RAA 2017 Vol. **17** No. **3**, 30 (10pp) doi: 10.1088/1674–4527/17/3/30
© 2017 National Astronomical Observatories, CAS and IOP Publishing Ltd.
http://www.raa-journal.org *http://iopscience.iop.org/raa*

Research in Astronomy and Astrophysics

Particle swarm optimization based space debris surveillance network scheduling

Hai Jiang[1,2,3], Jing Liu[1,2,3], Hao-Wen Cheng[1,3] and Yao Zhang[1,2,3]

[1] National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100012, China; *jhai@nao.cas.cn*
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Space Debris Observation and Data Application Center, China National Space Administration, Beijing 100012, China

Received 2016 September 2; accepted 2017 January 12

Abstract The increasing number of space debris has created an orbital debris environment that poses increasing impact risks to existing space systems and human space flights. For the safety of in-orbit spacecrafts, we should optimally schedule surveillance tasks for the existing facilities to allocate resources in a manner that most significantly improves the ability to predict and detect events involving affected spacecrafts. This paper analyzes two criteria that mainly affect the performance of a scheduling scheme and introduces an artificial intelligence algorithm into the scheduling of tasks of the space debris surveillance network. A new scheduling algorithm based on the particle swarm optimization algorithm is proposed, which can be implemented in two different ways: individual optimization and joint optimization. Numerical experiments with multiple facilities and objects are conducted based on the proposed algorithm, and simulation results have demonstrated the effectiveness of the proposed algorithm.

Key words: methods: data analysis — observational catalogs — telescopes — techniques: radar astronomy

1 INTRODUCTION

With the increase of human space activities, the population of space objects has consistently risen during the last several decades, and created an orbital debris environment that poses increasing impact risks to existing space systems, including human space flights and robotic missions (Kessler & Cour-Palais 1978; Liou & Johnson 2006). There are currently more than 20 000 tracked objects in Earth orbit, only 1300 of which are active spacecrafts. Tracking space debris is an extremely important task for maintaining the safety and viability of manned and unmanned spacecrafts. The sensors used to track these objects are mechanical/phased-array radars and optical telescopes. Unfortunately, insufficient resources exist to easily make enough observations to track every object's orbit with desired accuracy. This presents a significant challenge to the current space debris surveillance network. One of the challenges of sensor scheduling in the space surveillance network (SSN) is the large number of resident space objects (RSOs) that must be tracked in such a way as to produce acceptable orbital state accuracy to support conjunction analysis and collision avoidance maneuver planning (Duncan & Wysack 2011).

The goal of sensor scheduling is to allocate a collection of resources in a manner that most significantly improves the ability to predict and detect events involving the RSOs. For effectively allocating these resources, capabilities of the sensors, the system's historical performance and orbital accuracy requirements for cataloging are three main factors which should be taken into consideration.

The current US SSN tasking system only takes a limited number of factors into account (Wilson 2004), producing only prioritized lists of the RSOs assigned to each sensor, with requested collection number of tracks for each object. The creation of a timeline for the collection of observations is left to the individual sensor sites. There are a number of other techniques that have been proposed to address the optimization of SSN sensor tasking. Miller (2007)'s work is well known and is based on marginal analysis of the energy dissipation rate of orbital objects.

Covariance-based scheduling also has been proposed by Hill et al. (2010). Combinations of multiple algorithms were also proposed including the algorithm for bottleneck avoidance (Stottler 2012). Herz & Stoner (2013) recently proposed an optimization algorithm inherent in its COTS product with a proprietary optimization search.

To the authors' knowledge, there are issues that still need to be addressed with the current process: (1) the schedule may not be globally optimized since scheduling occurs only locally without reference to what is assigned or scheduled at other sensor sites; (2) the effects of other observations, which can provide complementary observations of the object and improve the object's orbit precision, of the same object by other specific sensors at specific times should be considered.

Particle swarm optimization (PSO) is an evolutionary computation technique inspired by the behavior of bird flocks, which was first introduced by Kennedy & Eberhart (1995). PSO chooses the path of cooperation over competition, so the PSO population is stable and individuals are neither destroyed nor created. Individuals are affected by the best performance of their neighbors, and individuals eventually converge on optimal points in the problem domain. Several investigations have been undertaken recently to improve the performance of the original PSO (Clerc 1999; Clerc & Kennedy 2002; Marini & Walczak 2015). PSO is capable of producing low cost, fast and robust solutions to several complex problems and it has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control and other areas.

We took advantage of PSO. A scheduling algorithm based on PSO is developed that takes as input a catalog of space debris and produces a globally optimized schedule for each sensor site as to what objects to observe, which can be implemented in two different ways. The algorithm is able to schedule a better number of observations targeting each object with the same sensor resources and make those observations more efficient. The results would be increased accuracy of the space catalog with fewer lost objects and the same set of sensor resources. The algorithms have been tested with simulated data, and the results presented in this paper show the efficiency of the proposed algorithms.

The rest of the paper is organized as follows: The next section introduces the formulation of this problem. Some criteria for optimal scheduling are given in Section 3, while the proposed scheduling algorithm based on PSO is presented in Section 4. Then, Section 5 describes the test experimental settings, and experimental results of the proposed algorithm. Finally, Section 6 summarizes the contribution of this paper and conclusions.

2 PROBLEM FORMULATION

Space debris surveillance network scheduling is a problem of optimally scheduling the surveillance time period for space surveillance facilities and space debris. Due to limitations imposed by their orbits, space debris can only be tracked by surveillance facilities in some specific periods. These periods are usually called visible windows.

The formulation of a scheduling problem requires the specification of visible window set $\Xi = \{\epsilon_1, \cdots, \epsilon_D\}$. A visible window $\epsilon_d = [\kappa_d, \eta_d]$ in Ξ represents a specific period that an object α_m from the object set $\mathrm{T} = \{\alpha_1, \cdots, \alpha_M\}$ can be observed by a facility β_n from the facility set $\mathrm{F} = \{\beta_1, \cdots, \beta_N\}$, where κ_d and η_d are the start time and end time of the visible window $\epsilon_d (1 \leq d \leq D)$, respectively. As shown in Figure 1, there is a one-way mapping from the visible window set to object set and to facility set, such that each visible window maps to one and only one object and one facility.

The facilities used to track space debris are mechanical/phased-array radars and optical telescopes. Some of them can only track one or a limited number of objects at a time, but there are a large number of space debris in Earth orbit, and the durations of visible windows may conflict.

Figure 2 illustrates the conflicts of a single-object tracking facility. In this situation, the decision maker faces the problem of how to allocate available resources to optimize some metrics such as efficiency, cost, information return, etc., while satisfying all constraints levied on the resources as well as the tasks. When there are unresolvable conflicts, the decision maker must accept certain tasks and reject others. In many cases the availabilities of resources become a direct constraint on the problem.

Without loss of generality, some assumptions are made to simplify the scheduling process as follows:

– Only a limited number of objects can be tracked by a facility at a time.
– There should be no more than one tracking task scheduled in each visible window.
– For effective tracking, the time length of each track provided by a facility must be longer than the facility's minimal length of working time τ.

3 CRITERIA FOR SCHEDULING

The goals of space debris surveillance network scheduling are to help in making the utilization of a surveillance

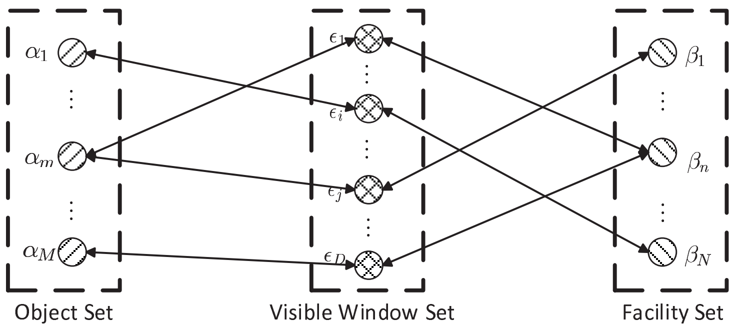

Fig. 1 Relationships among an object set, a facility set and a visible window set.

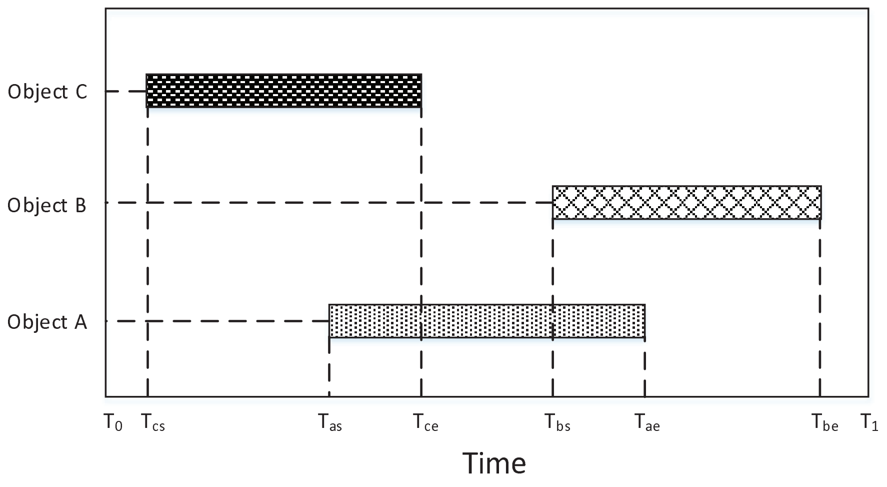

Fig. 2 Conflicts among different visible windows of a single-object tracking facility.

network more efficient and to improve the quality of the element sets in a satellite catalog. The surveillance performance on each object and the balance of resources available for surveillance facilities are two main factors, which should be carefully considered. On the basis of these considerations, two different criteria are introduced to evaluate the performance of the scheduled visible window set $\Xi = \{\epsilon_1, \cdots, \epsilon_D\}$ generated by an object set $\mathrm{T} = \{\alpha_1, \cdots, \alpha_M\}$ and a facility set $\mathrm{F} = \{\beta_1, \cdots, \beta_N\}$ in this section.

3.1 Scheduling Score

In order to evaluate the performance of a scheduled visible window set Ξ on each object, we define a scheduling score function $D_{\mathrm{S}}(\Xi; \mathrm{T})$ which sums the performance score of each object. It can be expressed as

$$D_{\mathrm{S}}(\Xi, \mathrm{T}) = \sum_{m=1}^{M} \rho_m \cdot \mathcal{F}\left(\sum_{d=1}^{D} \Phi_{d,m} \left(\eta_d - \kappa_d\right), \mu_m\right), \tag{1}$$

where ρ_m is the priority coefficient of object α_m; Φ is a $D \times M$ object decision matrix, $\Phi_{d,m} = 1$ only if visible window ϵ_d is scheduled for object α_m, otherwise $\Phi_{d,m} = 0$; κ_d and η_d are the start time and end time of the visible window ϵ_d, respectively; $\mathcal{F}(t, \mu)$ is a function to evaluate the efficiency of the total surveillance time t on its corresponding object, which is defined as

$$\mathcal{F}(t, \mu) = \begin{cases} 0, & t < \mu, \\ 1 + \sigma \frac{t-\mu}{\mu}, & t \geq \mu, \end{cases}$$

where μ is the minimal amount of surveillance time required for effective observation of an object, and σ is a redundancy coefficient in [0, 1].

3.2 Balance Level

Each facility has its own separate requirements and missions. Task scheduling is up to each individual facility to schedule the times for each observation it will perform during the day. Usually, some specific facilities will take precedence from experience in case of a scheduling conflict. We define the balance coefficient of facility β_k as λ_k ($\lambda_k \geq 1$), where facilities with higher balance coefficients will take priority in case of conflicts. We define the balance level function $D_{\rm B}(\Xi;{\rm F})$ as

$$D_{\rm B}(\Xi,{\rm F}) = \frac{\mathcal{P}\left(\beta_1,\cdots,\beta_N\right)}{\mathcal{P}_{\max}} \tag{2}$$

and $\mathcal{P}\left(\beta_1,\cdots,\beta_N\right)$ is a product relating the balance coefficient and utilization ratio of all facilities, which can be expressed as

$$\mathcal{P}\left(\beta_1,\cdots,\beta_N\right) = \left(\prod_{k=1}^{N}\left(1+\lambda_k\cdot U_k\right)\right)^{\frac{1}{N}}.$$

Here N is the number of facilities and U_k is the utilization ratio of facility β_k, which is defined as

$$U_k = \frac{\sum\limits_{d=1}^{D}\Psi_{d,k}\cdot(\eta_d-\kappa_d)}{\sum\limits_{d=1}^{D}\sum\limits_{n=1}^{N}\Psi_{d,n}\cdot(\eta_d-\kappa_d)},$$

where Ψ is a $D\times N$ facility decision matrix. $\Psi_{d,n}=1$ only if visible window ϵ_d is scheduled for facility β_n, otherwise $\Psi_{d,n}=0$; and $\mathcal{P}_{\max}$ is the theoretical maximum of $\mathcal{P}\left(\beta_1,\cdots,\beta_N\right)$, which can be expressed as

$$\mathcal{P}_{\max} = \left(\prod_{k=1}^{N}\left(1+\frac{\lambda_k}{N}\cdot\left(1+\sum_{i=1}^{N}\frac{1}{\lambda_i}-\frac{N}{\lambda_k}\right)\right)\right)^{\frac{1}{N}}.$$

4 PSO BASED SCHEDULING ALGORITHM

4.1 Review of PSO

PSO is an evolutionary computation technique. In PSO, each candidate solution is a particle and represents a point in a D-dimensional space, where D is the number of parameters to be optimized. Accordingly, the position of the i-th particle of the swarm at iteration t can be represented by a D-dimensional position vector $X_i^t=(x_{i1}^t,\cdots,x_{iD}^t)$. The velocity of the particle is denoted by $V_i^t=(v_{i1}^t,\cdots,v_{iD}^t)$. The best position the i-th particle has experienced is

$$P_i^t=(P_{i1}^t,\cdots,P_{iD}^t),$$

and the best position of all particles explored so far is

$$P_g^t=(P_{g1}^t,\cdots,P_{gD}^t).$$

The position of the particle and its velocity are updated using the following equations:

$$\begin{aligned} v_{id}^{t+1} &= v_{id}^t + c_1\cdot r_1\cdot(P_{id}^t-x_{id}^t) \\ &\quad + c_2\cdot r_2\cdot(P_{gd}^t-x_{id}^t), \qquad (3)\\ x_{id}^{t+1} &= x_{id}^t+v_{id}^{t+1}, \qquad (4)\end{aligned}$$

where c_1 and c_2 are acceleration coefficients, which are real-valued and usually in [0, 4], c_1 is a cognitive coefficient that quantifies how much the particle trusts its experience, c_2 is a social coefficient that quantifies how much the particle trusts its best neighbor, and r_1 and r_2 are random numbers generated from a uniform distribution in [0, 1].

In order to avoid the swarm diverging due to scattering of the new velocity, an inertia weight factor is introduced by Shi & Eberhart (1998). The inertia weight ω creates a tendency for the particle to continue moving in the same direction it was going previously. Accordingly, the velocity update equation is modified to

$$\begin{aligned} v_{id}^{t+1} &= \omega(t+1)\cdot v_{id}^t + c_1\cdot r_1\cdot(P_{id}^t-x_{id}^t) \\ &\quad + c_2\cdot r_2\cdot(P_{gd}^t-x_{id}^t). \qquad (5)\end{aligned}$$

Several studies (Eberhart & Shi 2001; Arumugam & Rao 2006; Bansal et al. 2011) have shown that a dynamical adjustment to the value of $\omega(t)$ may significantly improve the convergence properties of PSO. A linearly decreasing strategy is commonly used in inertia weight $\omega(t)$ adjustment, which is shown in the following equation

$$\omega(t)=\omega_{\max}-\frac{\omega_{\max}-\omega_{\min}}{t_{\max}}\cdot t. \tag{6}$$

Bansal et al. (2011)'s study shows that the adoption of the combination $\omega_{\max}=0.9$ and $\omega_{\min}=0.4$ may achieve better performance.

4.2 Proposed Algorithm

The PSO based task scheduling algorithm can be divided into the following four steps:

(1) Initialize all related parameters and compute all visible windows;
(2) Decode the visible windows to the execution status through a particle decoding process;
(3) Execute the PSO process. Calculate the fitness value of each particle, and find the best particle;
(4) Check the stopping criteria. If the pre-set maximum number of generations is reached or if no improvement to the best solution is obtained after a given number of iterations, then the process is terminated. Otherwise, go back to Step 2.

This algorithm can be implemented in two different ways. The first one is individual optimization, which will implement this algorithm for the tasks of each sensor separately and then get the overall result; the other one is joint optimization, which will implement the algorithm for the tasks of all sensors.

The flow chart of the proposed algorithm is shown in Figure 3. Some key technologies for the implementation of the proposed technique are described in the following sub-sections.

4.2.1 Particle decoding

The optimization will be done on a visible window set $\Xi = \{\epsilon_1, \cdots, \epsilon_D\}$ generated by an object set $\mathrm{T} = \{\alpha_1, \cdots, \alpha_M\}$ and a facility set $\mathrm{F} = \{\beta_1, \cdots, \beta_N\}$. A particle applied to the surveillance task scheduling scenario is defined as $\mathbf{X} = \{x_1, \cdots, x_D\}$, where D is the number of visible windows, and x_d denotes the priority coefficient of the d-th visible window.

Particle decoding is the process of decoding a visible window set to execution status according to the value of the particle, and also generating the object decision matrix Φ and the facility decision matrix Ψ for fitness computation. The definitions of object decision matrix Φ and facility decision matrix Ψ are the same as those that were introduced in Section 3.

Since the d-th visible window is linked to a specific object and a specific facility, it can only be scheduled for that object and facility or not. There is one and only one mapping both from $\mathbf{X}$ to Φ and from $\mathbf{X}$ to Ψ in the de-conflict process.

4.2.2 De-conflict process

De-conflict is a process of eliminating the conflicts and finding a solution that obeys all physical constraints. It plays an essential role in the step of decoding particles and consists of the following steps:

(1) Initialization. Set all of the elements of object decision matrix Φ and facility decision matrix Ψ to 0.
(2) Select a visible window from the visible window set. Select a visible window ϵ_d with the maximal priority coefficient value x_d from set Ξ, and determine its corresponding object α_m and facility β_n.
(3) Determine the beginning time of the selected window. Set the beginning time of the selected visible window equal to its own beginning time κ_d.
(4) Determine the end time of the selected window. The determination process contains two steps. First, find the minimal value t_0 of the maximum possible beginning times of all visible windows that conflict with ϵ_d and have end times no less than $\kappa_d + 2\tau$, and then set the end time of ϵ_d to the minimal value of t_0 and η_d. The whole process can be mathematically expressed as

$$\eta_d = \min\left\{\eta_d, \min\left\{\eta_h - \tau \,\middle|\, \begin{array}{l} \epsilon_h \in \Gamma_d^c \\ \eta_h \geq \kappa_d + 2\tau \end{array}\right\}\right\},$$

where η_d is the end time of window ϵ_d, η_h is the end time of window ϵ_h, Γ_d^c denotes a visible window set which contains all visible windows that conflict with the window ϵ_d and τ is the facility's minimal length of working time.
(5) Update the visible window set. Trim the conflicting part of each visible window that conflicts with ϵ_d, keep the longer part of the window instead of the old window and then delete all visible windows in Ξ whose time spans are shorter than the facility's minimal length of working time τ. Record the result of the scheduled window ϵ_d and delete ϵ_d from Ξ.
(6) Update object decision matrix and facility decision matrix for fitness computation. Since the visible window ϵ_d is scheduled for observation, and its corresponding object and facility are α_m and β_n respectively, we have to set $\Phi_{d,m} = 1$ and $\Psi_{d,n} = 1$.
(7) Go to Step 2 until Ξ is empty.

4.2.3 Fitness function

The performance of surveillance targeting each object and balancing the resources consumed by all facilities are two main considerations that determine how to schedule the surveillance tasks. So, we define the fitness function as the multiplication of the scheduling score function and the balance level computation function, which can be expressed as

$$\mathrm{Fitness}(\mathrm{T}, \mathrm{F}, \Xi) = D_{\mathrm{S}}(\Xi, \mathrm{T}) \cdot D_{\mathrm{B}}(\Xi, \mathrm{F}) \tag{7}$$

where the definitions of T, F and Ξ are the same as those in Section 2, and $D_{\mathrm{S}}(\Xi, \mathrm{T})$ and $D_{\mathrm{B}}(\Xi, \mathrm{F})$ are defined with Equations (1) and (2).

5 EXPERIMENTAL RESULTS

In this section, we verify the performance of the proposed scheduling methods by comparing their scheduling results with those results computed with other methods.

Two tracking radars are selected in these experiments. Detailed information on the radars is shown in Table 1. Objects' orbits are generated using two line elements (TLEs) accessed from the Space-Track website (www.space-track.org) on 2014 August 18. Of all the TLEs available in the catalog, active satellites in the altitude range below 2000 km with radar cross sections greater than 1 square meter are selected in the following experiments. The runs span a 30-minute (2014–

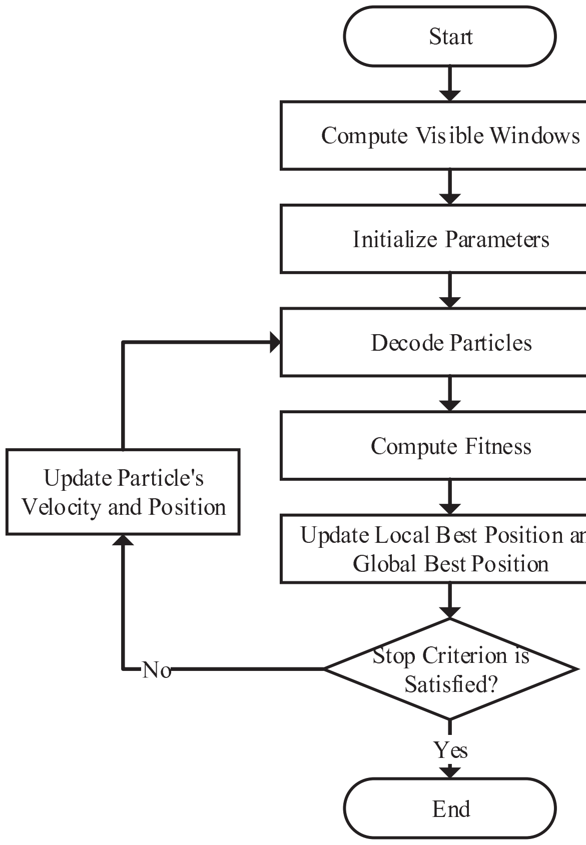
Start
Compute Visible Windows
Initialize Parameters
Decode Particles
Compute Fitness
Update Local Best Position and Global Best Position
Stop Criterion is Satisfied?
Update Particle's Velocity and Position
No
Yes
End

Fig. 3 Flow chart of the proposed scheduling algorithm.

Table 1 Information on Selected Sensors

Parameters	Radar 1	Radar 2
Latitude (°)	40	30
Longitude (°)	116	106
Altitude (m)	0	0
Detection Sensitivity ($\mathrm{km\ m^{-2}}$)	5000	5000
Azimuth Limitations (°)	60∼180	60∼180
Elevation Limitations (°)	30∼60	30∼60
Minimal Track Period (s)	60	60

08–18 00:00:00 ∼2014–08–18 00:30:00) simulation period. All visible windows among all selected objects and radars are analyzed with PROOF (Krag et al. 2000). The 43 resulting objects have a total of 50 visible windows longer than 60 seconds. Detailed information is shown in Table 2. The minimal surveillance time is set to 60 seconds; the priority coefficients of all objects are set to 1 except for object 25676, whose priority coefficient is set to 5; the balance coefficients of two radars are both set to 1 and we assume that each radar can track only one object at a time.

The first method is approximately the same as what was mentioned in Wilson (2004). It schedules the tasks according to the time order as well as the objects' priority coefficients, and priority goes to objects with earlier start time when there are conflicts among those objects with the same priority coefficients. Table 3 shows those tasks scheduled with the first method for both Radar 1 and Radar 2. It has a fitness value of 35.6568, which was calculated with Equation (7).

Both individual optimization and joint optimization methods are run for the same parameters, constraints and cost functions. The methods start by initializing a group of 500 particles, with random positions in a 50-dimensional hyperspace, constrained between zero and one in each dimension. A set of random velocities is also initialized with values in $[-1, 1]$. The termination criterion is set to the iteration number exceeding its maximal iteration threshold. Other parameters selected in the experiments are as follows: maximum number of iterations = 100, constant inertia weight $\omega_{\max} = 0.9$ and $\omega_{\min} = 0.4$, minimal amount of surveillance time of object $\mu = 60$, redundancy coefficient $\sigma = 0.1$ and facil-

Table 2 All Detectable Tasks of the Two Radars

Radar No.	NORAD ID	Beginning Time	End Time	Radar No.	NORAD ID	Beginning Time	End Time
1	13718	0.000	67.480	1	21087	1551.787	1686.885
1	05731	0.000	86.221	1	10491	1587.434	1678.959
1	25624	0.000	141.069	1	04507	1593.907	1800.000
1	25943	47.445	129.719	1	38745	1735.184	1800.000
1	25078	161.558	339.050	2	07768	0.000	61.796
1	25943	223.880	482.285	2	13718	0.000	80.933
1	14624	233.807	462.879	2	21032	0.000	152.055
1	21014	241.094	352.755	2	25943	0.000	200.881
1	38744	309.336	622.153	2	13148	35.500	148.466
1	20774	346.552	579.779	2	29712	227.769	379.733
1	32264	513.474	698.864	2	18957	245.967	543.041
1	27422	550.954	678.228	2	20774	265.628	603.640
1	25909	562.832	790.880	2	25676	266.376	763.613
1	25676	568.067	1029.247	2	38744	291.034	717.228
1	22689	578.510	875.903	2	10744	453.005	599.984
1	21031	609.591	936.783	2	32264	496.748	616.302
1	25944	642.864	823.016	2	22565	521.121	645.543
1	05104	667.658	791.423	2	16969	563.639	677.971
1	25040	732.229	839.581	2	12903	723.024	830.365
1	27534	886.147	1068.241	2	20233	822.453	1150.251
1	24870	1083.209	1246.203	2	24772	862.345	996.767
1	25039	1227.461	1388.719	2	19573	865.685	991.944
1	20510	1255.724	1335.430	2	14401	1403.692	1621.622
1	03576	1314.054	1635.846	2	21666	1509.006	1637.356
1	02801	1444.044	1664.759	2	25679	1648.143	1745.250

Table 3 Tasks Scheduled with the First Method

Radar No.	NORAD ID	Beginning Time	End Time	Radar No.	NORAD ID	Beginning Time	End Time
1	13718	0.000	67.480	1	04507	1635.846	1800.000
1	25624	67.480	141.069	2	07768	0.000	61.796
1	25078	161.558	339.050	2	21032	61.796	152.055
1	25943	339.050	482.285	2	29712	227.769	379.733
1	38744	482.285	622.153	2	18957	379.733	543.041
1	32264	622.153	698.864	2	20774	543.041	603.640
1	25909	698.864	790.880	2	25676	603.640	763.613
1	25676	790.880	1029.247	2	12903	763.613	830.365
1	24870	1083.209	1246.203	2	20233	830.365	1150.251
1	25039	1246.203	1388.719	2	14401	1403.692	1621.622
1	03576	1388.719	1635.846	2	25679	1648.143	1745.250

ity's minimal length of working time $\tau = 60$. After all these particles are scored, the best performer is identified as the initial global best.

These two methods are tested with 100 runs of Monte Carlo (MC) simulations in which each simulation is performed with different seeds in generation of the positions and velocities.

Figures 4 and 5 present the scheduling performance of individual PSO optimization and joint PSO optimization in their first five trials, respectively. It has been observed that there is significant improvement in the performance of the individual PSO optimization method compared with the first method, and the performance of the joint PSO optimization method slightly outperforms the individual optimization method. The performances of both individual PSO optimization and joint PSO optimization are much better than the first method.

Tables 4 and 5 show the best scheduled results with the individual PSO optimization method and joint PSO optimization method in the 100 MC runs respectively. A

Table 4 The Best Scheduled Result with Individual PSO Optimization in 100 MC Runs

Radar No.	NORAD ID	Beginning Time	End Time	Radar No.	NORAD ID	Beginning Time	End Time
1	05731	0.000	69.719	1	21087	1551.787	1618.959
1	25624	69.719	141.069	1	04507	1618.959	1735.184
1	25078	161.558	241.094	1	38745	1735.184	1800.000
1	21014	241.094	352.755	2	13718	0.000	80.933
1	20774	352.755	562.153	2	25943	80.933	200.881
1	27422	562.153	638.864	2	25676	266.376	539.984
1	25676	638.864	730.880	2	38744	616.302	717.228
1	22689	791.423	875.903	2	32264	539.984	616.302
1	21031	875.903	936.783	2	12903	723.024	830.365
1	05104	730.880	791.423	2	20233	830.365	931.944
1	27534	936.783	1068.241	2	24772	931.944	996.767
1	24870	1083.209	1246.203	2	14401	1403.692	1509.006
1	20510	1255.724	1335.430	2	21666	1509.006	1637.356
1	03576	1335.430	1444.044	2	25679	1648.143	1745.250
1	02801	1444.044	1551.787				

Table 5 The Best Scheduled Result with Joint PSO Optimization in 100 MC Runs

Radar No.	NORAD ID	Beginning Time	End Time	Radar No.	NORAD ID	Beginning Time	End Time
1	13718	0.000	67.480	1	21087	1551.787	1618.959
1	25624	67.480	141.069	1	04507	1618.959	1735.184
1	25078	161.558	223.880	1	38745	1735.184	1800.000
1	25943	223.880	292.755	2	07768	0.000	61.796
1	14624	292.755	462.879	2	13148	61.796	140.881
1	20774	462.879	562.153	2	25943	140.881	200.881
1	27422	562.153	638.864	2	25676	266.376	539.984
1	25676	638.864	730.880	2	32264	539.984	616.302
1	05104	730.880	791.423	2	38744	616.302	717.228
1	22689	791.423	875.903	2	12903	723.024	822.453
1	21031	875.903	936.783	2	20233	822.453	931.944
1	27534	936.783	1068.241	2	24772	931.944	996.767
1	24870	1083.209	1246.203	2	14401	1403.692	1509.006
1	20510	1255.724	1335.430	2	21666	1509.006	1637.356
1	03576	1335.430	1444.044	2	25679	1648.143	1745.250
1	02801	1444.044	1551.787				

Table 6 Results of MC Simulation for Different Methods

Method Name	The Best	The Worst	The Average
The First Method	35.6568	35.6568	35.6568
Individual PSO Optimization	45.3073	36.2249	40.0822
Joint PSO Optimization	46.2060	42.1850	44.1912

comparison of the result also shows that performance of joint PSO optimization slightly outperforms the individual optimization.

The results are summarized in Table 6, including the best, worst and average of the total score for different methods. Compared with the first method, the improved performances of individual PSO optimization and joint PSO optimization range from 1.59% to 27.06% and from 18.31% to 29.59%, respectively. Results show that both the performance of individual PSO optimization and joint PSO optimization are always better than the first method, and the joint PSO optimization method can al-

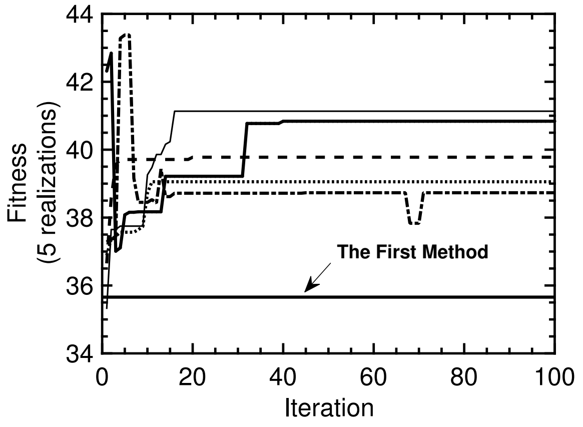

Fig. 4 Performance of individual PSO optimization.

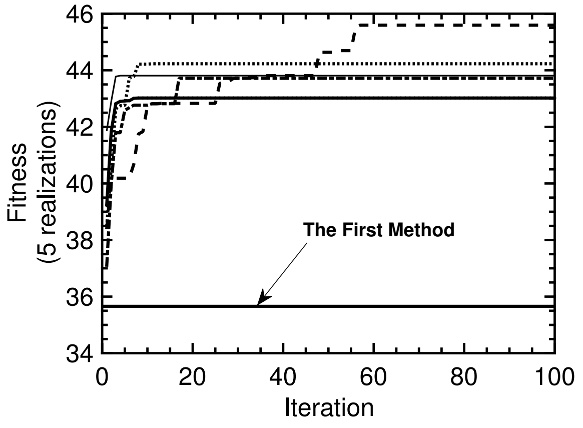

Fig. 5 Performance of joint PSO optimization.

ways guarantee a great (larger than 18%) performance improvement while the performance improvement by the individual PSO covered a very wide range. This is because the joint PSO optimization is a global optimization method while the individual PSO optimization is not really globally optimized. The performance variations of individual PSO and joint PSO optimizations are caused by several reasons. The primary one should be the initialization of particles, because the PSO algorithm cannot always achieve the global optimum but may fall into a local optimum in some cases.

6 CONCLUSIONS

In this paper, a new optimization technique is proposed for space debris surveillance network scheduling. This proposed algorithm is an integrated algorithm, which can optimally schedule the surveillance tasks automatically based on different evaluation criteria of the particles. Numerical experiments have been conducted with the proposed algorithm. It has been seen that the proposed technique with the PSO model not only computes fast but also gives much better performance (more than an 18% improvement) than the other method used in this work. The results show that the proposed algorithm can solve the task scheduling problem well. There may still have been some further improvements on the algorithm, such as observation geometry, which should be considered in scheduling as it may affect accuracy of the observed object's orbit. The computational complexity will increase along with increasing the number of surveillance tasks and so on. All of these issues will be our future research topics on surveillance task scheduling.

Acknowledgements This work was funded by the National Natural Science Foundation of China (Grant No. 11503044), and by the Young Researcher Grant of National Astronomical Observatories, Chinese Academy of Sciences.

References

Arumugam, M. S., & Rao, M. V. C. 2006, Discrete Dynamics in Nature and Society, 3, 638

Bansal, J. C., Singh, P. K., Saraswat, M., et al. 2011, in Proceedings of Third World Congress on Nature and Biologically Inspired Computing (NaBIC), 633

Clerc, M. 1999, in Proceedings of the 1999 Congress on Evolutionary Computation (CEC), 1957

Clerc, M., & Kennedy, J. 2002, IEEE Transaction on Evolutionary Computation, 6, 58

Duncan, M., & Wysack, J. 2011, in Proceedings of Advanced Maui Optical and Space Surveillance Technologies Conference, E53

Eberhart, R. C., & Shi, Y. 2001, in Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, 1, IEEE, 94

Herz, A., & Stoner, F. 2013, in Advanced Maui Optical and Space Surveillance Technologies Conference, E74

Hill, K., Sydney, P., Hamada, K., et al. 2010, in Paper AAS 10–150 presented at the AAS/AIAA Space Flight Mechanics Conference, February, 14

Kennedy, J., & Eberhart, R. 1995, in Neural Networks, 1995 Proceedings., IEEE International Conference on, 4, IEEE, 1942

Kessler, D. J., & Cour-Palais, B. G. 1978, J. Geophys. Res., 83, 2637

Krag, H., Beltrami-Karlezi, P., Bendisch, J., et al. 2000, Acta Astronautica, 47, 687

Liou, J.-C., & Johnson, N. L. 2006, Science, 311, 340

Marini, F., & Walczak, B. 2015, Chemometrics and Intelligent Laboratory Systems, 149, 153

Miller, J. G. 2007, Military Operations Research, 12, 57

Shi, Y., & Eberhart, R. 1998, in Evolutionary Computation Proceedings, 1998, IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, IEEE, 69

Stottler, R. 2012, in Proceedings of the AIAA Aerospace Conferences, 2434

Wilson, B. L. 2004, in Proceeding of 14th AAS/AIAA Space Flight Mechanics Conference, 377