The Role of Central Nodes in Multi-Task Allocation for Distributed Space-Based Observation Systems

Vincenzo Messina
Technical University of Munich
Ottobrunn, Germany
vincenzo.messina@tum.de

Marco Camporeale
Technical University of Munich
Ottobrunn, Germany
m.camporeale@tum.de

Rob Vingerhoeds
Fédération ONERA - ISAE-SUPAERO - ENAC, Université de Toulouse
Toulouse, France
rob.vingerhoeds@isae-supaero.fr

Alessandro Golkar
Technical University of Munich
Ottobrunn, Germany
golkar@tum.de

ABSTRACT

This work presents a reward-based selective propagation algorithm that enables distributed agents to evaluate, exchange, and bid task-specific reward functions while minimizing communication overhead. Distributed observation satellites in a Walker-Delta constellation exchange multiple tasks of observing target objects in Sun-Synchronous Orbits. We analyze the impact of the number of satellites, the number of target objects, and the fraction of central nodes on the task's diffusion, task performance quality, and bidding status over time. We investigate constellations with up to 5,000 satellites and 500 concurrent target objects, with a fraction of the central nodes ranging from 1% to 100% of the total number of satellites. Results show that the proposed framework maintains high performance and fast convergence even under heavy task loads, with convergence times below 100 s for networks with as few as 5% central nodes. Increasing the number of satellites reduces the required central node fractions to achieve faster convergence, suggesting that a minimum absolute number of central nodes is required to achieve maximum network responsiveness for the given constellation geometry. We identified a power-law relationship among network size, central node fraction, and link efficiency, offering valuable insights into architectural trade-offs for future Distributed Satellite Systems.

KEYWORDS

Distributed Satellite Systems, Task Allocation, Decentralized Networks, Large-Scale Simulation, Object Detection

ACM Reference Format:

Vincenzo Messina, Marco Camporeale, Rob Vingerhoeds, and Alessandro Golkar. 2026. The Role of Central Nodes in Multi-Task Allocation for Distributed Space-Based Observation Systems. In *Appears at the International Workshop on Autonomous Agents and Multi-Agent Systems for Space Applications (MASSpace-26). Held as part of the Workshops at the 25th International Conference on Autonomous Agents and Multiagent Systems., Paphos, Cyprus, May 2026*, IFAAMAS, 9 pages.

Appears at the International Workshop on Autonomous Agents and Multi-Agent Systems for Space Applications (MASSpace-26). Held as part of the Workshops at the 25th International Conference on Autonomous Agents and Multiagent Systems., S. Chien, G. Picard, I. Zilberstein (Chairs), May 2026, Paphos, Cyprus. © 2026 Copyright held by the owner/author(s).

1 INTRODUCTION AND BACKGROUND

Distributed Satellite Systems (DSS) are increasingly adopted for earth observation missions due to their inherent advantages in scalability, resilience, and spatial coverage. As mission objectives expand to include the simultaneous observation of large numbers of dynamic target objects, efficiently allocating tasks across DSS has emerged as a key systems engineering challenge [2].

This paper investigates the impact of central nodes on the performance of distributed multi-task allocation in large-scale DSS. We extend the decentralized task allocation framework, presented in [8], to a multi-task reward-based selective propagation framework in which satellites exchange compact task-level reward metrics, represented by the Reward Function (RF), instead of full state information, enabling distributed bidding under intermittent communication and without prior knowledge of other agents' capabilities. We refer to central nodes as satellites that act as hubs for task coordination and data propagation; task propagation occurs only when at least one of the communicating satellites is a central node. We consider a Walker-Delta Constellation with up to 5,000 observer satellites that need to observe up to 500 objects in space, considered as target objects that represent debris, other satellites, or natural bodies, distributed along Sun-synchronous orbits (SSO). The results focus on network performance, represented by the evolution of the maximum reward, the bidding status over time, and the diffusion of task knowledge. We analyze different levels of centralization by varying the fraction of central nodes relative to the total number of satellites. We observe a power law among network size, the fraction of central nodes, and the link efficiency, defined as the ratio of the total number of links n_L before convergence to the number of tasks n_T. The term DSS refers to satellite systems comprising two or more spacecraft that communicate to accomplish the mission goal [7]. They dynamically form networks of heterogeneous satellites designed to reduce revisit times, cover large areas at higher resolution, or minimize data access latency. This is essential for real-time Earth Observation missions, for monitoring deforestation, studying sea-ice coverage, tracking agricultural evolution and weather impacts, as well as for humanitarian aid, governments, and NGOs to monitor oceans and conflict zones [1]. Centralized architectures are

vulnerable to single points of failure and are fundamentally incompatible with federated systems; they are not beneficial for resource allocation in planning and scheduling when the number of satellites increases [2]. Different examples of DSS are implemented nowadays, such as satellite constellations for communication, such as Starlink and OneWeb, or for Earth Observation, such as PlanetLabs [1]. Many studies focus on the importance of satellite collaborations. Dynamic target (DT) is an example that highlights the importance of DSS, in which look-ahead sensor data is acquired, rapidly analyzed, and used to drive subsequent observations by the same spacecraft or another spacecraft in a trail. Use cases for such a capability include: cloud avoidance [4], storm hunting, search for planetary boundary layer events [14], plume study, and beyond [3]. Federated Satellite Systems (FSS) extend this paradigm further, enabling satellites from different operators to autonomously share resources through standardized protocols [11]. This improves mission efficiency through resource pooling and enhanced resilience [9], while reducing barriers to entry for smaller operators.

Schaefer et al. [12] demonstrate that inter-satellite collaboration reduces revisit time and increases sample count for wildfire-observing missions, highlighting the utility of coordinated DSS operations. Existing methods have already been developed and investigated in the field of DSS task scheduling.

Parjan et al.[10] propose a broadcast-based coordination methodology (BD) in which satellites announce their task selections to the network without requiring detailed knowledge of other agents' capabilities. In this approach, each satellite independently evaluates its suitability for available tasks using local heuristics, then broadcasts status information, such as task-satisfaction confirmations or priority indicators. These methods achieve near-centralized performance but face communication bandwidth challenges as the network size grows and assume that all satellites know the tasks at the problem's initialization.

Zilberstein et al. [15] extend the BD method to formulate satellite scheduling problems as distributed constraint optimization problems (DCOPs). DCOP frameworks use Geometric Neighborhood Decomposition heuristics to subdivide the global problem into manageable subproblems. This approach reduces messaging overhead by limiting coordination to satellites within smaller neighborhoods, whose boundaries are determined by orbital mechanics and communication capabilities. After the subdivision, Neighborhood Stochastic Search (NSS), a method built on the earlier BD, solves these subproblems. DCOP demonstrates improved scalability compared to fully centralized approaches and BD, particularly in dense task environments where decomposition naturally aligns with spatial task clustering. They assume agents possess knowledge of other agents' capabilities and that all agents have immediate knowledge of all tasks. Many DCOP solvers require significant computational resources and extensive inter-agent communication, potentially exceeding the constraints of resource-limited satellites.

Li et al. [6] propose an extension of the Consensus-Based Bundle Algorithm (m-CBBA), where individual satellites determine the full task schedule locally and achieve mutual consensus through iterative communication. An extension of this algorithm is the Asynchronous m-CBBA (m-ACBBA), in which satellites predict communication window opportunities and account for their possible asynchrony. These methods are generally flexible and can guarantee highly-optimal solutions; however, the computational and communication overheads do not scale efficiently with the number of satellites and tasks. Table 1 summarizes the discussed frameworks, highlighting the strengths and limitations identified in each. While decentralized satellite task allocation methods offer clear benefits, they also entail inherent trade-offs, particularly in scalability and communication overhead as the number of satellites and tasks grows.

This study investigates how the fraction of central nodes f_{CN} affects convergence time, task knowledge diffusion, and communication efficiency in a large-scale distributed satellite system. While the analysis is conducted within the proposed selective propagation

Table 1: Comparison of Decentralized Satellite Task Allocation Methods

Method	*BD [10]*	*NSS/DCOP [15]*	*m-ACCBA [6]*	*RF Selective Propagation (Proposed Method)*
Approach	Broadcast satisfaction and contention information	Decomposition and stochastic search	Auction-based iterative bidding	Reward-based selective propagation
Communication	Broadcast with a large amount of information	Scoped to local neighborhood	Exchanges across the network scale quadratically with the number of tasks	Broadcast with reduced information: reward functions, satellite identifiers, and timestamps
Computation	Light, simple heuristics only	Moderate, local search per agent	Heavy, evaluates all task permutations	Per task, light: reward prediction only
Scalability	Medium to high, bandwidth-limited	Higher than BD through standardized decomposition	Limited by bandwidth and task count, mitigated by m-ACCBA	Very high, demonstrated with thousands of satellites
Robustness	Medium, dependent on broadcast reliability	High, independent subproblems	High, supports asynchronous communication	High, opportunistic communication and dynamic topologies
Task and DSS knowledge at initialization	Tasks: global; satellite state: local heuristics only	Tasks: global; satellite state: neighbors only	Tasks: global; satellite state: unknown	Tasks: known to one satellite; satellite state: unknown

framework, the findings offer broader architectural insights: the results suggest that introducing even a small fraction of relay-capable nodes can significantly reduce convergence time and communication overhead, pointing to partial centralization as a promising direction for improving the responsiveness of distributed task allocation methods in general.

The main contributions are: (1) an extension of the single-task selective propagation framework [8] to a multi-task setting with multiple concurrent observation targets; (2) a parametric study of f_{CN} on convergence time, knowledge diffusion, and communication efficiency across constellations of up to 5,000 satellites and 500 concurrent tasks; and (3) the identification of a power-law relationship between η_L and n_{CN}/n_T, with coefficients scaling systematically with constellation size. Section 2 presents the methodology, Section 3 the experimental setup and results, and Section 4 the conclusions.

2 METHODOLOGY

This section describes the methodology used to develop the framework to investigate the effect of central nodes on multi-task allocation across multiple satellite architectures.

2.1 Problem Definition

We address the allocation of multiple tasks T, defined by observations of target objects O in SSO, among a satellite set S with different numbers of central nodes CN. Satellites operate in a Walker-Delta constellation, carry imaging payloads, and evaluate task execution capability and quality over the mission interval $\{t_{in}, ..., t_{end}\}$. The simulation assigns satellites in S to each task in T to maximize observation quality for satellites with good resource availability. At each timestep, the simulation evaluates feasible inter-satellite communications between satellite-central node pairs $(S, CN) \in S$. Initially, one satellite knows the full task set T. This assumption models two representative operational scenarios: tasks may be uplinked from a ground station to a single contact satellite, or originated autonomously by a satellite in orbit upon detection of target objects. Relaxing this assumption to account for multiple simultaneous task injection points, or for dynamic task generation during mission execution, is left as a direction for future work.

In both cases, knowledge must propagate through the network. Upon receiving a task, a satellite computes its RF value and broadcasts the higher of the received and evaluated values.

We define the task allocation problem as follows:

(1) $\{t_{in}, ..., t_{end}\}$ is the overall mission duration for the performance of the observations, starting at time t_{in} and ending at time t_{end}. t is discretized by a time step size $\Delta t > 0$ to result in the finite number of discrete time steps $t = (t_{end} - t_{in})/\Delta t$. The set of time steps, defined as $t = \{1, 2, ..., t_{end}\}$, contains all time steps at which a task can be performed.

(2) $S = \{1, 2, ..., n_s\}$ is a set of satellites, in a Walker-Delta Constellation architecture, equipped with an observational payload. The initial parameters and variables for the communication, power, and data-handling subsystems are identical across all satellites. From now on, any variable with the subscript i is associated with the i-th satellite.

(3) $f_{CN} = \{0.01, ..., 1\}$ is the fraction of the number of central nodes with respect to the total number of satellites in the constellation. They are responsible for the task propagation and are equipped with an observation payload.

(4) $O = \{1, 2, ..., n_O\}$ is the set of target objects to be observed, equally distributed along various SSO. Each object defines one observation task, so $n_T = n_O$.

(5) T_i, for each satellite $i \in S$, is the *Global Task Register*, storing for each task k the best known reward value $T_i^{k,RF}$, the associated satellite identifier $T_i^{k,ID}$, and the timestamp $T_i^{k,t}$ at which the task can be performed.

(6) L_i^k is the *Local Task Register* of satellite i, storing its locally evaluated reward $L_i^{k,RF}$ and identifier $L_i^{k,ID}$ for task k.

2.2 Architectural Variables and Parameters

The architectural design space was structured around one primary design variable, the fraction of central nodes, as defined in Eq. (1).

$$f_{CN} \triangleq \frac{n_{CN}}{n_s} \tag{1}$$

with $f_{CN} \in \{0.01, ..., 1.0\}$. A set of secondary variables supports the exploration of different architectures, including the number of satellites and target objects, as shown in Table 2.

Table 2: Architectural Design Variables

Tier	*Variable*	*Values*
Primary	Fraction of Central Nodes	0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0
Secondary	Number of Satellites	100, 500, 1000, 2000, 3000, 5000
Secondary	Number of Target Objects	1, 5, 10, 50, 100, 500

2.3 Figure of Merit

We consider five Figures of Merit (FoM):

- **Time to Converge** t_c represents the time to converge to a single satellite aware of being the best for each task.
- **Task Intensity** T_{int} defined as the ratio n_T/n_s and measures the load level of the network in terms of tasks to be performed.
- **Maximum Reward Function** RF_{Max} represents the highest performance achieved in performing a certain task.
- **Link Efficiency** η_L defined as the ratio between the total number of links n_L that occurred before convergence and the number of tasks n_T.
- **Diffusion Time** t_D represents the time elapsed until all satellites hold the complete task set T, i.e., $T_i \neq \emptyset$ for all $i \in S$. It marks the end of the knowledge propagation phase, which precedes and bounds the bidding convergence phase ($t_D \leq t_c$).

2.4 Task Packets

One assumption made at this stage of the research, to simplify simulation development, is that tasks are not executed individually in the framework but rather collectively as a *task packet*. The task-specific information exchanged, aside from communication

protocol overhead, consists of the Two-Line Element (TLE) of the target object, the satellite identifier associated with the highest reward function value, and the corresponding best-known reward value. Table 3 reports the assumed values for each component.

Table 3: Data Size Components

Variable	*Value [Bytes]*
TLE target object (s_{TLE})	104
ID + RF (s_{ID} + s_{RF})	8
Transport Layer Protocol (s_{TLP})	34
Network Layer Protocol (s_{NLP})	24
Datalink Layer Protocol (s_{DLP})	35
Encoding Scheme Reed–Solomon (s_{ESRS})	13
TOTAL (D_s)	$112 \cdot n_T + 106$

As the number of tasks increases, the number of communication windows that are too short to send the entire set of tasks increases. To enable data exchange across a larger number of tasks, the communication protocol is assumed to follow principles similar to those of the Licklider Transmission Protocol (LTP) for deep-space communications. Satellite pairs initiate links and transmit data in minimum chunks corresponding to the amount transferable within a single timestep of duration $\Delta t = 1$ s. If a visibility window terminates before completion, satellites retain the already transmitted data and resume transmission during the next available contact.

2.5 Selective Propagation Algorithm

The selective propagation algorithm manages the information exchange and task allocation through pairs of satellites evaluated at each simulation timestep. A satellite pair (S_i, S_j) initiates communication only if at least one member is a central node, including pairs where both members are central nodes, neither satellite is processing previously received information, and the effective data rate of the inter-satellite link is strictly positive. Communication occurs only if one of the two satellites in the pair knows the tasks, or if there is a knowledge-mismatch condition, defined as either unequal task sets or disagreement on the best-known reward function value for at least one shared task. Data transfer accumulates across consecutive visibility windows until the full task set size is reached, at which point the final exchange is reached; partial transfers persist across interrupted links and resume upon reestablishment of contact.

Bidding Process. Upon receiving a task packet, the receiving satellite j enters a distributed single-item bidding process independently for each task k. The bid of satellite j for task k is its locally computed reward function value $RF_j^k(t)$, evaluated using Eq. (2) over the remaining mission interval $\{t, ..., t_{end}\}$. The satellite retains the task assignment; i.e., records itself as the best candidate, only if its locally computed bid strictly exceeds the best-known value received from the sender. Otherwise, the received entry is propagated unchanged. This mechanism enforces a greedy, communication-efficient consensus: at any point in time, each satellite stores the highest known bid for each task along with the identifier of the satellite that produced it.

During bidding, each satellite maintains a task buffer that temporarily stores newly received or locally updated task information within a timestep, isolating it from the global task register. The bidding logic separates incoming tasks into previously known and newly discovered subsets. For known tasks, larger reward values and their associated satellite identifiers overwrite buffered entries only when strictly greater values are received; ties are resolved by retaining the current buffered entry. For newly discovered tasks, the receiving satellite evaluates its local reward function over $\{t, ..., t_{end}\}$, then updates the buffered entry only if the locally computed value strictly exceeds the received one; ties are resolved in favor of the received entry, preserving the propagation direction. At the end of each timestep, buffered task information is promoted to the global task register for all satellites.

This mechanism enforces control over information diffusion, avoids redundant transmissions, limits communication to strictly necessary exchanges, and enables distributed convergence to a unique, best-performing satellite per task with the largest RF. Algorithm 1 summarizes the described selective propagation algorithm.

Reward Function: Evaluation of Observation Capability. Each satellite evaluates its capability for performing a given observation task k using a reward function RF, defined as in Eq. (2).

$$RF_i^k(t) \;=\; q_i\, a_i(t) \left[A\, p_i(t) \;+\; B\, ds_i(t) \;+\; C \left(1 - \left| \frac{\varphi_i^k(t)}{FoV_i} \right| \right) \right] \qquad (2)$$

For each satellite i, task k, and timestep t, the variables are defined as follows. $q_i \in \{0, 1\}$ is a binary indicator taking value 1 if the imaging payload is available and 0 otherwise; $a_i(t) \in \{0, 1\}$ takes value 1 if the satellite is not already committed to a concurrent task at timestep t and 0 otherwise. Both q_i and a_i act as multiplicative gates: if either is zero, the reward is zero regardless of the remaining terms. The remaining variables are continuous and defined over $[0, 1]$: $p_i(t) = E_{a,i,t}/E_{\max}$ is the fraction of stored energy available, where $E_{a,i,t}$ is the energy stored onboard satellite i at timestep t and $E_{\max}$ is the maximum storage capacity; $ds_i(t) = DS_{a,i,t}/DS_{\max}$ is the fraction of available onboard data storage, where $DS_{a,i,t}$ is the data storage available on satellite i at timestep t and $DS_{\max}$ is the total onboard storage capacity; and $(1 - |\varphi_i(t)/FoV_i|)$ is the normalized angular proximity of the target to the payload boresight, where $\varphi_i(t)$ is the angular separation between the pointing direction and the target position and FoV_i is the field of view of the imaging payload. The parameters A, B, and C are calibration weights that emphasize different aspects of the reward; in this study they are all set to 1. The maximum ideal value is $RF_{\text{ideal}} = 3.0$, attained when the payload is perfectly aligned with the target and the satellite has full energy, storage, and schedule availability. It shall be noted that this reward function definition has not significantly changed from previous work [8].

Convergence and Stopping Condition. The algorithm terminates when a global convergence condition is reached: for every task $k \in T$, all satellites in the network agree on the same satellite identifier as the best candidate, i.e., $T_{k,ID}^i = T_{k,ID}^j$ for all $i, j \in S$. This condition is checked at the end of each timestep after buffer promotion. When convergence is detected, the convergence time is

Algorithm 1 Selective Propagation Algorithm

Require: Satellites $S = \{1, \ldots, n_S\}$, relay-capable set $CN \subseteq S$, targets $O = \{1, \ldots, n_O\}$, connected to task set T, interval $[t_{in}, t_{end}]$, step Δt
Require: Coefficients A, B, C (default $A = B = C = 1$)
Ensure: Convergence time t_c; task register T_i for all $i \in S$
Note: $RF_i^k(t)$ is computed using Eq. (2) over $[t,\ t_{end}]$.
Init: Choose one satellite with T; all others start empty.
for $t \leftarrow t_{in}$ **to** t_{end} **step** Δt **do**
  Update states (positions, $p_i(t)$, $ds_i(t)$, $a_i(t)$ for all $i \in S$)
  for all pairs (i, j) with $i < j$ **do** ▷ i = sender, j = receiver
    if $i \in CN \ \vee \ j \in CN$ **then**
      if rate$(i, j) > 0$ **then**
        if knowledge mismatch between T_i and T_j, **or** $T_j = \emptyset$ **then**
          Satellite i sends task packet to satellite j;
            accumulate data until size D_s is reached
            (resume across interrupted links)
          if transfer complete **then**
            Satellite j enters processing mode (comm disabled for 1 s)
            for all received tasks k **do**
              if $k \in T_j$ **then** ▷ Task already known to j
                if $T_{k,RF}^{recv} > T_{k,RF}^{buf,j}$ **then**
                  $T_{k,RF}^{buf,j} \leftarrow T_{k,RF}^{recv}$
                  $T_{k,ID}^{buf,j} \leftarrow T_{k,ID}^{recv}$
                  $T_{k,t}^{buf,j} \leftarrow T_{k,t}^{recv}$
                end if ▷ Ties: keep current entry (no update)
              else ▷ Task newly discovered by j
                Compute $RF_j^k(t)$ using Eq. (2)
                $L_{k,RF}^{j} \leftarrow RF_j^k(t)$
                $L_{k,ID}^{j} \leftarrow j$
                if $L_{k,RF}^{j} > T_{k,RF}^{recv}$ **then**
                  $T_{k,RF}^{buf,j} \leftarrow L_{k,RF}^{j}$
                  $T_{k,ID}^{buf,j} \leftarrow j$
                  $T_{k,t}^{buf,j} \leftarrow t$
                else
                  $T_{k,RF}^{buf,j} \leftarrow T_{k,RF}^{recv}$
                  $T_{k,ID}^{buf,j} \leftarrow T_{k,ID}^{recv}$
                  $T_{k,t}^{buf,j} \leftarrow T_{k,t}^{recv}$
                end if ▷ Ties: received entry takes precedence
              end if
            end for
            Satellite j exits processing mode
          end if
        end if
      end if
    end if
  end for
  for all $i \in S$ **do**
    Promote buffer to T_i; clear buffer
  end for
  if converged (unique max $T_{k,RF}$ per k network-wide) **then**
    $t_c \leftarrow t - t_{in}$
    return t_c, $\{T_i\}_{i \in S}$
  end if
end for

recorded as $t_c = t - t_{in}$ and the simulation terminates. It should be noted that convergence of reward values alone is not sufficient: two satellites may store the same *RF* value for a task while associating it with different satellite identifiers, which would constitute a tie rather than a resolved allocation. The bidding process does not include a mechanism to discern between two identical RF values: an incoming reward value overwrites the buffered entry only if it is *strictly* greater. Consequently, an incoming RF equal to the currently stored value is discarded and the existing entry is retained unchanged, implying that the first satellite to propagate a given RF value for a task retains the assignment. This preserves the direction of propagation and ensures a unique identifier is stored network-wide upon convergence. Algorithm 1 summarizes the full procedure.

3 RESULTS

3.1 Experimental Setup

The simulation environment is configured as follows. Both satellites and targets are assumed to be in Low Earth Orbit. Table 4 summarizes the orbital parameters.

Table 4: Orbital Parameters for Satellites and Targets

Variable	*Satellites*	*Targets*
Type	Walker Delta	SSO
n_p [-]	10	n_T
a [km]	6923	7171
i [°]	53	98.55 ± 10
e [-]	0	0.0009
Ω [°]	$[0, 360]$	267
ω [°]	0	84
θ [°]	$[0, 360]$	$[0, 360]$

The constellation is composed of $n_p = 10$ orbital planes, among which the satellites are equally distributed. Each orbit is perfectly circular, with eccentricity $e = 0$, and has a semi-major axis $a = 6923$ km, corresponding to a constant orbital altitude of 550 km. The inclination i and the argument of periapsis ω are set to 53° and 0°, respectively, for all orbits. The initial true anomaly θ and the right ascension of the ascending node Ω take variable values. Given the total number of satellites n_S, satellite i is assigned Ω_i and θ_i according to Eq. (3) and Eq. (4).

$$\Omega_i = \frac{2\pi}{n_p} \left\lfloor \frac{i}{n_S/n_p} \right\rfloor, \tag{3}$$

$$\theta_i = \frac{2\pi}{n_S/n_p} \left(i \bmod \frac{n_S}{n_p} \right). \tag{4}$$

The outputs of (3) and (4) are converted from radians to degrees. The orbital parameters of the satellites correspond to the first group of Starlink satellites [13]. Each target defines a distinct orbital plane, with the reference plane corresponding to that of Envisat, characterized by an inclination of 98.55°. Target inclinations are equally spaced within a range of ±10° around this reference value. Initial target true anomalies are uniformly distributed over 360°. These represent orbits commonly used for Earth Observation satellites and those with the highest density of space debris [5]. The simulator precomputes all orbital states and retrieves them during execution. Each satellite carries identical subsystems. The electrical power subsystem includes a 0.4×0.3 m solar panel with 22% efficiency, packing factor 0.9, maximum stored energy $E_{\max} = 84$ kJ, operational power demand 28.5 W, and eclipse power demand 5.0 W, with variation across satellites depending on the orbit. The optical payload has a 5° field of view, aperture diameter $d_l = 9$ cm, and wavelength limits $\lambda = 400$–700 nm; assuming a target size $d_T = 10$ m, the Rayleigh criterion gives a maximum observable distance $d_{\max} = d_T d_l / (1.22\,\lambda) \approx 860$ km. The onboard computer

requires 1 s to evaluate reward functions, during which communication is disabled. Inter-satellite communications operate at 437 MHz with 9.6 kHz bandwidth, 2 W transmit power, and receiver sensitivity −151 dBW. Satellites establish communication only when the effective data rate, computed from the link budget, Shannon capacity, and bit error rate, is strictly positive; otherwise the link is inactive. All parameter values are selected to align with typical specifications for a 6U CubeSat. All satellites maintain a fixed attitude aligned with the positive z-axis of the Earth-Centered Inertial frame, with payload boresight constant over time. At each timestep of $\Delta t = 1$ s, the simulator updates the status of each satellite.

3.2 Comparison with Existing Methods

The results are interpreted in the context of Table 1. The proposed framework differs from the baseline methods along three dimensions directly observable in the results below.

Regarding *task knowledge at initialization*: BD [10], NSS/DCOP [15], and m-CBBA [6] assume global task knowledge from the outset. The proposed method relaxes this assumption entirely; only one satellite holds the full task set at initialization. The knowledge diffusion results in Section 3.3 quantify this cost: diffusion time t_D is the primary bottleneck, not the bidding process itself.

Regarding *scalability*: The proposed method evaluates only a single reward value per task per satellite, demonstrating convergence at up to 5,000 satellites and 500 concurrent tasks, a scale not demonstrated by any baseline method.

Regarding *centralization*: none of the baseline methods model or vary the degree of centralization as an architectural parameter. This study quantifies how f_{CN} affects convergence time, diffusion, and communication efficiency; a design dimension absent from the existing literature. The results show that $f_{CN} \geq 0.05$ is sufficient for fast convergence, suggesting partial centralization as a scalable architectural strategy applicable beyond the specific algorithm studied here.

3.3 Individual Run Analysis

We introduce the results of a selected individual run, for reference and to set a standard for the general performance of the framework. We select the constellation with 5,000 satellites, of which 10% (500) are Central Nodes, and with a set of 50 target objects.

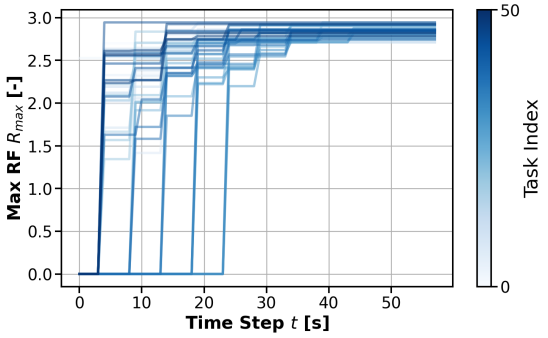


Figure 1: Reward function evolution over time for $n_S = 5000$, $n_T = 50$, $f_{CN} = 0.1$.

Figure 1 shows the maximum RF value per task, which reaches 75% of the theoretical maximum of 3.0 after 5 s; no further improvements are found after 45 s despite the network not yet having converged. Figure 2 shows knowledge diffusion, which completes at $t_D = 48$ s, only 10 s before full network convergence, indicating that task knowledge diffusion, not the bidding process, is the primary bottleneck under the current assumptions. Figure 3 shows the bidding status, which rises to a peak of over 40 simultaneous

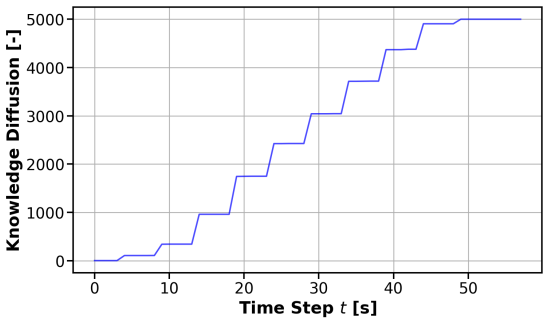


Figure 2: Knowledge diffusion over time for $n_S = 5000$, $n_T = 50$, $f_{CN} = 0.1$.

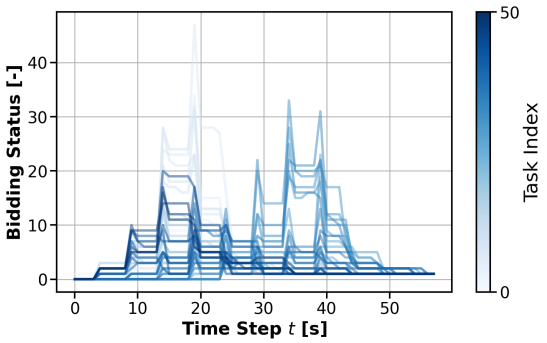


Figure 3: Bidding status over time for $n_S = 5000$, $n_T = 50$, $f_{CN} = 0.1$.

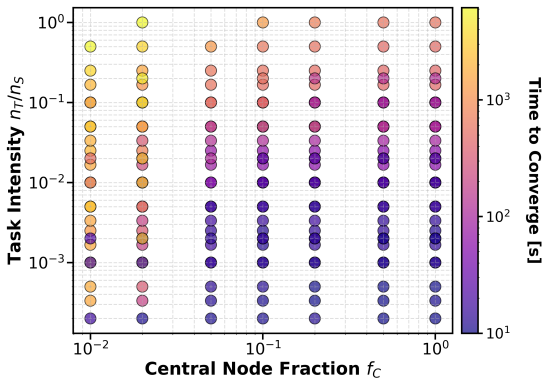


Figure 4: Performance Landscape of the studied collaborative framework, on log-log scale.

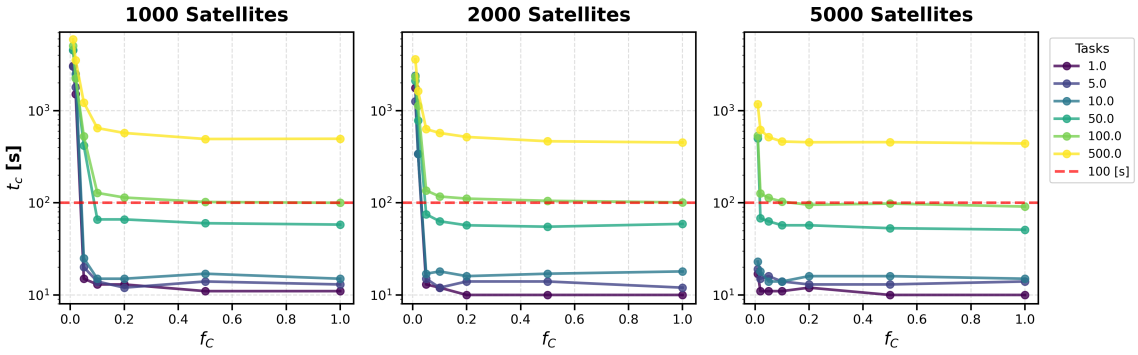


Figure 5: Time to converge t_c plotted against f_{CN} for $n_S \in \{1000, 2000, 5000\}$.

candidates per task during the initial propagation phase, then decreases nonlinearly until a single satellite per task is aware of being the best candidate.

3.4 Network Performance

The performance landscape plot in Figure 4 maps system behavior in a reduced two-parameter space. The horizontal axis represents the fraction of central nodes f_{CN}. The vertical axis represents task intensity, defined as the task load ratio n_T/n_S. Each data point encodes the convergence time t_c through color, with yellow indicating larger t_c values and purple indicating smaller t_c values. The surface shows a continuous dependence of network responsiveness on both f_{CN} and n_T/n_S. Increasing task intensity increases convergence time. Increasing f_{CN} decreases convergence time. The plot enables identification of the minimum f_{CN} to meet a target responsiveness level for a given task load. Configurations with $f_{CN} \geq 0.05$ maintain $t_c < 100$ s across the explored range of n_T/n_S.

Figure 5 reports convergence time t_c as a function of the central node fraction f_{CN} for multiple task counts n_T and three constellation sizes n_S. Each subplot corresponds to a fixed value of n_S.

The curves exhibit saturation behavior for $f_{CN} > 0.1$, where further increases in f_{CN} yield marginal reductions in t_c. Larger constellations show steeper reductions of t_c with increasing f_{CN}, indicating lower required central node fractions to achieve minimum convergence times as n_S increases. Increasing n_T reduces the variability of t_c with respect to f_{CN} for fixed n_S. The observed saturation suggests the existence of a minimum absolute number of central nodes required to achieve maximum network responsiveness for the given constellation geometry. A detailed analysis of the communication topology is outside the scope of this study.

Another performance metric considers the ratio of central nodes to tasks, n_{CN}/n_T. Figure 6 reports convergence time t_c (left panel) and link efficiency η_L (right panel) as functions of n_{CN}/n_T, with all axes in logarithmic scale.

The left panel shows that configurations with $n_{CN} < n_T$ do not achieve $t_c < 100$ s. This threshold appears invariant across the tested values of n_S, indicating a scale-independent constraint within the explored parameter range. The observed threshold can be understood as a large-scale network effect: as the number of tasks n_T increases, the task packet size D_s grows proportionally, requiring longer or more numerous communication windows to complete each transfer. With fewer central nodes than tasks ($n_{CN} < n_T$), the relay capacity of the network is insufficient to absorb this increased communication demand within the available contact opportunities, and the cumulative delay across the diffusion graph pushes convergence beyond 100 s. This effect is scale-independent: it persists across all tested values of n_S, suggesting that the ratio n_{CN}/n_T captures a fundamental constraint on the relay capacity of the network relative to its task load. Validation outside this range requires simulations with larger DSS.

Each horizontal stripe of points corresponds to a fixed number of tasks n_T, which compares with Figure 5. Movement along a stripe toward lower n_{CN}/n_T corresponds to decreasing n_{CN}. Increasing n_S reduces the sensitivity of t_c to variations in n_{CN} for fixed n_T. The color gradients converge toward linear trends as n_S increases, indicating reduced variability of convergence time with respect to central node count in larger constellations.

Communication Efficiency. The right panel of Figure 6 reports the link efficiency $\eta_L = n_L/n_T$, the ratio of total links before convergence to number of tasks. Lower η_L indicates higher information yield per communication event. As shown in Figure 6, η_L grows log-linearly with n_{CN}/n_T, with larger constellations exhibiting steeper slopes due to greater communication demand. This motivates the power-law fit:

$$\eta_L = A \times \left(\frac{n_{CN}}{n_T}\right)^b \tag{5}$$

Parameters A and b were identified for each constellation size n_S and plotted in Figure 7 as functions of n_S. The coefficient b follows a linear dependence on n_S (Eq. (6)), while A follows a power-law dependence on n_S (Eq. (7)). Linear regression yields p-values of 5.187×10^{-3} for b and 1.659×10^{-3} for A.

$$b = 4.15 \times 10^{-5}\, n_S + 0.617 \tag{6}$$

$$A \approx 1.1 \times n_S^{0.81} \tag{7}$$

Table 5 reports the total number of links n_L established before convergence under the current communication assumptions. The n_L increases for larger decentralized constellation. However, we see a maximum n_L at $n_T = 10$ and $f_{CN} = 1.0$, rather than at a boundary of the $n_T \times f_{CN}$ parameter space. This pattern holds across all tested

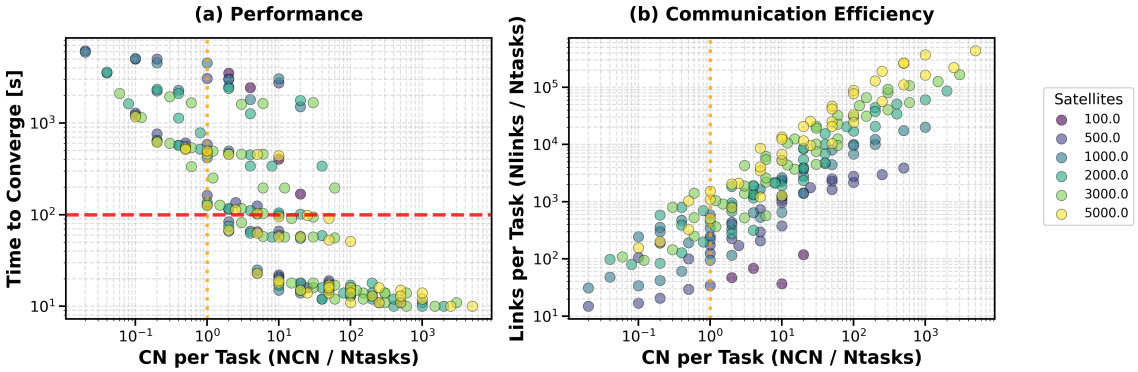
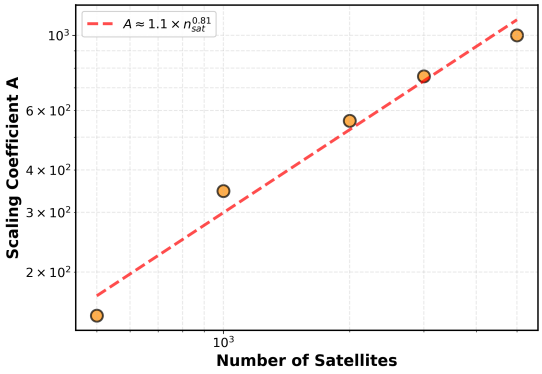
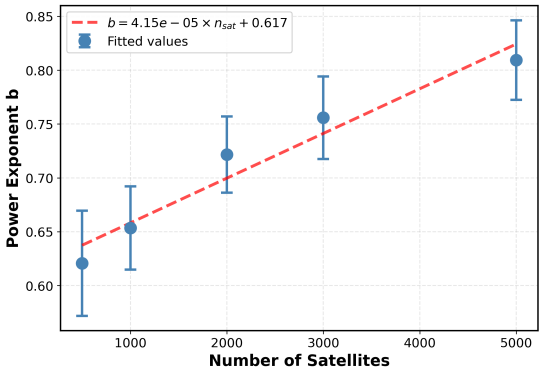


Figure 6: Time to converge t_c (on the left) and link efficiency η_L (on the right) against CN to task ratio. Axes are all in log-scale.



(a) Scaling coefficient A.



(b) Power exponent b.

Figure 7: Power-law coefficients A (top) and b (bottom) for the relation between η_L and n_{CN}/n_T as functions of n_S.

values of n_S and needs to be further investigated. Tables for other constellation sizes are omitted for brevity. For $n_T \leq 10$, the task packet size permits complete transmission within a single 1 s link, the minimum link duration imposed by the simulation timestep $\Delta t = 1$ s. Increasing n_T increases n_L, as satellites require additional communication rounds to reach agreement on multiple tasks. For $n_T > 10$, task packet sizes exceed the single-step transmission capacity. Satellites complete data exchange only after full task set transfer. Convergence time increases, link durations increase, and the number of completed links decreases. Parallel bidding across the network advances consensus during partial transmissions; some unfinished links are no longer needed yet still occur. This effect reduces n_L despite higher task counts.

Table 5: Number of Links for n_T and f_{CN} (5000 Satellites). Values are $\times 10^5$

f_{CN}	**0.01**	**0.02**	**0.05**	**0.10**	**0.20**	**0.50**	**1.00**
n_T							
1	0.25	0.33	0.56	1.1	1.6	2.2	4.4
5	0.62	1.0	2.1	3.9	6.5	1.3	1.8
10	0.86	1.4	2.6	4.7	8.9	19	26
50	0.76	1.0	2.0	3.4	5.8	10	12
100	1.0	1.1	2.1	3.5	5.2	9.6	10
500	0.79	1.0	1.6	2.5	4.0	6.0	6.4

The results show that $f_{CN} \geq 0.05$ maintains $t_c < 100$ s across all task intensities, and that n_{CN}/n_T is a scale-independent constraint: constellations with $n_{CN} < n_T$ consistently fail to converge within 100 s regardless of n_S. Larger constellations require smaller f_{CN} fractions for equivalent responsiveness, and communication efficiency follows a power-law in n_{CN}/n_T with coefficients scaling systematically with n_S.

4 FUTURE WORK AND CONCLUSION

Future work includes investigations into different constellation architectures, additional performance metrics such as system complexity and robustness, and determination of the optimal degree of centralization for specific use cases such as planetary boundary layer monitoring and wildfire detection. We will study resilience under random and targeted central-node failures. Extensions to dynamic task generation and multiple simultaneous task injection points will relax the single-origin assumption adopted here. We also plan to incorporate realistic inter-satellite network protocols to improve simulation fidelity. Overall, the results demonstrate that reward-based selective propagation enables fast, scalable, and communication-efficient multi-task allocation, highlighting the importance of partial centralization as a practical architectural strategy for future distributed satellite systems.

REFERENCES

[1] C. Araguz, E. Bou-Balust, and E. Alarcón. 2018. Applying autonomy to distributed satellite systems: Trends, challenges, and future prospects. *Systems Engineering* 21 (2018), 401–416. https://doi.org/10.1002/sys.21428

[2] M. K. Ben-Larbi, K. Flores Pozo, T. Haylok, M. Choi, B. Grzesik, A. Haas, D. Krupke, H. Konstanski, V. Schaus, S. P. Fekete, C. Schurig, and E. Stoll. 2021. Towards the automated operations of large distributed satellite systems. Part 1: Review and paradigm shifts. *Advances in Space Research* 67, 11 (2021), 3598–3619. https://doi.org/10.1016/j.asr.2020.08.009

[3] Alberto Candela, Jason Swope, and Steve A. Chien. 2023. Dynamic Targeting to Improve Earth Science Missions. *Journal of Aerospace Information Systems* 20, 11 (2023), 679–689. https://doi.org/10.2514/1.I011233

[4] Steve Chien, Itai Zilberstein, Alberto Candela, David Rijlaarsdam, Tom Hendrix, Aubrey Dunne, Aragon Oriol, and Miquel Juan Puig. 2025. Flight of Dynamic Targeting on the CogniSAT-6 Spacecraft. arXiv preprint. https://arxiv.org/abs/2509.05304

[5] Jian Huang, Xiangxu Lei, Bin Li, Jizhang Sang, and Hongkang Liu. 2022. Towards Fast and Reliable Evaluation of Detection Performance of Space Surveillance Sensors. *Remote Sensing* 14 (01 2022), 483. https://doi.org/10.3390/rs14030483

[6] Guoliang Li. 2020. Online scheduling of distributed Earth observation satellite system under rigid communication constraints. *Advances in Space Research* 65, 11 (2020), 2475–2496. https://doi.org/10.1016/j.asr.2020.02.018

[7] L. M. Marrero, J. C. Merlano-Duncan, J. Querol, S. Kumar, J. Krivochiza, S. K. Sharma, S. Chatzinotas, A. Camps, and B. Ottersten. 2022. Architectures and Synchronization Techniques for Distributed Satellite Systems: A Survey. *IEEE Access* 10 (2022), 45375–45409. https://doi.org/10.1109/ACCESS.2022.3169499

[8] Vincenzo Messina and Alessandro Golkar. 2025. Advancing Federated Satellite Systems Performance: A Collaborative Method for Improved Object Detection in Space. In *AIAA SCITECH 2025 Forum*. American Institute of Aeronautics and Astronautics, Orlando, Florida, USA. https://doi.org/10.2514/6.2025-0588 AIAA Paper 2025-0588.

[9] S. Oh and D. Vasisht. 2024. A Call for Decentralized Satellite Networks. In *Proceedings of the 23rd ACM Workshop on Hot Topics in Networks*. Association for Computing Machinery, New York, NY, USA, 25–33. https://doi.org/10.1145/3696348.3696896

[10] S. Parjan and S. A. Chien. 2023. Decentralized Observation Allocation for a Large-Scale Constellation. *Journal of Aerospace Information Systems* 20, 8 (2023), 447–461. https://doi.org/10.2514/1.I011215

[11] J. A. Ruiz-De-azua, N. Garzaniti, A. Golkar, A. Calveras, and A. Camps. 2021. Towards federated satellite systems and internet of satellites: The federation deployment control protocol. *Remote Sensing* 13 (2021), 1–22. https://doi.org/10.3390/rs13050982

[12] Ryan S. Schaefer and Paul T. Grogan. 2024. Collaborative Constellation Analysis Framework for Wildfire Observing Missions. In *Proceedings of the IEEE Aerospace Conference*. IEEE, Big Sky, Montana, USA, 1–11. https://doi.org/10.1109/AERO58975.2024.10521362

[13] Calum Spring-Turner and Raj Thilak Rajan. 2022. Performance Bounds for Cooperative Localisation in the Starlink Network. arXiv:2207.04691 [eess.SY] https://arxiv.org/abs/2207.04691

[14] Josue I. Tapia and Paul T. Grogan. 2023. Dynamic Targeting for Precipitation Observing Missions: Integrating the GEOS-5 Nature Run Data Set. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2023)*. IEEE, Pasadena, California, USA, 3764–3767. https://doi.org/10.1109/IGARSS52108.2023.10282353

[15] Itai Zilberstein, Ananya Rao, Matthew Salis, and Steve Chien. 2025. Decentralized, Decomposition-Based Observation Scheduling for a Large-Scale Satellite Constellation. *Journal of Artificial Intelligence Research* 82 (January 2025), 169–208. https://doi.org/10.1613/jair.1.16997