ECOBAS Working Papers

2017 - 07

Title:

INTEGRATING FORECASTING IN METAHEURISTIC METHODS TO SOLVE DYNAMIC ROUTING PROBLEMS: EVIDENCE FROM THE LOGISTIC PROCESSES OF TUNA VESSELS

Authors:

Carlos Groba
Universidade de Vigo

Antonio Sartal
Universidade de Vigo

Xosé H. Vázquez
Universidade de Vigo

ECOBAS
Economics and Business Administration for Society
www.ecobas.es

Integrating forecasting in metaheuristic methods to solve dynamic routing problems: Evidence from the logistic processes of tuna vessels

Groba, Carlos* Sartal, Antonio† Vázquez, Xosé H.‡

Abstract

The multiple Traveling Salesman Problem (mTSP) is a widespread phenomenon in real-life scenarios, and in fact it has been addressed from multiple perspectives in recent decades. However, mTSP in dynamic circumstances entails a greater complexity that recent approaches are still trying to grasp. Beyond time windows, capacity and other parameters that characterize the dynamics of each scenario, moving targets is one of the underdeveloped issues in the field of mTSP. The approach of this paper harnesses a simple prediction method to prove that integrating forecasting within a metaheuristic evolutionary-based method, such as genetic algorithms, can yield better results in a dynamic scenario than their simple non-predictive version. Real data is used from the retrieval of Fish Aggregating Devices (FADs) by tuna vessels in the Indian Ocean. Based on historical data registered by the GPS system of the buoys attached to the devices, their trajectory is firstly forecast to feed subsequently the functioning of a genetic algorithm that searches for the optimal route of tuna vessels in terms of total distance traveled. Thus, although valid for static cases and for the Vehicle Routing Problem (VRP), the main contribution of this method over existing literature lies in its application as a global search method to solve the multiple TSP with moving targets in many dynamic real-life optimization problems.

Keywords

multiple traveling salesman problem, genetic algorithm, vehicle routing problem, fish aggregating devices, moving targets

*cgroba@uvigo.es; University of Vigo.

†antoniosartal@uvigo.es; University of Vigo.

‡Corresponding author: xhvv@uvigo.es; University of Vigo; Facultade de Economía. Rúa Leonardo da Vinci. 36310 Vigo. Spain. tel: +34986812479; fax: +34986812401.

1 Introduction

This paper addresses the synergies in combining a predictive technique with a metaheuristic evolutionary-based method to solve the multiple Traveling Salesman Problem (mTSP) with moving targets (mTSP-MT). The mTSP-MT is the generalization of the well-known Traveling Salesman Problem (TSP). It deals with multiple salesmen, and targets (e.g., customers or objects) are not fixed. As in any TSP, however, the aim is to minimize the total distance traveled by all salesmen.

The mTSP-MT is therefore more suitable than the ordinary TSP for a wider range of real-world problems. In fact, this is the method used, for example, in the defense sector to protect an airport or a security zone from mobile intruders (raiders, animals, vehicles, etc.), or in the logistics sector to supply a fleet of boats or mobile ground units (Stieber & Fügenschuh, 2017; Stieber, Fügenschuh, & Yuan, 2015). It can also be applied to the Vehicle Routing Problem (VRP) with multiple vehicles, time windows or capacity restrictions (Bae & Chung, 2017; Sundar, Venkatachalam, & Rathinam, 2017). New potential uses are also emerging every day in mobility and delivery services (e.g., delivery services, real-time mobility requirements, drones scheduling and collaboration, etc.) conducted by companies such as Uber and Amazon (M. B. Menezes, Ketzenberg, Oliva, & Metters, 2015).

Whereas the diverse perspectives and problem-solving methods have helped practitioners and scholars to address a multitude of TSPs, including mTSPs in various industries (M. B. Menezes et al., 2015), the literature on mTSP-MT is still scarce. This is possibly due to the greater complexity of this type of problems compared to conventional TSP approximations, which may also explain why ad hoc experiments with many restrictions (small distances, planned routes, fixed starting-points, etc.) have often ended with few or not applicable solutions for real-life problems. For example, C.-H. Liu (2013) and Jiang, Sarker, and Abbass (2005) proposed various solutions for the mTSP-MT problem by narrowing the scope of analysis to one dimension and limiting the working speed. Similarly, other studies have restricted the positions of the salesmen (e.g., by having them start at the same point, located in the middle of the area) or the possible targets' movements (e.g., by forcing customers to move in a structured path) (M. B. Menezes et al., 2015; Stieber & Fügenschuh, 2017). In addition, regarding the calculation methodology, the most recent approaches have worked on a real-time basis and have been recalculated to find the changes between nodes (Hajjam, Créput, & Koukam, 2013; Zhou, Kang, & Yan, 2003). However, they have not anticipated the targets' future movement, so the optimal solutions appear only when the changes are communicated and the algorithms have been recalculated.

Against this background, the paper addresses the problem of moving targets by combining a predictive technique (Newton's movement equation) with a genetic algorithm (GA). This new approach, which could be named genetic algorithm based in multiple-trajectory prediction (GAMTP), yields a generic solution that not only suits dynamic and static scenarios, but it also applicable to any real-world problem with multiple travelers and mobile targets. GAMTP thus combines prediction and GA in a single method to reach a better global optimization solution than when GAs alone used without internalizing prediction.

Since the objective of this work is to prove that integrating forecasting within a metaheuristic method (e.g., genetic algorithms) better results are achieved than in the simple non-predictive version; we chose Newton's movement equation as our predictive technique over other techniques because it offers a quick, short-term prediction even when provided with very little information (Groba, Sartal, & Vázquez, 2015). Analogously, GAs were chosen as metaheuristic method for three reasons: (1) they are evolutionary, which is mandatory for the algorithm implementation; (2) they show fundamental properties in terms of robustness and statistical convergence; and (3) they can reach a solution within an acceptable computational time (Jih & Hsu, 2004).

Data come from different group of tuna vessels retrieving their fish aggregating devices (FADs) in the Indian Ocean during April in 2017. FADs drift in the sea and provide an artificial substrate for attaching organism such as algae and invertebrates. This phenomenon probably stimulates a food chain that attracts different type of fish. Tuna also tends to gather beneath them. All FADs are attached to a buoy with a Global Positioning System (GPS) that transmits its coordinates every 12 hours. The people steering the vessels, which work in groups, need to design their routes to recover the constantly moving FADs in order to minimize the total distance traveled. We compare our results with both the most commonly used method, the nearest neighbor (NN) strategy, and a classic mTSP approach based on GA (i.e., without prediction) (Bjarnadottir, 2004).

The paper is organized as follows. The next section provides a review of the literature. Sections 3 and 4 describe, respectively, the data and methodology. Section 5 introduces the model and presents the experimental design. Section 6 discusses results and, finally, Section 7 concludes by highlighting the paper's main contributions and its implications.

2 Literature Review

TSP is a well-known of Combinatorial Optimization (CO) problems that is NP-hard (Garey & Johnson, 1983), and in which, assuming that $P \neq NP$, no polynomial time algorithm exists (Karp, 1972).

An extension of TSP involves more than one salesman (mTSP), and assumes that each city must be visited exactly once and by only one salesman (Bektas, 2006a; Venkatesh & Singh, 2015). Thus, given a start-and-end point (a depot), a set of n cities to be visited by one salesman, and m salesmen (where $n > m$ the optimal), then the mTSP consists of finding routes for all m salesmen such that the cost of visiting all cities is minimized. The cost can be defined in terms of distance, time, or other criteria. Thus, although mTSP is NP-hard like TSP but entails a more complicated problem because cities must be assigned firstly to each salesman, and then the optimum order is subsequently determined for each salesman. Two main versions of the mTSP can be defined based on the number of depots.

In the first version all m salesmen start and end at one depot. In the second version every salesman begins and ends at a different depot. We address the second variant, which represents a more generalized situation that aims at minimizing the total distance the salesmen travel (i.e., the total length of all routes). This approach reflects the strand of literature dealing with multiple Traveling Salesman Problem (Bektas, 2006b).

Furthermore, the solution shows an additional trait: targets are not fixed and can vary their positions over time. This variant is known as mTSP with moving targets (mTSP-MT), which is a dynamic generalization of the mTSP that makes the problem more suitable to a wider range of real-world situations in various industries. In fact, many mTSP-MT applications exist, for example, in supply logistics (Stieber et al., 2015), robotic patrolling (Pushkarini Agharkar & Bullo, 2015), scheduling and routing (e.g., bank-crew scheduling, workload balancing, and school-bus routing), as well as in the defense sector (e.g., the multiple-weapons-to-multiple-targets assignment problem) (Stieber & Fügenschuh, 2017). On a broad perspective, one of the fields that could highly benefit from this type of analysis in the future is the transportation and delivery service (including unmanned aerial vehicle services) leaded by companies such as Uber, Amazon and others (Agatz, Bouman, & Schmidt, 2016; Dorling, Heinrichs, Messier, & Magierowski, 2017).

Curiously enough, however, there are relatively few approaches to solve mTSP-MTs. Again, this is possibly due to the greater complexity inherent to the dynamics of mTSP-MT in comparison with traditional TSP (Garcia-Najera & Bullinaria, 2011; Hajjam et al., 2013) or the simpler moving-target TSP (e.g. a supply ship that resupplies patrol boats as they work, a fishing boat collecting its catch at sea, or an airplane that must intercept a number of mobile ground units) (Groba et al., 2015; Helbing & Tilch, 1998). There are also variants of the TSP-MT (including one with resupply) in which the salesmen must return to the depot after intercepting each target (Jiang et al., 2005; Jindal & Kumar, 2011; L. Liu, Wang, & Yang, 2009).

Whereas this specific literature has helped researchers to explore the TSP-MT, it has proceeded so far with a high number of restrictions in order to reach a feasible result (Blum & Roli, 2003b; Helvig, Robins, & Zelikovsky, 2003). The cost of this strategy, nevertheless, is that they often end up with few or not applicable real-world solutions. For instance, Jiang et al. (2005) described a solution approach based on GA with a fixed number of cities in which the target moved at a constant velocity, which in fact is a very common assumption in this area. The same speed restriction has been considered in moving-target TSP situations (Helbing & Tilch, 1998; Helvig et al., 2003). Similarly, other studies have restricted the salesmen's position (e.g., by requiring that they start at the same point in the middle of the area) or the possible targets' movements (e.g., by having the customers only move in structured paths) (T. Menezes, Tedesco, & Ramalho, 2006; Stieber et al., 2015). In the same way, Jindal and Kumar (2011) assumed all targets started from their starting position, were only in one dimension, and moved with constant velocity. In general terms, therefore, this literature shows that the available research can hardly solve practical problems; rather, it is mainly focused on providing structure and analyzing variants of the TSP that answer specific questions in made-to-measure approximations of reality.

Similar arguments hold for VRP (Braekers, Ramaekers, & Van Nieuwenhuyse, 2016; Toth & Vigo, 2014), which could be considered as a generalization of mTSP with particular applications to transport and logistic (Montoya-Torres, Franco, Isaza, Jiménez, & Herazo-Padilla, 2015). The main difference between the classic moving TSP and the moving VRP is that the VRP can include additional restrictions beyond distance, such as added vehicle capacity, time constraint, a known non-negative demand for each depot, and a non-negative cost for each route (Eksioglu, Vural, & Reisman, 2009). Nevertheless, just as it happens with mTSP, research on VRP with moving targets is still underdeveloped. The existing literature focuses on Unmanned Aerial Vehicles (UAVs) (including combat UAVs), surveillance missions, and military needs, but none of these studies offers a generic solution with no restrictions (Geng, Zhang, Wang, Fuh, & Teo, 2014; Shetty, Sudit, & Nagi, 2008; Shima & Schumacher, 2005). Thus, although our research hinges on mTSP-MT, it can also contribute to solve VRP problems (Cattaruzza, Absi, Feillet, & González-Feliu, 2017).

Summing up, there is a gap in the literature on mTSP and VRP with regard to dynamic scenarios. Research so far has worked with basic settings and simplified parameters that not only lead to a continuous recalculation of the solution, but make this very same solution of limited application to real-life situations. Our proposal, however, addresses simultaneously three key issues that characterize any real-world situation: (1) multiple targets for (2) multiple salesmen and (3) in dynamic scenarios. This makes it a generalized solution for static and dynamic scenarios of mTSP and VRP, and opens multiple real-world applications in many scientific and business fields, from medicine or physics to production and logistics.

3 Data

3.1 Introduction: FAD recovery for tuna vessels

The global tuna fishery is one of the largest in the world. Aggregate catches of tuna and associated species, including swordfish and other billfishes, reached a record level of 6.6 million tons in 2010 (Food and Agriculture Organization of the United nations, FAO, 2012). The most widely used and fastest-growing fishing gear for targeting tuna is the purse seine. Since the 1950s, purse-seining vessels have benefited from the adoption of power blocks, increases in fish-holding capacity and freezing technology, improvements in tuna-locating techniques (e.g., helicopters, bird sonar, GPS, and most recently, UAVs), and the use of FADs (Miyake, Guillotreau, Sun, & Ishimura, 2010).

FADs are human-made structures that facilitate the attraction and aggregation of ocean-going pelagic fish such as tuna (Rajeswari, 2009), so fishermen have traditionally seeded them throughout the oceans to make their job more efficient. Still today, most FADs are handcrafted. This is a relevant issue because this specific trait make FADs drift on diverse courses and at different speeds. Furthermore, courses and speeds change with time because they depend on the sea currents and on superficial wind. For instance, currents beneath FADs move in a range from 0.2 knots[1] to 2 knots.

Each FAD is tied to a satellite buoy that sends information to the seiners from the FAD's on GPS position, battery level and water temperature. Simultaneously, most of the buoys are also equipped with echo sounders that transmit information about the aggregated biomass beneath them (Castro, Santiago, & Santana-Ortega, 2001). Vessels receive messages on all these parameters from the buoys at least twice a day via real-time satellite communication systems, such as Argos, Inmarsat, Orbcomm or Iridium (Moreno, Dagorn, Sancho, & Itano, 2007). Seiners can thus track FADs and the biomass beneath them in real time.

Information and communication technologies have thus expanded significantly the number of purse seiners using FADs during the last 15 years (Hallier & Gaertner, 2008). Some estimates suggest there are around 120,000 FADs deployed in the oceans. It is not surprising, consequently, that the use of FADs has been restricted in recent years to control fishing activity and to maintain better stocks of fishing resources. Organizations such as the Indian Ocean Tuna Commission apply normative rules to limit the quantity of FADs that each vessel can handle. For this and other reasons, each tuna vessel may share its FADs information among groups of two to five vessels, depending on the size of the company. A single vessel, on the other hand, can handle as many as hundreds of FADs.

Once the basic organization of the industry has been presented, it is worth noting that fuel consumption has been shown to be a major contributor to the operating costs of tuna fishing vessels, typically representing between 30 and 75% of total operating costs (Miyake et al., 2010). Vessels that fish tuna with purse seine have an average fuel-use intensity, weighted by landings, of 368 L/t (Parker, Vázquez-Rowe, & Tyedmers, 2015). Given a moment t, therefore, the challenge for tuna vessels is to figure out the best route for each vessel to maximize fishing and minimize travel. There are no restrictions regarding where the vessels are at time t or where they have to finish their tours. Under these circumstances, the majority of tuna vessels follow the Nearest Neighbour (NN) strategy, which is basically easy to implement. However, when the number of FADs to recover is greater than one, the constant movement of FADs make the NN method less useful.

Figure 1 represents the problem that the vessels face. Many FADs are drifting in the ocean, and some vessels that share FAD information need to plan their recovery. The FAD current position is represented in black; the white part reflects the drift in the last days. We can also observe the current positions of three vessels (red, yellow and green symbols) at time t. Hence, the figure presents the difficult problem that tuna vessels face when designing a common recollection strategy. If the high fuel consumption of tuna vessels is considered, the practical side of the challenge becomes probably more clear.

[1] The knot is a unit of speed equal to one nautical mile per hour, exactly 1.852 km/h

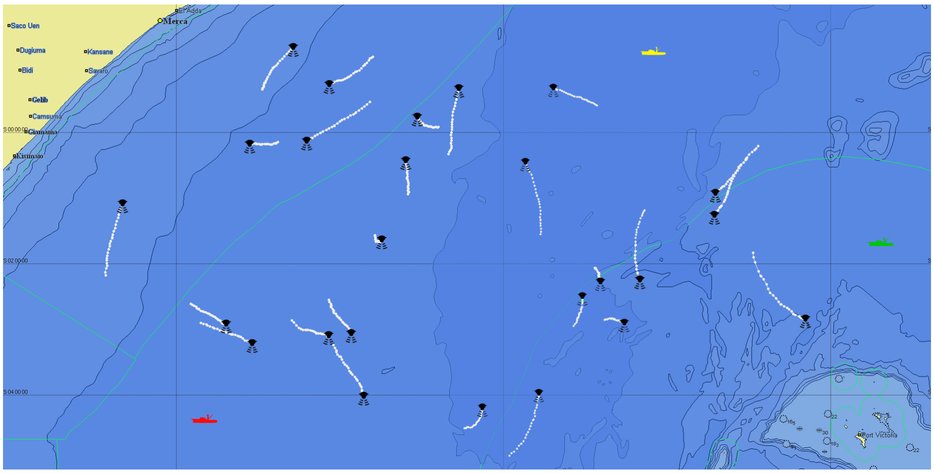

Figure 1: Drifting FADs and three vessels in the Indian Ocean.

Table 1: Mathematical symbols used.

Symbol	Description
n	number of cities or FADs
m	number of salesmen or vessels
t	time
r	number of predictions. Evolution of time t
f_i^t	position of FAD i at time t
$\hat{f}_i^r$	estimated position of FAD i at time r
v_i	initial position of vessel i
s_i	speed of vessel i
z_i	number of objects (FADs) that salesman (vessel) i has to recover
z_0	index of the initial FAD that a vessel has to pick up
z_f	index of the final FAD that a vessel has to pick up

The mathematical notations used in the article in Table 1 can help readers understand the proposed solution.

3.2 Data on buoys and vessels

3.2.1 FADs input

Our scenario is composed of n drifting objects (FADs), which are labeled f_i: $(f_1, f_2, \dots, f_n)$. Since we know their last transmitted coordinates as well as their last h positions, the input is an $n \times (h + 1)$ matrix:

$$\begin{pmatrix} f_1^t & f_1^{t-1} & f_1^{t-2} & \cdots & f_1^{t-h} \\ f_2^t & f_2^{t-1} & f_2^{t-2} & \cdots & f_2^{t-h} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_n^t & f_n^{t-1} & f_n^{t-2} & \cdots & f_n^{t-h} \end{pmatrix},$$

where the first column comprises all the objects at the current time t; the second column comprises all the objects at $t-1$, and so on and so forth throughout the matrix until the last column, which comprises the objects at $t-h$. Each object or FAD f_i^t is composed of two coordinates (latitude and longitude), which allows to represent the position of the FAD in the map: $f_i^t = (latitude_i^t, longitude_i^t)$. The variable h can be different for each FAD, and it depends on when it was released into the sea. In this case, having the last three positions of each FAD is enough (i.e., t, $t-1$ and $t-2$), so the FADs input matrix is as follows:

$$\begin{pmatrix} f_1^t & f_1^{t-1} & f_1^{t-2} \\ f_2^t & f_2^{t-1} & f_2^{t-2} \\ \vdots & \vdots & \vdots \\ f_n^t & f_n^{t-1} & f_n^{t-2} \end{pmatrix}$$

3.2.2 Vessels and fishing-information input

Given m vessels, the following inputs are needed:

- Initial position: $v_i = (v_{lat}, v_{lon})$, where $v_1 \neq v_2 \neq \ldots \neq v_m$.
- Average speed (s_i), in knots, when traveling from one object to the next. For simplicity, we use the same s_i for all vessels, but this value could be different in each case.
- Fishing time (FT) by object, with two possibilities:
  - The fishing time is the same for all objects: $FT_1 = FT_2 = \ldots = FT_n$.
  - The fishing time can be different for each object: $FT_1 \neq FT_2 \neq \ldots \neq FT_n$.

The fishing time could also be vessel-dependent instead of FAD dependent, using the same fishing time for all vessels ($FT_1 = FT_2 = \ldots = FT_m$) or a different fishing time for each vessel ($FT_1 \neq FT_2 \neq \ldots \neq FT_m$). For the sake of simplicity, our solution uses the same fishing time for all FADs.

4 Methodology

Based on historical data provided by GPS buoys tied to FADs, the paper proposes a new approach that combines a metaheuristic method with a predictive technique. We first estimate the trajectories of FADs and then run a genetic algorithm to determine the best possible route considering the FADs future locations. These steps are explained separately below in different subsections. The last one describes the final solution, which implements the GA with a multiple-trajectory prediction.

4.1 The predictive technique: estimating FADs future position

Once the information inputs are available, the next step is to predict the next position of each FAD. This challenge could be met with several methods that simulate and predict current movements in the sea (Özgökmen, Griffa, Mariano, & Piterbarg, 2000); however, most of them use complex mathematical models based on data that are difficult to obtain for a tuna vessel. On the one hand, FADs are hand-made objects, often built by fishermen themselves with a bamboo framework (about $3 \times 1.5\,\mathrm{m}$). Their manufacture therefore introduces a significant variability that makes the mathematical modeling of their movement very complex. On the other hand, underwater nets are commonly attached underneath these FADs. The length of these nets has progressively increased and can reach today a depth of 50 m in the Eastern Pacific (Fonteneau & Pianet, 2000). In any case, no two FADs can be assumed to drift identically under the same environmental conditions, which makes the modeling to predict future positions based on standard Lagrangian buoys a futile exercise.

Under these circumstances, a feasible alternative that needs little information and computational power is Newton's motion equation. It is thus used here to predict the future position of each FAD, because the last three positions of each object would be enough to obtain a rather accurate prediction in the short-term. In fact, the equation effectiveness decreases with time because the error has a cumulative effect. As soon as the buoys transmit their subsequent positions, however, the algorithm can update the last position and can be executed again to predict the best route. So if the prediction for a specific FAD is not accurate enough to predict the best set of routes, the solution can be updated when the next message is received, therefore showing a better optimal route if one exists. It is in this sense worth noting that the goal here is to predict where FADs will be in the near future (not only its next position, but many future positions). Newton's motion equation is thus applied between an initial point and a current or final point:

$$y_{t+1} = y_t + v_t \Delta_t + \tfrac{1}{2} a_t \Delta_t^2, \tag{1}$$

where

- y_t is the position at the end of the interval (displacement).
- v_t is the velocity at the end of the interval t.
- Δ_t is the time interval between the initial and current states.
- a_t is the acceleration at time t.

The variable t is considered discrete. Knowing the position of a FAD requires that the attached buoy transmits its coordinates through satellite communication. Since this is costly, the number of transmissions of each buoy is limited normally to twice a day. With this information, in order to calculate the future positions of an specific FAD (f_i) at time $t + x : x \in \{1, \ldots, r\}$ we only need its last three positions at time $t, t-1$, and $t-2$.

$$
\begin{aligned}
\hat{f}_{t+1} &= f_t + v_t \Delta_t + \frac{1}{2} a_t \Delta_t^2 \\
\hat{f}_{t+2} &= \hat{f}_{t+1} + \hat{v}_{t+1} \Delta_t + \frac{1}{2} \hat{a}_{t+1} \Delta_t^2 \\
&\vdots \\
\hat{f}_{t+r} &= \hat{f}_{t+r-1} + \hat{v}_{t+r-1} \Delta_t + \frac{1}{2} \hat{a}_{t+r-1} \Delta_t^2
\end{aligned}
$$

It is important to note that previous predictions are used to calculate new prediction values, so it is easy to understand that results worsen as this simple approach makes more predictions. Long term predictions would accordingly need a different prediction method indeed.

4.2 Algorithm design

mTSP is a class of NP-hard Combinatorial Optimization (CO) problem. Complete algorithms are guaranteed to find every finite size instance of a CO problem, but might need exponential computation time in the worst-case scenario. Approximate methods, by constrast, do not ensure optimal solutions but alternatively offer good solutions in a significantly reduced amount of time (Blum & Roli, 2003a).

In the last 20 years, a new kind of approximate algorithms has emerged that basically tries to combine basic heuristic methods in higher level frameworks aimed at efficiently and effectively exploring a search space. These methods are nowadays commonly called metaheuristics, and refer to an iterative generation process that guides a subordinate heuristic by combining different concepts for exploring and exploiting the search space. Furthermore, learning strategies are used to structure information in order to find efficiently near-optimal solutions (Osman & Laporte, 1996).

There are many heuristic methods to solve optimization problems like mTSP. Some examples include Ant Colony Optimization algorithms (Kuo & Zulvia, 2017; Mavrovouniotis & Yang, 2013), Particle Swarm Optimization algorithms (Du & Li, 2008; Lynn & Suganthan, 2017), Simulated Annealing (Song, Lee, & Lee, 2003) or Artificial Neural Networks (Yegnanarayana, 2009). Evolutionary algorithms (EA) are also a particular type of metaheuristic methods that are inspired by natural, self-organized systems and biological evolution. Examples of these algorithms include Genetic Algorithms (Deng, Liu, & Zhou, 2015; Li, Shao, Zuo, & Huang, 2017) or Artificial Immune Systems (Alonso, Oliveira, & de Souza, 2015; Banerjee, 2017), which -as any EA- are flexible in the sense that they can be adapted to changing environments by exploiting information from earlier moves. EAs are therefore especially suitable for optimization problems in dynamic environments such as the ones studied here (H.-f. Wang, Wang, & Yang, 2007; Zhou et al., 2003). Furthermore, the selected algorithm must be evolutionary for the following reasons (Zhou et al., 2003):

- EAs employ population policy and each individual in the population is an alternative solution for a dynamic TSP.
- The population policy allows individuals to hold diverse information. Population diversity has been proved to be a very important factor for species' existence in changing environments. By easily integrating some diversity-preserving techniques, individuals are also enabled in the algorithm to quickly fit the dynamic environments.

Within this context, GAs can be viewed as an evolutionary process whereby a population of solutions evolves over a sequence of generations to achieve a near-optimal solution. They can therefore be defined as computer programs that evolve in ways that resemble natural selection, solving complex problems that even their creators do not neccesarily have to understand completely (Holland, 1992). They were first introduced by Holland (1975) to address optimization problems using techniques inspired by natural evolution (Winter, Periaux, Galan, & Cuesta, 1996), and this idea has led to many theoretical developments over the last 40 years (Reinelt, 1994; Smith & Smith, 2002). In fact, GAs represent one of the most consolidated approaches to TSP (Potvin, 1996; Razali, Geraghty, et al., 2011), based on the following components (Srinivas & Patnaik, 1994):

- A genetic representation for the feasible solutions to the optimization problem
- A population of encoded solutions
- A fitness function that evaluates the optimality of each solution
- Generic operators that produce a new population from the existing population
- Control parameters

A population of solutions is maintained and a reproductive process allows parent solutions to be selected from the population. A *crossover* operator recombines portions from parent solutions to produce offspring, whereas a *mutation* operator maintains genetic diversity among solutions, preventing the GA to converge to a local minimum.

After using *crossover* and *mutation* on the initial population, the resulting offspring solutions exhibit some of the characteristics of each parent. The population components are then evaluated based on a given fitness function (in this case, the total distance traveled). Analogously to biological processes, the offspring with relatively good fitness levels are more likely to survive and reproduce, with the expectation that fitness levels throughout the population will improve as it evolves. Each solution is therefore composed of an array of numbers in which each number represents one of the targets in the route (the genes).

The potential of genetic algorithms (GAs) can be easily perceived in the wide use among scholars in such fields as Control engineering (Thomas & Poongodi, 2009), Economics (Chiang, 2005), Medicine (Kosakovsky Pond, Posada, Gravenor, Woelk, & Frost, 2006), Mechanical engineering (Bernardino, Barbosa, & Lemonge, 2007), etc. In the case of route optimization, GAs have several applications such as ship routing with time deadlines (Karlaftis, Kepaptsoglou, & Sambracos, 2009), vehicle routing with time windows (Baker & Ayechew, 2003; Garcia-Najera & Bullinaria, 2011; X. Wang & Regan, 2002), and VRP with loading constraints (Ruan, Zhang, Miao, & Shen, 2013). Although GAs are thus present in many route optimization problems, however, the TSP with dynamic targets is still an underdeveloped issue where GAs have hardly arrived.

GAs can nevertheless make an important contribution also in dynamic scenarios. To be sure, their near-optimal solution to a typical TSP problem is necessarily different from the one required when targets move. As explained above, however, GAs show an evolutionary behavior that allows to converge relatively fast towards a robust solution (Bjarnadottir, 2004; Jih & Hsu, 2004). Coherently, the GA used here can be fed with the trajectory prediction of FADs in order to identify the best route that each vessel within the fleet must follow.
In this paper, the specific properties of the GA reflected the need to minimize the distance that the vessels traveled throughout the recovering process. We summarize these properties below and, in the following subsections, provide a brief report on the algorithm design:

- Population size: $1,200$.
- Natural-selection mechanism: tournament selection.
- Tournament size: $1,200$ individuals selected in couples.
- Crossover type: two-part chromosome approach with $p = 0.8$.
- Mutation type: simple *mutation* between two elements with $p = 0.2$.
- Fitness: total distance traveled by all the vessels.
- Stopping criteria: $3,000$ iterations with no any fitness improvement.

4.2.1 Solution encoding

Each solution of the mTSP is a vector divided into two parts:

$$(f_1, f_2, \ldots, f_n \mid z_1, z_2, \ldots, z_m) \tag{2}$$

The first part of the vector $(f_1, f_2, \ldots, f_n)$ represents all the targets (FADs) that must be visited, and the second part $(z_1, z_2, \ldots, z_m)$ shows how many targets each vessel must visit. Thus, n is the quantity of FADs to recover, and m is the number of vessels that need to pick up those n objects. Note that $z_1 + z_2 + \ldots + z_m = n$. Each FAD, therefore, will be a point on the route and will be represented by a number. For instance, $(9, 4, 7, 3, 5, 1, 2, 6, 8 \mid 3, 4, 2)$ is a possible solution for a scenario with nine FADs and three vessels. This solution vector means that the first vessel has to recover three objects, $[9, 4, 7]$; the second vessel has to gather four objects, $[3, 5, 1, 2]$; and the third vessel has to collect two objects, $[6, 8]$.

4.2.2 Initialization

The initial population is chosen randomly with the aim of covering the entire search space. A random set of $1,200$ initial solutions has been set, which covers the problem perfectly (Yang, 1997). The fitness of each solution is measured as the total distance that the vessels travel to recover all the FADs.

The same method used by Carter and Ragsdale (2006) is employed here to generate the initial population. The first part of each chromosome is a randomly generated permutation of n cities. The greedy solutions are subsequently generated by examining the present location of each (vessel) and then calculating the unassigned city (FAD) that is closest to each salesman. Once a city is assigned to the closest salesman, the process continues until all cities are assigned to a salesman. This gives the GA a good starting point in the search space and improves the final results.

Additionally, as the sum of the positive integers in the second part of the chromosome must equal n, the m gene values (x_i) for the second part of the chromosome is generated as a discrete uniform distribution of random numbers between 1 and $n - \sum_{k=1}^{i-1} x_k$, for $i = 1$ to m (i.e., the maximum value for each successive gene value is based on n and the sum of the previous values). These values are then randomly assigned to the genes in the second part of the chromosome.

4.2.3 Prediction

Using the inputs of n drifting objects with known values for the current location and previous two positions, the r future positions for each object are estimated through Newton's motion equation. At first, this prediction is performed only once. The value of r will depend of the number of targets n and vessels m. As n increases, the vessels will need more time to recover the objects, so r typically grows as n grows, and decreases when m grows. As the vessels' speed (s_i) grows, r becomes smaller. There must be a balance between n, m, s_i, and the number of future predictions (r) that we can calculate. If there were more predicted positions than the problem requires, the algorithm will only use the future positions that it really needs; however, r must be large enough to determine the solution.

$$\begin{pmatrix} f_1^t & f_1^{t-1} & f_1^{t-2} \\ f_2^t & f_2^{t-1} & f_2^{t-2} \\ \vdots & \vdots & \vdots \\ f_n^t & f_n^{t-1} & f_n^{t-2} \end{pmatrix} \rightarrow \begin{pmatrix} \hat{f}_1^{t+1} & \hat{f}_1^{t+2} & \cdots & \hat{f}_1^{t+r} \\ \hat{f}_2^{t+1} & \hat{f}_2^{t+2} & \cdots & \hat{f}_2^{t+r} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{f}_n^{t+1} & \hat{f}_n^{t+2} & \cdots & \hat{f}_n^{t+r} \end{pmatrix} \tag{3}$$

4.2.4 Fitness

The fitness measure is the total number of miles traveled by all the vessels; i.e., the total travel distance.

4.2.5 Selection

The tournament *selection* method of individuals is based on their performance. With an initial random set of 600 pairs, the couple-tournament is used for the mating selection, so each pair competes with the others. The 600 winners of each tournament are selected to be the parent in the next generation (or offspring). This selection pressure allows the GA to improve the population fitness through successive generations. The algorithm needs probability values for the *crossover* and *mutation* in order to do so. Once the *crossover* and selection are made the offspring size will be again $1,200$ individuals.

4.2.6 Crossover

In the *crossover* method, each offspring inherits characteristics from its parents. *Crossover* causes a structured, yet randomized exchange of genetic material between solutions, with the possibility that 'good' solutions can generate 'better' ones (Srinivas & Patnaik, 1994). For the mTSP, the *crossover* approach is different from the TSP due to the complexity of the problem, i.e., different vehicles recovering different objects. A two-part chromosome is commonly used, therefore, instead of a one-part chromosome.
The two-part chromosome technique, as the name implies, divides the chromosome into two parts. The first part with length n represents a permutation of n cities, whereas the second part with length m reflects the number of cities assigned to each salesman. The *crossover* operation for the two-part chromosome is separated into two independent operations. The first operation uses an ordered *crossover* operator, while the second operation uses an asexual *crossover* operator to ensure that the second part of the chromosome remains feasible.

The solution implemented here uses a two-part chromosome approach that improves the GA's search performance when solving the multiple TSP (Shuai, Bradley, Shoudong, & Dikai, 2013). This method minimizes the size of the search space by reducing the redundant candidate solutions. The parameters of the GA were tested initially with the same probability (0.5-0.5); then the values were changed up and down by 0.1 until an optimum was reached in 0.8 and 0.2, respectively, for *crossover* and *mutation*. Lower *crossover* probabilities and higher values of *mutation* reduced the exploration capability of the GA and increased the amount of disruption to each gene in the solution. Similarly, higher *crossover* probabilities and lower *mutation* probabilities eliminate good candidates because of too much exploration and too much gene disruption, thus yielding worse results.

4.2.7 Mutation

Mutation is a genetic operator used to maintain genetic diversity from one generation of genetic algorithm chromosomes to the next one. These random changes will gradually add some new characteristics to the population that could not be supplied by the crossover, preventing the algorithm to avoid a local minimum when the population is too similar. The *mutation* probability, as noted above, has been set at 0.2.
There are many *mutation* types; in the simplest form, only one chromosome is mutated (bit string *mutation*), but there are more complex approximations as well (Flip bit, Boundary, Gaussian, etc.). This paper uses the simplest type, swap *mutation*, to randomly exchange two elements of the route. Notice, however, that this *mutation* is only applied for the first part of the chromosome; the second part continues without changing.

4.2.8 Execution of the algorithm and stopping criterion

The execution of the proposed algorithm must be carried out in a centralized way because the final, unique solution must be transmitted to all m vessels at the same time. The vessels' operators can all therefore know exactly which FADs have been assigned to each vessel. The algorithm cannot be executed on board because this is a metaheuristic approach, so the final solution could yield a different solution for each vessel. For this reason, the algorithm should be executed only once, either in the main office or on just one of the vessels. Results would be subsequently communicated to all vessels.

The execution of the algorithm is halted when, after $3,000$ loops, there is no fitness improvement. This number of maximum loops has been set so that the execution time of the GAMTP solution can vary from 1 to 10 min, depending on the number of FADs and vessels (n and m). This time constraint actually represents a fairly quick response since the company will organize the vessels' work for a week or more each time.

The algorithm has been developed in C# Programming Language and integrated into the MSB software, which is used by tuna vessels to receive the information from their buoys. It has been developed in a decoupled way where both, the collecting method (NN, GA or GAMTP) and the GAMTP parameters, can be easily changed. The solution might be useful for any type of tuna fishing company, regardless of the number of vessels or FADs. Furthermore, the algorithm can be specialized for specific tasks such as prioritizing the recovery of particular targets, working with time windows or finding the optimal speed of each vessel, which saves even more fuel. Concerning the computational experiments, they were carried out on a Windows 10 operating system with an Intel Core i7 running at 2.7 GHz and 16 GB of RAM.

5 Implementing the GA with a multiple-trajectory prediction

Based on the inputs and techniques described, this subsection explains how an evolutionary algorithm can be combined with a prediction technique (in this case, Newton's motion equation) to improve results when targets are constantly moving.

The addition of prediction is a bold departure from the initial state of the genetic algorithm, which evolves through multiple intermediate solutions to reach a final route for each of the vessels. Accordingly, the method described here, GAMTP (Genetic Algorithm based in Multiple-Trajectory Prediction), evolves from scratch to provide a set of routes based on the FADs predicted movement, which feeds the GA through the fitness-calculation process.

Algorithm 1 shows the pseudo-code of our GAMTP solution, and Algorithm 2 details how the fitness is calculated for each solution.

Data: n FADs positions (f_i) at time, $t, t-1$, and $t-2$:

$$\begin{pmatrix} f_1^t & f_1^{t-1} & f_1^{t-2} \\ f_2^t & f_2^{t-1} & f_2^{t-2} \\ \vdots & \vdots & \vdots \\ f_n^t & f_n^{t-1} & f_n^{t-2} \end{pmatrix}$$

Data: m vessels positions: $(v_1, v_2, \ldots, v_m)$
Data: Vessels speed (s) and fishing time for each FAD
Result: Final route for each vessel
for $i \leftarrow 1$ ***to*** n **do**
    using input $(f_i^t, f_i^{t-1}, f_i^{t-2}) \rightarrow$ calculation of the r future positions of each target f_i: $(\hat{f}_i^{t+1}, \hat{f}_i^{t+2}, \ldots, \hat{f}_i^{t+r})$;
end
GA initialization: k first solutions calculated;
for $i \leftarrow 1$ ***to*** k **do**
    fitness(i)
end
while *Stopping criteria not reached* **do**
    parent selection;
    new offspring generation: k solutions calculated;
    for $i \leftarrow 1$ ***to*** k **do**
        fitness(i)
    end
end
final route = best fitness;

Algorithm 1: GAMTP algorithm.

In Algorithm 1, different FADs and vessels positions are used as inputs to calculate the quasi-optimal best route. The first step is to predict r future positions for each FAD movement, so that the first set of solutions is created. Hence, following the rules of GAs based on the fitness of each solution, the population evolves towards the final route. It is worth noting that the singularity of the proposal with regard to existing literature is in how the fitness is calculated, as explained in detail in Algorithm 2.

The fitness algorithm calculates how well each solution matches the predicted FADs locations, as calculated previously $(\hat{f}^{t+r})$; this prediction is combined with the vessels' speed and fishing times to approximate the real route that each vessel is going to travel in the dynamic FAD context.

In order to determine the fitness of a given solution it is necessary to calculate the distance traveled by each vessel. The variables z_0, z_i and z_f are used to select the FADs that each vessel has been assigned to recover. Based on their speed, each vessel will

Data: Solution input: $(f_1, f_2, \ldots, f_n \mid z_1, z_2, \ldots, z_m)$
Data: FAD position matrix:

$$\begin{pmatrix} f_1^t & \hat{f}_1^{t+1} & \hat{f}_1^{t+2} & \cdots & \hat{f}_1^{t+r} \\ f_2^t & \hat{f}_2^{t+1} & \hat{f}_2^{t+2} & \cdots & \hat{f}_2^{t+r} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{f}_n^t & \hat{f}_n^{t+1} & \hat{f}_n^{t+2} & \cdots & \hat{f}_n^{t+r} \end{pmatrix}$$

Data: Vessels position inputs: $(v_1, v_2, \ldots, v_m)$
Data: Vessels speed (s) and fishing time (the same for all vessels)
Result: fitness of the solution $\equiv$ total distance traveled
variables initialization: $z_0 = 1$;

```
for i ← 1 to m do
    v = v_i: initial position of the vessel i;
    r = 0, d_i = 0: time and distance equal to zero;
    z_f = (z_0 + z_i) − 1;
    for j ← z_0 to z_f do
        nt = time to recover f̂_j^{t+r} at s knots from position v;
        nd = distance traveled from position v to f̂_j^{t+r} at s knots;
        v = predicted position of f̂_j^{t+r}: update the vessel's position;
        r = r + nt+ fishing time: update the time;
        d_i = d_i + nd: update the distance;
    end
    z_0 = z_i + 1;
end
fitness = Σ_{i=1}^{m} d_i
```

Algorithm 2: Fitness algorithm.

recover its FADs at their estimated position, depending on their arrival time, which starts at t and evolves with $t+r$. Finally, the fitness of the solution is measured as the sum of the distance that all m vessels traveled ($\sum_{i=1}^{m} d_i$).

Note that we ignore a FAD movement whenever a vessel is traveling to recover it. The rest of the FADs, however, are assumed to continue moving while the vessels are traveling or fishing, as shown in the fitness calculation. The rationale of this mathematical simplification is to avoid calculating the collision vector from the vessel to the object when it is moving. The only challenge of the alternative approach has to do with the time calculation, which should improve the result of the final solution because it would be more accurate than the current time calculation.

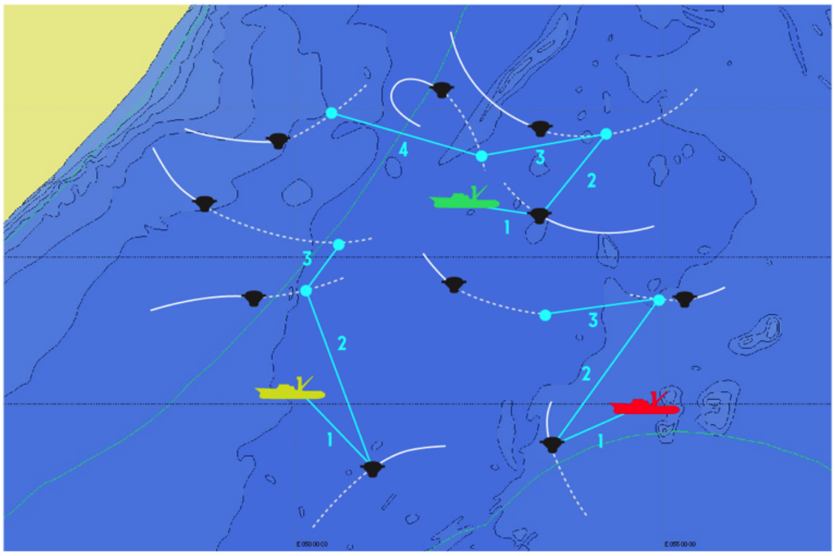

Figure 2: GAMTP algorithm procedure.

Figure 2 shows how GAMTP solution works. It includes the vessels and FADs initial positions at time t, as well as the final positions of each FAD. Figure 2 is an example of how the final fitness would be calculated, as the positions where the vessels recover the FADs depend on the time of recovery. By selecting one vessel as starting point, we can see that the first vector is to travel directly to the first FAD; however, the second iteration finishes at the FAD's expected location at time $t+x : x \in \{1, \ldots, r\}$. The same procedure holds for the rest of the objects.

Note that, when $f^t = f^{t-1} = f^{t-2}$, then $f^t = \hat{f}^{t+1} = \ldots = \hat{f}^{t+r}$. This means that, if there is no FADs movement, and if the fishing time is equal to zero, our GAMTP solution is identical to the classical GA approach for the mTSP. Thus, our contribution is a generalization of all the solutions, from the particular case when the targets are not moving and the vessels do not spend any time recovering targets, to the case where targets are moving and different fishing times exists. Thus, for the implementation of the classic mTSP based on GA, no target movement is assumed for the fitness calculation, so the resulting solution is compared to our GAMTP solution. This creates a static *vs.* dynamic comparison, through which it is easy to see that the total distance traveled using GAMTP and standard mTSP without prediction is very different.

6 Computational results

In this section we discuss the improvement achieved by addressing dynamic mTSP with the method proposed: GAs based on Multiple Trajectory Prediction (GAMTP). Initially we compare the Nearest Neighbor (NN) strategy, which is the method normally used by tuna vessels, with GAMTP. Then, we compare the performance of GAMTP method with mTSP solved only by genetic algorithms (GA), i.e., without prediction.

It is worth recalling that we use real data from different tuna fishing companies. In order to test our model, and with scientific purposes exclusively, Marine Instruments provided us with anonymous real data from several tuna vessels fishing in the FAO capture zone no. 57 (Eastern Indian Ocean) from April 9th to April 23rd. According to internal company records, in this area there are about 40 vessels operating at the same time. Taking into account a maximum retrieving of 14 FADs per vessel and week, it would be possible to collect a maximum of 1,120 FADs in this period of time. The sample used in our simulations consists of 150 randomly selected FADs using the MSB software (platform to receive and visualize the buoys data) from Marine Instruments. This quantity represents a percentage of 8.0% with a level of significance greater than 10% (sampling error of 7.45%). Regarding the working conditions of the simulations, they are the following:

- Vessels speed (average): 12 knots.
- FADs speed = FADs have different speeds, randing from 0.2 knots up to 2 knots.
- Fishing time: 3 h for each object.
- Number of vessels: 2, 3, and 4 (the typical range per group in practice).
- Number of FADs: from 20 to 36.
- Number of FADs/vessel: from 5 (20 FADs for 4 vessels) to 14 (28 FADs for 2 vessels).

Considering this scenario, the total distance traveled by the group of vessels is calculated based on both the number of vessels (two, three or four) and the number of FADs (from 20 to 36). We have performed 10 measurements in each experiment, varying the positions of the FADs and the vessels, to obtain representative mean values for each case (Table A1 in Appendix describe the design of the simulation tests). Once the values for each of the experiments are obtained, the mean is calculated and results from GAMTP are compared with the NN strategy and GA (without prediction) respectively. The results (average, standard deviation and the improvement percentage achieved between each two methods) are shown in Table 2. We can observe that our GATP method is always better than the NN and GA for recovering from 20 to 36 FADs.

Table 2: Computational results.

			Total distance traveled (nautical miles)			Improvement Comparison	
Num. Vessels	**Num. FADs**		**NN**	**GA**	**GAMTP**	**GAMTP vs NN**	**GAMTP vs GA**
2	20	$\bar{x}$	12,084.7	10,764.4	10,211.6	15.5%	7.3%
		σ	2,118.7	1,601.2	1,384.6		
	24	$\bar{x}$	11,549.6	10,649.9	9,765.3	15.4%	8.8%
		σ	992.5	648.8	875.9		
	28	$\bar{x}$	13,600.9	12,597.8	11,778.4	13.4%	7.0%
		σ	975.8	901.6	567.5		
	32	$\bar{x}$	11,040.1	10,588.4	9,997.7	9.4%	5.6%
		σ	1100.5	891.2	603.7		
	36	$\bar{x}$	15,121.4	15,172.3	13,942.8	7.8%	8.1%
		σ	1713.0	1050.9	590.1		
3	20	$\bar{x}$	13,876.2	11,310.6	10,918.4	21.4%	5.1%
		σ	2,011.0	535.8	507.4		
	24	$\bar{x}$	14,091.9	12,193.4	11,615.3	17.6%	6.2%
		σ	1,424.1	518.8	426.0		
	28	$\bar{x}$	13,505.8	11,858.3	11,357.3	15.9%	5.0%
		σ	3,024.6	2,497.3	2,403.0		
	32	$\bar{x}$	12,273.5	11,367.6	10,446.8	14.9%	8.1%
		σ	1,178.3	605.1	884.2		
	36	$\bar{x}$	11,142.6	10,418.8	9,628.5	13.6%	7.6%
		σ	989.2	667.8	761.0		
4	20	$\bar{x}$	13,197.0	10,279.2	9,662.6	26.8%	6.7%
		σ	1,364.0	425.6	488.4		
	24	$\bar{x}$	13,666.9	11,202.4	10,506.6	23.2%	7.0%
		σ	1,742.6	342.2	555.2		
	28	$\bar{x}$	15,526.4	13,151.2	12,381.4	20.2%	5.9%
		σ	994.9	464.7	490.6		
	32	$\bar{x}$	13,897.0	12,572.3	11,710.7	15.7%	6.9%
		σ	1,412.1	519.4	528.6		
	36	$\bar{x}$	13,607.7	12,260.1	11,552.2	15.1%	5.8%
		σ	980.4	618.3	477.4		

These results are supported statistically (Table 3 and 4). The comparison has been performed using a repeated-measures analysis of variance (repeated-measures ANOVA) for each experiment. Repeated-measures ANOVA must be used when, as here, we analyze differences in mean scores under three or more different conditions. One of the main conditions with these study designs is that the same individuals (vessels here) are being measured more than once on the same dependent variable (i.e. why it is called repeated measures) and the independent variable has different categories (here, the number of vessels per group). According to this, Table 3 shows significant differences among the methods used (Sig. $= 0.000 < 0.05$), and the results are consistent since the four most widely used tests for ANOVA (Phillai's trace, Wilks' Lambda, etc.) all indicate a very high significance. Secondly, getting deeper into how the means of the three methods are significantly different, Table 4 supports our descriptive analysis based on a Pairwise comparison: the average GAMTP distances are statistically lower than those obtained with NN and GA for the three group of vessels.

In brief, the GAMTP strategy yields better results for all experiments (two, three, and four vessels) relative to other common optimizing strategies (Figure 3). We prove

Table 3: Multivariate test (ANOVA)

Experiment	Type	Effect	Value	F	Hipothesis df	Error df	Sig.
Exp. 1:2 vessels	Method	Pillai's Trace	0.884	98.597	2.00	26.00	.000
		Wilks' Lambda	0.116	98.597	2.00	26.00	.000
		Hotelling's Trace	7.584	98.597	2.00	26.00	.000
		Roy's Largest Root	7.584	98.597	2.00	26.00	.000
Exp. 2:3 vessels	Method	Pillai's Trace	0.845	70.924	2.00	26.00	.000
		Wilks' Lambda	0.155	70.924	2.00	26.00	.000
		Hotelling's Trace	5.456	70.924	2.00	26.00	.000
		Roy's Largest Root	5.456	70.924	2.00	26.00	.000
Exp. 3:4 vessels	Method	Pillai's Trace	0.866	83.682	2.00	26.00	.000
		Wilks' Lambda	0.134	83.682	2.00	26.00	.000
		Hotelling's Trace	6.437	83.682	2.00	26.00	.000
		Roy's Largest Root	6.437	83.682	2.00	26.00	.000

Table 4: Pairwise comparison (ANOVA)

Num. Experiment	Method (a)	Method (b)	Mean Diff. (a-b)	Std. error	Sig.	95% Confidence Interval	
						Lower Bound	Upper Bound
Exp. 1:2 vessels	NN	GA	1,074.37	206.848	.000	546.424	1,602.363
		GAMTP	1,941.233	171.268	.000	1,504.068	2,378.378
	GA	NN	-1,074.37	206.848	.000	-1,602.363	-546.424
		GAMTP	866.867	106.990	.000	593.742	1,139.916
	GAMTP	NN	-1,941.233	171.268	.000	-2,378.378	-1,504.068
		GA	-866.867	106.990	.000	-1,139.916	-593.742
Exp. 2:3 vessels	NN	GA	2,037.196	267.707	.000	1,353.884	2,720.508
		GAMTP	2,650.947	281.720	.000	1,931.869	3,370.025
	GA	NN	-2,037.196	267.707	.000	-2,720.508	-1,353.884
		GAMTP	613.751	59.118	.000	462.854	764.648
	GAMTP	NN	-2,650.947	281.720	.000	-3,370.025	-1,931.869
		GA	-613.751	59.118	.000	-764.648	-462.854
Exp. 3:4 vessels	NN	GA	2,585.811	234.998	.000	1,985.989	3,185.634
		GAMTP	3,334.864	256.734	.000	2,679.565	3,990.172
	GA	NN	-2,585.811	234.998	.000	-3,185.634	-1,985.989
		GAMTP	749.057	102.720	.000	486.868	1,011.246
	GAMTP	NN	-3,334.868	256.734	.000	-3,990.172	-2,679.565
		GA	-749.057	102.720	.000	-1,011.246	-486.868

therefore that integrating forecasting within a metaheuristic method, such as genetic algorithms, can yield better results than their simple non-predictive version in a dynamic scenario. Undoubtedly, however, to achieve even better results, the prediction method should be adapted to the specific characteristics of each environment. Here, for example, improving the accuracy of long-term forecasting for FAD trajectories would require a prediction technique that contemplates the ocean currents, winds or the Coriolis Effect, among other factors. In fact, the results comparison suggests that the relative improvement decreases as the collection time increases (i.e., the ratio FADs/vessel).

Number of FADs collected by vessel (FADs/Vessel)

Vessels | FADs	20	24	28	32	36
2	10	12	14	16	18
3	7	8	9	11	12
4	5	6	7	8	9

First comparison: GAMTP vs NN (a)

Vessels | FADs	20	24	28	32	36
2	15.5%	15.4%	13.4%	9.4%	7.8%
3	21.4%	17.6%	15.9%	14.9%	13.6%
4	26.8%	23.2%	20.2%	15.7%	15.1%

Second comparison: GAMTP vs GA (b)

Vessels | FADs	20	24	28	32	36
2	7.3%	8.8%	7.0%	5.6%	8.1%
3	5.1%	6.2%	5.0%	8.1%	7.6%
4	6.7%	7.0%	5.9%	6.9%	5.8%

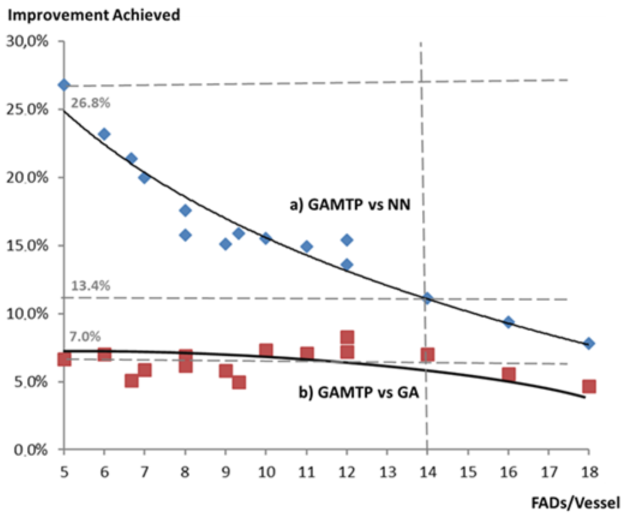

Figure 3: GAMPT improvements evolution as the ratio of FADs/vessel increases (built from Table 2).

Figure 3 describes this situation. First, from Table 2, the FADs/vessel ratio and improvement reached for each experiment were calculated (tables on the left); then these values were ordered according to this ratio and plotted. Figure 3 shows how lower collection rates (5 FADs/vessel) result in GAMTP results being as much as 26% better than NN results, and simultaneously verifies that for higher working ratios (14 FADs/vessel), this improvement is reduced up to 13%. A similar behavior shows the comparison with GA. Even in those simulations with 16 or 18 FADs/vessel (32 and 36 FADs for 2 vessels) results indicate that the GAMPT could reach a worse performance than the procedure without prediction (Figure 3). While this industry never reaches ratios as high as 18 FADs per vessel in this period of time, these simulations allow us to enrich and understand better the limitations of this proposal. This is due to the weakness of Newton's motion equation when making predictions in the long term, which obviously aggravates results as the number of FADs that each vessel retrieves increases. This reinforces the importance of prediction in the algorithm: although GAMTP yields better results than common strategies, the improvements decrease as the number of FADs increases.

7 Conclusions

This paper proposes a new method to address the multiple traveling salesman problem with moving targets (mTSP-MT). Unlike previous analyses, where the focus was on ad hoc quasi-experiments, moving targets are addressed here using an unrestricted generic solution that combines a metaheuristic method with a predictive technique. Particularly, the method first estimates the trajectory of each target using Newton's motion equation to feed then a GA that searches for the optimal route based on the total distance traveled by all salesmen. Hence the name "Genetic Algorithm based in Multiple Trajectory Prediction" (GAMTP). Based on historical GPS data for tuna fishing FADs, results show that the total distance traveled is always shorter with GAMTP than with other common methods used by ship-owning companies, such as NN or GAs without prediction. Integrating forecasting within a metaheuristic method (e.g., genetic algorithms) therefore yields better results than in the simple non-predictive version.

From an academic perspective, GAMTP can be seen as a generalization of classic approaches for solving the mTSP problem. With static objects, the predicted next position is the same in GAMTP and GA. However, in dynamic scenarios with moving objectives, GAMTP diverges because it evolves and continues optimizing the route by considering the future movement of each target. GAMTP could thus be used generically (in static and dynamic situations), as a suitable solution to obtain near-optimal routes.

In general terms, therefore, the paper opens up a set of possibilities for a wide range of real-world situations. GAMTP allows not just tuna vessels but any group of agents following moving targets (e.g., UAVs, autonomous devices and surveillance vehicles) to minimize the total distance traveled. Furthermore, within each of these possible fields of utilization, the algorithm could be specialized for specific tasks, such as prioritizing the recovery of particular targets (i.e., the ones that have more tuna beneath them), working with certain time windows for FAD recovery (e.g., tuna vessels do not fish at night), or finding the optimal speed of each vessel, which saves even more fuel. This should also open a new space for research. Furthermore, other interesting adaptations could be found in static scenarios with fixed destinations but where forecasting is especially relevant (e.g. situations where wind, currents, traffic jams, etc., are a critical factor). Similarly it is worth noting that GAMTP would allow to address the new potential uses that are emerging in mobility (e.g., delivery services, real-time mobility requirements, drones scheduling and collaboration, etc.) conducted by companies such as Uber and Amazon.

From an economic point of view, GAMTP can produce a considerable reduction in fleets' direct costs. In a typical situation with a fleet of 3 vessels, each working 24 h/day for 5 days, consumes approximately 1,000 USD/h (Parker et al., 2015). According to this, the average savings per campaign would be up to 64,800 USD. In addition, the reduction of the distance traveled not only affects the consumption of fuel with the estimations just shown, but also facilitates a more efficient utilization rate of the vessels thanks to a faster recovery process. Traveling is waste from an operational excellence perspective, whereas fishing, the output to maximize, is directly related to the number of FADs recoveries. It is worth noting, however, that given the chaotic nature of ocean currents and despite of the sophistication of the forecasting method, the GAMTP solutions add less value for long-term predictions. In fact, as shown in Figure 3, lower collection rates (e.g. 5 FADs/vessel) have greater improvements over standard techniques than do higher ratios (18 FADs/vessel).

Finally, taking into account the urge to stimulate sustainable business operations, the use of the GAMTP strategy in fishing companies would directly reduce CO_2 emissions by and average of 18% at current rates (considering the distance saved by a tuna vessel). This is undoubtedly a very significant improvement given that climate change represents one of the main challenges for humanity today (Howard-Grenville, Buckle, Hoskins, & George, 2014).

Acknowledgements

The authors wish to thank Marine Instruments for the buoys data. We also thank Iago Paz and participants at ACEDE 2018 for several useful suggestions. This research benefited from the financial support of the Spanish and Galician Governments through -respectively- grants ECO2016-76625-R and GRC2014/021.

References

Agatz, N., Bouman, P., & Schmidt, M. (2016). Optimization approaches for the traveling salesman problem with drone. *Transportation Science*. doi: http://dx.doi.org/10.2139/ssrn.2639672

Alonso, F., Oliveira, D., & de Souza, A. Z. (2015). Artificial immune systems optimization approach for multiobjective distribution system reconfiguration. *IEEE Transactions on Power Systems*, *30*(2), 840–847.

Bae, J., & Chung, W. (2017). A heuristic for a heterogeneous automated guided vehicle routing problem. *International Journal of Precision Engineering and Manufacturing*, *18*(6), 795–801.

Baker, B. M., & Ayechew, M. (2003). A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, *30*(5), 787–800.

Banerjee, S. (2017). An artificial immune system approach to automated program verification: Towards a theory of undecidability in biological computing. *PeerJ Preprints*, *5*, e2690v1.

Bektas, T. (2006a). The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, *34*(3), 209 - 219. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0305048304001550` doi: http://dx.doi.org/10.1016/j.omega.2004.10.004

Bektas, T. (2006b). The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, *34*(3), 209–219.

Bernardino, H. S., Barbosa, H. J., & Lemonge, A. C. (2007). A hybrid genetic algorithm for constrained optimization problems in mechanical engineering. In *Evolutionary computation, 2007. cec 2007. ieee congress on* (pp. 646–653).

Bjarnadottir, Á. S. (2004). Solving the vehicle routing problem with genetic algorithms. *Ph.D Thesis, Technical University of Denmark, Denmark*.

Blum, C., & Roli, A. (2003a). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, *35*(3), 268–308.

Blum, C., & Roli, A. (2003b, September). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.*, *35*(3), 268–308.

Braekers, K., Ramaekers, K., & Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, *99*, 300–313.

Carter, A. E., & Ragsdale, C. T. (2006). A new approach to solving the multiple traveling salesperson problem using genetic algorithms. *European Journal of Operational Research*, *175*(1), 246 - 257. Retrieved from http://www.sciencedirect.com/science/article/pii/S0377221705004236 doi: http://dx.doi.org/10.1016/j.ejor.2005.04.027

Castro, J. J., Santiago, J. A., & Santana-Ortega, A. T. (2001). A general theory on fish aggregation to floating objects: an alternative to the meeting point hypothesis. *Reviews in fish biology and fisheries*, *11*(3), 255–277.

Cattaruzza, D., Absi, N., Feillet, D., & González-Feliu, J. (2017). Vehicle routing problems for city logistics. *EURO Journal on Transportation and Logistics*, *6*(1), 51–79.

Chiang, C.-L. (2005). Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels. *IEEE transactions on power systems*, *20*(4), 1690–1699.

Deng, Y., Liu, Y., & Zhou, D. (2015). An improved genetic algorithm with initial population strategy for symmetric tsp. *Mathematical Problems in Engineering*, *2015*.

Dorling, K., Heinrichs, J., Messier, G. G., & Magierowski, S. (2017). Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *47*(1), 70–85.

Du, W., & Li, B. (2008). Multi-strategy ensemble particle swarm optimization for dynamic optimization. *Information sciences*, *178*(15), 3096–3109.

Eksioglu, B., Vural, A. V., & Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, *57*(4), 1472–1483.

Fonteneau, P., & Pianet. (2000). A worldwide review of purse seine fisheries on fads.

Food and Agriculture Organization of the United nations, FAO. (2012). The state of world fisheries and aquaculture 2012. *FAO, Rome 2012*.

Garcia-Najera, A., & Bullinaria, J. A. (2011). An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. *Computers &*

Operations Research, *38*(1), 287–300.

Garey, M. R., & Johnson, D. S. (1983). Crossing number is np-complete. *SIAM Journal on Algebraic Discrete Methods*, *4*(3), 312–316.

Geng, L., Zhang, Y., Wang, J., Fuh, J. Y., & Teo, S. (2014). Cooperative mission planning with multiple uavs in realistic environments. *Unmanned Systems*, *2*(01), 73–86.

Groba, C., Sartal, A., & Vázquez, X. H. (2015). Solving the dynamic traveling salesman problem using a genetic algorithm with trajectory prediction: An application to fish aggregating devices. *Computers & Operations Research*, *56*, 22–32.

Hajjam, A., Créput, J. C., & Koukam, A. (2013). From the tsp to the dynamic vrp: An application of neural networks in population based metaheuristic. *In Metaheuristics for Dynamic Optimization*, 309–339.

Hallier, J.-P., & Gaertner, D. (2008). Drifting fish aggregation devices could act as an ecological trap for tropical tuna species. *Marine Ecology Progress Series*, *353*, 255–264.

Helbing, D., & Tilch, B. (1998, Jul). Generalized force model of traffic dynamics. *Phys. Rev. E*, *58*, 133–138. Retrieved from https://link.aps.org/doi/10.1103/PhysRevE.58.133 doi: 10.1103/PhysRevE.58.133

Helvig, C. S., Robins, G., & Zelikovsky, A. (2003). The moving-target traveling salesman problem. *Journal of Algorithms*, *49*(1), 153–174.

Holland, J. (1975). Adaptation in natural and artificial systems, university of michigan press. *Ann Arbor, MI*, *1*(97), 5.

Holland, J. (1992). Genetic algorithms. *Scientific american*, *267*(1), 66–73.

Howard-Grenville, J., Buckle, S. J., Hoskins, B. J., & George, G. (2014). Climate change and management. *Academy of Management Journal*, *57*(3), 615–623.

Jiang, Q., Sarker, R., & Abbass, H. (2005). Tracking moving targets and the non-stationary traveling salesman problem. *Complexity International*, *11*(2005), 171–179.

Jih, W.-R., & Hsu, Y. (2004). A family competition genetic algorithm for the pickup and delivery problems with time window. *Bulletin of the College of Engineering, National Taiwan University*, *90*, 121–130.

Jindal, P., & Kumar, A. (2011). Multiple target intercepting traveling salesman problem 1.

Karlaftis, M. G., Kepaptsoglou, K., & Sambracos, E. (2009). Containership routing with time deadlines and simultaneous deliveries and pick-ups. *Transportation Research Part E: Logistics and Transportation Review*, *45*(1), 210–221.

Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of computer computations* (pp. 85–103). Springer.

Kosakovsky Pond, S. L., Posada, D., Gravenor, M. B., Woelk, C. H., & Frost, S. D. (2006). Gard: a genetic algorithm for recombination detection. *Bioinformatics*, *22*(24), 3096–3098.

Kuo, R., & Zulvia, F. E. (2017). Hybrid genetic ant colony optimization algorithm for capacitated vehicle routing problem with fuzzy demand?a case study on garbage

collection system. In *Industrial engineering and applications (iciea), 2017 4th international conference on* (pp. 244–248).
Li, T., Shao, G., Zuo, W., & Huang, S. (2017). Genetic algorithm for building optimization: State-of-the-art survey. In *Proceedings of the 9th international conference on machine learning and computing* (pp. 205–210).
Liu, C.-H. (2013). *The moving-target traveling salesman problem with resupply* (Tech. Rep.). Technical report, The National Chung Cheng University Library. http://ccur. lib. ccu. edu. tw/handle/987654321/7877.
Liu, L., Wang, D., & Yang, S. (2009). An immune system based genetic algorithm using permutation-based dualism for dynamic traveling salesman problems. *In Applications of Evolutionary Computing*, 725–734.
Lynn, N., & Suganthan, P. N. (2017). Ensemble particle swarm optimizer. *Applied Soft Computing*, *55*, 533–548.
Mavrovouniotis, M., & Yang, S. (2013). Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors. *Applied Soft Computing*, *13*(10), 4023–4037.
Menezes, M. B., Ketzenberg, M., Oliva, R., & Metters, R. (2015). Service delivery to moving demand points using mobile servers. *International Journal of Production Economics*, *168*, 158–166.
Menezes, T., Tedesco, P., & Ramalho, G. (2006). Negotiator agents for the patrolling task. *Advances in Artificial Intelligence-IBERAMIA-SBIA 2006*, 48–57.
Miyake, M., Guillotreau, P., Sun, C., & Ishimura, G. (2010). Recent developments in the tuna industry. *FAO, Rome 2012*.
Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F., & Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, *79*, 115–129.
Moreno, G., Dagorn, L., Sancho, G., & Itano, D. (2007). Fish behaviour from fishers' knowledge: the case study of tropical tuna around drifting fish aggregating devices (dfads). *Canadian Journal of Fisheries and Aquatic Sciences*, *64*(11), 1517–1528.
Osman, I. H., & Laporte, G. (1996). *Metaheuristics: A bibliography.* Springer.
Özgökmen, T. M., Griffa, A., Mariano, A. J., & Piterbarg, L. I. (2000). On the predictability of lagrangian trajectories in the ocean. *Journal of Atmospheric and Oceanic Technology*, *17*(3), 366–383.
Parker, R. W., Vázquez-Rowe, I., & Tyedmers, P. H. (2015). Fuel performance and carbon footprint of the global purse seine tuna fleet. *Journal of Cleaner Production*, *103*, 517–524.
Potvin, J.-Y. (1996). Genetic algorithms for the traveling salesman problem. *Annals of Operations Research*, *63*(3), 337–370.
Pushkarini Agharkar, S. D. B., & Bullo, F. (2015). Vehicle routing algorithms for radially escaping targets. *SIAM Journal on Control and Optimization*, *53*, 21.
Rajeswari, G. (2009). *Fish aggregating devices.* Central Institute of Fisheries Technology.
Razali, N. M., Geraghty, J., et al. (2011). Genetic algorithm performance with different selection strategies in solving tsp. In *Proceedings of the world congress on engineering*

(Vol. 2, pp. 1134–1139).

Reinelt, G. (1994). *The traveling salesman: computational solutions for tsp applications.* Springer-Verlag.

Ruan, Q., Zhang, Z., Miao, L., & Shen, H. (2013). A hybrid approach for the vehicle routing problem with three-dimensional loading constraints. *Computers & Operations Research*, *40*(6), 1579–1589.

Shetty, V. K., Sudit, M., & Nagi, R. (2008). Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles. *Computers & Operations Research*, *35*(6), 1813–1828.

Shima, T., & Schumacher, C. (2005). Assignment of cooperating uavs to simultaneous tasks using genetic algorithms. In *Proc. aiaa guidance, navigation, and control conference and exhibit. san francisco.*

Shuai, Y., Bradley, S., Shoudong, H., & Dikai, L. (2013). A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. *European Journal of Operational Research*, *228*(1), 72 - 82. Retrieved from http://www.sciencedirect.com/science/article/pii/S0377221713000908 doi: http://dx.doi.org/10.1016/j.ejor.2013.01.043

Smith, G. C., & Smith, S. (2002). An enhanced genetic algorithm for automated assembly planning. *Robotics and Computer-Integrated Manufacturing*, *18*(5), 355–364.

Song, C.-H., Lee, K., & Lee, W. D. (2003). Extended simulated annealing for augmented tsp and multi-salesmen tsp. In *Neural networks, 2003. proceedings of the international joint conference on* (Vol. 3, pp. 2340–2343).

Srinivas, M., & Patnaik, L. M. (1994). Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, *24*(4), 656–667.

Stieber, A., & Fügenschuh, A. (2017). The multiple traveling salesmen problem with moving targets and nonlinear trajectories.

Stieber, A., Fügenschuh, A., & Yuan, Z. (2015). *School taxi routing for children with special needs.* Helmut-Schmidt-Univ., Univ. der Bundeswehr Hamburg.

Sundar, K., Venkatachalam, S., & Rathinam, S. (2017). Analysis of mixed-integer linear programming formulations for a fuel-constrained multiple vehicle routing problem. *Unmanned Systems*, 1–11.

Thomas, N., & Poongodi, D. P. (2009). Position control of dc motor using genetic algorithm based pid controller. In *Proceedings of the world congress on engineering* (Vol. 2, pp. 1–3).

Toth, P., & Vigo, D. (2014). *Vehicle routing: problems, methods, and applications.* SIAM.

Venkatesh, P., & Singh, A. (2015). Two metaheuristic approaches for the multiple traveling salesperson problem. *Applied Soft Computing*, *26*, 74 - 89. Retrieved from http://www.sciencedirect.com/science/article/pii/S1568494614004827 doi: http://dx.doi.org/10.1016/j.asoc.2014.09.029

Wang, H.-f., Wang, D.-w., & Yang, S.-x. (2007). Evolutionary algorithms in dynamic environments. *Control and Decision*, *22*(2), 127.

Wang, X., & Regan, A. C. (2002). Local truckload pickup and delivery with hard time window constraints. *Transportation Research Part B: Methodological*, *36*(2), 97–112.

Winter, G., Periaux, J., Galan, M., & Cuesta, P. (1996). *Genetic algorithms in engineering and computer science.* John Wiley & Sons, Inc.

Yang, R. (1997). Solving large travelling salesman problems with small populations. *Second International Conference on Genetic Algorithms in Engineering Systems*, 157–162.

Yegnanarayana, B. (2009). *Artificial neural networks.* PHI Learning Pvt. Ltd.

Zhou, A., Kang, L., & Yan, Z. (2003). Solving dynamic tsp with evolutionary approach in real time. In *Evolutionary computation, 2003. cec'03. the 2003 congress on* (Vol. 2, pp. 951–957).

A Appendix

Table A1: Experiment design

Num. Vessels	Num. FADs	Num. Tests	NN	mTSP-GA	GAMTP
		Test 1	Total distance traveled 1	Total distance traveled 1	Total distance traveled 1
2	20	...	...	...	...
		Test 10	Total distance traveled 10	Total distance traveled 10	Total distance traveled 10
		Test 1	Total distance traveled 1	Total distance traveled 1	Total distance traveled 1
2	24	...	...	...	...
		Test 10	Total distance traveled 10	Total distance traveled 10	Total distance traveled 10
		Test 1	Total distance traveled 1	Total distance traveled 1	Total distance traveled 1
2	28	...	...	...	...
		Test 10	Total distance traveled 10	Total distance traveled 10	Total distance traveled 10
		Test 1	Total distance traveled 1	Total distance traveled 1	Total distance traveled 1
2	32	...	...	...	...
		Test 10	Total distance traveled 10	Total distance traveled 10	Total distance traveled 10
		Test 1	Total distance traveled 1	Total distance traveled 1	Total distance traveled 1
2	36	...	...	...	...
		Test 10	Total distance traveled 10	Total distance traveled 10	Total distance traveled 10
		Test 1	Total distance traveled 1	Total distance traveled 1	Total distance traveled 1
3	20	...	...	...	...
		Test 10	Total distance traveled 10	Total distance traveled 10	Total distance traveled 10
		Test 1	Total distance traveled 1	Total distance traveled 1	Total distance traveled 1
3	24	...	...	...	...
		Test 10	Total distance traveled 10	Total distance traveled 10	Total distance traveled 10
		Test 1	Total distance traveled 1	Total distance traveled 1	Total distance traveled 1
3	28	...	...	...	...
		Test 10	Total distance traveled 10	Total distance traveled 10	Total distance traveled 10
		Test 1	Total distance traveled 1	Total distance traveled 1	Total distance traveled 1
3	32	...	...	...	...
		Test 10	Total distance traveled 10	Total distance traveled 10	Total distance traveled 10
		Test 1	Total distance traveled 1	Total distance traveled 1	Total distance traveled 1
3	36	...	...	...	...
		Test 10	Total distance traveled 10	Total distance traveled 10	Total distance traveled 10
		Test 1	Total distance traveled 1	Total distance traveled 1	Total distance traveled 1
4	20	...	...	...	...
		Test 10	Total distance traveled 10	Total distance traveled 10	Total distance traveled 10
		Test 1	Total distance traveled 1	Total distance traveled 1	Total distance traveled 1
4	24	...	...	...	...
		Test 10	Total distance traveled 10	Total distance traveled 10	Total distance traveled 10
		Test 1	Total distance traveled 1	Total distance traveled 1	Total distance traveled 1
4	28	...	...	...	...
		Test 10	Total distance traveled 10	Total distance traveled 10	Total distance traveled 10
		Test 1	Total distance traveled 1	Total distance traveled 1	Total distance traveled 1
4	32	...	...	...	...
		Test 10	Total distance traveled 10	Total distance traveled 10	Total distance traveled 10
		Test 1	Total distance traveled 1	Total distance traveled 1	Total distance traveled 1
4	36	...	...	...	...
		Test 10	Total distance traveled 10	Total distance traveled 10	Total distance traveled 10