ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

UDC 004.896:629.735.05

O Babilunga, PhD, Assoc. Prof.,
D. Dmytrenko,
O. Andriianov, PhD, Assoc. Prof.
Odessa Polytechnic National University, Shevchenko Ave. 1, Odesa, Ukraine, 65044, e-mail: 4068167@stud.op.edu.ua

A METHOD FOR RATIONAL ROLE ALLOCATION IN A COMPUTER-VISION-ENABLED SWARM OF UNMANNED AERIAL VEHICLES UNDER RESOURCE CONSTRAINTS

О. Бабілунга, Д. Дмитренко, О. Андріянов. **Метод раціонального розподілу ролей у рої безпілотних літальних апаратів із комп'ютерним зором з урахуванням обмежених ресурсів.** Розподіл ролей у рої БПЛА з бортовим комп'ютерним зором ускладнюється тим, що «важкі» алгоритми сприйняття споживають енергію, займають обчислювальний ресурс і підвищують вимоги до радіоканалу. Для малорозмірних платформ це напряму пов'язано з обмеженнями розміру, маси та потужності (SWaP) і, як наслідок, зі скороченням часу польоту. Більшість практичних евристик вибирає виконавців ролей за геометрією або поточним зарядом і тому не враховує взаємозв'язок між енергетикою, обчисленнями та зв'язністю. У роботі запропоновано децентралізовану процедуру призначення ролей (Scout, Mapper, Relay, Worker), у якій кожен апарат формує ставку з урахуванням прогнозованих витрат на пропульсію, обчислення, сенсори та передачу даних, а також ризику зростання багатострибкової відстані до базової станції. Призначення виконується аукціонною процедурою з локальним узгодженням переможців між сусідніми агентами. Запроваджено правила допустимості, що відсікають призначення ролей при недостатньому запасі батареї або дефіциті обчислювального бюджету, і штраф за перемикання ролей для зменшення осциляцій. Метод перевірено у відтворюваній симуляції на ПК для роїв із 20 і 50 апаратів у межах області 1000×1000 м за обмеженого радіуса зв'язку та гетерогенних обчислювальних можливостей; оцінювалися SR, Time, CR, Conn і Sw. У складнішому сценарії (N=20) запропонований підхід забезпечує завершення місії (SR=1.00), тоді як геометрична евристика та статичний розподіл не досягають порогу (SR=0). Порівняно з SoC-евристикою метод зменшує Sw приблизно у 7–8 разів за збереження 100% успішності та прийнятної зв'язності з базою. Запропонована модель вартості може бути використана в системах керування роєм і надалі розширена на сценарії колективного сприйняття, коли частина апаратів виконує обчислювально важку обробку зору.
Ключові слова: ройова робототехніка, розподіл ролей, безпілотні літальні апарати, комп'ютерний зір, енергоменеджмент, обчислювальні обмеження, зв'язність мережі, децентралізована координація, моделювання, симуляційний експеримент

O. Babilunga, D. Dmytrenko, O. Andriianov. **A Method for Rational Role Allocation in a Computer-Vision-Enabled Swarm of Unmanned Aerial Vehicles Under Resource Constraints.** Role allocation in unmanned aerial vehicle swarms equipped with onboard vision is constrained not only by geometry but also by size, weight and power limits typical for small platforms. Vision pipelines increase energy draw, occupy the onboard compute budget and raise the demand on the wireless link. Heuristics that assign roles solely from distance or state of charge often overlook this coupling and may trigger unnecessary re-assignments. We propose a decentralized auction-based role allocation scheme (Scout, Mapper, Relay, Worker) where each vehicle computes a bid from (*i*) predicted energy cost for propulsion, sensing, computation and radio, and (*ii*) the risk of increasing the multi-hop distance to a base station. Feasibility checks prevent assigning vision-heavy roles to agents with insufficient battery reserve or compute capacity, and a switching penalty reduces oscillations. The approach is evaluated in a reproducible PC simulation (1000×1000 m area, limited communication range and heterogeneous computing capabilities) with swarms of 20 and 50 UAVs. We report mission success rate, completion time, mission progress, average base connectivity and the number of role switches. In the more challenging N=20 case the proposed method completes the mission (SR=1.00) while the distance-based and static baselines fail (SR=0). Compared with the battery-only heuristic, it preserves the same success rate but reduces role switching by about 7–8 while maintaining acceptable connectivity. For N=50 all methods succeed, yet the proposed approach keeps the number of switches at 310±52 versus 2386±320 for the battery-only baseline. The cost model can be integrated into swarm management systems and extended to collaborative perception, where a subset of vehicles performs vision-heavy processing and shares compact results under progressive battery depletion.
Keywords: swarm robotics, role allocation, unmanned aerial vehicles, computer vision, energy management, computational constraints, network connectivity, decentralized coordination, simulation, mission planning

1. Introduction

Unmanned aerial vehicles (UAVs) equipped with onboard computing modules (single-board computers, embedded GPUs, etc.) are increasingly used for computer-vision tasks: object detection and tracking, mapping, VIO/SLAM-based localization, and elements of collaborative perception. Integrating such algorithms increases energy consumption and the load on the onboard computer, and also raises the requirements for the wireless link. For small platforms this directly manifests as size, weight and power (SWaP) constraints: additional mass and power draw reduce endurance and decrease the available control authority margin.

In swarm systems these constraints interact with formation control, network connectivity mainte-

DOI: 10.15276/opu.2.72.2025.11

nance, and the distribution of functions among agents. Surveys in swarm robotics and formation control emphasize that scaling a swarm is not possible without decentralized coordination and information-exchange protocols [1, 2]. Role allocation enables specialization of subsets of vehicles: some perform scouting and data collection, others relay traffic and maintain the network, while others execute compute-intensive vision tasks. Common approaches include contract-net mechanisms, auction-based methods, and consensus algorithms such as CBBA [3, 4].

This paper proposes a decentralized role allocation approach for a UAV swarm with onboard vision that simultaneously accounts for predicted energy costs, available computing resources, and the risk of degrading multi-hop connectivity to a base station. In practical terms, an agent's bid is not reduced to "closer/farther" or "higher/lower SoC"; instead, it includes the expected costs of propulsion, data processing, sensing, and communication, as well as a penalty for frequent role switching. The novelty of the approach lies in combining these components within a single auction procedure together with feasibility rules that prevent assigning resource-intensive roles to energy- or compute-limited platforms. The workflow of the method is shown in Fig. 1.

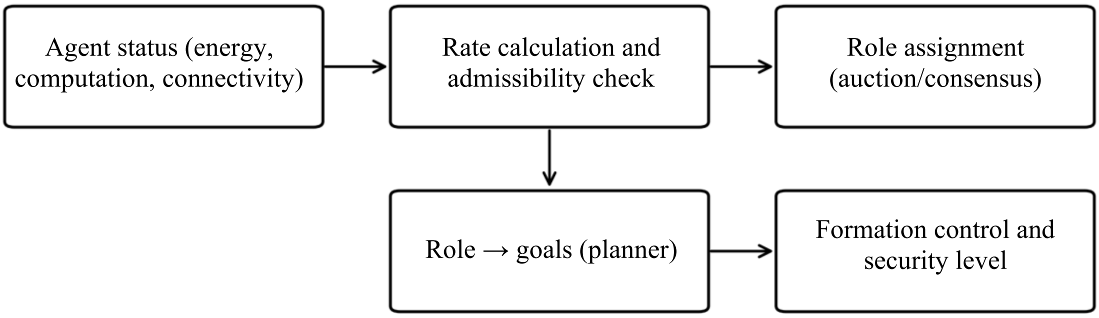

Fig. 1. Workflow of role allocation considering energy, computation, and connectivity

2. Literature review and problem statement

Designing UAVs with onboard computers requires reconciling size-weight and energy constraints with the demands of vision algorithms. Energy consumption and power-supply selection for UAVs are actively discussed in the literature, particularly in the context of mission endurance and flight profiles [5]. At the same time, modern camera-based navigation and mapping systems (VIO/SLAM), such as VINS-Mono and ORB-SLAM2, demonstrate high accuracy but require stable computing performance and, in collective settings, efficient mechanisms for data exchange [6, 7]. A related direction is energy-efficient sensing and computing (e.g., event cameras), which can potentially reduce power demand [8].

For UAV swarms, key challenges include formation control, cooperative trajectory planning, and maintaining network connectivity during maneuvering. Survey works [1, 2] summarize distributed control and coordination approaches and highlight the role of communication topology. Role and task allocation in multi-robot systems is often implemented via market-based mechanisms and auctions, which ensure scalability but require a well-defined cost function [3, 4]. However, in many publications the role/task cost is dominated by geometric or temporal factors (distance, time-to-arrival), while the energy price of onboard computation and the risk of losing connectivity to the base station are accounted for only indirectly or ignored. This gap defines the focus of the present study.

Classical works on Multi-Robot Task Allocation (MRTA) provide formal taxonomies and criteria for comparing task-allocation algorithms, which is useful for a correct problem formulation in UAV swarms [9, 10]. For scenarios with dynamic communication constraints, extensions of CBBA have been proposed that consider time windows, resource costs, and changing network topology [11], as well as approaches that reduce inter-agent communication to accelerate convergence [12] and methods that explicitly account for limited bandwidth and collisions in a shared channel [13]. A review of task allocation methods specifically for UAV systems is given in [14]. Despite substantial progress, most works still lack an integrated treatment of three groups of constraints: (*i*) energy (SoC and cost prediction), (*ii*) onboard computing (load from computer-vision pipelines), and (*iii*) connectivity requirements for delivering data to a base station. The proposed approach focuses on this integrated setting.

Problem statement. Let the swarm consist of N agents. For the i-th agent, the state $s_i(t)$ includes, in particular, the remaining battery charge $\mathrm{SoC}_i(t)$, the effective performance of the onboard computer

C_i, and local information about neighbors in the network. The goal is to assign roles $r_i(t) \in$ {Scout, Mapper, Relay, Worker} so as to maximize mission progress under the following constraints:

1) $SoC_i(t)$ does not drop below SoC_{min} given predicted expenditures;
2) roles with vision load do not exceed the available compute budget;
3) multi-hop connectivity to the base station is maintained to deliver data.

The problem is complicated by the fact that energy cost and compute demand depend on the role, while communication topology changes as agents move. For quantitative evaluation, we use PC-based simulation; examples of UAV simulation environments include AirSim and RotorS [15, 16].

3. Aim and objectives

The aim of this study is to develop a role allocation method for a UAV swarm with onboard vision computing that enables rational use of energy, computing, and communication resources.

To achieve this aim, the following tasks are addressed:

– define role profiles in terms of computing demand, traffic, and typical energy (SoC) consumption;
– build a resource-aware role cost model (energy + computation + connectivity) and feasibility rules;
– develop an auction-based role assignment procedure with a switching penalty;
– define a simulation evaluation protocol and compare the proposed method with baseline heuristics.

4. Materials and methods

The proposed method treats role assignment as a local decision made by each agent based on its own state and messages from neighbors. Every Δt seconds, an agent checks role feasibility with respect to battery reserve and compute budget, computes bids, and exchanges them with neighbors within communication range. After local agreement on winners (given role quotas), assignments are activated for the next control interval, while a switching penalty reduces oscillations in role composition.

4.1. Roles and resource profiles. Four roles are defined in the swarm: Scout, Mapper, Relay, and Worker. Each role is described by a profile of compute requirements, traffic demand, and typical relative state-of-charge consumption. The profiles used in this paper are listed in Table 1 (all values are given in relative units).

Table 1

Role profiles and resource requirements

Role	Main tasks	Compute demand (rel.)	Traffic (rel.)	SoC drain, 1/s
Scout	object search/detection, video data collection	1.5	1.0	0.0019
Mapper	SLAM/mapping, local map alignment	1.2	1.2	0.0016
Relay	packet relaying, connectivity maintenance	0.8	1.5	0.0012
Worker	tracking/coverage, auxiliary tasks	0.5	0.3	0.0010

4.2. Connectivity and local network model. The communication structure is modeled as an undirected graph $G(t) = (V, E(t))$, where vertices V correspond to agents and an edge $(i, j) \in E(t)$ exists if the distance between agents does not exceed the communication range R. The base station B is connected to agent i if $d(i, B) \leq R$. An agent is considered connected to the base if there exists a path from i to B in the graph (multi-hop). The fraction of agents connected to the base is defined as $C_B(t) = N_B(t)/N$, where $N\_B(t)$ is the number of connected agents at time t. To assess network risk, we use the hop distance $h_i(t)$, i.e., the length of the shortest path from agent i to the base in $G(t)$ (if no path exists, we set $h_i(t) = \infty$). We deliberately use hop distance as a simple local metric: it can be obtained from neighbor messages without computationally expensive global network estimates.

4.3. Energy model and cost prediction. The agent's energy expenditure is decomposed into components associated with flight (propulsion), computation, communication, and sensing. At the level of instantaneous power:

$$P_i(t) = P_{i,\text{prop}}(t) + P_{i,\text{comp}}(r_i,t) + P_{i,\text{comm}}(r_i,t) + P_{i,\text{sens}}(r_i,t) . \quad (1)$$

To check feasibility and compute bids, we estimate the energy "price" of a role over a horizon H. In the simplest case, we use a prediction of the average power for role r and obtain:

$$E_{i,\text{pred}}(r,t) = P_i(r,t)\cdot H \; . \tag{2}$$

4.4. Computing constraints and load. Because some roles (Scout, Mapper) execute vision algorithms, we introduce the effective compute capability C_i of agent i and a relative role load $C(r)$. The normalized compute utilization is defined as:

$$u_i(r,t) = C(r)\,/\,C_i \, . \tag{3}$$

4.5. Role feasibility rules. Before participating in the auction, an agent discards roles that cannot be executed given the minimum battery reserve and the available compute resources. For role r, the feasibility conditions are:

$$\text{SoC}_i(t) - E_{i,\text{pred}}(r,t)\,/\,E_{i,\text{bat}} \geq \text{SoC}_{\min}, \; u_i(r,t) \leq 1 \, . \tag{4}$$

4.6. Bid function and switching penalty. For each feasible role, the agent forms a bid $b_i(r,t)$, which is maximized in the auction procedure (5). In (5), $U_i(r,t)$ is the role utility, $\bar{E}_i(r,t)$ is the normalized estimate of energy expenditure over horizon H, $\acute{L}_i(r,t)$ is the normalized estimate of the risk of degrading connectivity to the base station, and $J_{i,\text{sw}}$ is the penalty for changing roles (to reduce assignment oscillations). Normalization is introduced so that no term dominates solely due to its scale: $\bar{E}_i(r,t)=E_{i,\text{pred}}(r,t)/(\text{SoC}_i(t)-\text{SoC}_{\min}+\varepsilon)$, $\acute{L}_i(r,t)=\min(1,\ h_i\text{\^{}pred}(r,t)/h_{\max})$, where h_i^pred is the predicted shortest-path length (in hops) from agent i to the base after one motion step in role r, h_max is a threshold (in our experiments $h_{\max}=6$), and ε=10^−3. For brevity, the arguments (r,t) in the right-hand side of (5) are omitted.

$$b_i(r,t) = U_i - \alpha\cdot\bar{E}_i - \beta\cdot u_i - \gamma\cdot\acute{L}_i - J_{i,sw} \, . \tag{5}$$

4.7. Auction-based role assignment. At each replanning step t, the agent computes bids for all feasible roles and sends to its neighbors messages of the form ⟨role, bid, agent identifier, timestamp⟩. Agreement is implemented as an iterative max-consensus procedure: for each role and its quota K_r, agents exchange the best known candidates over L_c communication rounds, after which a consistent winner list is formed. Conflicts are resolved by priority (higher bid; if equal, lower identifier). In our experiments we use L_c=5 (Table 2), which provides stable agreement within the local communication range.

Table 2

Simulation parameters

Parameter	Symbol	Value
Mission area size	L	1000×1000 m
Speed (simplified)	v	5 m/s
Energy prediction horizon	H	20 s
Consensus iterations	L_c	5
Agent compute resources (rel.)	C_i	40%: 1.5; 60%: 1.0
Simulation time	T	600 s
Replanning step	Δt	5 s
Initial state of charge	$\text{SoC}_i(0)$	1.0
Mission progress threshold	M_{th}	1000 a.u.
Communication range	R	180 m
Minimum battery reserve	$\text{SoC}_{\min}$	0.10
Number of agents	N	20 and 50
Role quotas (N=20)	K	Scout=2, Mapper=2, Relay=3
Role quotas (N=50)	K	Scout=5, Mapper=5, Relay=7

The penalty $J_{i,\text{sw}}$ reduces frequent re-assignments and the associated overhead (communication/replanning) and decreases the risk of control instability. The overall role-consensus procedure is given as pseudocode in Algorithm 1 – Decentralized SWaP-aware role allocation (CBBA-like):

Input: roles R with quotas K_r; weights w=(α,β,γ,η); prediction horizon H; threshold $\text{SoC}_{\min}$; communication graph $G(t)$.

Output: role assignment $r_i(t)$ for each agent i.

ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

For each replanning epoch t:

1. (Evaluation) each agent i forms the set of feasible roles $F_i(t)$ and bids $b_i(r,t)$ according to (5).
2. (Initial bid) agent i ranks roles in descending order of b_i and chooses the current proposal $r_i \leftarrow$ argmax_r $b_i(r,t)$.
3. Repeat until convergence or maxiter:
   3.1. (Exchange) send neighbors a message $m\_i=(i, r_i, b_i(r_i), t)$.
   3.2. (Local ranking) based on received $\{m\_j\}$, build for each role r a candidate list and locally select winners W_r – the K_r best bids (ties are resolved by ID).
   3.3. (Max-consensus) agree on Wr with neighbors; if Wr changes, continue iterations.
   3.4. (Try alternatives) if agent i is not in W_{ri}, remove ri from the local list and choose the next best role; if $F_i(t)$ is empty, set $ri \leftarrow$ Worker.
4. After convergence, each agent adopts role $ri(t)$ according to the agreed winner sets Wr.

Simulation protocol and baselines. The evaluation is performed in an agent-based PC simulation over a time interval T=600 s with replanning every Δt=5 s. The motion and network models are described in Sections 4.2…4.4, while the vision workload is parameterized (without running real inference) through the compute and bandwidth requirements in Table 1. For a fair comparison, all methods use the same feasibility rules (4) with respect to battery reserve and compute budget; the methods differ in the bidding/selection criterion and in the re-assignment mechanism. The mission completion threshold is set to M_{th}=1000 arbitrary units. The experiment parameters are given in Table 2.

Mission progress model. Let $n_S(t)$ and $n_M(t)$ denote the number of agents with roles Scout and Mapper, respectively, that have a path to the base station (i.e., $h_i(t) \le h_{\max}$). The accumulated mission progress $M(t)$ (arbitrary units) is updated discretely with step Δt according to:

$$M(t+\Delta t) = M(t) + \Delta t \cdot (w_S \cdot n_S(t) + w_M \cdot n_M(t)) . \quad (6)$$

Initially, $M(0)$=0, and the mission is considered completed if $M(t) \ge M_{th}$.

The proposed approach is compared with three baseline methods: (1) “Distance” – role assignment using a geometric heuristic with quotas (Scout – farthest from the base; Mapper – closest; Relay – intermediate distances); (2) “Battery level” – role assignment based solely on maximum SoC, without explicit connectivity consideration and without a switching penalty; (3) “Static roles” – a fixed role distribution at the start, without replanning. The proposed method is denoted as “SWaP-aware”.

5. Results and discussion

5.1. Evaluation metrics. We use the following metrics: SR – share of successful runs (reaching the threshold $M_$th within time T); Time – mean completion time (s); CR – relative mission progress at the end of the simulation ($M(T)/M_$th); Conn – mean fraction of agents connected to the base ($c\_B(t)$); Sw – total number of role switches in the swarm. Additionally, to characterize resource use we report: ΔSoC_avg – mean battery charge consumed per agent per run; SoC_min,end – minimum SoC among agents at completion; Viol – number of violations of feasibility rules (4); VisionHigh – share of Scout/Mapper assignments to agents with sufficient compute resources. Summary results are given in Tables 3 and 4, and an ablation study is reported in Table 5.

Table 3

Comparison of methods by main metrics (mean ± std over 12 runs)

Method	N	SR	Time, s	CR	Conn	Sw
SWaP-aware	20	1.00	443.8±22.4	1.00	0.92±0.03	115±18
Distance	20	0.00	600.0	0.84	0.78±0.05	344±41
Battery level	20	1.00	395.0±19.1	1.00	1.00±0.00	953±175
Static roles	20	0.00	600.0	0.77	0.82±0.04	0
SWaP-aware	50	1.00	182.1±14.7	1.00	0.87±0.04	310±52
Distance	50	1.00	276.7±18.9	1.00	0.79±0.05	365±49
Battery level	50	1.00	160.0±12.4	1.00	1.00±0.00	2386±320
Static roles	50	1.00	275.8±16.4	1.00	0.83±0.04	0

5.2. Quantitative comparison. Table 3 shows that for the N=20 swarm the static distribution and the “Distance” heuristic do not reach the threshold within time T (SR=0.00), whereas SWaP-aware and “Battery level” achieve SR=1.00. At the same time, SWaP-aware substantially reduces assignment oscillations: Sw=115±18 versus 953±175 for the SoC-only heuristic (about 8.3× fewer switches), while maintaining high mean connectivity (Conn=0.92±0.03). Table 4 shows that all methods satisfy the feasibility rules (Viol=0) and assign Scout/Mapper only to agents with sufficient compute resources (VisionHigh=100%). From an energy perspective, the SoC-only heuristic predictably maintains a higher minimum charge (SoC_min,end≈0.28), but at the cost of a much larger number of re-assignments.

Table 4

Energy and computing metrics (mean ± std over 12 runs)

Method	*N*	ΔSoC_{avg}	$SoC_{min,end}$	Viol	VisionHigh, %
SWaP-aware	20	0.52±0.04	0.16±0.03	0	100
Distance	20	0.71±0.05	0.10±0.00	0	100
Battery level	20	0.47±0.06	0.28±0.04	0	100
Static roles	20	0.70±0.04	0.10±0.00	0	100
SWaP-aware	50	0.22±0.02	0.65±0.04	0	100
Distance	50	0.33±0.03	0.48±0.06	0	100
Battery level	50	0.19±0.03	0.72±0.03	0	100
Static roles	50	0.33±0.03	0.47±0.05	0	100

Table 5

Ablation study of SWaP-aware (*N*=20; mean ± std over 12 runs)

Variant	SR	Time, s	CR	Conn	Sw
Full (α,β,γ,η)	1.00	443.8±22.4	1.00	0.92±0.03	115±18
No network term (γ=0)	0.75	548.0±34.6	0.93±0.05	0.80±0.06	138±25
No switching penalty (η=0)	1.00	412.5±20.1	1.00	0.91±0.04	740±120

For *N*=50 all methods reach the threshold (SR=1.00), yet differences appear in assignment stability and network quality. The SoC-only heuristic achieves Conn≈1.00, but at the cost of a very large number of switches (Sw=2386±320). SWaP-aware reduces Sw to 310±52 (about 7.7× fewer switches) while preserving high connectivity (0.87±0.04). The ablation study (Table 5) shows that removing the network term (γ=0) degrades connectivity and success rate, while removing the switching penalty (η=0) causes a sharp increase in Sw. To illustrate the mission progress dynamics for *N*=20, Fig. 2 shows a representative single run.

Figure 2 shows a typical example (one run) of the mission progress dynamics for *N*=20. In this run, the adaptive methods “SWaP-aware” and “Battery level” reach the completion threshold M_{th}, while the “Distance” and “Static roles” baselines exhibit long plateaus due to episodes of degraded network connectivity. Aggregate (mean ± std over 12 runs) values of CR and Time are reported in Table 3. Base-station connectivity for the same run is shown in Fig. 3.

Figure 3 illustrates the influence of role assignment on maintaining connectivity to the base station. In the shown run, “SWaP-aware” provides more stable connectivity compared to the “Distance” baseline, for which deeper and longer intervals of reduced connectivity are observed. This degrades data delivery to the base and slows down mission progress in Fig. 2. The number of role switches (Sw) is reported in Table 3.

5.3. Discussion and limitations. The results indicate that for swarms with onboard computers it is important to jointly consider energy, computing load, and connectivity. The ablation study (Table 5) shows that the network term $\gamma\acute{L}\_i$ is critical for sparse-topology scenarios (*N*=20), while the switching penalty η limits oscillations without a noticeable loss in SR. Nevertheless, the study has several limitations: (1) the vision profiles are parameterized and do not include real neural-network inference; (2) the communication and motion models are simplified and do not account for obstacles, varying throughput, or delays; (3) the evaluation is simulation-only, without hardware validation. Future work includes integration with AirSim/RotorS and experiments with real onboard computers and cameras. It is also worth noting that in the current configuration computing constraints do not become a bottleneck (Table 4: Viol=0; VisionHigh=100%), therefore their impact should be further evaluated in scenarios with stronger compute heterogeneity and possible throttling.

ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

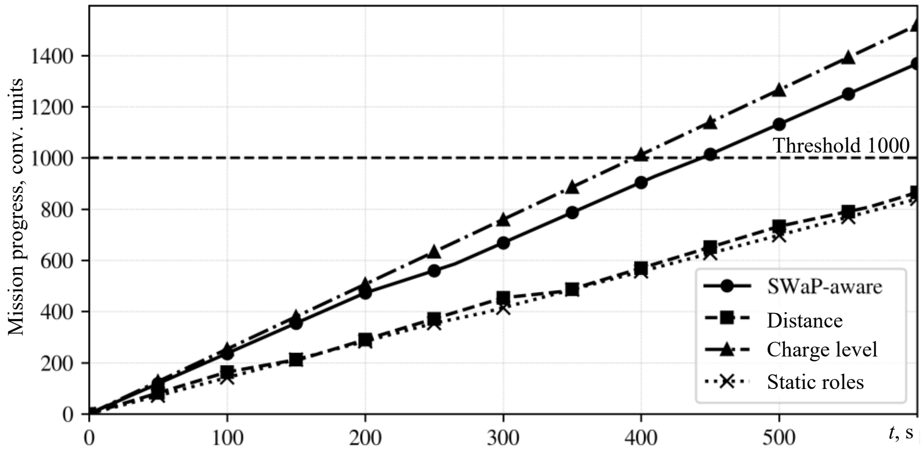

Fig. 2. Mission progress over time for *N*=20 (single run; threshold *d* = 1000)

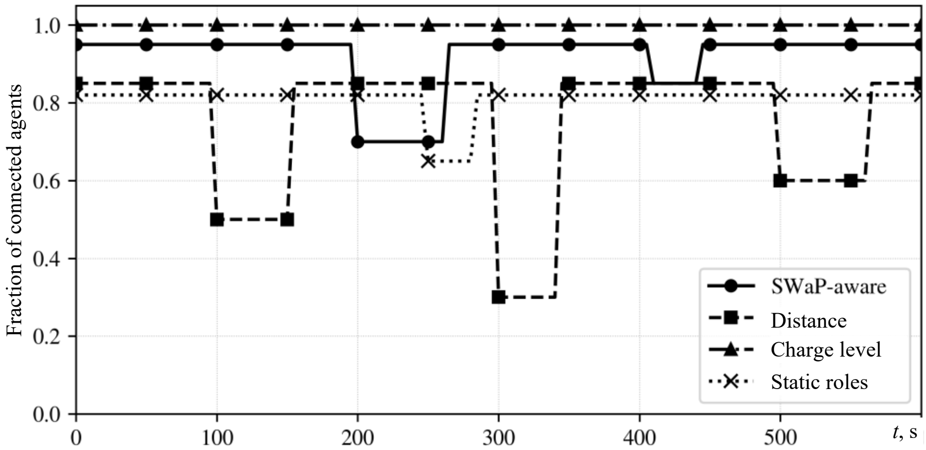

Fig. 3. Base-station connectivity over time for *N*=20 (fraction of connected agents; same run as in Fig. 2)

6. Conclusions

In summary, this paper proposes a decentralized role allocation method for a computer-vision-enabled UAV swarm that combines energy, computing, and network factors within a single assignment procedure. The main results can be summarized as follows:

– role profiles and a resource cost model covering propulsion, computation, sensing, and data transmission are formulated;

– feasibility rules and a bid function with normalized penalties for energy, network risk, and role switching are proposed;

– in simulation, the method achieves SR=1.00 for *N*=20 and *N*=50; in the more challenging *N*=20 scenario it outperforms the geometric heuristic and the static distribution (SR=0), and compared with the SoC-only heuristic it substantially reduces role switching (115±18 vs. 953±175 for *N*=20 and 310±52 vs. 2386±320 for N=50) while maintaining acceptable connectivity;

– future steps are outlined: more realistic channel and delay models, integration with AirSim/RotorS, and hardware experiments on onboard platforms.

Література

1. A Survey on Aerial Swarm Robotics / S.-J. Chung, A. A. Paranjape, P. Dames et al. *IEEE Transactions on Robotics*. 2018. Vol. 34, no. 4. P. 837–855. DOI: 10.1109/TRO.2018.2857475.
2. Oh K.-K., Park M.-C., Ahn H.-S. A Survey of Multi-Agent Formation Control. *Automatica*. 2015. Vol. 53. P. 424–440. DOI: 10.1016/j.automatica.2014.10.022.

3. Choi H.-L., Brunet L., How J. P. Consensus-Based Decentralized Auctions for Robust Task Allocation. *IEEE Transactions on Robotics*. 2009. Vol. 25, no. 4. P. 912–926. DOI: 10.1109/TRO.2009.2022423.
4. Market-Based Multirobot Coordination: A Survey and Analysis / M. B. Dias, R. Zlot, N. Kalra, A. Stentz. *Proceedings of the IEEE*. 2006. Vol. 94, no. 7. P. 1257–1270. DOI: 10.1109/JPROC.2006.876939.
5. Boukoberine M. N., Zhou Z., Benbouzid M. E. H. A Critical Review on Unmanned Aerial Vehicle Power Supply and Energy Management: Solutions, Strategies, and Prospects. *Applied Energy*. 2019. Vol. 255. 113823. DOI: 10.1016/j.apenergy.2019.113823.
6. Qin T., Li P., Shen S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics*. 2018. Vol. 34, no. 4. P. 1004–1020. DOI: 10.1109/TRO.2018.2853729.
7. Mur-Artal R., Tardós J. D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*. 2017. Vol. 33, no. 5. P. 1255–1262. DOI: 10.1109/TRO.2017.2705103.
8. Fully neuromorphic vision and control for autonomous drone flight / F. Paredes-Vallés, J. J. Hagenaars, J. Dupeyroux et al. *Science Robotics*. 2024. Vol. 9. eadi0591. DOI: 10.1126/scirobotics.adi0591.
9. Gerkey B. P., Matarić M. J. A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *The International Journal of Robotics Research*. 2004. Vol. 23, no. 9. P. 939–954. DOI: 10.1177/0278364904045564.
10. Korsah G. A., Stentz A., Dias M. B. A Comprehensive Taxonomy for Multi-Robot Task Allocation. *The International Journal of Robotics Research*. 2013. Vol. 32, no. 12. P. 1495–1512. DOI: 10.1177/0278364913496484.
11. Decentralized Planning for Complex Missions with Dynamic Communication Constraints / S. Ponda, J. Redding, H.-L. Choi et al. *Proc. of the 2010 American Control Conference (ACC)*. 2010. P. 3998–4003. DOI: 10.1109/ACC.2010.5531232.
12. Kim K.-S., Kim H.-Y., Choi H.-L. Minimizing Communications in Decentralized Greedy Task Allocation. *Journal of Aerospace Information Systems*. 2019. Vol. 16, no. 8. P. 340–345. DOI: 10.2514/1.I010624.
13. Raja S., Habibi G., How J. P. Communication-Aware Consensus-Based Decentralized Task Allocation in Communication Constrained Environments. *IEEE Access*. 2022. Vol. 10. P. 19753–19767. DOI: 10.1109/ACCESS.2021.3138857.
14. Skaltsis G. M., Shin H.-S., Tsourdos A. A Review of Task Allocation Methods for UAVs. *Journal of Intelligent & Robotic Systems*. 2023. Vol. 109. Article 76. DOI: 10.1007/s10846-023-02011-0.
15. Shah S., Dey D., Lovett C., Kapoor A. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. *Field and Service Robotics*. Springer, 2017. P. 621–635. DOI: 10.1007/978-3-319-67361-5_40.
16. Furrer F., Burri M., Achtelik M., Siegwart R. RotorS — A Modular Gazebo MAV Simulator Framework. *Simulation, Modeling, and Programming for Autonomous Robots*. Springer, 2016. P. 595–625. DOI: 10.1007/978-3-319-26054-9_23.

References

1. Chung, S.-J., Paranjape, A. A., Dames, P., Shen, S., & Kumar, V. (2018). A survey on aerial swarm robotics. *IEEE Transactions on Robotics*, *34*(4), 837–855. https://doi.org/10.1109/TRO.2018.2857475.
2. Oh, K.-K., Park, M.-C., & Ahn, H.-S. (2015). A survey of multi-agent formation control. *Automatica*, *53*, 424–440. https://doi.org/10.1016/j.automatica.2014.10.022.
3. Choi, H.-L., Brunet, L., & How, J. P. (2009). Consensus-based decentralized auctions for robust task allocation. *IEEE Transactions on Robotics*, *25*(4), 912–926. https://doi.org/10.1109/TRO.2009.2022423.
4. Dias, M. B., Zlot, R., Kalra, N., & Stentz, A. (2006). Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, *94*(7), 1257–1270. https://doi.org/10.1109/JPROC.2006.876939.
5. Boukoberine, M. N., Zhou, Z., & Benbouzid, M. E. H. (2019). A critical review on unmanned aerial vehicle power supply and energy management: Solutions, strategies, and prospects. *Applied Energy*, *255*, Article 113823. https://doi.org/10.1016/j.apenergy.2019.113823.
6. Qin, T., Li, P., & Shen, S. (2018). VINS-Mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, *34*(4), 1004–1020. https://doi.org/10.1109/TRO.2018.2853729.

ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

7. Mur-Artal, R., & Tardós, J. D. (2017). ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, *33*(5), 1255–1262. https://doi.org/10.1109/TRO.2017.2705103.
8. Paredes-Vallés, F., Hagenaars, J. J., Dupeyroux, J., Stroobants, S., Xu, Y., & de Croon, G. C. H. E. (2024). Fully neuromorphic vision and control for autonomous drone flight. *Science Robotics*, *9*(86), Article eadi0591. https://doi.org/10.1126/scirobotics.adi0591.
9. Gerkey, B. P., & Matarić, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, *23*(9), 939–954. https://doi.org/10.1177/0278364904045564.
10. Korsah, G. A., Stentz, A., & Dias, M. B. (2013). A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research*, *32*(12), 1495–1512. https://doi.org/10.1177/0278364913496484.
11. Ponda, S., Redding, J., Choi, H.-L., How, J. P., Vavrina, M., & Vian, J. (2010). Decentralized planning for complex missions with dynamic communication constraints. In *Proceedings of the 2010 American Control Conference (ACC)* (pp. 3998–4003). IEEE. https://doi.org/10.1109/ACC.2010.5531232.
12. Kim, K.-S., Kim, H.-Y., & Choi, H.-L. (2019). Minimizing communications in decentralized greedy task allocation. *Journal of Aerospace Information Systems*, *16*(8), 340–345. https://doi.org/10.2514/1.I010624.
13. Raja, S., Habibi, G., & How, J. P. (2022). Communication-aware consensus-based decentralized task allocation in communication constrained environments. *IEEE Access*, *10*, 19753–19767. https://doi.org/10.1109/ACCESS.2021.3138857.
14. Skaltsis, G. M., Shin, H.-S., & Tsourdos, A. (2023). A review of task allocation methods for UAVs. *Journal of Intelligent & Robotic Systems*, *109*, Article 76. https://doi.org/10.1007/s10846-023-02011-0.
15. Shah, S., Dey, D., Lovett, C., & Kapoor, A. (2017). AirSim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics* (pp. 621–635). Springer.
16. Furrer, F., Burri, M., Achtelik, M., & Siegwart, R. (2016). RotorS—A modular Gazebo MAV simulator framework. In *Simulation, Modeling, and Programming for Autonomous Robots* (pp. 595–625). Springer. https://doi.org/10.1007/978-3-319-26054-9_23.

Бабілунга Оксана Юріївна; Oksana Babilunga, ORCID: https://orcid.org/0000-0001-6431-3557
Дмитренко Дмитро Олександрович; Dmytro Dmytrenko, ORCID: https://orcid.org/0009-0004-9948-9682
Андріянов Олександр Вікторович; Oleksandr Andriianov, ORCID: https://orcid.org/0000-0001-7037-0523

Received November 05, 2025

Accepted December 01, 2025