An integrated multi-population genetic algorithm for multi-vehicle task assignment in a drift field

Bai, Xiaoshan; Yan, Weisheng; Ge, Shuzhi Sam; Cao, Ming

Published in:
Information Sciences

DOI:
10.1016/j.ins.2018.04.044

An integrated multi-population genetic algorithm for multi-vehicle task assignment in a drift field[☆]

Xiaoshan Bai[a,c,*], Weisheng Yan[a], Shuzhi Sam Ge[b], Ming Cao[c]

[a] *School of Marine Science and Technology, Northwestern Polytechnical University, 127 West Youyi Road, Xi'an, 710072, China*

[b] *Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576*

[c] *Faculty of Science and Engineering, University of Groningen, Groningen 9747 AG, The Netherlands*

Abstract

This paper investigates the task assignment problem for a team of autonomous aerial/marine vehicles driven by constant thrust and maneuvering in a planar lateral drift field. The aim is to minimize the total traveling time in order to guide the vehicles to deliver a number of customized sensors to a set of target points with different sensor demands in the drift field. To solve the problem, we consider together navigation strategies and target assignment algorithms; the former minimizes the traveling time between two given locations in the drift field and the latter allocates a sequence of target locations to each vehicle. We first consider the effect of the weight of the carried sensors on the speed of each vehicle, and construct a sufficient condition to guarantee that the whole operation environment is reachable for the vehicles. Then from optimal control principles, time-optimal path planning is carried out to navigate each vehicle from an initial position to its given target location. Most importantly, to assign the targets to the vehicles, we combine the virtual coding strategy, multiple offspring method, intermarriage crossover strategy, and the tabu search mechanism to obtain a co-evolutionary multi-population genetic algorithm, short-named CMGA. Sim-

[☆] This work is supported by the National Natural Science Foundation of China (Grant Nos. 51579210 and 61633002) and the China Scholarship Council (Grant No. 201506290081).

[*] Corresponding author

Email address: xiaoshan.bai@rug.nl (Xiaoshan Bai)

ulations on sensor delivery scenarios in both fixed and time-varying drift fields are shown to highlight the satisfying performances of the proposed approach against popular greedy algorithms.

Keywords: Task assignment; Time-optimal path planning; Drift field; Autonomous vehicles; Multi-population genetic algorithm

1. Introduction

The task assignment problem is usually referred to how to assign a number of agents to perform a set of tasks [10] while minimizing the total traveling distance [34] or time [29] of the agents. It arises in a range of applications of multi-agent systems, such as unmanned aerial vehicles (UAVs) [22, 35], unmanned marine vehicles [5, 24], ground vehicles/robots [4, 6] and sensor networks [9, 27]. People often use the metric matrices to specify the costs of assigning each target to each agent, which are usually independent of how the assigned tasks are accomplished by the agents [7, 12, 19, 20, 28]. Considering the task priority and vehicle loading capacity, Shima et al. [28] utilized a genetic algorithm (GA) to assign tasks to multiple UAVs. Dahl et al. [7] developed a vacancy chain scheduling to formalize robot interactions for the multi-robot task assignment. Considering UAVs' turning radius constraint, Edison and Shima [11] integrated GAs with a Dubins car model to complete target assignment for multiple UAVs, for which by properly discretizing the visitation angle of each target into a finite set, the integrated GAs achieved suboptimal trajectory assignment. Based on coarsening and partitioning the standard utility matrix, Liu and Shell [19] utilized both centralized and decentralized approaches to realize large-scale task allocation. For solving heterogeneous multi-vehicle routing problems, gossip algorithms were used to minimize the maximum execution time of robots [12].

When an agent's traveling cost between two prescribed positions is affected by external time-varying disturbance or by vehicle's current workload as in a cargo delivery task, the metric matrix and the task assignment are no longer independent of each other. Consequently, the multi-agent task assignment problem consists of two coupled sub-problems, namely how to assign subtasks as sequences of target locations to individual agents and how to navigate an agent from its initial position to a target location optimally. There are some related research works [3, 15, 16, 32, 36], which considered both the target location assignment and path planning for the employed robots. By simply requiring moving in straight lines between prescribed positions, Han and Chung [15] employed an autonomous underwater vehicle (AUV) to optimally visit several target points taking into account the constraints of ocean currents and obstacles. Zhu et al. [36] employed multiple AUVs to visit several target points by using the velocity synthesis approach to enable each AUV to reach its targets along the shortest path in the time-varying (in a discrete time scale) 3-D underwater environment and using the self-organizing map neural network to realize the multi-AUV target point assignment. To minimize the probability of being detected, a variational dynamic programming method was applied to find a locally optimal path for a vehicle flying through a field of detectors [16].

In our previous work [13], a GA was combined with Lyapunov's direct-controller for motion control of a single-link flexible robot. In [33], the multi-AUV sensor delivery task was performed in temporally piece-wise constant ocean currents aiming at minimizing the total traveling time along the shortest paths, assuming constant vehicle speed. Time-optimal coverage control of multi-vehicle in a drift field was studied in [37] where the time-optimal paths were generated over discrete times. To deal with a much more challenging scenario, in this paper we investigate the task assignment problem for which several vehicles with sensor loading capacities need to deliver a number of sensors to a set of dispersed target points in a drift field as energy-efficiently as possible. Our main contributions are as follows. Firstly, based on the accessible area analysis and optimal control theory, a path planning algorithm is designed which can generate the time-optimal path for a vehicle traveling between two given positions in a drift field. Here the effect of the workload of the weight of the carried sensors on the speed of the vehicles is considered, which leads to a more realistic model for the general cargo delivery problem. Secondly, we propose a co-evolutionary multi-population genetic algorithm (CMGA) that integrates the intermarriage crossover mechanism with a multiple offspring strategy, and utilizes a virtual coding technique together with a tabu mechanism. The algorithm can construct sequences of target points for individual vehicles efficiently. Lastly, a novel solution to the challenging sensor delivery task assignment problem is obtained after incorporating the optimal navigation law and the CMGA algorithm.

The rest of this paper is organized as follows. In Section 2, the formulation of the task assignment problem is given for multiple vehicles in a drift field. Section 3 presents the path planning algorithm which generates the optimal navigation control law and in Section 4 the task assignment algorithm CMGA is discussed. We present the simulation results in Section 5 and conclude the paper in Section 6.

2. Problem formulation

2.1. Problem description

A fleet of aerial/marine vehicles located at a base station needs to deliver customized sensors to a set of n targets that are randomly dispersed in a planar drift field while trying to minimize the total traveling time. Each target demands a specific number of sensors to be delivered, and the vehicles are constrained by their limited sensor loading capacity as each sensor has a certain weight. The net speed of each vehicle is affected by the speed of the currents in the drift field and the number of sensors it carries.

2.2. Formulation as an optimization problem

We use the vector $\vec{v}_c = [v_{cx}(x, y, t), v_{cy}(x, y, t)]^T$ to describe the drift velocity of the field with respect to some coordinate system fixed to the ground. We assume that the vehicles are driven by constant thrust, and consequently their velocity $\vec{v}$ obtains its maximum speed v_{max} relative to the field when carrying no sensors [25, 28, 30]. The influence of the number of carried sensors on the speed of the vehicle is assumed to be given by

$$v(t) = v_{max} - s_r(t)/L, \tag{1}$$

where L is a positive integer describing the sensor loading capacity and $s_r(t)$ is the number of sensors that the vehicle carries at time t. Since the dimension of the drift field is significantly larger than the vehicles' size, we assume that the vehicles are not subject to turning ratio constraints.

The kinematics of each vehicle are

$$\dot{x} = v(t)\cos\psi + v_{cx}(x, y, t), \qquad \dot{y} = v(t)\sin\psi + v_{cy}(x, y, t), \tag{2}$$

where ψ is the vehicle's navigation angle. We label the targets by $1, \ldots, n$ and the base station by 0 and let $\mathcal{V} = \{0, 1, \ldots, n\}$ be the set of these indices. For each $i \in \mathcal{V} \setminus \{0\}$, let s_i be the number of sensors that target i demands. We assume that $s_i \leq L$ for all i so that each target can always be served by a single vehicle. We require that each target can be served by only one vehicle, so the maximum required number of vehicles is n. We assume that there are enough number of sensors at the base station to be delivered. We use $\mathcal{A} = \{1, \ldots, m\}$ to denote the indices of the minimal set of vehicles. Then $m \leq n$, and given the sensor demand s_i, m is at least

$$\left\lceil \frac{1}{L} \sum_{i \in \mathcal{V} \setminus \{0\}} s_i \right\rceil, \tag{3}$$

where for a positive number a, the ceiling function $\lceil a \rceil$ returns the smallest integer that is greater than or equal to a.

For any pair of $i, j \in \mathcal{V}$, let t_{ij} denote the minimum time needed for a vehicle to travel from i to j using a properly designed navigation control. Obviously $t_{ii} = 0$ for each i. Let $\sigma_{ijk} : \mathcal{V} \times \mathcal{V} \times \mathcal{A} \to \{0, 1\}$ be the path-planning mapping that maps the indices $i, j \in \mathcal{V}$ of the starting and ending locations and $k \in \mathcal{A}$ of the kth vehicle to a binary value, which equals one if and only if it is planned that vehicle k travels between distinct locations i and j. So $\sigma_{iik} = 0$ for all $i \in \mathcal{V}$ and $k \in \mathcal{A}$. To minimize the totaled traveling time by all the vehicles is to minimize $\sum_{i \in \mathcal{V}, j \in \mathcal{V}, k \in \mathcal{A}} \sigma_{ijk} t_{ij}$. However, the minimization must be done subject to a number of constraints to ensure the validity of the path planning and task assignment. To describe some of the constraints clearly, we need to introduce the task-assignment mapping $\mu_{ik} : \mathcal{V} \times \mathcal{A} \to \{0, 1\}$ that maps the indices $i \in \mathcal{V}$ of the ith target and $k \in \mathcal{A}$ of the kth vehicle to a binary value, which equals one if and only if it is vehicle k that serves target i. We first present the overall constrained minimization problem below and then explain one by one what each constraint means. We aim to minimize

$$f = \sum_{i \in \mathcal{V}, j \in \mathcal{V}, k \in \mathcal{A}} \sigma_{ijk} t_{ij} + M \sum_{k \in \mathcal{A}} \max \left\{ \sum_{i \in \mathcal{V} \setminus \{0\}} s_i \mu_{ik} - L, 0 \right\}, \tag{4}$$

subject to

$$\begin{aligned}
\sum_{j \in \mathcal{V}, k \in \mathcal{A}} \sigma_{0jk} &= m; && (5)\\
\sum_{i \in \mathcal{V}, k \in \mathcal{A}} \sigma_{i0k} &= m; && (6)\\
\sum_{k \in \mathcal{A}} \mu_{ik} &= 1, \qquad \forall i \in \mathcal{V} \setminus \{0\}; && (7)\\
\sum_{j \in \mathcal{V}} \sigma_{ijk} &= \mu_{ik}, \qquad \forall i \in \mathcal{V},\ \forall k \in \mathcal{A}; && (8)\\
\sum_{i \in \mathcal{V}} \sigma_{ijk} &= \mu_{jk}, \qquad \forall j \in \mathcal{V},\ \forall k \in \mathcal{A}. && (9)
\end{aligned}$$

The objective function (4) seeks to minimize the total traveling time of the vehicles, where M is a large punishing factor used to guarantee that the number of sensors delivered by each vehicle is always below the vehicle's loading capacity. Constraints (5) and (6) ensure that all m vehicles leave from and return to the base station; (7) requires that each target is served and only served by one vehicle; and (8) and (9) ensure that it is the same vehicle arrives at and leaves from a target.

Remark 1. *If ignoring the effect of the carried sensors and the field currents on the speed of the vehicles, the sensor delivery task assignment problem just presented reduces to the extensively studied vehicle routing problem (VRP) [31], which is well known to be NP-hard.*

After formulating the sensor delivery task as a constrained minimization problem, we present in the following section a component of the path planning that is critical for solving the overall optimization problem.

3. Path planning algorithm given the starting and target positions

To plan the optimal path that minimizes the traveling time in a given field with currents, we first look at the accessible region of a vehicle starting from an arbitrary position. Then using optimal control theory, we plan the path and construct the navigation rule that guides a vehicle to travel between two given positions following the path using the minimum time.

3.1. Accessible region analysis

As before, $\vec{v}_c$ is used to denote the velocity of the current, which may change with location and time; its amplitude is v_c. As in (2), the vehicle's velocity relative to the field is $\vec{v}$ with the amplitude $v(t)$, and the vehicle's maximum speed is $v_{\max}$ which is only possible when the vehicle is free of load. Similarly, we use $\vec{v}_n$ to denote the vehicle's net velocity with amplitude v_n. The relative speed of the vehicle $v(t)$ is affected by the vehicle's load: since the speed of the vehicle relative to the field takes the possible values $v(t) \in \{v_{\max}, v_{\max} - \frac{1}{L}, v_{\max} - \frac{2}{L}, \ldots, v_{\max} - 1\}$, the minimum net speed

$$v_{n,\min} = v_{\max} - 1 - v_{c,\max},$$

where $v_{c,\max}$ is the maximum speed of the current at all locations in the field over all time. Obviously, the vehicle can reach all locations of the field given enough time if $v_{n,\min} > 0$. For this reason, we make this standing assumption for the rest of the paper.

Assumption 1: *It holds for all locations of the field and all time that $v_{n,\min} > 0$.*

Consequently, with this assumption, each vehicle can travel from any given position to any given other target location and the traveling time depends on the path planned and the associated navigation rule, which will be discussed in the following subsection.

3.2. Optimal navigation law

We now show how to navigate a vehicle between any two given positions within minimum time.

Theorem 1. *Under Assumption 1, for a vehicle with kinematics (2), to travel with the minimum time between any given starting and ending positions of the drift field with the current velocity $\vec{v}_c$, the rate of change of the vehicle's optimal navigation angle ψ^* must satisfy*

$$\begin{aligned}\dot{\psi}^* &= -\frac{\partial v_{cx}(x,y,t)}{\partial y}\cos^2\psi^* + (\frac{\partial v_{cx}(x,y,t)}{\partial x} - \frac{\partial v_{cy}(x,y,t)}{\partial y})\sin\psi^*\cos\psi^* \\ &\quad + \frac{\partial v_{cy}(x,y,t)}{\partial x}\sin^2\psi^*. \end{aligned} \tag{10}$$

Proof. Let t_0 and t_f be the starting and finishing times respectively. Then to minimize the traveling time, is to minimize the objective function

$$J = \int_{t_0}^{t_f} dt = t_f - t_0.$$

Define the corresponding Hamiltonian to be

$$\begin{aligned} H(t,[x,y]^T,\lambda,\psi) &= 1 + \lambda^T[\dot{x},\dot{y}]^T \\ &= 1 + \lambda_1(v(t)\cos\psi + v_{cx}(x,y,t)) + \lambda_2(v(t)\sin\psi \\ &\quad + v_{cy}(x,y,t)), \end{aligned} \tag{11}$$

where $\lambda = [\lambda_1, \lambda_2]^T$ is the two-dimensional Lagrangian multiplier. From Pontryagin's minimum principle of variational analysis in optimal control theory [18, P188], it must be true that the optimal Lagrangian multiplier λ^* and the optimal navigation angle ψ^* satisfies

$$\dot{\lambda}^* = -\frac{\partial H}{\partial [x,y]^T} \tag{12}$$

$$0 = \frac{\partial H}{\partial \psi^*} \tag{13}$$

Since (13) holds for all $t \geq t_0$, the time derivative of its right-hand side must also be zero. So we have

$$\dot{\lambda}_1^* \sin\psi^* + \lambda_1^* \dot{\psi}^* \cos\psi^* = \dot{\lambda}_2^* \cos\psi^* - \lambda_2^* \dot{\psi}^* \sin\psi^*.$$

Combining with what can be obtained from (12)

$$\begin{array}{rcl} \dot{\lambda}_1^* & = & -\lambda_1^* \dfrac{\partial v_{cx}(x,y,t)}{\partial x} - \lambda_2^* \dfrac{\partial v_{cy}(x,y,t)}{\partial x} \\ \dot{\lambda}_2^* & = & -\lambda_1^* \dfrac{\partial v_{cx}(x,y,t)}{\partial y} - \lambda_2^* \dfrac{\partial v_{cy}(x,y,t)}{\partial y}, \end{array}$$

we have that when $t_f - t_0$ is minimized, (10) must hold under the optimal navigation control ψ^*. □

Because of Assumption 1, we know that a solution ψ, and thus the optimal solution ψ^*, always exist. Theorem 1 gives a necessary condition on $\dot{\psi}^*$; what remains to be determined is the initial orientation $\psi^*(0)$. After knowing $\psi^*(0)$ and $\dot{\psi}^*$, the optimal navigation angle $\psi^*(t)$, $t > 0$, can be determined through the integration of $\dot{\psi}^*$ over t. However, to determine $\psi^*(0)$ with the initial position $[x_0, y_0]^T$ and the finishing position $[x_f, y_f]^T$, one needs to solve the two-point boundary problem:

$$\begin{cases} x_f = x_0 + \int_{t_0}^{t_f} [v(t) \cos(\psi(0) + \int_{t_0}^{t} \dot{\psi}^* d\tau) + v_{cx}(x,y,t)] dt, \\ y_f = y_0 + \int_{t_0}^{t_f} [v(t) \sin(\psi(0) + \int_{t_0}^{t} \dot{\psi}^* d\tau) + v_{cy}(x,y,t)] dt. \end{cases} \tag{14}$$

The solution $\psi(0)$ to (14) in general can only be found numerically. For the computational example we will show later in the paper, because of the special structures of current velocity $[v_{cx}(x,y,t), v_{cy}(x,y,t)]^T$ of the drift field, some simplification of the computation can be done. In the next section, we discuss the task assignment algorithm.

4. Task assignment algorithm CMGA

Multi-population genetic algorithms (MGAs) have been applied before to solve the task assignment problem for UAVs [8, 14, 23]. The most important characteristic of MGAs is the subpopulation utilization where individuals in each subpopulation evolve themselves and different subpopulations exchange information by sharing the global best-performing individuals [2]. Based on the route information obtained from the path planing algorithm, we design the co-evolutionary multi-population genetic algorithm (CMGA) for the target point assignment. The CMGA consists of three components, which are population initialization, local evolution operators, and the global operators. Now we explain the algorithm in more details.

4.1. Virtual coding strategy

The chromosome encoding is the most critical part of deriving a generic algorithm. Inspired by [13], we encode the CMGA chromosomes as real (decimal) integer arrays based on the target number $\{1, ..., n\}$. To use one dimensional chromosome to represent one solution to the task assignment of multiple vehicles, a virtual coding strategy is introduced, which inserts $m-1$ marker gene, coded by $\{n+1, ..., n+m-1\}$, into the n target point genes, where as before $m \geq 1$ is the number of vehicles. Then an extended set $\mathcal{V}'$ can be constructed, where $\mathcal{V}' := \mathcal{V} \cup \{n+1, ..., n+m-1\}$ includes the indices of the $m-1$ additional virtual base stations. The optimal traveling time t_{ij} in (4) associated with each arc changes to

$$t'_{ij} = \begin{cases} t_{ij} & for\ i,\ j \in \mathcal{V} \setminus \{0\}, \\ t_{i0} & for\ i \in \mathcal{V} \setminus \{0\},\ j \in \mathcal{W}, \\ t_{0j} & for\ i \in \mathcal{W},\ j \in \mathcal{V} \setminus \{0\}, \\ M & for\ i,\ j \in \mathcal{W}, \end{cases} \tag{15}$$

where $\mathcal{W} := \{0\} \bigcup \{n+1, ..., n+m-1\}$ is the set of vertices associated with the base station, and M is a large punishing value used to distinguish the base station from the targets.

As a result, the length l_{en} of the integer array chromosome is

$$l_{en} = |\mathcal{V}| - 1 + m - 1 = |\mathcal{V}| + m - 2, \tag{16}$$

where $|\mathcal{V}|$ is the cardinality of the vertex set $\mathcal{V}$ and $|\mathcal{V}| - 1$ is the number of the target points.

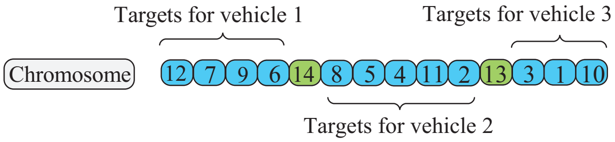

Fig. 1. Chromosome structure with 12 target points and 3 vehicles, where genes 13 and 14 are marker genes used as virtual base stations to separate the target points into three groups.

An example chromosome structure is presented in Fig. 1, which contains 12 target points and 3 vehicles. Firstly, the marker genes 13 and 14 separate the 12 target points into three gene segments where the target points within each segment is assigned to one vehicle. Moreover, the sequence of target points in one gene segment represents their serving sequence for one vehicle. Thus, each chromosome represents a candidate solution to the task assignment problem; for example, the routes of the three vehicles presented in Fig. 1 are vehicle 1: $0 \rightarrow 12 \rightarrow 7 \rightarrow 9 \rightarrow 6 \rightarrow 0$; vehicle 2: $0 \rightarrow 8 \rightarrow 5 \rightarrow 4 \rightarrow 11 \rightarrow 2 \rightarrow 0$; vehicle 3: $0 \rightarrow 3 \rightarrow 1 \rightarrow 10 \rightarrow 0$.

4.2. Population initialization

In the CMGA, N_{sub} subpopulations are utilized, each of which has the same number of chromosomes N_p. The initial chromosomes of each subpopulation are stochastically generated for which it is checked that in every chromosome each vehicle's sensor loading capacity is satisfied.

4.3. Fitness function

A fitness function evaluates a chromosome by assigning a numerical fitness value proportional to the chromosome's survival ability [21]. For chromosome j ($j = 1, ..., N_p$) in subpopulation i ($i = 1, ..., N_{sub}$), its fitness value $\bar{f}_{ij}$ is based on the objective f in (4):

$$\bar{f}_{ij} = 1/f. \tag{17}$$

From the initialization of the population, we get the initial target points' assignment and their serving sequence through distinguishing the marker genes in the chromosome, and thus compute f.

4.4. Local genetic operators

Now we discuss the genetic operators. We call those that exchange chromosome information within each subpopulation the *local* genetic operators, and call those between different subpopulations *global* genetic operators. The local operators consists of a selection operator, a crossover operator and a mutation operator.

4.4.1. Selection

Tournament selection is one of the most popular selection methods in genetic algorithms due to its efficiency and simplicity, which preserves gene diversity by giving a chance to all individuals to be selected [1, 26]. For the proposed multi-population genetic algorithm, we adopt the binary tournament selection that repetitively randomly chooses two chromosomes and selects the fitter one until the number of the chosen chromosomes reaches the initial population size.

4.4.2. Crossover

After the new generation is selected, a crossover operator takes action to execute the gene recombination. The basic idea is that mating parents pass their best genes to the offspring chromosomes with a high crossover probability P_c to improve the population survivability [21].

According to the classical two-point crossover mechanism [21], if a randomly generated value ranging from 0 to 1 is smaller than P_c, then the following crossover takes place. Firstly, two chromosomes having not undergone the crossover operation are randomly chosen to form a mating parent pair. Then, two crossover integer points cp_1, cp_2 and a receiving point rp, all ranging from 1 to l_{en}, are randomly produced for the gene exchange of the two parents as shown in Fig. 2. Different from the crossover operator of binary-coded chromosomes that only replaces the exchanged gene segments of the two mating

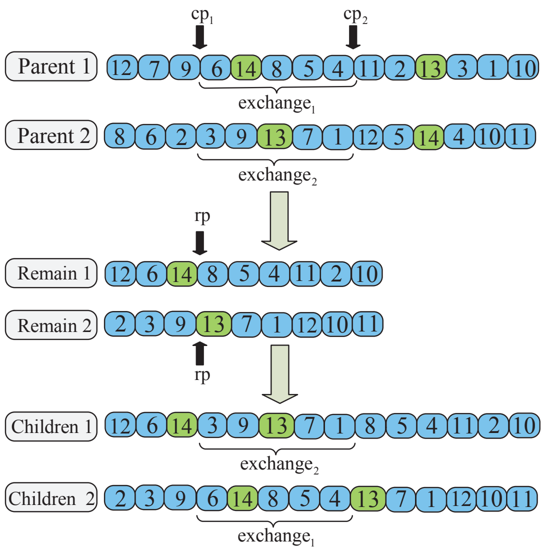

Fig. 2. Crossover operation I for parents with different chromosomes where cp_1 and cp_2 are crossover points and rp is the receiving point for the exchanged gene segment.

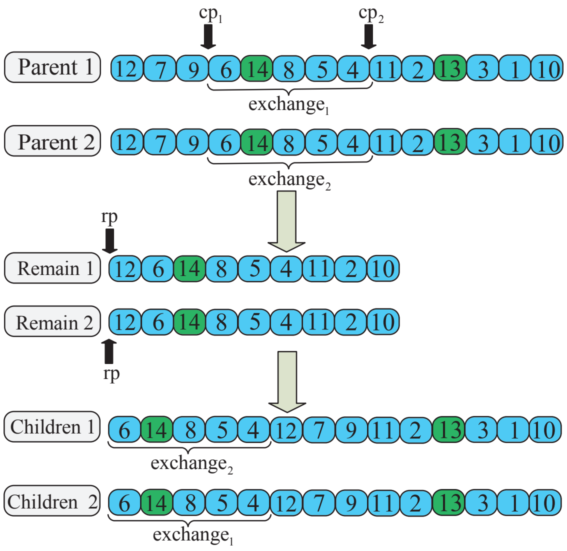

Fig. 3. Crossover operation II for parents with identical chromosomes: crossover points cp_1 and cp_2 are also randomly produced.

parents, the chromosomes coded here need to delete some genes before receiving the exchanged genes. The reason is that every target point should appear only once in a chromosome to ensure all the targets to be assigned and to avoid the same target being assigned to different vehicles.

This crossover strategy, however, makes no improvement if two mating parents have the same gene sequence. As a result, we adopt an crossover operator which can produce different offsprings even when two mating parents are identical. When two mating parents are chosen, a verifying operation is first done to determine whether the two parents are identical. If the two parents have different gene sequences, we use the crossover operator I illustrated in Fig. 2. Otherwise, the crossover operator II is employed which inserts the exchanged gene segment at the front of the chromosomes (see Fig. 3).

Inspired by the fact that many natural species have several offsprings for each mating parent pair in one generation, we propose a multiple offspring strategy for mating parents. To have a satisfying exploration of genes in the previous generation, the multiple offspring strategy gives each mating parent pair C_c crossover chances to produce more offsprings in one generation.

4.4.3. Mutation

Mutation is a genetic operator utilized to avoid premature convergence by preventing chromosomes stagnation at a local minimum [21]. For binary-coded chromosomes, the mutation operation is very simple since it only needs to change the value of the mutation gene from zero to one or vice versa. For integer-coded chromosomes, however, it is harder since every target point should appear only once in a chromosome.

As a result, we exchange the positions of two genes in a chromosome to undergo the mutation process. If a randomly generated value, ranging from 0 to 1, is smaller than the mutation probability P_m, two mutation positions m_1 and m_2 are randomly generated for one chromosome. The chromosome then exchanges the genes on these two sites, as presented in Fig. 4.

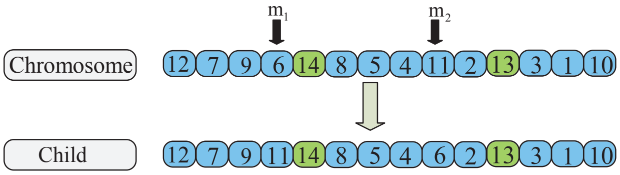

Fig. 4. Mutation operator for a chromosome: randomly generated mutation points m_1 and m_2 determine which genes experience the mutation.

4.5. Global genetic operators

To make different subpopulations co-evolutionary, all the subpopulations of the traditional MGAs share the global best-performing individual in each generation, known as the elitism sharement [21]. Besides usage of the elitism sharing, we propose an intermarriage crossover strategy to keep chromosome information exchange between individuals from different subpopulations.

4.5.1. Intermarriage crossover

For each subpopulation i $(i = 1, ..., N_{sub})$, we randomly choose N_c individuals from it. Then, the chosen individuals undergo the crossover operation. Finally, we insert the best N_c offsprings into each subpopulation.

4.5.2. Elitism sharement

By preserving the global best-performing chromosome, the elitism sharement mechanism enables GAs to improve the fitness value [21]. At each generation, we collect the global best individual from all the subpopulations declared as the elite, and share it among the subpopulations. In each subpopulation where the elite is to be added, the global best chromosome replaces the worst chromosome.

Remark 2. *In the selection and elitism sharing processes, the calculation of fitness (17) for each individual chromosome in each iteration is based on the optimal navigation control law (10).*

4.6. Tabu search strategy

From Remark 2, it is important to notice that the initial number of carried sensors, the target serving sequence, and the currents affect the optimal time for one vehicle to travel between two positions. As each variable mentioned above has multiple choices and depends on time, the time-based cost matrix $C = (t_{ij})_{\forall i,j \in \mathcal{V}'}$ in (15) for each individual chromosome may be different. With the evolution of the chromosomes, however, the optimal traveling time for one vehicle which carries a certain number of sensors at certain time t to travel between two target points might have already been calculated in the former generations. What's more, it is computationally expensive to calculate the optimal traveling time for one vehicle to travel between each two consecutive target points of each chromosomes at each generation.

Thus, we use a tabu matrix T_m to record the traveling information of all the vehicles with each row of the tabu matrix $T_m(i)$ being

$$T_m(i) = [c_p, g_p, v(t), s_t, e_t, n_a], \tag{18}$$

where c_p and g_p are two-dimensional row vectors representing the current position and the goal position of one vehicle respectively, $v(t)$ is the corresponding speed, s_t and e_t are the times one vehicle moves from the c_p and reaches the g_p, and n_a is the initial navigation angle.

Then, when calculating the optimal traveling time between two consecutive target points in each route of every chromosome in time-invariant currents, we first check whether the vehicle's current condition $c_p, g_p, v(t)$ is recorded in one row of the T_m. In time-varying currents, each vehicle's $c_p, g_p, v(t), s_t$ are checked. If that information has already been recorded in the T_m, we directly use the traveling cost $(e_t - s_t)$ and n_a; Otherwise, we calculate them and put them in a new row of the tabu matrix. That are the steps of our tabu strategy.

4.7. Stopping criteria

In each generation, CMGA applies the global operators to enable the gene information get exchanged between chromosomes from different subpopulations

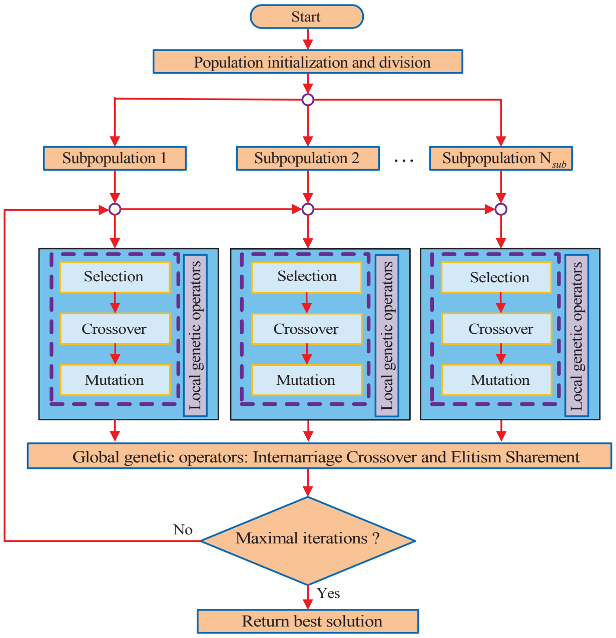

Fig. 5. The flowchart of CMGA.

after employing the local operators. If the maximal iteration is not reached, each subpopulation keeps its chromosomes resulting from the local operators and the global operators. Then, the subpopulations undergo the local operators and the global operators iteratively until reaching the maximum number of iterations. The flowchart of CMGA is shown in Fig. 5.

5. Simulation

To investigate the validity of the integrated algorithm, the task assignment for multiple vehicles is implemented on different environmental scenarios. First, CMGAs under different numbers of subpopulations are applied to several VRP benchmarks. Hereafter, the CMGA with the best performance is integrated with the path planning method to perform the sensor delivery task assignment in both time-invariant and time-varying drift fields. Finally, after relaxing the sensor loading constraint of the vehicles, the integrated CMGA is tested on the

Table 1. The average total traveling distance (m) of 100 simulations for the vehicles guided by the algorithms under five VRP benchmarks. A-n32k5 means 5 vehicles located at one depot need to deliver cargoes to 31 target locations, CMGA$_2$ means the number of subpopulation of CMGA is 2, and the best objective for the instances already known is B_o.

Instance	B_o	GA	GAMO	CMGA$_2$	CMGA$_3$	CMGA$_4$	CMGA$_5$	CMGA$_6$
A-n32k5	784	793.3	791.3	787.1	788.6	787.7	786.7	786.8
A-n44k6	937	1057.8	998.3	987.3	987.2	988.9	986.7	994.2
A-n48k7	1073	1139.7	1134.3	1121.1	1122.2	1123.1	1123.6	1126.1
A-n55k9	1073	1275.0	1208.1	1121.5	1118.5	1118.7	1123.4	1124.2
A-n63k10	1314	1498.3	1454.4	1389.1	1386.8	1387.6	1385.3	1385.9

resulting target visiting assignment problem where the variation of the solution quality with an increasing number of employed vehicles is investigated. All the experiments have been performed on an Intel Core $i5-4590$ CPU 3.30 GHz with 8 GB RAM, with algorithms compiled by Matlab under Windows 7.

5.1. CMGA's performance test

The performances of the CMGAs with different numbers of subpopulations are compared with that of a standard GA, and a GA using multiple offspring strategy (GAMO) on several Capacitated VRP benchmarks, denoted by A-n32k5, A-n44k6, A-n48k7, A-n55k9 and A-n63k10 whose information is available at $http : //neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/$. The numbers of chromosomes for the GA and GAMO on each applied benchmark are approximately 3 times the size of the benchmark, which are 120, 180, 180, 180 and 240, respectively. The CMGAs with the numbers of subpopulations ranging from 2 to 6 are tested, where the initial chromosomes in the GA are evenly divided to each subpopulation, i.e., $N_p = 40$ for A-n32k5 if $N_{sub} = 3$. Inspired by the fact that each pair of mating parents has multiple children in each generation, C_c is set to 10 to make each mating parent pair have multi-

Table 2. The average computation time (s) for the algorithms to converge on the VRP benchmarks.

Instance	GA	GAMO	CMGA$_2$	CMGA$_3$	CMGA$_4$	CMGA$_5$	CMGA$_6$
A-n32k5	101.8	154.6	40.0	46.7	46.6	40.2	44.0
A-n44k6	227.5	360.2	164.6	149.1	165.2	150.1	159.0
A-n48k7	262.4	392.6	200.2	189.8	188.5	194.2	181.9
A-n55k9	326.0	581.1	226.5	264.7	252.0	263.2	253.7
A-n63k10	492.5	771.5	441.1	435.6	429.1	422.7	416.1

ple crossover chances. According to the immigration rate of several developed countries, the number of individuals in each subpopulation undergoing the intermarriage crossover is set to 10% of the initial number of chromosomes in the subpopulation, which determines the N_c. The crossover rate and mutation rate for all the GAs are $P_c = 0.95$ and $P_m = 0.1$ respectively as suggested by [17, 26]. All the GAs terminate at the maximal iteration number 350. The punish variable M in (4) is set to 1000000.

The speeds of all the vehicles are normalized to 1 which is not affected by the number of cargoes carried on the vehicles, thus leading to the objective function (4) of a feasible solution to be the total traveling distance of the vehicles. Each algorithm is executed for 100 runs on each of the benchmarks. The average total traveling distance of the vehicles guided by the algorithms on each benchmark is presented in Table 1, where the published minimal total traveling distance for each benchmark is noted as B_o. Table 1 first shows that GAMO has better performance compared with the standard GA as its total traveling distances are closer to the minimal. The reason is that applying the multiple offspring strategy enables GAMO to have a better exploration of the solution space for each two mating chromosomes as the desirable genes of the mating parents are inherited by the offsprings with bigger probability. Furthermore, the CMGAs with different numbers of subpopulations perform better than GAMO, which shows the feasibility of CMGA. This is partly due to the fact that using mul-

Table 3. The average total traveling time (s) of the vehicles guided by the integrated CMGA$_5$ and the greedy algorithm on the sensor delivery task assignment problem under different instances where $n12m3$ means the number of targets is 12 and the number of employed vehicles is 3.

Algorithm	$n12m3$	$n21m5$	$n30m6$	$n44m8$	$n60m10$
Integrated greedy algorithm	4285.4	7208.5	9482.0	12705.7	15996.1
Integrated CMGA$_5$	3413.0	5719.8	7616.9	10716.0	12842.4

tiple subpopulations makes CMGA search for the solution in more directions, and less vulnerable to the local optimal. The results show that CMGA with 5 subpopulations, denoted by CMGA$_5$, has the best performance among the CMGAs. However, there is no significant difference between the CMGAs. Table 2 shows the corresponding average computation time (s) for the algorithms to converge on the benchmarks, where the multi-population CMGAs have less computation times compared with GA and GAMO. That shows the scalability of CMGA.

5.2. Sensor delivery task assignment for multiple vehicles in drift fields

In this section, the designed algorithms are employed to solve the task assignment problem where a set of target locations need some sensors to be delivered by vehicles located at a depot. Five instances of the problem, each with a certain number of targets in $\{12, 21, 30, 44, 60\}$, are tested where the sensor demands for the targets are randomly generated between 1 to 30. The sensor loading capacity of each vehicle is set to 100. Then, according to (3), the number of vehicles to be employed in each instance is set to be $3, 5, 6, 8, 10$, respectively. Finally, five instances of the sensor delivery task assignment problem are created, namely $n12m3$, $n21m5$, $n30m6$, $n44m8$ and $n60m10$ where $n12m3$ means the number of targets is 12 and the number of employed vehicles is 3. Due to the desirable performance among the CMGAs in the above section, CMGA$_5$ is integrated with the optimal path planning algorithm to deal with the sensor

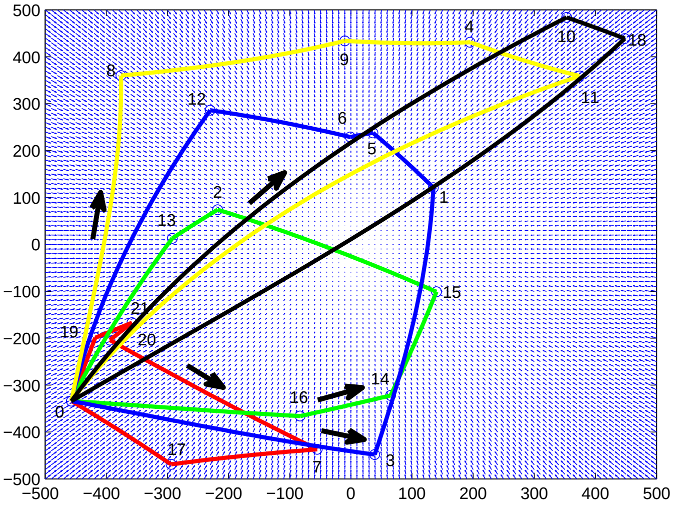

Fig. 6. The integrated CMGA for 5 vehicles initially located at depot 0 to deliver sensors to 21 target locations in the time-varying drift field $\vec{v}_c = [(-200x - 0.4ty) * 10^{-6}, (0.4tx - 200y) * 10^{-6}]^T$. The total traveling time of the vehicles is 6371.7s.

delivery task assignment for vehicles in both time-varying and time-invariant drift fields. The total numbers of chromosomes in CMGA$_5$ for the five instances are respectively 60, 90, 90, 120 and 180 while the other parameters are kept the same.

The integrated CMGA$_5$ is compared with an integrated greedy algorithm where vehicles always move towards the unassigned target location with the smallest traveling time. For each instance, 10 scenarios of the initial positions of the targets and the depot are randomly dispersed in an area of 1 km × 1 km. The first two instances are assumed to be in a time-varying drift field of $\vec{v}_c = [(-200x - 0.4ty) * 10^{-6}, (0.4tx - 200y) * 10^{-6}]^T$ and the others are in a time-invariant drift field given by $\vec{v}_c = [(0.3x + 0.2y) * 10^{-3}, (-0.2x + 0.3y) * 10^{-3}]^T$ [3]. The motion dynamics of the vehicles are described by (2) and their maximum speed v_{max} is $2m/s$. The average total traveling times of the vehicles guided by

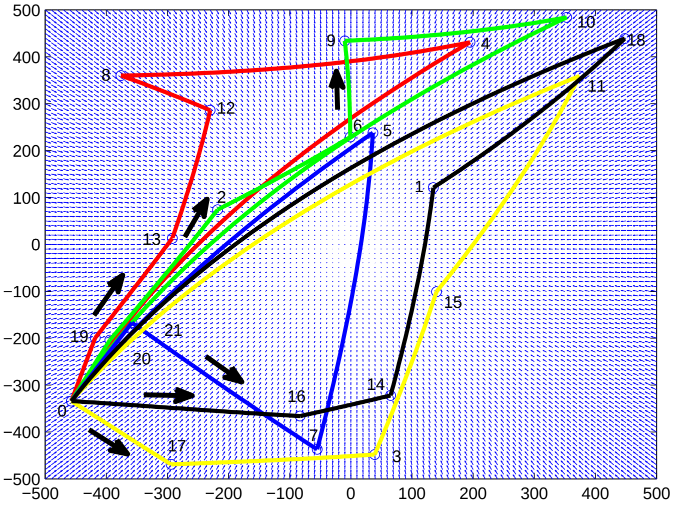

Fig. 7. The integrated greedy algorithm for 5 vehicles initially located at depot 0 to deliver sensors to 21 target locations in the time-varying drift field $\vec{v}_c = [(-200x - 0.4ty) * 10^{-6}, (0.4tx - 200y) * 10^{-6}]^T$. The total traveling time of the vehicles is 7063.9s.

the algorithms are presented in Table 3, which shows the better performance of the integrated CMGA$_5$ over the integrated greedy algorithm.

For one scenario of the instance $n21m5$, the numbers of sensors to be delivered to the 21 target points are 19, 11, 29, 22, 17, 16, 14, 21, 26, 22, 18, 19, 23, 18, 12, 27, 24, 20, 15, 24 and 15 respectively. The vehicles' routes resulting from the integrated CMGA$_5$ and greedy algorithm for the scenario are shown in Fig. 6 and Fig. 7 respectively, in which the small blue arrows are the current vectors at time $t = 1$ and the big black arrows are the traveling directions of the vehicles. Compared with Fig. 7, Fig. 6 shows that the integrated CMGA$_5$ can lead to satisfying solution to the task assignment problem where the path between two target locations takes the advantage of the currents.

Table 4. The average total traveling time (s) for vehicles guided by the integrated CMGA$_5$ to visit a set of target locations on 10 simulations, where $n50$ means the number of target locations to be visited is 50 and $m1$ means the number of employed vehicles is 1.

Instance	$m1$	$m2$	$m3$	$m4$	$m5$	$m6$	$m7$	$m8$
$n50$	5896.7	5785.1	5735.3	5678.5	5624.0	5586.0	5592.3	5519.1
$n70$	7943.0	7767.3	7523.6	8017.7	7832.8	8056.3	7929.9	8204.3
$n90$	8671.4	8892.5	8894.8	8659.7	9079.2	8971.0	8809.3	9204.2
$n100$	9404.7	9318.6	9838.0	9244.2	9280.2	9344.5	9138.7	9287.5
$n110$	10727.6	10871.1	10709.9	10709.8	10425.0	10334.5	10281.4	10946.9
$n120$	12148.7	11438.3	11456.5	11657.1	11021.4	11086.2	11536.5	11653.0

5.3. Target visiting assignment for multiple vehicles in a time-invariant drift field

Relaxing (1) and (3), the senor delivery assignment problem is transferred to the target visiting assignment problem where a set of target locations need to be visited by a fleet of vehicles located at a depot. The integrated CMGA$_5$ is tested on the relaxed problem in the time-invariant drift field of $\vec{v}_c = [(0.3x + 0.2y) * 10^{-3}, (-0.2x + 0.3y) * 10^{-3}]^T$, and the vehicles travel with the constant speed $v = 1m/s$. Several instances with different numbers of target locations $n \in \{50, 70, 90, 100, 110, 120\}$ are carried out where the target locations are randomly dispersed in an area of 1 km × 1 km. The influence of the number of employed vehicles on the total traveling time to visit all the targets is investigated. For each instance, simulations of different numbers of employed vehicles, m ranging from 1 to 8, are performed.

The total numbers of chromosomes in the integrated CMGA$_5$ for the 6 instances are respectively 150, 210, 270, 300 , 330 and 360 while the other parameters are kept the same. For each scenario, 10 runs of the integrated CMGA$_5$ are performed. The average total traveling time of the vehicles guided by the integrated algorithm on each scenario is listed in Table 4. For the instance with target number $n = 50$, the total traveling time of the vehicles is the smallest when the number of employed vehicle is the biggest, i.e. $m = 8$. However, for the other instances it is not the case. The table shows that a bigger number of employed vehicles does not necessarily lead to a smaller total traveling time of the vehicles, which helps practitioners to make the choice of the number of vehicles employed to perform the task assignment problem.

6. Conclusions

In this paper, we investigated the task assignment problem for multiple vehicles in a drift field. A path planning method was first designed which enables the vehicles to move between two prescribed positions in a drift field with the minimal time. For the sensor deliver task assignment, the effect of carried sensors on the speed of the vehicles was considered when designing the optimal navigation law. The traveling time resulted from the path planning method provided the route information for the target points assignment. In addition, a multi-population genetic algorithm namely CMGA was designed to assign the vehicles to the target points. The performance of CMGA under different numbers of subpopulations is investigated. Furthermore, integrating the path planning method and the CMGA efficiently solved the sensor delivery task assignment to multiple vehicles in both time-invariant and time-varying drift fields. We have also tested the performance of the integrated CMGA on the target visiting assignment problem where different numbers of vehicles are employed, and found that a bigger number of employed vehicles does not necessarily lead to a smaller total traveling time. The proposed algorithms will be extended for task assignment of vehicles in drift fields where currents might lead to some area unreachable by the vehicles.

References

[1] C. W. Ahn, R. S. Ramakrishna, A genetic algorithm for shortest path routing problem and the sizing of populations, IEEE transactions on evo-

lutionary computation 6 (6) (2002) 566–579.

[2] E. Alba, Parallel metaheuristics: a new class of algorithms, vol. 47, John Wiley & Sons, 2005.

[3] E. Bakolas, P. Tsiotras, Optimal partitioning for spatiotemporal coverage in a drift field, Automatica 49 (7) (2013) 2064–2073.

[4] G. Binetti, D. Naso, B. Turchiano, Decentralized task allocation for surveillance systems with critical tasks, Robotics and Autonomous Systems 61 (12) (2013) 1653–1664.

[5] E. Bovio, D. Cecchi, F. Baralli, Autonomous underwater vehicles for scientific and naval operations, Annual Reviews in Control 30 (2) (2006) 117–130.

[6] H.-L. Choi, L. Brunet, J. P. How, Consensus-based decentralized auctions for robust task allocation, Robotics, IEEE Transactions on 25 (4) (2009) 912–926.

[7] T. S. Dahl, M. Matarić, G. S. Sukhatme, Multi-robot task allocation through vacancy chain scheduling, Robotics and Autonomous Systems 57 (6) (2009) 674–687.

[8] M. Darrah, W. Niland, B. Stolarik, Increasing UAV task assignment performance through parallelized genetic algorithms, in: AIAA Infotech@ Aerospace 2007 Conference and Exhibit, 2815, 2007.

[9] B. Dieber, C. Micheloni, B. Rinner, Resource-aware coverage and task assignment in visual sensor networks, Circuits and Systems for Video Technology, IEEE Transactions on 21 (10) (2011) 1424–1437.

[10] R. J. Duro, M. Graña, J. de Lope, On the potential contributions of hybrid intelligent approaches to multicomponent robotic system development, Information Sciences 180 (14) (2010) 2635–2648.

[11] E. Edison, T. Shima, Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms, Computers & Operations Research 38 (1) (2011) 340–356.

[12] M. Franceschelli, D. Rosa, C. Seatzu, F. Bullo, Gossip algorithms for heterogeneous multi-vehicle routing problems, Nonlinear Analysis: Hybrid Systems 10 (2013) 156–174.

[13] S. Ge, T. Lee, G. Zhu, Genetic algorithm tuning of Lyapunov-based controllers: An application to a single-link flexible robot system, IEEE Transactions on Industrial Electronics 43 (5) (1996) 567–574.

[14] J. Gu, M. Gu, C. Cao, X. Gu, A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem, Computers & Operations Research 37 (5) (2010) 927–937.

[15] J. Han, J. Ok, W. K. Chung, An Ethology-Based Hybrid Control Architecture for an Autonomous Underwater Vehicle for Performing Multiple Tasks, Oceanic Engineering, IEEE Journal of 38 (3) (2013) 514–521.

[16] B. Jiang, A. N. Bishop, B. D. Anderson, S. P. Drake, Optimal path planning and sensor placement for mobile target detection, Automatica 60 (2015) 127–139.

[17] S. Karakatič, V. Podgorelec, A survey of genetic algorithms for solving multi depot vehicle routing problem, Applied Soft Computing 27 (2015) 519–532.

[18] D. E. Kirk, Optimal control theory: an introduction, Courier Corporation, 2012.

[19] L. Liu, D. A. Shell, Large-scale multi-robot task allocation via dynamic partitioning and distribution, Autonomous Robots 33 (3) (2012) 291–307.

[20] L. Liu, D. A. Shell, An anytime assignment algorithm: From local task swapping to global optimality, Autonomous Robots 35 (4) (2013) 271–286.

[21] M. Mitchell, An introduction to genetic algorithms, MIT press, 1998.

[22] S. Moon, E. Oh, D. H. Shim, An integral framework of task assignment and path planning for multiple unmanned aerial vehicles in dynamic environments, Journal of Intelligent & Robotic Systems 70 (1-4) (2013) 303–313.

[23] E. Osaba, E. Onieva, R. Carballedo, F. Diaz, A. Perallos, X. Zhang, A multi-crossover and adaptive island based population algorithm for solving routing problems, Journal of Zhejiang University SCIENCE C 14 (11) (2013) 815–821.

[24] D. Park, J. Han, W. K. Chung, Optimal mission planning for underwater environment, in: Ubiquitous Robots and Ambient Intelligence (URAI), 2013 10th International Conference on, IEEE, 554–555, 2013.

[25] D. Rao, S. B. Williams, Large-scale path planning for Underwater Gliders in ocean currents, in: Australasian Conference on Robotics and Automation (ACRA), Sydney, Citeseer, 2009.

[26] N. M. Razali, J. Geraghty, et al., Genetic algorithm performance with different selection strategies in solving TSP, in: Proceedings of the world congress on engineering, vol. 2, International Association of Engineers Hong Kong, 1134–1139, 2011.

[27] K. Shi, H. Chen, Y. Lin, Probabilistic coverage based sensor scheduling for target tracking sensor networks, Information sciences 292 (2015) 95–110.

[28] T. Shima, S. J. Rasmussen, A. G. Sparks, K. M. Passino, Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms, Computers & Operations Research 33 (11) (2006) 3252–3269.

[29] S. L. Smith, F. Bullo, Monotonic target assignment for robotic networks, Automatic Control, IEEE Transactions on 54 (9) (2009) 2042–2057.

[30] M. Soulignac, Feasible and optimal path planning in strong current fields, Robotics, IEEE Transactions on 27 (1) (2011) 89–98.

[31] P. Toth, D. Vigo, Vehicle routing: problems, methods, and applications, vol. 18, Siam, 2014.

[32] Y. Xu, K. Li, J. Hu, K. Li, A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues, Information Sciences 270 (2014) 255–287.

[33] W. Yan, X. Bai, X. Peng, L. Zuo, J. Dai, The routing problem of autonomous underwater vehicles in ocean currents, in: OCEANS 2014-TAIPEI, IEEE, 1–6, 2014.

[34] J. Yu, S.-J. Chung, P. G. Voulgaris, Target assignment in robotic networks: Distance optimality guarantees and hierarchical strategies, Automatic Control, IEEE Transactions on 60 (2) (2015) 327–341.

[35] L. Zhong, Q. Luo, D. Wen, S.-d. Qiao, J.-m. Shi, W.-m. Zhang, A task assignment algorithm for multiple aerial vehicles to attack targets with dynamic values, Intelligent Transportation Systems, IEEE Transactions on 14 (1) (2013) 236–248.

[36] D. Zhu, H. Huang, S. X. Yang, Dynamic task assignment and path planning of multi-AUV system based on an improved self-organizing map and velocity synthesis method in three-dimensional underwater workspace, Cybernetics, IEEE Transactions on 43 (2) (2013) 504–514.

[37] L. Zuo, J. Chen, W. Yan, Y. Shi, Time-optimal coverage control for multiple unicycles in a drift field, Information Sciences 373 (2016) 571–580.