AntMapper: An Ant Colony-Based Map Matching Approach for Trajectory-Based Applications

Yue-Jiao Gong, *Member, IEEE*, En Chen, *Member, IEEE*, Xinglin Zhang, *Member, IEEE*, Lionel M. Ni, *Fellow, IEEE*, and Jun Zhang, *Fellow, IEEE*

***Abstract*— Many trajectory-based applications require an essential step of mapping raw GPS trajectories onto the digital road network accurately. This task, commonly referred to as map matching, is challenging due to the measurement error of GPS devices in critical environment and the sampling error caused by long sampling intervals. Traditional algorithms focus on either a local or a global perspective to deal with the problem. To further improve the performance, this paper develops a novel map matching model that considers local geometric/topological information and a global similarity measure simultaneously. To accomplish the optimization goal in this complex model, we adopt an ant colony optimization algorithm that mimics the path finding process of ants transporting food in nature. The algorithm utilizes both local heuristic and global fitness to search the global optimum of the model. Experimental results verify that the proposed algorithm is able to provide accurate map matching results within a relatively short execution time.**

***Index Terms*— Ant colony optimization, big trajectory data, GPS, map matching, road network.**

I. Introduction

THE proliferation of GPS-embedded vehicles has enabled crowdsourcing of a large amount of vehicle trajectory data. The "big trajectory data" facilitates the emergence of many data-driven trajectory-based applications, such as route recommendation [1], traverse time estimation [2], traffic dynamics analysis [3], fraud detection [4], and urban planning [5]. These applications require an important preprocessing step, commonly referred to as *Map Matching*, to map each raw trajectory obtained by the GPS onto the road network.

Map matching is always indispensable and crucial because of the following two reasons. First, the tracking data produced by GPS is not accurate. Even using the state-of-the-art devices, *measurement error* can be detected due to the urban canyon effect or some other signal interferences. Meanwhile, the limited storage and transmission bandwidth restrict the sampling intervals to a range from tens of seconds to a few minutes, resulting in *sampling error* of the trajectories. Map matching is therefore needed to correct these errors. Second, many applications are based on the statistical flow data on the digital road network. It is more important to identify which road nodes and segments are traversed by vehicles than to obtain the specific geographic coordinates of vehicles. Therefore, another effect of map matching is to discretize the GPS coordinates into a series of road segment IDs, which plays an indispensable role in statistical analysis of big trajectory data.

Because of the ubiquity and importance of map matching in trajectory-based applications, significant attention has been paid to this area recently. The operation of map matching can be performed either in an online mode that handles streaming GPS data or an offline mode that possesses the integrated GPS trajectory information. Existing approaches conduct map matching in a *local*, *incremental*, or *global* manner, differing in the portion of trajectory considered in each matching step. Local and incremental methods can be applied to both online and offline modes, whereas the global methods are suitable only for the offline mode [6]. This paper focuses on offline map matching. Typical techniques include point-to-segment projection [7], Fréchet distance analysis [8], weight-based method [9]–[12], hidden Markov model (HMM) [13]–[15], Bayesian Inference [16], [17], belief function theory [18], and fuzzy control theory [19]–[21]. However, the local and incremental map matching algorithms endure low accuracy, whereas the global algorithms are much more time consuming. How to develop a powerful algorithm that possesses both good effectiveness and high efficiency in map matching remains to be a challenging task.

Now, let us imagine the situation in which we human beings are engaged in matching the trajectory points to the digital map. In most cases, a human will exhibit the following behavioral pattern deliberately or inadvertently. During the decision process, s/he will alternatively zoom out the map to obtain the global shape information of the entire trajectory and zoom in the map to look into local details of the GPS locations and the surrounding road networks, until s/he is satisfied with the matching results. In other words, both local/incremental and global perspectives will be used when performing manual map matching. Mimicking such human intelligence, this paper develops a novel map matching approach that utilizes

Manuscript received April 19, 2016; revised October 2, 2016, December 28, 2016, and March 3, 2017; accepted April 14, 2017. Date of publication May 23, 2017; date of current version February 1, 2018. This work was supported in part by the National Natural Science Foundation of China (NSFC) Youth Project 61502542 and Project 61502178, in part by the NSFC Key Project 61332002, in part by the Macao FDCT Grant 149/2016/A, and in part by the University of Macau Grant SRG2015-00050-FST. The Associate Editor for this paper was Z. Ding.

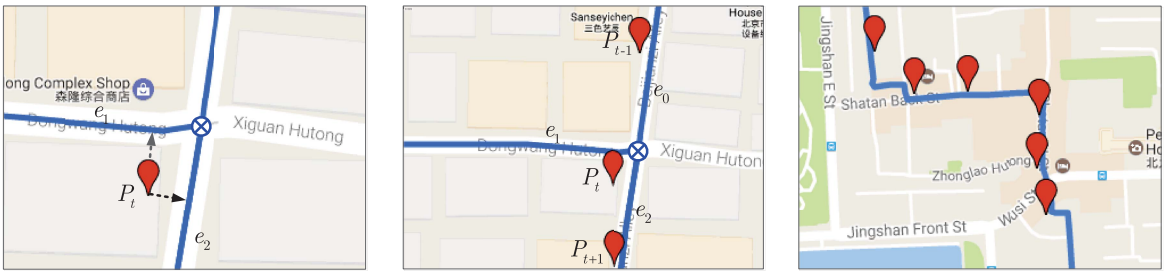
Y.-J. Gong, X. Zhang, and J. Zhang are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: gongyuejiao@gmail.com; zhxlinse@gmail.com; junzhang@ieee.org).

E. Chen is with the Fok Ying Tung Graduate School, The Hong Kong University of Science and Technology, Hong Kong, and also with Deepera Co., Ltd., Shenzhen 518060, China (e-mail: echen@ust.hk).

L. M. Ni is with the Faculty of Science and Technology, University of Macau, Taipa 999078, Macau (e-mail: ni@umac.mo).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2017.2697439



(a) Local perspective (b) Incremental perspective (c) Global perspective

Fig. 1. Illustration of map matching algorithms from local, incremental, and global perspectives.

local/incremental (denoted as "local" for simplification in the following statements of this section) and global information alternatively to improve the identification accuracy.

To realize this goal, an ant colony optimization (ACO) algorithm is designed for map matching, which, to the best of our knowledge, is the first attempt in this area. ACO is a swarm intelligence-based optimization method inspired by the cooperative foraging behavior of ants in nature, and it is originally designed to solve the *Traveling Salesman Problem (TSP)* [22], [23]. The motivation and potential benefits of applying ACO to the map matching problem are presented as follows. First, the solution construction of ACO considers both heuristic and pheromone information, which is well-suited to the requirement of employing both local and global information in the proposed map matching technique, as will be explained in the following sections. Second, ACO belongs to the category of global optimization methods. The algorithm is capable of locating the global optimum, and thus it can provide better matching results than the local search-based methods that return locally optimal solutions. Third, and importantly, the time complexity of ACO is sublinear to the problem size. Thus, although considering both local and global information, the proposed ant algorithm possesses high time efficiency. Experimental results on synthetic and real-life data verify the promising performance of the proposed algorithm.

The rest of this paper is organized as follows. Section II reviews related works. Section III describes the system model we developed for map matching, followed by the implementations of an ACO-based solver in Section IV. In Section V, experiments and comparisons are presented and discussed. Finally, conclusions are drawn in Section VI.

II. Related Work and Challenges

In the literature, a number of map matching approaches have been developed, which are generally divided into four categories by Quddus *et al.* [24]: geometric algorithms [7], [8], topological algorithms [9], [11], probabilistic algorithms [13]–[15], and advanced algorithms [19]–[21]. In this paper, we categorize the algorithms into local, incremental, and global groups by the different ranges of sight used in performing the matching operation.

- Local map matching algorithms [7], [25] consider a single GPS point on the trajectory at a time, and hence they are context free. As illustrated in Fig. 1(a), these algorithms are identical to constantly zooming in the underlying map until that only one GPS point is located within the sight, and then they perform map matching of the point in such a local scenario. Although possessing advantages of being fast and easy to implement, the local algorithms endure a low correct identification rate because they ignore temporal and spatial relationships between points on the trajectory. For example, in the case of Fig. 1(a), it is difficult to determine which edge, e_1 or e_2, should point P_t be matched to. To relieve this problem, in [26], after using a local hardware-based mechanism to correct the GPS errors, a data-driven algorithm is performed to improve the map matching accuracy.
- Incremental map matching algorithms [9], [11], [12], [18]–[21] take into account both the current GPS point and its link to the neighboring points, whose range of sight is slightly bigger than that of local algorithms. By taking incremental links into consideration, it is possible to utilize more information, such as topological information, travel distance, and transition probability, to improve the performance of map matching. As illustrated in Fig. 1(b), it is now clear that P_t should be located on edge e_2 rather than edge e_1, when the positions of P_{t-1} and P_{t+1} are considered. Nevertheless, the correct identification rate of incremental algorithms decreases with the prolonging of the sampling interval. Besides, for the algorithms which construct the matched path in a greedy manner, the accumulated error during the path construction will lower the effectiveness of incremental map matching.
- Global map matching algorithms [8], [27] possess a global perspective as in Fig. 1(c) and map the entire trajectory to a path in the road network based on similarity/distance measures. Recently, supervised learning has been applied to discover driver preferences from historical trajectory data, so as to utilize more prior information [17], [28]. The global information makes the algorithm less sensitive to the sampling rate. Moreover, as GPS points on the trajectory are considered in batches, global map matching does not encounter the error propagation problem of the incremental methods. However, in concurrence with the improved accuracy level, existing global map matching algorithms incur much higher computational costs than the other methods. Besides, the HMM-Viterbi algorithms [13]–[15] are sometimes categorized as global map matching methods since they align the entire trajectory to the road network. Nevertheless, based on the Markov assumptions, the

TABLE I
MAIN NOTATIONS

Symbol	Description
T, P	GPS trajectory, digital path
$p_i = (x_i, t_i, a_i)$	GPS sampling point, x_i is GPS coordinates, t_i is time stamp, a_i denotes other attributes
Δt	sampling interval
$G(V, E)$	road network, V is vertex set, E is edge set
c_i^j, C_i	candidate matched point, the set of candidate matched points (for GPS point x_i)
n	the number of GPS points in a trajectory
m	the number of candidate points in a candidate set
H, H_{norm}	heuristic, normalized heuristic
h_1, h_2, h_3	geometric error, heading error, routing error
ϵ	a small constant value
μ	mean best error of all candidate matches
ϕ_i, φ_i^j	heading of p_i, heading of c_i^j
F_{norm}	fitness of a path
Υ, Θ	feature vectors of trajectory and path
$(\Delta x_i.\text{lng}, \Delta x_i.\text{lat})$	geographic coordinate features of GPS point x_i
$(\Delta c_i.\text{lng}, \Delta c_i.\text{lat})$	geographic coordinate features of matched point c_i
$(LB.\text{lng}, LB.\text{lat})$	lower bound of the area
v_i	velocity of the vehicle at t_i
r_i	estimated routing distance between x_i and x_{i+1}
ρ_i	routing distance between the matched point c_i and c_{i+1} on the road network
$r_{\text{sum}}, \rho_{\text{sum}}$	summation of r_i, summation of ρ_i
u	the number of ants in the colony
J	transition probability
β	amplification coefficient
q_0	probability of exploitation in state transition
α, ρ	global evaporation rate, local evaporation rate
τ_0, τ	initial pheromone value, current pheromone value
P_{bsf}	the best-so-far path
g	the maximum number of iterations
$\lvert \cdot \rvert_{\measuredangle}$	radian difference between two angles
$\lVert \cdot \rVert_{gcd}$	great circle distance on earth
$\lVert \cdot \rVert_{route}$	distance of the shortest route on the road network
$rank_{\downarrow}(\cdot)$	ranking function in descending order
$S_m(\cdot)$	similarity metric / minimum ratio

algorithms are, in some sense, performed in an incremental manner.

In the following statements, for the convenience of description, the local and incremental algorithms are aggregated into a single category denoted as "local" since they both focus on local details of the trajectory and road network.

From the above review, it can be noticed that the local and global types of map matching methods both have pros and cons, which are complementary to some extents. Hence, a potential research direction is to hybridize these two kinds of methods. Moreover, as described in Section I, assuming that the map matching is performed manually, the individual responsible for the task proceeds with local and global matching behaviors alternatively. From this perspective, the concurrent use of local and global information is also interesting since it mimics human intelligence. However, combining the two methods is nontrivial, since traditional optimizers deal with only local heuristics or a global energy function. The solution proposed in this paper achieves this goal by adopting the nature-inspired ACO algorithm. The state transition in our proposed algorithm is based on two trails: *Heuristic* and *Pheromone*. On the one hand, the heuristic contains local information, whose guiding effect is similar to those of the traditional algorithms. We make an improvement by incorporating a rank function to make the heuristic be nonparametric and hence be robust for different working scenarios. On the other hand, the pheromone is self-adapted by the ant colony according to the global fitness information aggregating all local information and a similarity measure together. This way, at each local transition step, the ant is able to have a global sense of the trajectory matching quality. Because the concurrent use of heuristic and pheromone, the algorithm combines local and global information of the model to improve the map matching performance.

Fig. 2. A schematic diagram of the input and output of map matching.

At the end of this section, we would also like to introduce the recent emergence of vehicular cooperative map matching (VCMM) method [29], [30]. Different from the classical map matching model, in VCMM, each vehicle not only uses its own GPS information but also receives the GPS data from the nearby vehicles, so as to cooperatively obtain better results. This kind of model needs an additional vehicle-to-vehicle (V2V) communication channel to exchange data between vehicles, which is beyond the scope of this paper.

III. MODEL

This section presents the preliminaries of map matching and elaborates the design of our proposed information and system model. For clarity, the notations used are summarized in Table I.

A. Data Representation and Problem Formulation

As shown in Fig. 2, the fundamental goal of map matching is mapping the GPS trajectory data obtained by probe vehicles to the digital road network, in order to obtain a digital path that matches the raw trajectory.

Definition 1 (GPS Trajectory): A GPS trajectory is a GPS sequence $T : p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$ with $p_i = (x_i, t_i, a_i)$ $(i = 1, 2, \ldots, n)$ representing a sampling point, where x_i and t_i denote the GPS coordinates and time stamp respectively, and a_i represents some other attributes at t_i such as heading and velocity. Typically, the time interval between any two consecutive points p_i, $p_{i+1} \in T$ $(i < n)$ cannot exceed the *sampling interval* (denoted as Δt) of the trajectory, i.e., $t_{i+1} - t_i \leq \Delta t$.

Definition 2 (Road Network): A road network is a directed graph $G(V, E)$, where V is the vertex set representing the road intersections (or the starting and ending points of road segments), and $E = \{v_i \rightarrow v_j \mid v_i, v_j \in V\}$ is the edge set representing all the road segments in the map.

Definition 3 (Digital Path): Given a road network $G(V, E)$, a digital path is a sequence $P : e_1 \rightarrow e_2 \rightarrow \cdots \rightarrow e_n$, where e_i $(i = 1, 2, \ldots, n)$ satisfies $e_i \in E$ and denotes the road segment that is passed by the vehicle at time stamp t_i. Any two elements, e_i and $e_{i+1} \in P$ $(i < n)$, should be either connectable or the same in G.

Definition 4 (Candidate Matched Set): Given a GPS trajectory T, for each coordinates x_i of $p_i \in T$, we first calculate a set of candidate matches within the vicinity of the point. As an example, in Fig. 3, three road segments (e_1, e_2, e_3) surrounding the GPS coordinate x_i constitute three candidate matched edges of the point. Let c_i^1, c_i^2, and c_i^3 be the point-to-segment projection of x_i on e_1, e_2, and e_3, respectively. The three projected points are identified as the candidate matched points of x_i, with geometric errors $\|x_i - c_i^{\{1,2,3\}}\|_{gcd}$, where $\|\cdot\|_{gcd}$ denotes the great circle distance on earth. The vicinity area can be determined either by a given radius or by a maximum number of candidates, m. In this paper, we use the second rule, i.e., for each GPS point p_i, its candidate matched set C_i is consisted of the m closest edges (and/or points according to the practical requirement) with the smallest geometric errors.

Given the above definitions, we can define the map matching problem studied in this paper as follows:

Definition 5 (Map Matching): Given a GPS trajectory T, the map matching task is to select an optimally matched edge (and/or point) from the candidate set C_i of each point $p_i \in T$, so as to constitute a digital path P on the road network.

From the above definitions, it can be seen that map matching is an optimization problem, which is suitable to solve by an optimization algorithm.

B. Information Model

The proposed method utilizes 1) local geometric and topological information as *heuristic* and 2) the aggregation of the heuristics and a global similarity measure as *fitness* to facilitate finding the optimal map matching results.

1) Local Heuristic: The heuristic information on each possible link of the path defines the potential of the link to be chosen when constructing the optimal path.

Definition 6 (Heuristic): Given two candidate point-to-segment projected points c_i^s and c_{i+1}^t of consecutive points p_i and p_{i+1} on the trajectory, the local heuristic of constructing a path traversing from c_i^s to c_{i+1}^t is defined as

$$H(c_i^s \to c_{i+1}^t) = rank_{\downarrow}\Bigg(\prod_{j=1,2,3} (h_j(c_i^s \to c_{i+1}^t) + \epsilon) \text{ in } \Big\{ \prod_{j=1,2,3} (h_j(c_i^k \to c_{i+1}^l) + \epsilon) \mid \forall c_i^k \in C_i, \forall c_{i+1}^l \in C_{i+1} \Big\} \Bigg) \tag{1}$$

where h_1, h_2, and h_3 represent the geometric error, heading error, and routing error defined in (2)-(4), respectively; ϵ is a small constant set to 0.01 to avoid the situation where a zero factor eliminates the effect of the other two factors; the $rank_{\downarrow}(\cdot)$ function returns the ranking of the values inside the parentheses in the descending order.

The effects of using the rank function here are two-fold. First, as the descending rank function is a monotone decreasing function, the link with a smaller error value tends to possess higher heuristic, which is hence more likely to be chosen in the path construction process. More importantly, the rank-based nature makes the calculated heuristic be nonparametric, i.e., the value depends neither on the error distribution nor on the parameters. In this way, the developed algorithm can possess better robustness in different scenarios.

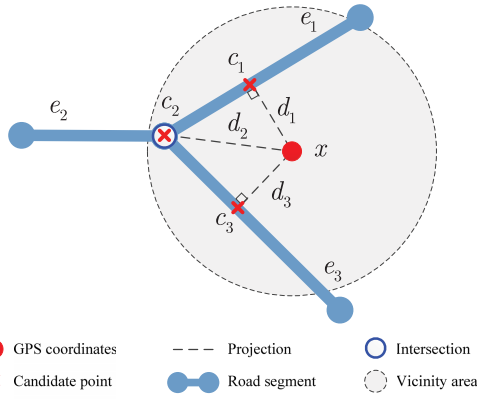


Fig. 3. Illustration of matching a GPS point to the surrounding road segments.

The geometric error h_1, heading error h_2, and routing error h_3 are defined as

$$h_1(c_i^s \to c_{i+1}^t) = \frac{\max\{\|x_i - c_i^s\|_{gcd}, \|x_{i+1} - c_{i+1}^t\|_{gcd}\}}{\mu} \tag{2}$$

$$h_2(c_i^s \to c_{i+1}^t) = \max\{|\phi_i - \varphi_i^s|_{\measuredangle}, |\phi_{i+1} - \varphi_{i+1}^t|_{\measuredangle}\} \tag{3}$$

$$h_3(c_i^s \to c_{i+1}^t) = \frac{\|c_i^s \to c_{i+1}^t\|_{route}}{\|x_i - x_{i+1}\|_{gcd}} \tag{4}$$

In (2), μ is a mean error parameter calculated by $\mu = (1/n) \cdot \sum_{j=1}^{n} (\|x_j - c_j^{k^*}\|_{gcd})$, where $c_j^{k^*}$ is the nearest projected neighbor of x_j. In (3), ϕ_i and ϕ_{i+1} denote the actual headings of the vehicle at points p_i and p_{i+1}; φ_i^s and φ_{i+1}^t denote the angles of the candidate edges on the road network; $|\cdot|_{\measuredangle}$ indicates the radian difference between two angles. It can be seen that the geometric and heading errors of a link are both dominated by the worse match at its source or target point. The bad matching result on either of these two points can lead the link to have a low heuristic value. In addition, the routing error on the link is calculated according to (4), where $\|c_i^s \to c_{i+1}^t\|_{route}$ is the traverse distance of the shortest route from c_i^s to c_{i+1}^t in the road network, $\|x_i - x_{i+1}\|_{gcd}$ is the great circle distance between x_i and x_{i+1} on earth.

After calculating the heuristic, we further employ a normalization scheme based on the arc tangent function to normalize the heuristic value into the range [0, 1]:

$$H_{norm}(c_i^s \to c_{i+1}^t) = 2 \cdot \arctan(H(c_i^s \to c_{i+1}^t))/\pi \tag{5}$$

where H_{norm} denotes the normalized local heuristic information to guide the path construction.

Note that our information model requires the heading data of vehicles. The heading information is useful for resolving ambiguities in discriminating two opposing carriageways. If the heading information is not available, an alternative way is to estimate it by consecutive GPS coordinates. The estimation could be inaccurate, especially when the sampling interval is long (our experimental investigation has verified this point). Fortunately, today most devices for tracking GPS data, such

as smartphones and navigators, are equipped with compass to measure the heading data.

2) Global Fitness: The term "fitness" is used to describe the ability of an organism to survive in a given environment in biology. More recently, this term has been widely adopted by the computing community and, more specifically, the computational intelligence (CI) community as a figure of merit to describe how close a solution is to achieve the objective of solving a problem. In this paper, we use fitness to denote the quality of global map matching of the entire trajectory.

Definition 7 (Fitness): Given a raw trajectory $T : p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$ and a digital path $P : e_1 \rightarrow e_2 \rightarrow \cdots \rightarrow e_n$ of the same length with the sequence of projected points $c_1 \rightarrow c_2 \rightarrow \cdots \rightarrow c_n$, the fitness of mapping T to P is the aggregation of the average heuristic value on each link $(c_i \rightarrow c_{i+1})$ and a global similarity measure:

$$F_{norm}(P) = \left(\frac{\sum_{i=1}^{n-1} H_{norm}(c_i \rightarrow c_{i+1})}{n-1} + S_m(\Upsilon(T), \Theta(P)) \right) / 2 \quad (6)$$

where $\Upsilon(T)$ and $\Theta(P)$ represent the feature vectors of T and P respectively, $S_m(\cdot)$ denotes a similarity metric that quantifies the correlation between T and P.

In the areas of pattern recognition and image processing, various metrics have been developed for years to quantify the similarity between the sequences of measurements for two given objects [31], [32]. Here in the proposed map matching method, we consider the raw trajectory and the matched path as two patterns, and use the *minimum ratio* [31] to evaluate the similarity between them.

Definition 8 (Minimum Ratio): If a pattern $Y = \{\beta_1, \beta_2, \ldots, \beta_l\}$ is a noise version of a pattern $X = \{\alpha_1, \alpha_2, \ldots, \alpha_l\}$, the minimum ratio quantifying the closeness of Y to X is defined as

$$S_m(X, Y) = \frac{1}{l} \sum_{i=1}^{l} \min \left\{ \frac{\alpha_i}{\beta_i}, \frac{\beta_i}{\alpha_i} \right\} \quad (7)$$

where l stands for the length of the feature vectors extracted from the objects (which is different from the trajectory length n in map matching). The more similar the patterns X and Y are, the larger the value $S_m(X, Y)$ is. Although simple in formulation, the minimum ratio, as a similarity metric, has the following properties [32]:

- A Graceful Range: $S_m \in [0, 1]$.
- Reflexivity: $S_m(X, Y) = 1$ if and only if $X = Y$.
- Symmetry: $S_m(X, Y) = S_m(Y, X)$.
- Triangle Inequality: $S_m(X, Y)S_m(Y, Z) \leq [S_m(X, Y) + S_m(Y, Z)]S_m(X, Z)$.

Moreover, as the ratio of intensity is used, this metric is independent with the scale of different features.

In the proposed method, the feature vector of the trajectory T is in a form of

$$\Upsilon(T) = \{\Delta x_1.\text{lng}, \Delta x_1.\text{lat}, \Delta x_2.\text{lng}, \Delta x_2.\text{lat}, \ldots, \Delta x_n.\text{lng}, \Delta x_n.\text{lat}, r_1, r_2, \ldots, r_{n-1}, r_{\text{sum}}\}$$

Here $\Delta x_i.\text{lng} = (x_i.\text{lng} - LB.\text{lng})$ and $\Delta x_i.\text{lat} = (x_i.\text{lat} - LB.\text{lat})$ represent the geographic coordinate features, where

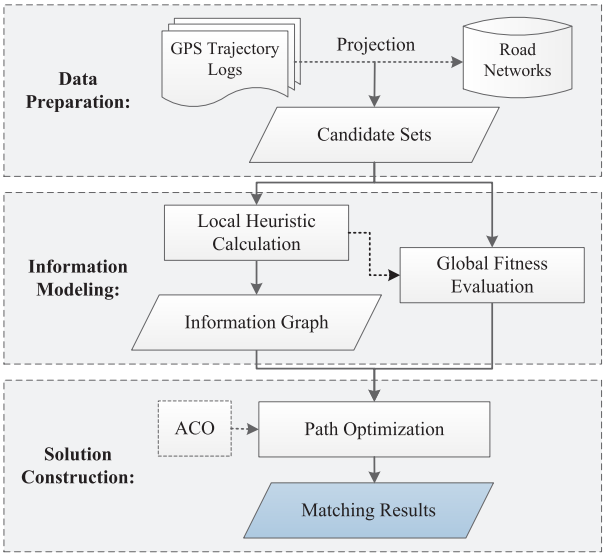


Fig. 4. Framework of the proposed map matching method.

$(x_i.\text{lng}, x_i.\text{lat})$ and $(LB.\text{lng}, LB.\text{lat})$ denote the (longitude, latitude) pairs of the position x_i and a lower bound LB of the area, respectively; r_i is the (roughly) estimated routing distance between x_i and x_{i+1} calculated in (8); r_{sum} is the summation of r_i.

$$r_i = \frac{(v_{i+1}/2 + v_i/2)}{2} \cdot (t_{i+1} - t_i) \quad (8)$$

Note that we divide each velocity by 2 by considering the acceleration and deceleration process of vehicles.

Correspondingly, the feature vector of the path P is defined as

$$\Theta(P) = \{\Delta c_1.\text{lng}, \Delta c_1.\text{lat}, \Delta c_2.\text{lng}, \Delta c_2.\text{lat}, \ldots, \Delta c_n.\text{lng}, \Delta c_n.\text{lat}, \rho_1, \rho_2, \ldots, \rho_{n-1}, \rho_{\text{sum}}\}$$

where $\Delta c_i.\text{lng} = (c_i.\text{lng} - LB.\text{lng})$ and $\Delta c_i.\text{lat} = (c_i.\text{lat} - LB.\text{lat})$ denote the geographic coordinate features of c_i; $\rho_i = \|c_{i+1} - c_i\|_{route}$ is the routing distance between c_i and c_{i+1} in the road network; and ρ_{sum} is the summation of ρ_i.

Now consider the fitness function (6). First, as we have $H_{norm}(c_i \rightarrow c_{i+1}) \in [0, 1]$ and $S_m(\Upsilon(T), \Theta(P)) \in [0, 1]$, it can be deduced that the fitness value of a path satisfies $F_{norm}(P) \in [0, 1]$. Second, note that the formulation is generic, providing flexibility in implementing the similarity measure. The optimizer used in this paper allows substituting the current implementation of $S_m(\cdot)$ by using different feature vectors of T and P or adopting a different similarity metric instead of the minimum ratio in the future.

C. System Model

Fig. 4 shows the framework of the proposed map matching method, which is composed of three modules: data preparation, information modeling, and solution finding. First, in the data preparation module, the GPS coordinates of the raw trajectory are projected onto the road network to build the candidate matched sets. Second, in the information modeling module, the heuristic of each candidate link is calculated and

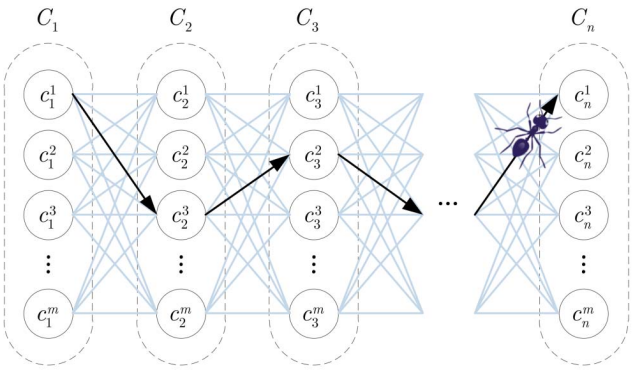


Fig. 5. Illustration of an ant constructing a path on the information graph. The candidate solution obtained is $c_1^1 \rightarrow c_2^3 \rightarrow c_3^2 \rightarrow \cdots \rightarrow c_n^1$.

stored in an information graph (as shown in Fig. 5, each vertex is a candidate matched points), and the global fitness function is formulated. Note that we should not evaluate the global fitness of a path in this module since we do not build a candidate path until the third module starts. Finally, in the solution finding module, a colony of ants is dispatched to cooperatively search the best path that matches the raw trajectory the most, taking into account both the heuristic on the information graph and the fitness evaluation of the paths built. We are going to describe this module, the AntMapper, in the next section.

IV. Ant Colony-Based Map Matching Algorithm

The main idea of ACO is derived from the natural phenomenon of how an ant colony transport food between the food source and the nest. It has been found by the biologists that, although ants have an undeveloped visual system, they are capable of finding the shortest path when transporting food. This relies on a medium named *Pheromone*. On the one hand, the ants will deposit pheromone on the ground they have passed. On the other hand, the amount of pheromone accumulated on the ground plays a decisive role in helping ants to select their paths. In the very beginning, the ants select the paths in a completely random manner. Since the ant colony move at approximately the same velocity, per unit time, a shorter path would be visited by more ants and hence it accumulates more pheromone than a longer path does. Guided by the pheromone, the followers are more likely to select the shorter paths, which further differentiates the amount of pheromone on different paths. Meanwhile, the pheromone on all paths evaporates at the same rate. In this way, after a short time period, all the ants will traverse the shortest path between the food source and the nest, since the pheromone value on this path now dominates those on the other paths.

Simulating the above natural intelligent process, ACO is originally designed to solve the shortest path problems such as TSP. Owing to the conceptual simplicity, global optimization ability, and high operational efficiency, the algorithm has lately been successfully applied to many other optimization problems in various fields, such as vehicle routing [33]–[35], railway scheduling [36]–[38], uplink scheduling [39], and project scheduling [40].

Algorithm 1 AntMapper

```
/* Initialization */
Set the pheromone on all links of the information graph to an equal initial value;
/* Main Loop */
while Termination Rule is not met do
    Place each ant at a random starting point in the candidate set of the first GPS point (i = 1);
    for i = 2 to n do
        Perform State Transition Rule to move each ant to the next point;
        Perform Local Update Rule to update the pheromone on the link which is just passed by each ant;
    end for
    Evaluate the fitness values of all ants according to (6);
    Identify the best-so-far path P_bsf;
    Perform Global Update Rule to update the pheromone on path P_bsf;
end while
return Path P_bsf.
```

In this paper, we apply ACO to deal with the map matching problem and obtain an AntMapper approach. After obtaining the candidate sets and constructing the information graph as described in Section III, the AntMapper works as follows. In the initialization stage, each link on the information graph is assigned with an equal amount of pheromone value. Then, a colony of u ants are dispatched to accomplish the map matching task, where the colony size u is a parameter of ACO. Each ant is initially placed at a random candidate matched point in set C_1 for the first GPS point. Repeatedly, the ants apply a *State Transition Rule* to select the next candidate point based on the heuristic and pheromone information. While walking on the information graph, the ants modify the amount of pheromone on the links they pass according to a *Local Update Rule*. Once entire paths for the trajectory are built, the ant colony applies a *Global Update Rule* to further amend the pheromone based on the evaluated fitness values. Fig. 5 depicts an example of the path construction of an ant on the information graph, with the candidate solution being $c_1^1 \rightarrow c_2^3 \rightarrow c_3^2 \rightarrow \cdots \rightarrow c_n^1$. Afterwards, the ants start a new round of path construction iteratively, until a *Termination Rule* is met. The procedure of AntMapper is shown in **Algorithm 1**. Next, we will describe the state transition, global and local update, and termination rules, respectively.

A. State Transition Rule

This rule determines how an ant on one position uses heuristic and pheromone to select its next position.

Definition 9 (Transition Probability): For an ant that is currently positioned at point c_i^s, the transition probability of selecting c_{i+1}^t $(t = 1, 2, \cdots, m)$ from the candidate set C_{i+1} as its next position is

$$J(c_{i+1}^t) = \frac{H_{norm}(c_i^s \rightarrow c_{i+1}^t) \cdot [\tau(c_i^s \rightarrow c_{i+1}^t)]^\beta}{\sum_{k=1}^{m} (H_{norm}(c_i^s \rightarrow c_{i+1}^k) \cdot [\tau(c_i^s \rightarrow c_{i+1}^k)]^\beta)} \tag{9}$$

where $\tau\ (c_i^s \rightarrow c_i^t)$ is the pheromone deposited on the link $(c_i^s \rightarrow c_{i+1}^t)$, the exponential coefficient β is used to amplify the relative importance of the pheromone with respect to the heuristic in calculating the probability.

It can be seen that the link with a larger heuristic value and a greater amount of pheromone is more favourable.

Definition 10 (State Transition): In state transition, the ant positioned at c_i^s selects the next point c_{i+1}^t according to

$$c_{i+1}^t = \begin{cases} \arg\max_{c_{i+1}^k \in C_{i+1}} J(c_{i+1}^k), & \text{if } q \leq q_0 \\ \qquad \text{(exploitation)} & \\ random\_proportional\_selection, & \text{otherwise} \\ \qquad \text{(biased exploration)} & \end{cases} \tag{10}$$

where q is a random number uniformly generated in $[0, 1]$, q_0 is a parameter of ACO that determines the probability of an ant to perform exploitation. If $q \leq q_0$, the ant selects the point with the largest transition probability. Otherwise, with probability $(1 - q_0)$, the ant conducts the biased exploration where a random proportional selection rule (such as the roulette wheel selection [41], [42]) is used to choose the next point based on the transition probability.

B. Pheromone Update Rule

At the beginning, the pheromone on all links is initialized to a very small value τ_0, which is also a lower bound of the pheromone. Then, the ants perform a global update rule and a local update rule defined as follows to modify the amount of pheromone on different links during the optimization process.

1) Global Update Rule: In each iteration, the u ants conduct one round of path construction and evaluate the fitness values according to (6). The ant that constructs the best-so-far path P_{bsf} with the highest fitness value is considered as the dominating ant. Only the dominating ant is allowed to deposit pheromone on the path.

Definition 11 (Global Update): For each link $(c_i^s \rightarrow c_{i+1}^t)$ in P_{bsf}, the pheromone value is updated by

$$\tau(c_i^s \rightarrow c_{i+1}^t) = (1 - \alpha) \cdot \tau(c_i^s \rightarrow c_{i+1}^t) + \alpha \cdot F_{norm}(P_{bsf}) \tag{11}$$

where $\alpha \in (0, 1)$ is a parameter defining the global evaporation rate of pheromone.

It can be seen that, the amount of pheromone reinforcement depends on the fitness of P_{bsf}. The better the path is, the more pheromone the links would receive. These links are identified as promising links that will attract more ants in the following search process.

2) Local Update Rule: On the other hand, the local update rule is executed by all ants right after each state transition step to shuffle the desirability of different paths.

Definition 12 (Local Update): Every time an ant passes a link $(c_i^s \rightarrow c_{i+1}^t)$, it modifies the pheromone value on the link by

$$\tau(c_i^s \rightarrow c_{i+1}^t) = (1 - \rho) \cdot \tau(c_i^s \rightarrow c_{i+1}^t) + \rho \cdot \tau_0 \tag{12}$$

where $\rho \in (0, 1)$ is the local evaporation rate of pheromone.

It can be seen that, by performing the local update rule, the link that has been passed by an ant losses some pheromone (it becomes closer to τ_0). Hence, the probability of choosing the same link for the following ants is reduced. The local update rule enhances the search diversity of ants and improves the global optimization ability of the algorithm.

C. Termination Rule

In accordance with traditional ACO algorithms, the AntMapper iterates the aforementioned path construction and pheromone update steps until a maximum number of iterations (denoted as g) is met. Considering the candidate position selection in the information graph as the elementary operation, the time complexity of the optimal path finding process of AntMapper is $O(g \cdot u \cdot n)$.

D. Discussions

At the end of this section, we compare ACO with the widely used Viterbi algorithm and summarize the advantages and limitations of using ACO for optimization.

1) Advantages: First, the Viterbi algorithm adopts the hidden markov model (HMM), which assumes that the observed sequence is based on memoryless noise. Namely, the ith observation depends only on the ith transition from point p_i to point p_{i+1}. Instead, ACO considers the global fitness function defined in Eq. (6), which involves all transition steps simultaneously. The pheromone information on the links is updated according to the fitness function (see the global update rule). ACO optimizes the fitness globally without making the similar assumptions as HMM, which enhances the effectiveness of map matching.

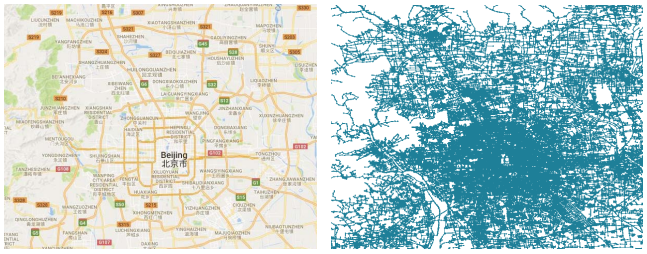
Second, the transition and emission probabilities used in the HMM-Viterbi algorithm are usually unknown a-priori and pre-estimated according to man-made rules. In comparison, ACO is an apriori metaheuristic algorithm. The pheromone is updated by the ant colony during the search process and is self-adapted to the problem nature. Guided by the pheromone, ACO is able to discover the underlying structure of the problem space, and it is more robust for a variety of problem instances.

Third, using ACO, the proposed map matching algorithm has advantages of high flexibility or expandability. It allows us to add, subtract, or modify the model without changing the solution-finding component of the system. For example, the system can be easily expanded by using a more complicated similarity measure instead of the minimum ratio to quantify the similarity between two data sequences.

2) Limitations: Although ACO is shown to possess powerful global optimization ability by a mass of experiments in various application fields [33]–[40], so far there is no complete mathematical proof to show that ACO will definitely find the global optimum. A few existing proofs made some assumptions to simplify the algorithm structures [43]. Thus, the performance of AntMapper is empirically derived.

V. Experimental Study

In this section, we conduct experiments on both synthetic and real data to study the performance of the proposed



(a) A map scene of Beijing (b) The digital road network of Beijing

Fig. 6. Road network of Beijing.

algorithm. The experimental results of AntMapper is compared with a local algorithm based on point-to-segment projection (P2S) [7], an incremental algorithm based on fuzzy logic (FL) [21], and a Viterbi algorithm based on HMM [14]. We introduce and describe the data instances, experimental setup, and results in detail in the rest of this section.

A. Data Preparation

As depicted in Fig. 6, the road network of Beijing is used as the underlying map for map matching. The road network consists of 171,504 vertices and 433,491 road segments.

1) Synthetic Dataset: The synthetic data are generated by our program of simulating vehicles traversing on the road network.[1] At five different intervals, namely, Δt = 10s, 30s, 1min, 2min, and 5min, the information of the current vehicle is recorded. The trajectory lengths are $n = 10$ (for longer trajectories in real case, they can be divided in batches for processing). We add bivariate normal noise to the recorded GPS coordinates with standard deviation 4, 8, 12, 16, 20, and 24m. This way, according to the three-sigma rule of normal distribution, within a probability of 99.7%, the GPS error ranges from 0 to 72m. The average velocity of different vehicles varies from 20 to 80km/h. As for the heading, normal noise with an error range of $\pm 15°$ has been added to the raw data. For each sampling interval, 600 GPS trajectories are simulated, therefore the total number of synthetic test instances used in the experiments is 3,000.

2) Real-Life Dataset: In addition, we also retrieved real-life data from a taxi company in Beijing and selected 40 trajectories that cover a wide range of spatial area. To obtain the ground-truth labels, human efforts have been spent to match each taxi trajectory to the digital road network manually. The sampling intervals of these real taxis differ a lot, which are generally within the range of 10s to 1min. Note that, due to the Surveying and Mapping Law of China [44], [45], the digital road network we use is an encrypted one that contains offsets from reality.[2] These offsets are considered as additional spatial errors to be corrected in the map matching task, which further increases the difficulty of solving the problem.

[1]As the map matching tasks in the dense area are more difficult than those in the area where the road segments are sparse, the data are generated within the central area of Beijing to increase difficulty.

[2]In the data preprocessing step, we include a procedure to filter all the taxi trajectories by a unique global shift $(0.0061, 0.0013)$ and rotation $(2\pi \cdot 10^{-8})$ in the GPS coordinates. The parameters are roughly estimated based on our human vision to align the GPS points close to the road network, which partially relieves the encryption matter.

TABLE II
COMPARISONS OF CSI AT DIFFERENT SAMPLING INTERVALS

Δt	P2S	FL	HMM	AntMapper
10s	50.28%	91.73%	95.62%	97.80%
30s	51.87%	90.82%	95.25%	96.95%
1min	52.40%	87.70%	93.48%	95.90%
2min	53.03%	82.23%	90.70%	95.72%
5min	52.93%	74.85%	82.30%	93.97%
mean	**52.10%**	**85.47%**	**91.47%**	**96.07%**

The real-life instances with human-labelled ground truth are tested in the second part of our experiments.

B. Experimental Setup

The parameter settings of P2S, FL, and HMM are based on the values recommended in their original papers [7] [14], [21]. In the proposed AntMapper, the number of candidate segments for each GPS points is set as $m = 6$. The parameters involved in the ACO algorithms empirically adopt the values that are recommended in the literature, namely, the ant colony size $u = 10$, the amplification coefficient $\beta = 0.5$, the probability in the state transition $q_0 = 0.9$, the global and local evaporation rates $\alpha = \rho = 0.1$, the initial pheromone value $\tau_0 = F_{norm}(P_{nn})/n$ where P_{nn} is a path produced in a nearest-neighbor manner considering the local heuristic. Besides, the number of iterations in ACO is commonly set to a value increasing linearly with the problem dimensionality in the literature. In this work, the problem dimensionality is 10 and the number of iterations is set as $g = 100$.

All algorithms are coded in C++ and executed on the same platform. We compare the four algorithms in terms of both the average CPU execution time of mapping each trajectory and the map matching accuracy measured by an external index named *Correct Segment Identification* (CSI).

$$CSI = \frac{\#correctly\ matched\ segments}{\#all\ segments} \tag{13}$$

C. Results on Synthetic Instances

In this subsection, we first compare the accuracy and efficiency of the map matching algorithms. Then, AntMapper is investigated with different settings of m (the number of candidate segments for each GPS points). The reported results are obtained under the noise with a standard deviation of 8m. Further, instances with different noise ranges are tested to show the performance of AntMapper on degraded data.

1) Accuracy Measure: The average CSI values obtained by the algorithms at different sampling intervals are reported in Table II. It can be observed from the table that, the proposed AntMapper performs the best with respect to all sampling intervals, followed by HMM, FL, and P2S. Particularly, when the sampling interval is short (e.g., 10 and 30s), AntMapper, HMM, and FL are all competitive, and achieve an accuracy level greater than 90%. This is because that, the GPS trajectories carry relatively sufficient local geometric and topological information in such cases, which helps to correctly match

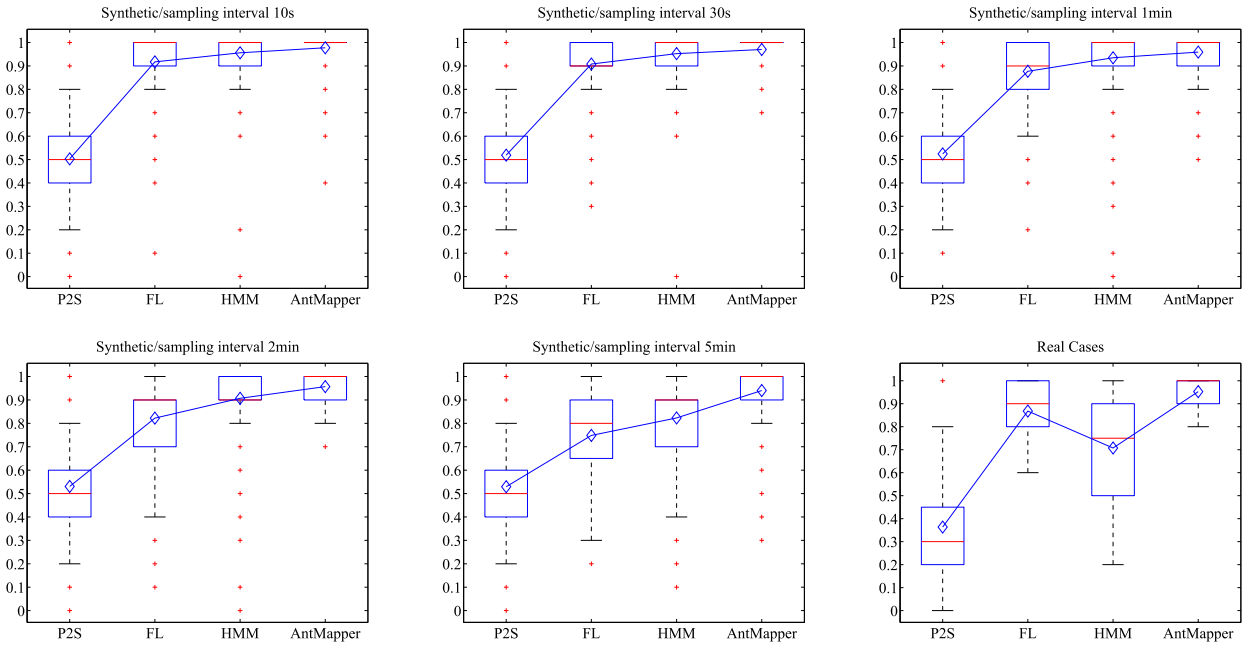


Fig. 7. Statistical results/box plots of the map matching accuracy on synthetic and real-life trajectory instances obtained by the four algorithms.

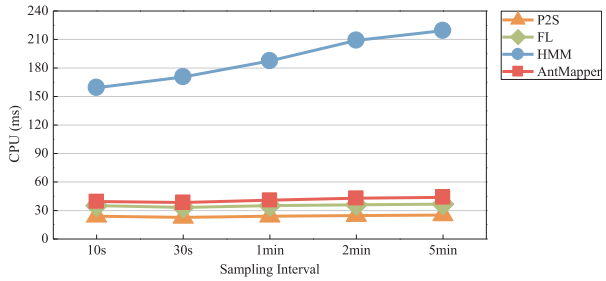


Fig. 8. Time efficiency of the map matching algorithms.

TABLE III
RESULTS OF HMMc, COMPARED WITH HMM AND AntMapper

	Algorithm	10s	30s	1min	2min	5min
CSI	HMMc	95.22%	95.12%	93.53%	90.55%	82.32%
	HMM	95.62%	95.25%	93.48%	90.70%	82.30%
	AntMapper	97.80%	96.95%	95.90%	95.72%	93.97%
CPU (ms)	HMMc	39.68	38.53	40.92	41.88	44.43
	HMM	159.29	170.62	187.44	209.08	219.38
	AntMapper	39.48	38.41	40.76	42.85	43.82

the segments. However, in many practical scenarios, long sampling intervals (or named low sampling rates) are used in order to save communication and energy costs. These instances endure higher local uncertainty, for which the local map matching algorithms become less effective. Considering the experimental results of Δt = 1, 2, and 5min, the performance of FL decreases to a great extent. In contrast, the HMM and AntMapper algorithms are more promising. The proposed AntMapper also outperforms HMM by utilizing both the local heuristics and the global similarity measure. Besides, the P2S algorithm obtains inferior results for all sampling intervals because it cannot distinguish directions for the two-way road segments.

In addition, we use box plots to illustrate the statistical results of 600 trajectories for each sampling interval, where the mean, median, upper quartile, lower quartile, maximum, minimum, and outliers are depicted. It can be clearly observed from Fig. 7 that AntMapper is more stable and reliable than the other algorithms.

2) Time Efficiency Measure: Fig. 8 shows the average CPU execution time of processing each trajectory by the four algorithms. Note that the local and incremental algorithms, P2S and FL, are extremely fast because they construct the path point-by-point in a greedy manner. However, the performance of the two algorithms, as presented above, is not very satisfactory, especially for the cases with long sampling intervals. The proposed AntMapper applies the ACO algorithm for optimal path finding, whose time complexity is $O(g \cdot u \cdot n)$. AntMapper can be considered as possessing a sublinear complexity to the problem size. Actually, the operation of ACO is very lightweight, while most computational overhead of AntMapper is derived from the distance calculations in the data preparation and the information modeling modules. This is why, despite running 100 iterations, AntMapper does not incur much more execution time than the P2S and FL. Considering the HMM algorithm, it uses a Viterbi algorithm (dynamic programming) to find the optimal path. Besides, it is to be noticed that, although Viterbi is a deterministic approach that can provide the optimal path, due to the simplifications made on the model, the HMM algorithm can still obtain suboptimal results [14]. The time complexity of the HMM-Viterbi is $O(n \cdot |S|^2)$, where S is the candidate state space in the hidden markov model. In [14], S is defined as the set of segments that are located in a circular area (with radius 200m) of each GPS location. As shown in Fig. 8, the algorithm is much more time-consuming than the others. Further, we replace the candidate selection of HMM by using the same nearest neighbor scheme of AntMapper and denote the algorithm

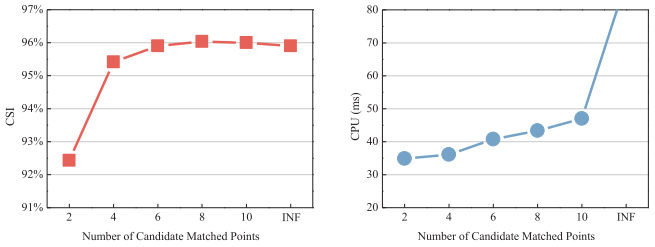


Fig. 9. Accuracy and time efficiency of AntMapper with respect to the number of candidates being considered.

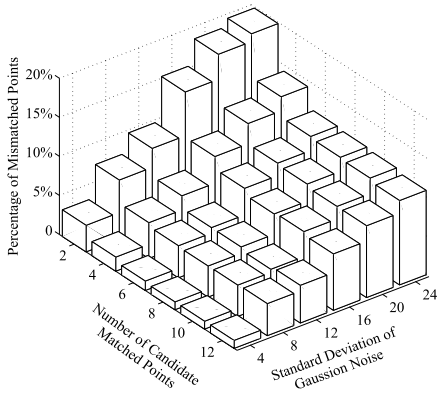


Fig. 10. Percentage of mismatched points for trajectories with different ranges of noises.

as HMMc. The results are repoted in Table III, where the results of HMM and AntMapper are also listed for the purpose of comparisons. It can be observed that HMMc are much efficient than the original HMM algorithm, whose execution time is similar to that of AntMapper. Besides, considering the accuracy measure, the results of HMMc and HMM are similar: on three datasets (10s/30s/2min), the results of HMM are slightly better than HMMc, whereas on the other two datasets (1min/5min), the HMMc are slightly better. The proposed AntMapper outperforms both HMMc and HMM with respect to all sampling intervals.

Apart from the algorithmic complexity, the execution time of map matching algorithms is also influenced by the data instances. For example, the map matching task for GPS points in a dense area in the road network is more difficult than those located in an area where the road segments are sparse. As we test a number of widely spread trajectories and then record the average, the influence of time coming from the data instances is very small. However, we may sometimes observe the influence, e.g., the small reduction for all P2S, FL, and AntMapper on the 30s dataset.

3) Investigation on the Number of Candidates: In the map matching model, the candidate matched points for each GPS coordinates is determined as the m-nearest neighbors of the GPS coordinates. The selection of m has effect on both the final map matching results and the computational costs of calculating the local heuristic between consecutive points. Here we execute AntMapper with $m = 2, 4, 6, 8, 10$ and a much larger value denoted as *INF*, so as to investigate the influence of this parameter. Taking $\Delta t = 1$min for instance, as reported in Fig. 9, the map matching accuracy of AntMapper increases progressively with m, but, meanwhile, the required execution time of the algorithm also increases. More specifically, it can be seen from Fig. 9 that, when using $m \geq 6$, the improvement on accuracy by further increasing m is slight. In fact, when using the *INF*, we can observe a small reduction of the accuracy value, because increasing m enlarges the search space of ACO, prolongs the required time for ant colony to approach the global optimum, and hence reduces the obtained solution accuracy. Therefore, we suggest using $m = 6$ in the experiments since this value can bring satisfactory results within short execution time. Nevertheless, for situations with high levels of GPS error, we can increase the value of m according to the practical requirements.

TABLE IV
CSI OF ANTMAPPER ON DEGRADED DATA INSTANCES ($m = 6$)

Std[1]	4	8	12	16	20	24
CSI	98.80%	95.90%	95.20%	91.80%	90.20%	88.40%

[1] Std denotes the standard deviation of noise.

4) Investigation on Degraded Data: In the above experiments, Gaussian noise with standard deviation 8m (i.e., error of 0 to 24m) is added to the data sequences, because it covers the amplitude of the localization errors for the majority of current GPS devices. Testing different noise ranges is also interesting, since it provides evidences of how the proposed algorithm will perform if the data are less accurate due to some severe signal interferences. Fig. 10 presents the test results by adding different magnitudes of noises and using different number of candidates for matching. As we can imagine, the map matching accuracy decreases with the increase of noise range. From Fig. 10, it can also be observed that AntMapper is not very sensitive to the setting of candidate number and using $m = 6$ contributes good results for different noise ranges. More specifically, the numerical results of instances with different ranges of noises when $m = 6$ are listed in Table IV. Compared with the CSI values listed in Table II, it can be noticed that, the performance of AntMapper on degraded data instances is competitive to the performance of other algorithms on normal instances. These results further verify the robustness of the proposed algorithm.

D. Results on Real-Life Instances

As described in Subsection V-A, the underlying map we used for matching is encrypted by shifting errors, which makes the map matching tasks become more challenging. The experimental results of P2S, FL, HMM, and AntMapper for the real-life data instances are reported in Table V. In addition, the box plots of more detailed statistical results are shown in Fig. 7. It can be observed that the best performed algorithm is still AntMapper, which achieves $CSI = 95.20\%$.

Besides, the FL algorithm maintains its performance as on the synthetic instances. The results of P2S decreases to 36.11%, which means that, due to shifting errors, the nearest point-to-segment projection is not the correct matching result in most cases. Note that the performance of HMM also decreases to a great extent. This may be because the algorithm makes two assumptions, which are not satisfied in the real-life

TABLE V
COMPARISON OF CSI ON REAL-LIFE INSTANCES

Algorithm	P2S	FL	HMM	AntMapper
CSI	36.11%	86.87%	70.71%	95.20%

instances tested in this paper: (1) The algorithm ignores any road segments that are located more than 200 meters away from the GPS coordinates. However, due to shifting errors on the map, some ground-truth segments may be neglected by the algorithm. (2) The algorithm assumes that the routes cannot be circuitous or strange, but we find that, by investigating the taxi trajectories, this kind of routes exists for the vacant taxis wandering around the road to seek for passengers.

AntMapper is more robust than the other algorithms owing to the following features: (1) The algorithm not only considers local geometric and topological information, but it also adopts a global similarity measure that quantifies the correlation between patterns of raw trajectory and digital path. (2) Ranking is adopted to make the error-based heuristic be nonparametric, and hence the algorithm no longer relies on the error distribution. (3) We apply the ant colony optimization (ACO), which is capable of searching the global optimum and thus providing better results than local search-based methods.

VI. CONCLUSIONS

This paper develops a novel map matching algorithm named AntMapper. In the proposed information model, the geometric, heading, and routing errors in matching the GPS coordinates constitute the local heuristic. We adopt a rank function to make the calculated heuristic be nonparametric, so that the algorithm is more robust in different scenarios. In addition, the model also adopts a global similarity measure, namely, the minimum ratio that quantifies the correlation between the patterns of the raw trajectory and the matched path. In this way, both the local geometric and topological information and the global shape information are taken into account. Then, in the solution finding module, the ant colony optimization algorithm is used, in which a group of ants are dispatched to search the optimal digital path that matches the GPS trajectory the most. AntMapper is tested on both synthetic and real trajectory instances, with performance compared with P2S, FL, and HMM algorithms. Experimental results show that AntMapper outperforms the others to a large extent.

Note that the algorithm developed in this paper handles the trajectories in batches, which is suitable for offline usage of trajectories such as traffic dynamics analysis, interests prediction, and urban planning. It would also be interesting to develop online map matching algorithm that handles streaming GPS coordinates, for which a typical application is the navigation system. Developing an online AntMapper is a promising future working direction.

REFERENCES

[1] W. Luo, H. Tan, L. Chen, and L. M. Ni, "Finding time period-based most frequent path in big trajectory data," in *Proc. ACM Int. Conf. Manage. Data (SIGMOD)*, 2013, pp. 713–724.

[2] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proc. 20th ACM Int. Conf. Knowl. Discovery Data Mining (SIGKDD)*, 2014, pp. 25–34.

[3] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen, "Learning the dynamics of arterial traffic from probe data using a dynamic Bayesian network," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1679–1693, Dec. 2012.

[4] S. Liu, L. M. Ni, and R. Krishnan, "Fraud detection from taxis' driving behaviors," *IEEE Trans. Veh. Technol.*, vol. 63, no. 1, pp. 464–472, Jan. 2014.

[5] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," in *Proc. 13th ACM Int. Conf. Ubiquitous Comput. (UbiComp)*, 2011, pp. 89–98.

[6] M. Hashemi and H. A. Karimi, "A critical review of real-time map-matching algorithms: Current issues and future directions," *Comput. Environ. Urban Syst.*, vol. 48, pp. 153–165, Sep. 2014.

[7] C. E. White, D. Bernstein, and A. L. Kornhauser, "Some map matching algorithms for personal navigation assistants," *Transp. Res. C, Emerg. Technol.*, vol. 8, no. 1, pp. 91–108, 2000.

[8] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk, "On map-matching vehicle tracking data," in *Proc. 31st Int. Conf. Very Large Data Bases (VLDB)*, 2005, pp. 853–864.

[9] N. R. Velaga, M. A. Quddus, and A. L. Bristow, "Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems," *Transp. Res. C, Emerg. Technol.*, vol. 17, no. 6, pp. 672–683, 2009.

[10] H. Wei, Y. Wang, G. Forman, Y. Zhu, and H. Guan, "Fast Viterbi map matching with tunable weight functions," in *Proc. 20th ACM Int. Conf. Adv. Geograph. Inf. Syst. (SIGSPACIAL)*, 2012, pp. 613–616.

[11] M. Quddus and S. Washington, "Shortest path and vehicle trajectory aided map-matching for low frequency GPS data," *Transp. Res. C, Emerg. Technol.*, vol. 55, pp. 328–339, Apr. 2015.

[12] M. Hashemi and H. A. Karimi, "A weight-based map-matching algorithm for vehicle navigation in complex urban networks," *J. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 573–590, Mar. 2016.

[13] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, "Map-matching for low-sampling-rate GPS trajectories," in *Proc. 17th ACM Int. Conf. Adv. Geograph. Inf. Syst. (SIGSPATIAL)*, 2009, pp. 352–361.

[14] P. Newson and J. Krumm, "Hidden Markov map matching through noise and sparseness," in *Proc. 17th ACM Int. Conf. Adv. Geograph. Inf. Syst. (SIGSPATIAL)*, 2009, pp. 336–343.

[15] G. Wang and R. Zimmermann, "Eddy: An error-bounded delay-bounded real-time map matching algorithm using HMM and online Viterbi decoder," in *Proc. 22nd ACM Int. Conf. Adv. Geograph. Inf. Syst. (SIGSPATIAL)*, 2014, pp. 33–42.

[16] C. Fouque and P. Bonnifait, "Matching raw GPS measurements on a navigable map without computing a global position," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 887–898, Feb. 2012.

[17] T. Hunter, P. Abbeel, and A. Bayen, "The path inference filter: Model-based low-latency map matching of probe vehicle data," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 507–529, Apr. 2014.

[18] F. Abdallah, G. Nassreddine, and T. Denoeux, "A multiple-hypothesis map-matching method suitable for weighted and box-shaped state estimation for localization," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1495–1510, Apr. 2011.

[19] M. A. Quddus, R. B. Noland, and W. Y. Ochieng, "A high accuracy fuzzy logic based map matching algorithm for road transport," *J. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 103–115, Apr. 2006.

[20] S. Syed and M. Cannon, "Fuzzy logic-based map matching algorithm for vehicle navigation system in urban canyons," in *Proc. ION Nat. Tech. Meeting*, vol. 1. San Diego, CA, USA, 2004, pp. 26–28.

[21] M. Ren and H. A. Karimi, "A fuzzy logic map matching for wheelchair navigation," *GPS Solutions*, vol. 16, no. 3, pp. 273–282, 2012.

[22] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.

[23] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behaviour," *Nature*, vol. 406, no. 6791, pp. 39–42, 2000.

[24] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transp. Res. C, Emerg. Technol.*, vol. 15, no. 5, pp. 312–328, 2007.

[25] G. Taylor, G. Blewitt, D. Steup, S. Corbett, and A. Car, "Road reduction filtering for GPS-GIS navigation," *Trans. GIS*, vol. 5, no. 3, pp. 193–207, 2001.

[26] H. Wu, W. Sun, and B. Zheng, "Is only one GPS position sufficient to locate you to the road network accurately?" in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. (UbiComp)*, 2016, pp. 740–751.
[27] R. R. Joshi, "A new approach to map matching for in-vehicle navigation systems: The rotational variation metric," in *Proc. IEEE Intell. Transp. Syst.*, Aug. 2001, pp. 33–38.
[28] M. Xu, Y. Du, J. Wu, and Y. Zhou, "Map matching based on conditional random fields and route preference mining for uncertain trajectories," *Math. Problems Eng.*, vol. 2015, pp. 1–13, Apr. 2015.
[29] M. Rohani, D. Gingras, and D. Gruyer, "A novel approach for improved vehicular positioning using cooperative map matching and dynamic base station DGPS concept," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 230–239, Jan. 2016.
[30] M. Rohani, D. Gingras, and D. Gruyer, "A new decentralized Bayesian approach for cooperative vehicle localization based on fusion of GPS and inter-vehicle distance measurements," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 2, pp. 85–95, Feb. 2015.
[31] A. A. Goshtasby, "Similarity and dissimilarity measures," in *Image Registration*. London, U.K.: Springer, 2012, pp. 7–66.
[32] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. Orlando, FL, USA: Academic, 2009.
[33] B. Yu and Z. Z. Yang, "An ant colony optimization model: The period vehicle routing problem with time windows," *Transp. Res. E, Logistics Transp. Rev.*, vol. 47, no. 2, pp. 166–181, 2011.
[34] G. Kim, Y. S. Ong, C. K. Heng, P. S. Tan, and N. A. Zhang, "City vehicle routing problem (city VRP): A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1654–1666, Aug. 2015.
[35] J. Cheng, J. Cheng, M. Zhou, F. Liu, S. Gao, and C. Liu, "Routing in Internet of vehicles: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2339–2352, Oct. 2015.
[36] W. Fang, S. Yang, and X. Yao, "A survey on problem models and solution approaches to rescheduling in railway networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 2997–3016, Jun. 2015.
[37] Y.-J. Zheng, M.-X. Zhang, H.-F. Ling, and S.-Y. Chen, "Emergency railway transportation planning using a hyper-heuristic approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 321–329, Jan. 2015.
[38] S. Su, T. Tang, and C. Roberts, "A cooperative train control model for energy saving," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 622–631, Feb. 2015.
[39] J. B. Wang, M. Chen, X. Wan, and C. Wei, "Ant-colony-optimization-based scheduling algorithm for uplink CDMA nonreal-time data," *IEEE Trans. Veh. Technol.*, vol. 58, no. 1, pp. 231–241, Jan. 2009.
[40] W.-N. Chen and J. Zhang, "Ant colony optimization for software project scheduling and staffing with an event-based scheduler," *IEEE Trans. Softw. Eng.*, vol. 39, no. 1, pp. 1–17, Jan. 2013.
[41] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin, Germany: Springer-Verlag, 1996.
[42] A. Lipowski and D. Lipowska, "Roulette-wheel selection via stochastic acceptance," *Phys. A, Statist. Mech. Appl.*, vol. 391, no. 6, pp. 2193–2196, 2012.
[43] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA, USA: MIT Press, 2004.
[44] A. Pasternack, "If you're a foreigner using GPS in China, you could a spy," Vice Media, New York, NY, USA, 2013.
[45] *Restrictions on Geographic Data in China Wikipedia*, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Restrictions_on_geographic_data_in_China

Yue-Jiao Gong (M'15) received the B.S. and Ph.D. degrees in computer science from Sun Yat-sen University, China, in 2010 and 2014, respectively. From 2015 to 2016, she was a Post-Doctoral Research Fellow with the Department of Computer and Information Science, University of Macau, Macau. She is currently an Associate Professor with the School of Computer Science and Engineering, South China University of Technology, China. Her research interests include evolutionary computation, swarm intelligence, and their applications to big data and intelligent transportation scheduling. She has authored over 50 papers, including 15 IEEE Trans. papers, in her research area. She currently serves as a Reviewer of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON CYBERNETICS, and IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. She is a member of the ACM.

En Chen received the master's degree from Fok Ying Tung Research Institute (Innovative Technologies Leadership Program), The Hong Kong University of Science and Technology, in 2015. She is currently a Project Manager with Deepera Co., Ltd., Shenzhen, China. Her research interests include big traffic data analysis and trajectory-based applications.

Xinglin Zhang (M'15) received the B.E. degree from the School of Software, Sun Yat-sen University, in 2010 and the Ph.D. degree from the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology in 2014. He is currently an Associate Research Fellow with the School of Computer Science and Engineering, South China University of Technology, China. His research interests include wireless ad-hoc/sensor networks, mobile computing and crowdsourcing. He is a member of the ACM.

Lionel M. Ni (F'94) received the Ph.D. degree in electrical and computer engineering from Purdue University in 1980. He was a Chair Professor of Computer Science and Engineering with The Hong Kong University of Science and Technology. He is currently a Chair Professor of the Department of Computer and Information Science and the Vice-Rector of Academic Affairs with the University of Macau. He has chaired over 30 professional conferences and has received eight awards for authoring outstanding papers. He is a fellow of the Hong Kong Academy of Engineering Science.

Jun Zhang (F'16) received the Ph.D. degree from City University of Hong Kong, Hong Kong, in 2002. He is currently a Changjiang Chair Professor with the School of Computer Science and Engineering, South China University of Technology, China. His research interests include computational intelligence, cloud computing, data mining, and power electronic circuits. He has authored over 200 technical papers in his research area. He was a recipient of the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011 and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and IEEE TRANSACTIONS ON CYBERNETICS.