}] < 0$, and

$\triangle \frac{du_{ij}}{dt} = \frac{du_{ij}}{dt} \mid_{t_1} - \frac{du_{ij}}{dt} \mid_{t_0} = \frac{du_{ij}}{dt} \mid_{t_1} = \triangle[\Psi_1(t) + \Psi_2(t)] \approx [\frac{\partial \Psi_1(t)}{\partial u_{ij}(t)} + \frac{\partial \Psi_2(t)}{\partial u_{ij}(t)}] \triangle u_{ij} = - \mid c \mid \triangle u_{ij}$.

That means $\frac{du_{ij}}{dt} \mid_{t_1}$ is always against $\triangle u_{ij}$, or in other words, the perturbation will be suppressed and hence the artificial mosquito m_{ij} will return to the original equilibrium point.

Whereas, for an equilibrium point on Segment II, because

$\frac{\partial \Psi_2(t)}{\partial u_{ij}(t)} > 1 - \gamma$ and $\frac{\partial \Psi_1(t)}{\partial u_{ij}(t)} = \gamma - 1 > 0$, we have

$c = [\frac{\partial \Psi_2(t)}{\partial u_{ij}(t)} + \frac{\partial \Psi_1(t)}{\partial u_{ij}(t)}] > 0$, and

$\frac{du_{ij}}{dt} \mid_{t_1} \approx [\frac{\partial \Psi_1(t)}{\partial u_{ij}(t)} + \frac{\partial \Psi_2(t)}{\partial u_{ij}(t)}] \triangle u_{ij} = \mid c \mid \triangle u_{ij}$

such that the perturbation is intensified and the artificial mosquito m_{ij} departs from the original equilibrium point on Segment II. Therefore, an equilibrium point on Segment II is unstable, and only an equilibrium point on Segment III, e.g., p_4 in Fig. 5, is stable with $u_{ij}(t) > 1, v_{ij}(t) = 1$. □

Proof of Lemma 2. Due to $\gamma > 1$ and $-\Psi_2(t) < 0$, an equilibrium point must be on Segment III, e.g., p_6 in Fig. 5. Moreover, as stated in the proof of Lemma 1, $\frac{\partial \Psi_2(t)}{\partial u_{ij}(t)} < 1$ for $u_{ij}(t) > 1$ guarantees any equilibrium point on Segment III to be stable, with $u_{ij}(t) > 1, v_{ij}(t) = 1$. □

Proof of Lemma 3. It is straightforward from the proof of Lemma 1 . □

Proof of Theorem 3. By Lemmas 1–3, the equilibrium points on Segment II and Segment III, except for the saddle point s_3 in Fig. 5, are unstable and stable, respectively. We denote the right side of Eq. (8) by RHS.

Sufficiency. Assume that $RHS = \frac{du_{ij}(t)}{dt} = \Psi_1(t) + \Psi_2(t) > 0$ holds for $u_{ij}(t) = 1, v_{ij}(t) = 1$. It follows that $\Psi_1(t) \neq -\Psi_2(t)$ for $u_{ij}(t) = 1, v_{ij}(t) = 1$, namely, it is impossible that the equilibrium point is the intersection point s_3 where $\Psi_1(t) = -\Psi_2(t)$. Since $RHS = \frac{du_{ij}}{dt} > 0$ at point $u_{ij}(t) = 1$, $v_{ij}(t) = 1$, the increase of $u_{ij}(t)$ from value 1 leads to the situation that the artificial mosquito m_{ij} converges to a stable equilibrium point on Segment III.

Necessity. Suppose that Eq. (8) has a stable equilibrium point. We need to prove that $RHS > 0$ holds for $u_{ij}(t) = 1$ and $v_{ij}(t) = 1$. By contrary, if there is $RHS \leq 0$ for $u_{ij}(t) = 1$ and $v_{ij}(t) = 1$, then the equilibrium point must be either at the point s_3 where $RHS = 0$, or on Segment II because $RHS = \frac{du_{ij}}{dt} < 0$ would give rise to the decrease of $u_{ij}(t)$ from value 1. Since the point s_3 and any equilibrium point on Segment II are all non-stable, we have a contradiction. □

Proof of Theorem 4. By Eqs. (1), (2), (3), (4) we have

$\Psi_2(t) = \{\lambda_1 + \lambda_2 - \lambda_3 \omega_{ij}^2(t) u_{ij}^2(t) - \lambda_4[(1 + \exp(-\zeta_{ij}(t) u_{ij}(t)))^{-1} - 0.5]^2\}[r_{ij}^2(t) + c_{ij}^2(t)][u_{ij}(t)]^2$ (21)

Note that $\omega_i^2(t), r_{ij}^2(t), c_{ij}^2(t) \leq 1$. By Theorem 1, for $u_{ij}(t) = 1, v_{ij}(t) = 1$, we have

$$\begin{aligned}\gamma > 1 - \Psi_2(t) &= 1 + \{-\lambda_1 - \lambda_2 + \lambda_3 \omega_{ij}^2(t) u_{ij}^2(t) \\ &\quad + \lambda_4[(1 + \exp(-\zeta_{ij}(t) u_{ij}(t)))^{-1} - 0.5]^2\}[r_{ij}^2(t) + c_{ij}^2(t)][u_{ij}(t)]^2 \\ &\leq 1 + 2[\lambda_3 + 0.25\lambda_4].\end{aligned}$$

□

Proof of Lemma 4. Using Eq. (21) and noting $\frac{d\omega_{ij}(t)}{du_{ij}(t)} = 2\omega_{ij}(t)u_{ij}(t)(\omega_{ij}(t)-1)$, we obtain

$$\begin{aligned}\frac{\partial\Psi_2(t)}{\partial u_{ij}(t)} &= \{2[-\lambda_1-\lambda_2+\lambda_3\omega_{ij}^2(t)u_{ij}^2(t)+\lambda_4[(1+\exp(-\zeta_{ij}(t)u_{ij}(t)))^{-1}-0.5]^2]\\ &+[-2\lambda_3\omega_{ij}(t)\tfrac{d\omega_{ij}(t)}{du_{ij}(t)}u_{ij}^2(t)-2\lambda_3\omega_{ij}^2(t)u_{ij}(t)-2\lambda_4\zeta_{ij}(t)\exp(-\zeta_{ij}(t)u_{ij}(t))[(1+\exp(-\zeta_{ij}(t)u_{ij}(t)))^{-1}-0.5]\\ &\times[1+\exp(-\zeta_{ij}(t)u_{ij}(t))]^{-2}](-u_{ij}(t))\}x_{ij}^2(t)[r_{ij}^2(t)+c_{ij}^2(t)][-u_{ij}(t)]\\ &= \{2[-\lambda_1-\lambda_2+\lambda_3\omega_{ij}^2(t)u_{ij}^2(t)+\lambda_4[(1+\exp(-\zeta_{ij}(t)u_{ij}(t)))^{-1}-0.5]^2]\\ &+[-4\lambda_3\omega_{ij}^2(t)u_{ij}^3(t)(\omega_{ij}(t)-1)-2\lambda_3\omega_{ij}^2(t)u_{ij}(t)-2\lambda_4\zeta_{ij}(t)\exp(-\zeta_{ij}(t)u_{ij}(t))[(1+\exp(-\zeta_{ij}(t)u_{ij}(t)))^{-1}-0.5]\\ &\times[1+\exp(-\zeta_{ij}(t)u_{ij}(t))]^{-2}](-u_{ij}(t))\}x_{ij}^2(t)[r_{ij}^2(t)+c_{ij}^2(t)][-u_{ij}(t)]\end{aligned}$$

Then from $\frac{\partial\Psi_2(t)}{\partial u_{ij}(t)} < 1$ for $u_{ij}(t) \geq 1^+$; and $u_{ij}(t) \leq \omega_{ij}(t), r_{ij}(t), c_{ij}(t) \leq 1$, we derive

$$\frac{\partial\Psi_2(t)}{\partial u_{ij}(t)} < 2[2(-\lambda_1-\lambda_2+\lambda_3+0.25\lambda_4)+4\lambda_3(2/3)^2(3/4)^3/12] < 1,$$

which leads to Eq. (19).

Similarly, from $\frac{\partial\Psi_2(t)}{\partial u_{ij}(t)} > 1-\gamma$ for $0^+ \leq u_{ij}(t) \leq 1^-$, we have

$$\frac{\partial\Psi_2(t)}{\partial u_{ij}(t)} > 4(-\lambda_1-\lambda_2-\lambda_3/4-\zeta_{ij}(t)\lambda_4) > 1-\gamma,$$

which leads to Eq. (20). By Theorem 4, therefore the conclusion is valid. □

Proof of Theorem 5. Straightforward, by Lemmas 4 and 5. □

Proof of Theorem 6. Using Eq. (21) and noting $\frac{d\omega_{ij}(t)}{du_{ij}(t)} = 2\omega_{ij}(t)u_{ij}(t)(\omega_{ij}(t)-1)$, we obtain

$$\begin{aligned}\frac{\partial\Psi_2(t)}{\partial u_{ij}(t)} &= \{2[-\lambda_1-\lambda_2+\lambda_3\omega_{ij}^2(t)u_{ij}^2(t)\\ &\quad+\lambda_4[(1+\exp(-\zeta_{ij}(t)u_{ij}(t)))^{-1}-0.5]^2]\\ &\quad+[-2\lambda_3\omega_{ij}(t)\tfrac{d\omega_{ij}(t)}{du_{ij}(t)}u_{ij}^2(t)-2\lambda_3\omega_{ij}^2(t)u_{ij}(t)\\ &\quad-2\lambda_4\zeta_{ij}(t)\exp(-\zeta_{ij}(t)u_{ij}(t))\\ &\quad[(1+\exp(-\zeta_{ij}(t)u_{ij}(t)))^{-1}-0.5]\\ &\quad[1+\exp(-\zeta_{ij}(t)u_{ij}(t))]^{-2}](-u_{ij}(t))\}\\ &\quad x_{ij}^2(t)[r_{ij}^2(t)+c_{ij}^2(t)][-u_{ij}(t)]\\ &= \{2[-\lambda_1-\lambda_2+\lambda_3\omega_{ij}^2(t)u_{ij}^2(t)\\ &\quad+\lambda_4[(1+\exp(-\zeta_{ij}(t)u_{ij}(t)))^{-1}-0.5]^2]\\ &\quad+[-4\lambda_3\omega_{ij}^2(t)u_{ij}^3(t)(\omega_{ij}(t)-1)-2\lambda_3\omega_{ij}^2(t)u_{ij}(t)\\ &\quad-2\lambda_4\zeta_{ij}(t)\exp(-\zeta_{ij}(t)u_{ij}(t))\\ &\quad[(1+\exp(-\zeta_{ij}(t)u_{ij}(t)))^{-1}-0.5]\\ &\quad[1+\exp(-\zeta_{ij}(t)u_{ij}(t))]^{-2}](-u_{ij}(t))\}\\ &\quad x_{ij}^2(t)[r_{ij}^2(t)+c_{ij}^2(t)][-u_{ij}(t)]\end{aligned}$$

For the force field F, we define a Lyapunov function $L(t)$ by

$$\begin{aligned}L(t) &= -\tfrac{1}{2}\sum_{i,k}(\gamma-1)v_{ij}(t)^2+\sum_{i,k}\int_0^t\tfrac{dv_{ij}(x)}{dx}\{-\lambda_1-\lambda_2\\ &+\lambda_3\omega_{ij}^2(x)u_{ij}^2(x)+\lambda_4[(1+\exp(-\zeta_{ij}(x)u_{ij}(x)))^{-1}-0.5]^2\}\\ &\quad[r_{ij}^2(x)+c_{ij}^2(x)][-u_{ij}(x)]^2dx.\end{aligned}$$

We hence have

$$|\ L(t)\ | \leq \sum_{i,k}\ (\ \gamma-1\ )\ |\ v_{ij}(t)^2\ |\ +\ \sum_{i,k}\ \int_0^t\ |\ \tfrac{dv_{ij}(x)}{dx}\ |$$

$$\cdot \mid \{ -\lambda_1 - \lambda_2 + \lambda_3 \omega_{ij}^2(x) u_{ij}^2(x) + \lambda_4[(1+\exp(-\zeta_{ij}(x)u_{ij}(x)))^{-1} - 0.5]^2\} \mid [r_{ij}^2(x) + c_{ij}^2(x)][-u_{ij}(x)]^2 dx.$$

Since condition (20) is valid, $v_{ij}(t) \leq 1$ and $u_i(t) \leq 1$, it follows that

$$\mid L(t) \mid \leq \sum_{i,k} (\gamma - 1) + \sum_{i,k} \gamma < m\, n\, \gamma$$

which implies that $L(t)$ is bounded.

In addition, we have

$$\begin{aligned}
\frac{dL(t)}{dt} &= -\sum_{i,k}(\gamma-1)v_{ij}(t)\frac{dv_{ij}(t)}{dt} \\
&\quad + \sum_{i,k}\frac{dv_{ij}(t)}{dt}\{-\lambda_1 - \lambda_2 + \lambda_3\omega_{ij}^2(t)u_{ij}^2(t) \\
&\quad + \lambda_4[(1+\exp(-\zeta_{ij}(t)u_{ij}(t)))^{-1} - 0.5]^2\} \\
&\quad [r_{ij}^2(t) + c_{ij}^2(t)][-u_{ij}(t)]^2 \\
&= -\sum_{i,k}\frac{dv_{ij}(t)}{du_i(t)}\frac{du_{ij}(t)}{dt}\{-u_{ij}(t) + \gamma v_{ij}(t) \\
&\quad + \{\lambda_1 + \lambda_2 - \lambda_3\omega_{ij}^2(t)u_{ij}^2(t) \\
&\quad - \lambda_4[(1+\exp(-\zeta_{ij}(t)u_{ij}(t)))^{-1} - 0.5]^2\} \\
&\quad [r_{ij}^2(t) + c_{ij}^2(t)][-u_{ij}(t)]^2\} \\
&= -\sum_{i,k}\frac{dv_{ij}(t)}{du_{ij}(t)}(\frac{du_{ij}(t)}{dt})^2 \leq 0.
\end{aligned}$$

Thus $L(t)$ will monotonically decrease with the elapsing time. □

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant No. 60905043, 61073107 and 61173048, the Innovation Program of Shanghai Municipal Education Commission, and the Fundamental Research Funds for the Central Universities.

References

[N] L. Adleman, Molecular computation of solutions to combinatorial problems, **Science** 266 (1994) 1021-1024.

[2] L.T. Berners, W. Hall, J. Hendler, N. Shadbolt, D. Weitzner, Creating a science of the Web, **Science** 313 (2006) 769-771.

[P] E. Bonabeau, M. Dorigo, G. Theraulaz, Inspiration for optimization from social insect behavior, **Nature** 406 (2000) 39-42.

[Q] R. Brooks, Artificial life: from robot dreams to reality, **Nature** 406 (2000) 945-947.

[R] P. Denning, Computing is a natural science, Communications of the ACM 50(2007) 13-18.

[6] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation 1 (1997) 53-66.

[7] R. Durbin, D. Willshaw, An analogue approach to the travelling salesman problem using an elastic net method, **Nature** 326 (1987) 689-691.

[8] S. Effati, M. Pakdaman, Artificial neural network approach for solving fuzzy differential equations, **Information Sciences** 180 **(2010)** 1434-1457.

[9] C.M. Fernandes, J.J. Merelo, A.C. Rosa, A comparative study on the performance of dissortative mating and immigrants-based strategies for evolutionary dynamic optimization, **Information Sciences** 181 **(2011)** 4428-4459.

[10] S. Forrest, Genetic algorithms - principles of natural-selection applied to computation, **Science** 261 (1993) 872-878.

[11] K. Helsgaun, An effective implementation of the Lin-Kernighan traveling salesman heuristic, European Journal of Operational Research 126 (2000) 106-130.

[12] T. P. Hong, Y. F. Tung, S. L. Wang, Y. L. Wu, M. T. Wu, A multi-level ant-colony mining algorithm for membership functions, **Information Sciences** 182 **(2012)** 3-14.

[13] L. Jia, I. Katsunobu, Q. S. Zhu, Fluctuation-driven computing on number-conserving cellular automata, **Information Sciences** 182 **(2012)** 266-276.

[14] S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, **Science** 220 (1983) 671-680.

[15] T.A.S. Masutti, L.N.d. Castro, A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem, **Information Sciences** 179 **(2009)** 1454-1468.

[16] J.Y. Potvin, The traveling salesman problem: A neural network perspective, ORSA J. Comput. 5 (1993) 328-347.

[17] G. Reinelt, TSPLIB-a traveling salesman problem library, ORSA Journal on Computing 4 (1996) 134-143.

[18] Y. Shi, H. C. Liu, L. Gao, G. H. Zhang, Cellular particle swarm optimization, **Information Sciences** 181 **(2011)** 4460-4493.

[19] C.F. Tsai, C.W. Tsai, C.C. Tseng, A new hybrid heuristic approach for solving large traveling salesman problem, **Information Sciences** 166 (2004) 67-81.

[20] X.F. Xie, J. Liu, Multiagent optimization system for solving the traveling salesman problem (TSP), IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 39 **(2009)** 489-502.

[21] R. Xu, D. Wunsch, R. Frank, Inference of genetic regulatory ntworks with recurrent neural network models using particle swarm optimization, IEEE/ACM Trans. on Computational Biology and Bioinformatics 4(2007) 681-692.