Ad Hoc & Sensor Wireless Networks, Vol. 28, pp. 257–287
Reprints available directly from the publisher
Photocopying permitted by license only

Social Group Search Optimizer Algorithm for Ad Hoc Network

XIANG FENG*, MEIYI MA, HUIQUN YU AND ZHE WANG

Department of Computer Science and Engineering, East China, University of Science and Technology

Received: October 10, 2012. Accepted: January 6, 2014.

Due to the dynamic structure in network topology and absence of a centralized administration in management, a specific routing algorithm satisfying the demands of QoS is required indeed in mobile Ad Hoc networks. A novel Social Group Search Optimizer algorithm is proposed by improving the GSO algorithm to a dynamic and discrete algorithm through the introducing of social behaviors. SGSO is divided into search and prey parts, where "search" is on duty to find the optimal solution effectively and "prey" is responsible for adjusting the algorithm to the dynamic change of objective functions. Dynamic Coupling Level is used to divide the Ad Hoc network and corresponding approaches and models based on SGSO are applied to routing algorithm, including the decision factor and local routing table. The convergence and correctness of our algorithm are verified mathematically and extensive experiments have been conducted to evaluate the efficiency and effectiveness of the proposed mechanism in mobile Ad Hoc networks. The results show that SGSO improves packet delivery ratio and reduces average end-to-end latency effectively, especially for large-scale and high-dynamic networks.

Keywords: Ad Hoc network, social behavior, social group searching optimization, dynamic network, quality of service

1 INTRODUCTION

Recently in computer science, there is an increased interest in intelligent algorithms that are inspired by the swarm intelligence and social behaviors

* Corresponding authors: Address: Department of Computer Science and Engineering, East China University of Science and Technology, Meilong Road 130, Shanghai 200237, PR China. Tel.: +86 15021382004. E-mail addresses: xfeng@ecust.edu.cn (X. Feng), mameiyi0106@163.com (M. Ma), yhq@ecust.edu.cn (H. Yu), wangzhe@ecust.edu.cn (Z. Wang).

Property	SGSO	GSO	PSO	ACO
Conceptual inspiration	Social behaviors	Animal searching behavior	Animal swarm behavior	Ants' path following behavior
Searching strategies	Search and prey	Producing,scrounging and ranging	Flocking	Following pheromone trails
Object	Dynamic or static	Static	Static	Static
Information communication	Internal and Public communications	Given out by producer	Given out by the best particle	Given out by pheromone
Suitable problems	Discrete,Continuous, high-dimensional, multimodal	Continuous, high-dimensional, multimodal	Continuous, unimodal	Discrete

TABLE 1
Comparisons Between SGSO, GSO, PSO and ACO

of animal and that can solve difficult problems [1-3]. On the one hand, some researchers focus on the improvement and applications of classic successful nature-inspired approaches including the Ant Colony Optimizer algorithm (ACO) [4], which has virtual ants as agents that communicate indirect way and uses randomly propagation rules that make difficult to understand algorithms and agents' behavior, and Particle Swarm Optimizer algorithm (PSO) [5], which is to simulate the social interaction behavior of birds flocking and fish schooling. On the other hand, some novel bio-inspired algorithms are proposed. By studying the evolution of "swarms", the artificial mosquitoes' microcosmic actions and macroscopic swarm intelligence, Mosquito Host-Seeking Algorithm (MHSA), inspired by the host-seeking behavior of mosquitoes, is proposed in [6]. Similarly, Artificial Bee Swarm Optimization (ABSO) is a recently invented algorithm inspired by the intelligent behaviors of honey bees such as collection and processing of nectar [7].

Group Search Optimizer algorithm, as a novel algorithm inspired by swarm intelligence and social behavior, has a great performance on some classical optimizer problems. It has proved to have much better convergence and efficiency in benchmark functions mathematically and experimentally in [8]. The comparison among PSO, ACO and GSO is shown in Table 1. Even though, GSO and other algorithms still have the following disadvantages prevent them from a variety of practical applications:

- They all have a good performance in continuously objective functions or discrete problems with little scale, but remain stuck in large scale discrete problems.
- The objective functions of problems or applications are static.
- They have tendency to fall into local optimal or fail to find the optimal solution in large scale problems.

However, with the development of hardware technology and increasing demand in network, large-scale and high-dynamic networks, such as Ad Hoc network, are much more popular in military and industry. Therefore, appropriate and specific routing algorithms are required to take place of the classical static routing algorithm eagerly. How to develop a proper routing approach satisfying the strict constraints of wireless dynamic network and QoS demands has attracted researchers' attention.

[9] studies how to support fair bandwidth allocation among all end-to-end flows in a multihop wireless network to achieve the global max-min fairness objective in bandwidth allocation. It develops a novel theory that maps the global max-min objective to four local conditions and designs a distributed rate adjustment protocol to achieve the global max-min objective through fully distributed operations. By modeling a link delay as a function of the signal to interference noise ratio of the receiving node in this link and its packet forwarding time and taking a weight sum of delay and energy consumption as weight of edge, Optimal Edge-cost Topology Control (OETC) algorithm and Distributed Symmetric Link Maintenance (DSLM) algorithm are proposed to obtain the minimum weight sum of any edge [10]. Unreliable and short-term connectivity can increase communication cost due to frequent failure and activation of links, and ineffective resource allocation can increase communication cost due to multi hop communication between dependent tasks. A two-phase resource allocation scheme to reduce communication cost between dependent tasks is proposed in [11]. [12] identifies the key issues that impact end-to-end connection performance when a DSA-enabled WLAN is integrated with the wired cloud and propose a new network management framework, called DSASync, to mitigate the identified performance issues. [13] studies the problem of congestion control and scheduling in ad hoc wireless networks that have to support a mixture of best-effort and real-time traffic and propose a model for incorporating the quality-of-service requirements of packets with deadlines in the optimization framework. A novel algorithm, named a distributed channel assignment control, is proposed in [14] that focus on performance enhancements related to QoS and mathematical analysis techniques for the channel bandwidth. This novel algorithm uses channel assignment control with a power control to reduce the negative effects induced by the quasi-exposed node problem, and then the channels are adaptively negotiated to allow communication in the interference region. Optimizing over the number of hops, single hop transmission based on Delay-reliability (D-R), and throughput-delay-reliability (T-D-R) tradeoffs is shown to be optimal for maximizing a lower bound on the transmission capacity in the sparse network regime under quality of service constraints in an ad hoc network [15]. Multicasting through Time Reservation using Adaptive Control for Energy efficiency [16] provides superior energy efficiency while

producing competitive QoS performance and bandwidth efficiency by enabling the nodes to switch to sleep mode frequently and by eliminating most of the redundant data receptions. Source Routing Mechanism (TSR) proposed in [17] provides a flexible and feasible approach to choose the shortest route that meets the security requirement of data packets transmission. It improves packet delivery ratio and reduces average end-to-end latency by evaluating the trustworthiness of nodes through their behaviors.

By analyzing the pros and cons of bio-inspired algorithms existing and social behaviors, as well as the characters and tackle key problems in Ad Hoc networks, we proposed a novel Social Group Search Optimizer algorithm by improving the GSO algorithm to a dynamic and discrete algorithm through the introducing of social behaviors. SGSO is divided into two stages, search and prey. In search part, GSO algorithm is maintained with the improved efficiency through internal and public behaviors. New strategy inspired by prey behavior of wolves is introduced to prey part of SGSO. SGSO has these advantages (Table 1):

- It has the ability to perform large-scale distributed parallel optimization;
- It has good performance in both dynamic and static problems.
- It can deal with both discrete and continuous problem.
- It can converge;
- It can describe complex behaviors and dynamics;
- It has a comprehensive optimization ability for multiple objectives;
- It has a powerful processing ability in a complex, high-dimensional and dynamic real-time changing environment;
- It is flexible and easy to adapt to a wide range of optimization problems.

Ad Hoc network is divided into four DC Levels according to its dynamic coupling level as well. Therefore, SGSO is applied to solve the dynamic and QoS-satisfied Ad Hoc network. Local routing table based on the theory of six degree space of separation and decision factor inspired by the decision behavior are also introduced to improve the effectiveness and efficiency of our algorithm. The mathematical proofs for convergence and correctness of our algorithm are provided. The scheme is validated in a simulation environment using various workloads and parameters and the results show that it can successfully fit the dynamic environment, and guarantee QoS by maintaining good delivery ratio, reducing total overhead, and enhancing delay.

The structure of the rest of the paper is as follows. In Section 2, social group behaviors are discussed and mathematical model for SGSO is built according to social group behaviors in Section 3. The problem model of Ad Hoc network is formalized and analyzed in Section 4 and solved by SGSO in Section 5. Section 6 provides the proofs of the correctness and convergence of our algorithm. In Section 7, we present simulation results which attest to the effectiveness and suitability of SGSO for Ad Hoc network. Finally, conclusions are drawn in Section 9.

2 SOCIAL GROUP BEHAVIOR

All the manifestations of the interaction and influence between the animals living together are defined as Social behaviors, including the dominance hierarchy sequence, communication behavior, courtship behavior, altruistic behavior, pro-kill behavior and other typical representative behaviors. Social behaviors not only have an irreplaceable role in nature for maintaining the stability of natural ecosystems, species diversity and many other aspects, but also can improve the performance and effectiveness of algorithm when they are introduced to intelligent algorithm.

2.1 Prey Behavior of Wolves

There're two kinds of social relationships in wolves, which are the relationship between wolves themselves, and relationship between them and other animals. Moreover, competition, mutual benefit and predation are the main acts between wolves and other animals. This paper focuses on the relationships and relevant behaviors between wolves and preys.

The wolf society organizations follow a strict social class model. Their social organization is divided into three levels: the first level is the leader of the wolves in the highest level - Alpha level, served by a pair of male wolf and female wolf; The second level is inferior to the leadership level of the chiefs - Beta level, served by a pair of male wolf and female wolf; remaining wolves in group belong to the third level, the lowest level of the wolves - Omega level. Wolves in different levels play different roles and have different social division of labor in activities. For example, Alpha wolves and Beta wolf can breed, while Omega wolves are mainly responsible for hunting. Similarly, there is a clear division of labor in the process of predation.

Wolves, as social animals, have strong team spirit. Predator is a typical social group behavior. The process of predator, including search for prey, pursuit of fleeing pry and hunting, is as shown in Figure 1. When wolves prey, they will search the prey firstly and select a target to follow. Then wait the right time to attack. If the prey hasn't found the wolves nearby, unfortunately, it will become the food of wolves immediately. However, if prey is lucky enough to sense them, they will flee and wolves have to chase. The chasing will end up as two kinds of results, which are getting and losing. During the pursuit, only one to three wolves chase per time to drag down prey and attack it together. If wolves lose the prey, they have to stop to search another prey

FIGURE 1
Prey Behavior of Wolves

again. The prey model is refined according to the prey process of wolves, and the flow is as shown in Figure 2.

The process of wolves' prey can be refined as an effective intelligent search algorithm. The prey can be regarded as the object, wolves as search individuals and the final goal is a quick and exact strategy to hunt. Compared with other intelligent algorithm inspired by animals, the most advantage of wolves is that their prey is dynamic. By observing the process of prey, we

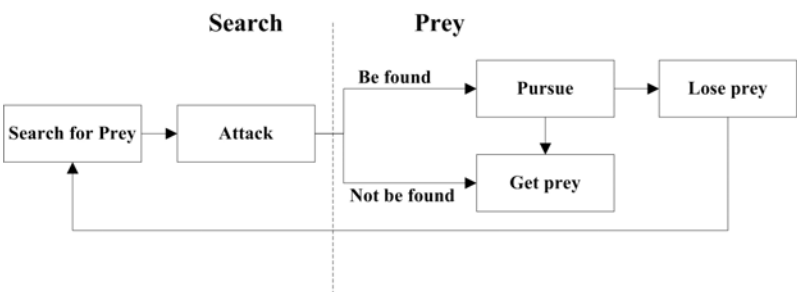


FIGURE 2
Flow of Wolf Prey

found that wolves' prey, which is different from some other lower animals, hasn't stopped by the time they find the prey. There is a process of chase, or prey, as well. If the prey hasn't found them, it will stay still and be caught, which corresponds to the situation of static object in search algorithm. However, on the contrary, under the long-term of nature selection, prey will react rapidly when the hunter nearby, and wolves have to chase on them with the higher or at least equal speed. This process can be extracted as the model of dynamic object problem.

For most algorithms, the algorithm inspired by wolves is different from other lower animals. For example, the Ant Colony Optimization algorithm, ants only need to search for the food in a certain place, namely the static object. They leave pheromone on the path to attract more ants. However, they don't have a plan B when the object moves, but only to search again. In this condition, their former work is completely wasted. Moreover, if the object moves faster than the speed of search, the algorithm may not find its position. On the contrary, wolves prey strategy compensates this problem with the dynamic prey. Re-searching only happens when they fail to pursuit.

In summary, social group search optimizer algorithm consists of two important parts, one is searching part to find the original position of prey, and the other is preying part to pursuit and catch the prey.

2.2 Internal & Public Behaviors

We shouldn't ignore that there is an important strategy when the wolves pursuit the prey. Only one to three wolves chase on them per time and pursuit in turn to drag down the prey. This is the biggest advantage of wolves than tiger or other animals, which prey alone. No matter prey or any other social group behaviors, the intelligence of group is the division of labor. If all the individuals in group work alone, the group benefit is just a total number or less. But if they have interactions, group benefit may be improved. However, over-interaction will also affect the results. Image that if wolves keep questioning their hunting strategy and comparing with others, instead of focusing on the prey, can they really catch the prey? It's only to give more fleeing time for prey. That situation also happens in the GSO algorithm. Individuals have to rank by the result of objective function right after iteration. New producer and followers are refined in each iteration. This is a typical over-interaction behavior. Too much time is spent on communication and unnecessary cost is taken by interaction and ranking.

Therefore, we suggest that the behaviors of group animals should be divided into two parts, internal behavior and public behavior. Internal behavior can't be affected by others, such as running or resting, while public behavior is interaction between individuals, such as running towards predator, or eliminate individuals which perform relatively bad. Appropriate internal behaviors should be added to the GSO algorithm. Insert the internal behaviors into public ones to make them hunting and interacting alternately in a proper order.

2.3 Decision Behavior

By the research of variety of decision behaviors of social animals and human beings, Franz J. Weissing proposed a leaders model, pointing out that there are a few individuals acting as leaders while others acting as followers. This leader-follower relationship is formed spontaneously and influent the behavior of social group. Amount of experiments and researches show that this model is available for any group, only with different influent factor deciding the leader for different group.

To reap the benefits of group living, some animals have to neglect their own references and following a leader. There is a famous coordination problem known to game theorists as the Battle of the Sexes, which imagines a married couple who want to spend the evening together. Husband and wife (the players) can either go to a football game or to the opera, but they cannot communicate with each other about where to meet. Neither wants to miss their partner by going to a different event from them. If that happens, both get a pay-off of zero. When they go to the same event, the wife would prefer the opera, whereas the husband would prefer the football game. When meeting at the same event, the players get the pay-offs 1 and 1-k (where $0 < k < 1$), depending on whether or not they realize their preferred option.

Leaders Model is proposed according to this game theory, and happens to fit for group with more than two individuals. Assuming that there are interactions between individuals in social groups, and they will either insist their preferred option or ignore it to follow others. Each player is characterized by a strategy, λ, corresponding to the player's probability of sticking to his or her preferred action. Individual with higher λ is leader, while with lower λ is follower.

A population of only leaders ($\lambda = 1$) is not evolutionistic stable, because they will never meet and will get a pay-off of zero. Likewise, a population of only followers ($\lambda = 0$) is not stable either, because the players will again miss each other and will get a pay-off of zero. Instead, the population will first evolve to an intermediate value of λ ($\lambda = 0.5$). However, this is not the final outcome. From the intermediate strategy, the population will diversify and evolve to a state where two strategies coexist – a leader strategy (say, $\lambda = 0.9$) and a follower strategy (say, $\lambda = 0.1$).

This model can be applied to lots of group behavior and provides a new direction of development for intelligent algorithms simulated biological behavior.

3 MATHEMATICAL MODEL FOR SOCIAL GROUP SEARCH OPTIMIZER

Social Group Search Optimizer algorithm (SGSO) is proposed by introducing the social group behaviors of wolves. The SGSO algorithm is an expansion and improvement of original Group Search Optimizer algorithm.

The SGSO algorithm is divided into two parts, search and prey. In the search part, we inherit the GSO only to improve it with internal behaviors. While in the prey part, a new pursuit strategy is introduced to adjust to the dynamic situation effectively.

3.1 Search: Improved Group Search Optimizer

The GSO algorithm divides individuals into three characters, producer, scroungers and dispersed members based on the Producer- Scrounger (PS) model. Three roles have different search strategies and find the location of prey by the guidance of objective function. The characters will be resigned each iteration according to their performance. However, the overhead cost by frequent interaction will reduce the efficiency. Therefore, more internal behaviors are inserted into public behaviors in SGSO to improve the performance of search part.

Search Space of SGSO:

The search space is an n-dimensional space, where the scanning field of vision is simplified and generalized to an n-dimensional space, which is characterized by maximum pursuit angle $\theta_{\max} \in \mathrm{R}^1$, and maximum pursuit distance $l_{\max} \in \mathrm{R}^1$. The ith member at the kth searching iteration has a current position $x_i^k \in \mathrm{R}^n$, a head angle $\varphi_i^k = (\varphi_{i1}^k, ..., \varphi_{i(n-1)}^k) \in R^{n-1}$. The search direction of the ith member, which is a unit vector $D_i^k(\varphi_i^k) = (d_{i1}^k, ..., d_{in}^k) \in R^n$ that can be calculated from φ_i^k via a polar to Cartesian coordinate transformation

$$d_{i1}^k = \prod_{q=j}^{n-1} \cos(\varphi_{iq}^k)$$

$$d_{ij}^k = \sin(\varphi_{i(j-1)}^k) \cdot \prod_{q=j}^{n-1} \cos(\varphi_{iq}^k)(j = 2, ...n-1)$$

$$d_{in}^k = \sin(\varphi_{i(n-1)}^k)$$

where $r_1 \in R^1$ is a normally distributed random number with mean 0 and standard deviation 1 and $r_2 \in R^{n-1}$ is a uniformly distributed random sequence in the range (0, 1).

This strategy is employed by SGSO to handle the bounded search space: when a member is outside the search space, it will turn back into the search space by setting the variables that violated bounds to its previous values.

Producer:

In the SGSO algorithm, at the *k*th iteration the producer X_p behaves as follows. The producer will scan at zero degree and then scan laterally by randomly sampling three points in the scanning field: one point at zero degree, one point in the right hand side hypercube and one point in the left hand side hypercube:

$$X_z = X_p^k + r_1 l_{\max} D_p^k(\varphi^k)$$

$$X_r = X_p^k + r_1 l_{\max} D_p^k(\varphi^k + r_2\theta_{\max}/2)$$

$$X_l = X_p^k - r_1 l_{\max} D_p^k(\varphi^k - r_2\theta_{\max}/2)$$

The producer will then find the best point with the best resource (fitness value). If the best point has a better resource than its current position, then it will fly to this point. Or it will stay in its current position and turn its head to a new randomly generated angle

$$\varphi^{k+1} = \varphi^k + r_2\alpha_{\max}$$

where $\alpha_{\max} \in R^1$ is the maximum turning angle. If the producer cannot find a better area after a iterations, it will turn its head back to zero degree

$$\varphi^{k+a} = \varphi^k$$

where $a \in R^1$ is a constant. During each searching bout, a number of group members are selected as scroungers. The scroungers will keep searching for opportunities to join the resources found by the producer.

Scroungers:

At the *k*th iteration, the area copying behavior of the *i*th scrounger can be modeled as a random walk toward the producer

$$X_i^{k+1} = X_i^k + r_3 \circ (X_p^k - X_i^k)$$

where $r_3 \in R^1$ is a uniform random sequence in the range (0,1). Operator "$\circ$" is the Hadamard product or the Schur product, which calculates the entrywise product of the two vectors. During scrounging, the *i*th scrounger will keep searching for other opportunities to join [2]. We modeled this behavior by turning the ith scrounger's head to a new randomly generated angle.

Dispersed Members:
At the kth iteration, it generates a random head angle φ_i; and then it chooses a random distance

$$l_i = a \cdot r_1 l_{\max}$$

and move to the new point

$$X_i^{k+1} = X_i^k + l_i D_i^k(\varphi^{k+1})$$

Social Interaction Behaviors:
By analyzing the behaviors of social group, we divide the behavior of SGSO into two parts: internal and public behaviors. Individuals execute their own strategy themselves without looking at the others in internal part and exchange search information to resign their character in public part. There is an interactional factor δ controlling the interaction, drawing a line between internal behavior and public behavior. Specifically, there is a public interactional behavior every δ internal search behavior.

This interaction improvement for SGSO is significant. Not only reduce the unnecessary interaction and increase the efficiency of search, but also enhance the parallelism of algorithm, as shown in Figure 3. Although GSO is

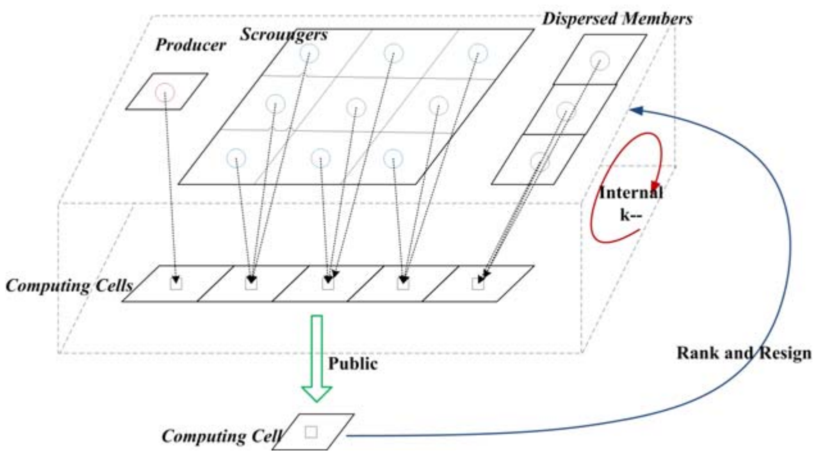


FIGURE 3
Parallel Structure of SGSO

famous for its parallel structure and well performance on great scale problem, individuals cannot be assigned to computation cells effectively with overmuch interaction. However, interactional factor δ gives the regular space for internal search. Producer, scroungers and dispersed members are allocated to different computation cells. After δ iterations, a computation cell is responsible for ranking individuals according to their values of objective function and resigning their characters by the ranking results. Repeating this internal search and public ranking until optimal result appears.

3.2 Prey: Dynamic Object

As we've claimed in last section, search part only finds the position of prey rather than catches it. If the prey realizes that it has been found, it will escape immediately. Individuals have to react quickly and effectively to catch them. Namely, when the objective function $f(X_1)$ changes to $f'(X_2)$, individuals in SGSO have to change their position and speed as well.

Assuming that Δf is the relative variable, thus,

$$\Delta f = f(X_2) - f'(X_1)$$

All the functional transformation can be summarized as the transformation of variable in different dimensions. Set two-dimension function $f = x_1^2 + x_2^2$ as a simple example, X-axis and Y-axis represent the independent variables and Z-axis represents the objective function, as shown in Figure 4. Thus the transformation of n-dimension function can be decomposed to *(n+1)*th-axis and handled respectively.

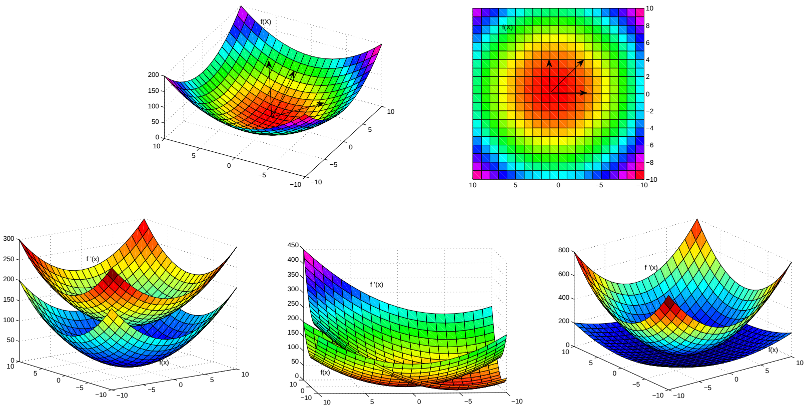
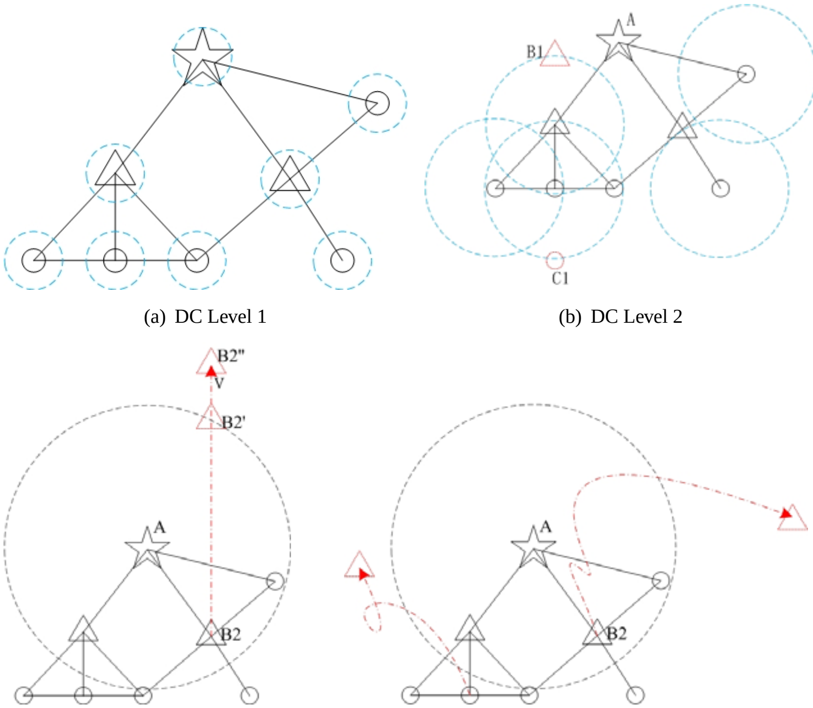


FIGURE 4
Function Transformations



(a) DC Level 1 (b) DC Level 2

(c) DC Level 3 (d) DC Level 4

FIGURE 5
Structures and Movement of Four Dynamic Coupling Levels

(1) If Δf is a constant, function transformation is alone the *(n+1)*th-axis, as shown in Figure 5 Left. Although the value of optimal solution may be changed, the position of optimal solution won't are exactly the same. Thus since the solution position hasn't been changed, there's no need to change the individuals in SGSO.

(2) If Δf isn't a constant, i.e. Δf is relevant to X, as shown in Figure 5 Middle. Although the value of optimal solution hasn't been changed, the position of optimal solution has changed. By this condition, the transformation of X should be discussed.

If the transformation ΔX is difference,

$$\Delta X = X_2 - X_1$$

ΔX should be decomposed to every axis.

$$\Delta x_i = x_{2i} - x_{1i}$$

Where, Δx_i is constant.

(3) If ΔX is multiple, i.e.

$$\Delta X = X_2/X_1$$

$\Delta x_i = x_{2i}/x_{1i}$, Δx_i is constant. In this condition, both the value and position of optimal solution will change, as shown in Figure 5 right.

In summary, the variable of the objective function can be represents as $f'(X) = f(\alpha X + \beta) + \theta$. According to the analysis above, θ doesn't have any influence on optimal solution, while, α and β do. The corresponding adjustment should be done with the SGSO, or it has to re-search.

When objective function changes to $f'(X)$,

$$f'(X) = f(\alpha X + \beta) + \theta$$

Every dimension of each individual in SGSO should make the equal transformation,

$$X' = \alpha X + \beta$$

It's worth noting that each dimension's transformation is different,

$$\Delta x_i = \Delta f * \cos \varphi_i$$

And $x_i{}' = (x_i - \beta_i)/\alpha_i$

Therefore, when the objective function has a transformation, individuals in SGSO should change and go on search part until they get the prey.

4 PROBLEM DESCRIPTION: AD HOC NETWORK

4.1 Dynamic Network

One of the biggest character of Ad Hoc network is dynamic. Here, we introduce a new definition to describe the dynamic networks, Dynamic Coupling Level (DC Level).

Therefore, the Ad Hoc network can be divided into four levels according to their dynamic degree. The higher level means higher dynamic. Characters for different dynamic coupling levels are shown in Table 2. DC Level 1 is the basic level with static or micro-movers, such as the temporary wireless networks for exhibition hall. Devices moving ruleless in a certain range, for example, the wireless sensor networks or emergence service, makes up the DC Level 2. In DC Level 3, device may move with a certain velocity in some

DC Level	Type	Problem	Application
1	Static or Micro-movers	Static	Exhibition Hall
2	Move in certain range	Range	Emergence Service
3	Move with certain velocity	Velocity	Military Activity
4	Move without constraints	Position	Communications in Battle

TABLE 2
Characters of Dynamic Coupling Levels

direction, which always happens in military activities. Setting communications in battle as an example, devices in DC Level 4 move ruleless without any constraints.

The structures and movement of four dynamic coupling levels are as shown in Figure 5, where the devices are represented by different graphic symbols, blue circles are the possible moving ranges of devices, grey circles are the transmission ranges, and the red symbols and arrows represent the possible positions after moving and moving directions respectively. Devices inside others' ranges are linked with lines. By observing these four figures, we can analyze the key problems and seek proper solutions for them. As for DC Level 1 in Figure 6(a), devices are static or barely moving, which makes the static route table important. Moving range and velocity are the key points for communication in Figure 6(b) and Figure 6(c), respectively. When it comes to ruleless moving without any constraints, the new position is of significant. The solutions will be discussed concretely in next section.

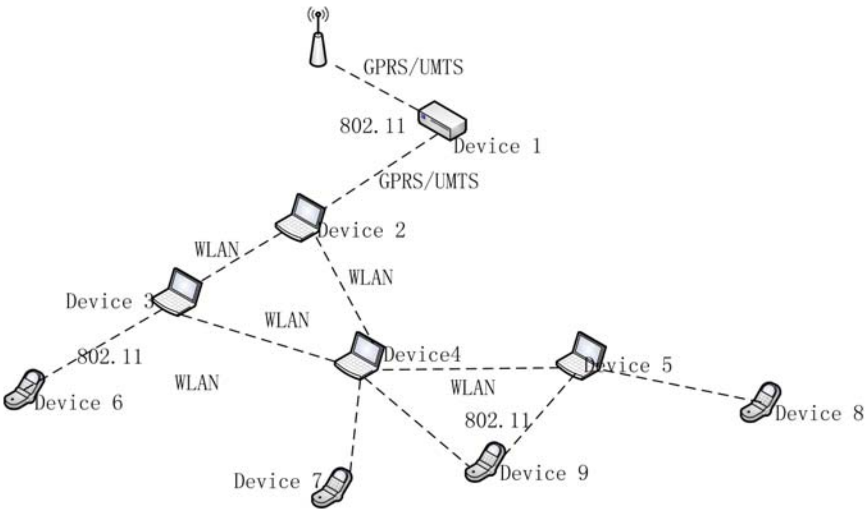


FIGURE 6
Ad Hoc Network

4.2 QoS-guaranteed Network

Besides dynamic character, QoS-guaranteed routing is another tackle key problem for Ad Hoc networks. Because the transmission range and bandwidth in dynamic networks are much more limited than static networks, and lead to more constraints. QoS-guaranteed routing algorithm is to find out the minimum cost path from source to destination satisfying constraints.

Definition 1. *Weighted graph* $\Omega = (\Lambda, \Phi)$ *is defined as the Ad Hoc Network, where* Λ *is the set of devices and* Φ *is the set of paths between devices.* $|\Lambda|$ *and* $|\Phi|$ *are the number of devices and path respectively. They are variable with the dynamic moving of devices. Paths are from Device* i *to other devices which are inside Device* i*'s range. For each path* $\phi_{ij} \in \Phi$ *and* $\phi_{ij} = (\lambda_i, \lambda_j), \lambda_i \in \Lambda, \lambda_j \in \Lambda$.

Definition 2. *Device is represented by a triple* $\lambda_i < Type,\ Bandwidth,\ Range >$*, where, Type=*$\{D, S\}$ *(D is short for Dynamic and S is short for Static) represents the dynamic type of device, Bandwidth is the capability of the device, and Range is the transmission of the device. The bandwidth, range and overhead of devices* λ_i *can also be represented as Bandwidth(*λ_i*), Range(*λ_i*) and Overhead(*λ_i*).*

Definition 3.

$$Overhead(path(\lambda_s, \lambda_d)) = \sum_{\lambda_i \in path(\lambda_s, \lambda_d)} Overhead(\lambda_i) + \sum_{\phi_{ij} \in path(\lambda_s, \lambda_d)} Overhead(\phi_{ij})$$

$$Delivery\_Ratio(\lambda_i) = \frac{Receive(\lambda_i)}{Forward(\lambda_i)} \times 100\%$$

Definition 4. (Objective Function)

The multi-objective function of Ad Hoc networks is defined by,

$$\begin{cases} f_1 = \min(Overhead(path(\lambda_s, \lambda_d))) \\ f_2 = \max(\sum Delivery\_Ratio(\lambda_i)) \end{cases}$$

While, with the constraints,

$$\begin{cases} Forward(\lambda_i) < Bandwidth(\lambda_i) \\ if path(\lambda_i, \lambda_j) exists,\ path(\lambda_i, \lambda_j) < Range(\lambda_i) \end{cases}$$

Set electronic communication network as an example, as shown in Figure 6. $|\Lambda| = 9$ and $|\Phi| = 10$ represent the number of devices and links. $\lambda_1 =< S, 100, 200 >$, $\lambda_2 =< D, 80, 150 >$ and $\lambda_8 =< D, 30, 80 >$ represent the property of devices. Device 1 is a typical static device in DC Level

1 and Device 2-5 are in DC Level 2, which perform a random moving in a certain range. Device 6-9 perform a random moving without certain range or in a direction with some velocity, namely, in DC Level 3 and DC Level 4.

5 SGSO FOR AD HOC NETWORK

5.1 Initialization: Discrete SGSO for Networks

We defined a weighted graph G=(V, E) as the Ad Hoc Network, where V is the set of nodes and E is the set of path between nodes. There are two initial matrixes representing the statu of network. Matrix $M_0 = [m_{ij}]$ represents the connection statu of wireless network. $m_{ij} = \{-1, 1\}$, where $m_{ij} = -1$ means that node j isn't inside the transmission range of Node i, and $m_{ij} = -1$, on the contrary, means that information can be delivered from Node i to Node j. The second matrix is overhead matrix $M_{overhead} = [m_{ij}]$, where $m_{ij} = c(i \neq j)$ represents the overhead from Node i to Node j and $m_{ij} = c(i = j)$ represents the overhead of Node i.

Let a two-dimensional adjacent 0-1 matrix X ($N \times N$) represents the solution of network route or path. The value of x_{ij} is either 0 or 1 where $x_{ij} = 1$ represents that there is a path between Node i and Node j, while $x_{ij} = 0$ represents Node i and Node j are not connected.

There should be only one element's value equals 1 in a row as the path is unique when we just consider the unicast route. If the number of nodes is N, the particles will move towards the origin on a N-dimensional space during the optimization process, where x_{ij} is a coordinate of the N-dimensional space.

First, initialize M particles in the N-dimensional space randomly. Each particle is a path and is a N-dimensional vector $X_i = (X_{i1}, X_{i2}, X_{i3}, \cdots, X_{iN})$. The $N \times N$ matrix X is also the position of particles in searching space. Secondly, improve GSO algorithm to 0-1 discrete model by Sigmoid function. According the feature of producer, here, we turn into value between 0 and 1, by

$$sig(r_1 l_{\max} D_p^k(\varphi^k)) = \frac{1}{1 + \exp(-r_1 l_{\max} D_p^k(\varphi^k))}$$

Thus we obtain the improved function of producer,

$$X_z = X_p^k + sig(r_1 l_{\max} D_p^k(\varphi^k))$$

$$X_r = X_p^k + sig(r_1 l_{\max} D_p^k(\varphi^k + r_2\theta_{\max}/2)))$$

$$X_l = X_p^k + sig(r_1 l_{\max} D_p^k(\varphi^k - r_2\theta_{\max}/2)))$$

Set the maximum number of every row 1 and others 0. Similarly, the same strategy is used for scrounges and dispersed members by Sigmoid function. The characters are resigned in each iteration. For example, supposing a network with 5 nodes, we obtain a solution matrix $X = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ after n iterations. $x_{12} = 1$ means there is a path from Node 1 to Node 2, and then we will check Row 2 to find out the path from Node 2 to next node. Path $1- >4- >3- >5$ thus is the available path from Node 1 to Node 5.

5.2 Solutions: under DC Levels

Ad Hoc network has been classified under four levels according to their dynamic couple levels in this paper. Therefore, we propose route approaches based on this level category using SGSO algorithm.

For *DC Level 1*, an on-table route is more appropriate since the table is barely changed. Even though, the network scale should also be considered. If the scale is relative small with N nodes, S $N \times N$ Matrix X_i are initialized. The objective function is static as well, by,

$$f_1 = \min(Overhead(path(\lambda_s, \lambda_d)))$$

$$\begin{aligned} &Overhead(path(\lambda_s, \lambda_d))) \\ &= \sum_{i,j \in node} (Overhead_{node(i)} + Overhead_{path(ij)}) \\ &= \sum_{i,j \in node} (M_{overhead} \times X_{ij}) \end{aligned}$$

Under the iterations of SGSO algorithm, Route solution Matrix X_i for Node i is obtained. Otherwise, if the network scale is large, the building of static table will take too much time. We prefer another local routing approach, which will be introduced in DC Level 4.

The key access to *DC Level 2* is the relationship between delivery range R and moving range r. We define that $R = \hbar r$, where $\hbar$ indicates coefficient between them. If $\hbar > 10$, assume that R is big enough to use the static table. Thus, we revise the delivery range to (R-2r) and continue the method of DC Level 1. While if $1 < \hbar < 10$, we prefer the dynamic part of SGSO to build route table. Namely, we change the objective function dynamic when the node moves. By this means, f_1 changed by the change of $Overhead_{path(ij)}$. Otherwise, if $\hbar < 1$ means that R is less than r and the moving range is so large that we classify this condition into DC Level 4.

DC Level 3 is similar with DC Level 2 in some degree, only with the different objective function. The Overhead of path change with a speed of v, thus, we define that

$$Overhead'_{path(ij)} = Overhead_{path(ij)} + vt.$$

By this way, we obtain a real-time route table for this node changing with time and iteration.

Finally, it comes to *DC Level 4*, the dramatic dynamic network. We introduce a new approach called Local routing table inspired by the theory of Six Degrees of Separation. Local routing table is built by combining the demand-driven routing algorithm and the table-driven routing algorithm. For each device, a fixed routing table with M (M>0) level is built where M is the number of route for least cost from this device to the destination node. This method is designed particular for large scale and high dynamic network.

The value of M is relevant to the total amount of nodes. Local routing table contains the node itself when M=1 and all the nodes when M=N. We will discuss the proper relationship between M and N in the simulation.

Supposing that X_{1n} is the shortest path from Node 1 to Node n through n hops. If Node 1' s routing table contains Node n, we will get the optimal solution easily. Otherwise search the routing table of *M*th hop, thus, the complex multi-hop searching problem is simplified into a combination of several groups

$$X_{1n} = \sum_{i=1}^{n-1} X_{i(i+1)}$$

If this path is through Node i and Node j, then

$$X_{1n} = X_{1i} + X_{ij} + X_{jn}$$

in this condition, minimizing X_{1i}, X_{ij} and X_{jn} can get the minimum of X_{1n}

$$X_{1n\min} = X_{1i\,\min} + X_{ij\,\min} + X_{jn\,\min}$$

Thus, we can get the best path by comparing the cost of these few group.

5.3 Efficiency: Decision Behaviors for QoS

According to Leader Model, individuals in the group whether to be leaders or followers depends on its decision factor λ. It's the same with Ad Hoc network. There are N nodes in the network with different range and bandwidth, as shown in Figure 6. We regard every node of the group as a individual in the leader model with a decision factor λ. Therefore, the decision factor will affect the transform decision by affecting the node's cost.

If λ=1, means this node is a leader and λ=0 means it's a follower. In the Ad Hoc network, the leader node is available and the follower node is saturation. When $0 < \lambda < 1$, the individuals can either choose their preferred option or copy the previous action of the other player depending on the value of λ. That means that this node has forwarded information but hasn't arrived at its bandwidth. Since the node has been occupied, the package loss may be higher than other path.

What makes a leader in the group? It is different from group to group. The decision factor of Ad Hoc network depends on the relationship between bandwidth and forwarding information, by,

$$\lambda_i = \cos(\frac{\pi}{2} \times \frac{x}{\min(bandwidth_i)})$$

Then the cost of the node,

$$node\_\cos t_i{}' = \frac{1}{\lambda_i} node\_\cos t_i$$

The objective function, by,

$$f = \sum (x_{ij}\, path\_overhead_{ij} + \frac{1}{\lambda_i} node\_overhead_i + \frac{1}{\lambda_j} node\_overhead_j)$$

6 MODEL ANALYSIS AND PROOFS

Convergence Theorem give a benchmark to verify the convergence of algorithms. To prove the convergence and correctness of our algorithm, we analyze the search space and function of producer and scroungers respectively according to the rules of Convergence Theorem.

Convergence Theorem: *Supposing that A is an algorithm on X and* Ω *is the set of solution, the initial point is given* $x^{(1)} \in X$ *, and iterations are as follows:*

If $x^{(k)} \in \Omega$ *, the iteration is finished; or, set* $x^{(k+1)} \in A(x^{(k)})$ *.*

Let k+1 substitutes k, and repeat the above process. Thus, we obtain the sequence $\{x(k)\}$*.*

Then set that

1. *Sequence* $\{x(k)\}$ *is contained in compact subset X;*

2. *There exists a continuous function, which is a decreasing function of* Ω *and A;*
3. *Mapping A is closed on the complement of* Ω.

The limitation of any convergent subsequence of sequence $\{x(k)\}$ *belongs to* Ω.

Lemma 1. *Supposing that the objective function's searching space is U and the producer's is P. If X represents the producer's location, then* $\forall X \in U \Rightarrow X \in P$.

Proof. As we known, the producer's searching methods,

$$\begin{aligned}
X_z &= X_p^k + r_1 l_{\max} D_p^k(\varphi^k) \\
&= X_p^k + r_1 \sqrt{\sum_{i=1}^{n} (U_i - L_i)^2} \sin(\varphi_{i(j-1)}^k) \prod_{q=j}^{n-1} \cos(\varphi_{iq}^k) \\
X_r &= X_p^k + r_1 l_{\max} D_p^k(\varphi^k + r_2 \theta_{\max}/2) \\
&= X_p^k + r_1 \sqrt{\sum_{i=1}^{n} (U_i - L_i)^2} \sin(\varphi_{i(j-1)}^k) \prod_{q=j}^{n-1} \cos(\varphi_{iq}^k + r_2 \theta_{\max}/2) \\
X_l &= X_p^k - r_1 l_{\max} D_p^k(\varphi^k - r_2 \theta_{\max}/2) \\
&= X_p^k + r_1 \sqrt{\sum_{i=1}^{n} (U_i - L_i)^2} \sin(\varphi_{i(j-1)}^k) \prod_{q=j}^{n-1} \cos(\varphi_{iq}^k - r_2 \theta_{\max}/2)
\end{aligned}$$

In an n-dimensional search space, the ith member at the kth searching iteration has a current position $x_i^k \in R^n$, a head angle $\varphi_i^k = (\varphi_{i1}^k, ..., \varphi_{i(n-1)}^k) \in R^{n-1}$. The search direction of the ith member, which is a unit vector $D_i^k(\varphi_i^k) = (d_{i1}^k, ..., d_{in}^k) \in R^n$ that can be calculated from φ_i^k via a polar to Cartesian coordinate transformation

$$d_{i1}^k = \prod_{q=j}^{n-1} \cos(\varphi_{iq}^k)$$

$$d_{ij}^k = \sin(\varphi_{i(j-1)}^k) \prod_{q=j}^{n-1} \cos(\varphi_{iq}^k)(j = 2, ...n - 1)$$

$$d_{in}^k = \sin(\varphi_{i(n-1)}^k)$$

$$l_{\max} = ||U - L|| = \sqrt{\sum_{i=1}^{n} (U_i - L_i)^2}$$

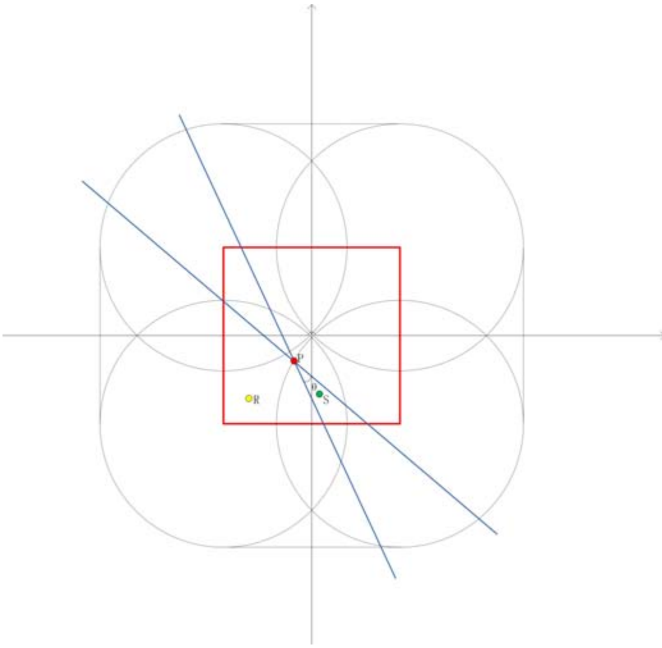


FIGURE 7
GSO's searching space in 2-dimension

L_i and U_i are lower and upper bound respectively, thus, $l_{\max}$ is the distance between the two most distant nodes in the searching space.

That means the producer's searching space is a sphere whose center is X_p^k and radius is $\sqrt{\sum_{i=1}^{n}(U_i - L_i)^2}$. An example of 2-dimension searching space is shown in Figure 7. Rectangular area ((-a, a), (-b, b)) is the objective function's searching space. The area in the dotted line is the producer's searching space, which is composed of four circles.

It turns out that $P \subset U$, thus, $\forall X \in U \Rightarrow X \in P$.

Lemma 2. *The value of producer's objective function decreases in every iteration by GSO, namely, if $Px(t)$ is the value of objective function, thus, $f(Px(t+1)) \leq f(Px(t))$.*

Proof. Supposing that there are only producers in the group, scourges and dispersed members are omitted. According to producers' methods,

$$Px(t+1) = \min(Px(t), X_z, X_r, X_l)$$

that means,

$$f(Px(t+1)) \leq f(Px(t))$$

Thus it turns out that, the producer's value of objective function decreases monotonically.

Lemma 3. *The scrounges converges to the producers, namely, $x_p \in Px, x_s \in Ps, x_s \to x_p$ stands for converging).*

Proof. Here, we set the dimension n = 1 to simplify the problem.

$$x(k+1) = x(k) + r_3 \times (x_p - x(k))$$

Since the movement of particles is a second-order difference equation, then the Z-transform leads to,

$$zY(z) - zy(0) = (1 - r_3)Y(z)$$

that is,

$$z - 1 + r_3 = 0$$

According to the Routh's Stability Criterion, necessary and sufficient condition for a second-order linear system is a condition that all the coefficients of characteristic equation are positive. With the result, we will then derive the condition to guarantee the stability of a difference equation (when the system continuous cyclical oscillation with the same amplitude and limited speed, it can be considered as critical stability.)

$$0 < r_3 < 2$$

When Equation (34) is satisfied and according to Z-transform final-value theorem,

$$x(k) = \lim_{z \to 1} ((z-1) \cdot X(z)) = x_p$$

That means scourges will converges to producer when $0 < r_3 < 2$.

Theorem 1. *GSO algorithm's searching space is global and the value of objective function decreases in every iteration.*

Proof.

1. Producer's searching space is global according to Lemma 1, and scourges search around the producer in angel θ. Meanwhile, dispersed members, by,

$$X_i^{k+1} = X_i^k + l_i D_i^k(\varphi^{k+1})$$
$$X_i^{k+1} = X_i^k + a \cdot r_1 l_{\max} \sin(\varphi_{i(j-1)}^{k+1}) \prod_{q=j}^{n-1} \cos(\varphi_{iq}^{k+1} - r_2\theta_{\max}/2)$$

   Dispersed members move randomly by angel φ^{k+1}, which is the optimal result in this iteration. The purpose of dispersed members is avoiding local optimum.
2. The value of producers' objective function decreases according to Lemma 2. When an iteration is finished,

$$X_{p+1} = \min\{X_1, X_2, ..., X_n\}a$$

   which means,

$$X_{p+1} \leq X_p$$

   In summary, the particle's searching space in GSO is global and the value of objective function decreases, namely, the correctness of GSO algorithm is verified and GSO can be applied to solving space-searching problem.

Theorem 2. *GSO algorithm is a global convergent algorithm.*

Proof.

1. According to Theorem 1, GSO algorithm's searching space is global and the value of objective function decreases in every iteration. Supposing that Sequence $\{x(k)\}$ is the solutions we obtained, thus it must be in the global searching space. Therefore, Sequence $\{x(k)\}$ is contained in compact subset X.
2. As we've analyzed, the function of producer and scroungers are both decreasing with the iteration. Furthermore, Dispersed members who perform random walk motions are the 20% worst performers in iteration. They can find out new position when producer is in local optimum. Therefore, there exists a continuous function, which is a decreasing function of Ω and A;

3. This strategy is employed by GSO to handle the bounded search space: when a member is outside the search space, it will turn back into the search space by setting the variables that violated bounds to its previous values. With the searching strategy and global searching space, mapping A is closed on the complement of Ω.

In summary, based on the Convergence Theorem, GSO algorithm is a global convergent algorithm. The conclusion is straightforward.

7 SIMULATIONS

Here, we give the experimental results, which serve four purposes. First, simulate the mobile Ad Hoc network satisfying QoS demands in different DC Levels. Secondly, using SGSO routing algorithm to build route table and compare delay with other classical algorithms. Thirdly, compare the performance in overhead of GSO and SGSO algorithm when building the dynamic route table to highlight the effectiveness of our algorithm. Finally, set simple examples to illustrate the improvement of local routing table and decision factor.

All the experiments presented in this section are completed on Windows 7, Visual Studio 2010 C# Windows Form.

7.1 Dynamic Networks

Since we've already divided the Ad Hoc networks into 4 levels according to their dynamic coupling levels and analyzed solutions on different situations, the simulations are proceeded and discussed under different levels.

DC Level 1: Robust global routing table is built by SGSO in DC Level 1 effectively. To verify the effectiveness and efficiency of SGSO routing algorithm experimentally, we compared the results by SGSO with the other classic algorithm, including the AODV, Genetic Algorithm (GA) [18] and Ant Colony Optimization algorithm (ACO) [19]. The end-to-end delay and delivery ratio shown in Figure 8 and Figure 9 illustrates the performance of our algorithm clearly. We can also draw the conclusion that the SGSO routing algorithm has better stability and higher efficiency than other algorithms.

DC Level 2 and Level 3: The experiments on DC Level 2 and Level 3 focus on the overhead of Ad Hoc networks when the structure of network changes during transmission.

Figure 10 shows the result of overheads of routing table by GSO and SGSO, respectively. The improvement performance is definitely clear. When

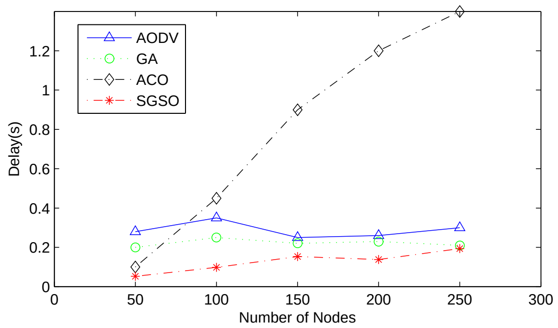


FIGURE 8
Number of Nodes vs Delay

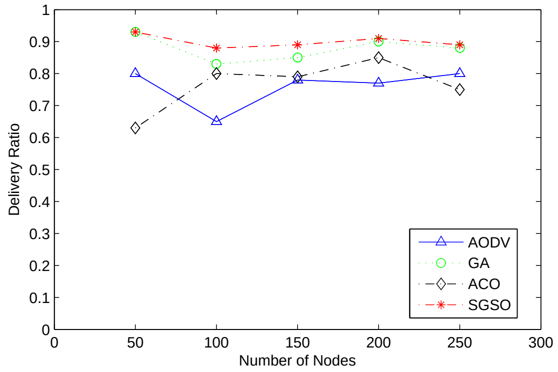


FIGURE 9
Number of Nodes vs Delivery Ratio

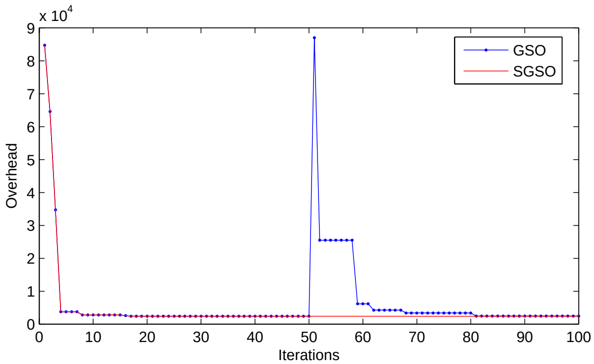


FIGURE 10
Overhead of routing table by GSO and SGSO

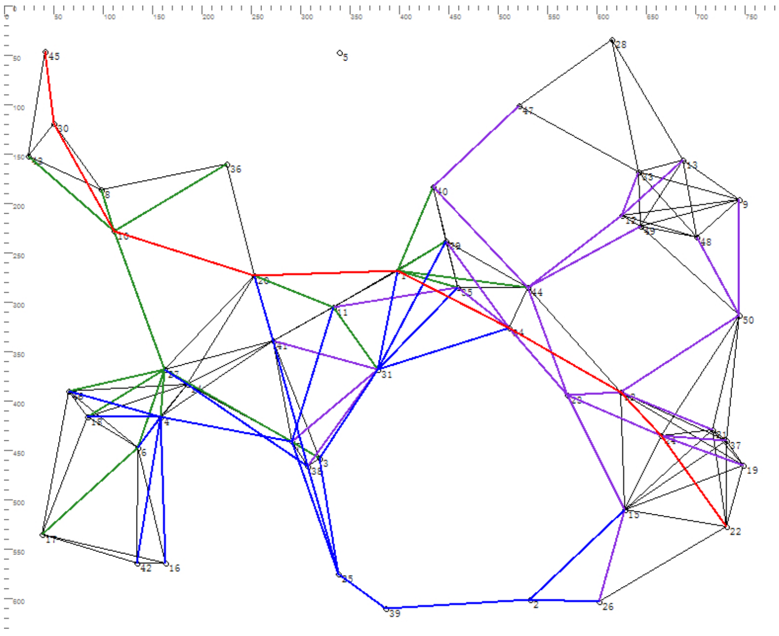


FIGURE 11
Local Routing and Transmission

the device moved in iteration 50, the original GSO algorithm searched from the beginning, while SGSO algorithm adapted to the new condition immediately.

DC Level 4: As for the ruleless moving in DC Level 4, local routing table is built. As shown in Figure 11, we built 3-level routing tables of 50 devices. Set Device 10, 23 and 25 as examples, local tables are highlight in the figure, which shows that their edges are linked. A transmission from device 45 to device 22 goes through 2 routing tables.

7.2 Decision Factor

Initialize 20 nodes in three device types. Device A contains Node 1, 2, 3 and Node 7, Node 4, 5, 6, 8, 9, 10, 11, 12 belong to Device B and the rest are Device C. The node's range and bandwidth are shown in Table 2.

We compare the nodes' information including bandwidth, decision factor, delivery information, node's overhead and path's overhead between network with decision factor and the one without decision factor. Figure 7 and Figure 8 are QoS-guaranteed Ad Hoc network experimental results with decision

Device	Range	Bandwidth	Nodes
A	180	200	1, 2, 3, 7
B	100	150	4, 5, 6, 8, 9, 10, 11, 12
C	50	100	13, 14, 15, 16, 17, 18, 19, 20

TABLE 3
Ranges and Bandwidth

	With Decision Factor					Without Decision Factor			
Sending	Receive	Sending	Delivery Ratio	D.F.	Overhead	Receive	Sending	Delivery Ratio	Overhead
10	10	10	100%	0.988	1	10	10	100%	1
30	30	30	100%	0.951	1.01	30	30	100%	1
50	50	50	100%	0.809	1.05	50	50	100%	1
70	70	70	100%	0.588	1.23	70	70	100%	1
90	90	90	100%	0.156	4.58	90	90	100%	1
95	95	95	100%	0.078	6.39	95	95	100%	1
96	96	96	100%	0.060	12.75	96	96	100%	1
98	96	96	100%	0.060	12.75	98	98	100%	1
100	96	96	100%	0.060	12.75	100	100	100%	1
110	96	96	100%	0.060	12.75	110	100	90.9%	1

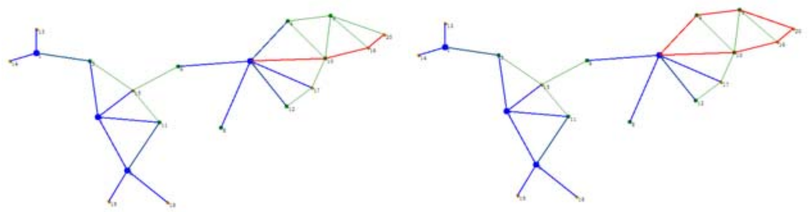
TABLE 4
Decision Factor Comparisons

factor and experimental data is in Table 3. By Node 16, when the amount of sending information is 50, decision factor is 0.7, and the node's overhead is 1, which is far away from affecting delivery. However, when the amount arrives at 50, decision factor is 0.06, which makes the node's overhead increasing sharply. For this reason, Node 9 will be chosen to replace Node 16 because of lower overhead even though Node 16's bandwidth hasn't arrived. It also avoids packet loss from overloading and increases delivery ratio.

Node 16's information is shown in Table 3. As shown in Table 3, decision factor's effectiveness is obvious only when the information forwarding arrive at 90%. Decision factor's sharply decreasing increase the node's overhead, which makes sure that its forwarding information is less than its bandwidth and forwarding rate is 100%. On the contrary, the nodes will stop forwarding information only after that's arrival at bandwidth, which decreases the forwarding rate and increase package loss rate.

8 CONCLUSION

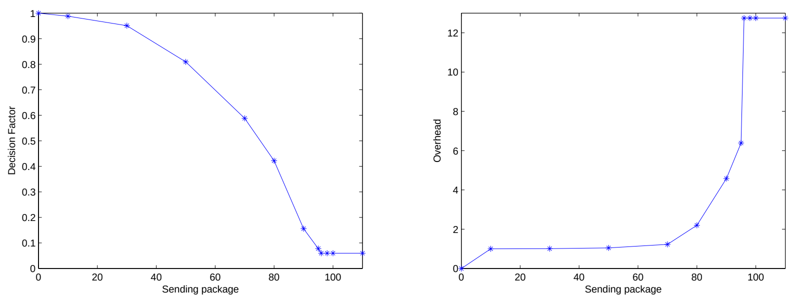
Mobile Ad Hoc networks are spontaneously deployed over a geographically limited area without well-established infrastructure. The networks work well



(a) 50 Packages (b) 98 Packages

FIGURE 12
Package Delivery from Node 3 to Node 20 with decision factor

only with the specific routing algorithm designing for dynamic structure and satisfying Quality of Service. A novel Social Group Search Optimizer algorithm by improving the GSO algorithm to a dynamic and discrete algorithm through the introducing of social behaviors. SGSO is divided into search and prey parts, where “search” is on duty to find the optimal solution effectively and “prey” is responsible for adjust the dynamic change of objective functions. SGSO has the ability to perform large-scale distributed parallel optimization and a powerful processing ability in a complex, high-dimensional and dynamic real-time changing environment. Dynamic Coupling Levels promote the routing algorithm based on SGSO to solve Ad Hoc network problems more particularly and effectively. To apply the SGSO into Ad Hoc network, decision factor and local routing table are also introduced to SGSO routing algorithm and successful performance is obtained, especially for high level dynamic networks. The convergence and correctness of our algorithm are verified mathematically and extensive experiments have been conducted



(a) Decision Factor (b) Overhead

FIGURE 13
Decision Factor and Node Overhead of Node 16

to evaluate the efficiency and effectiveness of the proposed mechanism in mobile Ad Hoc networks. The results show that SGSO improves packet delivery ratio and reduces average end-to-end latency effectively.

However, more practical simulations based on real data in life are still needed in future work. As a distributed and parallel algorithm, internal and public behaviors of SGSO should be arranged more clearly to elevate parallelism and reduce the executing time. Moreover, dynamic feature of SGSO can be improved to apply in more dynamic applications.

ACKNOWLEDGEMENT

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

[1] Giacomo G D, Patrizi F, Sardina S. Automatic behavior composition synthesis. *Artificial Intelligence*, 2013.3: 106–142.

[2] Yang R, Kiekintveld C, etal. Improving resource allocation strategies against human adversaries in security games: An extended study. *Artificial Intelligence*, 2013.2: 440–469.

[3] Rintanen J. Planning as satisfiability: Heuristics. *Artificial Intelligence*, 2012.12: 45–86.

[4] Junior L S, Nedjah N, de Macedo Mourelle L. Routing for applications in NoC using ACO-based algorithms. *Applied Soft Computing*, 2013.5: 2224–2231.

[5] Lin C C. Dynamic router node placement in wireless mesh networks: A PSO approach with constriction coefficient and its convergence analysis. *Information Sciences*, 2013.5: 294–308.

[6] Feng X, Lau F, Yu H. A Novel Bio-inspired Approach Based on the Behavior of Mosquitoes. *Information Sciences*, 2013.6: 87–108.

[7] Askarzadeh A, Rezazadeh A. Artificial bee swarm optimization algorithm for parameters identification of solar cell models. *Applied Energy*, 2013.2: 943–949.

[8] S. He, Q. H. Wu, and J. R. Saunders. Group search optimizer– an optimization algorithm inspired by animal searching behavior.*Evolutionary Computation*, 2009.10: 973–991.

[9] Zhang L, Luo W, Chen S, et al. End-to-end maxmin fairness in multihop wireless networks: Theory and protocol. *Journal of Parallel and Distributed Computing*, 2012.72: 462–474.

[10] Gui J, Liu A. A new distributed topology control algorithm based on optimization of delay and energy in wireless networks. *Journal of Parallel and Distributed Computing*, 2012.8: 1032–1044.

[11] Shah S C, Park M S. An effective and robust two-phase resource allocation scheme for interdependent tasks in mobile ad hoc computational Grids. *Journal of Parallel and Distributed Computing*, 2012.12: 1664–1679.

[12] Kumar A, Shin K G. DSASync: managing end-to-end connections in dynamic spectrum access wireless LANs[J]. *IEEE/ACM Transactions on Networking* , 2012.4: 1068–1081.

[13] Jaramillo J J, Srikant R. Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic.//*INFOCOM, 2010 Proceedings IEEE. IEEE*, 2010: 1–9.

[14] Kyungjun Kim, A distributed channel assignment control for QoS support in mobile ad hoc networks. *Parallel and Distributed Computing*, 2011.3: 335–342.

[15] Vaze R. Throughput-delay-reliability tradeoff with ARQ in wireless ad hoc networks. *Wireless Communications, IEEE Transactions on*, 2011.10: 2142–2149.

[16] Tavli B, Heinzelman W. Energy-efficient real-time multicast routing in mobile ad hoc networks. *Computers, IEEE Transactions on*, 2011.60: 707–722.

[17] Xia H, Jia Z, Li X, et al. Trust prediction and trust-based source routing in mobile ad hoc networks. Ad Hoc Networks, 2012.2. http://dx.doi.org/10.1016/j.adhoc.2012.02.009

[18] Baboo S S, Narasimhan B. Genetic Algorithm Based Congestion Aware Routing Protocol (GA-CARP) for Mobile Ad Hoc Networks. *Procedia Technology*. 2012.4: 177–181.

[19] Wang J, Osagie E, Thulasiraman P, et al. HOPNET: A hybrid ant colony optimization routing algorithm for mobile ad hoc network. *Ad Hoc Networks*. 2009.7(4): 690–705.