Multi-Satellite Task Allocation Algorithm for Earth Observation

Pratik Kumar Sinha
National Institute of Technology Durgapur
Durgapur, West Bengal, India
Email: pratik.sinha27@gmail.com

Animesh Dutta
National Institute of Technology Durgapur
Durgapur, West Bengal, India
Email: animesh.dutta@it.nitdgp.ac.in

***Abstract*—The recent increase in the number of satellites in orbit has made their control by ground stations inefficient. Moreover, the delays in ground station to satellite communication adversely affects the functioning and efficiency of satellite systems. The environment of the satellites is also highly dynamic. Thus, the allocation of tasks to satellites by ground stations is no longer feasible. In order to overcome these issues, this paper presents a Multi-Agent based modeling of satellite systems. The satellites are modeled as autonomous agents and can collaborate with other satellite agents in the multi-agent system. The tasks can thereby be allocated and executed by the agents, without repetitive involvement of ground stations. The allocation of tasks by the agents is modeled as a distributed constraint optimization problem. A distributed iterative algorithm, suited to the satellite domain, has been proposed to solve the optimization problem. The paper also presents an algorithm for agent coordination and negotiation. The multi-agent model along with the proposed algorithms eliminate the requirement of ground station control. This has been evaluated empirically and the results have been presented.**

***Index Terms*—Agent, Multi-agent, Earth observation, Satellite, DCOP, Constraint optimization**

I. Introduction

One of the most common application of a satellite is its use in the observation of Earth for various purposes such as weather prediction, disaster management etc. The quality of the images taken by the satellites plays a crucial role in determining the quality of information deduced from them. The satellites in the Low Earth Orbit (LEO) are known to provide better images than those in higher orbits and are therefore more preferred. Also, traditional satellites are now being replaced by small satellites. They are much cheaper, resulting in a sharp increase in the number of satellites in the LEO orbit. However, LEO satellites, owing to their high orbital velocity (around 7.5 Km/s), have very low pass over times, making task allocation difficult. Moreover, Earth observation tasks typically require the participation of multiple satellites working in dynamic environments which are prone to faults. Thus, the traditional method of manually allocating tasks to satellites (offline planning) cannot be used [1].

The task allocation problem in LEO satellites can be addressed by modeling the satellites as Agents in a Multi-Agent System (MAS). The multi-agent paradigm is commonly used in distributed system which require collaborative automation. Each satellite agent would be able to sense and adapt to the dynamics of the environment. They would also communicate and collaborate with other satellite agents (in the multi-agent system) to complete the tasks assigned. This type of modeling involves two parts. First, each satellite must be modeled as an agent and their behavior must be defined. Then, the protocols for communication, negotiation and collaboration among the agents have to be established. The Distributed Constraint Optimization (DCOP) algorithms are widely used to handle the negotiation amongst agents in a MAS. The modeling of a satellite system as a MAS may seem similar to modeling a Wireless Sensor Network (WSN). However, the two differ over the constraints involved, specifically that the satellites move in fixed orbits and their motion cannot be arbitrarily altered. Moreover, agent communication in satellite systems is different from that in WSNs. In this context, this paper presents the modeling of Earth observation LEO satellites as agents in a MAS along with algorithms to handle the negotiation and collaboration among the satellite agents.

The remainder of this paper is organized as follows: section II discusses the related work and presents a brief review of literature, section III defines the scope of this work, the system model and the problem statements are given in sections IV and V, the proposed algorithms are given in section VI. The simulation results and concluding remarks are given in section VII.

II. Related Work

The area of task allocation and task scheduling for satellites has received a lot of attention from the research community in the last few years. Many algorithms have been proposed for generating efficient schedules. Chen et al. [2] have proposed an algorithm for multi-satellite task scheduling. It uses the particle swarm optimization to improve the rate of convergence and avoid stagnation in genetic algorithm. Wu et al. [3]–[5] have proposed algorithms based on ant colony optimization with local iterative search. The algorithms use clustering and prioritization of tasks improve the schedule quality. Dynamic scheduling of observation tasks is done by Dishan et al. [6] using an integer programming model. The tasks are divided into sets with overlapping times of task arrival. Each set is individually scheduled, thereby changing the dynamic scheduling problem into several static scheduling problems. Wang et al. [7] presented a mathematical model for dynamic (real-

978-1-5090-2597-8/16/$31.00 ©2016 IEEE

time) task scheduling for multi-satellite systems. The work includes a heuristic algorithm based on the genetic algorithm. It generates schedules with optimal resource utilization and maximum profit.

Several DCOP algorithms have been proposed for handling coordination and negotiation in multi-agent systems. In the domain of satellite systems modeled as MAS, both time and resources are constrained. These systems specifically have constraints over computational, storage, communication and power resources. Julius Pfrommer [8] has presented a modified max-sum algorithm for loopy constrained graphs. The proposed method converts a loopy constraint graph into a factor tree graph. The max marginal can then be computed accurately. Also, the agents which cannot communicate directly are disconnected in the factor tree graph. This removes the necessity for their intercommunication. A dynamic programming (DP) based algorithm was proposed by Vinyals et al. [9]. The authors use heuristics to convert the constraint graphs into junction trees. The local objective functions of the agents are propagated throughout the junction tree. The agents can then infer the global objective function. Subsequently, they compute the respective optimal assignments. A DCOP formulation for sensors networks where energy resources are constrained is presented by Farinelli et al. [10]. The max-sum algorithm is used with factor graphs generated by focusing on only the critical sections of the constraint graph. This reduces the computational and communication complexity. The solution quality does not degrade even with very large numbers of agents. Zivan et al. [11] have explored multiple approximation DCOP algorithms. They have used exploration methods within a restricted solution space. This overcomes the drawback of such algorithms getting stuck at the local optima.

III. Scope of the Work

Over the past few years, a large amount of research work has been directed towards finding an efficient solution to the task allocation problem for Earth observation satellites. However, the problems specific to multi-satellite task allocation have not been addressed sufficiently. Most of the existing algorithms perform offline scheduling [2]–[5]. This is undesirable in multi-satellite systems. The algorithms which perform dynamic task allocation [6], [7] do not consider the related constraints properly. Also, the multi-agent based models [12] and related algorithms focus on working with specific problems such as constrained communication [8], speed, cost [9]–[11] or the quality of the solution [13]. Overcoming these challenges together is crucial to finding a good solution. With this background, this work presents the modeling of a multi-satellite system as a multi-agent system. The MAS model takes the important constraints of the multi-satellite domain into consideration. Additional constraints may be easily added to it. The paper also proposes algorithms for negotiation and collaboration among the agents using the DCOP paradigm. The algorithms work on the principle of the Max Gain Message (MGM) algorithm [14] and belong to the incomplete (approximate) class of DCOP algorithms. The proposed algorithms are anytime algorithms. They iteratively attempt to improve the quality of the solution.

IV. System Model

A set of tasks T consisting of k number of individual observation tasks T_i is considered:

$$T = \{T_i | 1 \leq i \leq k\} \tag{1}$$

The tasks in T are independent of each other and can be performed simultaneously. Each task T_i has its individual requirements and specifications. An observation task involves capturing one or more images. Every observation task will have the following properties:

1) Budget - Each task T_i has a fixed maximum budget Bud_i. It specifies the maximum cumulative cost that can be incurred by the agents while performing the task. The budget may be specified in terms of financial cost, cost of effort, cost of resources (such as power) etc. or a combination of various types of costs represented as a common unit.
2) Deadline of Task - Every task T_i has a deadline DoT_i. It is the time before which the task T_i must be completed by the agent or agents to whom the task is allocated. The deadline is specific to the task only and cannot be modified by the agents.

Each task T_i consists of c_i number of independent observation sub-tasks T_{i-j}. An observation sub-task involves the capturing only one image.

$$T_i = \{T_{i-j} | 1 \leq j \leq c_i\} \tag{2}$$

It is assumed here that the sub-tasks do not have any dependencies among them and can be performed in any order. However, if the sub-tasks are interdependent, this assumption can be omitted by considering the dependencies as properties of the specific sub-tasks or as constraints. Then, the dependency would be considered and appropriately handled by the agents during sub-task allocation. The sub-tasks T_{i-j} inherit the properties of the task T_i. An observation sub-task has the following properties:

1) Budget - The budget Bud_{i-j} allocated to the sub-task $T_{i-j} \in T_i$ is the maximum cost that can be incurred while performing the sub-task. The total budget of all sub-tasks T_{i-j} is less than or equal to the budget allocated to the task T_i.
2) Deadline - The sub-task T_{i-j} must be completed before the deadline DoT_{i-j}. The deadline of any subtask T_{i-j} must not exceed the deadline DoT_i of the task T_i.
3) Location - The location on the Earth, which has to be observed for sub-task T_{i-j}, is specified as a latitude Lat_{i-j} and a longitude Lon_{i-j}.
4) Start Time - The time ST_{i-j} at which the observation for sub-task T_{i-j} may begin.
5) End Time - The time ET_{i-j} by which the observation for sub-task T_{i-j} must finish.

A satellite system consisting of a set A with n number of satellites is considered. Each satellite in set A is modeled as an agent A_p.

$$A = \{A_p | 1 \leq p \leq n\} \tag{3}$$

Each agent A_p has the following properties:

1) Intention - The intention of an agent A_p, at time t, towards accepting a sub-task T_{i-j} with the goal of completing it. An agent may calculate its intention towards a sub-task T_{i-j} based on the sub-task properties such as the target location (Lat_{i-j},Lon_{i-j}), budget (Bud_{i-j}), deadline (DoT_{i-j}) etc. and its own capabilities.

$$intention^t_{A_p}(T_{i-j}) = \begin{cases} 1, & \text{if } A_p \text{ is willing} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

2) Positive Intention Sub-tasks - Each agent A_p maintains a set PI_{A_p} consisting of sub-tasks which are not yet allocated to any agent but are being negotiated for allocation. The set PI_{A_p} consists of only those unallocated sub-tasks towards which the agent A_p has intention 1. The task list of any agent is always a subset of a task set T_i.

$$PI_{A_p} \subset T_i, T_i \in T \tag{5}$$

3) Allocated Sub-tasks - The set AST_{A_p} consists of the sub-tasks allocated to agent A_p.
4) Known Tasks - The set KT_{A_p} consists of the tasks which an agent A_p has received information about, such that $KT_{A_p} \subset T$. This set is used any an agent to filter duplicate information about duplicate tasks.
5) Expected Cost - The cost which is expected to be incurred if agent A_p performs the sub-task T_{i-j} and the agent accepts the task at time t. This estimate is calculated by the agent A_p based on the its own capabilities and the properties of the sub-task.

$$expc^t_{A_p}(T_{i-j}) \rightarrow \Re \tag{6}$$

6) Expected Time of Task Completion - The time at which the sub-task T_{i-j} is expected to be completed if agent A_p accepts the task at time t. It is dependent on the capabilities of the agent A_p and the properties of the sub-task T_{i-j}.

$$expt^t_{A_p}(T_{i-j}) \rightarrow \Re \tag{7}$$

The (satellite) agents are constantly in motion, i.e. they are moving around the Earth in their fixed orbits. Each agent $A_p \in A$ can send messages directly to other agents within its communication range through the Inter-satellite Links (ISLs). The communication range of a satellite agent A_p at any particular time t is CR^t_p. The unit of measurement can be omitted by using a uniform system. Any two satellites which are within the communication range of each other would be able to communicate. The communication is disrupted when either or both move out of the others range.

The communication network is dynamic, i.e. the topology of the communication network or the availability of links between the agents changes over time. In such a case, maintaining coordination amongst all the agents is difficult (or may be impossible altogether). However, instead of considering all agents at the same time, only those agents which are interconnected (either directly or through other agents) are considered. This interconnection may last for a finite period of time. Consider a subset B of the agent set A, such that $B_t \subset A$ at time t. The physical distance between two agents $A_p \in B_t$ and $A_q \in B_t$, at time t, is given by the function:

$$dist(p, q, t) \rightarrow \Re \tag{8}$$

The communication network graph for the agents in set B_t at time t is:

$$G_t = \{V_t, E_t\} \tag{9}$$

$$E_t = \{< A_p, A_q > | A_p, A_q \in B_t\} \tag{10}$$

where V_t is the set of vertices representing the agents in B_t and E_t is the set of edges representing the communication links amongst the agents, at time t. Each edge $< A_p, A_q >\in E_t$ shows the presence of a two way communication links between the agents $A_p, A_q \in B_t$ such that:

$$\begin{aligned} CR^t_p &\geq dist(p, q, t) \\ CR^t_q &\geq dist(q, p, t) \end{aligned} \tag{11}$$

The communication graph G_t is valid only till all its vertices V_t and edges E_t remain unchanged. A time period c is considered during which G_t remains valid. Then,

$$G_{t+d} = G_t \tag{12}$$

where $0 \leq d \leq c$.

A task may be generated and transmitted to a satellite agent $A_p \in A$ by a ground station $GS_k \in GS$. The set GS consists of m ground stations which may interact with the agents in set A. A ground station $GS_k \in GS$ may also receive data from the agents in set A.

$$GS = \{GS_k | 1 \leq k \leq m\} \tag{13}$$

V. Problem Formulation

An observation task is generated by a ground station $GS_{source} \in GS$ and transmitted to an agent $A_{origin} \in A$. The agent A_{origin} performs the following actions on receiving the task from a ground station:

1) Broadcasting the task information to nearby agents, including itself.
2) Coordinating the negotiation and sub-task allocation process. It may also participate in the negotiation.

When an agent A_p receives a task from another agent, it performs two actions:

1) Calculating its intentions towards the sub-tasks of the task.
2) Negotiating with other agents for sub-task allocation.

Based on the above, the following problems have been identified:

1) Identification of neighbors - An agent A_p may have intention 1, from Eq. (4), towards a sub-task T_{i-j}. Another agent A_q may also have intention 1 towards the same sub-task T_{i-j}. In this case, agents A_p and A_q are neighbors. However, since the two agents calculated their intentions towards the sub-tasks individually, they are unaware of each other's intentions. Consequently, they cannot identify each other as neighbors. The identification of neighbors is important in multi-agent systems for negotiation and coordination among the agents. It is, therefore, necessary that the agents identify their respective neighbors.
2) Sub-task allocation - Once the agents have established their intentions towards the sub-tasks and have identified their neighbors, they must negotiate and coordinate among themselves to find the optimal sub-task allocation.

The allocation of sub-tasks is subject to the following constraints:

1) Total Cost of an Assignment - The total cost of an assignment is the sum of the expected costs of all allocated sub-tasks T_{i-j} present in the task list AST_{A_p} of agents A_p. This cost should not exceed the total budget Bud_i of the task T_i at any time t. The level of satisfaction of the constraint is inversely proportional to total cost, as given in Eq. (15).

$$Cost(T_i, A, t) = \left(\sum_{A_p \in A} \sum_{T_{i-j} \in AST_{A_p} \cap T_i} expc^t_{A_p}(T_{i-j}) \right) - Bud_i \quad (14)$$

2) Expected Time of Completion - The expected time of completion of a task T_i is the maximum expected time at which all sub-tasks T_{i-j} will complete. The expected time of completion of a task T_i should not exceed its deadline DoT_i. The constraint is better satisfied if all sub-tasks are completed as soon as possible, as given in Eq. (16).

$$Time(T_i, A, t) = \left(\max_{A_p \in A} \max_{T_{i-j} \in AST_{A_p} \cap T_i} expt^t_{A_p}(T_{i-j}) \right) - Bud_i \quad (15)$$

3) Exclusivity of allocation - A sub-task T_{i-j} allocated to an agent A_p should not be allocated to any other agent in set A. Also, a sub-task should not remain unallocated. In other words, a sub-task should be allocated to only one agent, as given in Eq. (18).

$$AllocCount(T_{i-j}) = \sum_{A_p \in A} \left| AST_{A_p} \cap \{T_{i-j}\} \right| \quad (16)$$

$$Exclusive(T_{i-j}, A) = \begin{cases} |A|, \text{if} AllocCount(T_{i-j}, A) < 1 \\ 1, \text{if } AllocCount(T_{i-j}, A) = 1 \\ AllocCount(T_{i-j}, A), \text{otherwise} \end{cases} \quad (17)$$

Based on the system defined in the previous section, the goal is to assign each sub-task T_{i-j} of a task T_i to an agent such that the assignment is optimal. The value generated by the objective function given in Eq. (19) must be minimized to obtain optimal assignments.

$$Optimal(T_i, A, t) = Cost(T_i, A, t) + Time(T_i, A_t) + \sum_{T_{i-j} \in T_i} Exclusive(T_{i-j}, A) \quad (18)$$

4) Communication constraints - An agent A_p may require to communicate with an agent $A_q \in A$. However, if the inquality given in 11 is not true, they would not be able to communicate. In such cases, A_{origin} would have to act as a message forwarding node. This would significantly increase the workload of A_{origin} with increase in the number of messages to be forwarded, which is undesirable. Therefore, coordination and negotiation algorithms must ensure that minimum number of messages require forwarding.

VI. Proposed Algorithms

An observation task T_i is sent by a ground station GS_k to a satellite agent A_{origin} within its communication range. The agent A_{origin} broadcasts the task information to nearby agents. When an agent A_p receives the task information, it executes Algorithm 1. In lines 4-7, A_p calculates its intentions towards every sub-task $T_{i-j} \in T_i$. The intentions are sent to A_{origin} in line 8 and a list of neighbors is received in line 11. It is possible that not all agents will have the same response times. Therefore, in line 12, A_p waits for a timeout, which is calculated as four times the propagation time to the farthest agent. If a new set of neighbors is received before the timeout, A_p updates its neighbor list and resets the timeout. Once the timeout expires, A_p starts updating the communication graph in line 13. The negotiations for task allocation is started simultaneously in line 14 by executing Algorithm 3.

Algorithm 2 is executed each time A_{origin} receives a list of intentions from an agent. It calculates the neighbor list for each known agent in lines 3-9 and sends the updated neighbor lists to all known agents.

Once all agents have identified their respective neighbors, each agent executes Algorithm 3 for negotiating task allocation. It is a distributed algorithm based on the Maximum Gain Message (MGM) algorithm [14]. MGM is used for finding approximate solutions to DCOP problems. The algorithm has been modified to suit the satellite domain. Improvements have been made for faster and better convergence towards finding a solution. The original MGM used only Gain messages. The proposed algorithm uses Gain and Loss values in lines 18-32 for improving the solution. The proposed algorithm also

Algorithm 1 DiscoverNeighbors

Input: An observation task $T_i \in T$ with a set of neighbors

$p \leftarrow$Identification number of the agent $A_p \in A$
$timeout \leftarrow (4 * CR_p^t)$/Propagation Speed
if $T_i \notin KT_{A_p}$ **then**
  for $\forall T_{i-j} \in T_i$ **do**
    if $intention_{A_p}^t(T_{i-j}) = 1$ **then**
      $PI_{A_p} \leftarrow PI_{A_p} \cup \{T_{i-j}\}$
    end if
  end for
  $send(Origin, A_p, PI_{A_p})$
end if
$neighbors_{A_p} \leftarrow receive(Origin)$
Wait for $timeout$ expiry
$UpdateCommGraph(neighbors_{A_p})$
$TaskNegotiation()$

Algorithm 2 CalculateNeighbor

Input: PI_{A_p}, $agents$, $neighbors$

$agents \leftarrow agents \cup \{A_p\}$
if $count(agents) > 1$ **then**
  for $\forall T_{i-j} \in PI_{A_p}$ **do**
    for $\forall A_q \in subtasks[T_{i-j}]$ **do**
      $neighbors[A_p][T_{i-j}] \leftarrow neighbors[A_p][T_{i-j}] \cup \{A_q\}$
      $neighbors[A_q][T_{i-j}] \leftarrow neighbors[A_q][T_{i-j}] \cup \{A_p\}$
    end for
    $subtasks[T_{i-j}] \leftarrow subtasks[T_{i-j}] \cup \{A_p\}$
  end for
  for $\forall A_p \in agents$ **do**
    $send(A_p, Origin, neighbors[A_p])$
  end for
end if

ensures, in lines 8-16, that the quality of solution does not degrade in any iteration. The various functions used in the algorithm are described in Table I.

VII. Simulation Results and Conclusion

A simulation environment was set up using actual satellite data of the Weather and Earth Resources LEO satellites [15]. A set of Earth observation tasks was generated. The objective of each task was to obtain images of a certain area of the Earth multiple times between a stipulated start and end time. Each task was divided into ten to fifty sub-tasks with the objective of capturing one image. The initial request to perform a task is sent by a ground station to any satellite agent within its range. The agents execute Algorithms 1 and 3 to construct the constraint and communication graphs and to allocate tasks. The quality of the allocation and the number of times the loop of Algorithm 3 (lines 3-32) was iterated has been shown in Fig. 1. The optimal quality has been given for reference. The graph also shows the total number of messages exchanged among the agents. The increase in the number of satellites

Algorithm 3 TaskNegotiation

Input: $neighbors_{A_p}$, PI_{A_p} of agent $A_p \in Active$

1: Generate random initial $assignment_{A_p}$
2: Set $quality_{A_p}$ and $quality_{old}$ to minimum
3: **repeat**
4:   Send $assignment_{A_p}$ and $quality_{A_p}$ to all neighbors $A_q \in neighbors_{A_p}$
5:   Receive $assignment_{A_q}$ and $quality_{A_q}$ from all neighbors $A_q \in neighbors_{A_p}$
6:   $quality_{new} \leftarrow \sum_{A_q \in neighbors_{A_p}} quality_{A_q}$
7:   $value \leftarrow CalcValue(assignment_{A_p})$
8:   **if** $quality_{new} > quality_{old}$ **then**
9:     **if** $Node_{best} = A_p$ **then**
10:      $assignment_{A_p} \leftarrow assignment_{new}$
11:    **end if**
12:    $quality_{old} \leftarrow quality_{new}$
13:    $assignment_{new} \leftarrow GenAsst(assignment_{A_p})$
14: **else**
15:    $assignment_{new} \leftarrow GenAsst(assignment_{new})$
16: **end if**
17: $value_{new} \leftarrow CalcValue(assignment_{new})$
18: $[Gain, Loss] \leftarrow CalcGainLoss(value, value_{new})$
19: Send $assignment_{new}$, $Gain$ and $Loss$ to all neighbors $A_q \in neighbors_{A_p}$
20: Receive $assignment_{A_q}$, $Gain_{A_q}$ and $Loss_{A_q}$ from all neighbors $A_q \in neighbors_{A_p}$
21: $Gain_{max}$, $Loss_{min}$, $Node_{best} \leftarrow$ Values of A_p
22: **for** $\forall A_q \in neighbors_{A_p}$ **do**
23:    **if** $Gain_{A_q} > Gain_{max}$ **then**
24:      $Gain_{max}$, $Loss_{min}$, $Node_{best} \leftarrow$ Values of A_q
25:    **else if** $Gain_{A_q} = Gain_{max}$ **then**
26:      **if** $Loss_{A_q} < Loss_{min}$ **then**
27:        $Gain_{max}$, $Loss_{min}$, $Node_{best} \leftarrow$ Values of A_q
28:      **else**
29:        $Gain_{max}$, $Loss_{min}$, $Node_{best} \leftarrow$ Values of A_p or A_q
30:      **end if**
31:    **end if**
32: **end for**
33: $quality_{A_p} \leftarrow CalcQuality(neighbors_{A_p}, Node_{max})$
34: **until** All neighbors $A_q \in neighbors_{A_p}$ are in range {By Eq. 11}

and Earth observation tasks has made offline allocation of tasks to satellites inefficient and difficult. The multi-agent technology has promising aspects in solving the problems in this domain. In this paper, a MAS modeling of satellites has been presented. The algorithms for defining the behavior of the satellite agents towards task allocation have also been proposed. They do not require any pre-processing [8] on the constraint or communication graphs. This gives the algorithms additional time to complete more number of iterations and thereby improve the solution quality. The satellites perform the task allocation without requiring ground station control.

TABLE I
LIST OF FUNCTIONS USED IN ALGORITHM 3

Function Name	Description
UpdateCommGraph	Checks for direct communication links with the given agents and updates the communication graph.
CalcValue	Tests the given assignment against the constraints given by Eq. 14, 15 and 17
GenAsst	Finds the assignment which best satisfies the constraints given by Eq. 14, 15 and 17. An assignment passed to the function is excluded from the assignment domain.
CalcGainLoss	Finds the Gain and Loss values between two assignment values.
CalcQuality	Finds the quality of an assignment based on the objective function given in Eq. 18.
Send	Send the given message to the destination. If a direct communication link is not available, i.e. Eq. 11 is not true, the message if sent to A_origin for forwarding.

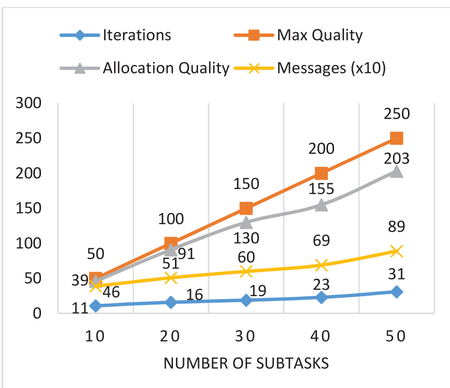

Fig. 1. Simulation result

On receiving a task, the agents negotiate among themselves to find a good quality allocation of tasks. The allocation quality, as shown in Fig. 1, remains close to the quality of the optimal allocation. The obtained allocation quality was within 80% of the optimal even with 50 sub-tasks. The number of messages exchanged grows linearly with the number of iterations. This has been verified through multiple simulations. Thus, the modeled system and the proposed algorithms deliver as expected with promising results. However, the proposed system failed to allocate some tasks in some simulations. This occurred in cases in which none of the agents which received the task information had the capability (or intention 1) towards the unallocated sub-tasks. Such unallocated sub-tasks must be sent to new and undiscovered agents for allocation. The development of algorithms for forwarding of unallocated tasks to undiscovered agents will be taken up in the future.

ACKNOWLEDGMENT

This research is supported by the Department of Electronics and Information Technology, Ministry of Communications and IT, Government of India under the Visvesvaraya PhD Scheme administered by Media Lab Asia.

REFERENCES

[1] Q. Hu, "High-efficiency mission planning and distribution system of the land observation satellites," *Global Journal of Researches In Engineering*, vol. 14, no. 7, 2015.

[2] Y. Chen, D. Zhang, M. Zhou, and H. Zou, "Multi-satellite observation scheduling algorithm based on hybrid genetic particle swarm optimization," in *Advances in Information Technology and Industry Applications*. Springer, 2012, pp. 441–448.

[3] G. Wu, M. Ma, J. Zhu, and D. Qiu, "Multi-satellite observation integrated scheduling method oriented to emergency tasks and common tasks," *Systems Engineering and Electronics, Journal of*, vol. 23, no. 5, pp. 723–733, 2012.

[4] G. Wu, J. Liu, M. Ma, and D. Qiu, "A two-phase scheduling method with the consideration of task clustering for earth observing satellites," *Computers & Operations Research*, vol. 40, no. 7, pp. 1884–1894, 2013.

[5] J. Wu, L. Liu, and X. Hu, "A mutation-based algorithm for deadline-constrained task scheduling in micro-satellite clusters," in *Natural Computation (ICNC), 2015 11th International Conference on*. IEEE, 2015, pp. 419–424.

[6] Q. Dishan, H. Chuan, L. Jin, and M. Manhao, "A dynamic scheduling method of earth-observing satellites by employing rolling horizon strategy," *The Scientific World Journal*, vol. 2013, 2013.

[7] J. Wang, X. Zhu, L. T. Yang, J. Zhu, and M. Ma, "Towards dynamic real-time scheduling for multiple earth observation satellites," *Journal of Computer and System Sciences*, vol. 81, no. 1, pp. 110–124, 2015.

[8] J. Pfrommer, "Distributed constrained optimization over constrained communication topologies," in *Proceedings of the 2014 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*, vol. 20. KIT Scientific Publishing, 2015, p. 77.

[9] M. Vinyals, J. A. Rodriguez-Aguilar, and J. Cerquides, "Constructing a unifying theory of dynamic programming dcop algorithms via the generalized distributive law," *Autonomous Agents and Multi-Agent Systems*, vol. 22, no. 3, pp. 439–464, 2011.

[10] A. Farinelli, A. Rogers, and N. R. Jennings, "Agent-based decentralised coordination for sensor networks using the max-sum algorithm," *Autonomous agents and multi-agent systems*, vol. 28, no. 3, pp. 337–380, 2014.

[11] R. Zivan, H. Yedidsion, S. Okamoto, R. Glinton, and K. Sycara, "Distributed constraint optimization for teams of mobile sensing agents," *Autonomous Agents and Multi-Agent Systems*, vol. 29, no. 3, pp. 495–536, 2015.

[12] P. Skobelev, E. Simonova, A. Zhilyaev, and V. Travin, "Multi-agent planning of spacecraft group for earth remote sensing," in *Service Orientation in Holonic and Multi-Agent Manufacturing*. Springer, 2016, pp. 309–317.

[13] B. Ottens, C. Dimitrakakis, and B. Faltings, "Duct: An upper confidence bound approach to distributed constraint optimization problems," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 1, no. EPFL-CONF-197504, 2012, pp. 528–534.

[14] R. Zivan, R. Glinton, and K. Sycara, "Distributed constraint optimization for large teams of mobile sensing agents," in *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on*, vol. 2. IET, 2009, pp. 347–354.

[15] "Space-track.org." [Online]. Available: https://www.space-track.org