

Research Article

Dynamic Fuzzy Logic-Ant Colony System-Based Route Selection System

Hojjat Salehinejad^{1,2} and Siamak Talebi^{1,2,3}

¹Department of Electrical Engineering, Shahid Bahonar University of Kerman, Kerman 76169-133, Iran

²International Center for Science, High Technology, and Environmental Sciences, Kerman, Iran

³Advanced Communications Research Institute, Sharif University of Technology, Tehran, Iran

Correspondence should be addressed to Hojjat Salehinejad, h.salehi@mail.uk.ac.ir

Received 8 June 2010; Accepted 23 August 2010

Academic Editor: Hani Hagrass

Copyright © 2010 H. Salehinejad and S. Talebi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Route selection in metropolises based on specific desires is a major problem for city travelers as well as a challenging demand of car navigation systems. This paper introduces a multiparameter route selection system which employs fuzzy logic (FL) for local pheromone updating of an ant colony system (ACS) in detection of optimum multiparameter direction between two desired points, origin and destination (O/D). The importance rates of parameters such as path length and traffic are adjustable by the user. In this system, online traffic data are supplied directly by a traffic control center (TCC) and further minutes traffic data are predicted by employing artificial neural networks (ANNs). The proposed system is simulated on a region of London, United Kingdom, and the results are evaluated.

1. Introduction

In route selection problems, typically a pair of origin and destination (O/D) is given, while there are many possible routes for selection. The objective is to find a route with the least cost, based on the costs calculated for different possible directions.

During the past years, many researchers have proposed methods for optimum route selection by considering some important parameters for city travelers. Some recent navigation systems have embedded algorithms that attempt to minimize journey distance and/or travel time. However, many drivers are now becoming increasingly concerned with rising fuel costs, waste of time in traffic congestions, and pollutant emissions.

Finding the shortest route between a pair of points is an NP-hard problem that requires enumerating all the possible routes. In addition, most users nowadays not only need routes with the shortest distance, but also require routes which can satisfy their other desires. Such users mostly need safe, low-traffic, and scenic routes with the fewest number of junctions to avoid traffic lights. The published research

papers in the literature have not addressed a fast, dynamic, low-complicated, and practical system that can be available almost anywhere and satisfy all these desires by comprising all important parameters.

The proposed fuzzy logic-ant colony system (FLACS) in this paper introduces a new dynamic multiparameter vehicle navigation system that satisfies the above claims. This system uses a combination of fuzzy logic (FL) and ant colony system (ACS) algorithm in order to find an optimum multiparameter route between a pair of O/D. An optimum route refers to a route that attempts to satisfy all desired parameters of a user. These parameters are “Distance,” “Traffic,” and “Incident Risk.” The set of parameters can also be extended by adding “Width” (number of lanes), “Quality” (medical treatment facilities, entertainments, and etc.), and number of “Traffic Lights” parameters. In this system, current traffic data are supplied by a traffic control center (TCC) and artificial neural networks (ANNs) are employed for traffic data estimation of coming minutes. As user might not like the first selected route, the proposed system is capable of considering previously selected routes by user and providing a ranking set of feasible routes.

The remainder of this paper is organized as follows. The next section reviews some route selection-related works. A brief review on basic principles of ACS as well as FL and ANN is presented in Section 3. Section 4 demonstrates details of the proposed system and simulation results are discussed in Section 5. Finally, the paper is concluded in Section 6.

2. Related Route Selection Works

Barth et al. in [1] have developed some environmentally friendly navigation techniques that focus on minimizing energy consumption and pollutant emissions. These methods combine sophisticated mobile source energy and emission models with route minimization algorithms that are used for navigational purposes. These methods have been applied on several case studies in southern California, USA. Authors of [2] have emphasized on more important parameters such as number of traffic lights, right turns, and stop signs for route selection. Although route planning has been widely studied, most of the available applications are primarily targeted at finding the shortest travel time or the shortest path routes, which is insufficient for dynamic route planning in real life scenario [3]. The approach in [4] provides an optimal route planning by making efficient use of underlying geometrical structure. It combines classical artificial intelligence exploration with computational geometry. Given a set of global positioning system (GPS) trajectories, the input is refined by geometric filtering and rounding algorithms. This method has some computational complexities as well as being depended on the GPS to maintain optimal route planning.

In [5], an approach combining offline precomputation of optimal candidate paths with online path retrieval and dynamic adaptation is proposed. Based on a static traffic data file, a partially disjoint candidate path set is constructed prior to the trip using a heuristic link weight increment method. This method satisfies reasonable path constraints that meet the driver preferences as well as alternative path constraints that limit the joint failure probability for candidate paths. This algorithm is tested on randomly generated road networks [5]. Routing vehicles based on real-time traffic conditions has presented significant reduction of travel time and, hence, cost in high volume traffic situations. The authors in [6] model the dynamic route determination problem as a Markov Decision Process (MDP) and present procedures for identifying traffic data that have no decision making value. These methods are examined based on actual data collected on a route network in southeast Michigan, USA [6]. In [7], historical data and experiences are used to decide an optimum route based on the analysis of unblocked reliability and the circuitous length. This can provide reasonable route guidance to the traveler, even without real-time traffic information. In [8], a novel concept of an intelligent navigator that can give a driver timely advice on safe and efficient driving is presented. From both the current traffic conditions obtained from visual data and the driver goals and preferences in driving, it autonomously generates advice and presents it to the driver. Two main components of this

intelligent navigator are the advice generation system and the road scene recognition system.

The FL system is a popular and powerful tool implemented by researchers for optimum route selection [9–11]. Teodorovic and Kikuchi have applied FL methodology for the first time in route selection [11]. Their proposed method only considers the travel time parameter and cannot be easily generalized to multiple routes [9, 10]. The proposed hybrid evolutionary algorithm for solving the dynamic route planning problem (HEADRPP) in [3] comprises a fuzzy logic implementation (FLI) and a graph partitioning algorithm (GPA) incorporated into a genetic algorithm (GA) core, and offers both optimized shortest path and shortest time routes to the user [3]. Kambayashi et al. have also employed a GA to find a quasioptimal route for the driver. They attempted to integrate uncomfortable turns into the conditions of their proposed GA based route selection algorithm [12].

By considering the route selection problem as a multi-criteria problem, an approach of ant colony optimization (ACO) is presented in [13] to solve a spatial reconfiguration problem of multimodal transport network. This work is consisted of real-time route planning with three criteria to be optimized: travel time, travel distance, and number of vehicles. However in this work, as an application of reconfiguration, the execution time is very important. Another ants algorithm-based approach is presented in [14], which is capable of processing a digital image and detecting tracks left by preceding vehicles on ice and snow. Salehinejad et al. have proposed a series of route selection systems based on ACS in [15–17]. In [17], a combination of A-Star (A*) algorithm and ACS is employed where the A* algorithm invigorates some paths pheromones in ants algorithm. In this work, A* algorithm is a prologue of the ants algorithm. It invigorates some produced directions by itself in order to help ants algorithm to recognize best direction with higher reliability and lower cost than pure ants algorithm. This is done by updating (increasing) pheromone amount of directions found in the A* algorithm. In [15, 16], the ACS employs ANNs to predict traffic data of further minutes in offline and online modes, respectively. Most data prediction techniques rely on accuracy of a plant model or knowledge of the stochastic processes [18–20]. This is while the ANNs have suggested an alternative approach in the literature. Some benefits of ANNs are relatively insensitive to erroneous or missing data and handling nonlinear systems, which is an important issue for treating highly dynamic traffic data [21, 22].

3. A Survey on Ant Colony System, Fuzzy Logic, and Artificial Neural Networks

3.1. Ant Colony System. The ACO is a class of algorithms whose first member called ant system (AS) was initially proposed by Dorigo et al. [23]. Although real ants are blind, they are capable of finding shortest path from food source to their nest by exploiting a liquid substance, called pheromone, which they release on the transit route. The developed AS strategy attempts to simulate behavior of real

ants with the addition of several artificial characteristics: visibility, memory, and discrete time to resolve many complex problems successfully such as the traveling salesman problem (TSP) [24], vehicle routing problem (VRP) [25], and best path planning [26]. Even though many changes have been applied to the ACO algorithms during the past years, their fundamental ant behavioral mechanism that is positive feedback process demonstrated by a colony of ants is still the same. Ants algorithm has also plenty of networking applications such as in communication networks [27] and electrical distribution networks [28]. Different steps of a simple ant colony system algorithm are as follows.

Problem Graph Depiction. Artificial ants move between discrete states in discrete environments. Since the problems solved by ACS algorithm are often discrete, they can be represented by a graph with N nodes and R routes.

Ants Distribution Initializing. A number of ants are placed on the origin nodes. The number of ants is often defined based on trial and error and number of nodes in the region.

Ants Probability Distribution Rule. Ants probabilistic transition between nodes can also be specified as node transition rule. The transition probability of ant k from node i to node j is given by

$$p_{ij}^k = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{h \notin \text{tabu}_k} (\tau_{ih})^\alpha (\eta_{ih})^\beta} & j \notin \text{tabu}_k, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where τ_{ij} and η_{ij} are the pheromone intensity and the cost of route between nodes i and j , respectively. Relative importance of τ_{ij} and η_{ij} are controlled by parameters α and β , respectively. The tabu_k is set of unavailable routes (visited nodes) for ant k .

Update Global Trail. When every ant has assembled a solution, at the end of each cycle, the intensity of pheromone is updated by a pheromone trail updating rule. This rule for ACS algorithm is given as

$$\tau_{ij}^{\text{new}} = (1 - \rho)\tau_{ij}^{\text{old}} + \sum_{k=1}^m \Delta\tau_{ij}^k, \quad (2)$$

where $0 < \rho < 1$ is a constant parameter named pheromone evaporation and m is number of ants. The amount of pheromone laid on the route between nodes i and j by ant k is

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{f_k} & \text{if route } (i, j) \text{ is traversed by,} \\ & \text{the } k\text{th ant (at the current cycle),} \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where Q is a constant parameter and f_k is the cost value of the found solution by ant k .

Stopping Procedure. This procedure is completed by arriving to a predefined number of cycles, or the maximum number of cycles between two improvements of the global best solutions.

3.2. Fuzzy Logic. The fuzzy set theory was initially introduced by Lotfi Zadeh in 1965. In this theory, the usual set theory is generalized so that an object cannot only be seen as an element of a set (membership value 1) or not an element of this set (membership value 0), but it can also have a membership value between 0 and 1. Therefore a fuzzy set is defined by its membership functions which are allowed to assume any value in the interval [0,1] instead of their characteristic function, which is defined to assume the values 0 or 1 only [29].

Later in 1970, Assilian and Mamdani developed the fuzzy control concept to control complex processes particularly when no strict model of the processes exists [30, 31]. Fuzzy control can be described as a means of control working with conditional sentences called linguistic “IF-THEN” rules rather than mathematical equations. The deduction of the rule is called inference and requires definition of a membership function characterizing this inference. This function determines the degree of truth of each proposition [31].

Different stages of a simple fuzzy logic controlling system are as follows.

Fuzzification. Each fuzzy system is realized in the form of fuzzy rules such as

$$\begin{aligned} \text{Rule I:} & \quad \text{IF } M \text{ is } p_1 \ \& \ N \text{ is } q_1 \ \text{THEN } R \text{ is } g_1, \\ \text{Rule II:} & \quad \text{IF } M \text{ is } p_2 \ \& \ N \text{ is } q_2 \ \text{THEN } R \text{ is } g_2, \end{aligned} \quad (4)$$

where M and N are variables of the condition part, R is variable of the action part, and p_i , q_i , and g_i are fuzzy parameters characterized by membership functions. The condition parts of control rules make use of measurements which are usually real numbers [30]. Figure 1 presents Mamdani’s approach to the fuzzy inference procedure for two rules and arbitrarily membership functions. By considering D_1 and D_2 as value domains, the real valued measurements m and n are matched to their corresponding fuzzy variables by determining their membership values as $\mu_{p_1}(m)$, $\mu_{q_1}(n)$, $\mu_{p_2}(m)$, and $\mu_{q_2}(n)$.

Fuzzy Inference Engine. By considering $M = m$ and $N = n$ for all the control rules in the rule base, the truth value of each rule in the premise is derived by building the conjunctions of the matching membership values as

$$\begin{aligned} \text{Rule I:} & \quad \mu_1 = \mu_{p_1}(m) \wedge \mu_{q_1}(n), \\ \text{Rule II:} & \quad \mu_2 = \mu_{p_2}(m) \wedge \mu_{q_2}(n), \end{aligned} \quad (5)$$

where the “ \wedge ” conjunction represents one of the Mamdani’s implication such as “min” function [30].

The truth degree of rules I and II is represented by μ_1 and μ_2 , respectively. These also define the membership values $\mu_{g_1}(R)$ and $\mu_{g_2}(R)$ of the fuzzy subsets $g_1(R)$ and $g_2(R)$ for

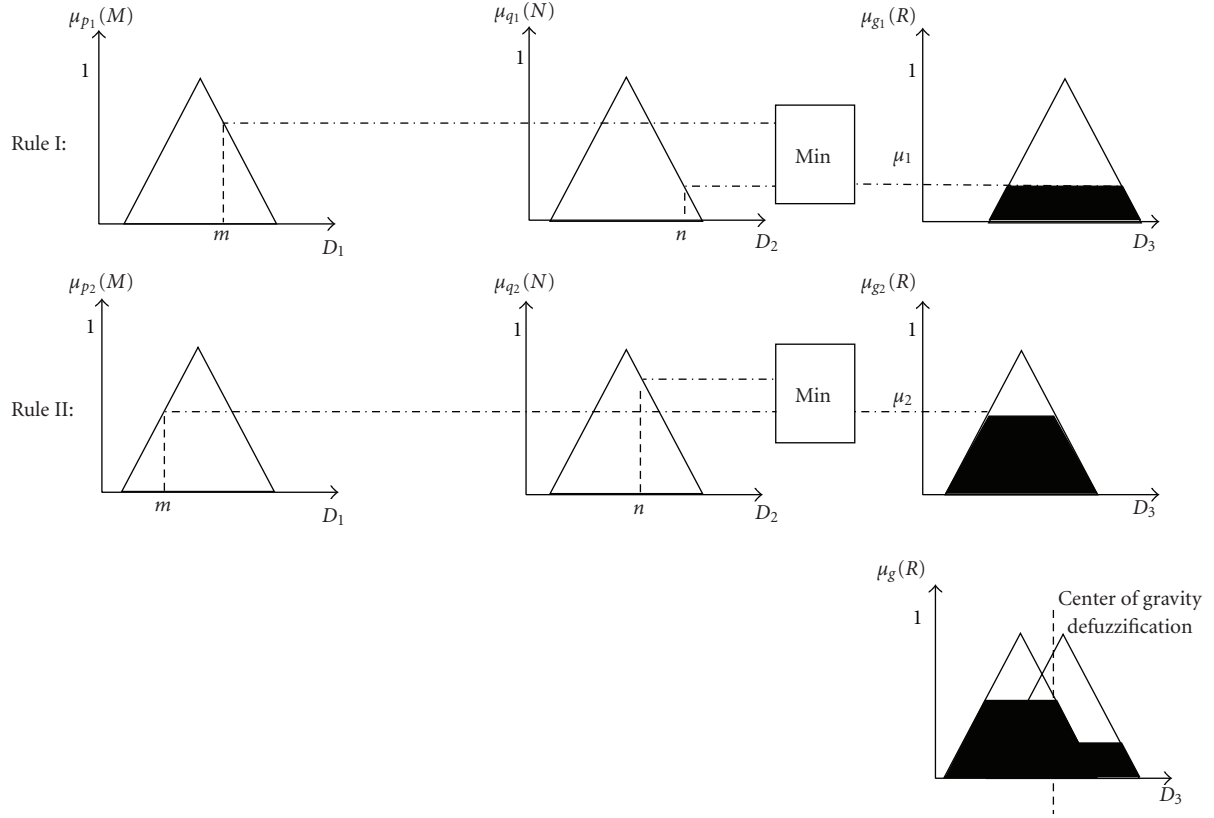


FIGURE 1: Fuzzy inference procedure for two rules based on Mamdani's type.

the measurement m and n , respectively. By considering D_3 as the value domain of the output variable, the fuzzy control output $g(R)$ is represented by the aggregation of all fuzzy subsets $g_i(R)$. Its membership values $\mu_{g_i}(R)$ are determined by disjunction of all the membership values $\mu_{g_i}(R)$ as

$$\mu_g(R) = \mu_{c_1}(R) * \mu_{c_2}(R), \quad (6)$$

where the “*” disjunction is the “max” function when used with Mamdani's implication [30, 31].

Defuzzification. The fuzzy result, which is outcome of the inferences, is transformed into a real value that can be used as control input. Since the desired output is a nonfuzzy outcome, a quantitative value of the control output is determined by defuzzifying $\mu_{g_i}(R)$. There are two common methods for defuzzification which are the “center of gravity” and “mean of maxima” methods. Interested readers are referred to [30] for more descriptions.

3.3. Artificial Neural Networks. The brain is an extensive structure consisted of many connected neural cells called “neurons.” In ANNs, it is claimed to imitate biological brain neural networks into mathematical model. A model of the brain connects many linear or nonlinear neuron models and processes information in a parallel distributed manner. Since neural networks have learning and self-organization

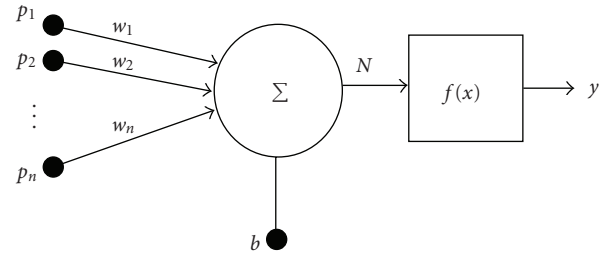


FIGURE 2: An artificial neuron model.

capabilities, they can adapt to changes in data and learning the characteristics of input signal. Such networks can learn a mapping between an input and an output space and synthesize an associative memory that retrieves the appropriate output when presented with the input and generalizes when presented with new inputs [32]. By considering the above characteristics, neural networks are employed today in many fields including pattern recognition, identification, speech, vision, and control systems [33].

A neuron with a single n -element input vector is shown in Figure 2 where p_1, p_2, \dots, p_n are individual element inputs and w_1, w_2, \dots, w_n are weights of connections. The ANNs can be trained to perform a particular function by adjusting the values of the weights [33]. The neuron unit has a bias b , which is summed with the weighted inputs to

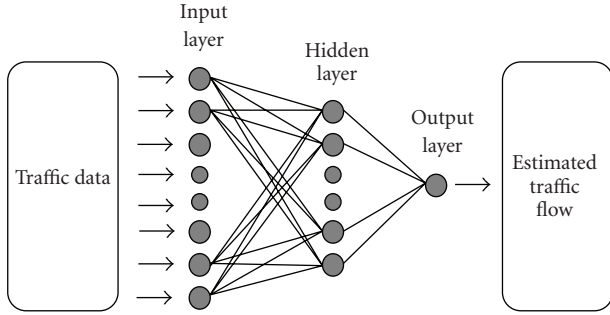


FIGURE 3: Employed hierarchical feed-forward ANN structure for traffic estimation.

form the net input N . Output of the neuron is the weighted sum of input signals as

$$y = f\left(\sum_{i=1}^n p_i \cdot w_i + b\right). \quad (7)$$

The neuron activation function is often a continuous and nonlinear function which is called sigmoid function and is defined as

$$f(x) = \frac{1}{1 + e^{-ax}}, \quad (8)$$

where a is a constant and $0 < f(x) < 1$.

One largely used category of the ANNs is called feed-forward net. This hierarchical structure is consisted of some layers without interconnections between neurons in each layer, and signals flow from input layer to output layer in one direction as in Figure 3. Readers are encouraged to refer to [32, 33] for more details.

4. Proposed Route Selection System

4.1. System Structure. The proposed system is executed locally for every single vehicle. It finds directions with minimum costs based on the importance rates of user desired parameters. Architecture of the proposed system is presented in Figure 4. In this system, the traffic signal is provided by a TCC and contains current traffic data which is updated regularly. Memory of the system comprises statistical data, average speed of the vehicle, current saved traffic data, current time, and so forth. System information like statistical data is available on smart cards for different cities. Therefore, user can use the system in other regions supported by a TCC just by purveying the appropriate smart card.

In this system, ANNs are used for traffic estimation of coming minutes. The manner depicted in [22] is followed for employing ANNs as traffic predictor in this work. The employed ANN is consisted of one hidden layer with m hidden neurons, n inputs, and one output as in Figure 3. By considering statistical traffic data of last n years in a typical time period as input, the ANN structure is trained for making traffic predictions. Then, it can estimate traffic

load of further minutes by considering current traffic data available by TCC as inputs. For a detailed description of the employed ANN, enthusiastic readers may refer to [22].

In Figure 4, time needed to move between two junctions, before movement of the vehicle is estimated in the “time delay estimation” block. This is due to having different traffic loads in different hours of day and night. Therefore, the system must have an estimate of arrival time to other junctions to use the appropriate data of the arrival time to that junction. This estimation is done by considering the vehicle average speed, the distance between junctions, and the current traffic flow. By taking into account the predicted traffic data and estimated journey time delay, correspondent predicted traffic data are used in the system.

The proposed system has the privilege of evading upcoming congestions. The system is aware of current vehicle location by using GPS. Therefore, if a congestion happens on the suggested direction, the system immediately recommends nearest direction to the user parameters and current direction based on the new conditions to evade upcoming congestion. The system also has the capability of considering the previously selected directions. This task is performed by updating pheromone amount of these directions in the ACS. In order to have a more user friendly system, it can provide the user some candidate directions to choose from. These attributes are optional for users and more features can be developed in future works. The next subsection discusses the proposed FLACS-based structure in details.

4.2. Fuzzy Logic-Ant Colony System-Based Model. The pseudocode of the proposed FLACS-based system is presented in Algorithm 1. Its different steps are described as follow.

Initialize. It is consisted of initial values of the algorithm parameters such as number of ants, evaporation coefficient, and average speed of the vehicle.

Locate Ants. Ants are located on the start point in this stage. An active ant refers to the ant, which has not arrived to the destination yet and is not blocked in a junction. Since each ant can traverse each junction once in each iteration, an ant is blocked in a junction when it has no chance of continuing its transition toward the destination and has no possible route to move backward. A blocked ant at a junction is depicted in Figure 5.

Construct Probability. In this step, the probability of each possible direct route is calculated based on its cost function for each active ant. The probability of displacing from junction i to junction j for ant k is as

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \prod_{l \in \text{parameters}} \xi_{ijl}^{-\alpha_l}}{\sum_{h \notin \text{tabu}_k} \tau_{ih} \prod_{l \in \text{parameters}} \xi_{ihl}^{-\alpha_l}} & j \notin \text{tabu}_k, \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

where τ_{ij} is the direct route pheromone intensity from junction i to j . Parameter α controls the importance of τ_{ij}

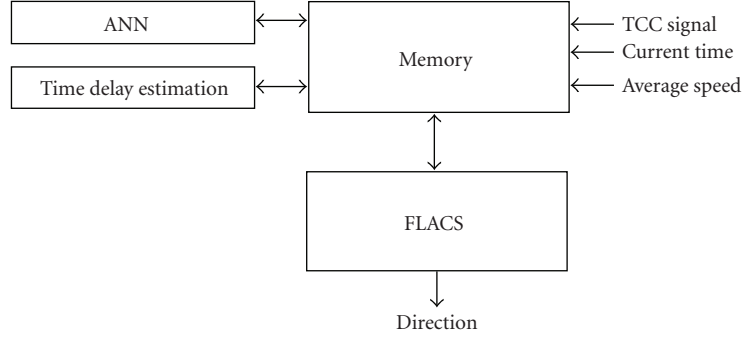


FIGURE 4: Block diagram of the proposed FLACS route selection system.

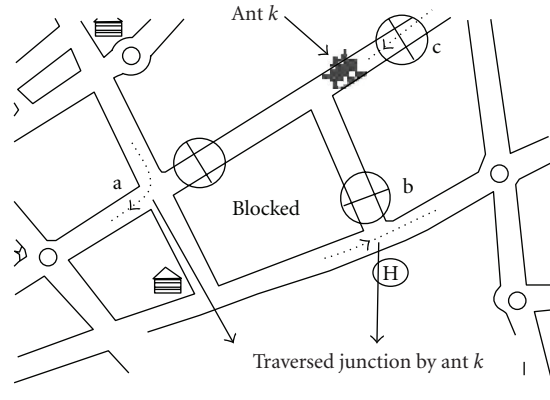
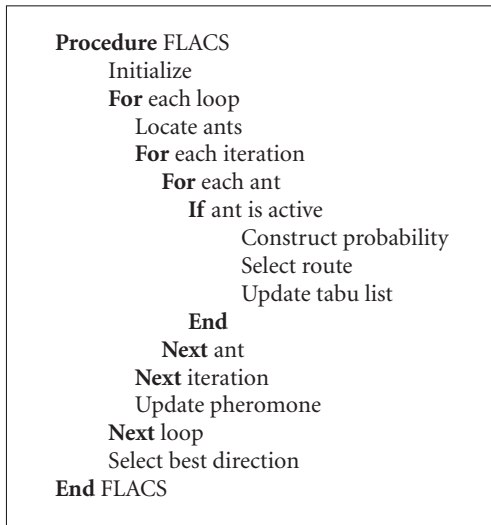


FIGURE 5: An ant on a blocked junction. The junctions “a” and “b” have been traversed by ant k before. Therefore, it cannot pass these junctions once more and is blocked in the current junction “c” in the current iteration.



ALGORITHM 1: Proposed FLACS algorithm in pseudocode.

and is set to 2 [22–24, 27]. The tabu_k list is the set of direct blocked routes (visited nodes). Parameters set is a collection of most important parameters for drivers taking journeys in metropolises. For more simplicity, the “Distance,” “Traffic

Flow,” and “Incident Risk” parameters are considered in this set. However, it can be developed by adding other parameters such as “Width” of the road (number of lanes), number of “Traffic Lights,” and “Road Quality” (medical treatment facilities, number of gas stations, and entertainments), [15–17]. Cost function of each parameter l is $\xi_{ijl}^{-\alpha_l}$, where ξ_{ijl} is normalized in $1 \leq \xi_{ijl} \leq 10$ and significance of each l is adjustable by α_l for all parameters. The considered parameters are described as follows.

(i) *Distance.*

$$\xi_{ij\text{distance}} = D(i, j, t), \quad (10)$$

where the $D(i, j, t)$ is distance between junctions i and j at time t . Since some routes are two-way or one-way in specific hours, this parameter is a function of time. Longer distance increases total cost and therefore, decreases the probability of the longer route selection in (9).

(ii) *Traffic Flow.*

$$\xi_{ij\text{traffic}} = F(i, j, t), \quad (11)$$

where $F(i, j, t)$ is traffic load of the route between junctions i and j at time t . Considering this parameter in the route selection systems has many benefits

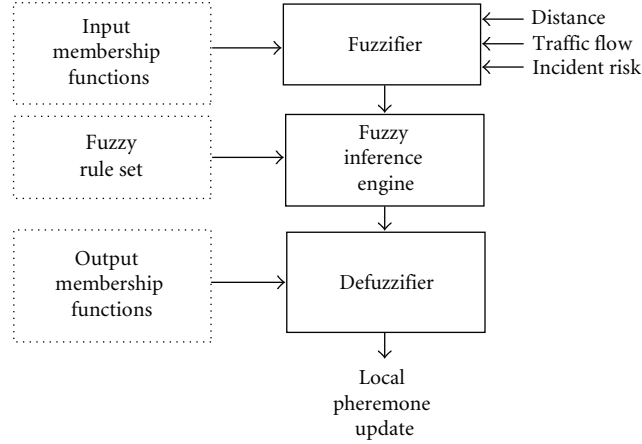


FIGURE 6: Architecture of the fuzzy logic system for FLACS.

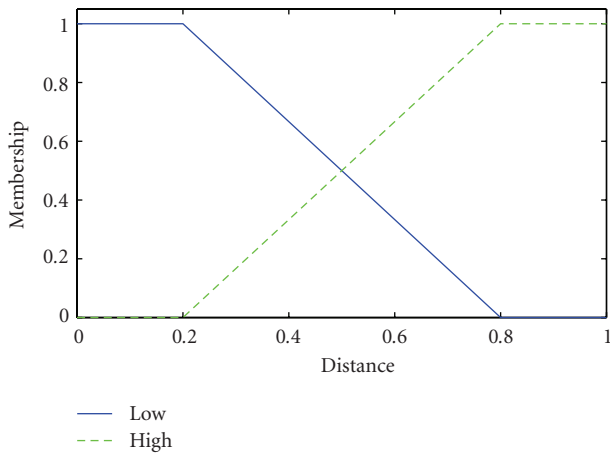


FIGURE 7: Membership function of “Distance” parameter.

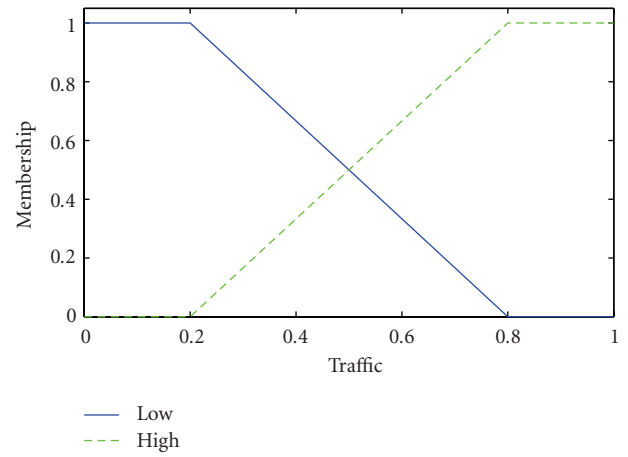


FIGURE 8: Membership function of “Traffic” parameter.

such as less air pollution, less time wastage in the traffic, and less gas usage. As traffic on a route grows, total cost increases and consequently it decreases the probability of selecting that route by the system in (9).

(iii) *Incident Risk.*

$$\xi_{ij\text{risk}} = R(i, j, t), \quad (12)$$

where $R(i, j, t)$ is risk of route between junctions i and j at time t . This parameter is a measure of incidents happening risk, which might occur on the route based on the statistical data. This parameter has direct relation with total cost, therefore risky routes have less selection probability in (9).

Select Route. A random parameter $0 \leq q \leq 1$ with uniform probability is compared with the parameter Q , where $0 \leq Q \leq 1$ and is usually fixed to 0.9 [34]. The comparison result between Q and q picks up one of the two selection methods

for the active ant to continue its route to the next junction as

$$j = \begin{cases} \arg \max(p_{ih}^k) & q > Q, \\ \text{Roulette Wheel}(p_{ih}^k) & \text{otherwise.} \end{cases} \quad (13)$$

If q is greater than Q , active ant selects the route with the highest probability, otherwise, Roulette Wheel rule is selected to choose the next junction through probabilities.

Update Tabu List. In this step, the route (selected node) which ant k has been chosen is added to the tabu list in order not to be selected again. Furthermore, its probability will not be calculated anymore.

If ant k has arrived at the destination or is blocked in a junction as in Figure 5, it is omitted from the active ant list. In other words, this step deactivates the blocked or arrived ant in the current iteration.

Update Pheromone. The ACS pheromone system is consisted of two main rules: first is applied whilst constructing

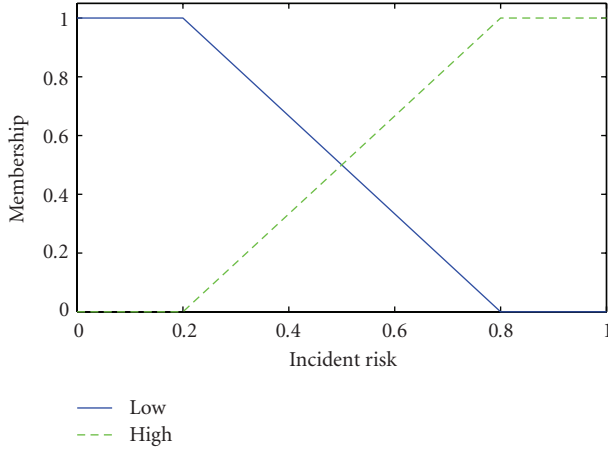


FIGURE 9: Membership function of “Incident Risk” parameter.

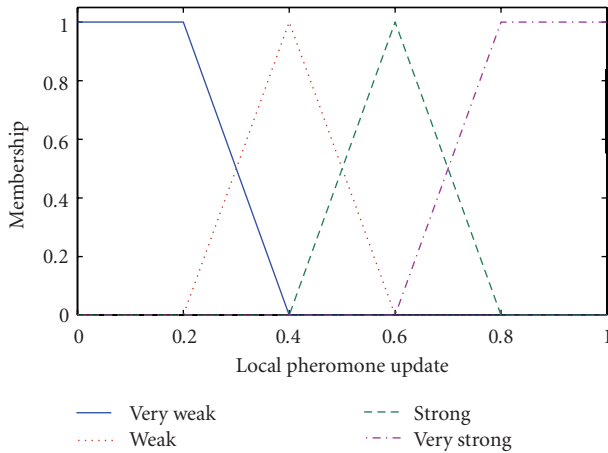


FIGURE 10: Membership function of local pheromone updating.

solutions (local pheromone update rule) and the other rule is applied after all ants have finished constructing a solution (global pheromone update rule). The pheromone amount of the route between junctions i and j is updated for ant k as

$$\tau_{ij}^{\text{new}} = \tau_{ij}^{\text{old}} + (10 \times \Delta\tau), \quad (14)$$

where $\Delta\tau$ is the amount of local pheromone updating. The value of $\Delta\tau$ is the output of a FL system. By considering the FL approach in Section 3.2, and Mamdani’s implication as our approach, structure of the employed FL system is illustrated in Figure 6. Inputs of the FL system are the total amount of “Distance,” “Traffic Flow,” and “Incident Risk” of the direction which ant k has selected. By considering computing complexities, only two input fuzzy sets, “Low” and “High,” are defined for each input. The corresponding Trapezoidal-shaped membership functions of the input variables are presented in Figures 7, 8, and 9. In these functions, it is assumed that if total distance of selected direction is more than 80% of maximum total visited distance by ants in the same loop, its membership function for “High” is unity whereas when total distance of selected direction is less than 20% of maximum total visited distance by ants in the

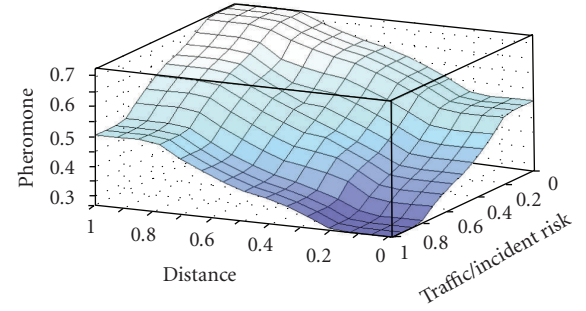


FIGURE 11: Surface-plot of the proposed FLACS system IF-THEN rules.

same loop, its membership function for “Low” is unity. The similar definitions are considered for the “Traffic Flow” and “Incident Risk” parameters. Four fuzzy sets are considered for the output variable as in Figure 10. Levels 1–4 represent different levels of pheromone density, which are “Very Weak,” “Weak,” “Strong,” and “Very Strong,” respectively. At the final stage, with respect to the most defuzzification techniques mentioned in Section 3.2, the “center of gravity” method is employed in this FL system in order to resolve a single output value from the fuzzy set.

Since important rates of parameters are different and are defined by different users, various fuzzy rules are predefined in the system. Therefore, according to the preferences of each user, appropriate fuzzy rules are loaded into the FL system. A surface-plot of the IF-THEN rules is presented in Figure 11. In this figure, rates of the considered parameters versus pheromone are presented. As an example, IF-THEN rules set of a specific parameters importance rate set is presented in Table 1. In this example, preferences of the user for the “Distance,” “Traffic,” and “Incident Risk” parameters are “High,” “Low,” and “Low,” respectively. Therefore, directions with closer costs to the preferences achieve more local pheromone update. This is while routes with the “Low,” “High,” and “High” preferences are achieving the least local pheromone update.

The traditional last step of each completed loop is global pheromone updating defined as

$$\tau_{ij}^{\text{new}} = \rho\tau_{ij}^{\text{old}}, \quad (15)$$

where $0 < \rho < 1$ is the evaporation coefficient and is usually set to 0.9 [15–17, 34].

Select Best Direction. After m loops, direction with the lowest cost from origin to destination is recommended by the system.

5. Simulation Results

Performance of the proposed FLACS route selection system versus ACS based and A* -ACS systems presented in [15–17] is evaluated and discussed in this section.

The proposed system is applied on a part of London, United Kingdom, routes network. The selected region is

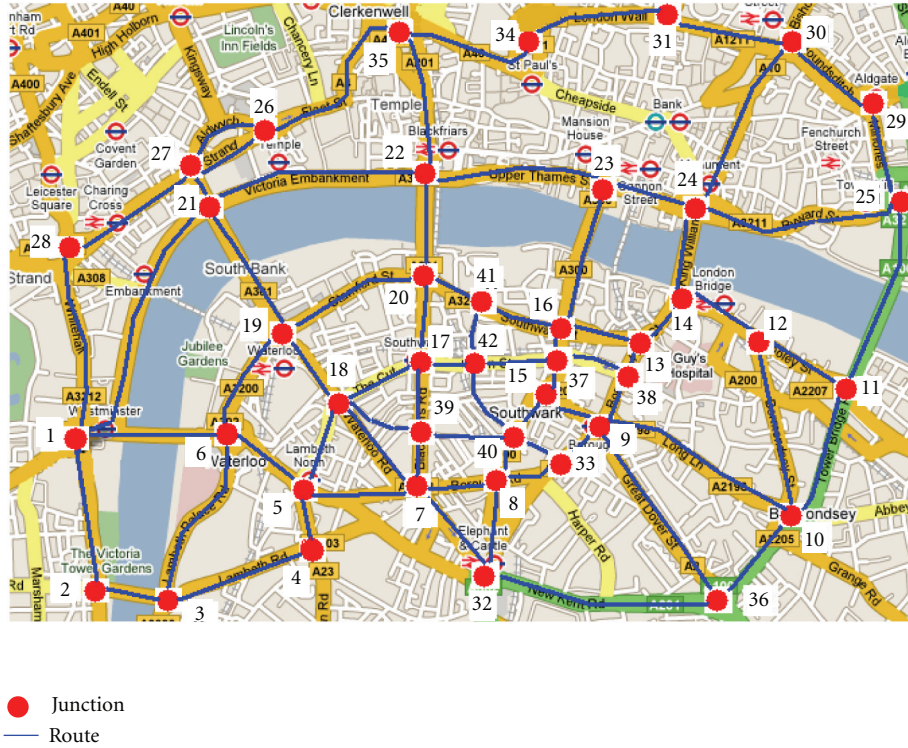


FIGURE 12: Selected region of London, UK, for simulations consisted of 42 junctions.

TABLE 1: The IF-THEN rules set for FLACS system.

Rule no.	Distance	IF Traffic	Incident Risk	THEN Local Pheromone Update
1	High	Low	Low	Very Strong
2	High	High	Low	Strong
3	High	Low	High	Strong
4	Low	Low	Low	Strong
5	Low	Low	High	Weak
6	Low	High	Low	Weak
7	High	High	High	Weak
8	Low	High	High	Very Weak

consisted of 42 junctions as in Figure 12. The statistical traffic data used in these simulations are provided by London traffic control center (LTCC). In all simulations, start time is 4:00 PM and average speed of vehicle is considered 40 km/h as default. The system is run in 30 loops with 15 initial ants as default. Number of running cycles, iterations, and ants are defined based on the number of junctions in the search area and therefore complexity of the region. In this system, the evaporation coefficient ρ is considered 0.9, $Q = 0.9$, and $\alpha = 2$. A desktop computer with Intel Core2Quad Q8300 2.5 GHz CPU and 3 MB of RAM is employed for simulations in MATLAB 2009b environment.

In the first simulation, 10 O/D pairs are randomly selected. Average of their cost averages for different number of cycles and a set of user preferences (Distance = High; Traffic = Low; Incident Risk = Low) is evaluated. As presented

in Figure 13, the considered systems have the most cost averages in the first cycles. By following an alternative descending manner, cost averages of ACS and A*-ACS systems almost arrive to a stable value by the 14th cycle. This is while the FLACS system has arrived to a stable cost average point, which is totally less than other two systems, by the 10th cycle. Since the FLACS system converges faster with a less cost average than the other two systems, this fact shows enhanced performance of the proposed system versus A*-ACS and ACS systems.

In general, the comparison between ACS and A*-ACS systems in Figure 13 illustrates that the A*-ACS algorithm has less costs average than ACS algorithm. This difference is due to local pheromone updating of the ACS by A* algorithm. In addition, comparison between A*-ACS and FLACS algorithms illustrates that the FLACS algorithm has

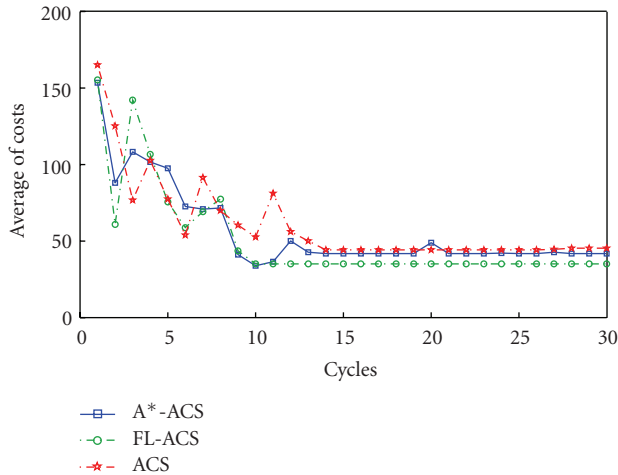


FIGURE 13: Average of 10 randomly selected O/D pairs cost averages for different number of cycles. A comparison between performance of ACS, A*-ACS, and FLACS algorithms is presented for user preference: Distance = High; Traffic = Low; Incident Risk = Low.

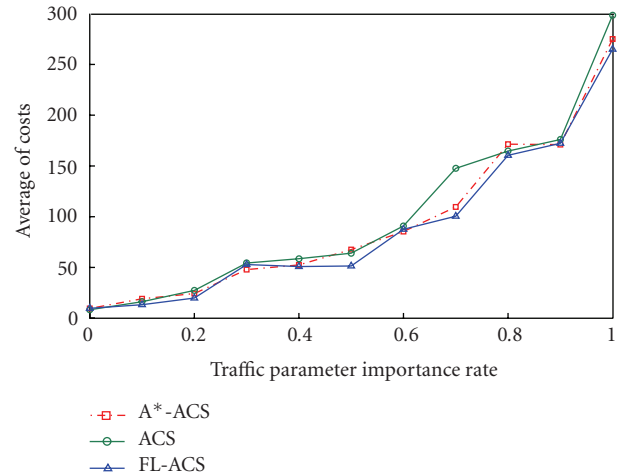


FIGURE 15: Average of 10 randomly selected O/D pairs cost averages for different “Traffic” parameter importance rates. A comparison between performance of ACS, A*-ACS, and FLACS algorithms is presented.

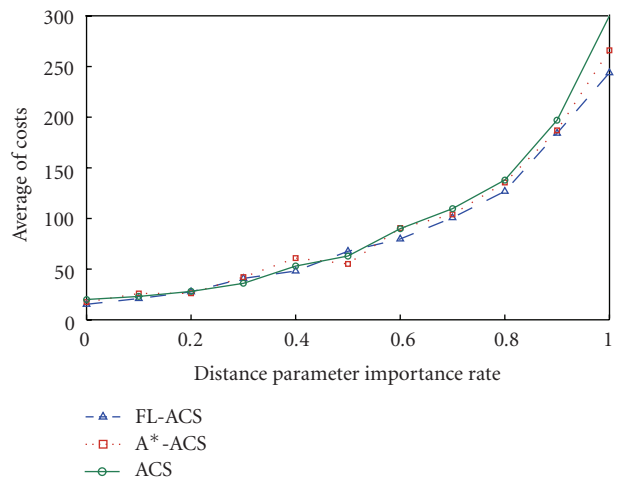


FIGURE 14: Average of 10 randomly selected O/D pairs cost averages for different “Distance” parameter importance rates. A comparison between performance of ACS, A*-ACS, and FLACS algorithms is presented.

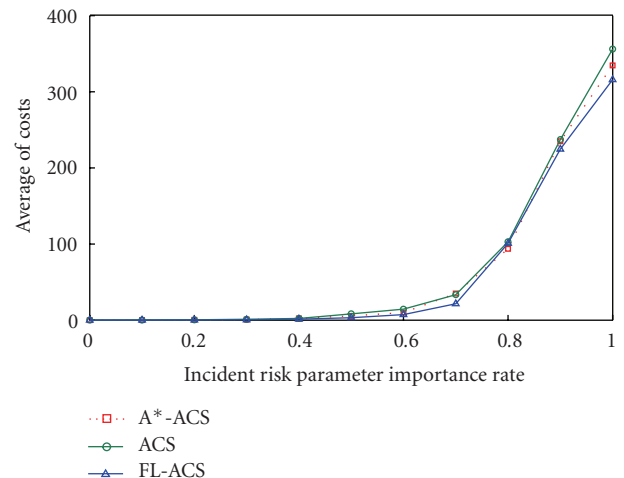


FIGURE 16: Average of 10 randomly selected O/D pairs cost averages for different “Incident Risk” parameter importance rates. A comparison between performance of ACS, A*-ACS, and FLACS algorithms is presented.

the least costs averages. This fact, demonstrates performance of FL in ACS local pheromone updating versus A* algorithm.

In another simulation, performance of the ACS, A*-ACS, and FLACS systems are compared for each “Distance,” “Traffic,” and “Incident Risk” parameters separately. For each parameter, average of 10 randomly selected O/D pairs cost averages is compared for different values of parameters importance rates, rated between 0 and 1. As it is illustrated in Figure 14, cost averages of all the three systems increase by increasing the importance rate of “Distance” parameter. Although the systems have almost similar behavior versus different “Traffic” parameter importance rates, but the ACS and FLACS algorithms have totally the most and the least average of costs, respectively. This is while it has been

demonstrated in [17] and also the previous simulation that the A*-ACS system has less average of costs than ACS system.

As analysis of cost averages for different importance rates of “Traffic” parameter demonstrates in Figure 15, the systems have different behavior for various values of this parameter. In fact, the alternations are less smoother than the “Distance” parameter. This is due to estimation of further traffic data by ANNs. Therefore, traffic data used in each system is not exactly same as the other system. By considering this fact, generally the FLACS algorithm has the least average of costs than the other algorithms. In a similar simulation, performance of the systems is studied for different values of “Incident Risk” parameter as in Figure 16. Similar to the “Distance” parameter, the systems behavior is smooth in a descending manner.

TABLE 2: Comparison between performance of ACS, A*-ACS, and FLACS algorithms for three different O/D pairs and user preference: Distance = High; Traffic = Low; Incident Risk = Low.

Route selection system	O/D Pair	Recommended direction	Cost (unitless)	Time (sec.)
ACS	6/13	6 → 5 → 18 → 17 → 42 → 37 → 38 → 13	24.257	3.211
	22/37	22 → 20 → 17 → 39 → 40 → 37	21.025	1.901
	20/36	20 → 17 → 42 → 40 → 15 → 9 → 36	27.241	3.320
A*-ACS	6/13	6 → 19 → 20 → 41 → 16 → 13	21.634	3.980
	22/37	22 → 23 → 16 → 37	19.452	2.412
	20/36	20 → 41 → 42 → 37 → 15 → 9 → 36	23.952	3.872
FLACS	6/13	6 → 19 → 20 → 41 → 16 → 13	16.548	3.412
	22/37	22 → 20 → 41 → 16 → 37	15.012	2.122
	20/36	20 → 41 → 42 → 40 → 33 → 9 → 36	19.856	3.490

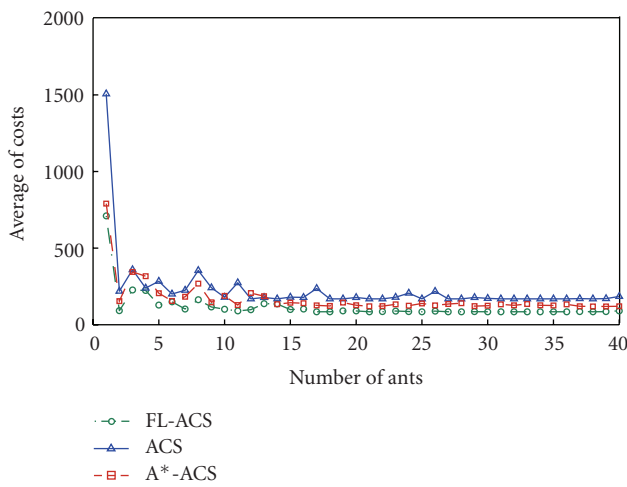


FIGURE 17: Average of 10 randomly selected O/D pairs cost averages versus different number of ants. A comparison between performance of ACS, A*-ACS, and FLACS algorithms is presented.

In Figure 17, average of 10 randomly selected O/D pairs cost averages versus different number of ants is studied. As this figure illustrates, by considering one ant in the systems as the worker, averages of costs are so high. By activating more ants, the systems averages of costs decrease while converging to a specific value of costs average. The convergence point for the FLACS, A*-ACS, and ACS systems is at 17, 17, and 18 number of ants, respectively. However, after this point, behavior of the FLACS system costs average is static, while the ACS system has more alternations than A*-ACS system. This figure clearly demonstrates performance of the proposed FLACS system versus other two systems.

In another study, performances of the systems are compared for three specific O/D pairs in Table 2. As it is indicated, the ACS has the most cost but less running time among other systems. This is while the A*-ACS, due to computational load of its A* search engine, has more running time but less cost than ACS. Even though the FLACS also has the extra FL system component respect to ACS, the simulation results demonstrate that the FLACS has less running time than A*-ACS and the least cost among

all systems. However, due to its FL system component, its running time is more than the ACS. In a specific case, the FLACS and A*-ACS have recommended similar direction for the O/D pair 6/13. However, the FLACS method has recommended this direction with less running time as well as less cost than the ACS method, which demonstrates performance of the FLACS system.

Totally, the systems have demonstrated almost the same growth pattern by parameters increment which is due to the common ACS core of these methods. However, the performance analyze of the proposed FLACS system for different values of “Distance,” “Traffic,” and “Incident Risk” parameters demonstrates its less cost average versus A*-ACS and pure ACS systems. This also means that employment of the FL technique in local pheromone updating has not forced or converged the FLACS to a specific result, but could help it to achieve optimum results with fewer costs than pure ACS and A*-ACS algorithms. The privilege of FLACS versus other methods is its dynamic behavior in different regions with different complexities due to existence of the FL part. Therefore, the system performs with fewer total costs as well as better performance.

6. Conclusion and Future Works

The proposed system in this paper introduces a dynamic route selection system which employs fuzzy logic (FL) and ant colony system (ACS) for multiparameter route selection in urban areas. This system considers a set of important parameter for city travelers: “Distance,” “Traffic,” and “Incident Risk.” However, this set can be developed by considering other parameters such as “Width” (number of lanes), “Quality” (medical treatment facilities, entertainments, etc.), and number of “Traffic Lights.”

In this work, costs of possible routes are calculated based on the adjusted desired parameters of the user. Then direction with the optimum cost is selected by using the proposed fuzzy logic-ant colony system (FLACS) algorithm. For real-time applications, fuzzy logic is considered as a management mechanism for the proposed ACS local pheromone updating. This technique improves ACS performance and prepares a real applicable dynamic system for different regions. This work can also be developed for daily life

usages by employing some advanced technologies such as vehicle-to-vehicle (V2V) communication and networking technologies. Another version of this work can also be developed for passengers, being available on their mobile cell phones or personal digital assistants (PDAs).

The proposed system can have lots of real-time applications for emergency services, tourist guides, and generally for anyone who wants to have a low-cost, safe, and comfortable journey in urban areas.

Acknowledgments

The authors would like to acknowledge the anonymous reviewers for their insightful suggestions. The authors would also like to appreciate customer service advisor of the London Traffic Control Center, United Kingdom, for providing traffic data of this project.

References

- [1] M. Barth, K. Boriboonsomsin, and A. Vu, "Environmentally-Friendly navigation," in *Proceedings of the 10th International IEEE Conference on Intelligent Transportation Systems (ITSC '07)*, pp. 684–689, October 2007.
- [2] H. Wang and B. Zhang, "Route planning and navigation system for an autonomous land vehicle," in *Proceedings of the 3rd International Conference on Vehicle Navigation and Information Systems (VNIS '92)*, pp. 135–140, Norway, 1992.
- [3] L. W. Lup and D. Srinivasan, "A hybrid evolutionary algorithm for dynamic route planning," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 4743–4749, Singapore, September 2007.
- [4] S. Edelkamp, S. Jabbar, and T. Willhalm, "Geometric travel planning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 5–16, 2005.
- [5] C. Yanyan, M. G. H. Bell, and K. Bogenberger, "Reliable pretrip multipath planning and dynamic adaptation for a centralized road navigation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 14–19, 2007.
- [6] S. Kim, M. E. Lewis, and C. C. White, "State space reduction for nonstationary stochastic shortest path problems with real-time traffic information," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 3, pp. 273–284, 2005.
- [7] C. Yanyan, L. Ying, and D. Huabing, "The model of optimum route selection in vehicle automatic navigation system based on unblocked reliability analyses," *IEEE Intelligent Transportation Systems Proceedings*, vol. 2, pp. 975–978, 2003.
- [8] J. Miura, M. Itoh, and Y. Shirai, "Toward vision-based intelligent navigator: its concept and prototype," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 2, pp. 136–146, 2002.
- [9] G. K. H. Pang, K. Takahashi, T. Yokota, and H. Takenaga, "Adaptive route selection for dynamic route guidance system based on fuzzy-neural approaches," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 6, pp. 2028–2041, 1999.
- [10] G. Pang and M.-H. Chu, "Route selection for vehicle navigation and control," in *Proceedings of the 5th IEEE International Conference on Industrial Informatics (INDIN '07)*, pp. 693–698, June 2007.
- [11] D. Teodorovic and S. Kikuchi, "Transportation route choice model using fuzzy interface technique," in *Proceedings of the 1st International Conference Uncertainty Modeling and Analysis: Fuzzy Reasoning, Probabilistic Models, and Risk Management*, pp. 140–145, Maryland University, College Park, 1990.
- [12] Y. Kambayashi, Y. Tsujimura, H. Yamachi, and H. Yamamoto, "Integrating uncomfortable intersection-turns to subjectively optimal route selection using genetic algorithm," in *Proceedings of the 5th IEEE International Conference on Computational Cybernetics (ICCC '07)*, pp. 203–208, Tunisia, October 2007.
- [13] S. Zidi, S. Maouche, and S. Hammadi, "Real-time route planning of the public transportation system," in *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC '06)*, pp. 55–60, Canada, September 2006.
- [14] A. Broggi, M. Cellario, P. Lombardi, and M. Porta, "An evolutionary approach to visual sensing for vehicle navigation," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 1, pp. 18–29, 2003.
- [15] H. Salehinejad, F. Pouladi, and S. Talebi, "A new route selection system: multiparameter ant algorithm based vehicle navigation approach," in *Proceedings of the International Conference on Computational Intelligence for Modeling, Control and Automation*, pp. 1089–1094, IEEE Computer Society, Vienna, Austria, 2008.
- [16] H. Salehinejad and S. Talebi, "A new ant algorithm based vehicle navigation system: a wireless networking approach," in *Proceedings of the International Symposium on Telecommunications (IST '08)*, pp. 36–41, Tehran, Iran, August 2008.
- [17] H. Salehinejad, H. Nezamabadi-Pour, S. Saryazdi, and F. Farrahi-Moghaddam, "Combined A*-ants algorithm: a new multi-parameter vehicle navigation scheme," in *Proceedings of the 16th Iranian Conference on Electrical Engineering (ICEE '08)*, pp. 154–159, Tehran, Iran, 2008.
- [18] S. Abed and C. Swann, "Analysis of freeway traffic timeseries data by using Box Jenkins techniques," *Transportation Research*, no. 72, pp. 1–9, 1979.
- [19] H. Nicholson and C. D. Swann, "The prediction of traffic flow volumes based on spectral analysis," *Transportation Research*, vol. 8, no. 6, pp. 533–538, 1974.
- [20] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transportation Research. Part B*, vol. 18, no. 1, pp. 1–11, 1984.
- [21] I. Ohe, H. Kawashima, M. Kojima, and Y. Kaneko, "A method for automatic detection of traffic incidents using neural networks," in *Proceedings of the Vehicle Navigation and Information Systems Conference in Conjunction with the Pacific Rim TransTech Conference, A Ride into the Future*, pp. 231–235, 1995.
- [22] C. Taylor and D. Meldrum, "Freeway data prediction using neural networks," in *Proceedings of the Vehicle Navigation and Information Systems Conference in Conjunction with the Pacific Rim TransTech Conference, A Ride into the Future*, pp. 225–230, 1995.
- [23] M. Dorigo, V. Maniezzo, and A. Colnari, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 26, no. 1, pp. 29–41, 1996.
- [24] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 1–24, 1997.
- [25] L. W. Dong and C. T. Xiang, "Ant colony optimization for VRP and mail delivery problems," in *Proceedings of the IEEE International Conference on Industrial Informatics (INDIN '06)*, pp. 1143–1148, August 2006.

- [26] C. L. Liu, "Best path planning for public transportation systems," in *Proceedings of the IEEE 5th International Conference on Intelligence Transportation Systems*, pp. 834–839, 2002.
- [27] M. Dorigo, "Ant foraging behavior, combinatorial optimization, and routing in communication networks," *Swarm Intelligence: From Natural to Artificial Systems*, pp. 25–107, 1999, Santa Fe Institute Studies in the Sciences of Complexity.
- [28] S. Favuzza, G. Graditi, M. G. Ippolito, and E. R. Sanseverino, "Optimal electrical distribution systems reinforcement planning using gas micro turbines by dynamic ant colony search algorithm," *IEEE Transactions on Power Systems*, vol. 22, no. 2, pp. 580–587, 2007.
- [29] CH. Schuh, "Fuzzy sets and their application in medicine," in *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS '05)*, pp. 86–91, Detroit, Mich, USA, 2005.
- [30] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proceedings of the Institution of Electrical Engineers*, vol. 121, no. 12, pp. 1585–1588, 1974.
- [31] P. Chemouil, J. Khalfet, and M. Lebourges, "Fuzzy control approach for adaptive traffic routing," *IEEE Communications Magazine*, vol. 33, no. 7, pp. 70–76, 1995.
- [32] T. Fukuda and T. Shibata, "Theory and applications of neural networks for industrial control systems," *IEEE Transactions on Industrial Electronics*, vol. 39, no. 6, pp. 472–489, 1992.
- [33] H. Demuth, M. Beale, and M. Hagan, *Neural Network Toolbox For Use with MATLAB®*, The MathWorks, 2006.
- [34] V. Maniezzo, L. M. Gambardella, and F. De Luigi, *Ant Colony Optimization, New Optimization Techniques in Engineering*, Springer, Berlin, Germany, 2004.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

