



Article

Fish-Inspired Task Allocation Algorithm for Multiple Unmanned Aerial Vehicles in Search and Rescue Missions

Amjaad Alhaqbani ^{1,2}, Heba Kurdi ^{1,3,*} and Kamal Youcef-Toumi ³¹ Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia; Aalhaqbani@kacst.edu.sa² Space and Aeronautics Research Institute, King Abdulaziz City for Science and Technology (KACST), P.O. Box 6086, Riyadh 11451, Saudi Arabia³ Mechanical Engineering Department, Massachusetts Institute of Technology (MIT), Cambridge, MA 02139, USA; youcef@mit.edu

* Correspondence: hkurdi@ksu.edu.sa

Abstract: The challenge concerning the optimal allocation of tasks across multiple unmanned aerial vehicles (multi-UAVs) has significantly spurred research interest due to its contribution to the success of various fleet missions. This challenge becomes more complex in time-constrained missions, particularly if they are conducted in hostile environments, such as search and rescue (SAR) missions. In this study, a novel fish-inspired algorithm for multi-UAV missions (FIAM) for task allocation is proposed, which was inspired by the adaptive schooling and foraging behaviors of fish. FIAM shows that UAVs in an SAR mission can be similarly programmed to aggregate in groups to swiftly survey disaster areas and rescue-discovered survivors. FIAM’s performance was compared with three long-standing multi-UAV task allocation (MUTA) paradigms, namely, opportunistic task allocation scheme (OTA), auction-based scheme, and ant-colony optimization (ACO). Furthermore, the proposed algorithm was also compared with the recently proposed locust-inspired algorithm for MUTA problem (LIAM). The experimental results demonstrated FIAM’s abilities to maintain a steady running time and a decreasing mean rescue time with a substantially increasing percentage of rescued survivors. For instance, FIAM successfully rescued 100% of the survivors with merely 16 UAVs, for scenarios of no more than eight survivors, whereas LIAM, Auction, ACO and OTA rescued a maximum of 75%, 50%, 35% and 35%, respectively, for the same scenarios. This superiority of FIAM performance was maintained under a different fleet size and number of survivors, demonstrating the approach’s flexibility and scalability.



Citation: Alhaqbani, A.; Kurdi, H.; Youcef-Toumi, K. Fish-Inspired Task Allocation Algorithm for Multiple Unmanned Aerial Vehicles in Search and Rescue Missions. *Remote Sens.* **2021**, *13*, 27. <https://dx.doi.org/10.3390/rs13010027>

Received: 25 November 2020

Accepted: 21 December 2020

Published: 23 December 2020

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Unmanned aerial vehicles (UAVs), originally developed for the military, have become preeminent platforms in civil applications, such as agriculture [1–3], transportation [4,5], mineral exploration [6–8], and search and rescue (SAR) operations [9–11]. SAR missions are generally time critical, where any delay can result in potential human losses, and such missions often occur in unfriendly environments that are likely to be detrimental to manned rescues [12]. Therefore, an increasing recognition of the potential of UAVs for SAR missions is evident in the literature (e.g., [13–19]). In fact, the usage of UAVs in such operations has already been demonstrated in the events of 9/11, after the collapse of the World Trade Center in New York [20]. Rescue teams also deployed UAVs in the aftermath of the 2008 Sichuan earthquake to identify choke points that could hinder the rescue efforts [21].

In a typical SAR mission, multiple UAVs are sent to a disaster area to locate and rescue survivors. The time and operational constraints of SAR missions and the limitations imposed on UAVs require efficient coordination to best allocate search and rescue tasks

to the UAVs within the same mission, which is known as the multi-UAV task allocation (MUTA) problem. MUTA is a special case of the NP-hard multi-robot task allocation (MRTA) problem [22–24]. In MUTA, robots fly, thereby introducing additional challenges in communications and increasing the level of uncertainty. Furthermore, MUTA constraints are more critical in SAR missions due to its life-and-death nature and absence of communication infrastructure. Due to the high complexity of the MUTA problem, it is often approached by using heuristics to ensure traceable solutions.

This study introduces a fish-inspired algorithm for the task allocation for multi-UAVs (FIAM) during an SAR mission, which exploits ideas from fish schooling and foraging behavior where fish aggregate in schools to search for food, and each school follows a single leader. The follower fish progressively abandons a futile leader, as forgetfulness significantly plays a role in this mechanism as fish have high forgetting rates of ‘bad memories’, and joins a more successful leader forming a massive school around them due to the lower forgetting rate of ‘good memories’. Fish do this with minimum communications among them [25]. Similarly, FIAM groups’ UAVs in small fleets with a leader of each to explore different regions of a disaster area. Leaders at regions with few survivors would need less followers and vice versa. The decisions of whether to lead or follow, whom to follow, when and for how long are very critical and are handled, in FIAM, autonomously, adaptively, and with minimum direct communications between the fleet members of SAR mission in hostile environments.

The primary assets of FIAM, compared with other MUTA approaches, are adaptive behaviors and autonomy of decision by each UAV based on situation awareness. Additionally, although fish behavior has previously been considered for various task assignment problems (e.g., [22–24]), it has not been particularly considered for resolving the MUTA problem. Above all, these benefits come at no cost of running time where FIAM was investigated in-depth by simulating SAR missions within a strictly controlled evaluation framework. FIAM performance was benchmarked against three long-standing MUTA heuristics, including opportunistic task allocation (OTA) [18], auction algorithm [26], and Ant-colony optimization (ACO) [27], as well as the recently introduced locust-inspired heuristic for MUTA (LIAM) [18]. The results demonstrated the superiority of FIAM over the benchmark algorithms, where it maintained a significantly higher percentage of rescued survivors and lower task completion time and running time. The main contributions of this study are as follows:

- A new algorithm inspired by fish schooling and foraging behavior, to address the special challenges of the task allocation problem, which is among the NP-hard problems.
- The customization of the proposed algorithm to address the extra challenging constraints of multi-UAV SAR missions.
- A strictly controlled experimental framework to thoroughly investigate the performance of the proposed algorithm at different problem scales.

The remainder of this study is organized as follows. In Section 2, the existing studies developed to address the problem of multi-robot and multi-UAV task allocation are reviewed. Section 3 describes the biological inspiration and the algorithm design. Sections 4 and 5 present the evaluation methodology and experimental results, respectively. Finally, Section 6 concludes the study and briefly discusses future research directions.

2. Related Work

Due to the wide range of applications of the task allocation problem, several approaches have been introduced in the literature to tackle its intrinsic challenges. Starting from the simple opportunistic allocation to advanced auction-based and bio-inspired approaches. The opportunistic allocation approach (OTA) follows a simple greedy strategy [18] where each task is allocated to the first available agent regardless of whether it is the best fit for this task in terms of the available resources and distance to reach. Despite its appealing simplicity, OTA is not widely adopted because it produces less optimum solutions. In contrast, auction-based algorithms allocate tasks to agents based on bidding [28]. Agents bid

for each task and the task is allocated to the highest bidder [29]. Bids are usually based on current available resources to the UAV and its distance from the task. Due to the efficiency of the auction-based approach, many of its variants have been proposed [26,30]. However, apparently, this approach becomes time and resource consuming for a large number of tasks and fleets of UAVs, where a quadratic time complexity $O(n^2m)$ (where n is the number of tasks, and m is the number of UAVs) was reported [31].

The bio-inspired algorithms employ ideas from biological systems in deciding task allocation. These algorithms considerably diverge depending on the species (e.g., ants, bees or locusts) that they simulate. Here, we focus on ants because ants have attracted widespread attention as a successful metaheuristic in many application domains, fish due to their direct relevance to this study, and locusts due to the latest interest in utilizing their behavior in the MUTA problem. AO is among the first and most successful proposals for bio-inspired metaheuristics. ACO is a probabilistic technique that is modeled based on the behavior of ants in colonies as they forage for food. The ants leave a trail of chemical pheromones to guide other ants to discovered food sources; paths with strong concentrations of pheromones are given priority to support the search for the shortest path between the colony's nest and the food source [32]. ACO has been used in approaching different optimization problems [33–35], including the task allocation problem. In [36], ACO was utilized for MRTA to allocate tasks to robots and determine each robot's task-processing sequence. The collection path of ACO (CPACO) [27], extends ACO by establishing a three-dimensional pheromone path to resolve the MUTA problem. The dynamic ant-colony labor division (DACL) algorithm [37] was proposed for the task allocation problem in battlefields with a swarm of multiple combat UAVs. However, ACO and its variations poorly perform in dynamic contexts, such as SAR missions, which is a common problem in all the metaheuristics [38], as recalculations are needed once a new task arrives, i.e., a new survivor discovered, which results in large computation overheads due to the exponential run time [39].

Several studies have used fish as an inspiration in approaching problems in complex optimization domains [40], such as networks [41–45], image processing [46–48], robotics, motion control [49–51], machine learning [52–54], path planning [55,56], industries [57], automation [58], and many other fields. However, few fish-inspired heuristics have been proposed, particularly targeting the task allocation problem. The artificial fish-swarm algorithm (AFSA) is among the earliest fish-inspired metaheuristics [59]. It is a swarm intelligence algorithm that derives its inspiration from certain behaviors of individual fish during the foraging process in nature. Each artificial fish in the algorithm searches for food by swarming, random moving, preying and following behaviors. An artificial fish has its own information, which comprises its current position, step length, the visual range, crowding factor, food concentration to decide which behavior to perform next. AFSA has several advantages, such as global search ability, fast convergence, parallelism and insensitivity to initial values. The original AFSA was modified for the distributed MRTA problem to minimize the digression between robots and desirability [60] by a resource-leveling method in which robots are considered resources. The algorithm performance was examined against particle swarm optimization and genetic algorithms where the results revealed its superiority in achieving faster global convergence and better robustness. In [61], a fish-swarm algorithm was proposed to schedule and allocate tasks to multiple robots. This approach consists of two main steps. In the first step, the optimal task execution sequence is calculated using AFSA. In the second step, the task sequence is randomly divided and allocated to the robots. The algorithm was compared with simulated annealing and the results showed that the fish-swarm algorithm consumed less time. However, similar to other metaheuristics, these fish-inspired algorithms depend on iterative calculations for the best solutions, making them prohibitively time consuming to be employed in SAR dynamic environments.

In contrast to the above reviewed metaheuristics, a locust-inspires problem-dependent heuristic (LIAM) [18], was recently introduced, which is tailored to the MUTA problem in multi-UAV SAR missions based on the locust elastic behavior [9,18]. Locusts exhibit

adaptable morphological and behavioral forms as they advance through their lifecycle. Their behavior is dramatically altered in response to internal and external stimuli between two main phases: solitarious and gregarious. Similarly, the LIAM divides the SAR mission into two phases, search (solitarious) and rescue (gregarious) phases. Additionally, it dynamically alters the behavior of the UAVs, based on their physical capabilities to mimic the locust behavior. The experimental results showed the superiority of the LIAM compared to auction, ACO, and OTA algorithms in terms of the percentage of rescued survivors and reduced task completion time.

Considering the aforementioned works, there have been significant interests in the task allocation problem. However, among the various forms of the problem, the MUTA, especially in dynamic and constrained environments, such as SAR missions, is understudied. Furthermore, the potential benefit of fish-inspired algorithms to approach this problem is still unexplored.

3. FIAM Algorithm

3.1. Biological Inspiration

Fish are perhaps the most successful vertebrates since ancient times; their enormous diversity results in an astounding array of physiological and behavioral adaptations that ensure their survival in nature [25]. Hence, it is unsurprising that they have attracted increased attention for designing robust and adaptive systems (e.g., [62–64]). Fish of the same species move in groups commonly known as schools. There are several benefits for being in a school, such as reduced aggression, defense against predators, and better food localization during foraging [65,66]. However, while many aspects of the behavior of fish schools have been used as the basis for inspiration in algorithms, a remarkable aspect that has been consistently overlooked in computational applications is the forgetting process of fish. Studies have shown that forgetting plays a significant role in fish activities, such as the food foraging process, which allows rapid and flexible adaptation to changes in the environment. They discovered that in fish foraging, forgetting might be equally significant to remembering. Fish widely vary in their memory window size, which is highly dynamic and influenced by the environmental variability [25].

Fish usually forage for food through individual sampling and following other foragers. Foraging divides a school of fish dynamically into a single sampler and its followers. The sampler fish actively searches for food sources and pursue prey, while followers simply go along and profit from the food the sampler finds. Followers tend to forget, and therefore leave an unsuccessful sampler and also tend to remember, and therefore stick with, a successful one. Every time the sampler leads its follower to a food source, they assess its profitability (i.e., prey size) and the school size is accordingly adjusted using the forgetting process exhibited by fish. If the food is plentiful, a follower's forgetting rate of the sampler is slower (due to the large memory window size), and followers will be strongly attached to their sampler, keeping a small distance from it and its follower, resulting in a coherent school. Conversely, if the sampler consistently leads the school to meager sources, the forgetting rate of the followers is faster (due to the small memory window size). In other words, fish forget “bad” memories faster, as their memory window size shrinks, than they do “good” ones, by expanding their memory window size. However, different fish in a school will have different forgetting rates of the same sampler. This leads to a progressive rather than sudden disintegration of less successful schools. When followers forget the sampler they were with, they make the decision of either random search for their own food sources (becoming a new sampler), or of joining a more profitable school. This means that a successful school gradually expands as followers of less successful school join and their forgetting rate slows due to abundant food. Therefore, small schools progressively combine with more successful schools to form a massive harmonious school [25,67,68], as shown in Figure 1.

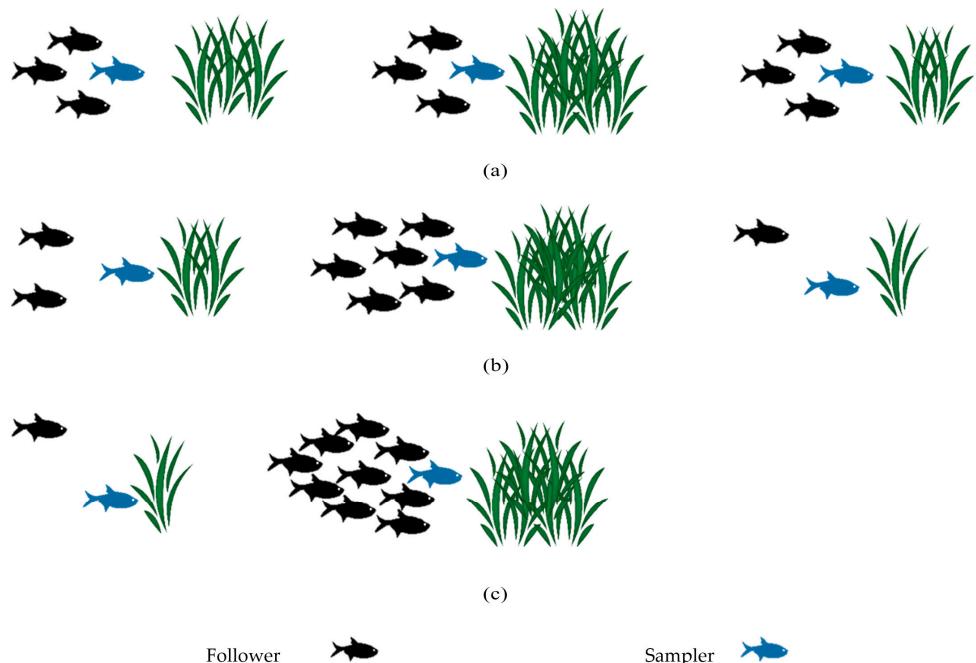


Figure 1. Fish-schooling behavior. (a) Three fish schools with variable food source profitability. (b) Followers from less profitable schools (left and right) join the most profitable school (middle). (c) Schools gradually combine into a large school.

From the above, one can build a mapping between the process of allocating UAVs to survivors, in an SAR mission, and the process of allocating fish to food sources, during the foraging of fish schools. Fish correspond to UAVs and food sources to survivors and the sea arena where fish live corresponds to a virtual representation of the disaster area internally maintained by each UAV. Additionally, the UAVs in FIAM are divided between two dynamically changing roles: Sampler UAVs and Follower UAVs, that behaviorally mimic sampler fish and follower fish, respectively.

3.2. System Overview

Similar to [18,69], we consider an SAR mission where the number, location, and lifetime of survivors are unknown *a priori*. Survivors are arbitrarily dispersed in the mission area with hotspots that indicate areas of higher survivor concentration to illustrate the fact that in real-life scenarios, people tend to gather in groups when they feel unsafe. A survivor can be in one of the four states: discovered, undiscovered, deceased, or rescued. At the start of the mission, the statuses of all survivors are unknown. This accordingly changes as the mission time elapses.

As illustrated in Figure 2, the mission area is logically structured as a grid of blocks with each block holding at most a single survivor. The grid is locally represented on each UAV, as an in-memory table which is periodically synchronized with updates from the other UAVs in the same mission. A set of $a \times a$ blocks constitute a single region where a is a positive integer > 1 . The value of a is determined during the initialization phase according to the size of the disaster area. Regions and blocks can be in one of the following states: unassigned, assigned or explored. An assigned block is a block that has been selected by a UAV but where the search for a survivor is yet to be completed. An assigned region is a region with at least one assigned block. A block is said to be explored when a search for a survivor has been completed. When all blocks in a region have been explored, then the region is said to be explored as well. At the start of the mission, the statuses of the regions and blocks are set to unassigned and change as the mission progresses.

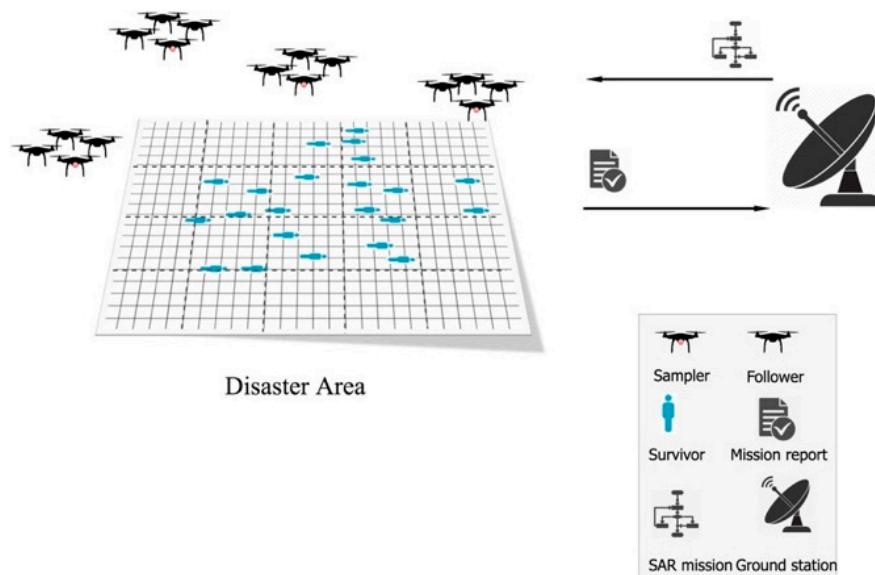


Figure 2. System overview. SAR, search and rescue mission.

The UAVs are divided into groups (schools). Each school is composed of multiple follower UAVs and a single sampler UAV. At the start of the system, the follower and sampler roles are randomly assigned to the UAVs in the fleet. Nevertheless, as the SAR mission progresses, the UAVs can dynamically switch between the two roles based on the sampler and follower UAV algorithms presented in Sections 3.3 and 3.4, respectively. We assume that the UAVs are battery-operated multi-rotors with similar capabilities to support role interchangeability and can autonomously fly without colliding. Before the beginning of any mission, the ground station determines the mission requirements, which include the parameter values needed to execute the mission, such as the mission duration, disaster area size and location, number of UAVs, their initial roles, battery capacity and maximum travel speed, number of charging stations and their locations. The ground station receives a complete mission report once the mission is completed or terminated.

3.3. Sampler UAV Algorithm

The sampler algorithm is illustrated in Figure 3. The sampler UAV starts by flying to the nearest unassigned block in the nearest unassigned region to search for survivors accordingly, the region and block states are changed from unassigned to assigned. Once a survivor is discovered, the survivor is rescued, the profitability of the region is increased by one unit and the block state is changed to be explored. The sampler UAV sequentially continues its search of unassigned blocks in the assigned region, once all blocks are explored in the assigned region, the sampler UAV starts searching for survivors in unassigned blocks in the nearest unassigned region. If all regions have already been assigned, the sampler UAV changes its role to follower UAV and joins the nearest sampler of the highest profitable region. The algorithm terminates when the UAV changes its role to follower UAV, and all blocks in all regions are assigned or when the mission time concludes.

3.4. Follower UAV Algorithm

The follower UAV algorithm explicitly adopts the forgetting strategy of fish behavior; hence, a UAV gradually abandons its sampler if its region is low in number of discovered survivors. This behavior is supported by the introduction of two variables: memory and forgetting rate. The memory variable is used in controlling the memorability of a sampler by its followers given the profitability of its region while the forgetting rate controls the memory size of followers.

A follower UAV starts with 100% memory capacity and a randomly generated forgetting rate, in the range of 10–20% (Figure 4). To search for survivors, the follower UAV flies to the nearest unassigned block in the direction of its assigned sampler. Upon the discovery of a survivor, the survivor is rescued, and the follower UAV increases the profitability of the region, which results in a deduction from the forgetting rate. In contrast, if the block does not hold a survivor, the follower memory value is decreased by an amount equal to the forgetting rate at the same time as the forgetting rate is increased. After each memory deduction, the minimum memory threshold is to be checked, if not reached, and there are more blocks in the current region the follower UAV continues searching by selecting the nearest unassigned block toward the sampler. However, if the minimum memory threshold has already been reached, the follower UAV will decide to either abandon its sampler or join a more successful sampler, if all regions are already assigned, or to change its role to a sampler and starts foraging on an unassigned region. This strategy ensures the discoverability of all survivors throughout the unassigned regions and blocks. The follower UAV algorithm terminates when the follower changes its role to a sampler, and all blocks in all regions have already been assigned or explored, or when the mission time concludes.

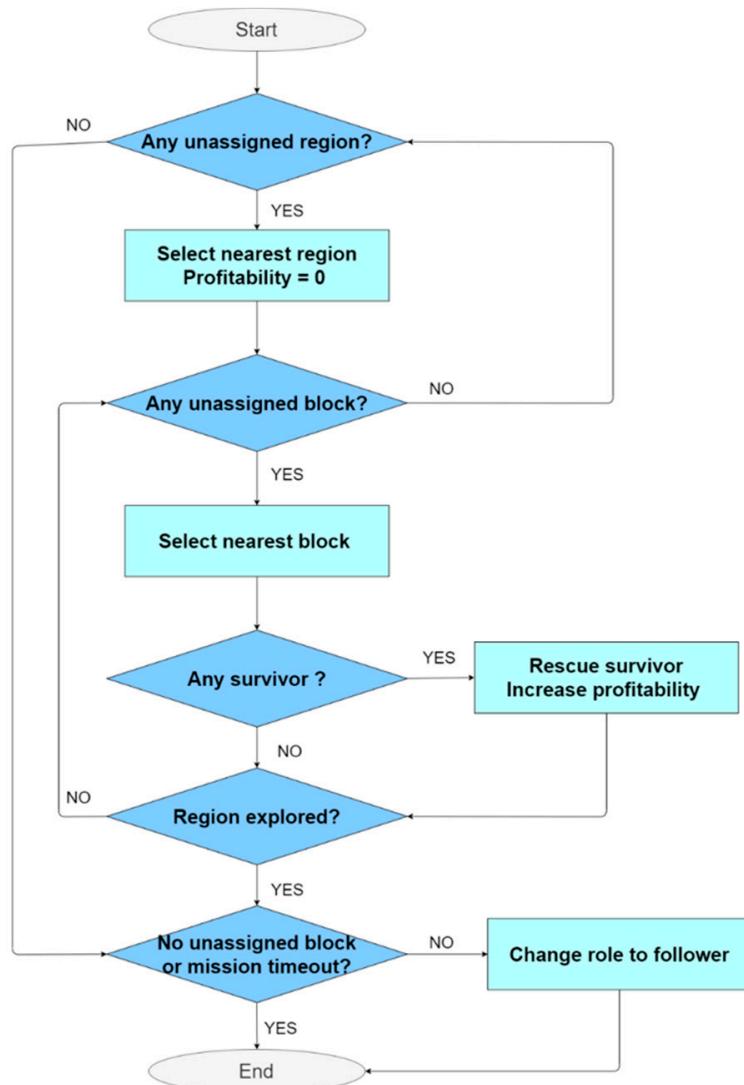


Figure 3. Sampler algorithm.

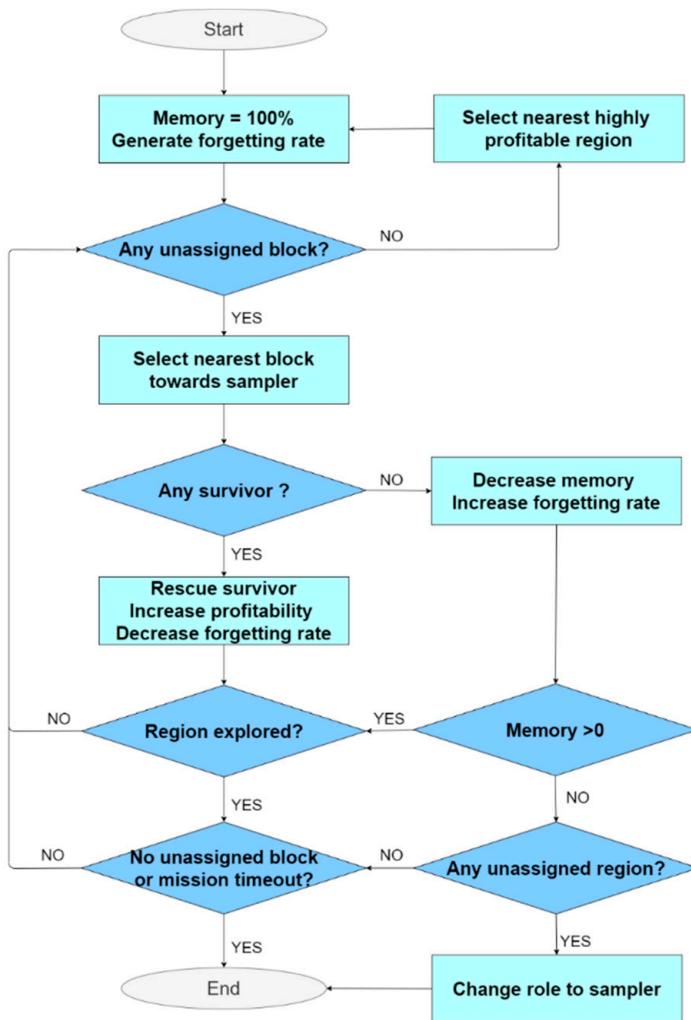


Figure 4. Follower algorithm.

4. Evaluation Methodology

The main objective of FIAM is to maximize the number of rescued survivors and minimize the mean rescue time at various problem scales in terms of the number of UAVs and number of survivors. A well controlled empirical evaluation framework was developed to evaluate the system performance by comparing it to three well established benchmark algorithms and a recently proposed locust-inspired MUTA heuristic: OTA scheme used as a baseline, an auction-based algorithm A combinatorial auction framework for decentralized task allocation, ACO [27], and a locust-based task allocation for multi-UAVs, LIAM [18]. Three performance metrics were measured for each algorithm, which include the percentage of rescued survivors, mean time to rescue a survivor, and algorithm-running time.

Due to the increased difficulty of the MUTA problem as the numbers of UAVs and survivor increase in SAR missions, we experimented with multiple values of each:

- The number of UAVs was logarithmically increased in a power of 2 to include the teams of 4, 8, 16, 32, 64, and 128 UAVs.
- The number of survivors was similarly logarithmically increased to a power of 2 to include 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, and 4096 survivors.

In a similar approach to [18], different scenarios were developed using a java-based multi-UAV mission simulator, MASPlanes++ [70,71], to represent a disaster area of 111×111 regions, with nine $3 \text{ m} \times 3 \text{ m}$ blocks on each region. The survivor critical window was set to 10–72 h. Each survivor has a randomly assigned maximum lifespan, and if the survivor

is not rescued before that time, they expire. The considered UAVs are battery-operated quadrotors with 40 km/h maximum speed and 15 min of travel battery capacity with 30 unit/s as the recharge ratio. Nine charging stations and ten survivors' hotspots with a radius of 200 m were arbitrary placed in the area. The power consumption for movement was uniform across all UAVs. Time penalties of 10 and 60 s were assumed for searching a block and rescuing a survivor, respectively. The power consumption during idle time and operation time was 1 unit of power/300 ms and 1 unit of power/100 ms, respectively. Furthermore, the power penalties of 5 and 10 units were assumed for searching a block and rescuing a survivor, respectively. The selected value for each parameter is presented in Table 1.

Table 1. Parameter settings of the evaluation environment.

Parameters	Settings
Disaster area size	111 × 111 regions, 9 blocks per region
Block size	3 m × 3 m
Hotspots	10; radius: 200 m
Critical survival window	10–72 h
Plane maximum speed	40 miles/hour
Battery capacity	9000 units (15 min of travel)
Battery recharge ratio	30 units/second
Search power consumption penalty	5 units of power
Search time penalty	10 s
Rescue power consumption penalty	10 units of power
Rescue time penalty	60 s
Idle power consumption	1 unit of power/300 milliseconds
Standard power consumption	1 unit of power/100 milliseconds
Charging stations	9 stations

5. Results and Discussion

The performance of FIAM was analyzed on different SAR mission scenarios against four benchmark algorithms: OTA strategy, auction algorithm, ACO, and LIAM. The following subsections examine the performance of the five algorithms considering three performance metrics: net throughput, mean rescue time and running time performance. All experiments were performed on the same hardware, which has a desktop-class quad-core processor with Intel i7-4790K processor which runs at 4.6 GHz and paired with 16 GB RAM operating at 1866 MHz. Each experiment was run at least ten times, and the results were averaged and presented in lin-log graphs defined as a logarithmic base-2 scale of the number of UAVs on the x axis by a linear scale for the performance measures (net throughput, mean rescue time, and run-time performance) on the y axis. Lin-log graphs employing a logarithmic axis allow for simultaneous comparisons of data points drawn from various UAVs.

5.1. Percentage of Rescued Survivors

The percentage of rescued survivors (net throughput) is plotted for the number of survivors in the range [1–4096] against the number of UAVs in lin-log graphs for FIAM, OTA, auction, LIAM, and ACO algorithms in Figure 5 with 12 subfigures, each of which is for a different number of survivors.

As shown in Figure 5, the proposed FIAM algorithm clearly outperformed, with high margins, the benchmark algorithms for the most challenging scenarios (i.e., scenarios with a limited number of UAVs searching large areas for only a small number of survivors), e.g., Figure 5a,b. In those scenarios, the UAVs in the benchmark algorithms were only able to rescue up to 50% of survivors. On the other hand, FIAM was able to rescue up to 100% of survivors. Though the margin of rescue decreased as scenarios got less challenging, e.g., Figure 5f–l, FIAM still outperformed the benchmarks. This demonstrates the efficiency of the FIAM-schooling behavior in SAR missions where survivors would congregate at

shelters or supply areas. Consequently, the schooling behavior of the UAVs in FIAM would dedicate more resources to these areas.

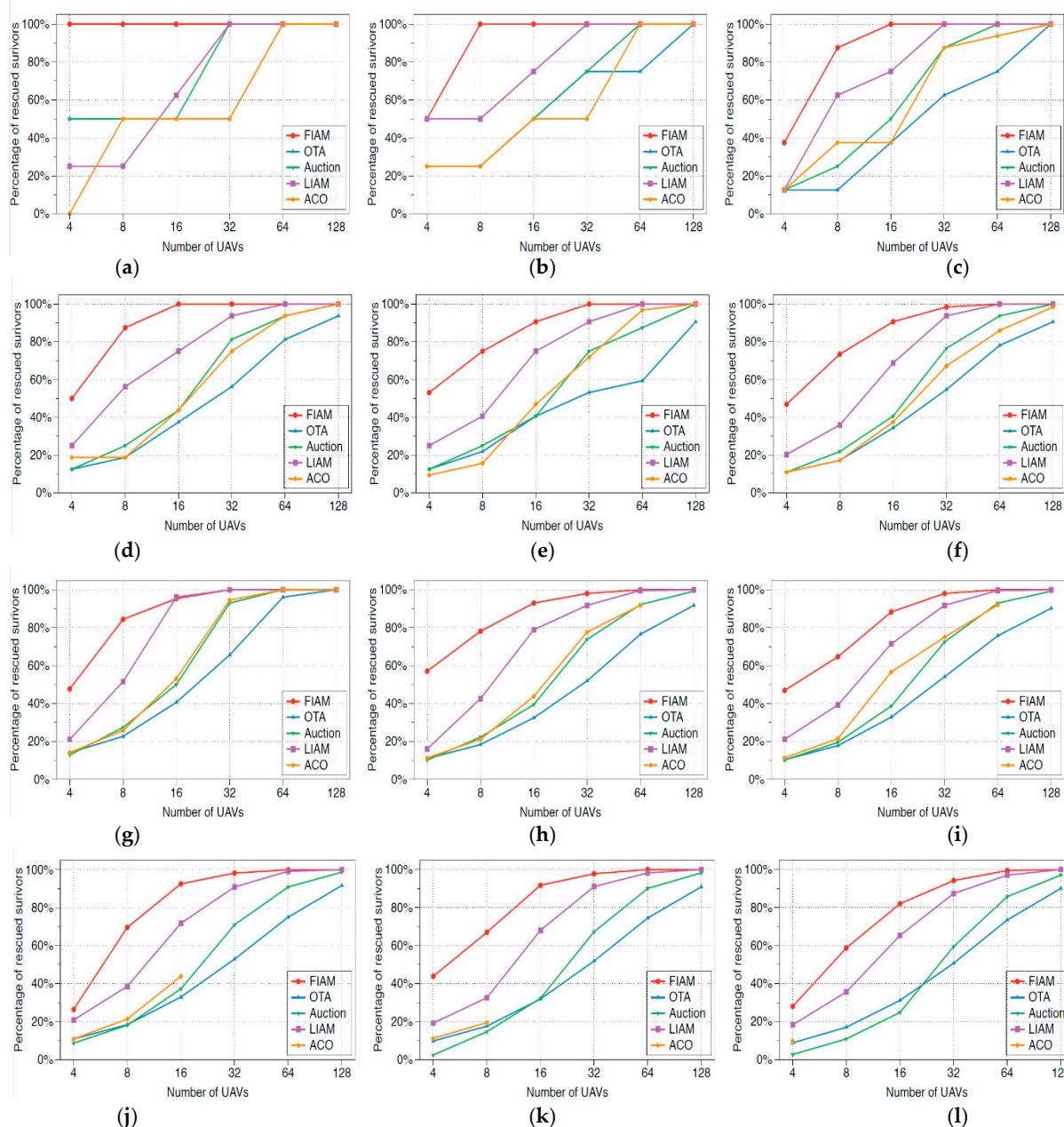


Figure 5. Net throughput as the number of survivors increased from 2 to 4096. (a) Net throughput for 2 survivors. (b) Net throughput for 4 survivors. (c) Net throughput for 8 survivors. (d) Net throughput for 16 survivors. (e) Net throughput for 32 survivors. (f) Net throughput for 64 survivors. (g) Net throughput for 128 survivors. (h) Net throughput for 256 survivors. (i) Net throughput for 512 survivors. (j) Net throughput for 1024 survivors. (k) Net throughput for 2048 survivors. (l) Net throughput for 4096 survivors. UAV, Unmanned Ariel Vehicle; FIAM, fish-inspired algorithm for task allocation in multi-UAV missions; OTA, opportunistic task allocation; ACO, ant colony optimization; LIAM, locust-inspired algorithm for MUTA problem.

LIAM proved to be the second most efficient among the benchmark algorithms. With the largest simulated fleet (128 UAVs), FIAM and LIAM were able to rescue all the survivors. However, FIAM was able to rescue more survivors with fewer UAVs throughout the different scenarios. This superiority of the FIAM performance over LIAM can be attributed to FIAM schooling behavior; when a shelter is found, a sufficient number of

UAVs is usually around to help rescue all the discovered survivors. This is in contrast to LIAM, where a UAV that finds a survivor needs to wait for a standby or a rescuer UAV, unless the algorithm has already reached the rescue phase. The performance results of OTA, auction, and ACO algorithms were comparable in most scenarios. However, ACO was less robust, causing the system to halt as it was unable to withstand the complexity of task allocation and incurred a large computational overhead, as shown in Figure 5j–l, with 1024, 2048, and 4096 survivors and 16, 8, and as few as 4 UAVs, respectively.

5.2. Mean Rescue Time

The rescue time represents the duration between the start of the simulation and the time when the survivor was rescued by a UAV. For each scenario, the rescue time was computed for each survivor, the mean was then calculated over several simulation iterations.

Figure 6 shows the 12 subfigures of the mean rescue times of FIAM, OTA, auction, LIAM, and ACO algorithms. Generally, the results show that FIAM outperformed OTA, auction, and ACO algorithms in terms of mean rescue time, with the measure's value decreasing as the numbers of survivors and UAVs were increased throughout the simulations. However, observing FIAM performance with a small number of UAVs (Figure 6a,b) indicates that FIAM takes a relatively longer time to rescue a survivor compared to the benchmarks. Regardless, these results should not be interpreted in isolation from Figure 5a,b, where FIAM comparatively rescued the largest number of survivors. In most other scenarios, when only four UAVs were used by FIAM, higher rescue times were also observed. This result could be attributed to the fact that the schooling behavior employed in FIAM cannot be demonstrated with a single fleet of UAVs (i.e., four UAVs, with one sampler UAV and three follower UAVs). In these scenarios, the UAVs will pursue survivors within the same region. If this region happens to exhibit low profitability then survivors in other regions would wait longer before being discovered. Similar to what is reported in Figure 5, ACO was halted early when the number of survivors increased to 256 or more (see Figure 6h–l) due to the complexity of the algorithm.

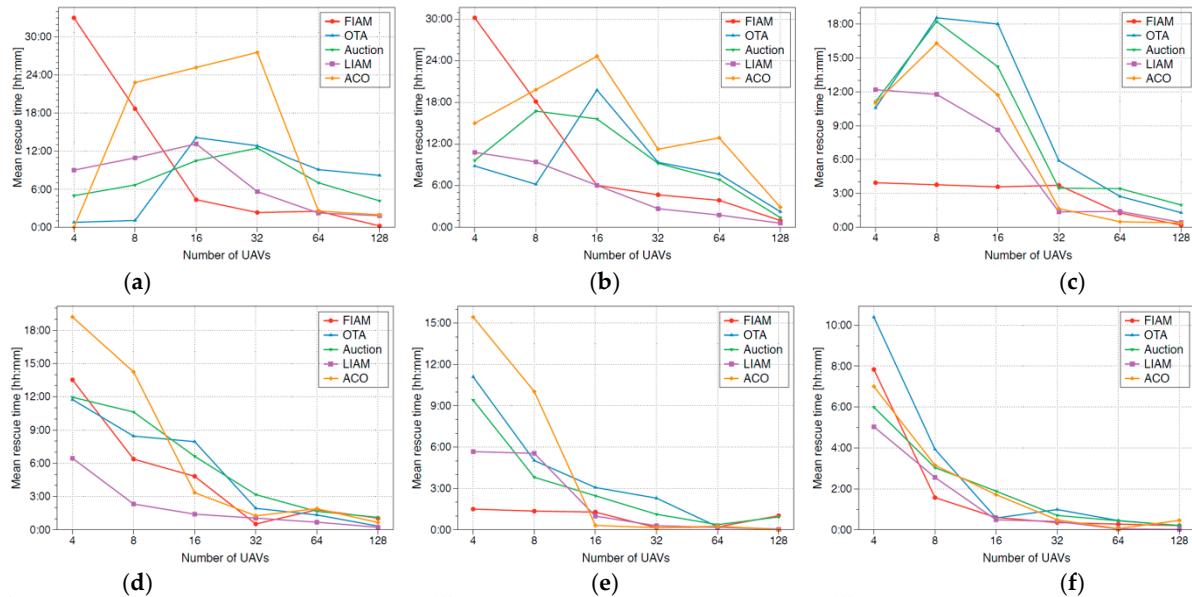


Figure 6. Cont.

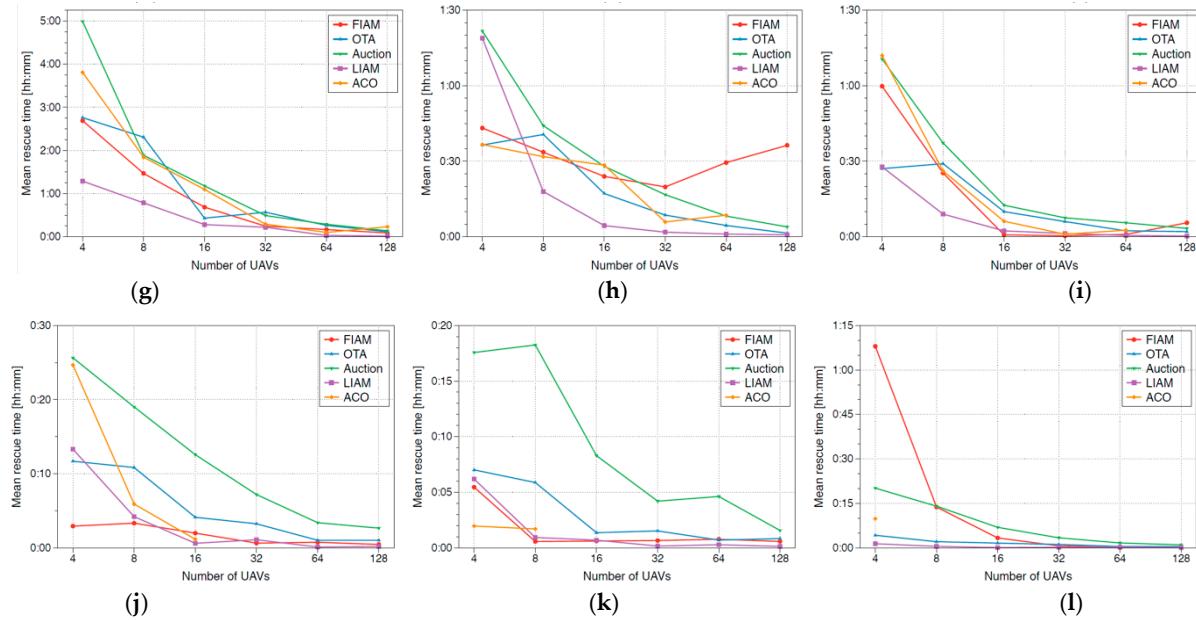


Figure 6. Mean rescue time as the number of survivors increases from 2 to 4096. (a) Mean rescue time for 2 survivors. (b) Mean rescue time for 4 survivors. (c) Mean rescue time for 8 survivors. (d) Mean rescue time for 16 survivors. (e) Mean rescue time for 32 survivors. (f) Mean rescue time for 64 survivors. (g) Mean rescue time for 128 survivors. (h) Mean rescue time for 256 survivors. (i) Mean rescue time for 512 survivors. (j) Mean rescue time for 1024 survivors. (k) Mean rescue time for 2048 survivors. (l) Mean rescue time for 4096 survivors. UAV, Unmanned Ariel Vehicle; FIAM, fish-inspired algorithm for task allocation in multi-UAV missions; OTA, opportunistic task allocation; ACO, ant colony optimization; LIAM, locust-inspired algorithm for MUTA problem.

5.3. Running Time Performance

By analyzing FIAM flowcharts, as shown in Figures 3 and 4, it can be deduced that both algorithms maintain linear running time performances, $O(n)$, where n is the number of blocks. However, previous studies show that auction and ACO algorithms suffer from quadratic [31] and exponential run times [47], respectively. These findings were experimentally confirmed by recording the running time performances of FIAM and the benchmark algorithms for each simulated scenario as illustrated in Figure 7. The dashed lines in the figures represent the performance readings that sharply extend beyond the range of values displayed on the vertical axis.

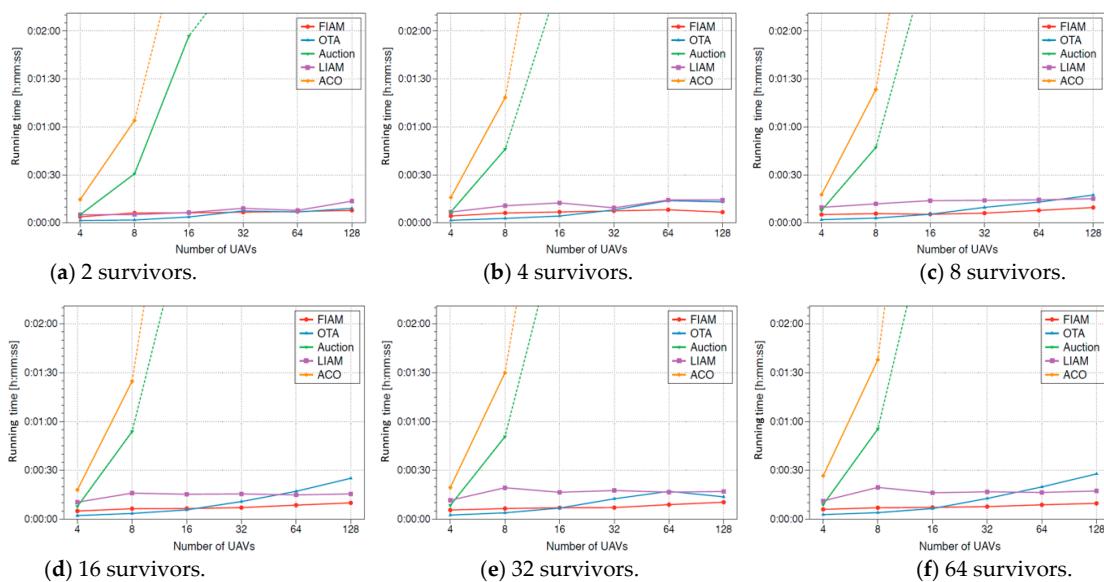


Figure 7. Cont.

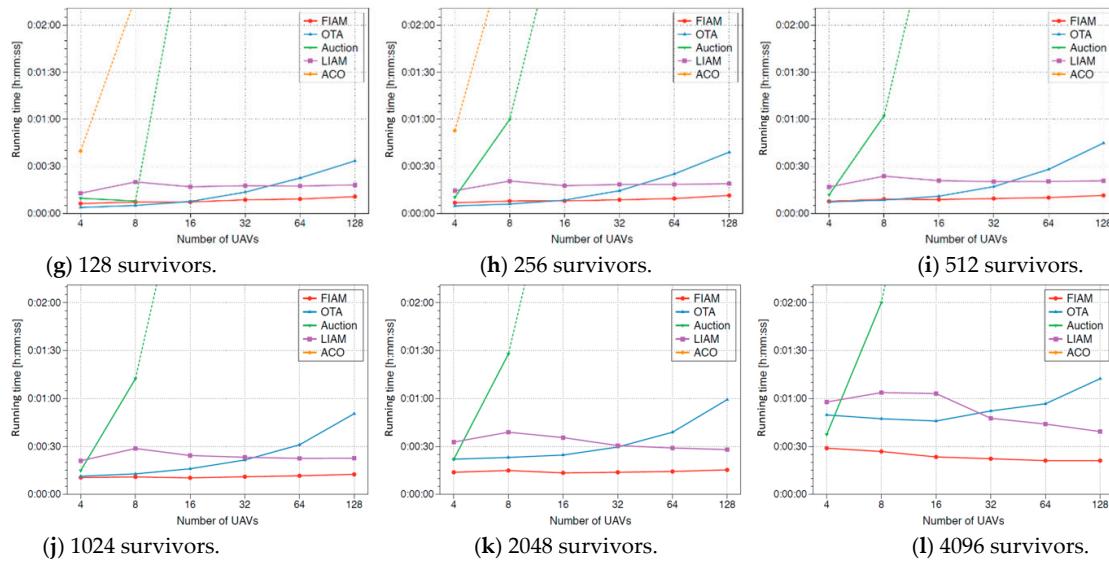


Figure 7. Running time performance as the number of survivors increases from 2 to 4096. (a) Running time for 2 survivors. (b) Running time for 4 survivors. (c) Running time for 8 survivors. (d) Running time for 16 survivors. (e) Running time for 32 survivors. (f) Running time for 64 survivors. (g) Running time for 128 survivors. (h) Running time for 256 survivors. (i) Running time for 512 survivors. (j) Running time for 1024 survivors. (k) Running time for 2048 survivors. (l) Running time for 4096 survivors. UAV, Unmanned Ariel Vehicle; FIAM, fish-inspired algorithm for task allocation in multi-UAV missions; OTA, opportunistic task allocation; ACO, ant colony optimization; LIAM, locust-inspired algorithm for MUTA problem.

As shown in Figure 7, the running time performance of the proposed algorithm was comparable to those of OTA, auction, and LIAM when the number of deployed UAVs was small (no more than 4). Subsequently, the running time performance of the auction increased sharply, while OTA, LIAM, and FIAM exhibited only a slight performance increase. The deteriorating performance of the auction algorithm was likely due to its inability to perform in a dynamic environment as new calculations are required every time a survivor is discovered. The running time performance of ACO proved to be the least efficient when the algorithm halted as the number of survivors was increased to 512–4096 survivors.

Furthermore, observing the running time performance of OTA, it can be deduced that the algorithm maintains the lowest running time in scenarios with limited numbers of UAVs (less than 32) and survivors (less than 512). This is because OTA implements a naive greedy strategy. Nevertheless, the OTA running time performance gradually increased as the number of survivors and UAVs were increased, an observation that is consistent with the finding in [18] where the complexity of $O(nm)$, rather $O(n)$ was suggested for OTA. In contrast, the running time performance of FIAM remained almost steady in all scenarios. This result is likely because the allocation of tasks in FIAM is a local inexpensive decision irrelevant to the number of UAVs or survivors.

6. Conclusions

To effectively conduct SAR missions, multiple UAVs are expected to be deployed and tasks must be properly allocated. In the literature, this is known as the MUTA problem, which is a challenge, especially as the numbers of UAVs and survivors increase. This study introduced FIAM, a novel low computational algorithm that addressed the MUTA problem based on the interesting schooling behavior of fish.

FIAM performance was evaluated on a simulated multi-UAV SAR mission at different scales of fleet size and number of survivors using a well controlled experimental framework. The algorithm performance was benchmarked to several MUTA algorithms. The experimental results show that FIAM can enhance the performance of SAR missions by remarkably increasing the percentage of rescued survivors, particularly when medium to large fleets (eight or more) of UAVs are deployed. Moreover, the mean rescue time for a sur-

vivor dramatically decreases when FIAM is used in most simulated scenarios. Furthermore, FIAM successfully maintained an impressively low running time performance.

Several limitations of the proposed method lend themselves to future work. The current results exhibited by FIAM in terms of mean rescue time are relatively high for a small UAV fleet; we intend to enhance FIAM by implementing an adoptive policy based on the fleet size. We also plan to examine FIAM performance at different scales of search areas and other multi-UAV-related problems, such as path findings and flight formation. Although FIAM was developed within the context of SAR missions, it is by no means limited to this application; it can also be applied in numerous missions that comprise executing two different groups of tasks under challenging or expensive circumstances. This includes detecting and treating missions in agriculture and find and track missions in military applications.

Author Contributions: Conceptualization, A.A.; formal analysis, A.A. and H.K.; funding acquisition, H.K.; investigation, A.A.; project administration, H.K.; resources, H.K. and K.Y.-T.; software, A.A.; supervision, H.K. and K.Y.-T.; writing—original draft, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research was supported by a grant from Researchers Supporting Unit, Project number (RSP-2020/204), King Saud University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Adão, T.; Hruška, J.; Pádua, L.; Bessa, J.; Peres, E.; Morais, R.; Sousa, J.J. Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry. *Remote Sens.* **2017**, *9*, 1110. [[CrossRef](#)]
2. Pietsch, M.; Henning, M.; Mader, D.W.; Westfeld, P.; Etterer, F. Using unmanned aerial vehicles (UAV) for monitoring biodiversity measures in periurban and agrarian landscapes. *J. Digit. Landsc. Archit.* **2018**, *3*, 273–282.
3. Kurdi, H.; AlDaood, M.F.; Al-Megren, S.; Aloboud, E.; Aldawood, A.S.; Youcef-Toumi, K. Adaptive task allocation for multi-UAV systems based on bacteria foraging behaviour. *Appl. Soft Comput.* **2019**, *83*, 105643. [[CrossRef](#)]
4. Menouar, H.; Guvenc, I.; Akkaya, K.; Uluagac, A.S.; Kadri, A.; Tuncer, A. UAV-enabled intelligent transportation systems for the smart city: Applications and challenges. *IEEE Commun. Mag.* **2017**, *55*, 22–28. [[CrossRef](#)]
5. Lee, H.I.; Yoo, D.W.; Lee, B.Y.; Moon, G.H.; Lee, D.Y.; Tahk, M.J.; Shin, H.S. Parameter-robust linear quadratic Gaussian technique for multi-agent slung load transportation. *Aerospace Sci. Technol.* **2017**, *71*, 119–127. [[CrossRef](#)]
6. Di Mauro, B.; Fava, F.; Ferrero, L.; Garzonio, R.; Baccolo, G.; Delmonte, B.; Colombo, R. Mineral dust impact on snow radiative properties in the European Alps combining ground, UAV and satellite observations. *J. Geophys. Res. Atmos.* **2015**, *120*. [[CrossRef](#)]
7. Malehmir, A.; Dynesius, L.; Paulusson, K.; Paulusson, A.; Johansson, H.; Bastani, M.; Wedmark, M.; Marsden, P. The potential of rotary-wing UAV-based magnetic surveys for mineral exploration: A case study from central Sweden. *Lead. Edge* **2017**, *36*, 552–557. [[CrossRef](#)]
8. Walter, C.; Braun, A.; Fotopoulos, G. High-resolution unmanned aerial vehicle aeromagnetic surveys for mineral exploration targets. *Geophys. Prospect.* **2020**, *68*, 334–349. [[CrossRef](#)]
9. Kurdi, H.; How, J.; Bautista, G. Bio-Inspired Algorithm for Task Allocation in Multi-UAV Search and Rescue Missions. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, San Diego, CA, USA, 4–8 January 2016.
10. Sampedro, C.; Rodriguez-Ramos, A.; Bavle, H.; Carrio, A.; de la Puente, P.; Campoy, P. A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques. *J. Intell. Robot. Syst.* **2019**, *95*, 601–627. [[CrossRef](#)]
11. Yu, X.; Li, C.; Zhou, J. A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios. *Knowl. Based Syst.* **2020**, *204*, 106209. [[CrossRef](#)]
12. Waharte, S.; Trigoni, N. Supporting Search and Rescue Operations with UAVs. In Proceedings of the 2010 International Conference on Emerging Security Technologies, Canterbury, UK, 6–7 September 2010.
13. Berger, J.; Lo, N. An innovative multi-agent search-and-rescue path planning approach. *Comput. Oper. Res.* **2015**, *53*, 24–31. [[CrossRef](#)]
14. Qi, J.; Song, D.; Shang, H.; Wang, N.; Hua, C.; Wu, C.; Qi, X.; Han, J. Search and rescue rotary-wing UAV and its application to the lushan ms 7.0 earthquake. *J. Field Robot.* **2016**, *33*, 290–321. [[CrossRef](#)]

15. Zhao, W.; Meng, Q.; Chung, P.W.H. A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario. *IEEE Trans. Cybern.* **2016**, *46*, 902–915. [[CrossRef](#)] [[PubMed](#)]
16. Silvagni, M.; Tonoli, A.; Zenerino, E.; Chiaberge, M. Multipurpose UAV for search and rescue operations in mountain avalanche events. *Geomat. Nat. Hazards Risk* **2017**, *8*, 18–33. [[CrossRef](#)]
17. Matos, A.; Silva, E.; Almeida, J.; Martins, A.; Ferreira, H.; Ferreira, B.; Alves, J.; Dias, A.; Fioravanti, S.; Bertin, D.; et al. Unmanned maritime systems for search and rescue. In *Search and Rescue Robotics—From Theory to Practice*; Cubber, G.D., Doroftei, D., Rudin, K., Berns, K., Serrano, D., Sanchez, J., Govindaraj, S., Bedkowski, J., Roda, R., Eds.; InTech: London, UK, 2017; pp. 77–92.
18. Kurdi, H.A.; Aloboud, E.; Alalwan, M.; Alhassan, S.; Alotaibi, E.; Bautista, G.; How, J.P. Autonomous task allocation for multi-UAV systems based on the locust elastic behavior. *Appl. Soft Comput.* **2018**, *71*, 110–126. [[CrossRef](#)]
19. Goian, A.; Ashour, R.; Ahmad, U.; Taha, T.; Almoosa, N.; Seneviratne, L. Victim localization in USAR scenario exploiting multi-layer mapping structure. *Remote Sens.* **2019**, *11*, 2704. [[CrossRef](#)]
20. Casper, J.; Murphy, R.R. Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2003**, *33*, 367–385. [[CrossRef](#)]
21. Murphy, R.R. *Disaster Robotics*; The MIT Press: Cambridge, UK, 2014.
22. Yan, Z.; Jouandeau, N.; Cherif, A.A. A survey and analysis of multi-robot coordination. *Int. J. Adv. Robot. Syst.* **2013**, *10*. [[CrossRef](#)]
23. Bekmezci, İ.; Sahingoz, O.K.; Temel, Ş. Flying ad-hoc networks (FANETs): A survey. *Ad Hoc Netw.* **2013**, *11*, 1254–1270. [[CrossRef](#)]
24. Korsah, G.A.; Stentz, A.; Dias, M.B. A comprehensive taxonomy for multi-robot task allocation. *Int. J. Robot. Res.* **2013**, *32*, 1495–1512. [[CrossRef](#)]
25. Brown, C.; Laland, K.; Krause, J. *Fish Cognition and Behavior*; Wiley-Blackwel: Oxford, UK, 2008.
26. Segui-Gasco, P.; Shin, H.; Tsourdos, A.; Seguí, V.J. A Combinatorial Auction Framework for Decentralised Task Allocation. In Proceedings of the 2014 IEEE Globecom Workshops (GC Wkshps), Austin, TX, USA, 8–12 December 2014.
27. Wang, L.; Wang, Z.; Hu, S.; Liu, L. Ant colony optimization for task allocation in multi-agent systems. *China Commun.* **2013**, *10*, 125–132. [[CrossRef](#)]
28. Macarthur, K. Multi-Agent Coordination for Dynamic Decentralised Task Allocation. Doctoral Thesis, University of Southampton, Southampton, UK, 2011.
29. Bertsekas, D.P. The auction algorithm: A distributed relaxation method for the assignment problem. *Ann. Oper. Res.* **1988**, *14*, 105–123. [[CrossRef](#)]
30. Johnson, L.; Choi, H.L.; How, J.P. The hybrid information and plan consensus algorithm with imperfect situational awareness. In *Distributed Autonomous Robotic Systems*; Chong, N.Y., Cho, Y.J., Eds.; Springer: Tokyo, Japan, 2016; pp. 221–233.
31. Ma, Y.; Li, B.; Zhang, Y.; Zhu, J. Efficient Auction Mechanism with Group Price for Resource Allocation in Clouds. In Proceedings of the 2014 Second International Conference on Advanced Cloud and Big Data, Huangshan, China, 20–22 November 2014; pp. 85–92.
32. Dorigo, M.; Maniezzo, V.; Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **1996**, *26*, 29–41. [[CrossRef](#)] [[PubMed](#)]
33. Greenfield, G.; Machado, P. Ant- and ant-colony-inspired ALife visual art. *Artif. Life* **2015**, *21*, 293–306. [[CrossRef](#)]
34. Shirkande, S.D.; Vatti, R.A. ACO Based Routing Algorithms for Ad-Hoc Network (WSN, MANETs): A Survey. In Proceedings of the 2013 International Conference on Communication Systems and Network Technologies, Gwalior, India, 6–8 April 2013; pp. 230–235.
35. Duan, H.; Yu, Y.; Zou, J.; Feng, X. Ant colony optimization-based bio-inspired hardware: Survey and prospect. *Trans. Inst. Meas. Control* **2010**, *34*, 318–333. [[CrossRef](#)]
36. Taixiong, Z.; Liangyi, Y. Optimal Ant Colony Algorithm Based Multi-Robot Task Allocation and Processing Sequence Scheduling. In Proceedings of the 7th World Congress on Intelligent Control and Automation, Chongqing, China, 25–27 June 2008; pp. 5693–5698.
37. Wu, H.; Li, H.; Xiao, R.; Liu, J. Modeling and simulation of dynamic ant colony's labor division for task allocation of UAV swarm. *Phys. A Stat. Mech. Appl.* **2018**, *491*, 127–141. [[CrossRef](#)]
38. Boussaïd, I.; Lepagnot, J.; Siarry, P. A survey on optimization metaheuristics. *Inf. Sci.* **2013**, *237*, 82–117. [[CrossRef](#)]
39. Sudholt, D.; Thyssen, C. Running time analysis of ant colony optimization for shortest path problems. *J. Discrete Algorithms* **2012**, *10*, 165–180. [[CrossRef](#)]
40. Neshat, M.; Sepidnam, G.; Sargolzaei, M.; Toosi, A.N. Artificial fish swarm algorithm: A survey of the state-of-the-art, hybridization, combinatorial and indicative applications. *Artif. Intell. Rev.* **2014**, *42*, 965–997. [[CrossRef](#)]
41. Faheem, M.; Gungor, V.C. Capacity and spectrum-aware communication framework for wireless sensor network-based smart grid applications. *Comput. Stand. Interfaces* **2017**, *53*, 48–58. [[CrossRef](#)]
42. He, M.; Liu, F.; Miao, Z.; Zhou, H.; Chen, Q. A mechanism of topology optimization for underwater acoustic sensor networks based on autonomous underwater vehicles. *Int. J. Distrib. Sens. Netw.* **2017**, *13*, 155014771668697. [[CrossRef](#)]
43. Zhang, K.; Jia, H.; Lv, H. Coverage-enhancing approach in multimedia directional sensor networks for smart transportation. *Multimed. Tools Appl.* **2016**, *75*, 17593–17615. [[CrossRef](#)]

44. El-said, S.A.; Osamaa, A.; Hassanien, A.E. Optimized hierarchical routing technique for wireless sensors networks. *Soft Comput.* **2016**, *20*, 4549–4564. [[CrossRef](#)]
45. Xia, J. Coverage Optimization Strategy of Wireless Sensor Network Based on Swarm Intelligence Algorithm. In Proceedings of the 2016 International Conference on Smart City and Systems Engineering (ICSCSE), Zhangjiajie, China, 25–26 November 2016; pp. 179–182.
46. Zhang, F.S.; Li, S.W.; Hu, Z.G.; Du, Z. Fish swarm window selection algorithm based on cell microscopic automatic focus. *Clust. Comput.* **2017**, *20*, 485–495. [[CrossRef](#)]
47. Qin, L.; Sun, K.; Li, S. Maximum Fuzzy Entropy Image Segmentation Based on Artificial Fish School Algorithm. In Proceedings of the 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 27–28 August 2016; pp. 164–168.
48. Ferreira, F.A.B.S.; Madeiro, F. A Fish School Search Based Algorithm for Image Channel-Optimized Vector Quantization. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016; pp. 1680–1685.
49. Dong, D.; Chen, W.; Hou, Z. Hydrodynamic Performance and Flow Control of Two Flexible Bodies in Fish Schooling-Like Configuration. In Proceedings of the 3rd International Conference on Control, Automation and Robotics (ICCAR), Nagoya, Japan, 24–26 April 2017; pp. 491–494.
50. Mao, J.; Xiao, G.; Sheng, W.; Qu, Z.; Liu, Y. Research on realizing the 3D occlusion tracking location method of fish's school target. *Neurocomputing* **2016**, *214*, 61–79. [[CrossRef](#)]
51. Zhang, F.; Ennasr, O.; Litchman, E.; Tan, X. Autonomous sampling of water columns using gliding robotic fish: Algorithms and harmful-algae-sampling experiments. *IEEE Syst. J.* **2016**, *10*, 1271–1281. [[CrossRef](#)]
52. Huang, Z.; Chen, Y.; Shi, X. A synergetic semantic role labeling model with the introduction of fluctuating force accompanied with word sense information. *Intell. Data Anal.* **2017**, *21*, 5–18. [[CrossRef](#)]
53. Carneiro, R.F.; Bastos-Filho, C.J.A. Improving the Binary Fish School Search Algorithm for Feature Selection. In Proceedings of the 2016 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Cartagena, Colombia, 2–4 November 2016; pp. 1–6.
54. He, S.; Belacel, N.; Chan, A.; Hamam, H.; Bouslimani, Y. A hybrid artificial fish swarm simulated annealing optimization algorithm for automatic identification of clusters. *Int. J. Inf. Technol. Decis. Mak.* **2016**, *15*, 949–974. [[CrossRef](#)]
55. Shen, H.; Chen, J.; Li, H.; Zhou, Z. Research on Real-Rime Flight Path Planning of UAV Based on Grey Prediction. In Proceedings of the 9th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 10–11 December 2016; pp. 62–67.
56. Ma, X.; Tang, R.; Kang, J.; Liu, Q. Optimizing Application Layer Multicast Routing via Artificial Fish Swarm Algorithm. In Proceedings of the 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Changsha, China, 13–15 August 2016; pp. 115–120.
57. de Albuquerque, I.M.C.; Filho, J.M.; de Lima Neto, F.B.; de Oliveira Silva, A.M. Solving Assembly Line Balancing Problems with Fish School Search Algorithm. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2016; pp. 1–8.
58. Sun, X.; Fang, H. Speed Governor PID Gains Optimal Tuning of Hydraulic Turbine Generator Set with an Improved Artificial Fish Swarm Algorithm. In Proceedings of the 2016 IEEE International Conference on Information and Automation (ICIA), Ningbo, China, 1–3 August 2016; pp. 2033–2035.
59. Li, X.; Shao, Z.; Qian, J. An optimizing method based on autonomous animate: Fish swarm algorithm. *Syst. Eng. Theory Pract.* **2002**, *22*, 32–38.
60. Tian, W.; Liu, J. An Improved Artificial Fish Swarm Algorithm for Multi Robot Task Scheduling. In Proceedings of the Fifth International Conference on Natural Computation, Tianjin, China, 14–16 August 2009; pp. 127–130.
61. Zheng, T.; Li, J. Multi-Robot Task Allocation and Scheduling Based on Fish Swarm Algorithm. In Proceedings of the 8th World Congress on Intelligent Control and Automation, Jinan, China, 7–9 July 2010; pp. 6681–6685.
62. Bshary, R.; Gingins, S.; Vail, A.L. Social cognition in fishes. *Trends Cogn. Sci.* **2014**, *18*, 465–471. [[CrossRef](#)]
63. Brownscombe, J.W.; Cooke, S.J.; Algera, D.A.; Hanson, K.C.; Eliason, E.J.; Burnett, N.J.; Danylchuk, A.J.; Hinch, S.G.; Farrell, A.P. Ecology of exercise in wild fish: Integrating concepts of individual physiological capacity, behavior, and fitness through diverse case studies. *Integr. Comp. Biol.* **2017**, *57*, 281–292. [[CrossRef](#)]
64. Filho, C.J.A.B.; Neto, F.B.D.L.; Lins, A.J.C.C.; Nascimento, A.I.S.; Lima, M.P. A Novel Search Algorithm Based on Fish School Behavior. In Proceedings of the 2008 IEEE International Conference on Systems, Man and Cybernetics, Singapore, 12–15 October 2008; pp. 2646–2651.
65. Reynolds, C.W. Flocks, Herds and Schools: A Distributed Behavioral Model. In Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, New York, NY, USA, 27–31 July 1987; pp. 25–34.
66. Lönnstedt, O.M.; Ferrari, M.C.; Chivers, D.P. Lionfish predators use flared fin displays to initiate cooperative hunting. *Biol. Lett.* **2014**, *10*, 20140281. [[CrossRef](#)] [[PubMed](#)]
67. Pitcher, T.J.; Magurran, A.E.; Winfield, I.J. Fish in larger shoals find food faster. *Behav. Ecol. Sociobiol.* **1982**, *10*, 149–151. [[CrossRef](#)]
68. Ward, A.J.W.; Krause, J.; Sumpter, D.J.T. Quorum decision-making in foraging fish shoals. *PLoS ONE* **2012**, *7*, e32411. [[CrossRef](#)] [[PubMed](#)]

69. Kurdi, H.; Al-Megren, S.; Aloboud, E.; Alnuaim, A.A.; Alomair, H.; Alothman, R.; Muhayya, A.B.; Alharbi, N.; Alenzi, M.; Youcef-Toumi, K. Bee-inspired task allocation algorithm for multi-UAV search and rescue missions. *Int. J. Bio Inspired Comput.* **2020**, *11*, 111. [[CrossRef](#)]
70. Pujol-Gonzalez, M.; Cerquides, J.; Meseguer, P. MAS-Planes: A Multi-Agent Simulation Environment to Investigate Decentralised Coordination for Teams of UAVs. In Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2014), Paris, France, 5–9 May 2014; pp. 1695–1696.
71. Aloboud, E. GitHub—xEbtesam/MASPlanes: MASPlanes Simulator for the Development of Distributed Coordination Algorithms, MASPlanes. 10 December 2016. Available online: <https://github.com/xEbtesam/MASPlanes> (accessed on 7 September 2020).