Information Technology & Tourism (2022) 24:265–298
https://doi.org/10.1007/s40558-022-00224-x

ORIGINAL RESEARCH

Automatic generation of sailing holiday itineraries using vessel density data and semantic technologies

Andreas Komninos[1,2] · Charalampos Kostopoulos[3] · John Garofalakis[1,2]

Received: 21 September 2021 / Revised: 18 February 2022 / Accepted: 26 February 2022 /
Published online: 15 March 2022

Abstract

Sailing holiday activities represent a significant portion of the Blue Economy growth in Europe and across the world. Due to the global financial crisis, yacht ownership has declined, but demand for such holiday products remained steady, therefore shifting the yachters profile towards younger and less experienced consumers who prefer to charter boats, rather than own one. Boat chartering offers more flexibility to explore different regions from year to year, but this means that significantly more time must be spent planning the route, since local experience is absent. The tourists' experience during the initial contemplation and planning phase, taking place weeks or months before an actual trip, and where a broad range of route options needs to be explored, could thus significantly benefit from support given by automated IT tools. Current literature demonstrates a complete lack of research in the development of itinerary recommendation systems in the context of sailing holidays. In this paper, we describe a methodology for the automatic generation of route recommendations, based on the semantic modelling of spatial data, and the determination of realistic sea route options, based on vessel density maps produced from raw AIS data. We demonstrate the implementation and results from this methodology using one of the most popular sailing regions of Greece, namely the Ionian Sea, as a case study.

Keywords Maritime trip planning · Vessel routing · Route planning · Itinerary recommendation · Genetic algorithms · Semantic spatial modelling

✉ Andreas Komninos
akomninos@ceid.upatras.gr

Charalampos Kostopoulos
info@optionsnet.gr

John Garofalakis
garofala@ceid.upatras.gr

[1] Computer Engineering and Informatics Department, University of Patras, 26504 Rio, Greece

[2] Diophantos Computer Technology Institute and Press, 26504 Rion, Greece

[3] OptionsNet IT Services and Consulting, Optionsnet, 121 Maizonos St., 26221 Patras, Greece

1 Introduction

One of the key components of the "Blue Economy", an expressed strategic priority of the European Union, is coastal tourism, which represents over 64% of the total workforce in the EU blue economy (2.8m persons), and contributes to more than 45% of blue economy's added value, at approximately €80bn.[1] The recreational boating industry, which employs an estimated 0.28m persons in 32,000 companies in boat manufacturing and tourism services, including marinas and boat charters, is an important sub-sector of coastal tourism. Compared to touring a region by car, cruise ship or island-hopping via ferry, sailing holidays are a low-impact and more sustainable mode of exploring coastal areas (Eijgelaar et al. 2021). It is estimated that European waters are home to over 6m boats and 36m boaters of European citizenship, travelling between over 10,000 marinas, while the direct economic impact reaches €28bn in total, of which €4bn in the marina industry and €6bn in the charter boat industry (European Commission 2020). The sector has a range of economic impacts in other areas related to coastal tourism, including accommodation, catering, recreational activities and shopping, as well as parts of the value chain that include repair and maintenance, dry berthing, training, retail of supplies and accessories and insurance (Ecorys 2015).

Boat charters are increasingly important in this sector, since the economic crisis of 2008 has drastically reduced private boat ownership and shifted owners' average age by at least 10 years (from 45 to 55). Crucially, interest in sailing holidays has remained consistently high, shifting the industry from reliance on boat owners, to consumers of sailing holidays as a service. The average age of the youngest such consumers is 31 years old, and the boat charter market is equally spread between bareboat (boat only, where at least one member of the renting party has a skipper's license) and crewed (boat, skipper and, optionally, other staff) rentals. This means that a large, if not the largest proportion of recreational boat skippers on such holidays are young and inexperienced seafarers. The versatility of renting vs. owning a boat also means that such holiday makers are not constrained to the geographical confines of specific regions, but can find themselves exploring unfamiliar waters from year to year and from country to country. Unsurprisingly, given these user-base characteristics, there is a high demand for the experience to be supported through innovative IT tools, including online booking and smartphone applications to manage the rental experience (Ecorys 2015).

Yacht tourism, and by extension its planning phase, is reported to be predominantly based on an extensive skillset which is developed only through experience, such as exposure to new areas and challenges, discussions with other sailors (Andersson 2007). Such skills can be supplemented or even developed through use of IT (e.g. reading other sailors' experiences and obtaining recommendations from online discussion forums), even though Ferrer-Rosell et al. (2017) found that pre-trip internet use for planning doesn't correlate with trip satisfaction. Although there

[1] Blue Economy Indicators Dashboard: https://blueindicators.ec.europa.eu/access-online-dashboard_en.

exists a range of applications to support the boat charter experience, such as online booking systems for boats and berth spaces, or various informational websites, the usage of such tools in the planning phase of a sailing trip itinerary remains obscure from a literature viewpoint. We haven't been able to locate any literature that examines tourists' behaviour during sailing holiday planning, however, anecdotal evidence in published articles related to marine tourism, indicate that planning relies on either advice from the charter company, the skipper (if one is booked) and personal research based on online information, such as, for example, basic sailing advice, sample itineraries and travel experiences from other tourists (Łapko 2019; Nualnim and Phuaksawat 2010; Pranita 2020; Strulak-Wójcikiewicz et al. 2020).

In reviewing, for the purposes of this work, a significant number of yachting holiday websites directed towards booking a yacht, booking berth spaces, or general destination information, we found more evidence that supports these insights. We noted the frequent presence of either pre-designed itineraries, as well as contact forms for a user to be able to input general information about their area of interest, type of boat and holiday dates, in order to receive a customised (manually composed) proposal in an asynchronous manner (i.e. after a few hours or days). There exist only a handful of computer-assisted sea route calculation tools. One such type of tools allows the manual design of itineraries, drawing line segments between a sequence of points on a map, without any automatic assistance for avoiding overland coordinates or automatic complex path generation (e.g. Sea-Seek[2]). Another type of tool, and probably the only one of its kind, allows the automated generation of complex route paths between a start and end point, that avoids land (e.g. Searoutes[3]). However, such tools do not solve the problem. Manual route design requires intimate knowledge of the sailing area to produce a *good* route under criteria such as safety, speed, ease of navigation etc. Automatic path calculation tools are able to produce a route without manual intervention, however the routes are not necessarily realistic for sailing purposes (e.g. they avoid close proximity to land, do not pass between islets etc.).

Finally, more evidence regarding the problems in trip planning come from our co-authoring partners who developed and operate the *SaMMY* platform[4] for berth space bookings, which is the premier platform of this type for Greece and Cyprus. One of the most frequent types of query, handled through the customer support team, relates to getting a good estimate of distances between marinas (in terms of time). Since many customers book berthing space in multiple marinas as part of a sequence of stops (i.e. making an itinerary), they need to be certain about the time it takes to go from one waypoint to the next, so that they can proceed with the reservation requests. Often, such users find conflicting information about distances in various websites, and need the assurance of an authority or knowledgeable expert. In fact, the problem of being able to provide a reliable answer to such queries formed the initial inspiration for the work described in this paper.

[2] Sea-Seek: http://www.sea-seek.com.

[3] Searoutes: http://www.searoutes.com.

[4] SaMMY: http://www.sammyacht.com.

As a result, we note the complete lack of availability of a software tool which would assist the sailing holiday planning process, by generating itinerary recommendations based on realistic routing options, and given some user-specified constraints such as the maximum trip duration, length of stay at each port of call etc. The work presented in this paper builds upon the findings and inspiration from previous literature to address the unique challenges in the context of sailing holiday planning, which, to the best of our knowledge, has not been investigated in the past.

In this paper, we describe the methodology for building such a system, based on openly available spatial datasets. We focus on the Ionian Sea in Greece as a case study, though our methodology can be scaled to cover the entire European continent. Itinerary recommendation is a well-studied area in the context of land-based travel, but it has been entirely overlooked in the important context of sailing holidays. Our work's main contribution to support a fundamental element in the operation of itinerary generation algorithms, which is entirely missing for sea-based travel. This element is to construct realistic and accurate route calculations (distances) for any two ports of call, and we solve this based on historical data on vessel density in areas of interest. We demonstrate how this ability can be integrated into a complete itinerary recommendation system that can run as a service, using a genetic algorithm (GA) approach to creating the itineraries. We describe our GA approach and show how it can be applied to the sailing itinerary problem to support uncertainty and relaxed requirements that are present in the planning phase of a trip. Through this approach, we attempt to optimise the travel experience by optimising routes for selecting marinas offering as great a variety of attractions as possible, while at the same time, respecting user-defined soft criteria such as trip segment duration and overall trip duration.

In the rest of the paper, we being with a review of related literature (Sect. 2) and then describe our methodology for addressing the challenges of berthing option selection, through use of semantic data representations using ontologies and the automatic population and construction of parts of the ontology from openly available POI data (Sect. 3). In Sect. 4, we describe our methodology for solving the sea routing problem using AIS data and a modified version of the A* algorithm, in order to generate realistic routes between berthing locations, to construct an appropriate distance matrix, and to find itinerary suggestions under user-specified trip criteria. The paper concludes with an overview of our contribution, limitations and further work in Sect. 5.

2 Related work

2.1 Planning a sailing holiday

The sailing tourism sector has three core components that constitute the visitation experience, namely (a) the destination; (b) berthing, and; (c) sailing and navigation (Pranita 2020). The traditional process of consumption of a sailing holiday product begins with a preparation phase, in which tourists collect information about the area of the intended visit, identify berthing location options to make stopovers, gather information about the amenities at or nearby berthing locations, plan routes between berthing locations to form an itinerary and discuss the various options available to them (Łapko 2019). Out of these activities, the most challenging parts of planning are *berthing* (finding suitable stopover points to include in the trip) and *sailing & navigation* (planning routes between the selected stopover options).

According to Hoch and Deighton, (1989), the planning, discussion and selection of vacation trips are activities that can be seen as positively contributing to the overall experience of a trip, by building anticipation and engaging in co-creation with other members of their group. In the context of trip planning, we can classify tourists as highly motivated but potentially unfamiliar with the problem domain (area of visit), or already familiar with it. A further factor relates to previous experience (low or high) with the activity (sailing holidays). Thus, in the case of tourists who are both unfamiliar and have little experience, as is increasingly the case with the boat charter industry clientele, the process of learning about destinations and how to travel at this information gathering stage can greatly benefit from management. Though yacht owners can be easily characterised into segments according to their stated preferences and desires when visiting a marina, customer research shows that renters expectations are difficult to estimate, as they, themselves, don't really know what to expect from their visit to a marina Paker and Vural (2016).

To date, there exists a major gap in current tourism literature with respect to sailing holiday planning and satisfaction factors, as acknowledged by Shen et al. (2021). In this respect, even though there exists practically no previous research on how sailing holidays are planned, it is easy to understand why many boat charter websites contain large sections with pre-compiled destination information and suggested itineraries. The presence of organised information is a way to manage the learning process and therefore offers some value to the prospective client. On the other hand, such ready-made solutions detract from the involvement in the planning activity, while effort invested in such activities is known to correlate with positive memories and satisfaction from the overall travel experience (Kim et al. 2012). From this perspective, we can assume that an interactive tool which allows for customised exploration of the various options, particularly the itinerary construction phase, can contribute positively to the tourists' experience, by facilitating the planning process for inexperienced persons, and allowing for creative synthesis between alternatives. Next, we provide a brief overview of related work to the problems of tourist itinerary recommendation, further delving into the subproblems of selecting berthing options, and determining sea routes, as pertinent to our problem domain.

2.2 Tourist itinerary recommendation

To a large extent, any tourist wishing to explore a destination for a period of time has to consider the formulation of an itinerary (i.e. which places to visit, and in what order). Solving the planning problem is confounded when travellers are inexperienced, in an unfamiliar area, or in an area that is dense with possible alternatives. Therefore, algorithmic approaches for the generation of itinerary recommendations have been explored in past literature, in an attempt to address the *Tourist*

Trip Design Problem, TTDP (Vansteenwegen and Van Oudheusden 2007). In TTDP, travel is modelled using a graph, where nodes represent POIs in a destination, and are connected by weighted edges representing the ability to travel between these. An extension to this problem is the Vacation Planning Problem (*VPP*), defined by (Gavalas et al. 2019), in which a tourist specifies a wider area to visit, which can include multiple destinations, and each destination is associated with a range of POIs. The TTDP and related extensions are derived from the *Orienteering Problem, OP*, inheriting the constraints that users have a pre-defined time budget which must cover travel to POIs and stay time at each POI, and that each node can be visited only once. The presence of a time budget means that a user cannot visit all POIs in a trip, and the objective is to find itineraries which maximise *profit*. This profit is a value attached to each node, i.e. a metric which is used as a proxy to measure the potential user benefit from having visited a POI. The quality of a recommendation is therefore assessed by the cumulative profit to be gained from it, when compared to other options. Extensive reviews of algorithmic approaches to the OP and TTDP are found in Gavalas et al. (2014), Gunawan et al. (2016), Lim et al. (2019), Tenemaza et al. (2020b).

In TTDP, the major constraint is the available time budget since any path must lead from the start to the end node within this budget. Therefore, the temporal length of any candidate path must fit into the relevant budget. To solve the problem, one must know the travel time between any pair of nodes and also the stay time at each node. Travel time can be computed using an assumed average speed based on transport mode, and known distance between nodes (Chen et al. 2015). Both these metrics can be derived from direction service APIs (e.g. Google Maps) or even modelled in a probabilistic manner (Zhang et al. 2015). More realistic calculations can involve historical traffic speed data or other estimates based on big data analytics (Cristian et al. 2021). The stay time at each node can be modelled statically or randomly, or it can be based on actual data, e.g. scraped from social networks on a POI-by-POI basis (Friggstad et al. 2018), or estimated based on the average stay time for a range of POIs belonging to the same category (Fogli and Sansonetti 2019). Further, suitable metrics for the profit value of each node must be determined. These metrics can be based on a range of objective and subjective observations. Such objective metrics can be the popularity of a POI in a social network, for example, measured by the number of likes, check-ins, photos taken at a location (Yochum et al. 2020; Fogli and Sansonetti 2019; Friggstad et al. 2018). Subjective metrics typically involve the construction of a user profile and their preferences on various POI categories, or metrics of the user's own past behaviour, such as check-ins to venues of similar categories in other locations (Fogli and Sansonetti 2019; Lim et al. 2018).

Ultimately the quality of the produced itinerary must be somehow evaluated, and this can be achieved in a number of ways, as described in Lim et al. (2019) and Fogli and Sansonetti (2019). In some literature examples, algorithm outputs are evaluated with real visitors, for example as in Tenemaza et al. (2020b). In the case of real-life travel itinerary data being available, traditional metrics such as precision, recall and F1-score can be used to measure the quality of a proposed itinerary. Heuristic-based evaluation can be used where actual tourist data is not available, using metrics such as the number of total POIs recommended (sometimes also viewed as a difference to the maximum number of POIs specified by a user), sum of POI popularity, alignment of recommended POI categories to user-stated interests, length of itinerary and diversity of POI categories in an itinerary.

In the context of sailing trip recommendations, there are two subproblems which merit further discussion. First, selecting appropriate berthing locations to use as graph nodes, and secondly to determine the distances (and hence, travel time) between them. We discuss previous work in these two areas next.

2.3 Modelling berthing options

In any POI-based tourism system, data on these POIs must be modelled in order to allow its spatial and semantic querying. For sailing, the main POI type of interest is berthing locatios (i.e harbours, marinas or other anchoring points), since they will be the main stopover points during the trip. A system can select berthing options from a large set in a given destination area, by applying filters on information about their location and on-site or nearby amenities. These filters can be used to reduce the set of candidate nodes in a classic TTDP application, and also to determine the individual profit for each node.

Yachters are not only interested in on-site amenities (e.g. availability of fuel, water, electricity, waste disposal, repair services etc.), but also nearby service offerings (Benevolo and Spinelli 2021; Mikulić et al. 2015; Paker and Vural 2016). Such offerings can relate to the proximity and type of catering options, entertainment, cultural and athletic activities, shopping opportunities and other factors that are related to the tourism industry. Therefore, a method to represent this information in a queryable form must be developed. One approach is to model locations as points of interest (POIs) in a relational database with spatial extensions. as in He et al. (2016) and Heikkinen et al. (2014). This approach requires the specification of a fixed schema, and is able to determine spatial relationships between POIs, such as answering queries about POI presence in a radius, or a bounding box. Another approach is to model POIs as semantic data, through ontologies. These representations offer greater flexibility in terms of representing levels of abstraction, can be used to integrate POI datasets in a vendor-agnostic manner, can be used for logical inferencing on new data, and to provide recommendations to users of POI-based services (Cabrera Rivera et al. 2015; Palumbo et al. 2019; Patroumpas et al. 2019). A further advantage of ontology-based representations is that they can be used to find POIs that match users' profiles and interests. An overview of related work using ontologies in the TTDP is given in Souffriau and Vansteenwegen (2010).

2.4 Determining sea routes

Contrary to terrestrial transportation, where the infrastructure (roads, railways) is fixed, sea routing represents an entirely different problem, since there are no "fixed" routes between any two ports. A sailing vessel is theoretically free to take any course between two given waypoints, within some physical constraints (e.g. it cannot travel overland). The challenge of determining sea routes is therefore more complex. One approach used in Kuhlemann and Tierney (2020) is to first draw a straight line between waypoints, and then iteratively “push” the midline of segments orthogonally towards the water (if they are overland) until a route that avoids all land has been generated. Another approach is to use Dijkstra’s or the A* shortest-path algorithm to find a route, by representing the world as a grid on which each tile is either navigable or not (i.e. terrain) (Kurosawa et al. 2020; Shin et al. 2020; Wang et al. 2018, 2019). Such algorithms can be modified in order to add further cost features in the distance function, incorporating information such as weather patterns and sea currents from external data sources. These modifications have been proven successful in automatically generating routes which are optimal according to some criterion (e.g. saving fuel, minimising travel time etc.). A third possibility is to use genetic algorithms in order to determine an optimal route under specific criteria (Kuhlemann and Tierney 2020; Wang et al. 2018, 2019). All these previous approaches have been applied with the underling assumption that the travelling vessel is able to cross any expanse of water (typically, a freighter or tanker). As such, the world is modelled as a large grid or graph, each grid square or graph node representing an area of several km^2. While this approach can work reasonably well at planetary scale, it is less appropriate for holiday sailing vessels, where better resolution is needed in order to better capture the coast-hugging and reef-avoiding behaviours of boats. To determine more realistic representations of viable sea routing graphs, researchers have very recently turned to the availability of Automatic Identification System data, which contains continuous information on vessel position and type (Filipiak et al. 2020; Sheng and Yin 2018). This data allow the construction of navigable graphs, which closely match actual practice as demonstrated by real vessels. The use of AIS data for sea route planning has been commercialised by SeaRoutes,[5] though the provided service is tailored towards the merchant navy (freight ship routing) and is entirely unsuitable for the sailing holiday context.

2.5 Summary

Our presented work is intended to support the broad exploration of alternatives, possibilities and options under various assumptions and preferences (e.g. available time, activity and amenity preferences, sailing time) in the stage where a sailing holiday is being *contemplated* and explored at an initial phase, especially by non-expert users. These planning phases could be weeks or months before the actual voyage. Therefore, we aim to support the users in answering broad-scope questions such as ”How much of this area could I explore in my available time?” rather than specific questions such as ”What should be my precise departure times in order to complete a chosen itinerary?”.

Before any of the algorithmic approaches, such as those found in the TTDP/OP literature, can be applied in the context of sailing holiday planning, there is a significant sub-problem that needs to be solved, namely the realistic calculation of travel

[5] Searoutes: http://www.searoutes.com.

time between stopover berthing locations. In this paper, our main contribution is to present a solution for the fundamental problem of determining graph node distances and hence travel time between nodes, in the context of sea-based travel. This problem does not exist in land-based travel and has not been addressed in TTDP-related literature. Previous literature addressing the sea route problem has only considered merchant navy vessels covering large distances and able to travel over any expanse of water. Thus, to our knowledge, the presented work work is novel since the generation of fine-grained realistic routes between ports of call in the context of sailing holidays has not been previously explored.

To demonstrate how our approach can fit into an integrated itinerary recommender system, we describe next the implementation of a basic, but complete itinerary suggestion service. This is based on the modelling of stopover berthing locations and nearby POIs using ontology-based knowledge representation, the leveraging of vessel-density data to create plausible routes between any two ports in an area of interest, and finally an itinerary generation system, based on a genetic algorithm that optimises for maximum POI diversity in visited marinas, while controlling for appropriate trip segment lengths and overall trip duration.

3 Modelling berthing locations

To address the problem of selecting from the set of marinas in a given area, we employ semantic modelling of the relevant location-based information. The rationale behind the use of semantic technologies rather than a relational database, is the flexibility of the former to afford logical relationships across the various data entities, therefore aligning the querying process in a way that resembles the rational process of the intended users. For this purpose, we developed an ontology using Protégé,[6] with four main entity types (classes). The ontology contains the concepts of `City`, `Marina`, Point of Interest (`POI`) and `Service`. We note here that we use the term `Marina` to cover the general notion of a berthing location. The classes are joined through three main relationships (object properties). The classes `POI` and `Marina` are related to the `City` class through the property *isLocatedIn* (e.g., a `POI` or a `Marina` instance is located in an instance of `City`). A `POI` is also related to `Marina` through the *isNearMarina* property (a `POI` instance is near a `Marina` instance). Finally, a `POI` is related to the `Service` class through the *offersPOIService* property (a `POI` instance offers a `Service` instance). The `Service` class is a superclass from which a range of subclasses inherit, in order to model the specific service types offered by various `POIs`. The developed ontology is shown in Fig. 1, depicting the four main classes, and examples of subclasses of the Service class. Each class contains a range of data properties, which are used to provide further information about the instances of each class. These data properties can be used in order to refine queries, so as to filter results according to specific criteria. The data properties are shown in Table 1.

[6] Protégé Ontology Editor: https://protege.stanford.edu/.

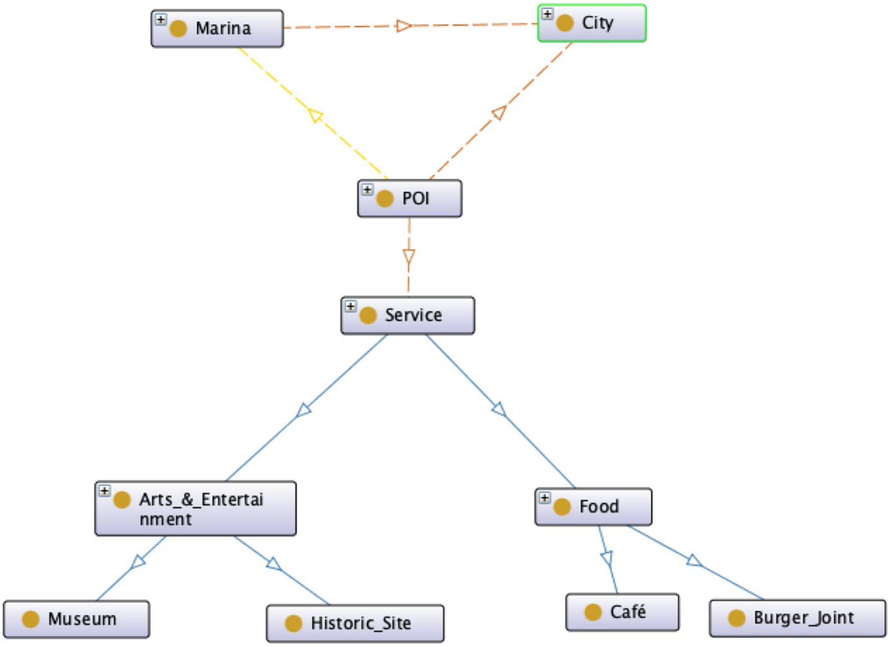


Fig. 1 Ontology overview

Table 1 Data properties and their relation to each class type

Data property	Data type	Applies to class
Latitude	Decimal	`POI, Marina, City`
Longitude	Decimal	`POI, Marina, City`
State	String	`City`
PlaceName	String	`POI, Marina, City`
Price range	Integer	`POI, Service, Marina`
Marina electricity	Boolean	`Marina`
Marina water	Boolean	`Marina`
Marina WiFi	Boolean	`Marina`
Marina firstaid	Boolean	`Marina`
Marina wastedisposal	Boolean	`Marina`
Marina parking	Boolean	`Marina`
Marina toilets	Boolean	`Marina`

This ontology allows for the formulation of generic search queries on the stored data, supporting a range of broad or narrower scope queries that fit the planning goals of a user. For example, using the ontology, it is easy to formulate very broad queries, such as to *"Retrieve all **Marinas** located in a list of given islands"*, or to refine it further by asking to *"Retrieve all **Marinas** located in a list of given islands that have nearby **POIs** offering the **Service** type **Food**"* (using modelled relationships), and even further by specifying data property criteria, such as geographical coordinates: *"Retrieve all* ***Marinas*** *located in a list of given islands that have nearby* ***POIs*** *offering the* ***Service*** *type* ***Food*** *with a latitude/longitude between* [$X1$, $Y1$] *and* [$X2$, $Y2$]".

3.1 Populating the ontology

To develop the ontology, we began by manually defining the four main classes (`City`, `Marina`, `POI` and `Service`) along with their data properties in Protégé's ontology editor. We did not define the `Service` subclasses, since this was achieved automatically in a later step, as will be explained in Sect. 3.1.3. The process of enriching the ontology with class instances (data) and `Service` class subclasses, involved the following steps: (1) Populate the ontology with `Marina` instances, (2) Populate with `City` instances and link to related `Marina` instances, (3) Generate the `Service` subclass taxonomy and (4) Populate with `POI` instances, linking to related `City`, `Marina` and `Service` subclass.

In order to populate the ontology with instances of `Marina`, we obtained a list of all 153 licensed tourist ports in Greece, through the Greek Ministry of Tourism website.[7] For the rest of the data, we considered various openly queryable datasets, including Google Maps, Facebook and Foursquare. We settled to mostly use the Foursquare Places API,[8] given its comprehensive category taxonomy, which is more fine-grained than that of other data sources. We also employed the Google Maps API for the purposes of geocoding coordinates, as will be explained next. We obtained data employing the Python *foursquare*[9] client (v1!2020.1.30) and *Google Maps*[10] client (v4.4.5). The population of the ontology with class instances, and with new classes and relationships as required, was done using the *OWLReady2* Python library[11] (v0.31). Throughout the project, the Python language version used was v3.8.5.

3.1.1 Obtaining marina data

According to Greek legislation, there exist three categories of tourist port: Fully licensed marinas, harbours and anchorages, and hotel ports (this type was abolished in 2012). Each port type is offered as a set of pins on an embedded interactive Google Maps frame (Fig. 2). We extracted the data in KML format, which includes the place name and geographical coordinates. The main difference between fully licensed marinas and harbour and anchorage types, in terms of touristic interest, is the requirement for the existence of first-aid facilities and parking space in

[7] Licensed Tourism Ports in Greece: https://mintour.gov.gr/ependyseis/toyristikoi-limenes/chartes-toyristikon-limenon/.

[8] Foursquare Places API: https://developer.foursquare.com/docs/places-api/.

[9] foursquare Python library: https://pypi.org/project/foursquare/.

[10] Google Maps Python library: https://github.com/googlemaps/google-maps-services-python.

[11] OWLReady2: https://pypi.org/project/Owlready2/.

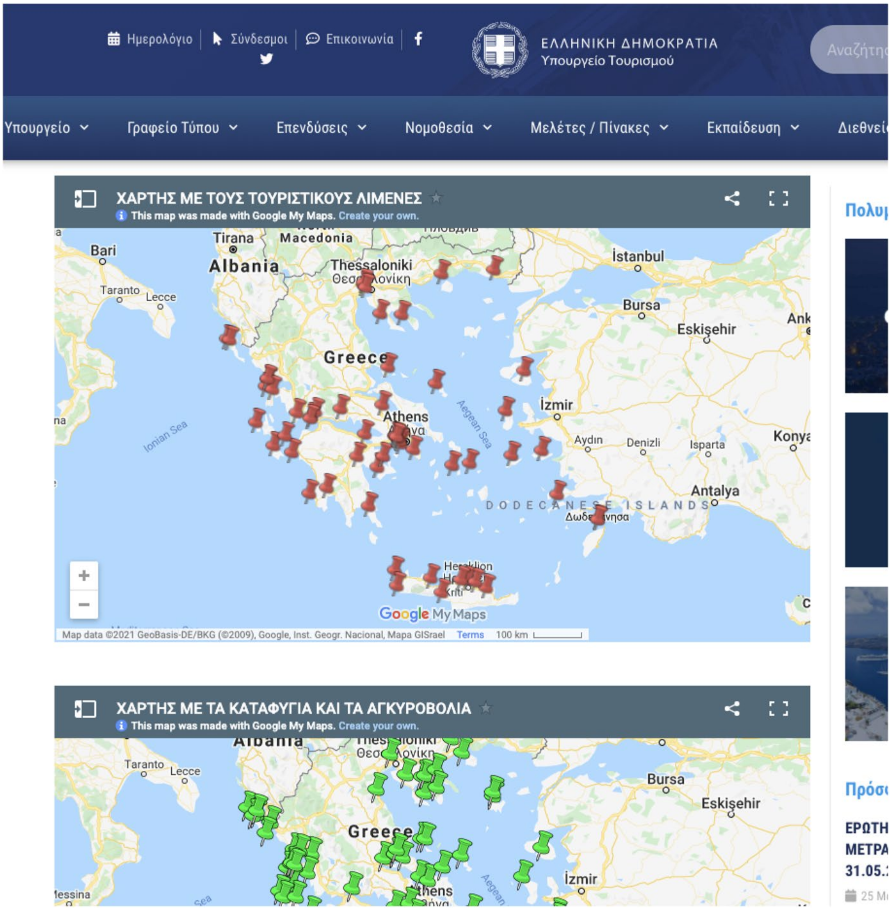
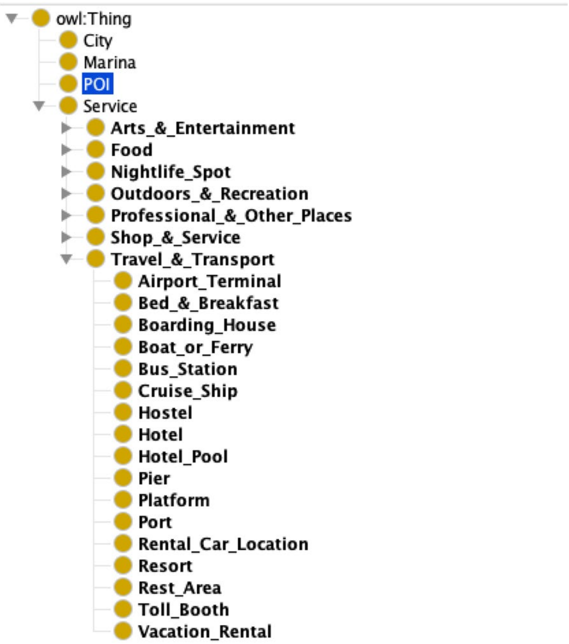


Fig. 2 Data offered in the Greek ministry of tourism website. Screen captured 25 May 2021

marinas. Both types have a requirement for water, electricity and communication facilities, waste disposal and toilets. As a result, we determined the relevant property values in Table 1 to True or False according to legislation. For hotel ports, we determined that all data properties would be set to False, since it is unknown what regulations currently apply to these types, after their abolishment. The related KMLs were converted into CSV format and a Python script was written to generate the relevant Marina class instances, data property values and required relations inside the ontology.

3.1.2 Obtaining city data

For this step, we used the existing Marina instance CSV and searched for venues located around the coordinates of each marina, using the */explore* endpoint of the Foursquare API. Each retrieved venue includes a list of location data values, which



- owl:Thing
  - City
  - Marina
  - POI
  - Service
    - Arts_&_Entertainment
    - Food
    - Nightlife_Spot
    - Outdoors_&_Recreation
    - Professional_&_Other_Places
    - Shop_&_Service
    - Travel_&_Transport
      - Airport_Terminal
      - Bed_&_Breakfast
      - Boarding_House
      - Boat_or_Ferry
      - Bus_Station
      - Cruise_Ship
      - Hostel
      - Hotel
      - Hotel_Pool
      - Pier
      - Platform
      - Port
      - Rental_Car_Location
      - Resort
      - Rest_Area
      - Toll_Booth
      - Vacation_Rental

Fig. 3 Example of the 3-level Service taxonomy in our ontology

may contain values for ”city” and ”state”. Since not every venue contains these details, we iterated through the list of results (10 by default) for venues that contained this information. If found, we saved this information, otherwise we repeated the query to fetch the next set of results, until the information was found. This process resulted in obtaining a list of cities and states pertinent to each marina. Using the Google Maps API,[12] we geocoded the names of each city to obtain the city coordinates. Using this data, we populated the ontology with City instances.

3.1.3 Obtaining POIs and service taxonomy data

The next step was to determine an appropriate taxonomy for the Service class. For this, we employed the Foursquare */categories* API, which returns a JSON array of all categories and subcategories used by Foursquare (1145 in total, with 10 top-level categories, and up to 5 nesting levels). Obviously such a large number of categories is not necessary for our system, since, for example, many of these refer to location types that are not pertinent for our study area (e.g. “Apres-Ski bar”). Therefore we took a different approach, by first retrieving a list of all recommended POIs within 2km of each marina, using the */explore* endpoint, which returns, for each venue, its specific sub-category type. Then, we recursively searched through the entire category taxonomy, to find the top-level category in which the venue belongs. As a result, we obtain a list of 3614 POIs, their sub-category and the related top-level

[12] Google Maps Geocoding: https://developers.google.com/maps/documentation/geocoding/overview.

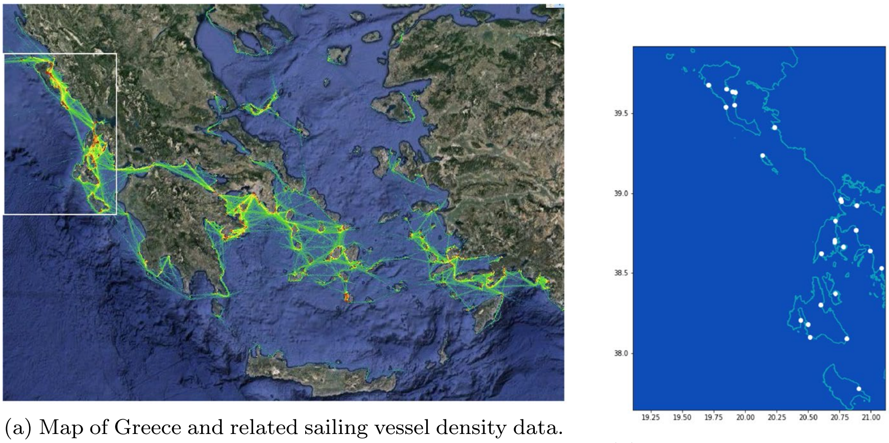
category. We noted that several POIs were present more than once in this dataset, owing to marinas being close to each other, and thus the same POI was present in multiple recommended venue results. Removing 239 redundant entries, we are left with a total of 3375 unique POIs. Then, in our ontology, we inserted all discovered top-level categories under the Service class, and a further hierarchical level of all sub-categories under each top-level category, therefore flattening the Foursquare taxonomy to 3 levels (Service, Top-Level, Sub-Category), as shown in Fig. 3. As a result we obtain 7 top-level categories (number of subcategories in parentheses) Arts and Entertainment (29), Food (65), Nightlife Spot (15), Outdoors and Recreation (45), Professional and Other places (2), Shop and Service (38) and Travel and Transport (17).

The related data for POIs was already obtained in the previous step, therefore for the last part of our process, all we needed to do was to insert the data into the ontology as POI class instances, paying attention to the generation of all necessary linkages to City, Marina and Service sub-classes as required. Each POI instance was also populated with the related values for its data properties, as retrieved from the Foursquare API (placeName, latitude, longitude, priceRange).

4 Generating trip recommendations

We now turn to the problem of trip planning in the context of a sailing holiday. More specifically, the problem we focus on is the recommendation of a sequence of berthing locations (itinerary), given a start and end destination, as well as a set of intermediate candidate berthing locations that meet the user's criteria (obtained by querying the ontology). The recommended itineraries must meet a set of user-specified constraints (e.g. maximum trip duration) and assumptions (e.g. average sailing speed, stay duration at each port). This problem represents a typical scenario in yachting tourism, where significant investment of time and effort in trip planning is required at the start of a journey to a new area.

Trip planning problems have traditionally been approached through a graph modelling perspective, where stops (waypoints) can be modelled as vertices in a graph, connected via edges. Contrary to the typical problem of trip planning in road networks, where each node is connected directly to a limited number of other nodes, in theory, each sea port (node) can be linked through a direct sea route to all other ports. This means that any graph representing the trip planning problem would be a fully-connected graph, with the number of possible routes from any node to any other exploding in size with the number of nodes. Of course, this is not really the case, since there exist several real-world constraints that place limits on the number of ports that can be considered as connected to any given port (e.g. the fuel capacity of a vessel, or the willingness of the captain and crew to spend more than a given amount of time at sea without mooring). Another difference is that while road, rail and even air routes are mostly well-defined and can be considered fixed, sea routes from one port to another can take any shape or form, therefore making the calculation of edge weights (distances between nodes) difficult to realistically estimate. In the next sections, we describe how we dealt with these issues.



(a) Map of Greece and related sailing vessel density data. The selected region of Ionian Sea is shown in the box outline

(b) Marina locations in the Ionian Sea

Fig. 4 Detailed overview of the paper's case study area

In Sect. 3.1 we described our methodology for data capture covering the entire country of Greece. For the rest of this paper, we focus our analysis on a specific sailing region in Greece, namely the Ionian sea, in order to keep visualisations at a scale which can be clearly presented in the paper. This region is one of the most popular sailing destinations in the country (Fig. 4a). We select a bounding box of latitude between 37.647399 and 39.915594 and longitude between 19.107729 and 21.112372, representing an area of 43,768.14 km^2. In total, there are 27 marinas in the region (Fig. 4b).

4.1 Determining plausible sea routes through AIS data

One approach to generate sea routes is to model distances between ports using a simple algorithm that considers (and avoids) the shape of coastlines, sailing along them as required, while taking a direct line approach between sections of open-sea areas, as described in (Kuhlemann and Tierney 2020). This approach works well for large commercial vessels which seek to minimise travel distances (and costs), but it is likely unsuitable for itinerary planning in the context of a sailing holiday. Skippers familiar with a region, know to avoid certain parts of the sea, due to known prevalent weather and other conditions, while preferring longer routes and indirect approaches in order to include scenic trip parts and other preferences. This "knowledge" could be captured if it were somehow possible to obtain coordinate logs from many vessels over a long period of time, which would capture the popular sea routes in a given area. In the last two decades, vessels have increasingly employed the Automatic Identification System (AIS), which continuously reports location data, as part of international conventions to improve maritime safety. In Greece, the use of AIS systems is mandatory in recreational vessels that are chartered out for tourism, as a means of enforcing both safety and appropriate taxation.

Real-time and historical AIS data is not openly available, but can be acquired through paid subscriptions from private companies. On the other hand, the European Maritime Observation and Data Network (EMODnet)[13] has recently (2019) offered access to an open data product based on AIS, which calculates vessel density maps in the form of a 1 × 1 km square grid covering the entire EU. The data can be exported in geoTIFF format (a TIFF image which contains georeferenced information, including projection, coordinate systems, ellipsoids, datums and all other information to determine spatial reference for the image data). In this case, downloaded geoTIFFs effectively represent vessel densities in each pixel of the image, with values ranging from 0-255. Therefore, the downloaded geoTIFF can be simply considered as a 2D array of values, representing vessel density in a geo-referenced grid, where it is possible to easily convert between pixel coordinates and spatial coordinates. In these geoTIFFs, pixels with a value of zero represent either inland regions, or sea regions where sailing vessels do not ordinarily pass through. Simple thresholding manipulations on the 2D array can be applied to turn pixels with a low vessel density into such areas as well. In effect, we can treat this 2D array of values as a 2D map that depicts pixels with obstacles (zeros) and without obstacles (non-zero values), and then employ well-known path-finding algorithms (e.g. Dijkstra's or A*) to find routes between any two sea ports.

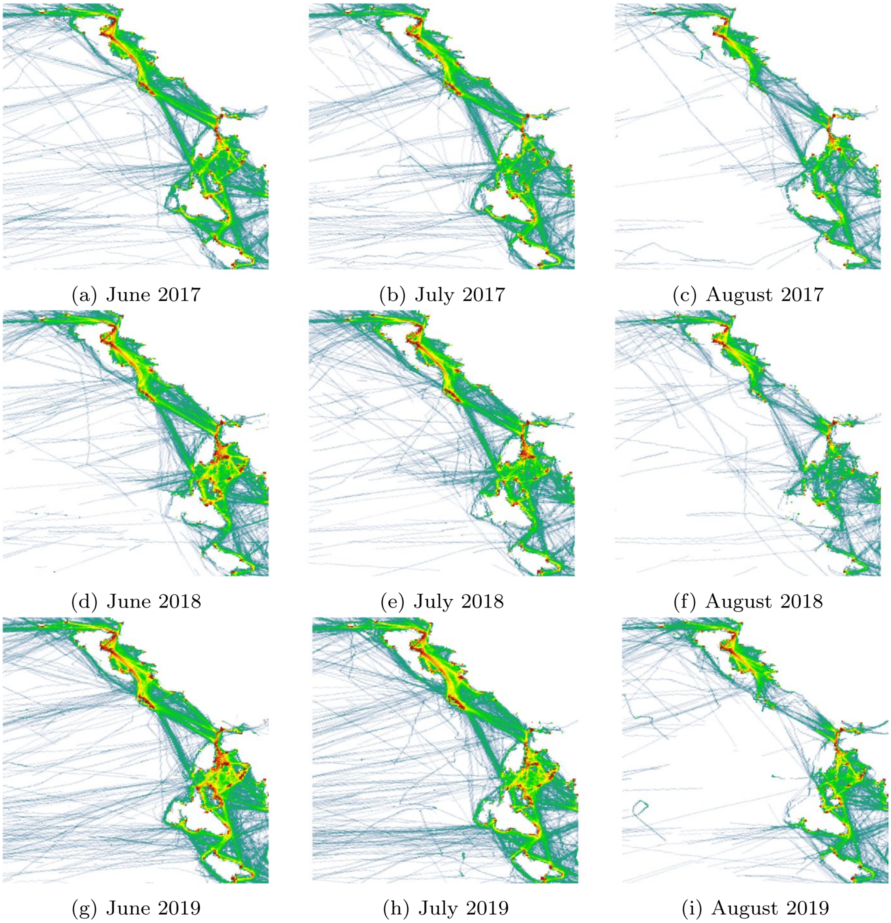
Therefore, to derive plausible sea routes between ports, we employ the following strategy. First, we download and process vessel density data from EMODnet. Then, we calculate plausible routes between all ports using the A* algorithm. Finally, we derive a distance matrix table from the calculated routes. In this section, we provide some more details about how these steps were implemented, and in the next section, we describe how used the derived distance matrix to assign weights to the edges of a graph, connecting the various ports, to solve the trip planning problem.

4.1.1 Obtaining EMODNet data

The data provided by EMODnet can be downloaded by specifying a coordinate bounding box (area of interest), the vessel category, and period of interest (aggregates by month and year, or aggregates just by year). Since the AIS data specification provides a field for vessel type, it is possible to filter data by selecting logs generated by the "Sailing" vessel category only, which is ideal for our purpose, since we are not interested in data generated by other vessel types (e.g. fishing boats, cargo ships, passenger or cruise ships). Therefore we downloaded the relevant sailing vessel geoTIFF data for the months of June, July and August in the years 2017, 2018 and 2019, a total of 9 geoTIFFs (Fig. 5). Data for 2020 is also available, but we did not use these since that year's tourism patterns cannot be considered normal, due to the COVID'19 pandemic circumstances. Manipulation of the geoTIFF files was done with the *Rasterio* library (v1.2.3).

The 2D value arrays extracted from these geoTIFFs were averaged into a single array. Further, we applied a thresholding value of 10, in order to reduce the

[13] EMODnet: https://www.emodnet-humanactivities.eu/about.php.



(a) June 2017 (b) July 2017 (c) August 2017

(d) June 2018 (e) July 2018 (f) August 2018

(g) June 2019 (h) July 2019 (i) August 2019

Fig. 5 Vessel densities in June, July, August in 2017, 2018, 2019

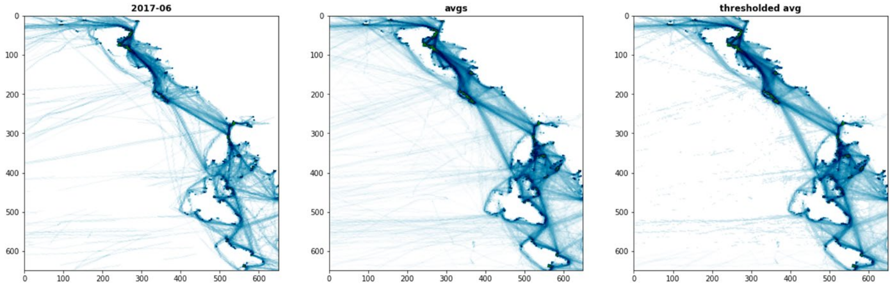


Fig. 6 Example of a single month's vessel density data (June 2017, left), the averaged data (center) and the resulting vessel density map after thresholding ($t = 10$, right)

complexity of the 2D map, while maintaining the important sea routes in the averaged data. As can be seen in Fig. 6, the coastal outlines are preserved, while most of the irrelevant sea routes are eliminated from the map.

4.1.2 Route generation between ports

Next, we feed the list of marinas and the resultant 2D map into the A* algorithm, in order to find plausible routes between them. Notably here, we noticed that 1 marina was located in areas where there is no vessel density data—upon further inspection, one marina (Pentanti, Corfu) is a location which obtained a license in 2008 but has not yet built, thus it was removed from the set. Further, two marinas share the same pixel coordinates (they are effectively the same marina spread over two adjacent locations in the same city), and therefore we merged them into one entity.

The A* algorithm is parametrised so as to allow movement from any pixel to any other in 8 directions (horizontal, vertical and diagonal). The routes are returned as a sequence of pixel coordinates. To find the total distance of a route, we simply sum the Haversine distance between the geographical coordinates at the centre of subsequent pixels in a route. The conversion process between pixels and geographical coordinates is easy to achieve (a benefit of the georeferenced data in geoTIFFs), using the *Rasterio* library.

The use of the diagonal distance guarantees the discovery of the shortest path between any two marinas, however, to determine the most plausible route, we examined the notion of taking into account the vessel density in adjacent pixel options as an additional heuristic. As such, we effected a slight modification to the A* algorithm, in the step of calculating the heuristic scores while determining the best neighbouring node. Here, the original algorithm takes into account the cost of moving between adjacent cells, which has a maximum of $\sqrt{2}$ since each movement can be made by 1 pixel in 8 directions. We extend this heuristic to also take into account the adjacent node vessel density data, using a bias towards each of the two metrics, distance and density. To calculate the heuristic value $h_{(n)}$ we use:

$$h_{(n)} = b \times \frac{V_{(i,n_i)}}{V_{max}} + (1 - b) \times \frac{d_{(i,n_i)}}{d_{max}}$$

where $b \in [0, 1]$ is the bias weight, $V_{(i,n_i)}$ is the vessel density difference between a node i and its neighbour n_i (we normalise this by dividing by $V_{max} = 255$, which is the maximum possible difference), and $d_{(i,n_i)}$ is the movement cost (distance) between the a node and its neighbour, which we also normalise by dividing by $d_{max} = \sqrt{2}$. Therefore, we combined the distance heuristic with vessel density, each criterion being weighted to the sum of 1. An example of the difference in the algorithm's behaviour with different bias weights is shown in Fig. 7.

4.1.3 Constructing the distance matrix

Distance matrices can be used as a way to pre-compute and quickly retrieve distances between any two nodes in a graph, and hence are often crucial for performance in

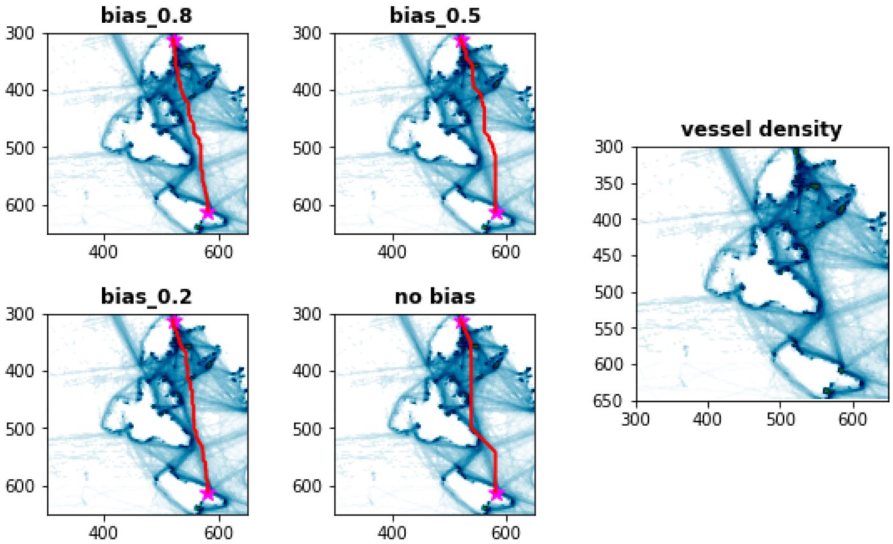


Fig. 7 Example of a route between Lefkada and Kerkyra (Corfu) calculated with A* on the thresholded data, using various vessel density selection bias levels

Table 2 Route statistics under various algorithm settings for trips between top-10 marinas

	b = 0.8	b = 0.5	b = 0.2	simple A*
Mean route length (km)	97.469	97.469	97.467	97.466
σ	61.956	61.955	61.955	61.964
Mean route pop. score	17853.277	18132.882	18059.902	16319.858
σ	9673.762	9827.957	9697.695	8365.670

the operation of route-finding systems. To generate the distance matrix to be used in our system, we need to determine the most appropriate value of bias with appropriate metrics. One metric is the total length of the proposed route (as described above), and a further metric is the total vessel density in the pixels which make up the route. The latter metric is indicative of the ability of the algorithm to select route options closest to the actual route behaviour demonstrated in the AIS data.

To measure the algorithm's overall performance, we selected four bias values $b \in [0.0, 0.2, 0.5, 0.8]$, with $b = 0.0$ corresponding to plain A*, and ran route calculations under these settings for the routes between the top 10 marinas in terms of vessel density in our dataset (45 routes). We excluded one route because it referred to a special case of two adjacent marinas which were located on the same map pixel (essentially the city of Aktio, which has two marinas located right next to each other). As can be seen in Table 2, all algorithms produced similar average route distances. On the other hand, we note that the average vessel density in the routes selected by the various options differ. A Wilcoxon signed rank test (following

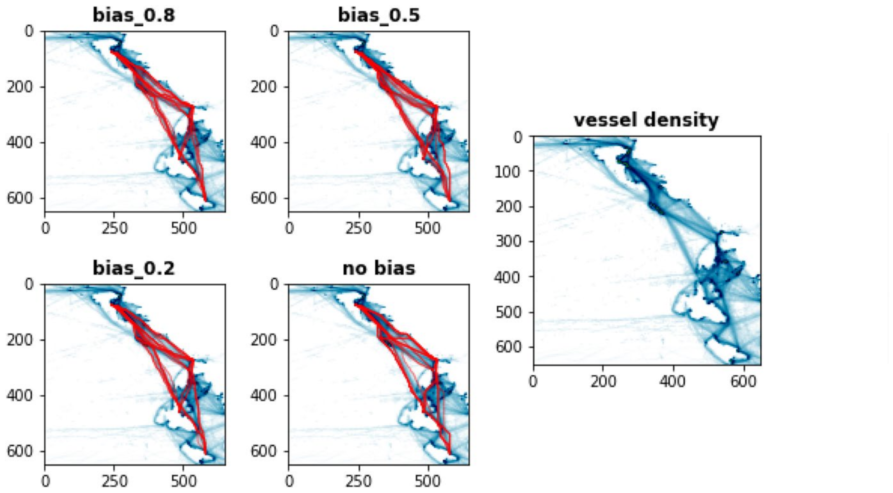


Fig. 8 Coverage of all routes between top-10 marinas

Shapiro-Wilk normality test) with post-hoc Bonferroni correction (setting the p-value threshold to 0.0083), demonstrates that there exists a statistically significant difference between the plain A* implementation and b = 0.8 (p = 0.001, Z = -3.350); plain A* and b = 0.5 (p = 0.004, Z = −2.896). The difference between plain A* and b = 0.2 is marginally non-statistically significant (p = 0.009, Z = −2.620). All other differences were not statistically significant. However, due to the slightly better nominal score with b = 0.5, we select this option for further steps of calculating the final distance matrix, containing all 351 possible routes between any two marinas. The overall coverage of the routes selected by the algorithm options in this comparison is shown in Fig. 8.

4.2 Trip recommendation

Having calculated and saved the distance matrix between all marinas as described in the previous section, we can now treat the route recommendation problem in the classical manner of finding paths along a weighted graph, where the graph nodes are the marinas, connected by weighed edges, where the edge weight is the distance between any two marinas, as calculated in Sect. 4.1. The problem at hand is roughly the same as the TTDP: A route must be recommended to a user, from a large range of alternatives, given a set of constraints of which the trip time budget is perhaps the most important. In our case, since the system is aimed at the initial exploratory phases of a trip, where the user is exploring options without very fixed requirements, the recommendation system should explore a range of options which may be slightly outside the constraints input by the user. For example, a user might specify a trip duration of 7 days, but this could be treated as a *suggestion*, rather than a hard limit, since some very attractive solution could be present if the trip were to be extended by 1 day. To address the problem, we employ a genetic algorithm (GA)

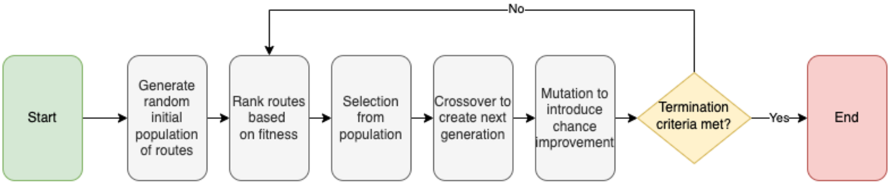

Start
Generate random initial population of routes
Rank routes based on fitness
Selection from population
Crossover to create next generation
Mutation to introduce chance improvement
Termination criteria met?
No
Yes
End

Fig. 9 Outline of the GA steps

Table 3 Trip selection parameters

Parameter	Description	Unit	Sample value
Start_p	Starting point of the trip	Nominal	”Lefkada”
End_p	Ending point of the trip	Nominal	”Zakynthos”
Max_sailtime	Preferred maximum duration of each sailing route segment	Hours	8
Pref_sailtime	Preferred minimum duration of each sailing route segment	Hours	4
Sail_speed	Average sailing speed	Km/h	8
Trip_days	Total duration of the trip	Days	14

approach, as in Tenemaza et al. (2020a). Genetic algorithms have been employed to solve many variations of the travelling salesman problem, including TTDP (e.g. Chen et al. (2013), Zheng et al. (2017)) and have demonstrated the achievement of a good balance between quality of results and speed of execution. Here, we employ a GA to solve the problem of a round-trip design (i.e. a user sailing off from a given marina, visiting several distinct locations and returning to the same marina as they started). The algorithm's main stages are outlined in Fig. 9.

4.2.1 Parameters for executing the GA

There are two sets of parameters that need to be specified for the GA to run. First, parameters that are input by the user are required in order to set up the trip constraints. These exogenous parameters are shown in Table 3. A starting point is required, but not an ending point. If the latter is not provided, a round-trip is assumed. For the rest of the paper, though our algorithm supports both options, we proceed with the design of round-trips only, since this is the most typical use case for a tourist. A user can also input the maximum and minimum sailing duration for each segment of the trip. The latter represents the tourist's estimate of the amount of sailing they'd like to do ideally per day, while the former is the absolute maximum sailing they would be willing to do for any single segment.

Next, a set of parameters for the algorithm's execution must be set by the programmer, as shown in Table 4. These are mostly self-explanatory, though the

Table 4 Genetic algorithm parameters

Parameter	Description	Unit	Sample value
pop_size	Size of the initial population of routes	Integer	200
elite_size	Number of ”elite” routes to keep for breeding the next generation	Integer	3
mutation_rate	Probability of route gene mutation	Floating point	0.01
n_solutions	Maximum number of candidate routes to return	Integer	5

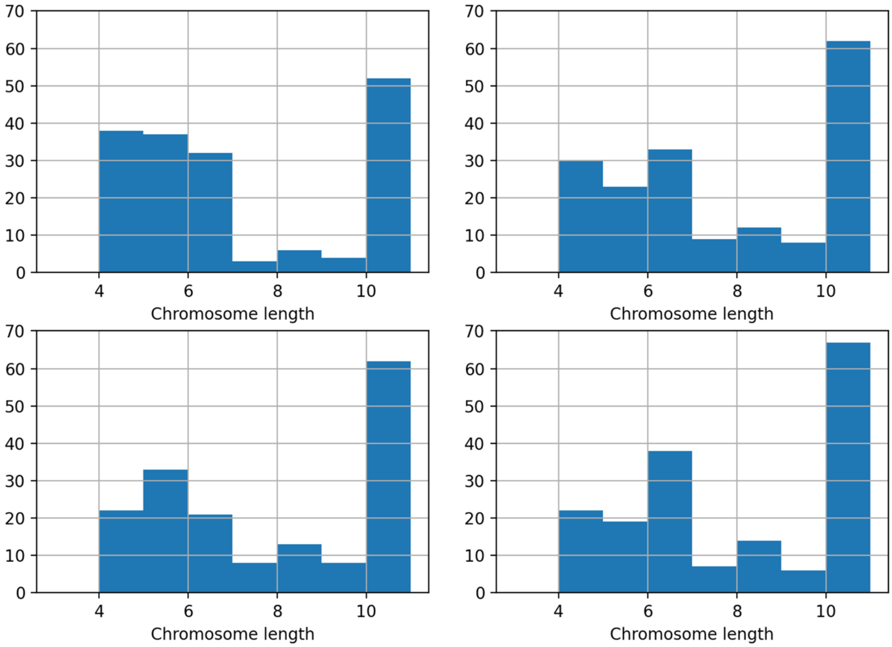


Fig. 10 Distribution of route chromosome lengths for 4 random initial populations

n_solutions parameter is worth noting as the stopping condition for the algorithm (it stops after only N solutions are left in the population).

4.2.2 Initial population

Each candidate route is represented by a ”chromosome”, which is simply a list of the marinas to be visited, in sequence. For the generation of chromosomes, we assume that each marina is an overnight stopover location. Therefore we create chromosomes that have a minimum and maximum length of $r_{min}, r_{max} = trip\_days \pm \frac{2 \times trip\_days}{3}$. Chromosome lengths are biased for the initial population to create a short route of random length between $r_{min} + 0.2 \times (r_{max} - r_{min})$ with probability $p_s = 0.4$, a long route of random length between $r_{max} - 0.2 \times (r_{max} - r_{min})$ with probability $p_l = 0.4$ and anything in between with a probability $p_n = 0.2$. This setup produces a range of chromosomes with various route lengths (Fig. 10), and the bias between long and short routes, as will be described next, helps the convergence towards the user's specified trip duration through crossover.

4.2.3 Route fitness

In genetic algorithms, the fitness function describes the optimisation objective, as used in the TTDP. An appropriate fitness function allows us to quantitatively assess a chromosome (i.e., route's) suitability as a candidate under the user specified criteria. To define this fitness function, we begin by assigning a score to each marina in a route, which is akin to the "profit" collected by the user for visiting it (as per the TTDP). In TTDP literature, profit is typically inferred from statistics based on social network user interactions with a POI (e.g. likes, check-ins, photos taken). However, these metrics may be unsuitable for sailing trips. First, not all ports exist as POIs in social networks. For those that do, interactions can be made by anyone (e.g. the marinas in most cities are a popular area for strolling and have several cafe/bar establishments—the vast majority of tips and comments on Foursquare and Facebook are from locals who frequent the area). Therefore the number of interactions does not cover the same quality semantics such as for museums (interactions by locals are fewer than those by tourists), or restaurants (interactions by locals can overwhelm those by tourists). One could consider a proxy metric such as summing the number of POIs around a marina, or the check-ins at these venues, e.g. as in Gavalas et al. (2019). To avoid the problems associated with inferring profit from social network interactions, we use instead a metric of diversity (Shannon Index, $H' = -\sum_{i=1}^{R} p_i \times ln(p_i)$), which is often used in the context of ecology science to measure biodiversity (number of species, and individuals per species). This metric quantifies entropy, and though originally intended for strings of text, has been widely adapted in ecology to obtain a measure of the number of individuals (POIs) and species (POI categories) in an environment (around a marina). A high H' demonstrates larger diversity, while, at minimum, it can take the value 0 (only one species present). Therefore, in our case, a marina with a high H' is more worth visiting, as there is a wider diversity of attractions around it. Conceptually thus, a route's profit $P(r)$ is the sum of the profits accrued by visiting each marina i $P(r) = \sum_{i=1}^{N} H'_i$.

We accept that a profit by visiting a marina can be accrued without penalty, if the segment leading from the current location to that marina is within the bounds placed by the exogenous parameters *pref_sailtime, max_sailtime*. Otherwise, we detract from the profit to be accrued, according to the amount of time the segment is over, or under the exogenous parameter limits. In fact, we administer a bonus to the H' if within these limits, a penalty if under these limits (a shorter route at least allows for more relaxation time in the area) and a harsher penalty if over the limits (since this becomes tiring, and prevents users from having enough time to explore the destination). As is shown in Eq. 1, a bonus/penalty modifier $M(i)$ for the segment leading to marina i is calculated based on the temporal length t_i of the journey to reach marina i.

$$M(i) = \begin{cases} -H'_i \times |\frac{t - pref_sailtime}{pref\_sailtime}| & \text{if } t_i < pref\_sailtime \\ -2H'_i \times |\frac{t - max_sailtime}{max\_sailtime}| & \text{if } t_i < max\_sailtime \\ 0.2 \times H'_i & \text{if } pref\_sailtime \leq t_i \leq max\_sailtime \end{cases} \tag{1}$$

Therefore, the total route profit accrued for visiting each marina in the route's chromosome sequence becomes $P(r) = \sum_{i=1}^{N} H'_i + M(i)$. Furthermore, we assume that each marina represents an overnight destination. Therefore, after calculating route profits based on H', we also apply a bonus/penalty mechanism on the duration of the route t_r, by giving a bonus if the route is a precise fit to the trip_days parameter specified by the user, or a penalty according to the number of days it is off, as shown in Eq. 2.

$$B(r) = \begin{cases} 0.5 \times P(r) if t_r = trip\_days \\ -0.2 \times |t_r - trip\_days| & \text{if } t_i < trip\_days \\ -0.2 \times |t_r - trip\_days| & \text{if } t_i > trip\_days \end{cases} \tag{2}$$

As such, the final value V of a route r becomes $V(r) = P(r) + B(r)$

4.2.4 Selection, crossover and mutation

For the purposes of selecting routes from the population, these are first ranked according to fitness. We use elitism, which means that N most fit routes (as defined by the parameter elite_size) will always be selected for the production of the next generation. From the remaining population, we discard the bottom 20% and assign the rest a fitness-weighted probability of being selected for the production of the next generation, alongside the elites (also termed "Roulette wheel" selection). This ensures that some variability remains in the population.

The selected population is now ready for the breeding process (or "crossover"). In previous GA applications to the TTDP such as Zheng et al. (2017), chromosomes of the same length were crossed-over, but in our case we have a need to be able to breed across chromosomes of different lengths. We use an approach called partially-mapped crossover, restricting the size of cutting points according to the length of the shortest chromosome, and ensuring that the resulting child has a length between the longer and shorter parent. The process is shown in Fig. 11 and works as follows. In the first phase, marked (1) in Fig. 11, we determine the cutting points where chromosomes will be exchanged, by selecting a random start and end gene (position) in the shorter parent. These genes can't be the first or last one since we are looking to preserve the origin and destination marina in the child. In phase (2), an empty shell for the child chromosome is created, with a length equal to the rounded average of the length of its parents. The first and last position are occupied by the origin and destination city, which is the same in both parents. In phase (3) we transfer the genes from the shorter parent which reside inside the cutting region. Then we transfer

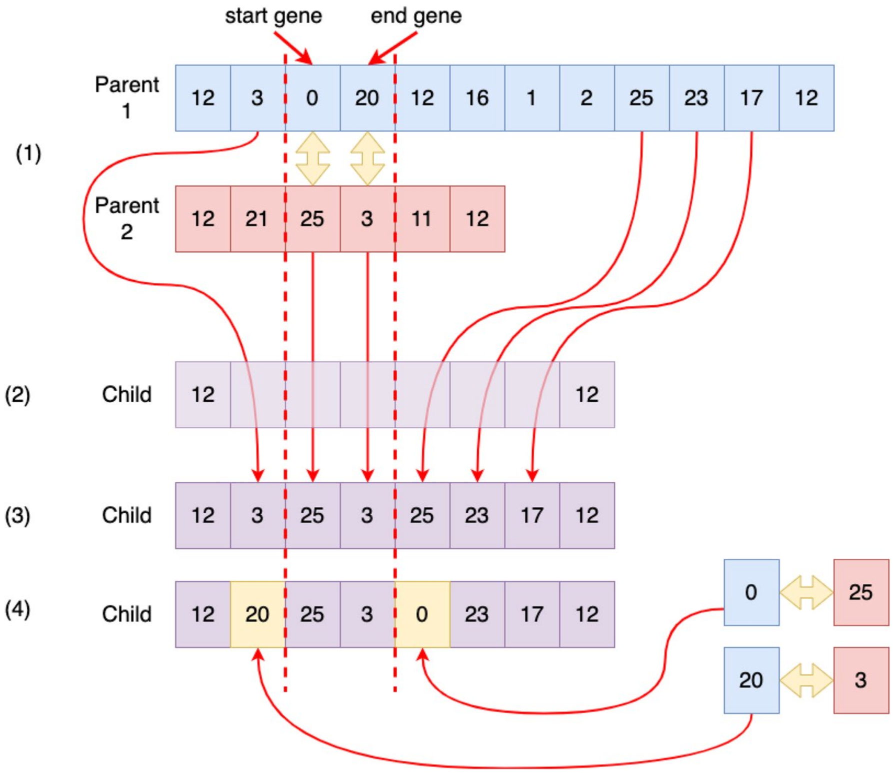


Fig. 11 Illustration of the partial-mapping crossover process for uneven-length chromosomes

the genes from the longer parent. Those preceding the cutting region are transfered as-is, while for the subsequent region, we start backwards from the longer parent and transfer as many as there are spaces in the child's shell. The final stage (4) is where we apply partial mapping to ensure the route does not contain any duplicates. The map is derived from the matching of chromosomes between the two parents, within the cutting region, and map-based transformations are executed only on the genes outside the cutting section (except, of course, the first and last ones).

Once children have been created, a mutation round is applied to the new population. For each route chromosome, we assign each of its genes a probability that it is swapped for another random gene inside the same chromosome. Of course, we exclude the possibility that it might be swapped for the start or end gene since we always want our routes to start and end in the same place. This provision can be removed if the origin/destination marinas are not a hard requirement for the user. Finally, at the end of the mutation round, we scan through the population and remove any possible duplicate routes that might have been generated.

Table 5 Algorithm evaluation parameters

Parameter	Description	Unit	Value
Max_sailtime	Preferred maximum duration of each sailing route segment	Hours	8
Pref_sailtime	Preferred minimum duration of each sailing route segment	Hours	4
Sail_speed	Average sailing speed	Km/h	8
Trip_days	Total duration of the trip	Days	6
pop_size	Size of the initial population of routes	Integer	200
elite_size	Number of ”elite” routes to keep for breeding the next generation	Integer	3
mutation_rate	Probability of route gene mutation	Floating point	0.02
n_solutions	Maximum number of candidate routes to return	Integer	5

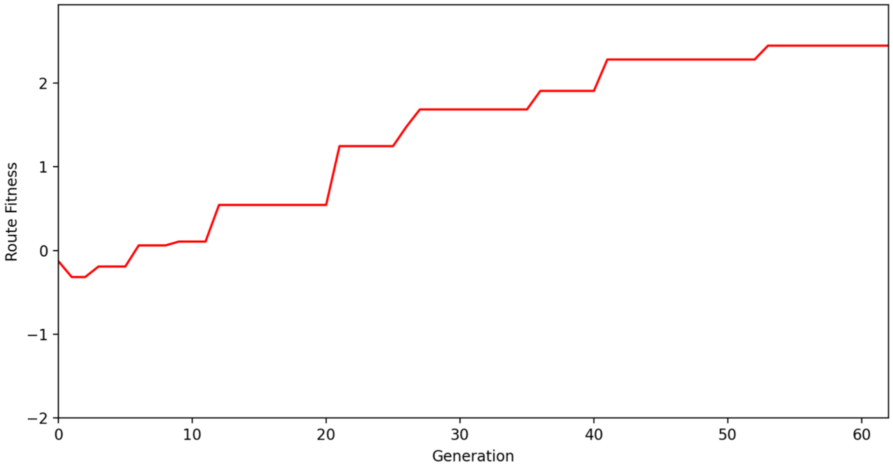
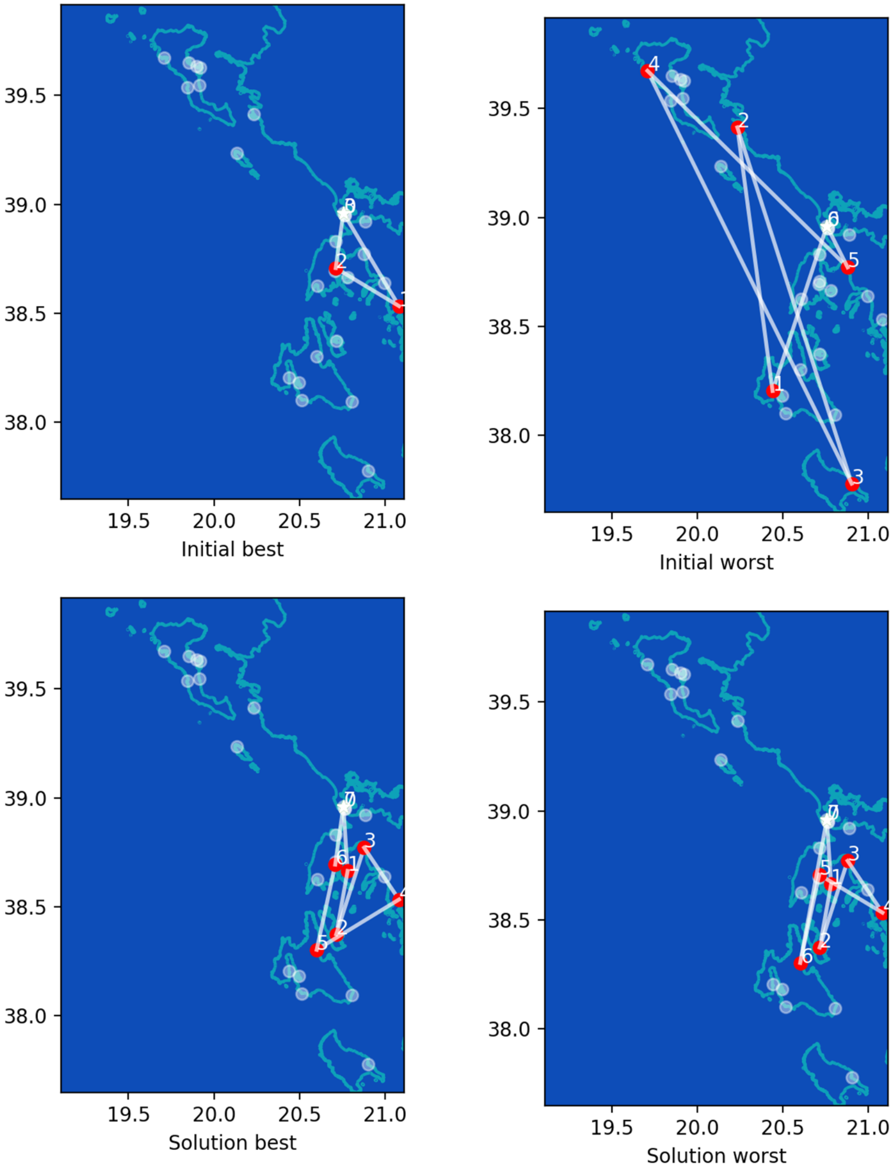


Fig. 12 Illustration of the GA execution progress and best-route fitness improvement using Preveza marina as the origin, and parameters as per Table 5

4.3 Results

We evaluated the algorithm’s ability to generate itineraries using the parameter values shown in Table 5, and for various marinas as the origin. As shown in Fig. 12 the algorithm makes good progress towards converging to solutions over the generations. Figure 13 shows the best and worst routes generated from the random initial population and the final set of solutions after execution of the GA. As we can see, in the random population the best solution was a very short trip (just 2 ports) while the worse was a longer trip which contained distances far beyond the maximum daily sailing time. The final set of solutions converges towards routes which fit the number of days specified by the user, and contain non-trivial (i.e. not too short) sailing distances between visited ports.



Initial best

Initial worst

Solution best

Solution worst

Fig. 13 Illustration of the best (left) and worst (right) routes in the initial random population (top) and final set of solutions (bottom) using Preveza marina as the origin, and parameters as per Table 5

One characteristic of the GA is that the final set of solution depends on the initial population that was generated. Due to randomness in the generative process, it is likely that multiple runs in with the same parameters will yield different results. To illustrate, Fig. 14 shows the results of 20 runs of the GA and some metrics for the best recommended route in each run. The top-left panel shows the number of

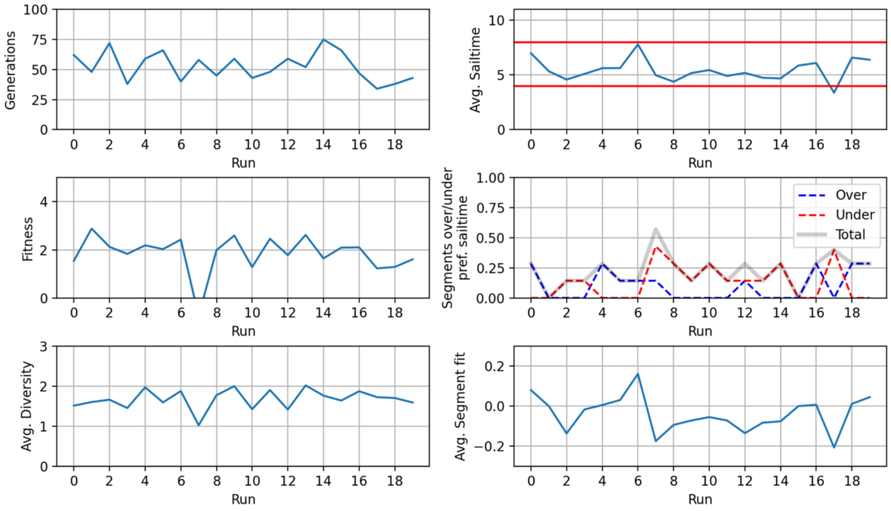


Fig. 14 Metrics on the best recommended route over 20 runs, using Preveza marina as the origin, and parameters as per Table 5

generations needed to complete the run. As we can see, these fluctuate due to the randomness in the number of routes that go forward to make the next generation during the selection process. In the mid-left panel, the fitness of the best route is shown. While there is some fluctuation, this hovers around the same performance, with the exception of one run, where the best route had a low (in fact, negative) fitness value. The bottom left panel shows the average H' (without bonus/penalties) across the ports in the top route.

A better understanding on how fitness values were derived comes from the right hand panels. The top right panel shows the average sailing time per route segment, while the two red lines show the user-specified *pref_sailtime* and *max_sailtime* parameters. We note that the algorithm produces routes that manage to stay within these confines, therefore we expect that not many penalties will be applied on the routes. The mid-right panel shows the percentage of segments in each route that were over, or under the thresholds specified by the user. We note that in most cases the proportion is less than 25%, and with most such segments being under the threshold, only small penalties are attracted. The last panel (bottom right) quantifies how much these segments were over or under the user thresholds on average. We can see that the average "fit" with the specified thresholds is generally negative, meaning that any segments over the threshold were only slightly so, where as those under the threshold were so by approximately 5–15%.

5 Discussion and conclusions

Summarising, we presented our methodology for automatically recommending sailing itineraries for holiday makers. We address the problem of determining berthing options by generating a queryable ontology of location-based data, populated through openly available sources. For the problem of route recommendation, our system leverages vessel density maps derived from AIS data, in order to generate realistic routes between any two waypoints in the system. Our solution is based on a modification of the A* algorithm's heuristic function, to bias path selection towards regions of increased vessel density, therefore producing paths that are balanced between reducing distance and following established navigation routes. The distance matrix generated from this process, paired with the resulting marina subset from queries directed towards the ontology, allows for the dynamic generation of graphs that can be used to provide customised itinerary recommendations.

To the best of our knowledge this paper is the first to address the problem of itinerary recommendation in the context of sailing holiday planning. In doing so, we believe that we can positively contribute towards the planning process of both experienced and novice tourists interested in sailing holidays, and therefore the further development of blue growth. Our work is a first step towards this direction, and therefore contains some limitations and scope for further work, which we outline next.

- Travel time calculations: The vessel density maps we depend on are currently only available for Europe through EMODnet. Since the EMODnet methodology for producing this data is openly publishable, given an appropriate level of financial investment in obtaining global AIS data, it would be possible, in the future, to produce vessel density maps with worldwide coverage, or at least, to apply the methodology to other geographic areas of interest. Another point to note is that travel time in our system is derived by assuming an average speed of sailing across any route segment. In reality, actual speed can be influenced by a range of parameters, not least of which is prevalent weather conditions. Future work can include more realistic estimates of achievable sailing speeds based on historical wind speed and direction data. Our system takes the average over the whole summer period to produce the vessel density map, but a modification could be to select data from only those periods in which the user is interested in travelling. In a sense, the present routing patterns would reflect the sailors' choices based on prevalent weather conditions over that period, and therefore would render routing options more realistic for the end user. However, EMODnet data is only available on a month-by-month basis (therefore posing an issue for users who wish to travel across two month interval periods, e.g. 28 June–12 July), and a custom way of generating vessel density maps from AIS data would be needed to enable such options.
- Itinerary recommendations: In this paper we demonstrated how our travel time calculations for sea travel can power a functional recommendation system. An obvious next step would be to improve the recommendation algorithm in sev-

eral ways. One obvious approach for future work is to extend the calculations for the fitness function by including more information about the destination. For this, there are further challenges to be solved. First, the challenge of calculating appropriate profits to each marina needs to be addressed. While ad-hoc assumptions based on quantitative measures such as social network interactions (check-ins, geotagged photos, likes), nearby POI counts or POI diversity metrics can be adopted, these may not necessarily be appropriate since there exists no literature to evaluate the importance of these surrounding POIs to yachters. The work by Shen et al. (2021) is an important first step towards understanding this problem, but the results are reported from relatively few participants (404) and are aggregated in overly coarse categories. A second challenge would be to adjust collected profits according to stated user preferences (profiles) for various POI categories or other aspects of the journey. In our system we have modelled stay time as a static value assumed equal for all stopovers (each marina is an overnight stop-over destination). In reality, users may prefer to spend longer in some places compared to others, for example, a marina which has many touristic attractions nearby can be used as a "base" for nearby explorations, either on land (e.g. museums, entertainment, hospitality) by sea (nearby beaches, anchorages), for several days. It would be interesting to examine the results by adapting stay time according to the density of nearby POIs, either as an absolute value, or as they align with users' preferences, and thus create itineraries with uneven stopover durations. Finally, it would be interesting to apply an approach such as for the VPP (Gavalas et al. 2019), in order to create detailed itineraries that span both in-land and sea-based activities, which would be a useful tool at a later stage of the pre-trip planning process. In this case, it would be interesting to add the thousands of beaches, fishing villages or other temporary anchoring places where a yacht could potentially settle for a short period of time, and attempt to generate hour-by-hour detailed itineraries, that might be more helpful in the next stage of a planning process.

- Evaluation of itinerary recommendations: Another limitation is the quality assessment of the produced routes for the generation of the distance matrix, and the itinerary recommendations. For the former, we employed a metric based on the popularity of routes derived from vessel density data. We consider this to be a realistic metric since it helps us determine how well the routes we produce fit the observed vessel behaviour, however, it would be good to validate the generated routes between marinas with experienced skippers, who have the best local knowledge. Similarly, we would like to validate the itineraries with experienced chartering professionals and experienced skippers, in order to assess the quality of the recommendations produced by the system, and also to obtain prospective customers' opinions on the recommendations as well. Therefore, we are working towards piloting our itinerary recommendation system through integration with the *SammyYacht* platform, developed by the company affiliated with our 2nd co-author. The platform is the premier berth space booking system for Greece and Cyprus, handling over 33% of licensed berth spaces in these countries, and with an average of 7,500 users in the summer period. This large user base will assist deployment of this pilot version and user-based evaluation, through the platform's website and mobile app.

- Scaling up: As discussed above, dynamic distance matrix calculations for each user-specified period can take significant amounts of time and are therefore not ideal from a service provision point of view. For example, distance matrix calculation took approximately 7 minutes for the work presented in this paper (27 marinas, using a 650 × 650 pixel map), using a relatively low-spec machine (Xeon E-2224 4-cores/4-threads, 8Gb RAM). A possible solution to this problem would be to have pre-calculated distance matrices at various temporal resolutions, to enable fast recommendation generation. Similarly, the genetic algorithm coded in Python, offers no parallelism currently. A typical run is between 6–15 seconds, and while reasonably fast, it's not fast enough for responding to users in real-time. The shortcomings and limitations of our approach can be addressed by carefully considering how to scale the system according to service provision requirements.

Overall our work presents a novel contribution towards addressing the problem of automatic itinerary recommendations for sailing holiday planning. The problem is largely neglected in existing literature but remains a fertile domain for future work, not just from a computer science perspective, but also given the importance of this tourism product choice for a significant portion of the market. We hope that our outline of future work will inspire further work in this area.

Acknowledgements Research in this paper was co-funded by the Hellenic Government and the European Regional Development Fund under the NSRF 2014-2020 program (Research and Innovation Strategies for Smart Specialisation - RIS3, Western Greece Region, SaMMY+ project, project ID: DEP5-0017992).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Andersson TD (2007) The tourist in the experience economy. Scand J Hos Tour 7(1):46–58. https://doi.org/10.1080/15022250701224035

Benevolo C, Spinelli R (2021) Benefit segmentation of pleasure boaters in Mediterranean marinas: a proposal. Int J Tour Res 23(1):134–145. https://doi.org/10.1002/jtr.2403. https://onlinelibrary.wiley.com/doi/abs/10.1002/jtr.2403

Cabrera Rivera L, Vilches-Blázquez LM, Torres-Ruiz M, Moreno Ibarra MA (2015) Semantic recommender system for touristic context based on linked data. In: Popovich V, Claramunt C, Schrenk M, Korolenko K, Gensel J (eds) Information fusion and geographic information systems (IF&GIS' 2015): deep virtualization for mobile gis, lecture notes in geoinformation and cartography. Springer International Publishing, Cham, pp 77–89, https://doi.org/10.1007/978-3-319-16667-4_5

Chen C, Zhang D, Guo B, Ma X, Pan G, Wu Z (2015) TripPlanner: personalized trip planning leveraging heterogeneous crowdsourced digital footprints. IEEE Trans Intell Transp Syst 16(3):1259–1273. https://doi.org/10.1109/TITS.2014.2357835

Chen JH, Chao KM, Shah N (2013) Hybrid Recommendation System for Tourism. In: 2013 IEEE 10th International Conference on E-Business Engineering, pp 156–161, https://doi.org/10.1109/ICEBE.2013.24

Cristian A, Marshall L, Negrea M, Stoichescu F, Cao P, Menache I (2021) Multi-itinerary optimization as cloud service. Commun ACM 64(11):121–129. https://doi.org/10.1145/3485626

Ecorys (2015) Study on the competitiveness of the recreational boating sector. Tech. rep., European Competitiveness and Sustainable Industrial Policy Consortium, Rotterdam, https://www.europeanboatingindustry.eu/images/Members_Documents/151124%20Competitiveness%20Study.pdf

Eijgelaar E, Peeters P, Neelis I, de Bruijn K, Dirven R (2021) Travelling large in 2019: the carbon footprint of Dutch holidaymakers in 2019 and the development since 2002. Tech. rep., BUas (Breda University of Applied Sciences)

European Commission (2020) The EU Blue Economy Report 2020. Tech. rep., Publications Office of the European Union, Luxembourg, https://blueindicators.ec.europa.eu/sites/default/files/2020_06_BlueEconomy-2020-LD_FINAL-corrected-web-acrobat-pro.pdf

Ferrer-Rosell B, Coenders G, Marine-Roig E (2017) Is planning through the Internet (un)related to trip satisfaction? Inform Technol Tour 17(2):229–244. https://doi.org/10.1007/s40558-017-0082-7

Filipiak D, Węcel K, Stróżyna M, Michalak M, Abramowicz W (2020) Extracting maritime traffic networks from ais data using evolutionary algorithm. Bus Inform Syst Eng 62(5):435–450. https://doi.org/10.1007/s12599-020-00661-0

Fogli A, Sansonetti G (2019) Exploiting semantics for context-aware itinerary recommendation. Person Ubiquit Comput 23(2):215–231. https://doi.org/10.1007/s00779-018-01189-7

Friggstad Z, Gollapudi S, Kollias K, Sarlos T, Swamy C, Tomkins A (2018) Orienteering algorithms for generating travel itineraries. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Association for Computing Machinery, New York, NY, USA, WSDM '18, pp 180–188, https://doi.org/10.1145/3159652.3159697

Gavalas D, Konstantopoulos C, Mastakas K, Pantziou G (2014) A survey on algorithmic approaches for solving tourist trip design problems. J Heuris 20(3):291–328. https://doi.org/10.1007/s10732-014-9242-5

Gavalas D, Konstantopoulos C, Pantziou G, Vathis N (2019) An efficient heuristic for the vacation planning problem. In: 2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON), p 0803–0808, https://doi.org/10.1109/UEMCON47517.2019.8993068

Gunawan A, Lau HC, Vansteenwegen P (2016) Orienteering problem: a survey of recent variants, solution approaches and applications. Eur J Oper Res 255(2):315–332. https://doi.org/10.1016/j.ejor.2016.04.059. https://www.sciencedirect.com/science/article/pii/S037722171630296X

He Y, Zheng Y, Deng J, Pan H (2016) Design and implementation of a POI collection and management system based on public map service. In: 2016 Fourth International Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS), pp 197–200, https://doi.org/10.1109/UPINLBS.2016.7809971

Heikkinen A, Okkonen A, Karhu A, Koskela T (2014) A distributed POI data model based on the entity-component approach. In: 2014 IEEE Symposium on Computers and Communications (ISCC), pp 1–6, https://doi.org/10.1109/ISCC.2014.6912570

Hoch SJ, Deighton J (1989) Managing what consumers learn from experience. J Market 53(2):1–20. https://doi.org/10.1177/002224298905300201

Kim JH, Ritchie JRB, McCormick B (2012) Development of a scale to measure memorable tourism experiences. J Travel Res 51(1):12–25. https://doi.org/10.1177/0047287510385467

Kuhlemann S, Tierney K (2020) A genetic algorithm for finding realistic sea routes considering the weather. Journal of Heuristics 26(6):801–825. https://doi.org/10.1007/s10732-020-09449-7

Kurosawa K, Uchiyama Y, Kosako T (2020) Development of a numerical marine weather routing system for coastal and marginal seas using regional oceanic and atmospheric simulations. Ocean Eng 195:106706. https://doi.org/10.1016/j.oceaneng.2019.106706. https://www.sciencedirect.com/science/article/pii/S0029801819308200

Łapko A (2019) The possibility of using online tools to increase the attractiveness of a nautical tourism product. DIEM : Dubrovnik Int Econ Meet 4(1):127–134, https://hrcak.srce.hr/228698

Lim KH, Chan J, Leckie C, Karunasekera S (2018) Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. Knowl Inform Syst 54(2):375–406. https://doi.org/10.1007/s10115-017-1056-y

Lim KH, Chan J, Karunasekera S, Leckie C (2019) Tour recommendation and trip planning using location-based social media: a survey. Knowl Inform Syst 60(3):1247–1275. https://doi.org/10.1007/s10115-018-1297-4

Mikulić J, Krešić D, Kožić I (2015) Critical factors of the maritime yachting tourism experience: an impact-asymmetry analysis of principal components. J Travel Tour Mark 32(sup1):S30–S41. https://doi.org/10.1080/10548408.2014.981628

Nualnim N, Phuaksawat C (2010) Customer's selection criteria when booking yacht online. Master's thesis, Mälardalen University

Paker N, Vural CA (2016) Customer segmentation for marinas: evaluating marinas as destinations. Tour Manag 56:156–171. https://doi.org/10.1016/j.tourman.2016.03.024. https://www.sciencedirect.com/science/article/pii/S0261517716300395

Palumbo R, Thompson L, Thakur G (2019) SONET: a semantic ontological network graph for managing points of interest data heterogeneity. In: Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Geospatial Humanities, Association for Computing Machinery, New York, NY, USA, GeoHumanities '19, pp 1–6, https://doi.org/10.1145/3356991.3365474

Patroumpas K, Skoutas D, Mandilaras G, Giannopoulos G, Athanasiou S (2019) Exposing points of interest as linked geospatial data. in: proceedings of the 16th international symposium on spatial and temporal databases, association for computing machinery, New York, NY, USA, SSTD '19, pp 21–30, https://doi.org/10.1145/3340964.3340976

Pranita D (2020) How digital capabilities can influence the co-creation of the yacht-tourism experience: a case study of indonesia's marine tourism destinations. In: 3rd International Conference on Vocational Higher Education (ICVHE 2018), Atlantis Press, pp 395–403, https://doi.org/10.2991/assehr.k.200331.172, https://www.atlantis-press.com/proceedings/icvhe-18/125938340

Shen Y, Kokkranikal J, Christensen CP, Morrison AM (2021) Perceived importance of and satisfaction with marina attributes in sailing tourism experiences: a kano model approach. J Outdoor Recreat Tour 35. https://doi.org/10.1016/j.jort.2021.100402. https://www.sciencedirect.com/science/article/pii/S2213078021000384

Sheng P, Yin J (2018) Extracting shipping route patterns by trajectory clustering model based on automatic identification system data. Sustainability 10(7):2327. https://doi.org/10.3390/su10072327. https://www.mdpi.com/2071-1050/10/7/2327

Shin YW, Abebe M, Noh Y, Lee S, Lee I, Kim D, Bae J, Kim KC (2020) Near-optimal weather routing by using improved A* algorithm. Appl Sci 10(17):6010. https://doi.org/10.3390/app10176010. https://www.mdpi.com/2076-3417/10/17/6010

Souffriau W, Vansteenwegen P (2010) Tourist trip planning functionalities: State–of–the–art and future. In: Daniel F, Facca FM (eds) Current trends in web engineering, Springer, Lecture Notes in Computer Science, p 474–485, https://doi.org/10.1007/978-3-642-16985-4_46

Strulak-Wójcikiewicz R, Wagner N, Łapko A, Hącia E (2020) Applying the business model canvas to design the e-platform for sailing tourism. Proc Comput Sci 176:1643–1651. https://doi.org/10.1016/j.procs.2020.09.188. https://www.sciencedirect.com/science/article/pii/S1877050920320901

Tenemaza M, Luján-Mora S, De Antonio A, Ramírez J (2020) Improving itinerary recommendations for tourists through metaheuristic algorithms: an optimization proposal. IEEE Access 8:79003–79023. https://doi.org/10.1109/ACCESS.2020.2990348

Tenemaza M, Luján-Mora S, De Antonio A, Ramírez J (2020) Improving itinerary recommendations for tourists through metaheuristic algorithms: an optimization proposal. IEEE Access 8:79003–79023. https://doi.org/10.1109/ACCESS.2020.2990348

Vansteenwegen P, Van Oudheusden D (2007) The mobile tourist guide: an or opportunity. OR Insight 20(3):21–27. https://doi.org/10.1057/ori.2007.17

Wang H, Mao W, Eriksson L (2019) A three-dimensional Dijkstra's algorithm for multi-objective ship voyage optimization. Ocean Eng186. https://doi.org/10.1016/j.oceaneng.2019.106131. https://www.sciencedirect.com/science/article/pii/S0029801819303208

Wang Y, Zhang R, Qian L (2018) An improved A* algorithm based on hesitant fuzzy set theory for multi-criteria arctic route planning. Symmetry 10(12):765. https://doi.org/10.3390/sym10120765. https://www.mdpi.com/2073-8994/10/12/765

Yochum P, Chang L, Gu T, Zhu M (2020) An adaptive genetic algorithm for personalized itinerary planning. IEEE Access 8:88147–88157. https://doi.org/10.1109/ACCESS.2020.2990916

Zhang C, Liang H, Wang K, Sun J (2015) Personalized trip recommendation with poi availability and uncertain traveling time. In: Proceedings of the 24th ACM International on Conference on

Information and Knowledge Management, Association for Computing Machinery, New York, NY, USA, CIKM ’15, pp 911–920, https://doi.org/10.1145/2806416.2806558

Zheng W, Liao Z, Qin J (2017) Using a four-step heuristic algorithm to design personalized day tour route within a tourist attraction. Tour Manag 62:335–349. https://doi.org/10.1016/j.tourman.2017.05.006. https://www.sciencedirect.com/science/article/pii/S0261517717301061