This is an original manuscript of an article published by Taylor & Francis in Cybernetics and Systems on 27 Jul 2023, available at: https://doi.org/10.1080/01969722.2023.2240645.

Article Information

Article Type:	research-article
Journal Title:	Cybernetics and Systems
Publisher:	Taylor & Francis
DOI Number:	10.1080/01969722.2023.2240645
Volume Number:	0
Issue Number:	0
First Page:	1
Last Page:	18
Copyright:	© 2023 Taylor & Francis Group, LLC

Genetic Algorithm-Based Task Assignment for Fleet of Unmanned Surface Vehicles in Dynamically Changing Environment

Left running head: M. DVORAK ET AL.

Short title : Cybernetics and Systems: An International Journal

AQ0

Miroslav Dvorak[a], **Petr Dolezel**[a], **Dominik Stursa**[a] and **Mohamed Chouai**[b] AQ1

[a]Faculty of Electrical Engineering, University of Pardubice, Pardubice, Czech Republic;

[b]Alfred Wegener Institute Helmholtz Centre for Polar and Marine Research, Bremerhaven, Germany

Corresponding Author

CONTACT Petr Dolezel petr.dolezel@upce.cz Faculty of Electrical Engineering, University of Pardubice, Studentska 95, 532 10 Pardubice, Czech Republic.

Abstract

Unmanned vehicles are gaining the attention of professional operators and the general public. The implementation of unmanned vehicles is evident in, among other fields, emergency management, agriculture, traffic monitoring, post-disaster operations, and delivery of goods. Naturally, a group of unmanned vehicles can cooperatively complete operations more proficiently than a single vehicle. However, several issues must be resolved before a stable and reliable group of unmanned vehicles can be generally deployed to solve tasks in civil infrastructures and in industrial facilities. Here, a framework for the guidance of a fleet of unmanned surface vehicles is proposed. The framework utilizes several levels of control, namely Global Planning Level, Local Planning Level, and Low-Level Control. While the individual vehicles are completely autonomous in their operational locomotion and obstacle avoidance (low-level control and local planning), the task assignment for each vehicle (or group of them) is provided by a global planning process, based on the genetic algorithm.

The framework provides a concept to solve complex tasks for the fleet of unmanned surface vehicles (USVs). This includes, but is not necessarily limited to, a dynamically changing environment, different types of USVs with special abilities, multiple types of areal restrictions and obstacles, different restrictions for individual USVs, cooperation of multiple USVs to solve their subtasks, energy consumption optimization, etc. The framework can be advantageously applied to tasks such as warehouse logistics, surface maintenance, area exploration, etc. At the end of the study, the application of the framework is presented using a simulated example of cooperative problem solving using six vehicles.

KEYWORDS

Fleet management control system; fleet of unmanned vehicles; genetic algorithm; global planning

Note: Any change made here needs to be made in the corresponding section at the end of the article.

University of Pardubice 10.13039/501100016365

The work has been supported by the SGS grant at the University of Pardubice. This support is very gratefully acknowledged.

Introduction

In recent years, the rapid and remarkable growth of unmanned vehicles (UV) has attracted many researchers and industrial experts. The implementation of UVs is evident in such applications which are dangerous, impossible, or tedious for human operators. Furthermore, fleets of UVs provide better precision and reliability, while reducing the training procedure of qualified human personnel (Guney and Raptis 2021).

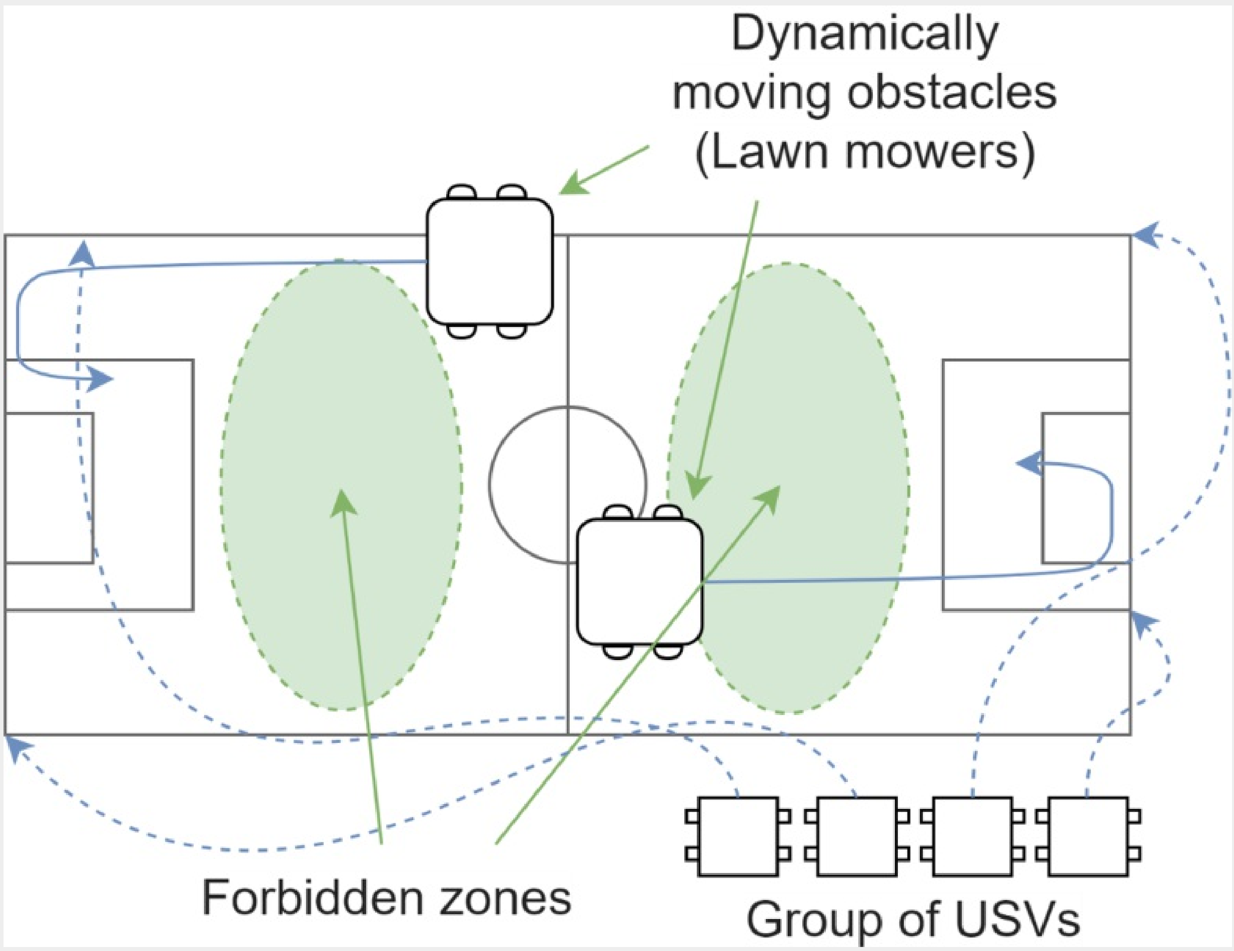
The application of UVs includes, among other areas, emergency management (Arafat and Moh 2019), agricultural spheres (Dolezel et al. 2021), wildfire (Bailon-Ruiz, Bit-Monnot, and Lacroix 2022), traffic monitoring (S. Lee, Kim, et al. 2022), search operations (Zhang et al. 2022), post-disaster operations (Arafat and Moh 2018), and delivery of goods (S. Y. Lee, Han, et al. 2022).

Along with the increasing number of UVs, the demand to simultaneously operate a group of autonomous mobile units in civil infrastructures and industrial facilities naturally rises. This phenomenon is called a fleet management control problem (Andreasson et al. 2015). Usually, a fleet management control system (FMCS) manages all tasks related to the control of any UV in the controlled area. In other words, the FMCS is expected to perform decision making, regarding the best strategies to solve the problems associated with its functioning, such as routing, scheduling, mutual cooperation of UVs, and conflict avoidance. This should be guaranteed regardless of the number of UVs (Vivaldini et al. 2015).

Although UVs and fleets of UVs are in common use, some challenges still persist and need to be addressed (Poudel and Moh 2022). The list of open issues includes the requirement of avoiding inter-vehicle collisions, high-safety integrity of UV fleets applied in civil infrastructures, the scarcity of communication band-width for efficient data sharing, and the demand for coordinating behaviors of UVs to achieve optimal efficiency (Miao, Wang, and Pang 2022). In accordance with the last item on this list, the issue of unmanned surface vehicle (USV) fleet motion planning and task assignment within an area with dynamically moving obstacles and forbidden zones, is addressed in this study. This article is an extension of work originally presented in the 16th International Conference on Soft Computing Models in Industrial and Environmental Applications (Dvorak et al. 2022). While a general solution concept was presented and one iteration of this concept was simulated in the original paper, here the whole concept is implemented for the complex problem of USV fleet motion in an environment with dynamically moving obstacles and restricted zones.

The USV motion planning problem has been comprehensively examined in literature. A good overview is given, for example, in the freely available Ph.D. thesis by (Singh 2019). Various proposed solutions can be principally divided according to their completeness, computational complexity, and optimality (Parker 2009). In addition, big differences between the requirements for the FMCS occur according to the particular task and environment. Here, an intuitive approach to a FMCS, based on a genetic algorithm, is proposed. The proposed FMCS is designed to iteratively divide the task into a set of specific subtasks, where each subtask is assigned to an individual USV or a group of cooperating units capable of solving the subtask. Then, the USVs are guided to perform a collision-free locomotion from their initial location to their respective subtask. Finally, each USV begins to solve its subtask. Meanwhile, the task or the environment may change, or various unexpected events (including failures) of individual USVs may occur. Therefore, the proposed FMCS continuously evaluates the current state variables, and adapts the subtasks of any USV in a response to dynamically changing conditions of the environment. A real-life example of the considered FMCS is depicted in Figure 1.

Figure 1. Example of the proposed functionality of the fleet management control system. A group of unmanned surface vehicles is expected to line the pitch. To deal with this goal, the pitch is divided into submaps and each USV is assigned to line its part of the pitch. However, the pitch contains forbidden zones, which need to be avoided by the USVs. Furthermore, the pitch is simultaneously mowed with lawn-mowers. Hence, the unmanned vehicles must avoid contact with lawn-mowers and other obstacles, which can be static or dynamically moving. Due to these circumstances, the pitch division is performed every defined period of time under the current environment conditions, in order to get the close-to-optimal solution.



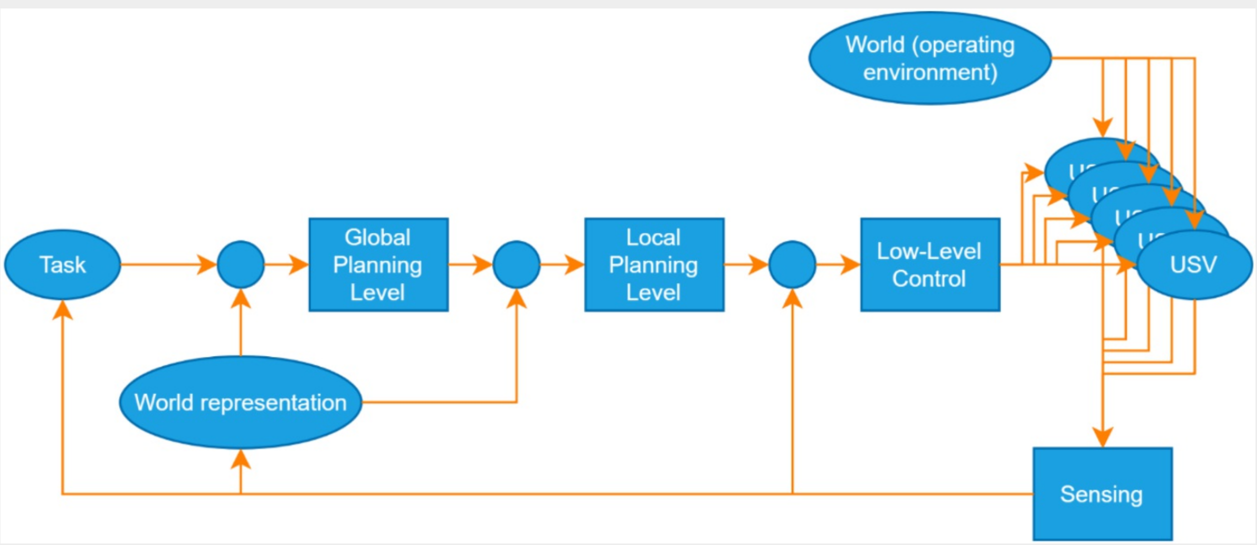
The rest of the study is structured as follows. The next section generally defines the intended framework and the optimization algorithm that is implemented to find the appropriate assignment of tasks to each USV. Then, the utilization of the framework in an applied context is presented in order to provide a set of specific examples to follow. The study is concluded with a discussion and possible ways of further research.

Materials and Methods

Fleet Management Control System Framework

The FMCS designed according to this framework is expected to guide a fleet of USVs in order to solve a defined task. This framework defines the operations of the FMCS into several levels. Specifically, these levels are Global Planning Level, Local Planning Level, and Low-Level Control (adapted according to Moysiadis et al. 2020). The diagram of the FMCS framework is illustrated in Figure 2.

Figure 2. A diagram of the fleet management control system framework. A collective task is processed by the Global Planning Level module, which provides the subtasks for each USV. Then, the Local Planning Level module provides the specific orders for each USV in order to solve each subtask. The orders are executed using the Low-Level Control module. In this level, each USV also perceives the local environment (sensing block) and reacts to possible obstacles or other events. This operational information is provided as feedback to the fleet management control system to adapt the world representation for further planning, and to adapt the task itself.



Low-Level Control

At this level, the orders from the higher level are directly executed by each USV. This often includes following a defined path, loading/unloading cargo, manipulating an object, cooperating with another USV, etc. Furthermore, the reactive behavior of the USVs, often related to the safety of each USV and the operational environment, is addressed at this level. The control is based on direct information from various perception sensors since the quantities such as distance, touch, position, and pose are crucial for this control level. The reaction of a USV is mostly determined by a control law based on some classical approach (PID control, 2-state control, etc.) (Comasolivas et al. 2015).

Local Planning Level

Local planning is designed to deliver a sequence of instructions to each USV in order to solve its subtask. The local planning level is strictly task-relevant, and it cannot be directly generalized. Typically, it involves tasks such as generating the reference movement trajectory for each USV, searching the optimal route for retrieving multiple items, planning for multi-USV collaboration to solve a subtask, etc. The planning algorithm uses the information from the world representation model rather that from hardware sensors. For the implementation, advanced control laws (predictive control, LQ control) are regularly considered in combination with a selected state-space search technique, and a dynamic or kinematic model of the USV (Sharma, Dusek, and Honc 2017).

Global Planning Level

A global task is divided into subtasks that can be solved by individual USVs or groups of cooperating USVs at this level. This division is made on the basis of world representation, which is reconstructed from the a priori information as well as from the hardware sensors. Depending on how accurate the required results need to be, the world representation can be designed at different levels of abstraction. This can be a coarse grid of discrete integer values, or a sophisticated dynamic model containing detailed 3D objects including their equations of motion.

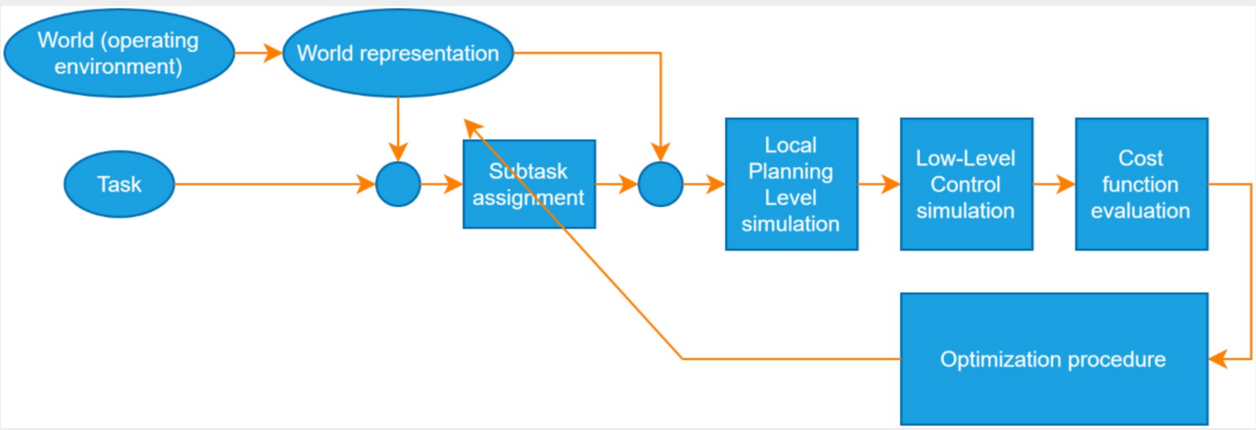
Since a dynamically changing environment and external factors are often considered, this framework performs the division into the subtasks periodically, based on the current state of the task, to keep the subtasks as efficient and up-to-date as possible.

Formally, the problem consists of USV assignment for each subtask. Initially, let $V = \{v_1, v_2, \ldots, v_N\}$ be the set of USVs at some position S denoted by planar coordinates and orientation $(x_{Si}, y_{Si}, \alpha_{Si})$, $i = 1, 2, \ldots, N$. Let $T = \{t_1, t_2, \ldots, t_M\}$ be the set of subtasks that together form a global task. Additionally, let the p_{ij} be a path of USV v_i to reach the destination to complete the task t_j from its current position. Then, the solution to the problem is to assign one or more vehicles with their paths to each problem, i.e., to form a set of tuples $(t_j, v_{i1}, p_{i1j}, v_{i2}, p_{i2j,\ldots})$, $j = 1, 2, \ldots, M$. Note that:

- It is not necessary for all USVs to be assigned to a task.
- Multiple types of USVs can be used within a task.
- Some tasks can only be accomplished by specific USVs.
- Some tasks can only be accomplished by a group of cooperating USVs.

From the computational point of view, the Global Planning Level includes a repeatedly executed optimization procedure, which minimizes the cost function. The cost function takes into account the total cost to solve the problem. This can be, for example, the total time required to solve a task, total resource consumption, etc. Each execution of the optimization procedure is based on the current state of the USVs, changes in the environment, and also on the current degree of goal achievement. In order to successfully find a solution, the Global Planning Level module needs to assess a complex simulation of the environment (referred to as world representation), as well as the models of each USV and associated Local Planning Level model. Note that all relevant features of the task or the environment, which may affect the solution, need to be included in the world representation. These features include special abilities of individual USVs, any type of areal restrictions and obstacles, various specific restrictions for individual USVs, cooperation of multiple USVs necessary to solve their subtasks, etc. A flowchart of the optimization procedure is shown in Figure 3.

Figure 3. A flowchart of the optimization procedure in the Global Planning Level. The cost function is evaluated based on the complex simulation of the Local Planning Level module and Low-Level Control module, as well as the operating environment representation.

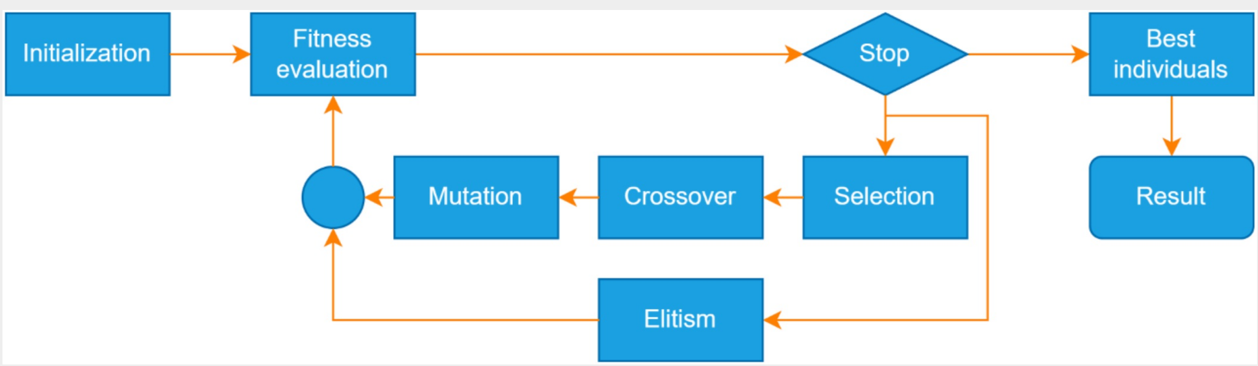


The optimization procedure for global planning may be established based on a number of algorithms and their combinations. A survey of various approaches can be found in (Shelkamy et al. 2020), (Seenu et al. 2020). The presented framework implements a genetic algorithm-based approach (GA). Since a GA has excellent parallel capabilities, it efficiently provides acceptable solutions that improve over time. It can also optimize various kinds of tasks including continuous, discrete, and multi-objective problems with arbitrary constraints. This decision is further supported by a survey of state-of-the-art applications of the genetic algorithm to related topics. Park et al. (2021) proposed mission planning optimization for USVs using a specifically designed GA, Liu et al. (2019) dealt with evolution-based unmanned aerial vehicles path planning in a complex environment. Guo et al. (2019) used an improved genetic algorithm for the path planning of USVs, and Han et al. (2021) introduced a modified GA for task assignment for a heterogeneous unmanned aerial vehicle system. A comprehensive survey of global planning algorithms for task assignment is available in (Poudel and Moh 2022).

Genetic Algorithm

The family of GAs is probably the most commonly implemented representative of the group of evolutionary algorithms. It is a stochastic population-based search technique for finding a close-to-optimal solution of an optimization problem, based on a natural selection process and genetic operators (Goldberg and Holland 1988). The GA uses a population of chromosomes representing potential solutions to the task. In each generation, the GA creates a new generation of possible solutions by selecting chromosomes according to their level of fitness, which is related to the cost function of the solution. The selected chromosomes are then bred together using genetic operators; mainly crossover and mutation. After a number of generations, this process is expected to lead to close-to-optimal solutions. A typical structure of the genetic algorithm is shown in Figure 4.

Figure 4. A structure of a genetic algorithm. Initially, the first generation of individuals representing the potential solutions is generated. Then, the algorithm cycle begins with fitness evaluation of each individual. Statistically more suitable individuals are selected for crossover and mutation. In parallel, a small group of best individuals is directly moved to the next generation. The multiple repetition of the algorithm cycle is expected to provide a close-to-optimal solution of the task.



The specific operators of the GA are defined in relation to the solved task. In order to gain an efficient variant of GA, the genetic operators need to cover all aspects of the considered USVs in the fleet (physical properties and limitations, special capabilities, operating costs, etc.), as well as the specifics of the environment. The design of task-specific genetic operators is demonstrated in the experimental section below.

Utilization of the Framework in an Applied Context

Here, a detailed implementation of the genetic algorithm-based FMCS framework is established. In order to show the particular aspects of the FMCS design, various implementations of the genetic algorithm are tested, based on the specific task for a fleet of USVs to solve.

Problem Formulation

The aim of this demonstrative problem is to guide a fleet of USVs to draw a pattern in 2D space. The environment includes forbidden zones and dynamically moving obstacles, as shown in Figure 4. Six USVs are considered to fulfill this task, and their positions and orientations at the beginning of the experiment are also indicated in the mentioned figure.

In order to keep this study clear and consistent, the fleet of the USVs consists of the same type of vehicles, all are capable of moving in any direction at a constant speed, and all are capable of avoiding obstacles immediately in front of the vehicle. Each USV is equipped with a device that allows the USV to draw a line, i.e., a part of the required pattern, on the surface. The path of the dynamic obstacles is given by polygonal lines, as seen in Figure 4. Both dynamic obstacles move at a constant speed equal to the maximum speed of the USVs.

Implementation of the Fleet Management Control System Framework

The Global Planning Level, Local Planning Level, and Low-Level Control need to be implemented in order to successfully fulfill the task of guiding the fleet. The Global Planning Level is expected to divide the global task into subtasks, and to assign them to the USVs. In this specific case, it means to divide the whole area to be drawn into subareas and to guide each USV to manage its subarea. The Local Planning level is supposed to plan a path for each USV to draw all parts of the pattern in the assigned subarea. The Low-Level Control module, which ensures safe execution of orders from higher levels, is strictly connected to the hardware implementation of the vehicles and is not dealt with here. Other modules are discussed below.

Local Planning Level Implementation

The aim of the module for this experiment is to plan the path for each USV to draw all parts of the pattern in the subarea, which is assigned to the USV in the Global Planning Level. The path planning is performed using a world representation, which is expected to include the information about the pattern to be drawn, free space, and prohibited zones (static forbidden zones and dynamically moving obstacles). In this demonstrative experiment, the world representation is abstracted as the array of 80 columns and 60 rows. Zeros, ones, and twos in this array represent free slots, the parts of the pattern which must be drawn, and obstacles, respectively. Each USV is represented as a discrete entity, able to perform any of the four following steps: move one step forward; turn left and move one step forward; turn right and move one step forward; turn back and move one step forward. Furthermore, this USV model solves its subtask using the greedy algorithm (Mahmud et al. 2012). Namely, it repeatedly locates the nearest part of the pattern and moves there, in order to draw it. A simple example is depicted in Figure 6.

The planned path including the pattern drawing commands is passed to the Low-Level Control module.

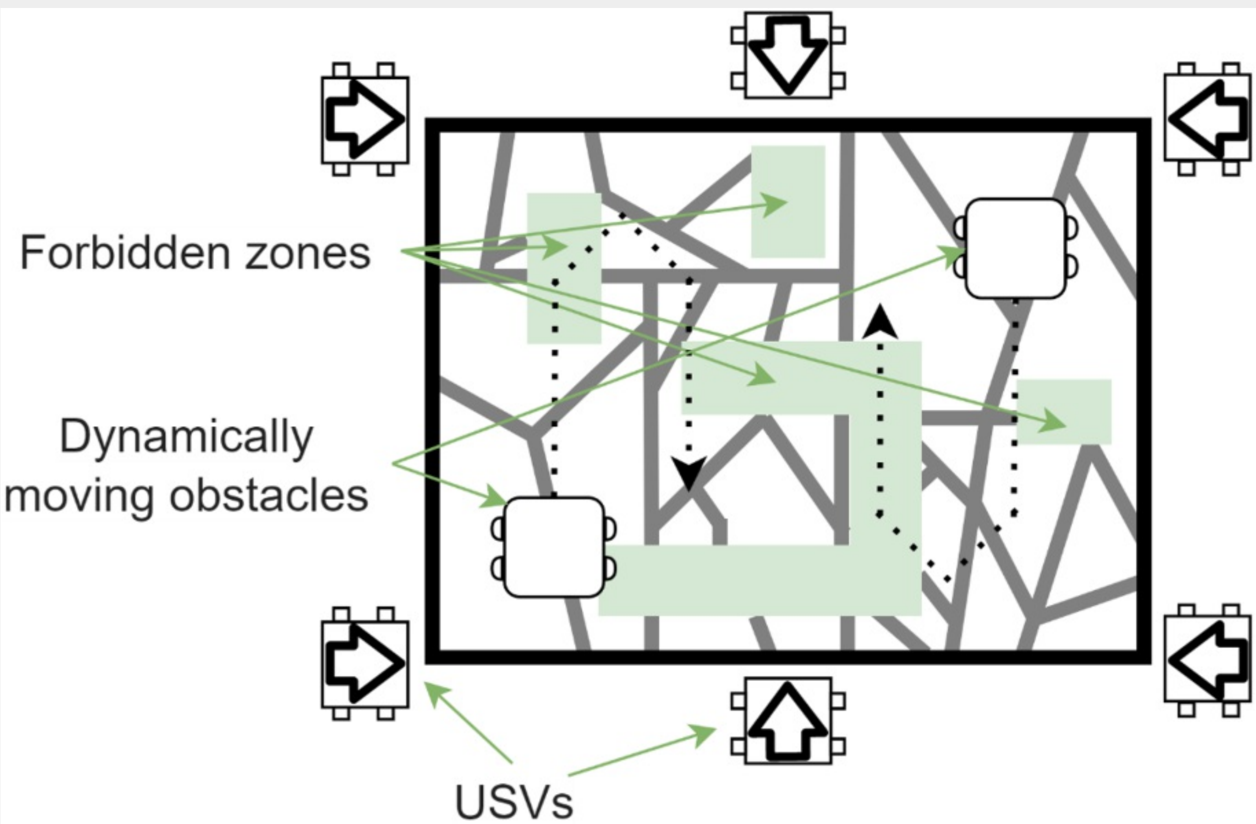
Global Planning Level Implementation

As mentioned above, this module includes a repeatedly executed optimization procedure based on a GA. The specific parts of the GA are proposed in order to achieve the defined aim in the demonstrative example. These are summarized below.

Chromosome Representation

Each chromosome is expected to represent a possible solution to the problem. In this case, the aim is especially to divide the pattern shown in Figure 5 into six parts, each to be fulfilled by one USV. Since obstacles may dynamically change their locations, and the USV may also be an obstacle to others, the obstacles are omitted from chromosome representation.

Figure 5. The aim for a fleet of USVs. Gray lines should be drawn by the USVs. Forbidden zones are prohibited for the USVs to enter. Dynamically moving obstacles follow the path defined by dotted arrows, but the path is not known for the FMCS. Initial positions and orientations of each USV are marked as arrows.



In this experiment, three competitive chromosome representations are considered. The first one divides the pattern using three lines. Each line is defined by two 2D points. Hence, the chromosome is described by 12 parameters. This representation is able to divide the pattern in up to 7 closed subpatterns, divided by straight boundaries. Since only six USVs are available, the last two subpatterns are merged. This representation cannot produce all possible shapes of sub-patterns, but it is expected to achieve a very time-efficient convergence.

The second variant of chromosome representation randomly divides the pattern into six subpatterns with pixel-wise resolution. Hence, each chromosome is composed of six subpatterns, where each subpattern is defined by a 2D array of zeros and ones. In each array, one means that this part of the pattern should be drawn by a particular USV. In order to define a chromosome, the pattern needs to be split up into indivisible segments, which are then disseminated randomly into six subpatterns. For the purposes of this experiment, in accordance with the representation used at the Local Planning Level, the pattern is divided into 80 × 60 segments, i.e., 80 columns and 60 rows. Contrary to the previous chromosome representation, this one can produce any possible type of assignment, including non-convex and disconnected subpatterns.

The third variant is similar to previous one. However, in this case, the assumption is expressed, that it is not advantageous to divide the pattern with pixel-wise resolution. On the contrary, it is expected that it would be suitable for the USV to finish any straight line, which had begun to be drawn. Hence, the pattern is allowed to be divided only at points where any line in the pattern begins or ends. Although this approach can also produce non-convex and disconnected sub-patterns, it is not able to divide straight lines in the pattern.

In Figure 7, a sample example of the chromosome representation approach is illustrated. In order to keep the clarity, the pattern is divided into only two subpatterns.

Fitness Evaluation

Each chromosome in the population needs to be evaluated by its fitness value in order to perform other genetic operators, such as selection and elitism. For this purpose, a world representation including predictions of the dynamically moving obstacles, and the models of the USVs need to be designed. Additionally, these models have to be computationally simple enough, since they are intended to be used for a large number of fitness evaluations during the genetic algorithm process. In this demonstrative example, the models are adopted from the Local Planning Level. Therefore, the number of movement steps is determined for each USV to fulfill its subtask.

Since six USVs are assigned with their subtask in this study, it is necessary to unite the costs of each USV. For this purpose, the costs of the slowest USV are considered as the defining characteristic. In other words, the total number of steps to solve the global task is minimized. Hence,

$$fitness = \max(costs_i), i = 1, 2, \ldots, 6,$$

where $costs_i$ is the number of steps of the USV_i performed to solve its subtask. Lower fitness is better, in this case. Apparently, different approaches such as minimal variance or sum of costs, may be implemented. However, if the sum of the steps of all the USVs were chosen instead of the maximum, the result could be in the worst case, only one active USV, while the other USVs would remain inactive.

Note that the implemented world representation is a very simple abstraction of the real world, and the fitness function value obtained is therefore only an estimate of the true cost (see Figure 6). However, it should be stressed that this value is used to optimally assign sub-tasks to individual USVs, not to predict or determine actual costs.

Figure 6. The world representation and demonstration of the greedy algorithm. In this situation, the USV (arrow), moves to [4, 6], turns right, moves to [4, 7], turns left, moves to [3, 7], [2, 7], [1, 7], turns left, moves to [1, 6], [1, 5], [1, 4], [1, 3], and [1, 2] to solve the subtask.

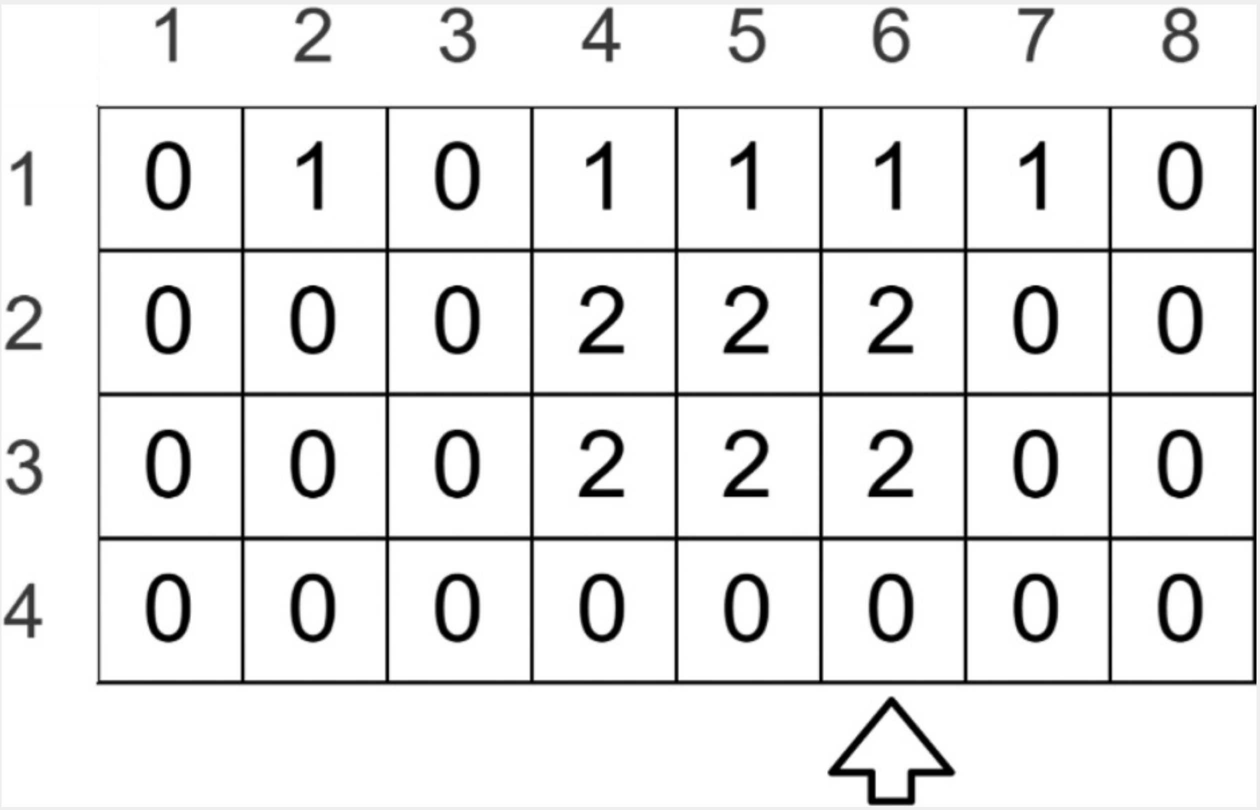
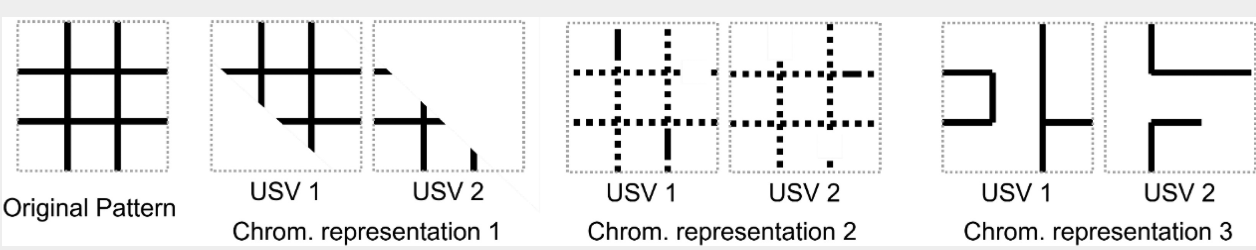


Figure 7. Considered variants of the task assignment according to the chromosome representation. The left image represents the pattern to be drawn. Three pairs of divided images represent considered variants of the chromosome representation for two USVs.



Selection

Selection is a mechanism, where individual chromosomes are selected for breeding. Tournament selection is chosen in this case. Specifically, *n* tournaments are arranged, where *n* is the number of chromosomes in the population. Two chromosomes, randomly taken from the population, participate in each tournament, and the winner of each tournament is selected for breeding. Besides, the best chromosome in the current population is also directly replicated into an offspring population.

Crossover

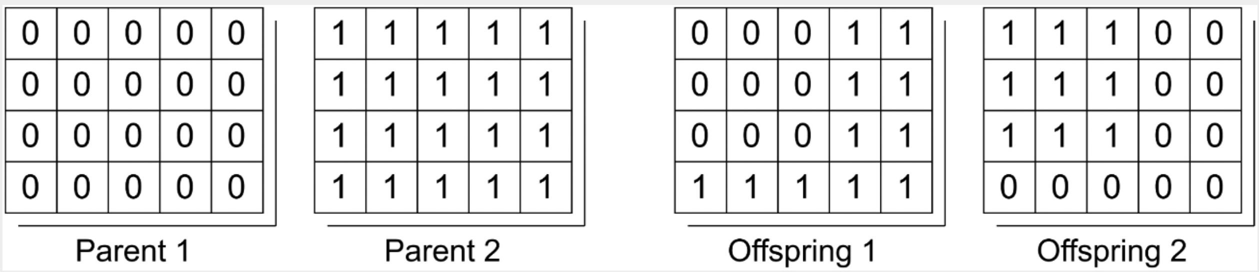
Crossover is a genetic operator used to combine two selected chromosomes to provide a pair of offspring chromosomes. In this experiment, the chromosomes are represented either by 12 float parameters (three lines for the first variant of the chromosome representation) or by six arrays of zeros and ones. For the former possibility, a blended crossover with a random alpha is applied. Specifically, two parents with 12 parameters $x_{i,1}^p, x_{i,2}^p$ produce two offspring chromosomes, as follows.

$$x_{i,1}^o = x_{i,1}^p - \alpha\left(x_{i,2}^p - x_{i,1}^p\right), x_{i,2}^o = x_{i,2}^p - \alpha\left(x_{i,2}^p - x_{i,1}^p\right), i = 1, 2, \ldots, 12,$$

where $x_{i,1}^o, x_{i,2}^o$ are parameters of offspring chromosomes, and $\alpha \in [0.25, 0.75]$ is a random parameter.

Considering the latter possibility, where the chromosomes are represented by six arrays of zeros and ones, a multi-point crossover is selected. The number of crossover points is chosen randomly between one and five. In order to keep the legitimacy of the offspring chromosomes, the crossover position needs to be kept across all six arrays. A demonstrative example of the crossover mechanism is shown in Figure 8.

Figure 8. Offspring mechanism for the chromosomes represented by six arrays of zeros and ones. Here, only one crossover point with the position [3, 3] is demonstrated.



Analogously to the initialization mechanism, either a totally random position of a crossover point, or only crossover points situated at positions where any line in the pattern begins or ends, are considered.

Mutation

In order to maintain genetic diversity within the population, a mutation operator is implemented. In this study, a mutation on each chromosome with 5% probability is implemented. For the chromosomes defined as an array of 12 float numbers, the mutation randomly generates one of 12 parameters. For the chromosomes represented as six arrays of zeros and ones, a rectangle of random size is swapped between two selected arrays.

Additionally, the diversity of the population is further increased by replacing 5% of the worst chromosomes with the freshly initiated ones.

Results

In order to demonstrate the proposed FMCS framework, a set of experiments defined above was performed. Specifically, four distinctive experiments were carried out and evaluated according to the particular combination of chromosome representation, initialization, and crossover in the Global planning level.

Note that the diagram in Figure 3 was strictly followed during the experiments, and each session of the genetic algorithm-based optimization procedure was executed with the following parameters: 100 generations, 150 individual chromosomes in each generation, mutation probability equal to 0.05, and 5% of the worst individuals at the end of each generation replaced by freshly initialized chromosomes. Moreover, since the dynamically changing environment is considered in the presented FMCS framework, the reassignment of the subtasks is performed repeatedly with a defined period of time during the actual execution of the subtasks by the USVs. In other words, the optimization procedure in the Global Planning Level module is run repeatedly with evolved world representation. This included partial fulfillment of the task, different initial positions of the USVs, different positions of the obstacles, etc.

In this set of experiments, re-run was performed five times after every 50 steps, in terms of the abstraction of the world representation described in the Local Planning Level module.

The concatenated courses (five runs together) of the fitness function for the best chromosomes and for the mean chromosomes are shown in Figure 9. Moreover, the number of steps necessary to fulfill the task (in terms of the abstraction described above) was 390 for variant (a) in Figure 9, 610 for variant (b) in Figure 9, 363 for variant (c) in Figure 9, and 393 for variant (d) in Figure 9. The number of steps performed by each USV to fulfill the task for every variant is shown in Table 1. To suggest a visualization of how the individual USVs performed the task, the paths of the three selected vehicles are shown in Figure 10. This figure corresponds to Figure 5. Visualization of all six vehicles would make the given figure unbearably cluttered. Nevertheless, it can be observed that all USVs effectively cover the required pattern and avoid the forbidden zones. Although the moving obstacles are not visualized in the figure for the purpose of clarity, there was also no collision between the USVs and the moving obstacles.

Figure 9. Fitness courses for (a) chromosome representation using three lines, (b) chromosome representation by six matrices with pixel-wise resolution, (c) and (d) chromosome representation by six matrices, the division is made at points, where the straight lines in the pattern begin or end. In (c) the crossover points are totally random, while in (d), the crossover points are situated at positions, where the straight lines in the pattern begin or end.

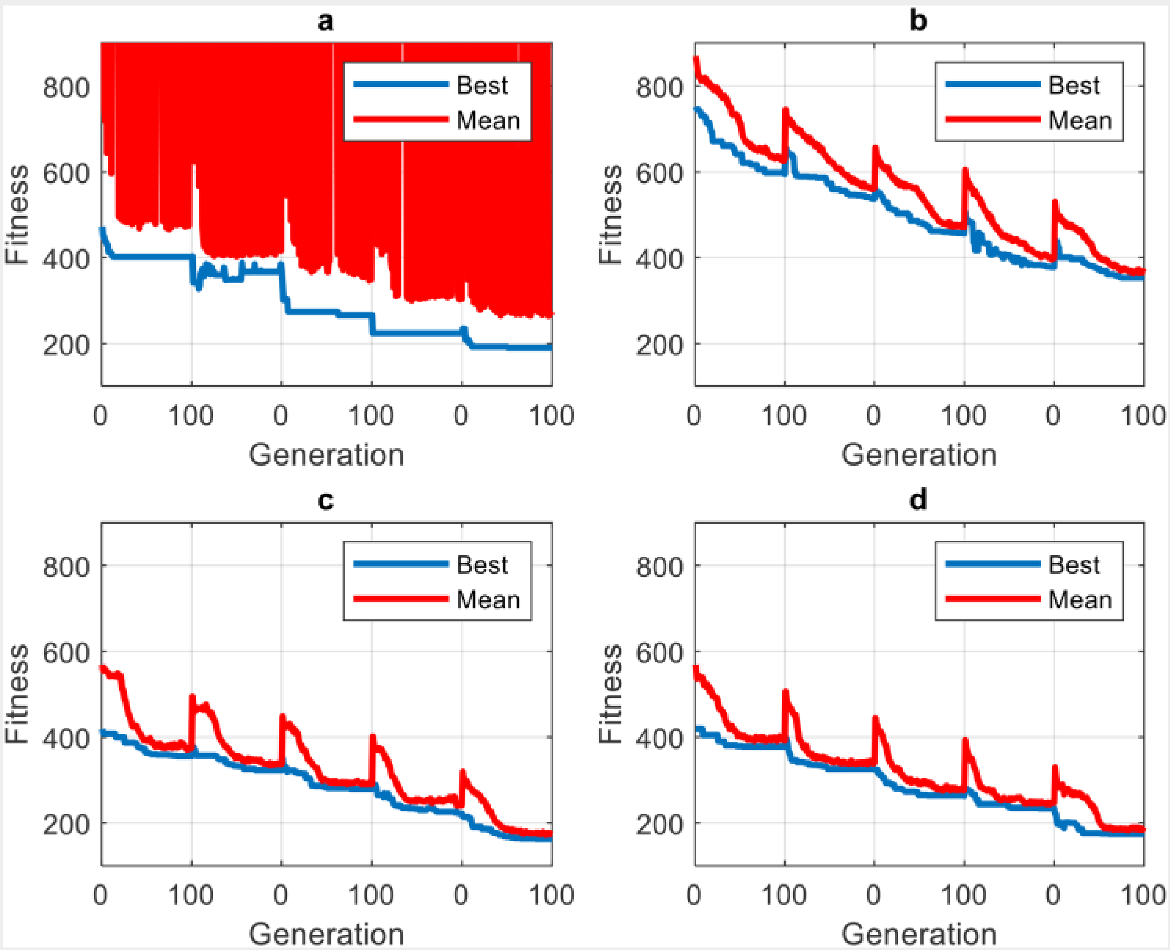
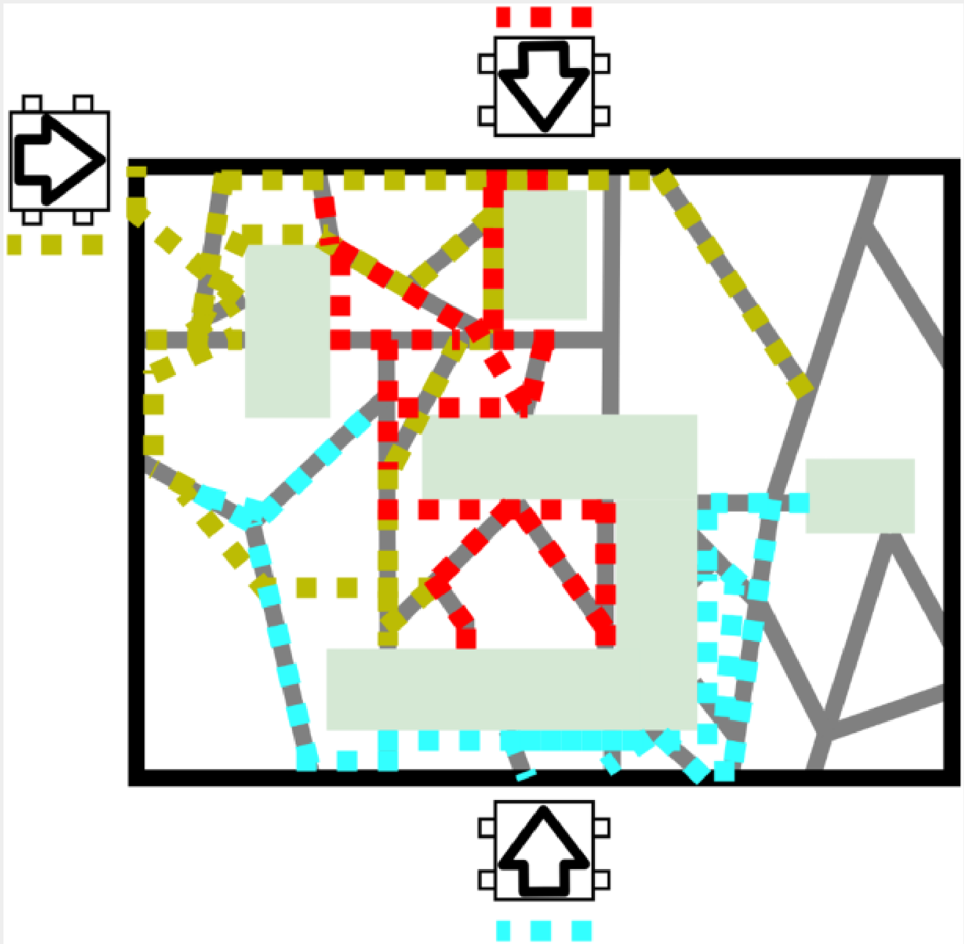


Figure 10. Visualization of the path of three selected USVs to fulfill their subtasks.



Note: The table layout displayed in ‘Edit’ view is not how it will appear in the printed/pdf version. This html display is to enable content corrections to the table. To preview the printed/pdf presentation of the table, please view the ‘PDF’ tab.

Table 1. The number of steps necessary to fulfill the task for each USV.

Variant	USV1	USV2	USV3	USV4	USV5	USV6
a	367	390	328	354	384	334
b	538	546	610	603	528	550

c	359	353	363	323	344	363
d	373	374	351	393	346	359

Place the cursor position on table column and click 'Add New' to add table footnote.

Discussion

Looking at Figure 9, it is apparent that the Global Planning Level using the division with three lines provides the most suitable convergence speed of the best individual. On the other hand, this representation often leads to results that are not feasible to plan in the Local Planning Level Module (see the fluctuations of the mean chromosome fitness). This behavior is particularly inappropriate when using lower numbers of individuals and generations.

Chromosomes represented by six matrices with pixel-wise resolution give the worst results in terms of the fitness function value and the convergence rate. Although this is the most generally defined GA session, it is obviously not suitable for the considered task.

The remaining two approaches deliver similar results. In this study, these methods converge more slowly than the first approach, but there are no individuals in the population unable to complete the task, which would be beneficial especially when using a smaller number of individuals.

These observations are, of course, strongly dependent on the specific task and experimental setup. The family of GAs allows significantly wider options, and it is possible that with a differently chosen initialization method, crossover operator, or mutation approach, the resultant findings would be different. However, the purpose of this set of experiments was to demonstrate the implementation of the FMCS framework rather than to find the optimal variant of the genetic algorithm for the considered problem. It should be pointed out that the presented framework inherently allows significantly more complex tasks for the USV fleet to be solved. These tasks include different types of USVs with special abilities, multiple types of areal restrictions and obstacles (including different restrictions for individual USVs), cooperation of multiple USVs to solve their subtasks, energy consumption optimization, etc.

Conclusion

Task assignment algorithms play a crucial role in unmanned vehicle fleet operations for proper cooperation and coordination, and they definitely need further research. In this study, a genetic algorithm-based approach to task assignment for a fleet of unmanned surface vehicles is introduced. Specifically, a variant of a fleet management control system, based on three levels of control, is examined. The top planning level is based on the specifically adapted genetic algorithm with a custom technique for chromosome representation. This level of control divides the global task into subtasks, and assigns these subtasks to individual unmanned vehicles (or groups of them), while utilizing various types of vehicles with special abilities, multiple types of areal restrictions and obstacles, different restrictions for individual vehicles, cooperation of multiple vehicles to solve their subtasks, energy consumption, etc. The possibility of applying the presented system is demonstrated on the pattern drawing problem in an environment with forbidden zones and dynamically moving obstacles using a fleet of six unmanned surface vehicles.

Future research will focus on establishing a formal protocol for communication between the different levels of control within the framework and other modules. Another goal is to create a comprehensive test polygon for real-world verification of the framework implementation.

Disclosure Statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this study.

Note: this Edit/html view does not display references as per your journal style. There is no need to correct this. The content is correct and it will be converted to your journal style in the published version.

References

Andreasson, H., A. Bouguerra, M. Cirillo, D. N. Dimitrov, D. Driankov, L. Karlsson, A. J. Lilienthal, F. Pecora, J. P. Saarinen, A. Sherikov, et al. 2015. Autonomous transport vehicles: Where we are and what is missing. *IEEE Robotics & Automation Magazine* 22 (1):64–75. doi: 10.1109/MRA.2014.2381357.

Arafat, M. Y., and S. Moh. 2018. Location-aided delay tolerant routing protocol in UAV networks for post-disaster operation. *IEEE Access* 6:59891–906. doi: 10.1109/ACCESS.2018.2875739.

Arafat, M. Y., and S. Moh. 2019. Localization and clustering based on swarm intelligence in UAV networks for emergency communications. *IEEE Internet of Things Journal* 6 (5):8958–76. doi: 10.1109/JIOT.2019.2925567.

Bailon-Ruiz, R., A. Bit-Monnot, and S. Lacroix. 2022. Real-time wildfire monitoring with a fleet of UAVs. *Robotics and Autonomous Systems* 152:104071. doi: 10.1016/j.robot.2022.104071.

Comasolivas, R., J. Quevedo, T. Escobet, A. Escobet, and J. Romera. 2015. Low level control of an omnidirectional mobile robot. 2015 23rd Mediterranean Conference on Control and Automation (MED), 1160–1166. IEEE. doi: 10.1109/MED.2015.7158912.

Dolezel, P., D. Stursa, D. Honc, J. Merta, V. Rozsivalova, L. Beran, and I. Hora. 2021. Counting livestock with image segmentation neural network. In *15th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2020)*, ed. Á. Herrero, C. Cambra, D. Urda, J. Sedano, H. Quintián, and E. Corchado, 237–44. Springer International Publishing. doi: 10.1007/978-3-030-57802-2_23.

Dvorak, M., P. Dolezel, D. Stursa, and M. Chouai. 2022. Guidance of unmanned surface vehicle fleet using genetic algorithm-based approach. In *16th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2021)*, ed. H. Sanjurjo González, I. Pastor López, P. García Bringas, H. Quintián, and E. Corchado, (187–97. Springer International Publishing. doi: 10.1007/978-3-030-87869-6_18.

Goldberg, D. E., and J. H. Holland. 1988. Genetic algorithms and machine learning. *Machine Learning* 3 (2/3):95–9. doi: 10.1023/A:1022602019183.

Guney, M. A., and I. A. Raptis. 2021. Dynamic prioritized motion coordination of multi-AGV systems. *Robotics and Autonomous Systems* 139:103534. doi: 10.1016/j.robot.2020.103534.

Guo, H., Z. Mao, W. Ding, and P. Liu. 2019. Optimal search path planning for unmanned surface vehicle based on an improved genetic algorithm. *Computers & Electrical Engineering* 79:106467. doi: 10.1016/j.compeleceng.2019.106467.

Han, S., C. Fan, X. Li, X. Luo, and Z. Liu. 2021. A modified genetic algorithm for task assignment of heterogeneous unmanned aerial vehicle system. *Measurement and Control* 54 (5–6):994–1014. doi: 10.1177/00202940211002235.

Lee, S. Y., S. R. Han, and B. D. Song. 2022. Simultaneous cooperation of Refrigerated Ground Vehicle (RGV) and Unmanned Aerial Vehicle (UAV) for rapid delivery with perishable food. *Applied Mathematical Modelling* 106:844–66. doi: 10.1016/j.apm.2022.02.024.

Lee, S., S. Kim, and S. Moon. 2022. Development of a car-free street mapping model using an integrated system with unmanned aerial vehicles, aerial mapping cameras, and a deep learning algorithm. *Journal of Computing in Civil Engineering* 36 (3):04022003. doi: 10.1061/(ASCE)CP.1943-5487.0001013.

Liu, X., X. Du, X. Zhang, Q. Zhu, and M. Guizani. 2019. Evolution-algorithm-based unmanned aerial vehicles path planning in complex environment. *Computers & Electrical Engineering* 80:106493. doi: 10.1016/j.compeleceng.2019.106493.

Mahmud, M. S., U. Sarker, M. M. Islam, and H. Sarwar. 2012. A greedy approach in path selection for DFS based maze-map discovery algorithm for an autonomous robot. 2012 15th International Conference on

Computer and Information Technology (ICCIT), 546–550. IEEE. doi: 10.1109/ICCITechn.2012.6509798.

Miao, R., L. Wang, and S. Pang. 2022. Coordination of distributed unmanned surface vehicles via model-based reinforcement learning methods. *Applied Ocean Research* 122:103106. doi: 10.1016/j.apor.2022.103106.

Moysiadis, V., N. Tsolakis, D. Katikaridis, C. G. Sørensen, S. Pearson, and D. Bochtis. 2020. Mobile robotics in agricultural operations: A narrative review on planning aspects. *Applied Sciences* 10 (10):3453. doi: 10.3390/app10103453.

Park, J., S. Kim, G. Noh, H. Kim, D. Lee, and I. Lee. 2021. Mission planning and performance verification of an unmanned surface vehicle using a genetic algorithm. *International Journal of Naval Architecture and Ocean Engineering* 13:575–84. doi: 10.1016/j.ijnaoe.2021.07.002.

Parker, L. E. 2009. Multiple mobile robot teams, path planning and motion coordination in. In *Encyclopedia of complexity and systems science*, ed. R. A. Meyers, 5783–800. New York: Springer New York. doi: 10.1007/978-0-387-30440-3_344.

Poudel, S., and S. Moh. 2022. Task assignment algorithms for unmanned aerial vehicle networks: A comprehensive survey. *Vehicular Communications* 35:100469. doi: 10.1016/j.vehcom.2022.100469.

Seenu, N., R. M. Kuppan Chetty, M. M. Ramya, and M. N. Janardhanan. 2020. Review on state-of-the-art dynamic task allocation strategies for multiple-robot systems. *Industrial Robot: The International Journal of Robotics Research and Application* 47 (6):929–42. doi: 10.1108/IR-04-2020-0073.

Sharma, K. R., F. Dusek, and D. Honc. 2017. Comparitive study of predictive controllers for trajectory tracking of non-holonomic mobile robot. 2017 21st International Conference on Process Control (PC), 197–203. IEEE. doi: 10.1109/PC.2017.7976213.

Shelkamy, M., C. M. Elias, D. M. Mahfouz, and O. M. Shehata. 2020. Comparative analysis of various optimization techniques for solving multi-robot task allocation problem. 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), 538–543. IEEE. doi: 10.1109/NILES50944.2020.9257967.

Singh, Y. 2019. Cooperative swarm optimisation of unmanned surface vehicles. Ph.D. thesis, University of Plymouth.

Vivaldini, K. C. T., L. F. Rocha, M. Becker, and A. P. Moreira. 2015. Comprehensive review of the dispatching, scheduling and routing of AGVs. In *CONTROLO'2014 – Proceedings of the 11th Portuguese Conference on Automatic Control*, ed. A. P. Moreira, A. Matos, and G. Veiga, 505–14. Cham: Springer International Publishing. doi: 10.1007/978-3-319-10380-8_48.

Zhang, M., W. Li, M. Wang, S. Li, and B. Li. 2022. Helicopter–UAVs search and rescue task allocation considering UAVs operating environment and performance. *Computers & Industrial Engineering* 167:107994. doi: 10.1016/j.cie.2022.107994.

Author Query

1. **Query** [AQ0] : Please review the table of contributors below and confirm that the first and last names are structured correctly and that the authors are listed in the correct order of contribution. This check is to ensure that your names will appear correctly online and when the article is indexed.

Sequence	Prefix	Given name(s)	Surname	Suffix
1		Miroslav	Dvorak	
2		Petr	Dolezel	
3		Dominik	Stursa	
4		Mohamed	Chouai	

Response by Author: "Ok"

2. **Query** [AQ1] : Please check the author names, affiliations, and corresponding details have been typeset correctly and correct if this is inaccurate.
Response by Author: "Ok"