DOUBLE LAYER PROGRAMMING MODEL
TO THE SCHEDULING OF REMOTE SENSING DATA
PROCESSING TASKS

MIN-FAN HE
School of Mathematics and Big Data
Foshan University
Foshan 528000, China

LI-NING XING
College of Systems Engineering
National University of Defense Technology
Changsha 410073, China

WEN LI
Business School of Hunan University
Changsha 410082, China

SHANG XIANG
College of Systems Engineering
National University of Defense Technology
Changsha 410073, China

XU TAN
School of Software Engineering
Shenzhen Institute of Information Technology
Shenzhen 518172, China

Abstract. Remotely sensed data are widely used in disaster and environment monitoring. To complete the tasks associated with processing these data, it is a practical and pressing problem to match the resources for these data with data processing centers in real or near-real time and complete as many tasks on time as possible. However, scheduling remotely sensed data processing tasks has two phases, namely, task assignment and task scheduling. This paper presents a model using bilevel optimization, which considers task assignment and task scheduling as a single problem. Using this architecture, a mathematical model for both levels of the problem is presented. To solve the mathematical model, this paper presents a cooperative coevolution algorithm that combines the advantages of a very fast simulated annealing algorithm with a learnable ant colony optimization algorithm. Finally, the effectiveness and feasibility of the proposed approach compared with the conventional method is demonstrated through empirical results.

2010 Mathematics Subject Classification. 90B35.
Key words and phrases. Bilevel optimization, scheduling problem, remote sensing data processing.
* Corresponding author: Wen Li.
1. Introduction. There has been a rapid increase in the number of onboard satellite sensors as a result of the emergence of satellite technologies. Remotely sensed data originate from multiple sources and are complex in structure. Hence, processing these data using scheduling algorithms of the Earth observing satellites is becoming more important [39, 20, 4, 5]. Recently, remotely sensed data have been widely used in disaster and environmental monitoring [27, 6, 9]. However, as the resolution of remotely sensed images becomes much higher, the automatic scheduling of remotely sensed data processing is becoming a challenge and a severe problem [21].

The main objective of remotely sensed data processing task comprehensive scheduling (RSDPTCS) is to assign remotely sensed data processing resources in a way that ensures the optimal use of limited resources to satisfy multifarious remotely sensed data processing requests from customers. RSDPTCS consists of two phases, namely, task assignment and task scheduling. Task assignment is responsible for rationally distributing the remotely sensed data tasks from the stations to the processing centers. Task scheduling is responsible for rationally distributing the remotely sensed data tasks to processing resources given the constraints of each processing center. The objective of the problem is to complete as many remotely sensed data processing tasks on time as possible. The research of this problem has important theoretical and practical value.

So far, the study of task scheduling and data processing has led to fairly mature approaches. Most of the research focuses on topics such as problem decomposition and recombination [34, 42, 15, 7, 25], algorithm optimization [31, 37, 16, 35, 28, 40, 18, 26], and model building [13, 23, 24, 10]. Over the past few decades, RSDPTCS has attracted increasing interest in the mission planning and scheduling area. Aloisio presented an architecture that shares the resources of different agencies and attaches importance to the access policies defined by the owners of the resources in a grid environment [1, 2, 3]. Karamchand established a cost model based on a centralized scheduling mode to evaluate continuous remotely sensed data processing [17]. Tehranian presented a robust distributed processing framework for processing remotely sensed data obtained from the NOAA satellite [30]. Xiang discussed the task management and scheduling techniques of the general high performance preprocessing system (GHIPS) and presented a task-level and algorithm-level parallel scheduling strategy to resolve the problem of remotely sensed data processing [36]. Zhu developed a framework for a distributed computing system for remotely sensed images to support distributed and high-performance computing for geospatial images. Zhu further designed a new algorithm that supports parallel computing with a dynamic workload balance for large images [41]. Chen and Dong designed a hierarchical cluster to create a global task scheduler for the workflow process, improved the remote sensing quantitative retrieval system performance [8, 11, 12]. Ma proposed task-tree based mosaicking for remote sensed imagery at a large scale. The approach uses dynamic DAG scheduling to obtain an optimized schedule on a multi-core cluster with minimal completion time [22]. Recent years, the method of machine learning has been gradually applied to solve the scheduling problem [29].

The research mentioned above only considers the problem of scheduling the remotely sensed data processing task and ignores the importance of task assignment. The goal of this study is to address both parts of the problem by building a mathematical model and using a bilevel optimization approach to propose an algorithm for solving it.
2. Problem formulations. The RSDPTCS problem has a hierarchical nature: the upper level is the task-assignment level and the lower level is the task-scheduling level. As Fig. 1 shows, these levels are interoperable and collaborative decision-making processes.

2.1. Upper-level optimization. The task assignment process that occurs at the upper level can be described as follows: processing centers form a cooperative plan to allocate the remotely sensed data processing task set on the basis of their own capabilities and the task constraints. By predicting the task transmission and completion times, a schedule is created that has the minimum expected penalty for tasks that are not completed on time.

Suppose $\mathcal{M}_{PC} = \{\mathcal{M}_{PC1}, \mathcal{M}_{PC2}, \ldots, \mathcal{M}_{PCm}\}$ denotes the set of processing centers, and the number of processing centers is $m$. Let $T = \{T_1, T_2, \ldots, T_{N_T}\}$ denote the set of tasks, and the number of tasks is $N_T$. Here, $T_i$ is a multi-component system:

$$T_i = \langle\text{TaskID}, \omega_i, Size_i, \text{Type}_i, T_{idf}, T_{idl}, L_{imin}, L_{imax}\rangle$$

where TaskID, $\omega_i$, and Type$_i$ denote the identifier, priority, and data type of the task, respectively, $T_{idf}$ and $T_{idl}$ denote the predefined time and deadline of the task,
and $L_{i \text{min}}$ and $L_{i \text{max}}$ denote the initial level and final level of task production. Let $\alpha_i^m = 0, 1$ denote the coefficient of transmission $T_i$ to $M_{PC_m}$. Then, $\beta_i^m = 1$ indicates that $M_{PC_m}$ can process $T_i$. Moreover, $t_{itr}$ is the transmission time of task $T_i$ and is $t_{itr} = t_{itr} + t_{delay}$, where $t_{itr} = N_i T_{bit} + t_{tran}$ is the transmission delay, $N_i$ is data size of task $T_i$, $L_i^m$ is the distance between the station and processing center, and $t_{delay}$ is the network delay, which is acquired by the adaptive model in [32]. Let $t_{icomp}$ denote the estimated completion time of $T_i$.

The mathematical model of task assignment is as follows:

$$f = \min \left( \sum_{i=1}^{N_T} \omega_i \max \left( 0, (T_{idl} - t_{itr} - t_{icomp}) \right) \right)$$  \hspace{1cm} (1)$$

$$\text{s.t. } \sum_{i=1}^{N_T} \alpha_i^m \beta_i^m \leq 1, \forall T_i \in T, m \in M_{PC}$$
$$\sum_{i=1}^{N_T} \alpha_i^m = 1, \forall T_i \in T, m \in M_{PC}$$ \hspace{1cm} (2)$$

Equation 1 expresses the objective of task assignment, i.e., to minimize the late-completion penalty. Equation 2 indicates that every task only can be transferred to one processing center. Here, $t_{icomp}$ can be acquired by low-level processing.

2.2. Low-level optimization. At the lower level, the task scheduling of remotely sensed data can be treated as the processing of data in a remotely sensed data processing grid consisting of resource processing nodes. The processing of each task involves the execution of dissimilar operations on the resource processing nodes in a sequence. Different resource nodes possess different processing capabilities (processing speed, and processing type). They are able to process only one operation at a time. Each operation utilizes only one resource node to finish the processing and is non-preemptive in nature. The schedule of remotely sensed data processing tasks is formed that minimizes the expected late-completion penalty, maximum computation time, and balances the resource node load.

Let $M_{PR} = \{M_{PR_1}, M_{PR_2}, \ldots, M_{PR_m^r}\}$ and $m^r$ denote the set and number of process resources respectively. $T^m = \{T_1^m, T_2^m, \ldots, T_n^m\} \subset T$ denotes the tasks received by processing center $M_{PC_m}$. Every task $T_i^m$ consists of a number of procedures, i.e.,

$$\text{Proc} = \{\text{ProcID}, \text{TaskID}, \omega_i^l, t_{impt}, t_{start}, t_{end}, \text{FarProcID}\}$$

where ProcID denotes $p(i, l, j)$ and $\omega_i^l$ is the priority of the procedure, which is initially assigned to be the priority of the task, but can change at a later time. In addition, $t_{impt}$ is the minimum processing time of $p(i, l, j)$ and is denoted as $t_{p(i, l, j)}$, $t_{start}$ is the start processing time of $p(i, l, j)$, and $t_{end}$ is the end processing time of $p(i, l, j)$. FarProcID is the procedure at the next level up. The meta-task is denoted by $(p(i, l, j), M_{PRm^r})$, $t_{B(p(i, l, j))}$ denotes the waiting time of $p(i, l, j)$, and $S_p(i, l, j, m^r) = \{0, 1\}$ denotes the coefficient of $p(i, l, j)$ processed on $m_{PRm^r}$.

The mathematical model of the task scheduling is expressed as:

$$f_1 = \min \left( \sum_{i=1}^{n^r} \omega_i^l \max \left( 0, (c_i - T_{idl}) \right) \right)$$  \hspace{1cm} (3)
\[
\min f_2 = \min \left( \max_{i=1}^{n'} c_i \right)
\]

\[
\min f_3 = \min (\sum_{k=1}^{m'} \Delta E_k)
\]

\[
\text{s.t. } \sum_{i=1}^{n} \sum_{l=0}^{t} \sum_{j=1}^{J} S_{p(i,l,j),k} \leq 1
\]

\[
\sum_{k=1}^{m} S_{p(i,l,j),k} = 1
\]

\[
t_s(p(i,l,j)) \geq 0
\]

\[
t_e(p(i,l,j)) > 0
\]

\[
t_{p(i,l,j)} = t_e(p(i,l,j)) - t_s(p(i,l,j)) > 0
\]

\[
t_{B(p(i,l,j))} = \begin{cases} 
    t_s(p(i,l,j)) - t_e(p(i,l,j-1)), & \forall j - 1 > 0 \\
    t_s(p(i,l,j)) - t_e(p(i,l-1,\max\{j\})), & \forall j - 1 \leq 0 \& l \neq 3 \\
    t_s(p(i,l,j)) - t_e(p(i,l-2,\max\{j\})), & \forall j - 1 \leq 0 \& l = 3 
\end{cases}
\]

\[
t_{B(p(i,0,j))} = 0
\]

where \(c_i\) denotes the completion time of \(T_i^{n'}\). Equations 3,4,5 express the objectives of the task scheduling: to minimize the late-completion penalty, minimize the maximum computation time, and balance the resource node load. Equation 6 indicates that each resource can only handle at most one procedure at a time. Equation 7 specifies that each procedure must be executed uninterrupted on a given resource. Equations 8,9,10 are the processing time constraints of the procedure. Equations 11,12,13 are the sequence constraints of the procedure.

3. **Cooperative evolution algorithm.** Figure 2 presents the cooperative coevolution algorithm proposed in this study for solving the mathematical model using bi-level programming. A cooperative coevolution algorithm is an algorithm combining the advantages of very fast simulated annealing and learnable ant colony optimization. The main part of the algorithm is the very fast simulated annealing algorithm. Learnable ant colony optimization is used to computation the completion task time and the scheduling plan.

The cooperative coevolution algorithm adopts parallel processing in the following two ways: 1) during task assignment, the data stations send the initial distribution plan (the remotely sensed data processing tasks) in parallel to the processing centers and 2) after initial distribution plan has been received, the processing centers computation the scheduling plans and lag costs in parallel and send feedback simultaneously. Based on the feedback, the algorithm adjusts the distribution plan until it generates the optimized plan.
3.1. Prediction of available time windows for the processing resources and center. The predicted available time windows of the processing resources and centers are based on the prediction of occupied resources and faults:

1) Occupied resources prediction: the processing resources are used to process the remotely sensed data task, so this processing time is occupied time.
2) Fault prediction: when processing resources are not operational, they cannot be used and the tasks cannot be processed. Faults can be predicted using a Markov model. By analyzing the fault logs of resources, dynamic status space without faults and transition matrix are built into the Markov model to predict the availability of the resources and center.

The available time windows of the processing resources and centers are the times other than the occupied and fault times.

3.2. Plan evolution. According to following rules for plan evolution, the cooperative coevolution algorithm modifies distribution plan as follows:

1) Choose the best solution \( S_{\text{best}} \);
2) Choose the worst tasks (i.e., a late-completion time greater than 0) in \( S_{\text{best}} \). If the quantity of tasks in \( S_{\text{best}} \) is \( S \), then the total quantity of the worst tasks is 0.1S;
3) Delete these worst tasks in \( S_{\text{best}} \) and redistribute them (to any processing centers except the current one) to generate the new distribution plan \( S_{\text{new}} \);
4) Compare with results of \( S_{\text{best}} \) and \( S_{\text{new}} \) and choose the better solution.

3.3. Learnable ant colony optimization algorithm. Based on the methods in [33, 38], a learnable ant colony optimization algorithm is suggested to solve the low-level model using a coarsely parallel computation:

1) Attractive factor: let \( \alpha_{p(i,l,j),m}^s \) be the attractive factor of an \( s \)-type pheromone for the selection of processing resource \( m \) for \( p(i,l,j) \) in available time window set \( \Pi_{p(i,l,j)} \). Hence,

\[
\alpha_{p(i,l,j),m}^s = \frac{\tau_{p(i,l,j),m}^s}{\sum_{h \in \Pi_{p(i,l,j)}} \tau_{p(i,l,j),h}^s}
\]

where \( \tau_{p(i,l,j),m}^s \) is the \( s \)-type pheromone.

2) Rejection factor: let \( \beta_{p(i,l,j),m}^s \) be the rejection factor of an \( s \)-type pheromone for the selection of processing resource \( m \) for \( p(i,l,j) \) in available time windows set \( \Pi_{p(i,l,j)} \). Hence,

\[
\beta_{p(i,l,j),m}^s = \sum_{h \neq s} \frac{\tau_{p(i,l,j),h}^s}{\sum_{h \in \Pi_{p(i,l,j)}} \tau_{p(i,l,j),h}^s}
\]

The learnable ant colony optimization algorithm is performed in two steps:

1) State transition rule

The operations are randomly allocated by the ants to the machines proportionally as per the state transition rule. The proposed method adopts a pseudorandom proportional state transition rule to enhance the efficiency of the optimization algorithm [14].

\[
Pr_{p(i,l,j),k}(t) = \begin{cases} 
J', & \text{if } q_0 \leq q \\
J, & \text{if } q_0 > q 
\end{cases}
\]

\[
J = \arg \max_{k \in \text{allow}(p(i,l,j),t)} \{ Pr_{p(i,l,j),k}(t) \}
\]
\[ J' = \begin{bmatrix} \tau_{p(i,l,j),k}(t) \alpha_{p(i,l,j),k}^{s} \beta_{p(i,l,j),k}^{s} / \sum_{h\notin \text{tabu}} \tau_{p(i,l,j),k}(t) \end{bmatrix} \left[ \begin{array}{c} P_i \\ Q_i \end{array} \right] \left[ \begin{array}{c} a \\ b \\ c \end{array} \right] \]  

where \( Pr_{p(i,l,j),k}(t) \) is the probability that procedure \( p(i,l,j) \) selects resource to process at time \( t \), \( \tau_{p(i,l,j),k}(t) \) is the pheromone, \( P_i \) is the priority of \( p(i,l,j) \), \( Q_i \) is the data capacity of \( p(i,l,j) \), \( a \) is the pheromone value, \( b \) is the priority value of \( p(i,l,j) \), \( c \) is the data size value of the corresponding task for \( p(i,l,j) \). Parameters \( a, b, \) and \( c \) are set heuristically.

2) Pheromone update rule

In the proposed method, we adopted a policy associated with a global update as well as local update. The pheromone is updated as follows:

\[ \tau_{p(i,l,j),k}(t + \Delta t) = (1 - \rho_{local}) \tau_{p(i,l,j),k}(t) + \rho_{local} \tau_{p(i,l,j),k}(\Delta t) \]  

\[ \tau_{p(i,l,j),k}(t + \Delta t) = \begin{cases} 
(1 - \rho_{global}) \tau_{p(i,l,j),k}(t) + \frac{Q}{J_{\text{total}}} & \text{if } J_{\text{total}} < J_{\text{worse}} \\
(1 - \rho_{global}) \tau_{p(i,l,j),k}(t) + \frac{Q}{J_{\text{worse}}} & \text{otherwise} 
\end{cases} \]

where \( \rho_{local} \) is local the volatility of the pheromones, \( \rho_{global} \) is the global volatility of the pheromones, \( Q \) is total pheromones, \( f \) is the objective function.

4. Results and discussions.

To evaluate the performance of the algorithm, we performed the following simulation experiments. Table 1 shows the distances between stations and processing centers used in the evaluation.

| station1 | station2 |
|---------|---------|
| PC1     | PC2     |
| PC3     | PC1     |
| 3420    | 3568    |
| 2954    | 2486    |
| 1471    | 1770    |

**Table 1.** Distances between stations and processing centers

Both stations have the same number of tasks \( (m = 50, 100, 200) \) and the processing centers have the same number of processing resources \( (n = 5, 10, 25) \).

In the present study, we set the processing start time of a task as a random number in the interval \([9:00:00, 9:00:10]\) and the deadline of a task as a random number in the interval \([10:30:00, 11:00:00]\) to generate tasks at different scales from 9:00:00 h to 11:00:00 h in a day. We assigned the processing resource nodes and capabilities randomly. We formulated nine problem instances and ran each of them 10 times. We compare our results to those obtained using the limited concentration model [19].

The empirical results obtained from the execution of the proposed approach are depicted in Figs. 3C6. The results are compared with respect to maximum computation time, average resource node load, late completion time, and processing time. The results indicate that the proposed approach is better than the representative model in the field (the limited concentration model).
Figure 3. Maximum computation time comparison

Figure 4. Average resource node load comparison

Figure 5. Late completion comparison
5. Conclusions. In the RSDPTCS problem, it is quite challenging to coordinate objects and constraints. This paper presented a model using bilevel optimization, which considers task assignment and task scheduling as a whole. At the upper level, processing centers make a cooperative plan to allocate the remotely sensed data processing tasks based on their own capabilities and constraints. At the lower level, the processing of each task involves the execution of dissimilar operations on the resource processing nodes in a sequence. We mathematically modeled the two levels of the problem, and proposed a cooperative coevolution algorithm that combines the advantages of very fast simulated annealing and learnable ant colony optimization to solve the proposed model. Experimental results comparing the results of the proposed method with those of the limited concentration model show the effectiveness and feasibility of our approach.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (No. 61773120, 61525304, 71690233, 71331008, U1501254 and 61472089). This work is also supported in part by the Foundation for the Author of National Excellent Doctoral Dissertation of China (2014-92), the Outstanding Youth Fund Project of Hunan Provincial Natural Science Foundation (S2015J5050), Guangdong Science and Technology Plan (2015B010131015 and 2015B010108006), and Shenzhen Basic Research Project for Development of Science and Technology (JCYJ20160530141956915).

REFERENCES

[1] G. Aloisio, M. Cafaro and R. Williams, The digital Puglia project: An active digital library of remote sensing data, in Proceedings of the 7th International Conference on High Performance Computing and Networking, HPCN-Europe, 1593 (1999), 563–572.
[2] G. Aloisio, M., Cafaro, R. Williams and P. Messina, A Distributed Web-Based Metacomputing Environment, in International Conference and Exhibition on High-Performance Computing and Networking, HPCN-Europe, 1225 (1997), 480–486.
[3] G. Aloisio, M. Cafaro, S. Fiore and G. Quarta, A Grid-Based Architecture for Earth Observation Data Access, in ACM Symposium on Applied Computing, Santa Fe, New Mexico, USA, 2005.
[4] G. Bai, L. Xing and Y. Chen, Scheduling multi-platforms collaborative disasters monitoring based on coevolution algorithm, *Research Journal of Chemistry and Environment*, **16** (2012), 43–50.

[5] G. Bai, L. Xing and Y. Chen, The knowledge-based genetic algorithm to the disasters monitoring task allocation problem, *Research Journal of Chemistry & Environment*, **16** (2012), 27–34.

[6] P. Caccetta, S. Collings, K. Hingee, D. McFarlane and X. Wu, Fine-Scale Monitoring of Complex Environments Using Remotely Sensed Aerial, Satellite, and Other Spatial Data, *2011 International Symposium on Image and Data Fusion*, Tengchong, Yunnan, (2011), 1–5.

[7] Y. Cao, S. Wang, L. Kang and Y. Gao, A TQCS-based service selection and scheduling strategy in cloud manufacturing, *The International Journal of Advanced Manufacturing Technology*, **82** (2016), 235–251.

[8] Z. Chen, Y. Xue, J. Dong, J. Liu and Y. Li, The task scheduling for Remote Sensing Quantitative Retrieval based on hierarchical grid computing platform, *2012 IEEE International Geoscience and Remote Sensing Symposium*, Munich, (2012), 487–490.

[9] P. Chinnasamy and M. G. Sunde, Improving spatiotemporal groundwater estimates after natural disasters using remotely sensed data—a case study of the Indian Ocean Tsunami, *Earth Science Informatics*, **9** (2016), 101–111.

[10] S. D. Dao, K. Abhary and R. Marian, An integrated production scheduling model for multi-product orders in VCIM systems, *International Journal of System Assurance Engineering and Management*, **8** (2017), 12–27.

[11] J. Dong, Y. Xue, Z. Chen, H. Xu and Y. Li, Analysis of remote sensing quantitative inversion in cloud computing, *2011 IEEE International Geoscience and Remote Sensing Symposium*, Vancouver, BC, (2011), 4348–4351.

[12] J. Dong, Y. Xue, Z. Chen, H. Xu, Y. Li and C. Wu, A study of grid workflow dynamic customization for remote sensing quantitative retrieval, *2012 IEEE International Geoscience and Remote Sensing Symposium*, Munich, (2012), 5382–5385.

[13] Y. Guan, W. Xiao, R. Cheung and C. Li, A multiprocessor task scheduling model for berth allocation: Heuristic and worst-case analysis, *Operations Research Letters*, **30** (2002), 343–350.

[14] R. He and L. Xing, A learnable ant colony optimization to the mission planning of multiple satellites, *Research Journal of Chemistry and Environment*, **16** (2012), 18–26.

[15] T. Jena and J. R. Mohanty, GA-based customer-conscious resource allocation and task scheduling in multi-cloud computing, *Arabian Journal for Science and Engineering*, (2017), 1–16.

[16] A. Kamalinia and A. Ghaffari, Hybrid task scheduling method for cloud computing by genetic and DE algorithms, *Wireless Personal Communications*, to be published.

[17] S. Karamchand, *Operator Scheduling in a Distributed Stream Management System for Remotely Sensed Imagery*, Ph.D thesis, University of California in Davis, USA, 2006.

[18] I. Kucukkoc and D. Z. Zhang, Integrating ant colony and genetic algorithms in the balancing and scheduling of complex assembly lines, *The International Journal of Advanced Manufacturing Technology*, **82** (2016), 265–285.

[19] W. Li, Y.-W. Chen, J.-F. Li and P. Gao, Remotely sensed data processing task scheduling based on limited concentration model, *PIC2010*, Shanghai, Chinn, (2001), 15–18.

[20] J. F. Li and L. Xing, Integrative Forest Fire Monitoring System Framework Design Aim to Early Warning Tasks, *Disaster Advances*, **5** (2012), 726–729.

[21] H. Liu, Y. Fan, X. Deng and S. Ji, Parallel Processing Architecture of Remotely Sensed Image Processing System Based on Cluster, *2009 2nd International Congress on Image and Signal Processing*, Tianjin, (2009), 1–4.

[22] Y. Ma, L. Wang, A. Y. Zomaya, D. Chen and R. Ranjan, Task-Tree Based Large-Scale Mosaicking for Massive Remote Sensed Imagery with Dynamic DAG Scheduling, *IEEE Transactions on Parallel and Distributed Systems*, **25** (2014), 2126–2137.

[23] P. Myszkowski, M. E. Skowroński, L. P. Olech and K. Osiłikiptono, Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem, *Soft Computing*, **19** (2015), 3599–3619.

[24] S. Nguyen, Y. Mei and M. Zhang, Genetic programming for production scheduling: A survey with a unified framework, *Complex & Intelligent Systems*, **3** (2017), 41–66.

[25] S. Nguyen, A learning and optimizing system for order acceptance and scheduling, *The International Journal of Advanced Manufacturing Technology*, **86** (2016), 2021–2036.
[26] S. K. Panda, I. Gupta and P. K. Jana, A multiprocessor task scheduling model for berth allocation: heuristic and worst-case analysis, *Information Systems Frontiers*, to be published.

[27] T. J. Pultz and R. A. Scofield, Applications of remotely sensed data in flood prediction and monitoring: Report of the CEOS Disaster Management Support Group flood team, *IEEE International Geoscience and Remote Sensing Symposium*, 2 (2002), 768–770.

[28] M. A. Salido, J. Escamilla, A. Giret and F. Barber, A genetic algorithm for energy-efficiency in job-shop scheduling, *The International Journal of Advanced Manufacturing Technology*, 85 (2016), 1303–1314.

[29] Y Shen, N Zhao, M Xia and X Du, A deep Q-learning network for ship stowage planning problem, *Polish Maritime Research*, 24 (2017), 102–109.

[30] S. Tehraniana, Y.-S. Zhao, T. Harveya, A. Swaroopa and M. Keith, A robust framework for real-time distributed processing of satellite data, *Journal of Parallel Distributed Computing*, 66 (2006), 403–418.

[31] J. T. Tsai, J. C. Fang and J. H. Chou, Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm, *Computers & Operations Research*, 40 (2013), 3045–3055.

[32] Z. Wang, *Predictive Control of Networked Control Systems*, Ph.D thesis, Zhe Jiang University, Zhe Jiang, China, 2008.

[33] L. Wen, G. Peng, C. Ying-Wu and L. Ju-Fang, Approach of remotely sensed data processing task scheduling problem based on ant colony optimization, *Proceedings of 2011 International Conference on Modelling, Identification and Control*, Shanghai, (2011), 532–536.

[34] C. S. Wong, F. T. S. Chan and S. H. Chung, A joint production scheduling approach considering multiple resources and preventive maintenance tasks, *International Journal of Production Research*, 51 (2013), 883–896.

[35] S. Wu, P. Zhang, F. Li, F. Gu and Y. Pan, A hybrid discrete particle swarm optimization-genetic algorithm for multi-task scheduling problem in service oriented manufacturing systems, *Journal of Central South University*, 23 (2016), 421–429.

[36] B. Xiang, G. Q. Li, D. S. Liu and J. S. Li, Research on task management and scheduling of high performance remote sensing satellite ground pre-processing system, *Journal of Astronautics*, 29 (2008), 1443–1446.

[37] Y. Xu, K. Li, J. Hu and K. Li, A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues, *Information Sciences*, 270 (2014), 255–287.

[38] K. Yang and L. Xing, The Learnable Ant Colony Optimization to Satellite Ground Station System Scheduling Problems, *Przeglad Elektrotechniczny*, 88 (2012), 62–65.

[39] F. Yao and L. Xing, The Model, Algorithm and Application to Scheduling Problem of Agile Earth Observing Satellites, *Disaster Advances*, 5 (2012), 1112–1116.

[40] H. Zheng, L. Wang and X. Zheng, Teaching–learning-based optimization algorithm for multi-skill resource constrained project scheduling problem, *Soft Computing*, 21 (2017), 1537–1548.

[41] M. Zhu, G. Wang and T. Oyana, Parallel spatiotemporal autocorrelation and visualization system for large-scale remotely sensed images, *The Journal of Supercomputing*, 59 (2012), 83–103.

[42] X. Zuo, G. Zhang and W. Tan, Self-adaptive learning pso-based deadline constrained task scheduling for hybrid iaas cloud, *IEEE Transactions on Automation Science and Engineering*, 11 (2014), 564–573.

Received June 2017; revised December 2017.

E-mail address: 2482042568@qq.com
E-mail address: xinglining@gmail.com
E-mail address: rainstorm0329@hotmail.com
E-mail address: xiangshang165@163.com
E-mail address: tanxu_nudt@yahoo.com