GA-Guided Task Planning for Multiple-HAPS in Realistic Time-Varying Operation Environments

Jane Jean Kiam
Bundeswehr University Munich
Neubiberg, Germany
jane.kiam@unibw.de

Valerie Hehtke
Bundeswehr University Munich
Neubiberg, Germany
valerie.hehtke@unibw.de

Eva Besada-Portas
Universidad Complutense de Madrid
Madrid, Spain
ebesada@ucm.es

Axel Schulte
Bundeswehr University Munich
Neubiberg, Germany
axel.schulte@unibw.de

ABSTRACT

High-Altitude Pseudo-Satellites (HAPS) are long-endurance, fixed-wing, lightweight Unmanned Aerial Vehicles (UAVs) that operate in the stratosphere and offer a flexible alternative for ground activity monitoring/imaging at specific time windows. As their missions must be planned ahead (to let them operate in controlled airspace), this paper presents a Genetic Algorithm (GA)-guided Hierarchical Task Network (HTN)-based planner for multiple HAPS. The HTN allows to compute plans that conform with airspace regulations and operation protocols. The GA copes with the exponentially growing complexity (with the number of monitoring locations and involved HAPS) of the combinatorial problem to search for an optimal task decomposition (that considers the time-dependent mission requirements and the time-varying environment). Besides, the GA offers a flexible way to handle the problem constraints and optimization criteria: the former encodes the airspace regulations, while the latter measures the client satisfaction, the operation efficiency and the normalized expected mission reward (that considers the wind effects in the uncertainty of the arrival-times at the monitoring-locations). Finally, by integrating the GA into the HTN planner, the new approach efficiently finds overall good task decompositions, leading to satisfactory task plans that can be executed reliably (even in tough environments), as the results in the paper show.

CCS CONCEPTS

• **Theory of computation** → **Evolutionary algorithms**; • **Computing methodologies** → **Planning under uncertainty**; *Multi-agent planning*;

KEYWORDS

Genetic Algorithms, Multiobjective Constrained Optimization, Mission Task-Planning, HAPS

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '19, July 13–17, 2019, Prague, Czech Republic

© 2019 Copyright held by the owner/author(s). Publication rights licensed to the Association for Computing Machinery.

ACM ISBN 978-1-4503-6111-8/19/07...\$15.00

<https://doi.org/10.1145/3321707.3321768>

Figure 1: HAPS platform, Zephyr 7 at launch ©ADS GmbH

ACM Reference Format:

Jane Jean Kiam, Eva Besada-Portas, Valerie Hehtke, and Axel Schulte. 2019. GA-Guided Task Planning for Multiple-HAPS in Realistic Time-Varying Operation Environments. In *Genetic and Evolutionary Computation Conference (GECCO '19)*, July 13–17, 2019, Prague, Czech Republic. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3321707.3321768>

1 INTRODUCTION

High-Altitude Pseudo-Satellites (HAPS) are fixed-wing, completely solar-powered High-Altitude Long-Endurance (HALE) Unmanned Aerial Vehicles (UAVs), which have been developed as an alternative to satellites with fix orbits or aircrafts with permanent refueling needs to monitor ground activities. After the successful tests conducted on Zephyr (see Fig. 1) from Airbus [1] and the upcoming test flights with Odysseus from Boeing [3], the use of these types of platforms in the near future is highly anticipated.

However, given their lightweight build (~50 kg) and weak electro-motors (~1.7 kW), their mission success rate and their impairment risks highly depend on the weather conditions [21]. It is hence essential to consider the time-varying operation environment in the automation of the mission planning in order to ensure the plans feasibility, minimize their risks, optimize their rewards, and reduce operational costs and the operators' workload. To achieve it, weather forecast information can be incorporated into the planner in order to, for instance, predict HAPS mission performance (using the forecasted winds and the cloud coverage) or to identify dangerous zones (e.g. zones with Cumulonimbus clouds) [18, 19].

Moreover, due to the heterogeneous constraints stemming from the operation regulations/environment [11], platform dynamics [21] and mission requirements, existing works on pre-mission HAPS mission planning (i.e. prior to mission execution) often adopt (from

Figure 2: A typical ground-activity monitoring scenario

other robotics or autonomous driving planning architectures [20, 30]) a sequential hierarchical structure. Within it, task planning is performed using classical methods at the top strategic level, followed by motion planning at the lower tactical step to separate the discrete and continuous search spaces. This way of proceeding, used in the Hierarchical Task Network (HTN) planner for a single HAPS presented in [15, 17], is clearly practical for encoding HAPS air traffic regulations and operation protocols at different abstraction levels. However, the planning approaches in both works (brute-force search in [15] and Markov Decision Process (MDP) policies in [17]) do not scale up easily or become even intractable for scenarios with more mission areas and multiple HAPS. In this regard, this work focuses on the extension of automated offline task-planning to multiple HAPS, which is essential because, in practice, several HAPS will be included within the same monitoring operation in order to increase the coverage area (and/or the covering frequency).

In order to increase autonomy of the task-planning of multiple HAPS, we have decided to use a Genetic Algorithm (GA) to guide the search of the best plan of a HTN within the allocated planning time. The decision is based on the tested efficiency of this flexible optimizer over many other types of real mission planners involving aerial vehicles [5, 10, 24–26, 31]. Nevertheless, to the best of the authors' knowledge, this paper (which extends the ideas suggested briefly by the authors in [14]) is the first one that details how to incorporate and implement a GA for multiple HAPS mission planning. Besides, this new approach exploits the capabilities of GA for handling constrained complex combinatorial problems, which are essential to plan feasible and overall good HAPS task-plans for the operation (and to handle properly the HAPS and environment dynamics, safety and operation constraints, and time-dependent mission requirements).

Finally, it is worth noting that although the paper is focused in the development of a planner for a specific multi-HAPS monitoring problem, our approach could be used in other HTN-based planners and/or to incorporate weather forecast uncertainty into the evaluation of the solutions of different types of planners.

2 PROBLEM DESCRIPTION

As documented in [11], HAPS will operate within the controlled airspace. This implies that they can only fly within their allocated airspace, and that their plans must be determined pre-mission and

Table 1: Actions, task and mission areas constraints, rewards involved in HAPS monitoring missions

Main actions (tasks) involved in HAPS planning												
Tasks description			Task names			Task Constraints						
Fly in WA#			WA#			TC1, TC2						
Monitor MA#			MA#			TC1, TC2, TC3, TC4, TC5						
Fly to corridor C#			toC#			TC1, TC2						
Cross corridor C#			crossC#			TC1, TC2						
Fly to LOI#			toLOI#			TC1, TC2, TC3, TC6						
Monitor LOI#			monitorLOI#			TC1, TC2, TC3, TC6						
Task Constraint (TC) identifier meaning												
TC1 Area obstacle occlusion must be less than 30%.												
TC2 The wind magnitude $ v^{\text{wind}} $ must be less than 10 m/s.												
TC3 No more than one HAPS can coexist in this type of area.												
TC4 A MA cannot be used as a corridor (i.e. if a HAPS flies into the MA, it must try to carry out the mission).												
TC5 There must be an existing corridor that connects this MA to the previous mission element.												
TC6 LOIs cannot be repeatedly monitored at each MA visit.												
Mission Requirements (MR) for successful monitoring												
MR1 All the LOIs of the MA are visually recorded.												
MR2 Ground image coverage of each LOI of the MA is higher than the minimum threshold imposed by the contracting client.												
MR3 Images captured within each MA are taken within its corresponding monitoring time windows.												
MR4 MA has not been visited that day more times than allowed by its maximum revisit frequency.												
MR5 Time-lapse between two consecutive visits to the same MA is higher than its allowed minimum inter-visit time.												
Rewards to be given for each MA (for the scenario in Fig. 2)												
MA	1	2	3	4	5	6	7	8	9	10	11	12
Reward ($\times 10^3$)	4	50	100	20	3	5	15	3	13	18	20	10

communicated to the Air Traffic Control (ATC) authority. HAPS monitoring missions include: Locations of Interest (LOIs) that are ground areas to be observed, Mission Areas (MAs) that group LOIs from the same client with the same set of requirements, Corridors (Cs) to be used by the HAPS to fly from one MA to another, and designated unlocked aerospace Waiting Areas (WAs) within the corridors for HAPS to linger. A typical mission scenario is depicted in Fig. 2, where LOIs, MAs, WAs and Cs are respectively represented in green, blue, yellow, and gray.

Besides, multi-HAPS missions can be defined by the set of sequences of n time-stamped actions $\pi_n^h = (< a_0^h, t_0^h >, < a_1^h, t_1^h >, \dots, < a_n^h, t_n^h >)$ to be performed by each HAPS h . The actions a_t^h , partially summarized in Table 1, are tasks related to the different elements of the mission, and their starting time t_t^h and duration $\delta_t^h = t_{t+1}^h - t_t^h$ are random variables due to the variability of the environmental wind. Additionally, each type of action can only be performed if the Task Constraints (TC) identified in the last column of the upper part of the table are fulfilled. Furthermore, after each MA is *successfully* monitored, the HAPS team is rewarded a

Figure 3: Temporal hierarchical plan example for two HAPS. Tasks by HAPS1 are edged in blue, and by HAPS2 in black

payment determined according to the interest of each company in their MAs (with the values stored at the bottom of Table 1 for the scenario represented in Fig. 2). And exactly, a successful MA monitoring only occurs when all the Mission Requirements (MR), presented in the third section of Table 1, are fulfilled.

Finally, it is worth noting that HAPS fly at high altitudes (e.g. ~20 km), carry out their mission during several hours (e.g. 8 hours during daylight) under time-varying weather conditions, and use electro-optical mission cameras to monitor ground activities. Hence, mission success is affected by the clouds situated between the operating altitude and the ground, and by the wind in the operation area¹. Thus, the reward of a plan cannot be known exactly during the pre-planning phase, and has to be estimated (considering the problem uncertainties) by the optimization approach of the planner.

3 TASK-MISSION PLANNER

On the one hand, this section outlines the Hierarchical Task Network (HTN) methodology that allows our planner to decompose the mission into tasks of the HAPS domain. On the other hand, it details the GA (coding, evaluation, steps and operators) used to optimize the ordering of the actions of the mission plan.

3.1 Hierarchical Task Network

In a realistic application, specifically in aeronautics involving strict regulations, mission execution must comply with protocols, which can be understood as domain-specific knowledge in planning. HTN is widely used to solve such class of problem (e.g. naming just a few in [4, 12, 13, 29]), since it decomposes the mission into primitive tasks according to domain knowledge (e.g. constraints, orders of tasks, effects of actions, etc.). Most HTN planners use a top-down forward decomposing strategy, which 1) decomposes each task a_0 chosen at the highest abstract level into subtasks until all tasks stemming from a_0 become primitive tasks and which 2) sequentially repeats the process until the plan horizon T_{end} is reached.

To illustrate how we perform this process for multi-HAPS monitoring operation², Fig. 3 displays part of the temporal HTN used to solve the scenario depicted in Fig. 2. The mission is first decomposed, at the highest mission level, into sequences of MAs to monitor and

WAs to cross. At the following phase, we show how to monitor each MA#: first the HAPS flies to the intersection of the corridor with the MA# ($to_{C\#}$) and subsequently crosses it ($cross_{C\#}$); afterwards, the HAPS visits each LOI# in MA# by reiterating the fly to LOI ($to_{LOI\#}$) and monitor LOI ($monitor_{LOI\#}$) actions. The bottom decomposition levels dictates when to turn on/off the mission camera, scan the LOIs and communicate with the ground control station, etc.

It is worth noting that although the scheme in Fig. 3 displays both the HTN and its temporal deployment, most HTN planners only decide the sequence of tasks to execute and leave out its temporal details [23]. An exception is the HTN planner in [9, 12], which includes the temporal element (allowing concurrency, and time-dependent goals and constraints) in order to plan forest fire fighting strategies. Yet, little has been done on optimizing the combinatorial problem of HTN planning. In this regard, [23] proposes three kinds of searches for a task planning problem with multiple decomposition methods that require the derivation of a reasonable heuristic, which is not always trivial (especially in problems with heterogeneous inter-dependent objectives and constraints). Alternatively, [9] takes randomly one of the decomposition methods to break down to non-primitive task, while the temporal HTN planner used in [15] for single HAPS monitoring solves the combinatorial problem by brute-force search over all decomposition methods (which is a viable strategy for scenarios involving just a few MAs). However, as the complexity of the combinatorial problem grows exponentially with the number of HAPS and MA, brute-force search can no longer be completed within reasonable time and its implementation as an anytime approach (that ends after a given time with the best plan so far) does not ensure the quality of the proposed solution. Hence, to overcome this issue, this paper details a new approach based on GA for optimizing the decomposition of the mission at the highest two levels in the HTN.

3.2 GA-Guided Task Decomposition

Among all the HTN levels in the multi-HAPS monitoring planner, the highest level (summarizable as MAs and WAs sequences) and the middle one (summarizable as LOI sequences within each MA) spawn the biggest combinatorial problem. Not only are the possible orderings numerous, but the rewards received by the HAPS are also highly dependent on the decomposition performed at these levels. It is hence advisable to optimize the decomposition at these two levels with an efficient GA capable of handling successfully complex constrained combinatorial problems. In the following, we describe the main properties of the selected approach.

3.2.1 Decision Variables. In a high-level plan $\pi_n^h = (< a_0^h, t_0^h > , \dots, < a_n^h, t_n^h >)$ the actions a_i^h are either MA to monitor (MA#) or WA to cross (WA#), while their starting times t_i^h are affected by the scenario characteristics (winds and distances). Besides, in the middle-level plan of each MA, the sequence of actions ($to_{LOI\#}$, $monitor_{LOI\#}$) can be summed up by the visiting ordering of its LOIs, while its timing is also scenario dependent. Hence, we decide to encode the high-level actions (MA# and WA#) and the middle-level LOI ordering within the decision variables (genes) of the GA, and leave their timings out, to be estimated by the GA when required (at the different steps of the GA).

¹In particular the former affects MR2, while the latter affects t_i^h and MR3.

²Further details can be found in [15], where a similar HTN for single HAPS planning is presented.

Figure 4: GA solution encoding.

In particular, the high-level decision variables for each plan are stored in the GA in an array (with an element for each HAPS) of lists of integers (each associated to a MA# and WA# action). Besides, for each MA# action in the aforementioned lists, the GA expands an array of middle-level decision variables to encode the visiting ordering (permutation) of its corresponding LOIs³. Additionally, it is worth noting that the number of elements n in each high-level plan is variable, and determined by the GA in order to let the high-level plan span between the mission start and end times $[T_{start}, T_{end}]$, which are pre-fixed, for instance, by the hours of daylight. Finally, as an example, Fig. 4 shows the GA encoding of a possible solution for monitoring the scenario in Fig. 2 with two HAPS, where the first one follows the sequence of MA and WA stored in the green list, and the second one follows the sequence stored in the blue list, and where the LOIs visited within each MA are monitored according to their LOI's ordering emerging from it.

Besides, to estimate the timing of the high-level plan, we consider that it is not unusual to have wind of magnitude $|v^{wind}| \sim 5$ m/s at the altitude where HAPS operate. Thus, we assume that a HAPS, flying with a true air speed v^{tas} , has a ground speed between $[v_{min}, v_{max}] = [v^{tas} - |v^{wind}|, v^{tas} + |v^{wind}|]$ and the duration δ_i^h of each action a_i^h is uniformly distributed over $[d_i^h/v_{max}, d_i^h/v_{min}] = [\delta_i^{h,min}, \delta_i^{h,max}]$, where d_i^h is the distance travelled⁴ by HAPS h while executing action a_i^h . Besides, action a_i^h terminates at $t_{i+1}^h = t_0^h + \sum_{j=0}^i \delta_j^h$, where t_0^h is deterministic ($t_0^h = T_{start}^h$) and $\sum_{j=0}^i \delta_j^h$ follows the distribution of the sum of $i+1$ non-identically distributed uniform random variables δ_j^h . Hence, when required, we can estimate the median value of t_{i+1}^h taking into account that the density $P(t_{i+1}^h - t_0^h) = P(\sum_{j=0}^i \delta_j^h)$ can be calculated with Eq. (1), derived in [8] and where $\mathcal{V}^i$ stands for the set with all 2^i vectors of signs $\vec{\epsilon}^k = (\epsilon_0^k, \dots, \epsilon_i^k) \in \{-1, 1\}^{i+1}$, $m_{\delta_j^h}$ is the median value of $[\delta_j^{h,min}, \delta_j^{h,max}]$, $u_{\delta_j^h} = (\delta_j^{h,max} - \delta_j^{h,min})/2$, $f(\vec{\epsilon}^k, \delta_{0:i}^h) =$

$\sum_{j=0}^i \delta_j^h + \sum_{j=0}^i (\epsilon_j u_{\delta_j^h} - m_{\delta_j^h})$, and $i!$ is the factorial of i .

$$P\left(\sum_{j=0}^i \delta_j^h\right) = \frac{\sum_{\vec{\epsilon}^k \in \mathcal{V}^i} (f(\vec{\epsilon}^k, \delta_{0:i}^h))^i \cdot \text{sign}(f(\vec{\epsilon}^k, \delta_{0:i}^h)) \prod_{j=0}^i \epsilon_j^k}{i! 2^{i+2} \prod_{j=0}^i u_{\delta_j^h}} \quad (1)$$

3.2.2 Constraints evaluation and handling. The constraints of the problem are classified, as Table 1 shows, in two groups: those associated to the task (TC) and those related with the successful monitoring of the missions (MR). The distinction has further implications: TC are hard-constraints that feasible plans are required to fulfill, while MR are soft-constraints whose violation implies that the HAPS team does not receive the reward associated to the monitoring of the corresponding MA. Hence, we treat them differently in the GA: the number of times that the former are violated by a multi-HAPS mission plan is accumulated in the constraint violation criterion (CC), while the violation of the latter modifies the value of the reward objective function.

3.2.3 Objective Criteria. The GA evaluates each mission plan with the three objective criteria explained below and combines them into a single objective function using a weighted sum.

Expected Rewards per Hour per HAPS Criterion. In order to take into account the effect of the weather conditions (wind and clouds) in the monitoring success of the MAs, we estimate the expected reward of the highest level plan. To do it, we define $\tilde{\pi}_{n,i}^h = (<a_i^h, t_i^h>, \dots, <a_n^h, t_n^h>)$ as the partial plan from the i -th action for HAPS h and calculate the expected cumulative probabilistic reward, obtained when applying at state s_i^h (corresponding to the last monitored MA or WA) and time t_i^h the remaining plan $\tilde{\pi}_{n,i}^h$ under weather w_{t_i} , with⁵ Eq. (2), where $L(\mu|s_i^h, t_i^h, a_i^h, w_{t_i})$ is the likelihood that action a_i^h , performed at time t_i^h at state s_i^h , is successful ($\mu = \text{succ}$) under weather conditions at t_i (w_{t_i}) or not ($\mu = \text{fail}$); $P_\mu(t_{i+1}^h)$ is the probability density function of ending a_i^h at t_{i+1}^h , and $R(\mu, a_i^h, t_i, t_{i+1})$ is the immediate reward obtained when performing a_i^h between times t_i and t_{i+1} successfully ($\mu = \text{succ}$) or unsuccessfully ($\mu = \text{fail}$).

$$E(\Sigma|s_i^h, t_i^h, \tilde{\pi}_{n,i}^h, w_{t_i}) = \sum_{\mu \in \{\text{succ}, \text{fail}\}} L(\mu|s_i^h, t_i^h, a_i^h, w_{t_i}) \int_{\mathbb{R}} P(t_{i+1}^h) \cdot [R(\mu, a_i^h, t_i^h, t_{i+1}^h) + E(\Sigma|s_{i+1}^h, t_{i+1}^h, \tilde{\pi}_{n,i+1}^h, w_{t_{i+1}})] \cdot dt_{i+1}^h \quad (2)$$

Hence, on the one hand, Eq. (2) combines the immediate reward obtained after monitoring the selected mission area successfully (using the corresponding values at the bottom of Table 1) or unsuccessfully (using a zero reward) with the expected reward of the remaining action plan $\tilde{\pi}_{n,i+1}^h$, and, on the other hand, it weights them with the probability of finishing the action at the given times and the likelihood of performing the actions successfully and unsuccessfully. Moreover, this expected cumulative reward can be computed using a backward iteration approach that takes advantage of the fact that the immediate reward $R(\mu, a_i^h, t_i^h, t_{i+1}^h)$ is piecewise constant with respect to t_{i+1} , and hence $E(\Sigma|s_i^h, t_i^h, \tilde{\pi}_{n,i}^h, w_{t_i})$ is piecewise constant

³We use integers for MA#/WA# since the same MA and WA can be visited several times by the same HAPS in a mission, while we use permutations for LOIs as a MA is not successfully monitored unless all its LOIs have been visually recorded.

⁴Note that the total distance travelled to monitor a mission area depends on the ordering used to visit its LOI. Hence, distances d_i^h are affected, indirectly, by the LOI visiting order encoded by the GA for each MA action of the high-level plan.

⁵Note that Eq. (2) has the same structure as the habitual formulation used to obtain the expected reward in time-dependent Markov decision processes [7].

Figure 5: Crossover selection point example, in a high-level plan expanded over its tasks median estimated times.

too and the integration in Eq. (2) can be done piecewise with the analogy that $P(t_{i+1}^h) = P(t_{i+1}^h - t_0^h) = P(\sum_{j=0}^i \delta_j^h)$. Note also that this way of proceeding puts more emphasis on the rewards of the initial actions than on the final ones, which is especially useful when planning for long time lapses, where there is time to re-plan for the future during the plan execution.

Finally, and to account for the rewards of all the HAPS in the team, the expected reward objective criterion OC_{rew} adds up the cumulative probabilistic rewards of all HAPS and normalizes the obtained value by the total number of hours of plan durations⁶.

$$OC_{rew} = \frac{\sum_{h=1}^{n_{HAPS}} E(\Sigma | s_0^h, t_0^h, \pi_{n,0}^h, w_{t_0})}{\sum_{h=1}^{n_{HAPS}} (T_{end}^h - T_{start}^h)} \quad (3)$$

Effort Criterion. To keep the clientele satisfied, we require the HAPS team to perform monitoring missions as much of their time in the air as possible. If $\delta_{LOI\#}$ is the estimated median duration spent at each LOI monitored during the plan, the objective function representing the HAPS effort to satisfy the clients can be calculated with Eq. (4).

$$OC_{eff} = \frac{\sum_l \delta_{LOI_l}}{\sum_{h=1}^{n_{HAPS}} (T_{end}^h - T_{start}^h)} \quad (4)$$

Diversity Criterion. The plan must also satisfy a big and diverse clientele pool. That is, even when one client pays a much higher reward, the HAPS should monitor the MAs of all the clients. To evaluate this diversity criterion OC_{div} , we use the Simpson index [28], which indicates the probability of obtaining two different MAs when two random draws of tasks are sampled without replacement from the high-level mission plan. In other words, the criterion obtains the probability of *not* drawing the same MA when two of them are drawn without replacement from a given plan. This criterion is calculated with Eq. (5), where n_{MA} is the number of MAs (or clients), n_c is the number of occurrence of MA_c in the task plan, and N is the total number of rewarded MAs within the task plan.

$$OC_{div} = 1 - \frac{\sum_{c=1}^{n_{MA}} n_c(n_c - 1)}{N(N - 1)} \quad (5)$$

⁶The denominator considers that each HAPS can have a different mission start and end time.

3.2.4 Genetic Algorithm Implementation. The GA uses binary tournament selection for choosing the pairs of parents, performs a problem-specific crossover (explained below) with a high mating probability P_{xover} , mutates the high-level genes of the children with a low mutation probability P_{mut} , and selects the solutions of the survival population among the ones in the new and old population using the stochastic ranking procedure [27] with a low swap probability P_{swap} . Finally, the algorithm stops after a given number of generations.

Besides, the first population is randomly initialized taking into account the connections of MAs and WAs supported by the scenario (in order to increment the number of feasible solutions), the estimated duration of each high-level action δ_i^h and the mission span (i.e. plan horizon) $[T_{start}, T_{end}]$ (in order to determine the number of elements required in each π_n^h). Unfeasible solutions according to the task constraints (TC) are eliminated and regenerated, until the initial population is only formed by feasible solutions.

The implemented crossover is slightly unusual, given the fact that each solution stores the highest-level sequence of actions (i.e. MA# and WA#) of several HAPS and the middle-level LOIs permutation expansions of each MA of the highest-level action sequences. Moreover, the duration of the highest-level actions (as well as the middle ones) are different and random. To deal with that variability in the action durations we perform a crossover inspired in the approach used in [2]. In particular, we select randomly and uniformly a value within $[T_{start}, T_{end}]$ as crossing-time $t_{crossover}$, and then we identify within the list of highest-level actions of each HAPS the one with the nearest *estimated median* starting time, as illustrated in Fig. 5 (for one of the possible high-level solutions when monitoring the scenario in Fig. 2 with h HAPS). Next we perform the usual single-point crossover in the high-level lists of all the HAPS, maintaining in the children the existing LOIs ordering associated to each MA in the parents, unless the crossing element is the same MA in both solutions. In this special case, we perform instead a permutation order crossover on the middle-level LOIs of both MA.

Finally, it is worth noting that the genetic operators (crossover and mutation) can generate unfeasible solutions (which do not fulfill the task constraints). However, unfeasible solutions have the opportunity of surviving a few generations, thanks to the stochastic ranking procedure [27] used to sort and select the best solutions within the old and new population, in order to help the algorithm to move from one feasible region to another through an unfeasible one.

4 TESTS AND ANALYSIS

In this section, we analyze the performance of the GA used to guide the tasks decomposition in our HTN planner. To put forth the benefit of our new approach, we compare the results obtained over several scenarios when using the GA and when using brute-force search. The tests are run on a 4.00 GHz x 8 Intel Core i7-6700K CPU with 32GB RAM, using a Matlab⁷ implementation of the GA.

4.1 Test Scenarios

The scenarios under test share the layout of MAs, WAs, LOIs and Cs depicted in Fig. 2, which comprises an area of 600 km × 600 km

⁷The integral operations required in Equation 2 are implemented in C.

Figure 6: Cloud coverage, monitoring time windows and substantial wind of each MA of the scenarios

placed in the South of Germany, and selected due the availability of historical weather data (useful to realistically simulate the environment and flight dynamics [18]). LOIs and MAs dimensions have orders of magnitude of tens and of hundreds of kilometers respectively. The minimum time lapse between two visits to the same MA is set to two hours. Besides, as fixed-wing aircrafts, HAPS can fly typically at an optimal equivalent airspeed of 9 m/s [21] and cruise at the operating altitude at a $v^{\text{tas}} \sim 29$ m/s. Its ground speed can be obtained $v^{\text{gs}} = v^{\text{tas}} + v^{\text{wind}}$. Equipped with an electro-optical mission camera, they must perform the monitoring tasks during daylight, starting at 08:00 in the morning and finishing at 16:00 in the afternoon. Hence, at 07:00, while the HAPS still linger at a WA (HAPS-1 in WA13 and HAPS-2 in WA14) awaiting commands, a plan for the next 8 hours must be determined (and approved).

Scenario 1 and 2 (S1 & S2) use the same weather forecast, taken from the National Meteorological Service [32] for the 24th April of 2018, while Scenario 3 (S3) uses synthetic weather data to test the performance and convergence speed of the GA-guided planner on cloudy days (where the cloud coverage is higher than 50% for almost 50% time of the day). Both forecasts are summarized in Fig. 6 with other configuration data for each MA and over the plan horizon (08:00-16:00). More in detail, the blue-grey shades represent the mean cloud coverage (the darker the shade is, the higher the cloud coverage), while the red dots indicate that the mean of the wind magnitude is greater than 3 m/s, the blue (dark or light) rectangular edges denote the time windows at which the LOIs of the respective MAs are requested by the clients to be monitored. Besides, the MA names in black encompass the LOIs that can only be visited once

Table 2: GA parameterization

Generations = 50	Stochastic ranking swapping
Population size = 100	$P_{\text{swap}} = 0.2$
Tournament size = 3	Objective addition weights:
Crossover $P_{\text{xover}} = 0.9$	$w_{\text{rew}} = 1, w_{\text{div}} = 1000,$
Mutation ⁹ $P_{\text{mut}} = 0.2$	$w_{\text{eff}} = 2000 (P1, P2) \text{ or } 5000 (P3, P4)$

during the day, while those in red encompass LOIs that can be visited as frequently as possible.

Finally, in S1 only one HAPS is contracted to carry out the mission, while in S2 and S3 a second HAPS is incorporated (making the combinatorial problem at the highest decomposition level exponentially more complex and activating the constraint that prohibits the coexistence of multiple HAPS in the same MA).

4.2 Planner Configurations

The following planner configurations are used to perform the tests.

The first configuration (P1) uses the GA described in Section 3.2 to guide the search for the optimal decomposition at the mission level of the temporal HTN planner. As the performance of the GA depends on the values of its parameters, we have to choose their values after performing a statistical analysis of their influence in the solutions obtained over different scenarios [6]. In particular, we choose the ones summarized in Table 2 as the overall best configuration, after running each parameterization 20 times and analyzing the evolution of the mean, over the test runs, of the objective values of the best solution at each GA iteration.

In the second planner configuration (P2), the decomposition at the highest level of the temporal HTN planner is performed by brute-force search. Not only is the purpose of this configuration to show the computational benefits of using a GA to optimize the task decomposition, but also to determine the optimal solution and check if P1 converges towards it.

The third and fourth planner configurations (P3 and P4) substitute the probabilistic cumulative reward (Eq. (2)) used in the reward objective criterion (Eq. (3)) by the deterministic reward⁸ presented in Eq. (6) and obtained with a deterministic ending time $t_i^{h, \text{det}}$ for each action. Hence, P1 and P2 consider the wind effects in the randomness of the ending times of the high level actions, while P3 and P4 use a deterministic ending time obtained ignoring the wind uncertainty. Finally, except for the difference in the reward objective (and in the w_{eff} weight), P3 is similar to P1 (i.e. both use the GA described in Section 3.2 to guide the decomposition at the highest mission levels), while P4 is similar to P2 (i.e. both use brute-force search). Hence, the solution by P4 serves as the reference solution for P3 (as the solution of P2 does it for P1).

$$E(\Sigma | s_0^h, t_0^h, \pi_{n,0}^h) = \sum_{a_i \in \pi_{n,0}^h} L(\mu = \text{succ} | s_i^h, t_i^{h, \text{det}}, a_i^h, w_{t_i}) \cdot R(\mu = \text{succ}, a_i^h, t_i^{h, \text{det}}, t_{i+1}^{h, \text{det}}) \quad (6)$$

⁸This new expression can be obtained assuming that $P(t_{i+1}^h) = \delta(t_{i+1}^h - t_{i+1}^{h, \text{det}})$ and that the rewards are only obtained when the MA is successfully monitored.

4.3 Results and Analysis

P1 and P3 benefit from the fact that each GA iteration takes about 1-2 second, and obtain their solution in less than 2 minutes after 50 iterations. The time used to obtain, by brute-force search, the optimal solution in P2 and P4 is scenario-dependent: in S1 it took 2.67 (P2) and 2.33 (P4) hours, while in S2 and S3 it required 16-18 hours¹⁰. Hence, from the computational point of view, the GA clearly accelerated the solution identification within the HTN planner.

The performance and the plans by the four planner configurations in the three tested scenarios are summarized in Fig. 7.

The first row shows, in a different column for each scenario, the evolution of the best fitness value obtained by each GA planner over the number of iterations for P1 (in blue) and P3 (in red), and the maximum fitness value found after the brute-force search by P2 (orange) and P4 (green). Besides, and due to the randomness of the GA algorithm, a total of 20 tests were run for P1 and P3, in order to represent the mean (solid line) and standard deviation (shaded area) of the weighted objective combination of the best solution at each GA iteration. The graphics show that, as expected, the search for the optimal plans using GA converges much faster (in 20 iterations) for the single HAPS scenario (S1) than when multiple HAPS are involved (S2 and S3). Besides, the fitness values of P1 and P2 are lower than those of P3 and P4 mainly due to the fact that the reward of each task in the plans of P1 and P2 only contributes probabilistically to the fitness value (as described in Eq. (2)). Finally, although the GA planners are not always able to identify the best solution obtained by brute-force search, they are capable of obtaining a feasible overall-good solution in less than 2 minutes.

The second row shows the plans (only at the highest mission level, with 1-12 being the MAs and 13-16 being the WAs) found by each planner. The actions ordering are displayed taking into account the median time of the actions predicted by the temporal HTN planner for P1 and P2, and the deterministic timing (without wind) for P3 and P4. The labels on the y-axis 'P#-\$' identify the plans found using planner configuration P# for HAPS-\$. The plans shown for planner P2 and P4 are simply the optimal ones obtained by brute-force according to their corresponding objective functions. For P1 and P3, we show the best plan found most often¹¹ out of the 20 test runs (unless several ones are found with the same frequency, in that case, we show the one with the highest fitness value).

Finally, to test the validity and quality of the plans, they are executed in a 6-dimensional HAPS simulator built according to [21], which is coupled with a Flight Management System (FMS) with an integrated numeric planner that obtains feasible flight trajectories while minimizing travel time [16], and a 4-dimensional flight controller [22]. The weather information used in the simulation is

identical to the information used in the task planner, to isolate the test of task plan quality from the weather forecast quality (which is out of the scope of this work). The third row of Fig. 7 shows the timing of the plans during the simulations, and highlights in green the MAs that are successfully monitored, in red those that fail to be rewarded, and in white those that correspond to WAs. The results for S3 show the benefits of using a probabilistic arrival time (within Eq. (2)) in P1 and P2, versus using a deterministic arrival time (within Eq. (6)) in P3 and P4: as the plans of HAPS-1 by P3 and P4 (P3-1 and P4-1) ignore the existence of the wind, they cannot successfully monitor their initial MA8 (within its allowed time window) during the simulations. It is also worth noting that although not all MAs of the plans by P1 and P2 are successfully monitored, most failures occur later in the plans. This effect is expected in the plans by P1 and P2, which use Eq. (2) to consider the distribution of each MA arrival time (spread wider in the far future MAs of the plan). In other words, Eq. (2) ensures that, the further a MA is in the mission time, the less important its contribution to the probabilistic reward is, because the reliability of a near future task plan is more important than a far future one. This is a good planning strategy for HAPS operations, since re-planning is often required when new weather forecast update is available (e.g. typically hourly).

CONCLUSION

Temporal HTNs are convenient for task mission planning in missions subject to heterogeneous time-dependent constraints stemming from stipulated operation protocols, as well as physical and environmental restrictions. Hence, we use them for HAPS monitoring/mapping task planning, since their activities are required/allowed at selected time windows and their success rates depends on the time-varying cloud-coverage and wind. However, the search for the optimal task decomposition in the HTN is hard since 1) the solution space is huge (due to the number of HAPS, MAs and LOIs to monitor) and complex (due to the time-dependency between the tasks). Since defining a reasonable task decomposition heuristic is not trivial either, we have incorporated a GA to our planner. Their benefits are multiple: it is easy to encode the solutions of the highest HTN levels (as list of MA and WA actions, and permutations of LOIs); it is straightforward to incorporate, evaluate and handle as many constraints and objective criteria as needed; the GA parameters can be tuned to make it converge fast to overall-good task-plans; and if needed, for online re-planning, the GA can terminate quicker and return the best 'promising' decomposition identified so far. It is also worth noting that our planner uses a probabilistic objective function (that considers the randomness caused by the weather in the plan execution and success) that allows it to handle tough-weather environments. Finally, we believe that the GA-guided task decomposition used by our planner for multi-HAPS monitoring can also be applied to other HTN-based planners with a combinatorial problem in its task decomposition.

In the future, a method to determine the minimal number of HAPS (fixed in the current version of the planner) required to fulfill the mission will be developed. Our future efforts will also focus on speeding up the GA (using parallelization techniques) to let it handle, in a reasonable time, missions involving more elements (HAPS, MAs, LOIs, WAs and Cs); and on analysing the benefits of

⁹It is worth noting that although the value of P_{mut} could seem a little bit high, high-level plans often have around 5-8 elements (MA and WA#) within the sequence of each HAPS, and hence the selected value is roughly the inverse of the number of elements in the high-level plan for the single-HAPS scenario (and twice that value for the two-HAPS scenario). Besides, the statistical test performed to select the parameters of the GA showed that it was a good choice for the scenarios under tests.

¹⁰The brute-force search ran out of memory when 2 HAPS were involved, and, hence, read/write operations on the SSD harddisk were required.

¹¹This does not imply that the represented plan is necessarily found many times, but only that it is found more often than the best plans identified by the other runs of GA.

Figure 7: Performance test and plan quality analysis run for different task planner configurations in three different scenarios

treating the objective functions in a pareto-front fashion and of mutating the blue values of the middle-level (LOIs) genes.

REFERENCES

- [1] Airbus. 2018. Zephyr press release. <https://www.airbus.com/defence/uav/zephyr.html>. (2018).
- [2] B. Andrés-Toro, J. M. Girón-Sierra, P. Fernández-Blanco, J. A. López-Orozco, and E. Besada-Portas. 2004. Multiobjective optimization and multivariable control of the beer fermentation process with the use of evolutionary algorithms. *Journal of Zhejiang University-SCIENCE* 5 (2004), 378–389.
- [3] Aurora. 2018. Odyssey press release. <https://www.aurora.aero/odyssey-high-altitude-pseudo-satellite-haps/>. (2018).
- [4] J. Benton, D. Smith, J. Kaneshige, L. Keely, and T. Stucky. 2018. CHAP-E: A Plan Execution Assistant for Pilots. In *Int. Conf. on Automated Planning and Scheduling*.
- [5] E. Besada-Portas, L. de la Torre, J.M. de la Cruz, and B. Andres-Toro. 2010. Evolutionary Trajectory Planner for Multiple UAVs in Realistic Scenarios. *IEEE Transactions on Robotics* 26 (2010), 619–634.
- [6] E. Besada-Portas, L. de la Torre, A. Moreno, and J.L. Risco-Martin. 2013. On the performance comparison of multi-objective evolutionary UAV path planners. *Information Sciences* 238 (2013), 111–125.
- [7] J.A. Boyan and M.L. Littman. 2001. Exact Solutions to Time-Dependent MDPs. *Advances in Neural Information Processing Systems* 13 (2001), 1026–1032.
- [8] D.M. Bradley and R.C. Gupta. 2004. On the Distribution of the Sum of n Non-Identically Distributed Uniform Random Variables. *Annals of the Institute of*

- Statistical Mathematics* 54 (2004), 689–700.
- [9] L. Castillo, J. Fdrex-Olivares, O. García-Pérez, and F. Palao. 2006. Efficiently Handling Temporal Knowledge in an HTN Planner. In *Int. Conf. on Automated Planning and Scheduling*.
 - [10] Y. Eun and H. Bang. 2009. Cooperative Task Assignment/Path Planning of Multiple Unmanned Aerial Vehicles Using Genetic Algorithm. *Journal of Aircraft* 46 (2009), 338–343.
 - [11] J. Everaerts and N. Lewyckyj. 2011. Obtaining a Permit-To-Fly for a HALE-UAV in Belgium. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*.
 - [12] J. Fdrex-Olivares, L. Castillo, O. García-Pérez, and F. Palao. 2006. Bringing Users and Planning Technology Together. Experiences in SIADEX. In *Int. Conf. on Automated Planning and Scheduling*.
 - [13] I. Georgievski and M. Aiello. 2015. HTN Planning: Overview, Comparison, and Beyond. *Artificial Intelligence* 222 (2015), 124–156.
 - [14] J.J. Kiam, V. Hehtke, E. Besada-Portas, and A. Schulte. 2019. Hierarchical Planning Guided by Genetic Algorithms for Multiple HAPS in a Time-Varying Environment. In *2nd Int. Conf. on Intelligent Human Systems Integration*.
 - [15] J.J. Kiam and A. Schulte. 2018. Multilateral Mission Planning in a Vector Field with Dynamic Constraints. In *IEEE Int. Conf. on Systems, Man, and Cybernetics*.
 - [16] J. J. Kiam, E. Scala, M. Ramirez, and A. Schulte. 2018. Using a Hybrid AI-Planner to Plan Feasible Flight Paths for HAPS-Like UAVs. In *Workshop on Planning and Robotics (PlanRob) at ICAPS*.
 - [17] J. J. Kiam and A. Schulte. 2017. Multilateral Quality Mission Planning for Solar-Powered Long-Endurance UAV. In *IEEE Aerospace Conference*.
 - [18] M. Köhler, F. Funk, T. Gerz, F. Mothes, and E. Stenzel. 2017. Comprehensive Weather Situation Map Based on XML-Format as Decision Support for UAVs. *Journal of Unmanned System Technology* 5 (2017), 13–23.
 - [19] M. Köhler, T. Gerz, and A. Tafferner. 2016. Cb-like-Cumulonimbus Likelihood: Thunderstorm Forecasting with Fuzzy Logic. *Meteorologische Zeitschrift* 25 (2016), 1–19.
 - [20] O. Lima, R. Ventura, and I. Awaad. 2018. Integrating Classical Planning and Real Robots in Industrial and Service Robotics Domains. In *Proceedings of the 6th Workshop on Planning and Robotics (PlanRob) at ICAPS*.
 - [21] R. Müller, J. J. Kiam, and F. Mothes. 2018. Multiphysical Simulation of a Semi-Autonomous Solar Powered High Altitude Pseudo-Satellite. In *IEEE Aerospace Conf.*
 - [22] R. Müller and G. Looye. 2013. A Constrained Inverse Modeling Approach for Trajectory Optimization. In *AIAA Guidance Navigation and Control Conf.*
 - [23] D. Nau, T.C. Au, O. Ilghami, U. Kuter, J.W. Murdock, D. Wu, and F. Yaman. 2003. SHOP2: An HTN Planning System. *Journal of Artificial Intelligence Research* 20 (2003), 379–404.
 - [24] S. Perez-Carabaza, E. Besada-Portas, J.A. Lopez-Orozco, and J.M. de la Cruz. 2016. A Real World Multi-UAV Evolutionary Planner for Minimum Time Target Detection. In *Proc. of the 18th Conf. on Genetic and Evolutionary Computation*.
 - [25] S. Perez-Carabaza, E. Besada-Portas, J.A. López-Orozco, and G. Pajares. 2019. Minimum time search in real-world scenarios using multiple UAVs with onboard orientable cameras. *Journal of Sensors* (2019).
 - [26] C. Ramirez-Atienza, J. Ser, and D. Camacho. 2019. Weighted strategies to guide a multi-objective evolutionary algorithm for multi-UAV mission planning. *Swarm and Evolutionary Computation* 44 (2019), 480–495.
 - [27] T. P. Runarsson and X. Yao. 2000. Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. on Evolutionary Computation* 4, 3 (2000), 284–294.
 - [28] E. H. Simpson. 1949. Measurement of diversity. *Nature* 163 (1949), 688.
 - [29] E. Sirin, B. Parsia, D. Wu, J. Hendler, and D. Nau. 2004. HTN planning for Web Service composition using SHOP2. *Journal of Web Semantics* 1 (2004), 377–396.
 - [30] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel. 2014. Combined task and motion planning through an extensible planner-independent interface layer. In *IEEE Int. Conf. in Robotics and Automation*.
 - [31] Y. Wang, T. Kirubarajan, R. Tharmarasa, R. Jassemi-Zargani, and N. Kashyap. 2018. Multiperiod Coverage Path Planning and Scheduling for Airborne Surveillance. *IEEE Trans. Aerospace Electron. Systems* 54 (2018), 2257–2273.
 - [32] Weather Data. 2018. German National Weather Service. <https://www.dwd.de/EN>. (2018).