Research Article

Reinforcement learning versus swarm intelligence for autonomous multi-HAPS coordination

Ogbonnaya Anicho[1] · Philip B. Charlesworth[1] · Gurvinder S. Baicher[1] · Atulya K. Nagar[1]

Received: 12 January 2021 / Accepted: 12 May 2021
Published online: 26 May 2021

Abstract
This work analyses the performance of Reinforcement Learning (RL) versus Swarm Intelligence (SI) for coordinating multiple unmanned High Altitude Platform Stations (HAPS) for communications area coverage. It builds upon previous work which looked at various elements of both algorithms. The main aim of this paper is to address the continuous state-space challenge within this work by using partitioning to manage the high dimensionality problem. This enabled comparing the performance of the classical cases of both RL and SI establishing a baseline for future comparisons of improved versions. From previous work, SI was observed to perform better across various key performance indicators. However, after tuning parameters and empirically choosing suitable partitioning ratio for the RL state space, it was observed that the SI algorithm still maintained superior coordination capability by achieving higher mean overall user coverage (about 20% better than the RL algorithm), in addition to faster convergence rates. Though the RL technique showed better average peak user coverage, the unpredictable coverage dip was a key weakness, making SI a more suitable algorithm within the context of this work.

Keywords Swarm intelligence · Reinforcement learning · Multi-HAPS · Autonomous coordination

1 Introduction

This work presents the concluding results of the comparative analysis of Reinforcement Learning (RL) and Swarm Intelligence (SI) for autonomous coordination of multiple High Altitude Platform Stations. It complements some of the work and results published in previous papers [1, 2], which covered some introductory thoughts and key concepts like reward signal design. However, in this paper, the challenges of state-space size and the performance of the algorithms under extended simulation runs were explored. Addressing the state-space constraint further was necessary for ensuring that only the classical case of RL was implemented since the other option would have been using deep q-learning or higher RL implementations. In the previous studies, SI consistently showed some superior performance, and this paper was deemed necessary to ensure that the classical cases of both algorithms were compared eliminating biases based on improved versions of either RL or SI. This will establish a baseline against which future iterations of the algorithms will be measured in the context of this work.

The International Telecommunications Union (ITU) defines HAPS as 'a station located on an object at an altitude of 20 to 50 Km and at a specified, nominal, fixed point relative to the earth' [3]. This region of the atmosphere (known as the stratosphere) is characterised by mild wind activity suitable for hosting aerial platforms with minimal station keeping [4]. From that altitude, HAPS can provide persistent communications coverage to ground users

✉ Ogbonnaya Anicho, anichoo@hope.ac.uk; Philip B. Charlesworth, charlep@hope.ac.uk; Gurvinder S. Baicher, baicheg@hope.ac.uk; Atulya K. Nagar, atulya.nagar@hope.ac.uk | [1]Liverpool Hope University, Liverpool, United Kingdom.

while combining the technical advantages of terrestrial and satellite communication systems [4–6]. The potentials to project large footprints with low signal latency similar to terrestrial systems make HAPS quite suitable for communications services. Furthermore, it can be easily retrieved for maintenance, redeployed or refitted with new service payloads to meet various operational scenarios, unlike terrestrial or satellite systems [2, 7].

Unmanned HAPS can be considered distinct from other unmanned aerial vehicles (UAVs) in configuration, application and specifically operation altitude; other UAVs may be classified as Low Altitude Platforms (LAPs) [8]. Technically, HAPS differ from UAVs or LAPs [9], which operate within the troposphere, and typically have smaller footprints and lower endurance capabilities. The current capability for operating unmanned HAPS systems requires about two to four operators working on different aspects of the system e.g., mission planning, flight control and sensor operation; this mode of operation can be described as *many-to-one* ratio [10, 11]. This implies that implementing multiple HAPS with current operator configuration will be challenging in both technical and economical terms. In a multi-HAPS context, the operational complexity and cost will likely scale making the business case for such a solution unjustifiable. The implementation of multiple HAPS can significantly increase area coverage capacity, i.e., through a network of HAPS. Reversing the *many-to-one* ratio to *one-to-many* can be considered a key objective of the multiple HAPS coordination research. Incorporating some level of autonomy into the HAPS operating framework may mitigate the operating ratio issue. This will minimise or in some cases eliminate the use of human or manual input for multi-HAPS operations. Some of the contributions of this paper are;

- Analysis of the state-space constraint in the multi-HAPS problem context.
- Performance analysis of RL and SI for multi-HAPS coordination.
- Unique insights into challenges with designing and implementing RL for coordination.
- Highlighting challenges with multi-HAPS coordination in the communications coverage domain.

The paper is laid out as follows: Section 1 introduces the work and puts it in some context, while Sect. 2 covers related works and relevant background. Section 3 addresses key concepts of RL and SI. Section 4 summarises the modelling and simulation methodology applied in this work. In Sect. 5, the simulation results and analysis are covered in some detail, while Sect. 6 highlights key findings. Finally, Sect. 7 draws conclusions on the work and considers future work.

2 Literature review

The key motivation for solving the multiple HAPS coordination problem is to minimise human input in multi-HAPS/UAV networks, thereby improving efficiency and lowering operational cost. In the literature, very limited publications addressed HAPS specifically unlike UAV-themed research. However, reviewing UAV-themed publications provided useful context for aligning HAPS within the aerial vehicle space. The authors considered publications where SI or RL was applied to UAV-based scenarios and made useful extrapolations from those. The HAPS coordination problem can be abstracted as a biologically inspired swarm [12], simulating the HAPS platforms as biological agents foraging for food (ground users) within the area of interest. The multiple HAPS coordination problem also satisfies the general principles accepted for modelling the broad behaviour of swarms of homogeneous agents, mainly the proximity and quality principles which define how swarms develop objectives and respond to quality factors like food and safety [13]. The RL and SI were deemed suitable solution candidates for the research problem because of the prospect for achieving learning with RL, and the simple but powerful SI techniques proven from its use in swarm robotics [14]. In the literature, different applications of the SI technique to various problems are available, but this work addresses only UAV-related applications. For instance, Particle Swarm Optimisation (PSO) for coordinating multiple UAVs [15]; swarm intelligence for real-time UAV coordination for search operations [16]; and swarm intelligence-based coordination for UAV swarm self-segregation, aggregation and cohesion [17]. Other SI approaches include the use of coordination protocols comprising of SDN-based UAV communication and topology management algorithms [18]; a proactive topology-aware scheme tracking network topology changes [19] . Though in the literature different applications of SI in UAV coordination have been cited, the area coverage scenario using fixed-wing unmanned multiple HAPS platforms is largely unavailable.

In the RL domain, some relevant applications to the problem domain in the literature are distributed Multi-Agent Reinforcement Learning (MARL) algorithm proposed by [20]; adaptive state focus Q-learning by [21]; area coverage control in conjunction with reinforcement learning [22]; application of reinforcement learning (using Q-learning) to the flocking problem [23]; Apprenticeship Bootstrapping via Inverse Reinforcement Learning using Deep Q-learning (ABS via IRL-DQN) by [24]; and decentralised deep reinforcement learning algorithm [25]. It is not the aim of this paper to provide

an exhaustive list of all RL based UAV applications but to identify implementations that put this work in the right context, particularly for multi-HAPS implementations. More details of related works have been captured in the authors' preceding publication [26].

Autonomy as a concept has levels and largely contingent on application specifics, function and design considerations [2, 7, 27]. There are very few publications where the autonomy of HAPS for communications coverage has been treated specifically. Giagkos et.al [28] compared different approaches for the autonomous capability for unmanned aircraft used for communications. The approaches considered were a non-cooperative game (NCG) and the use of evolutionary algorithm (EA) to plan flying strategies for the unmanned aerial fleet, but the vehicles considered in this work were not solar-powered. Gangula et al. [8] proposed the application of low-altitude platform (UAV) relay for providing end-to-end connectivity and applying autonomous placement algorithms. Moraes & Freitas [29] considered the development of an autonomous and distributed movement coordination algorithm for UAV swarms though in the context of communication relay networks and exploratory area surveillance missions. Choi et al. [30] implemented a new coverage path planning (CPP) for multiple UAV scenario for an aerial imaging use case which differs from a communications area coverage scenario. Stenger et al. [31] highlighted the growing importance of autonomous operation for UAVs and investigated the use of a cognitive agent-based architecture *Soar* for the decision-making process in autonomous systems.

3 RL and SI—summary of concepts

In RL, a model of the environment is required in model-based (planned) methods; otherwise, the model-free (for explicitly trial-and-error learners) is used [32, 33]. The model-based RL approach employs the transition probability (TP) model, which requires the generation of the system's state transition model. This method can be computationally burdensome, especially for large-scale problems. However, the model-free approach does not require computation, storing and manipulating TPs but uses stochastic approximation [34]. RL problems can be formalised with the Markov Decision Process (MDP) framework and can be solved using specific approaches or algorithms. The three (3) main classes of algorithms that can be used to solve RL problems are Dynamic Programming (DP), Monte Carlo (MC) and Temporal Difference (TD) Learning [32, 35]. TD is a model-free approach that learns directly from experiences by updating state-values as they are visited. This means learning on the go, rather than wait till the end of the episode (e.g. Q-learning algorithm)

Classical Q-learning was adopted for this work due to its universal application and ease of design so far as the state-action space is computationally manageable [34]. The central idea in the Q-learning algorithm is to store the state-action pair value Q(s, a) called Q-values of each iteration as the agents interact with the environment. At the beginning of the simulation, the Q-values are initialised to zero and stored in a table or an array (each HAPS (agent) maintains its own Q-table). The agent visits some state s and takes action a, and then transits another state. The immediate reward gained from this action is stored and the Q-value updated using the following mathematical relationship [32, 34];

$$Q(s,a) \approx (1-\alpha)Q(s,a) + \alpha\left[r + \gamma\, max_{t+1}Q(s_{t+1}, a_{t+1})\right] \quad (1)$$

where r denotes the reward at time t, $0 < \alpha < 1$is a given learning rate and γ is discount factor. The expression is used to update the Q-table until the values converge to a near-optimal solution. In the simulation carried out, the HAPS are defined as agents and user mobility modelled as part of the environment and 'states' are mapped to preselected and fixed coordinates (i.e. beacons). Though the 'states' are mapped to fixed coordinates, the user distribution within these states is not fixed but follows the user mobility model. The HAPS (agent) could be in one of a set of states, $\{S_1, S_2, S_3, ...S_n\}$, where S_n is mapped as below

$$S_n : (\phi_n, \lambda_n) \quad (2)$$

where ϕ, λ are the coordinates (latitude and longitude) mapped to S_n. The model-free approach applied in this work does not require the computation, storage and manipulation of transition probabilities but uses stochastic approximation.

The agent can execute two action set:$\{A_1, A_2\}$ i.e. 'Relocate from' or 'Remain within' state S_n;

$$A_s : \{A_1, A_2\} \quad (3)$$

As mentioned earlier, the 'states' are mapped to fixed coordinates, if the decision to 'relocate from' a current state is taken, the agent transits to an initially randomly chosen next 'state' (i.e. coordinate), and subsequently greedily chosen. Conversely, if the agent decides to 'remain within' a current state it simply stays within the same coordinate.

Reward (or penalty) signals are fed back to the HAPS to reinforce actions that influence goals (e.g., maximise user coverage) positively or otherwise. A random walk user mobility model was implemented in this work. The reward signal is mapped directly to the number of users

(U) covered in each state (s) by the HAPS after taking a specific action (a) and is given by;

$$r(s,a) \mapsto U_s \tag{4}$$

The SI algorithm applied to this problem was a variant of the classical bee algorithm with modifications to suit the uniqueness of the application domain. In the context of this work, the algorithm was modelled such that the HAPS network is abstracted as a swarm (HAPS platforms) foraging for food (users) around the simulation scenario. Fundamental concepts central to achieving SI are self-organisation and division of labour [12, 13]; both of which are reflected in the logic behind the algorithm. The participating HAPS in the swarm exchange data as they explore the environment were akin to foraging. Theoretically, the source quality is measured and is a relation between gain and cost given as [36]:

$$\text{SourceQuality}_i = (\text{Gain}_i - \text{Costs}_i)/(\text{Costs}_i) \tag{5}$$

i.e. the gain versus cost computation. This is mapped to the computed distance to be travelled (cost) by HAPS in relation to the potential number of users that will be covered (gain).

4 Modelling and simulation background

The modelling and simulation process or methodology used for this research involved aggregating various models of key elements of the simulated phenomenon. Some key models simulated in software are the HAPS flight dynamics model, propulsion model, navigation model, inter-HAPS link and solar energy model. These models are simplified enough to meet the specific scope and interest of the research using standard mathematical and physics models of aerodynamics and communications without compromising theoretical or practical considerations. Further reference and details of the modelling and simulation methodology implemented in this work are covered in [7, 26, 37].

The parameters in Table 1 describe the HAPS system communications and link budget parameters which ultimately defines the profile of the service segment, e.g., HAPS communications payload power and link data rates. The link budget is based on a payload power of 80 Watts, with the simulated HAPS network supporting about 500 subscribers/users spread over a large area (typical coverage density profile for HAPS). In such thinly populated scenarios, terrestrial networks would not be economical and satellites may be too expensive and ineffective.

5 Results and analysis

A simulation of four (4) HAPS covering about 500 users moving randomly around a specified geographical area was performed, with each HAPS in the network carrying out its own Q-learning individually. The mobility model of the ground users is random and unknown to the HAPS.

Each HAPS is simulated at 20km altitude, 22 degrees elevation (angle from the user's local horizon to the HAPS), and 135 degrees HPBW, with a footprint covering about 7160km^2 and the total area of interest covering about 102,101km^2. The size of the coverage area and the HAPS footprint was carefully designed to effectively accommodate only 4 HAPS in order to allow room for testing out the coordination algorithm. The ground users are randomly distributed across the geographical area, and the ultimate goal is to maximise user coverage through autonomous coordination of the 4 HAPS.

5.1 Benchmarking and algorithm evaluation

In order to provide a method to validate the algorithms and simulation method, a benchmark was introduced. The benchmark was to evaluate the performance of the multiple HAPS network without a coordination algorithm. In

Table 1 HAPS system communications and link budget parameters

S/N	Item	Specification	Justification
1	Half power beam width (HPBW)	145 degrees	Specific to model
2	Normalised signal to noise ratio (Eb/No)	10 dB	Assumed for link
3	EIRP	Depends on slant range	Power to support 1 subscriber at edge of cover
4	Data rate	10 Kbit/s	Desired link data rate
5	HAPS transmitter antenna efficiency	0.75	Assumed for model
6	Ground receiver antenna gain	1	Assumed for model
7	Signal frequency	7 GHz	Assumed for model
8	System noise temperature	350K	Standard

this scenario, the HAPS randomly searched and explored the environment without any form of coordination or logic. This benchmark represents the worst-case scenario where no coordination algorithm is applied ('Do Nothing' approach), the HAPS essentially navigate randomly without cooperating or exchanging information with each other. The outcome of this experiment or scenario provided a means to compare and validate the performance of the RL and SI coordination algorithms as applied in this work. Also note that in computing coverage, no user can be covered by more than one HAPS at a time. This is also applicable to the HAPS; this way duplication of coverage is avoided. It is assumed in this simulation that hard hand-off is implemented in the system (as each HAPS completely releases a user, before the next HAPS attaches it). In the benchmark scenario, no coordination algorithm is applied; the performance of this benchmark approach is shown and analysed comparatively with the SI and RL methods at the end of this section.

5.2 RL algorithm—application, simulation and results

This section details important aspects of implementing the RL technique and provides insight based on the simulation and results analysis. In order to apply the technique correctly, all the elements and concepts of the technique were analysed and mapped to the HAPS scenario. Elements such as the state space, action set, reward signals and all relevant components of the RL methodology were analysed, and insights gained were directly applied. This approach enabled the use of empirical means to understand the impact of the parameters on different aspects of the algorithm's implementation.

5.2.1 Reward signal design and RL hyper-parameters

The reward signal design is very critical in any RL-based implementation. In this work, the reward signal was designed to reinforce more user coverage by the HAPS. In other words, the reward was directly translated from the user coverage metrics. The number of users covered by each HAPS in any given state is directly used as the reward signal. Essentially, the HAPS evaluated the utility of each state and action based on the user coverage metric. Hyper-parameters are those parameters that are fixed before the RL algorithm is applied to the simulation. Two hyper-parameters critical to the RL algorithm as applied to this work are epsilon-greedy (ϵ) and learning rate (α). The discount factor (γ), which is the third hyper-parameter was not necessarily of much significance in the context of this work, as the problem scenario by design favoured immediate rewards over delayed future rewards. Essentially, the HAPS prefer to get maximum immediate rewards (higher user coverage) over any form of delayed future rewards. In other problem domains, the reverse is the case, for instance, some games are designed to favour delayed future rewards. However, delayed future rewards do not benefit the HAPS coordination problem scenario which favours very low discount factor; discount factors range from 0 to 1. In this work, a discount factor of between 0 to 0.2 was used. In most experiments, the value was fixed at 0, emphasising immediate rewards over delayed future rewards. However, in some cases, the hyper-parameters were designed to decay over time (parameter tuning) and not necessarily fixed. The impact of high and low learning rates was tested and based on results parameter tuning was adopted where necessary. The details of the reward signal design and hyper-parameter tuning were covered in a previous paper [1].

5.2.2 Analysis of state-space dimension

In this work, the state-space problem was analysed further and defined by partitioning the area of interest into equally spaced beacons the size of towns or regions identified by their coordinates. The state-space sizes of 4, 8, 12, 16, 20 and 24 were arbitrarily chosen and defined as prefixed set of states (see Fig. 1). Essentially, it is like dividing up a location into very large regions of 4, 8, 12, 16, 20 or 24 identified by the unique coordinates of those regions. These beacons (or coordinates) remain fixed throughout the duration of any run of the simulation. The HAPS explore the area of interest by visiting the beacons (fixed coordinates) storing the 'value' of each visited 'state'. This technique reduced the states space to a manageable size while leveraging the mathematical and computational convenience of using geographical coordinates. In order to investigate the impact of state vectors on the performance of the algorithm, an experiment was designed to run different state-space vectors. Each state-space dimension was run under the same conditions and the global performance measured as shown in Fig. 2.

The global coverage for each predefined state space was measured and compared for any statistically significant results. The exploration–exploitation dilemma which defines the binary decision set of either to explore the environment or exploit current location had some effect on the results. The dynamics of the exploration–exploitation dilemma is a well-known and critical aspect of any RL implementation. The 'noisy' coverage plot in Fig. 2 highlights this phenomenon clearly, as the HAPS explore (relocates to new location) or exploit (remains in the same location) with varying outcomes on user coverage. The HAPS will randomly cover less or more users due to this exploration (relocation) or exploitation (remain) decision

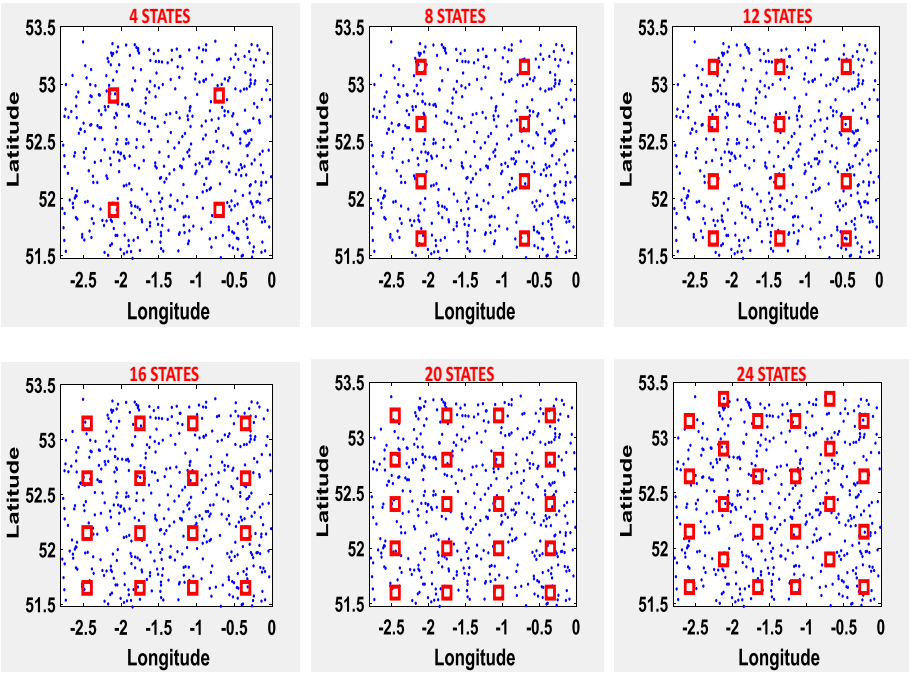

Fig. 1 States space sizes of 4, 8, 12, 16, 20 and 24 (fixed latitude and longitude coordinates)

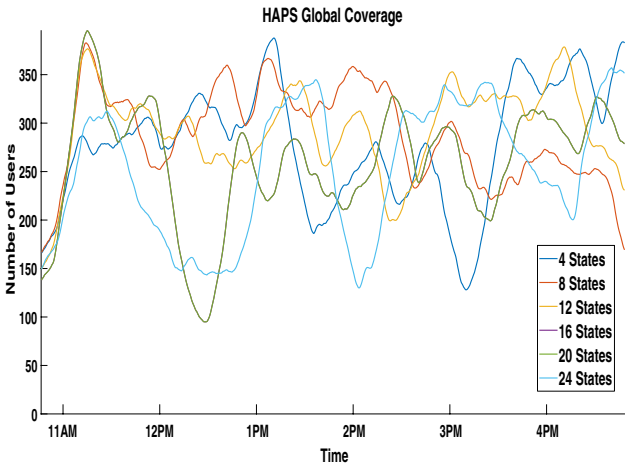

Fig. 2 Global coverage for 4, 8, 12, 16, 20 and 24 states space

loop. One challenge of the RL technique is to manage this 'dilemma' by finding the right exploration–exploitation balance. Choosing to either explore or exploit infinitely will negatively impact the algorithm, as the probability of remaining in suboptimal states becomes higher. The epsilon-greedy (ϵ-greedy) hyper-parameter controls the exploration–exploitation decision and determines how the HAPS balances this decision loop. Due to this noisy coverage signal, statistical tools were used to analyse the simulation results to establish any statistically significant outcomes.

5.2.3 Analysis of variance (ANOVA) test for RL states experiment

To further test the results of the different RL states and establish statistical significance in achieved global coverage for each state, an ANOVA test was carried out. The ANOVA data in Table 2 show the outcome of the mean variability among all the states (groups), i.e. 4, 8, 12, 16, 20 and 24. The one-way ANOVA tests the null hypothesis that all group means are equal against the alternate hypothesis that at least one group (state) is different from the other states [38]. The data in the ANOVA table (Table 2) show the source of the variability (Source), the sum of squares from each source (SS), degrees of freedom (df), mean squares for each source (MS), *F*-statistic (F) and Prob>F (*p* value) [38]. The row of the table provides information about variability between the groups (columns) and within the groups (error). The *F*-statistic is used to test the

Table 2 ANOVA data—coverage variance for all states space

Source	SS	df	MS	F	Prob>F
Columns	37293295.26	5	7458659.052	2045.863862	0.01
Error	481213956.3	131994	3645.72599		
Total	518507251.6	131999			

significance of any observed variability. The *p* value is then used to decide on the significance of this variability by comparing it with the significance level set for the experiment (normally 0.05 or 0.01). Essentially, the *p* value is the probability that *F*-statistic (F) can take a value that is larger than the computed test statistic [38]. For this work, a significance level of 0.01 is used; therefore, if the *p* value for the computed *F*-statistic is smaller than 0.01 (significance level), the null hypothesis is rejected. Using a 0.01 significance level implies only 1% risk (99% confidence level), while the commonly used 0.05 significance level implies about 5% risk (95% confidence level). From the ANOVA table (Table 2), computed for the RL states experiment, the *p* value is < 0.01, which implies that the null hypothesis can be rejected; the alternate hypothesis is, therefore, true, i.e. the group mean of at least one of the states is different. The box plot (Figure 3) provides graphical assurance that the group means are different, though the box plot shows the median of the groups (the line inside the box). The main aim of the test was to establish that there is a difference in the performance (group means) of the different states. However, to know which pairs of the different states (groups) are significantly different, a multiple comparison test was carried out (see Fig. 4). This test shows the comparison interval between the group means and provides a way to statistically and graphically establish which group means differ. Two groups are concluded to be significantly different if their intervals do not overlap (an overlap indicates no significant difference) [38].

From the comparison test, it can be seen that there is no overlap in the graph of the states. Though the states space performance showed marginal improvement across (group means vary slightly). However, the 12 states space vector showed better group mean (about 295 users) compared to the rest. The 16, 20 and 24 states had lower mean

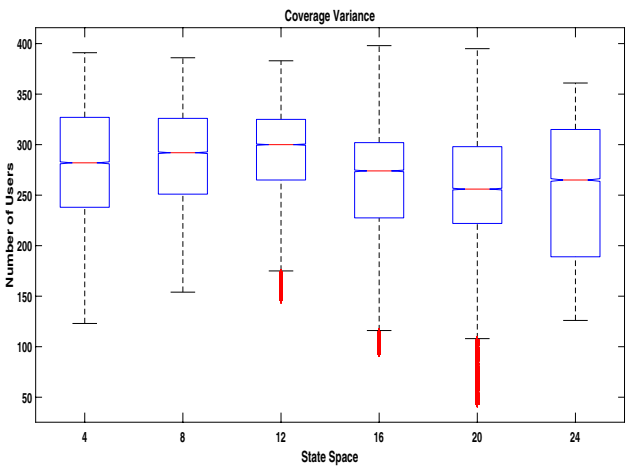

Fig. 3 Global coverage box plot for the different states space

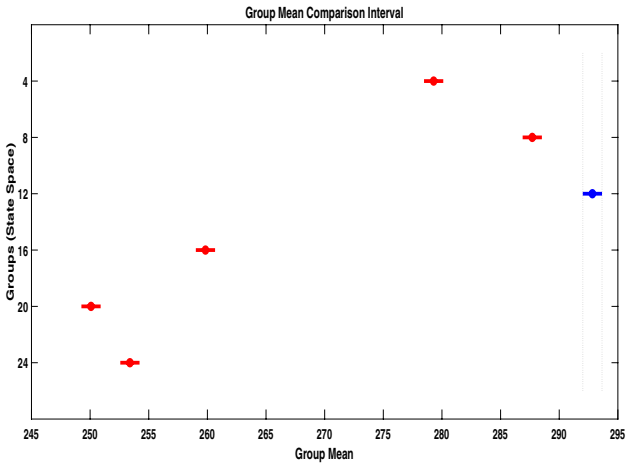

Fig. 4 Group mean comparison interval for the different states space

coverages of about 260, 250 and 254 users, respectively, while states space 4 and 8 had slightly better performance with a mean coverage of about 280 and 288 users, respectively. The statistical implication of the results and analysis strongly supports the alternate hypothesis that there is a difference in performance based on the variance of the group means. Furthermore, the multiple comparison test provided statistically supported evidence that the 12-state global coverage output is better with the group mean of almost 295 users representing almost 60% global coverage. Based on this result 12 states space partitions were used as the best case state-space definition within the context of this work. The results also suggested that coverage performance declined with an increase in the number of the states after 12 states i.e. increasing states space beyond 12 did not improve coverage. This may be explained by the increased proximity of the HAPS to each other as the states 'physical' size shrunk with increase in states. State-space design for HAPS-related problems can impact coordination performance and hinder user coverage goals as demonstrated by this experiment.

5.3 SI results and analysis

The results and analysis of the SI-based algorithm implemented for the coordination of the HAPS are explained in this section. Applying the SI algorithm in the same scenario with the RL and benchmark techniques will enable a fair comparison of their individual performances and capabilities. The behaviour and performance of the SI algorithm are captured graphically and analysed accordingly.

Figure 5 shows the behaviour of the SI algorithm over an extended run. As demonstrated from the graph, the SI algorithm converged to a solution early in the simulation.

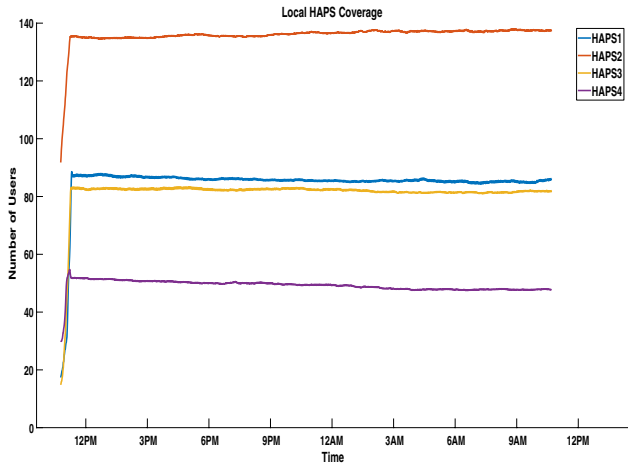

Fig. 5 SI local coverage—extended run (86400 time steps)

All the HAPS can be seen rising in coverage from the initial level at the starting epoch. Each HAPS converged to a solution within short periods and maintained consistent coverage all through the simulation time.

Unlike the RL method, the SI method was able to converge to a good solution at a relatively short time. The global coverage in Fig. 6 clearly highlights this trend.

5.4 Comparing performance of the algorithms

The sections above have been dedicated to investigating and analysing the behaviour of the various algorithms. At this point, statistical means will be used to compare the performance of the three methods—Benchmark, RL and SI techniques. The ANOVA test will be used for analysing the performance of the algorithms. The experiments were conducted under the same set of conditions. For this comparison, the global coverage was considered as this provided a high-level or global-level review of the performance of the algorithms.

The SI method converged faster (see Figure 7) as recorded in all previous runs. The SI algorithm also maintained better mean global coverage at 71%, while RL and Benchmark methods posted 51% and 44%, respectively. Statistically, the results as shown by the *p* value of zero (see Table 3) are significant with differences in the population means. The box plot of the extended run of the experiment provided the graphical assurance of the performance analysis. However, the SI method had a high number of outliers indicating that the HAPS at certain points covered an unusually small number of users and may not have attempted to relocate to correct this. The reluctance to relocate is due to the rules-based minimum distance of the SI-based algorithms. In the SI implementation, the HAPS are not allowed to relocate if the minimum distance constraint would be violated. As such, the HAPS remain in locations where user density may have shrunk due to user mobility. This is one area where the random exploration of the RL algorithm shows superiority as such suboptimality in coverage will be handled differently. This is the reason the RL result has far fewer outliers in comparison (Figs. 8, 9).

6 Discussion

The performance of the RL and SI algorithms as analysed earlier is discussed further below:

- Convergence: SI algorithm showed consistent convergence behaviour and consistently converged within 60–90 minutes. The RL algorithm showed weaker

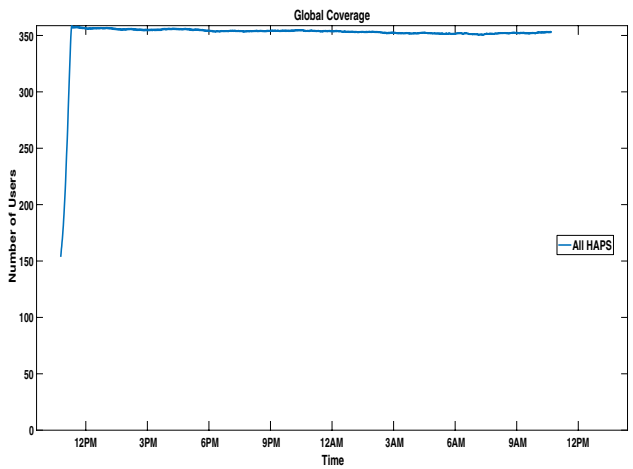

Fig. 6 SI global coverage—extended run (86400 time steps)

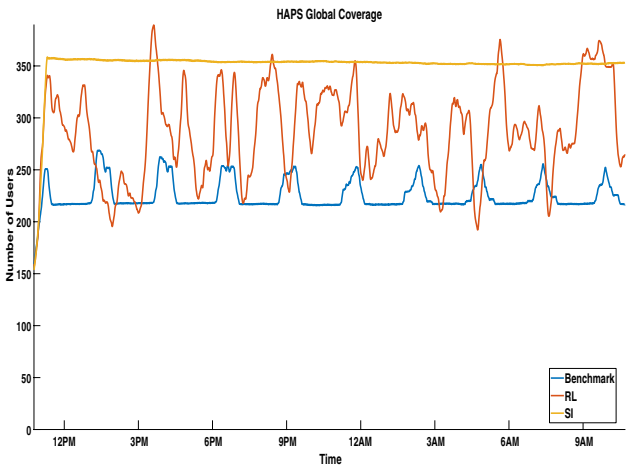

Fig. 7 Global coverage comparison of all algorithms—extended run (86400 time steps)

Table 3 ANOVA data: coverage variance—all algorithms (extended run)

Source	SS	df	MS	F	Prob>F
Columns	685948687.9	2	342974343.9	399462.9933	0.01
Error	222543571	259197	858.5885293		
Total	908492258.9	259199			

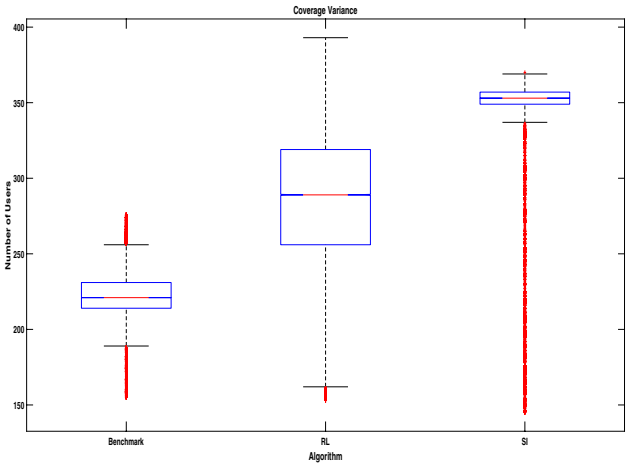

Fig. 8 Global coverage box plot: all algorithms—long run (86400 time steps)

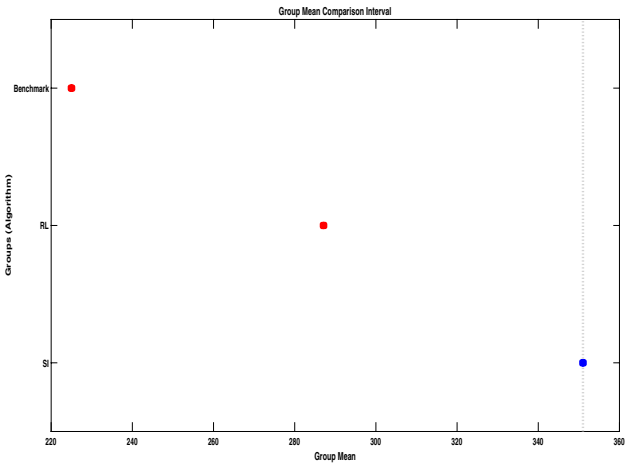

Fig. 9 Group mean comparison interval: all algorithms—long run (86400 time steps)

convergence behaviour as a result of its exploration strategy.

- Mean Global User Coverage: The SI algorithm consistently achieved better mean global user coverage of more than 70%, while RL recorded about 51%. The SI algorithm performed about 20% better than the RL technique.
- Peak Coverage: The RL algorithm achieved higher peak coverage of about 80% mean global coverage (though for short periods of time).
- Coverage Availability: The SI algorithm maintained better coverage availability due to consistent convergence, while the RL algorithm had more dips in coverage.
- Exploration Policy: The RL algorithm explored the environment more and was rewarded with peak coverages and equally 'punished'' with higher coverage dips.
- Exploitation Policy: SI exploited the environment aggressively and only explored very minimally.
- Comparison with Benchmark: RL and SI algorithms achieved higher mean coverage against the benchmark. SI performed about 27% better than the benchmark, while RL performed about 7% better.

6.1 RL and SI comparative analysis

The comparative analysis of the RL and SL algorithms reveals certain insights about their implementation within the multi-HAPS coordination problem context. Some of these novel insights are summarised below:

- The continuous state-space problem complicates the implementation of RL techniques but not a factor in SI implementation.
- The size of the state-space partition impacts the performance of the RL algorithm, so partitioning size has to be carefully selected.
- The main factor challenging the convergence of the RL algorithm was not the nonstationary stochastic environment but the exploration strategy of the HAPS.
- The SI techniques have superior convergence profile and could improve system reliability, an important factor for communications use cases.

7 Conclusions and future work

This work concludes the comparative analysis of the performance of Reinforcement Learning (RL) and Swarm Intelligence (SI) in the Multi-HAPS coordination problem context. In previous papers, the authors have laid out other important aspects of this analysis, for instance, reward signal designs, hyper-parameter impact on RL algorithms. This paper focused on addressing one of the critical

challenges highlighted from previous work, i.e. the continuous state-space challenge for RL implementation. The impact of using some form of partitioning was explored as a means of solving the continuous state-space problem. This was necessary to ensure that only the classical cases of both algorithms were compared.

In the final analysis, it was established from the work that SI-based approach performed better than classical RL techniques like Q-learning, covering more than 70% of the users consistently compared to 51 and 44% posted by the RL and Benchmark techniques, respectively. The SI algorithm demonstrated more stable and consistent results compared to the RL algorithm in the multi-HAPS coordination problem scenario; however, it is important to highlight the peak user coverage of the RL technique. This work reveals that SI may be best suited for the communications area coverage problem where reliability and network availability are key.

Future work will consider other variations or implementations of RL like Deep Q-Learning against SI techniques. This will involve going beyond the classical cases of both algorithms. The results from this work will serve as a baseline to measure improvements as higher versions of the algorithms are implemented. The rules-based approach of SI and its relatively low cost on computation resources will provide a good benchmark against learning algorithms and associated implementation overheads.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Anicho Ogbonnaya, Charlesworth Philip B, Baicher Gurvinder S, Nagar Atulya (2020) *Reinforcement Learning for Multiple HAPS/UAV Coordination: Impact of Exploration–Exploitation Dilemma on Convergence*, volume 1138 of Soft Computing for Problem Solving 2019. Advances in Intelligent Systems and Computing. Springer, Singapore
2. Anicho O, Charlesworth P. B, Baicher G. S, Nagar A, Buckley N (June 2019) Comparative study for coordinating multiple unmanned haps for communications area coverage. In 2019 International Conference on Unmanned Aircraft Systems (ICUAS), pages 467–474
3. International Telecommunications Union (ITU) (2016) Terms and definitions. Radio Regulations Articles
4. d Oliveira Flavio, Melo Francisco, Campos Tessaleno (2016) High-altitude platforms - present situation and technology trends. J Aerospace Technol Manag 8(249–262):09
5. David Grace, Mihael Mohorcic (2011) Broadband Communications via High Altitude Platforms. Wiley
6. ITU (2017) Identifying the Potential of New Communications Technologies for Sustainable Development. Working Group on Technologies in Space and the Upper-Atmosphere, Technical report, Broadband Commission For Sustainable Development
7. Anicho Ogbonnaya, Charlesworth Philip B, Baicher Gurvinder S, Nagar Atulya (November 2018) Integrating Routing Schemes and Platform Autonomy Algorithms for UAV Adhoc & Infrastructure Based Networks. In 28th International Telecommunication Networks and Applications Conference (ITNAC). 28th International Telecommunication Networks and Applications Conference (ITNAC), IEEE
8. Rajeev Gangula, Omid Esrafilian, David Gesbert, Cedric Roux, Florian Kaltenberger, Raymond Knopp, (06 2018) Flying Robots: First Results on an Autonomous UAV-based LTE Relay using OpenAirInterface. In SPAWC, (2018) 19th IEEE International Workshop on Signal Processing Advances in Wireless Communications, 25–28 June 2018. Kalamata, Greece, Kalamata, GREECE
9. Yong Zeng, Rui Zhang, Joon Lim Teng (2016) Wireless communications with unmanned aerial vehicles: opportunities and challenges. IEEE Commun Mag 54(5):36–42
10. Hehtke V, Kiam J.J, Schulte A (2017) An Autonomous Mission Management System to Assist Decision Making for a HALE Operator. Deutscher Luft-und RaumfahrtKongress
11. Chen Ting B (2016) *Management of Multiple Heterogenous Unmanned Aerial Vehicles Through Capacity Transparency*. PhD thesis, Queensland University of Technology
12. Amrita Chakraborty (2017) Kar Arpan Kumar. Swarm Intelligence, A Review of Algorithms. Springer
13. Hu Yichen (2018) Swarm Intelligence. http://guava.physics.uiuc.edu/~nigel/courses/569/Essays_Fall2012/Files/Hu.pdf. Accessed: 2018-10-23
14. Mullen R. J, Monekosso D. N, Barman S. A, Remagnino P (July 2009) Autonomous Control Laws for Mobile Robotic Surveillance Swarms. In 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pages 1–6
15. Diego Silva, Luiz Oliveira, Mariana Macedo (2012) Filho Carmelo (2012) On the analysis of a swarm intelligence based coordination model for multiple unmanned aerial vehicles. Brazilian Robotics Symposium and Latin American Robotics Symposium, IEEE
16. Varela Gervasio, Caamano Pilar, Orjales Felix, Deibe Alvaro, Lopez-Pena Fernando, Duro Richard (2011) Swarm Intelligence Based Approach for Real Time UAV Team Coordination in Search Operations. In Proceedings of the 2011 Third World Congress on Nature and Biologically Inspired Computing, pages 372–377
17. Caio Monteiro, Diego Silva, Carmelo Bastos-Filho (2013) On the Analysis of a swarm-intelligence coordination model for swarm robots
18. Dapper e Silva T, Emygdio de Melo C. F, Cumino P, Rosário D, Cerqueira E, Pignaton de Freitas E (2019) Stfanet: SDN-based

topology management for flying ad hoc network. IEEE Access 7:173499–173514
19. Hong L, Guo H, Liu J, Zhang Y (2020) Toward swarm coordination: topology-aware inter-UAV routing optimization. IEEE Trans Veh Technol 69(9):10177–10187
20. Xuan Pham Huy, La Hung, Feil-Seifer David, Nguyen Luan (03 2018) Cooperative and distributed reinforcement learning of drones for field coverage
21. Busoniu L, Schtter B, Babuska R (October 2005) Multiagent Reinforcement Learning with Adaptive State Focus. In K Verbeeck, K Tuyls, A Nowe, B Manderick, and B Kuijpers, editors, BNAIC 2005, pages 35–42. Proceedings of the 17th Belgium-Netherlands Conference on Artificial Intelligence
22. Adepegba Adekunle, Miah Suruz, Spinello Davide (2016) Multi-Agent Area Coverage Control using Reinforcement Learning. In Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Society Conference
23. Shao-Ming Hung, Sidney Givigi (2017) A Q-learning approach to flocking with UAVs in a Stochastic environment. IEEE Transac Cybernet 47(1):186–197
24. Nguyen Hung, Bui Lam, Garratt Matthew, Abbass Hussein (July 2018) Apprenticeship Bootstrapping: Inverse Reinforcement Learning in a Multi-Skill UAV-UGV Coordination Task. In M Dastani, G Sukthankar, E Andre, and S Koenig, editors, *Proceedings of the 17th International Conference on Autonomous and Multiagent Systems*, pages 2204–2206
25. Ye Y, Wei W, Geng D, He X (2020) Dynamic Coordination in UAV Swarm Assisted MEC via Decentralized Deep Reinforcement Learning. In 2020 International Conference on Wireless Communications and Signal Processing (WCSP), pages 1064–1069
26. Ogbonnaya Anicho, Charlesworth Philip B, Baicher Gurvinder S, Atulya Nagar (2019) Geographical considerations for implementing autonomous unmanned Solar-HAPS for communications area coverage. Data Sci J Comput Appl Inf 3(1):1–18
27. Chen Hai, Wang Xin min, Li Yan (2009) A Survey of Autonomous Control for UAV. IEEE Computer Society
28. Giagkos Alexandros, Wilson Myra, Tuci Elio, Charlesworth Philip (2016) Comparing Approaches for Coordination of Autonomous Communications UAVs. IEEE International Conference on Unmanned Aircraft Systems (ICUAS)
29. de Moraes RS, de Freitas EP (2017) Distributed control for groups of unmanned aerial vehicles performing surveillance missions and providing relay communication network services. J Intell Robotic Syst 92:645–656
30. Younghoon Choi, Youngjun Choi, Simon Briceno, Mavris Dimitri N (2019) Energy-constrained multi-UAV coverage path planning for an aerial imagery mission using column generation. J Intell Robotic Syst 97:125–139
31. Stenger A, Fernando B, Heni M (2012) Autonomous mission planning for UAVs: a cognitive approach. Deutscher Luft-und Raumfahrtkongress
32. Sutton Richard S, Barto Andrew G (2017) Reinforcement Learning: An Introduction. MIT Press
33. Gu Shixiang, Lillicrap Timothy, Sutskever Ilya, Levine Sergey (2016) Continuous Deep Q-Learning with Model-based Acceleration. In JMLR: W&CP, volume 48. Internation Conference on Machine Learning
34. Abhijit Gosavi (2017) A tutorial for reinforcement learning. Springer
35. Hung David Shao (2015) Reinforcement Learning Approaches to Flocking with Fixed-Wing UAVs in a Stochastic Environment. Master's thesis, Royal Military College of Canada,
36. Koc Ebubekir (2010) The Bees Algorithm: Theory, Improvements and Applications. PhD thesis, University of Wales, Cardiff, United Kingdom
37. Philip Charlesworth (2017) A solar aircraft model for simulations. Liverpool Hope University, Internal Publication
38. MathWorks (2019) One-Way Anova. https://uk.mathworks.com/help/stats/anova1.html. Accessed: 2019-03-04