

Research Article

Multisensors Cooperative Detection Task Scheduling Algorithm Based on Hybrid Task Decomposition and MBPSO

Changyun Liu,¹ Xiangke Guo,^{1,2} Zhihui Li,¹ Yingying Wang,¹ and Gang Wei¹

¹*Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China*

²*College of Electronic and Information Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191, China*

Correspondence should be addressed to Changyun Liu; icemissile@sina.com

Received 14 July 2016; Revised 8 October 2016; Accepted 26 October 2016; Published 12 February 2017

Academic Editor: Cheng-Tang Wu

Copyright © 2017 Changyun Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A multisensor scheduling algorithm based on the hybrid task decomposition and modified binary particle swarm optimization (MBPSO) is proposed. Firstly, aiming at the complex relationship between sensor resources and tasks, a hybrid task decomposition method is presented, and the resource scheduling problem is decomposed into subtasks; then the sensor resource scheduling problem is changed into the match problem of sensors and subtasks. Secondly, the resource match optimization model based on the sensor resources and tasks is established, which considers several factors, such as the target priority, detecting benefit, handover times, and resource load. Finally, MBPSO algorithm is proposed to solve the match optimization model effectively, which is based on the improved updating means of particle's velocity and position through the doubt factor and modified Sigmoid function. The experimental results show that the proposed algorithm is better in terms of convergence velocity, searching capability, solution accuracy, and efficiency.

1. Introduction

The early warning detection system of the ballistic missile defense system is composed of the space-based satellite, the ground radar, the sea-based radar, and so on. Due to the early warning equipment deployment and sensor detecting capability, multitactical ballistic missile (TBM) emerging randomly cannot be tracked by a single early warning equipment stably and continuously in the entire trajectory of TBM.

The goal of the multisensors cooperative detection is to ascertain the detection sequence and the detection time about TBM under the constraint of limited sensor resources and visualization period and to achieve the optimized match of sensor resources and detecting tasks, so that the optimum detecting profit is obtained.

Many researchers have done lots of researches on the resource scheduling of the early warning satellites in depth. In [1, 2], the parallel tabu genetic algorithm and improved particle swarm optimization algorithm are used, respectively, to resolve early warning satellites scheduling problems. A variable neighborhood heuristic algorithm for the dynamic

task planning of the Low Earth Orbit warning system is studied in [3]. The task decomposition method for early warning satellites scheduling is studied in [4, 5]. The above researches on the early warning satellites task scheduling is focused on the method of numerical solution and task decomposition, while the sensor resources schedule in BMDS should be the combined schedule containing the early warning satellites and ground radars and so on. Aiming at antimissile resource scheduling, the multitarget optimization model including many factors is established in [6], but the task decomposition method is different from the practical problem and the difference between early warning satellites and radars is not reflected in its scheduling model.

Sensor resources scheduling is a nonlinear multitarget combination optimization problem in essence. Because of the high velocity of TBM, it is vital to resolve the sensor scheduling problem in real time. Particle swarm optimization (PSO) [7–9] is easy and effective to use, so it has been widely applied in the sensor resources scheduling and the weapon-target allocation (WTA) [10]. Aiming at the optimization problems of the binary representations solution, BPSO is

proposed by Kennedy and Eberhart in [11]. And many improved BPSO algorithms aiming at the shortcomings of BPSO are presented in literature [12–15]. Among them, BPSO algorithm is modified in generation mechanism, update mechanism, and selection mechanism for multiobjective resource allocation problem in [13]. And a memetic binary particle swarm optimization algorithm is introduced in [15] to improve the local and global search performance.

Multisensors task scheduling is not only a nonlinear multitarget combination optimization problem, but also a real-time algorithm, so a multisensors task scheduling algorithm based on hybrid task decomposition and modified binary particle swarm optimization (MBPSO) is proposed. Aiming at the complex relationship between sensor resources and tasks, a hybrid task decomposition method is presented, so that the resource scheduling problem is decomposed into subtasks, and then the sensor resource scheduling problem is changed into the match problem of sensors and subtasks. Through the analysis of subtasks, the resource match optimization model based on the sensor resources and tasks is established, which contains several factors, such as the target priority, detecting benefit, handover times, and resource load. For the purpose of global optimization and real-time calculation, a match optimization model for MBPSO algorithm is proposed, which is based on the improved updating means of particle's through the doubt factor and modified Sigmoid function, multisensors cooperative detecting tasks optimization, and real-time solving are realized.

2. Cooperative Detecting Task Decomposition Based on Hybrid Task Decomposition

Antimissile multisensors cooperative detecting task has the following characteristics:

- (1) There are many different kinds of sensor resources in the ballistic missile defense system, so the visualization relationship between sensors and targets is complex when there are many TBM.
- (2) The flight range of the ballistic target is large, and it is easy to be out of the visual field of one sensor and enters into the visual field of another sensor.
- (3) The ballistic target has a low possibility to maneuver in the midcourse flight, so its motion trajectory can be predicted precisely.
- (4) The emergence time, position, number, and type of ballistic targets are difficult to predict, so the cooperative detecting tasks are random and dynamic.
- (5) The threat degree of multiple ballistic targets is different, so the priority of cooperative detection tasks is also different.

It can be concluded from the above task characteristics that one single cooperative detection task cannot be involved independently in the scheduling, the cooperative detection task should be decomposed into subtasks, and these subtasks can be used in scheduling.

2.1. Task Decomposition's Requirements. Through the task decomposition, the multisensors cooperative detecting tasks are decomposed into many basic and independent subtasks, which cannot be decomposed any longer. Some requirements to subtasks are as follows:

- (1) Completeness: cooperative detecting tasks are decomposed into many subtasks, and the sum of subtasks is equal to the cooperative detecting tasks.
- (2) Independence: the subtasks cannot be replaced by each other.
- (3) Granularity: the length of subtasks should be moderate. If the length of the subtasks is too long, it will occupy a sensor resource for a long time, which does not meet the requirement for optimization. If the length of subtasks is too short, it will not meet the requirements of estimation and prediction, which improves the complexity of optimization. Assume that the longest subtask is D_{\max} and the shortest is D_{\min} .
- (4) Interval: a certain interval is allowed among subtasks. If the interval is too long, it increases the trajectory prediction error and is more difficult for sensors to capture the target in the next subtask's time. So the interval cannot exceed the upper limit value.

2.2. Method of Hybrid Task Decomposition. The basic methods [4–6] of task decomposition include “the longest observation time” decomposition method, “evenly split” decomposition method, and “start-stop time” decomposition method. The characteristics of these methods are mainly as follows:

- (1) “The longest observation time” decomposition method: a sensor will be occupied for a long time, which not only decreases the detection efficiency of sensor resources but also increases the resource conflict.
- (2) “Evenly split” decomposition method: it is difficult to determine the length of subtasks. If the length of subtasks is too long, some tasks will not be completed by proper sensors, and if the length of subtasks is too short, the handover of sensors will be frequent.
- (3) “Start-stop time” decomposition method: a sensor detects the same target for a long time, which reduces the detection efficiency, and the number of subtasks would increase significantly with the increasing number of targets.

Due to the characteristics of the above task decomposition methods, the subtasks in a single task decomposition method would conflict each other, which results in the local optimization of resource scheduling, so the single task decomposition method cannot meet the requirement of task decomposition. Then combining “evenly split” decomposition method with “start-stop time” decomposition method, a hybrid task decomposition method for multisensors detecting multitarget is designed. Its main idea is as follows: firstly, using the “start-stop time” decomposition method, the whole task is decomposed into many subtasks according to the visualization start time and end time of all sensors

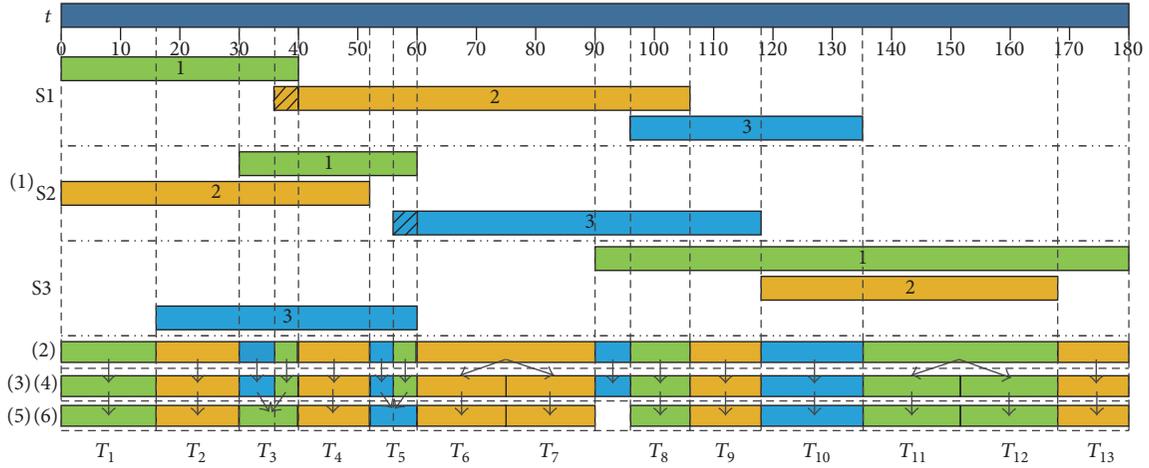


FIGURE 1: Process of hybrid task decomposition.

to all targets; secondly, using “evenly split” decomposition method, the subtasks are selected according to the constraint condition, if the length of subtasks is too long, it will continue to be split again, and if the length of subtasks is too short, it will be integrated. The specific diagram of hybrid task decomposition method is shown in Figure 1.

In Figure 1, S1 denotes sensor 1, S2 denotes sensor 2, S3 denotes sensor 2, and T1 to T13 denote subtask 1 to subtask 13, respectively. Green-1, orange-2, and blue-3 denote the interval of time; that is, target-1, target-2, and target-3 are in the detecting area of different sensor, respectively. For example, the green-1 in the area of S1 means that the interval of time of the target-1 in the sensor 1 detection area is from zero seconds to forty seconds, which denotes $[0, 40]$. The hybrid task decomposition is the basis of the match optimization model.

The detailed steps of hybrid task decomposition method are as follows.

Step 1 (calculating). According to the visual relationship between all sensors and all ballistic targets, the visualization interval of sensors and targets is determined.

Step 2 (sorting). The start time and end time in visualization interval of all sensors to all targets are marked, and the marked time is sorted in time from small to large, and then the two adjacent moments are synthesized into an interval; as a result, the whole tasks are decomposed into subtasks.

Step 3 (selecting). According to the third constraint condition of task decomposition, the subtasks between D_{\max} and D_{\min} are selected. If the length of subtasks is longer than D_{\max} , go to Step 4.

Step 4 (splitting). If the length of subtasks is longer than D_{\max} , the subtasks would be split continually, so the length of subtasks is in $[D_{\min}, D_{\max}]$. It is supposed that the length of subtasks is M , the task is split into N subtasks whose length

is D_z , the detailed method is as follows:

$$N = \begin{cases} \lceil M/D_{\min} \rceil & \text{if } M \bmod D_{\min} \neq 0 \\ M/D_{\min} & \text{if } M \bmod D_{\min} = 0 \end{cases} \quad (1)$$

$$D_z = M/N,$$

where $\lceil M/D_{\min} \rceil$ denotes the largest integer beyond M/D_{\min} .

Step 5. If the length of subtasks is less than D_{\min} , the processing step is as follows: (1) if the subtask is in existence independently, namely, the subtask length of the previous, and next subtask exceeds D_{\min} , the subtask would be deleted; (2) if the number of adjacent subtasks is more than two, it is supposed that the start time and the end time of i th subtask are t_{start}^i and t_{end}^i , respectively, and the time of the integrated subtask is t_{start} and t_{end} , respectively. So the integration method is as follows:

- (1) initialize: $t_{\text{start}} = t_{\text{start}}^1$, $i = 2$;
- (2) if $(t_{\text{end}}^i - t_{\text{start}}) \notin [D_{\min}, D_{\max}]$, go to (3); otherwise, go to (4);
- (3) $i = ++$, go to (2);
- (4) $k = i$, the k adjacent subtasks are integrated, and the visual interval of the integrated subtask whose length is less than D_{\min} would be deleted, as it be shown in slash part of Figure 1.

Step 6 (deleting). After subtasks are processed by Steps 4 and 5, the subtasks whose length cannot meet the third constraint condition is deleted.

3. Match Optimization Model of Resources and Subtasks

After the task is decomposed, the implementation of multi-sensor cooperative detecting task is transformed into the

implementation of its subtasks. In the interval of each subtask, there are two relationships between sensors and targets: (1) whether the target is in the visualization interval or not; (2) if the target is in the visualization interval of sensors, whether it is detected by the corresponding sensors or not. The multitargets subtasks optimization of multisensors resource is resolved through establishing the match optimization model of sensor resources and subtasks.

3.1. Description of the Problem. In order to facilitate the description of the problem, the following definitions are given.

Definition 1 (subtask detection matrix D^k). The subtask detection matrix denotes the matrix whose component denotes whether the target can be detected by sensor or not in the subtask interval of sensor, one denotes that the target can be detected by sensor, and zero denotes that the target cannot be detected by sensor.

Definition 2 (subtask scheduling matrix E^k). The subtask scheduling matrix denotes the matrix whose component denotes whether the detecting task of sensor is implemented or not in the subtask interval of sensor at the basis of subtask detection matrix, zero denotes that the sensor is to detect target, and one denotes that the sensor is not to detect target.

It is supposed that, in $[T_s, T_e]$, the number of ballistic targets is m , in addition, the number of LEO satellites is n_1 , and the number of ground radars is n_2 in BMDS. After the task is decomposed, TN subtasks are obtained in time sequence, and the k th subtask is in $[ST^k, ET^k]$, so the subtask detection matrix D^k and subtask scheduling matrix E^k are as follows:

$$\begin{aligned} D^k &= (d_{ij}^k)_{(n_1+n_2)m}, \\ E^k &= (e_{ij}^k)_{(n_1+n_2)m}. \end{aligned} \quad (2)$$

In $[ST^k, ET^k]$, if the j th ballistic target is in the visualization interval of the i th sensor, then $d_{ij}^k = 1$; otherwise $d_{ij}^k = 0$. In like manner, in $[ST^k, ET^k]$, if the i th ballistic target is detected by the j th sensor, then $e_{ij}^k = 1$; otherwise $e_{ij}^k = 0$ and $0 \leq e_{ij}^k \leq d_{ij}^k$.

In the scheduling interval $[T_s, T_e]$, the whole detection matrix D and the whole scheduling matrix E , respectively, are

$$D = [D^1, \dots, D^k, \dots, D^{TN}] \quad (3)$$

$$E = [E^1, \dots, E^k, \dots, E^{TN}]. \quad (4)$$

So the solution to the whole scheduling problem is to determine the value of each element in the whole scheduling matrix E .

3.2. Objective Function. The goal of multisensors cooperative detecting is to detect ballistic target stably and continuously. When the cooperative detecting task is scheduled, not only the property of targets but also the feasibility of detecting task to target is considered. So when the match optimiza-

tion objective function between sensors and subtasks is established, such factors as the target priority, detecting benefit, handover times, and resource load are considered comprehensively.

(1) *Target Priority.* Pr_j denotes the priority of the j th target and $0 \leq Pr_j \leq 1$.

(2) *Detection Benefit of Sensor to Target [16].* Detection benefit includes two factors: one is the distance of target to sensor, the closer the distance between sensor and the target is, the more advantage the target is detected by sensor; the other is detection angle of sensor to target, there is the optimal detection angle, and the larger the difference between the detection angle and the best detection angle is, the poorer the detection efficiency is. The distance detection benefit A_{ij}^k and angle detection benefit P_{ij}^k of the i th sensor to the j th target in the interval of the k th subtask are defined, respectively, as follows:

$$A_{ij}^k = \frac{D_{\max}^i - \bar{D}_{ij}^k}{D_{\max}^i} \quad (5)$$

$$P_{ij}^k = \frac{\max\{\theta_{\text{opt}}^i - \bar{\theta}_{ij}^k\} - (\theta_{\text{opt}}^i - \bar{\theta}_{ij}^k)}{\max\{\theta_{\text{opt}}^i - \bar{\theta}_{ij}^k\}}, \quad (6)$$

where D_{\max}^i is the maximum detection distance of the i th sensor, \bar{D}_{ij}^k is the average detection distance of the i th sensor to the j th target in the $[ST^k, ET^k]$, θ_{opt}^i is the best detection angle of the i th sensor, and $\bar{\theta}_{ij}^k$ is the average detection angle of the i th sensor to the j th target in the $[STime^k, ETime^k]$.

The normalized distance detection benefit A and normalized angle detection benefit P are as follows:

$$A = \frac{\sum_{i=1}^{(n_1+n_2)} \sum_{j=1}^m \sum_{k=1}^{TN} A_{ij}^k \cdot e_{ij}^k}{\sum_{i=1}^{(n_1+n_2)} \sum_{j=1}^m \sum_{k=1}^{TN} d_{ij}^k} \quad (7)$$

$$P = \frac{\sum_{i=1}^{(n_1+n_2)} \sum_{j=1}^m \sum_{k=1}^{TN} P_{ij}^k \cdot e_{ij}^k}{\sum_{i=1}^{(n_1+n_2)} \sum_{j=1}^m \sum_{k=1}^{TN} d_{ij}^k},$$

where $0 \leq A \leq 1, 0 \leq P \leq 1$.

(3) *Handover Rate of Sensor.* In the scheduling interval, the larger the number of total handover is, the more frequent the handover is, which would cause the loss of target more easily. The whole resources scheduling matrix (namely, E)'s corresponding element of adjacent subtask is calculated by "XOR," and then the number of "1" is added up as the total number of handover. The handover rate in the scheduling interval is defined as

$$B = \frac{\sum_{i=1}^{(n_1+n_2)} \sum_{j=1}^m \sum_{k=1}^{TN-1} e_{ij}^k \oplus e_{ij}^{k+1}}{(n_1 + n_2) \cdot m \cdot (TN - 1)}, \quad (8)$$

where $0 \leq B \leq 1$.

(4) *The Total Load Rate of Resource* [17]. The total load rate of sensor resource is defined as the ratio of total detection time and total visualization time of sensor to target in the [Time_s, Time_e], which is marked as L .

$$L = \frac{\sum_{i=1}^{n_1+n_2} \sum_{j=1}^m \sum_{k=1}^{TN} (ET^k - ST^k) e_{ij}^k}{\sum_{i=1}^{n_1+n_2} \sum_{j=1}^m \sum_{k=1}^{TN} (ET^k - ST^k) d_{ij}^k}, \quad (9)$$

where $0 \leq L \leq 1$; the smaller L is, the lighter the load of sensor resource is.

It is known from the above analysis that the matching optimization model of resources and subtasks contains such several factors as Pr_j , A , P , B , and L . Then the objective function is established through linear weighting of these factors:

$$Z = \alpha_1 Pr_j + \alpha_2 A + \alpha_3 P - \alpha_4 B - \alpha_5 L, \quad (10)$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4$, and α_5 are the weight coefficient, which are adjusted adaptively, and $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 = 1$.

3.3. Constraint Condition Analysis. The factors of matching optimization objective function in (10) include the target, sensor resource, and subtask. In the course of optimization solution of the objective function, it must be subject to the corresponding constraint conditions. So analysis of constraint conditions is the prerequisite for the optimization solution of objective function.

- (1) The tracking target capacity constraint of sensor.

The tracking target capacity of sensors is

$$\sum_{j=1}^m e_{ij}^k \leq R_i, \quad \forall i, k, \quad (11)$$

where R_i denotes the tracking target capacity of the i th sensor.

- (2) Binary detection constraint of LEO satellites.

The TBM must be detected by two satellites at the same time; namely,

$$\sum_{i=1}^{n_1} e_{ij}^k = \{0, 2, 4, \dots\}, \quad \forall j, k. \quad (12)$$

- (3) The number constraint of sensor.

$$\left\{ r_j^{s1} \leq \sum_{i=1}^{n_1} e_{ij}^k \leq r_j^{e1} \right\} \cap \left\{ r_j^{s2} \leq \sum_{i=n_1+1}^{(n_1+n_2)} e_{ij}^k \leq r_j^{e2} \right\}, \quad (13)$$

$\forall j, k$,

where r_j^{s1} and r_j^{e1} are the lower limit and upper limit of the satellites, respectively, and r_j^{s2} and r_j^{e2} are the lower limit and upper limit of the radars, respectively.

- (4) Feasibility constraint.

$$0 \leq e_{ij}^k \leq d_{ij}^k, \quad \forall i, j, k. \quad (14)$$

- (5) Solution space constraint.

$$\begin{aligned} d_{ij}^k &= \{0, 1\} \\ e_{ij}^k &= \{0, 1\}. \end{aligned} \quad (15)$$

4. Fast Computation for Resource Matching Model Based on MBPSO

In order to realize fast computation for the match optimization model, two more steps are involved:

- (1) For the optimization process of objective function is affected by the above constraints, these five constraints should be processed further to eliminate the corresponding effect.
- (2) A modified binary particle swarm optimization (MBPSO) is devised by modifying the updates on particle's velocity and position. On one hand, the modified update on particle's velocity is used to solve the problem that particle is easily trapped into local optimum in BPSO; on the other hand, the modified updates on particle's position are used to solve the problem that particles in BPSO cannot converge to the optimal solution.

4.1. Processing of Constraint Condition

4.1.1. Binary Matrix Encoding. The mapping relationship between solution matrix E and particle's position X is established according to resource matching model. A 0-1 binary matrix is built by binary encoding matrix E , whose dimension is $C \times (TN \cdot m)$ (where $C = R_1 + R_2 + \dots + R_{n_1+n_2}$). The encoding process is shown in Figure 2.

This encoding structure transforms the sensor with multitargets detecting capability into several sensors with the single-target detecting capability. The particle's position is $X = (x_{ij})_{C \times (TN \cdot m)}$ (if x_{ij} is equal to 1, this denotes that the j th target is detected by the i th sensor in the interval of subtask, while if x_{ij} is equal to 0, this denotes that the j th target is not detected by sensor i th sensor). The particles created with this encoding method are very intuitive, which are the one-to-one mapping relationship with total scheduling matrix E and denote the scheduling results directly without decoding computation.

The analysis shows that this binary matrix encoding will satisfy the constraint conditions (1) and (5) of resources optimization objective function.

4.1.2. Further Processing of Constraint Condition. In order to eliminate the influence of constraint condition (2), (3), and (4) to objective function, these three constraints should be further processed, respectively.

Sensor		Target										
		1	...	m	...	$(j-1)m+1$...	$j-m$...	$(TN-1)m+1$...	$TN-m$
S1	1	0	...	1	...	1	...	0	...	1	...	0
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	⋮	1	...	0	1	0	...	0	...	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	C	1	...	1	...	0	...	1	...	0	...	1

FIGURE 2: Encoding process of particle.

- (1) *Processing the Constraint (2)*. The deployment location of satellites is considered, the two satellites whose location are close to each other are integrated as one sensor, and the target capacity of combined satellites is the minimum of the two satellites.
- (2) *Processing the Constraint (4)*. The penalty function method is adopted, which is to check the position where the element is equal to 0 in total detection matrix D , and then the corresponding position' value in the distance detection benefit matrix and angle detection benefit matrix is set zero, which can eliminate the infeasible solution.
- (3) *Processing the Constraint (3)*. In the course of encoding and updating the particle, constraint (3) is inspected, if it is not conformed to constraint (3), then particle will be modified, and the modified steps are as follows.

Step 1. The number of "1" in the column of particle is added up, which is defined as N .

Step 2. If $N < (r_j^{s1} + r_j^{s2})$, the value of $(r_j^{s1} + r_j^{s2}) - N$ in this column is set to zero randomly, and then Step 4 is run; if $N \geq (r_j^{s1} + r_j^{s2})$, Step 3 is run.

Step 3. If $N > (r_j^{e1} + r_j^{e2})$, the value of $N - (r_j^{e1} + r_j^{e2})$ in this column is set to zero randomly.

Step 4. If all columns of particle are corrected, the step is jumped to Step 1; otherwise, the modified step is completed.

4.2. Modified BPSO Algorithm

4.2.1. Modified Method for Particle Velocity Updating Means. The particle velocity updating means in basic BPSO is as

follows:

$$v_{ij}^{k+1} = \omega v_{ij}^k + c_1 r_1 (p_{ij}^k - x_{ij}^k) + c_2 r_2 (p_{gj}^k - x_{ij}^k), \quad (16)$$

where k is the k th iteration, ω is the inertia weight, c_1 and c_2 are the cognition learning rate and social learning rate, respectively, r_1 and r_2 are random number between 0 and 1, $P_{ibest} = [p_{i1}, p_{i2}, \dots, p_{iD}]$ is the optimal value of personal, and $P_{gbest} = [p_{g1}, p_{g2}, \dots, p_{gD}]$ is the swarm optimal value.

The useful information is not made full use of in the iterative process of basic BPSO; namely, such factors as the particle's velocity, personal optimal solution, and global optimal solution are contained. The whole swarm would fall into local optima, when global optimal solution is invalid at a certain moment or global optimal solution is trapped into local optima. The personal optimal solution and global optimal solution are doubted through introducing the doubt factor $-\omega^k r_3 [p_{ij}^k + p_{gj}^k]$, which is to reduce the probability of the premature convergence when particle is fallen into local optima. And the modified velocity updating means is as follows:

$$v_{ij}^{k+1} = \omega^k v_{ij}^k + c_1 r_1 (p_{ij}^k - x_{ij}^k) + c_2 r_2 (p_{gj}^k - x_{ij}^k) - \omega^k r_3 (p_{ij}^k + p_{gj}^k), \quad (17)$$

where r_3 is random number between 0 and 1.

The modified velocity updating means of the particle are shown in Figure 3.

4.2.2. Modified Method of Particle Position Updating Means. The particle position updating means in basic BPSO are as

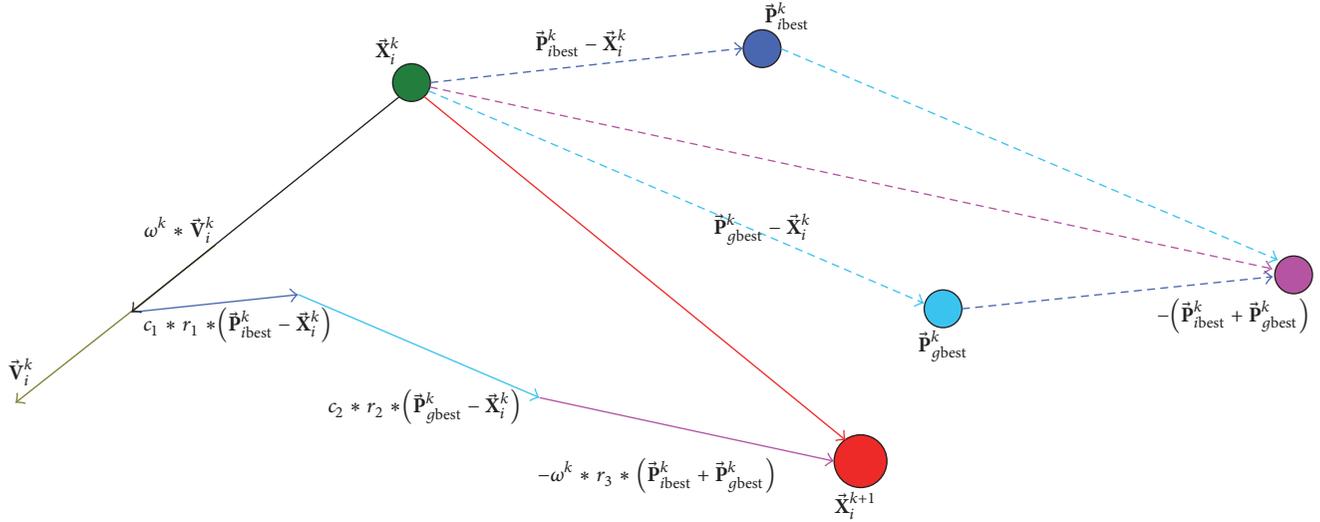


FIGURE 3: Graphical representation of the particle updating.

follows:

$$x_{ij}^{k+1} = \begin{cases} 1, & a < S(v_{ij}^{k+1}) \\ 0, & \text{otherwise,} \end{cases} \quad (18)$$

where a is random number between 0 and 1 and $S(v_{ij}^{k+1})$ denotes a Sigmoid fuzzy function (where $S(v_{ij}^{k+1}) = 1/1 + \exp(-v_{ij}^{k+1})$). $v_{ij}^{k+1} \in [v_{\min}, v_{\max}]$ denotes the probability of x_{ij}^{k+1} whose value is 1 or 0.

Sigmoid function is modified referring methods in [15], and the contrast diagram of the basic Sigmoid function and modified Sigmoid function is shown in Figure 4.

Analysis of Figure 4 shows that, in the Sigmoid function of BPSO, the particle velocity is larger and positive, which means that probability of 1 for the particle position is large; in the same way, the particle velocity is less and negative, which means that probability of 0 for the particle position is large, and the particle velocity is 0, which means that the probability of 0 and 1 for particle position is 0.5. While there is no difference between positive velocity and negative velocity in the modified Sigmoid function, a large absolute value of velocity means that the current particle's position is not suitable and a great movement is required to make the particle reach the optimal position. Also, a small absolute value of the velocity indicates that the current particle's position is close to the optimal position.

So the modified Sigmoid function is used to update particle's position in MBPSO algorithm; the modified velocity updating means is

$$S(v_{ij}^{k+1}) = E + (1 - E) \times |\tanh(v_{ij}^{k+1})| \quad (19)$$

$$x_{ij}^{k+1} = \begin{cases} \bar{x}_{ij}^k, & a < S(v_{ij}^{k+1}) \\ x_{ij}^k, & \text{other,} \end{cases} \quad (20)$$

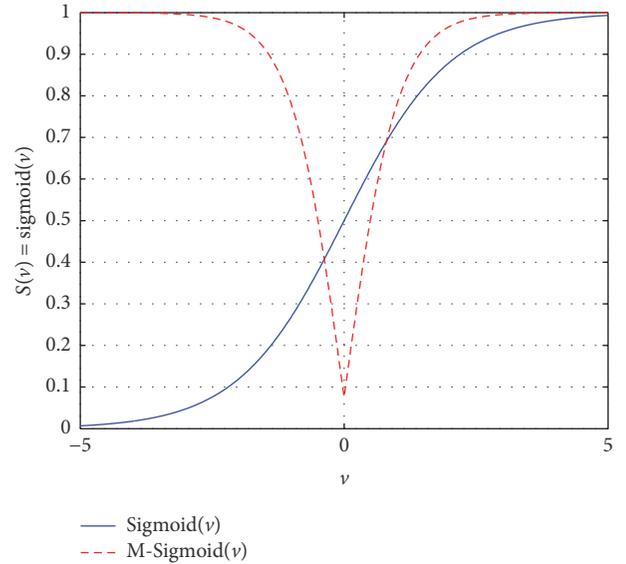


FIGURE 4: The contrast diagram of the basic Sigmoid function and modified Sigmoid function.

where $E = \text{erf}(NF/T^t) = (2/\sqrt{\pi}) \int_0^{NF/T^t} e^{-t^2} dt$ is a small value. If the optimal position of particles is not changed in several sequential iterations, E is added to accelerate the convergence rate and to escape from a local optima. NF is the number of generations that the optimal position is not changed in the sequential iterations, T^t is a time constant, and \bar{x}_{ij}^k is the complement of x_{ij}^k .

4.3. Algorithm Process. MBPSO algorithm process is as follows.

Step 1. Set parameter. Set ω , c_1 , c_2 , c_3 , the largest number of iterations Iter_{\max} , NF , T^t , population size, particle's dimension, and so on.

TABLE 1: Relevant parameters of TBM.

TBM	Launch time	Flight time (s)	Maximum height (km)	Priority
TBM1	18:00:00	930	551	2
TBM2	18:00:00	1032	615	1
TBM3	18:00:00	879	529	4
TBM4	18:00:00	911	543	3

Step 2. Initialize the particle swarm, the current personal optimal fitness value, and global optimal fitness value. The position and velocity of initial swarm are generated randomly and $k = 1$.

Step 3. Judge whether the particle meets constraint (3) or not; if constraint (3) is met, the processing step is gone to Step 4; otherwise, the particle is modified with the modified method.

Step 4. The fitness value of each particle is calculated by (10), then P_{ibest} and P_{gbest} are updated.

Step 5. The velocity of particle is updated by (17), and the position is updated by (19) and (20); then it is judged that whether the particle meets constraint (3) or not, if it meets, the process step is gone to Step 6; otherwise, the particle is modified with the modified method.

Step 6. If $k < \text{Iter}_{\max}$, then $k = k + 1$, and the process step is gone to Step 4; otherwise, the process step is gone to Step 7.

Step 7. Output P_{gbest} . The optimal scheduling matrix scheme is obtained; then the algorithm is ended.

5. Simulation Analysis

5.1. Simulation Conditions. In order to validate efficiency of the resources match model and MBPSO algorithm proposed in this paper, simulation scenario is supposed as follows:

- (1) Command and control system received early warning information of 4 TBMs at 18:00:30. Relevant information is shown in Table 1.
- (2) Two LEO satellites and three ground radars are deployed in early warning system. LEO satellite's parameters are as follows: orbit altitude is 1600 km, detection range is as far as 1400 km, field size is $20^\circ \times 10^\circ$, the best detection angle is 90° , target capacity is 2. Ground radar's parameters are as follows: two works on P -band, one works on X -band, and the detection range of three radars are, respectively, 3000 km, 4000 km, and 2000 km, the area of azimuth-angle is $\pm 60^\circ$, the area of pitch-angle is $1\text{-}90^\circ$, the best detection angle is 45° , 60° , and 80° , respectively, and target capacity of the radar is 3, 4, and 8, respectively.

Task scheduling time is from 18:06:30 to 18:08:30, in which sensors will detect targets and provide subsequent intercepting with significant instructions of TBM. The visualization interval of sensor to target is shown in Table 2.

TABLE 2: Visualization interval of sensor to TBM.

	TBM1	TBM2	TBM3	TBM4
Sat1	[0, 85]	[0, 102]	[0, 68]	[0, 89]
Sat2	[0, 120]	[0, 120]	[0, 120]	[0, 120]
PR1	[0, 85]	[0, 89]	[0, 72]	[0, 83]
PR2	[40, 120]	[46, 120]	[24, 120]	[29, 120]
XR1	[97, 120]	[110, 120]	[72, 120]	[83, 120]

5.2. Simulation Process. Assume D_{\max} is 20 s, D_{\min} is 10 s, and MacLong is 15 s. Nine subtasks are obtained by the task decomposition method proposed in this paper, whose corresponding interval, respectively, is [0, 12], [12, 24], [29, 40], [46, 57], [57, 68], [72, 83], [83, 97], [97, 110], and [110, 120]. Meanwhile, subtasks' detection matrix is as follows:

$$D^1 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$D^2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$D^3 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$D^4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$D^5 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{aligned}
 D^6 &= \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} & D^4 &= \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\
 D^7 &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} & D^5 &= \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\
 D^8 &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} & D^6 &= \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \\
 D^9 &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}. & D^7 &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}
 \end{aligned}
 \tag{21}$$

Assume that $\text{Iter}_{\max} = 50$, $NF = 5$, $c_1 = c_2 = c_3 = 2.05$, $T^l = 20$, swarm size is 30, weight coefficient $\alpha_1 = 0.3$, $\alpha_2 = 0.25$, $\alpha_3 = 0.2$, $\alpha_4 = 0.15$, $\alpha_5 = 0.1$, $r_j^{s1} + r_j^{s2} = 1$, and $r_j^{e1} + r_j^{e2} = 4$. According to resources match model in Section 2, the total scheduling matrix is solved by MBPSO algorithm. So subtasks' scheduling matrix is as follows:

$$\begin{aligned}
 D^1 &= \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} & D^8 &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} \\
 D^2 &= \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} & D^9 &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.
 \end{aligned}
 \tag{22}$$

5.3. Simulation Conclusion

5.3.1. Optimizing Scheduling Scheme. The total scheduling matrix D is transformed, and then the optimal match scheme is obtained, as shown in Figure 5.

Analysis of Figure 5 shows the following:

- (1) The optimal scheduling scheme meets the constraint of total detection matrix.
- (2) It is guaranteed that there is at least one sensor to detect each TBM, in addition to TBM3 which is not detected in period $[0, 29]$. Under the constraint of

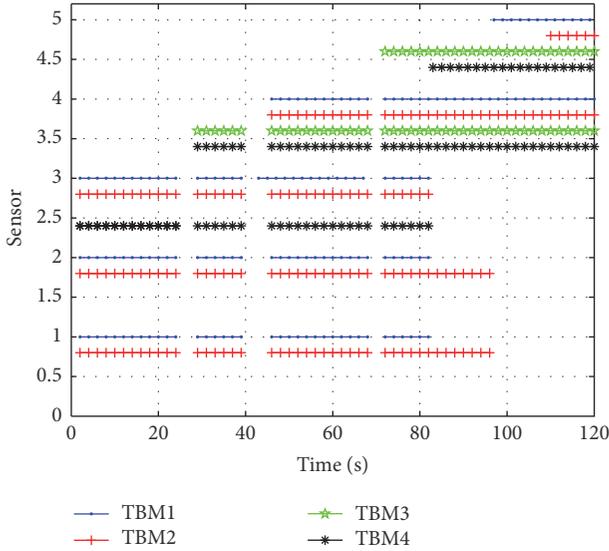


FIGURE 5: Optimal scheduling scheme of multisensor for four-TBM detection.

TABLE 3: Performance contrast of algorithms.

Algorithm	Average convergence iteration	Average computation time/s
BPSO	16	2.5382
NBPSO	32	3.0759
MBPSO-2	22	2.5968
MBPSO	18	2.6125

binary satellite detection, the TBM is detected by sensor 1 and sensor 2 for the meantime.

- (3) More sensors are needed to detect TBM2, 4 sensors at the most time, while less sensors are needed to detect TBM3, 2 at the most time. That is, TBM2's priority is the highest, and TBM3's priority is the lowest.
- (4) Besides intervals [24, 29] and [40, 46], detection of each sensor to TBM is not interrupted, which denotes that detection to TBM is continuous and has fewer handover times.

5.3.2. *Algorithm Performance Contrast.* The algorithm is verified in MATLAB R2013a on Pentium 3.2 GHz platform. The above match optimization model is solved by BPSO [11], NBPSO [18], MBPSO-2, respectively, and the proposed algorithm, and the performance of each algorithm is compared through calculating 100 times. MBPSO-2 is the algorithm which is obtained by improving the particle's velocity updating means in BPSO.

The optimum fitness value contrast of four kinds of algorithm is shown in Figure 6.

The performance contrast of four kinds of algorithm is shown in Table 3.

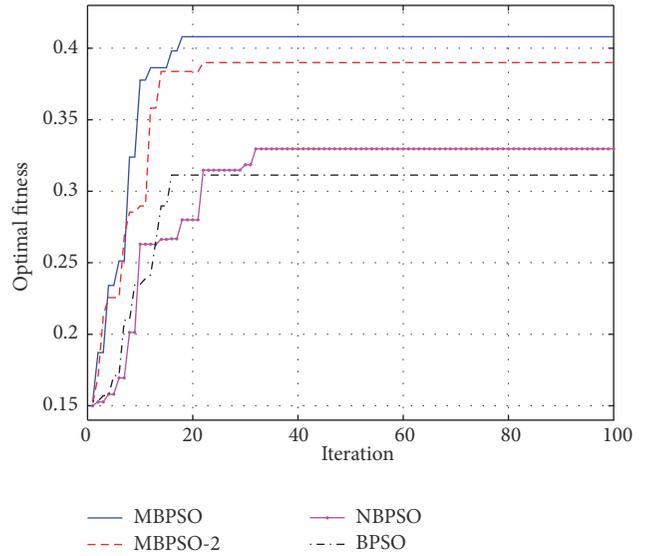


FIGURE 6: Contrast of optimal fitness value curves.

It can be analyzed from Figure 6 and Table 3 that the optimal fitness value of the MBPSO algorithm is better than that of the other three algorithms, and the iteration number of MBPSO is the least, which is 18, while the iteration number of MBPSO-2 and NBPSO is 22 and 32, respectively, which suggest that the real time of the MBPSO algorithm is the best among the BPSO, NBPSO, MBPSO-2, and MBPSO.

On one hand, the Sigmoid function in NBPSO is modified, and the optimal fitness value of NBPSO is better than BPSO, but NBPSO still easily traps into local optima due to its velocity updating means. On the other hand, the velocity updating means in MBPSO-2 are modified, and the optimal fitness value of MBPSO-2 is greatly improved, but the particle cannot be converged to the best solution because of the problem of basic Sigmoid function. With both the velocity and position updating means of particle modified in MBPSO, which makes it easier for particle to jump out of local optima, the convergence speed and search capability of MBPSO are better than those of the other three algorithms.

6. Conclusions

The resources scheduling problem for the multisensor cooperative detecting the ballistic target is researched in this paper. The multisensor cooperative detecting task is decomposed as many subtasks by hybrid task decomposition method. Then the resource match optimization model based on the resources and tasks is established. Finally, MBPSO algorithm is proposed to solve the match optimization model efficiently. Simulation results show that the proposed algorithm is better in terms of convergence velocity, searching capability, solution accuracy, and efficiency.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Authors' Contributions

Changyun Liu and Xiangke Guo have contributed equally to this paper.

Acknowledgments

This paper is supported by National Natural Science Fund of China (61503408) and Doctoral Foundation of Air Force Engineering University under Grant (KGD080914002).

References

- [1] Y. Zhi-Wei, N. Yi-Feng, and L. Han-Ling, "Study of sensor scheduling for early warning satellite based on parallel tabu genetic algorithm (PTGA)," *Journal of Astronautics*, vol. 24, no. 6, pp. 598–603, 2003.
- [2] T. Shao-Xun, Y. Xian-Qing, and L. Xue-Shan, "An Improved particle swarm optimization algorithm for early warning satellites scheduling problems," *Systems Engineering*, vol. 30, no. 1, pp. 116–121, 2012.
- [3] P. Jian, P. Zou, and W. Xiong, "Heuristic algorithm for dynamic task planning of early warning system of Low Earth Orbit," *Journal of Electronics and Information Technology*, vol. 35, no. 10, pp. 2438–2444, 2013.
- [4] F. Ming-Yue, L. Guo-Hui, and Y. Xian-Qing, "Rule-based scheduling method for sensors on early warning satellites basing on task analysis," *Computer Applications*, vol. 27, no. 12, pp. 3095–3098, 2007.
- [5] M. Feng, G. Liu, and X. Yi, "Sensor scheduling method for space-based early warning system," *Computer Engineering and Applications*, vol. 45, no. 2, pp. 225–228, 2009.
- [6] X.-S. Zhao, H.-X. Wang, and Y.-C. Cai, "Resource scheduling method in antimissile early warning campaign," *Systems Engineering and Electronics*, vol. 37, no. 6, pp. 1300–1305, 2015.
- [7] W.-L. Guan, Y.-H. Tan, H.-X. Zhang, and J.-L. Song, "Distribution system feeder reconfiguration considering different model of DG sources," *International Journal of Electrical Power & Energy Systems*, vol. 68, pp. 210–221, 2015.
- [8] M. Yusoff, J. Ariffin, and A. Mohamed, "DPSO based on a min-max approach and clamping strategy for the evacuation vehicle assignment problem," *Neurocomputing*, vol. 148, pp. 30–38, 2015.
- [9] L. Zhang, Y. Tang, C. Hua, and X. Guan, "A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques," *Applied Soft Computing Journal*, vol. 28, pp. 138–149, 2015.
- [10] Y.-J. Zheng, H.-F. Ling, J.-Y. Xue, and S.-Y. Chen, "Population classification in fire evacuation: a multiobjective particle swarm optimization approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 70–81, 2014.
- [11] J. Kennedy and R. C. Eberhart, "Discrete binary version of the particle swarm algorithm," in *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 4104–4108, Orlando, Fla, USA, October 1997.
- [12] A. H. El-Maleh, A. T. Sheikh, and S. M. Sait, "Binary particle swarm optimization (BPSO) based state assignment for area minimization of sequential circuits," *Applied Soft Computing*, vol. 13, no. 12, pp. 4832–4840, 2013.
- [13] K. Fan, W. You, and Y. Li, "An effective modified binary particle swarm optimization (mBPSO) algorithm for multi-objective resource allocation problem (MORAP)," *Applied Mathematics and Computation*, vol. 221, pp. 257–267, 2013.
- [14] B. Wei and J. Zhao, "Haplotype inference using a novel binary particle swarm optimization algorithm," *Applied Soft Computing*, vol. 21, pp. 415–422, 2014.
- [15] Z. Beheshti, S. M. Shamsuddin, and S. Hasan, "Memetic binary particle swarm optimization for discrete optimization problems," *Information Sciences*, vol. 299, pp. 58–84, 2015.
- [16] R. Jun-Liang, X. Qing-Hua, L. Qiang, and J. Zhe, "Resource scheduling method of missile defense early warning system based on self-adaptive probability particle swarm optimization," *Journal of Air Force Engineering University*, vol. 15, no. 6, pp. 45–48, 2014.
- [17] W. Bo, A. Wei, X. Kai et al., "Multi-object tracking sensor scheduling for Low Earth Orbit constellation based on multi-model," *Acta Aeronautica et Astronautica Sinica*, vol. 31, no. 5, pp. 946–957, 2010.
- [18] M. Rostami-Shahrbabak and H. Nezamabadi-Pour, "New approach to binary PSO algorithm," in *Proceedings of the 14th Iranian Conference on Electrical Engineering*, pp. 1–6, Tehran, Iran, May 2006.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

