



Dynamic routing with ant system and memory-based decision-making process

Sina Abolhoseini¹ · Ali Asghar Alesheikh¹

Accepted: 13 September 2020 / Published online: 26 September 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Dynamic routing is an essential tool for today's cities. Dynamic routing problems can be solved by modelling them as dynamic optimization problems (DOPs). DOPs can be solved using Swarm Intelligence and specially ant colony optimization (ACO) algorithms. Although different versions of ACO have already been presented for DOPs, there are still limitations in preventing stagnation and premature convergence and increasing convergence rate. To address these issues, we present an in-memory pheromone trail and an algorithm based on it (named AS-gamma) in the framework of ACO. In-memory pheromone trail is effectively increasing diversity after a change in an environment. Results of experimenting AS-gamma in three scenarios on a real-world transportation network with different simulated traffic conditions demonstrated the effectiveness of the presented in-memory pheromone trail method. The advantages of AS-gamma over three existing DOP algorithms have been illustrated in terms of solutions quality. Offline performance and accuracy measures indicate that AS-gamma faces less stagnation, premature convergence and it is suitable for crowded environments.

Keywords Dynamic optimization problem · Ant colony optimization · Routing · Memory-based decision-making

1 Introduction

Nowadays, vehicle navigation is an important task in large municipalities and faces great challenges in transportation networks (Mavrovouniotis et al. 2017a, b; Hwang and Jang 2019). Usually, congestions happen during rush hours or when accidents in different locations occur. These events change the state of transportation networks and make them dynamic. The distance between two consecutive changes in environment is called a state. Finding the optimal route in a dynamic transportation network is referred to as dynamic routing problem. Real-time information regarding traffic conditions of streets, can be provided using intelligent infrastructure-based services such as Ubiquitous Sensor Networks (USNs) (Sharif and Sadeghi-Niaraki 2017) or by infrastructure-dependent services such as Vehicle Ad-hoc Networks (VANETs) (Felicia and Lakshmanan 2016). Giving drivers real-time traffic information can be very helpful in providing the possibility of choosing better routes (Singh

et al. 2015). A better system will also compute detours and give drivers instructions to change their previous route. Currently, computers, networks and the Internet facilitate solving such problems ((Aghdam and Alesheikh 2018).

Solutions for static routing problems can be classified into three main classes: classical, heuristic and metaheuristic (Abolhoseini and Sadeghi Niaraki 2016). Classical algorithms search entire transportation networks that can be modelled as a graph with arcs and nodes as streets and intersections (e.g. Dijkstra, Bellman-Ford). Heuristic algorithms utilize additional information to guide their search toward better solutions and to avoid searching entire transportation networks (e.g. A*). A brief review of classical and heuristic static route finding algorithms can be found in (Abolhoseini and Sadeghi Niaraki 2016). Metaheuristic algorithms try to find best solutions by simulating natural phenomena (e.g. Neural Networks, Genetic Algorithm, Simulated Annealing, Ant Colony Optimization (ACO), Particle Swarm Optimization, Tabu Search). Amongst all well-known metaheuristic algorithms is ACO which was developed for discrete spaces. It is mostly used for discrete/combinatorial problems (Mavrovouniotis et al. 2017a, b). In ACO, ants search the surrounding of their nests and release a chemical named pheromone to help other ants use their

✉ Sina Abolhoseini
abolhoseini.sina@gmail.com

¹ Faculty of Geodesy and Geomatics Engineering, K. N. Toosi University of Technology, Tehran, Iran

experience and decide optimally (Abolhoseini and Sadeghi Niaraki 2017). In some studies, the concept of pheromone trails was used to navigate a group of robots to search an environment (Lima et al. 2016; Souza and Lima 2019). Souza and Lima added a tabu-based memory capability to robots in (TIACA) (Souza and Lima 2019) to improve the performance of their previous algorithm (IACA) (Lima et al. 2016). They came to the conclusion that TIACA algorithm significantly increased the performance of robots in a surveillance task compared to IACA.

Dynamic routing problems can be modelled as a dynamic optimization problem (DOP) (Mavrovouniotis, et al. 2017a, b). Although many different approaches have been proposed to solve DOPs by the ACO algorithm, there still exist problems on preventing stagnation and premature convergence whilst increasing convergence rate after changes in states of environments (Dorigo and Stützle 2019). A dynamic routing algorithm can be used in crowded environments to prevent traffic congestions and accidents if it can perform better in the mentioned issues.

Despite the fact that many approaches have been developed based on ACO to solve dynamic routing problems, they face stagnation, premature convergence and slow convergence rate when they are applied to real-world transportation networks. To address these problems, we present a new memory scheme algorithm named AS-gamma, in the framework of ant colony algorithm, where an independent pheromone trail and a variable for the memory parameter (γ) are suggested to deal with changes in transportation networks. The contributions of this research are as follows:

- We present an independent pheromone trail (we refer it to as *in-memory pheromone* trail) and a variable (γ) for transferring information from previous states of a dynamic environment to its new states. Pheromone trails from previous states optimize the algorithm to find optimum in the new states faster using the knowledge of the previous populations.
- With the proposed *in-memory pheromone* trail and the variable for memory, an algorithm named AS-gamma is suggested based on the Ant System algorithm to solve dynamic routing problems. New populations in AS-gamma are generated based on three parameters, heuristic value, pheromone trail value and *in-memory pheromone* trail value on each link.
- The performance and effectiveness of AS-gamma are evaluated on a real-world transportation network of a crowded city with simulated traffic condition in three scenarios (low, medium and high traffic condition). AS-gamma is compared to three existing state-of-art algorithms (pheromone conservation method (AS-MON) (Montemanni et al. 2005), a complete restart method (AS-Restart) (Jin et al. 2013) and a memory-based immi-

grants scheme algorithm (AS-IMG) (Mavrovouniotis and Yang 2015) in solving dynamic routing problem, which revealed the superiority of AS-gamma.

The rest of the paper is organized as follows. Section 2 assesses the related research in this field. In Sect. 3 after a brief explanation of the AS algorithm, AS-gamma is presented to solve the dynamic routing problem. Moreover, the evaluation measures are explained. Section 4 begins with the specific characteristics of a real-world dataset, as well as the process of parameter setting following by experimental results. Finally, in Sect. 5 the conclusions are presented.

2 Related work

In the past few years, a significant number of studies have been conducted for routing (Zhang and Zhang 2018). Finding optimal routes in a dynamic transportation network can be considered a dynamic optimization problem (DOP) (Mavrovouniotis et al. 2017a, b). A DOP can be defined as a sequence of static problem states to be optimized (Mavrovouniotis and Yang 2013a, b). Route finding and vehicle navigation can be categorized as a discrete problem where all optimization variables are discrete values (Mavrovouniotis and Yang 2011). The main difference between discrete and continuous problems is the finite search space in discrete problems. Usually optimization problems with network environments are modelled as weighted graphs (e.g. transportation networks, social networks, telecommunication networks, rail networks).

Many DOPs are NP-hard and they cannot be solved within a polynomial computation time (Manusov et al. 2018). Swarm intelligence algorithms have recently been developed extensively to deal with such problems. These algorithms have been applied to solve different DOPs for both discrete and continuous search spaces. Between all well-known metaheuristic algorithms, Ant Colony Optimization (ACO) was developed for discrete spaces. It is mostly used for discrete/combinatorial problems (Mavrovouniotis et al. 2017a, b). This algorithm is inspired from the nature of ants; it is population-based and iterative (Colorni et al. 1992).

Different strategies were utilized to solve dynamic discrete optimizations such as increase diversity after change, maintain diversity during execution, memory schemes and multiple population strategies (Mavrovouniotis et al. 2017a, b). *Increase diversity after a change* methods include complete restart and partial restart methods. Complete restart does not use the knowledge of ants from previous states of a problem (Jin et al. 2013). Some researchers used partial restart to improve the convergence of the ACO (Skinderowicz 2016). Partial restart

methods can keep the knowledge of previous states and increase diversity simultaneously in part of population to solve DOPs. Prakasam and Savarimuthu proposed a local restart strategy for efficient search of space during node replacement in dynamic Travelling Salesman Problem (TSP) (Prakasam and Savarimuthu 2019). Montemanni et al. divided dynamic problems into time slices and considered each time slice as a static optimization problem (Montemanni et al. 2005). Once the optimization of a time slice was completed, the pheromone trails that represented the quality of the solution, were passed on to the next time slice. They introduced a new parameter in updating the pheromone trails of next time slices to regulate pheromone conservation (Montemanni et al. 2005). Euchi et al. conducted a similar research and gained better results by adding 2-opt local search in their method (Euchi et al. 2015). Xiang et al. proposed a demand coverage diversity adaptation method to solve DVRP (Xiang et al. 2020). This method handled newly appeared customers far from planned routes based on their locations.

Maintain diversity during execution methods do not need detecting changes to increase the diversity of ants. For example, an algorithm was introduced by Mavrovouniotis et al. with an adaptive pheromone evaporation rate (ρ) to speed up the evaporation rate when the algorithm was approaching stagnation (Mavrovouniotis and Yang 2013a, b, 2014a). Immigrants schemes are very popular in solving DOPs with ACO. In these schemes, ants from the previous states were selected to deposit pheromone in new states (Mavrovouniotis and Yang 2015; Gao et al. 2016; Mavrovouniotis and Yang 2014a, b). Modification in the ants generation of ACO was also investigated by Liu (2005). This study explained that the random proportional rule for choosing edges by ants is inefficient at exploring paths in large Travelling Salesman Problems (TSPs). Therefore, they suggested a rank-based ACO based on rank-based nonlinear selection pressure function and a modified Q-learning method to solve reinforcement learning problems. In 2019, Chowdhury et al. presented a novel ACO framework to maintain diversity by transferring knowledge to current pheromone trails from latest states of problems. They developed Adaptive Large Neighbourhood Search (ALNS) based immigrant schemes (Chowdhury et al. 2019).

Memory scheme strategy methods transfer knowledge from previous states of environments to new states. The main idea was to store best solutions in memory and reuse them in new states (Mavrovouniotis and Yang 2012, 2015). In addition, memory-based immigrants schemes were presented to deal with dynamic changes during optimization. In these approaches, several best solutions from previous states were stored in an external memory (Mavrovouniotis and Yang 2013a, b). Part of the population in new states were generated using the solutions stored in the memory.

Multiple population methods allocate different populations to different parts of the space. In this way, the diversity is increased and maintained. Xu et al. used an enhanced ACO to prevent premature convergence in dynamic routing problems by fusing ACO with improved K-means and crossover operation (Xu et al. 2018). In (Yang et al. 2015) multi-colony ACO was used with different pheromone trails to maximize the search space. Garcia et al. used ACO algorithm to solve route finding for robots with obstacle avoidance. A memory capability was also added to avoid stagnation. Visited nodes with obstacles were temporally marked to avoid re-testing (Garcia et al. 2009). ACO has been widely used in multi root systems. de Almeida et al. presented an online path planner for multi-robot systems. They used the concept of inverted pheromone trails to avoid redundant exploration of regions and areas to explore an entire environment faster. It can be used for dynamic obstacles (de Almeida et al. 2019).

In spite of different approaches that have been developed to address routing problems by others, most of them confront stagnation, premature convergence and slow convergence rate when they are used in real-world situations. We recommended *in-memory pheromone trails* and an independent variable (γ) within the procedure of generating ants to control the impact of *in-memory pheromone trails*. As it can be seen in Fig. 1, two pheromone trails are deposited on each graph edges. One is the pheromone trail deposited by the current generation of ants to optimize the current state of the problem. The other one is the *in-memory pheromone* trail, which is from the previous

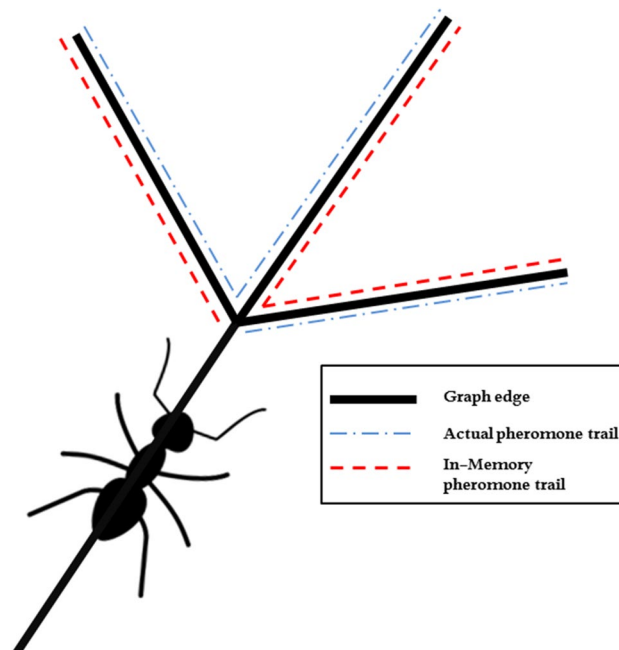


Fig. 1 Using the In-memory pheromone trail to decide which node to take as next

state of the environment. Both pheromone trails are used independently whilst generating ants. The AS-gamma algorithm is developed based on the *in-memory pheromone* trails and compared to the AS-IMG, AS-MON and AS-Restart algorithms. AS-IMG is an elitism-based immigrants ACO proposed by Mavrouniotis et al. to solve dynamic optimization problems, especially Dynamic Vehicle Routing Problem (DVRP) (Mavrouniotis and Yang 2013a, b). In AS-IMG, the knowledge from previous states of an environment was transferred to new states through creating a population of ants in new states based on the elite solutions of previous states. AS-MON is a method introduced by Montammini et al. to solve DVRP (Montammini et al. 2005). In this method, the knowledge of previous states of a problem was transferred to new states through initializing pheromone values on the edges of the graph of the problem based on previous states final pheromone trails with a procedure named pheromone conservation. AS-Restart is the classic Ant System algorithm without any modification. After each change in states of a problem, the algorithm optimizes the new states without any knowledge from its previous generations. In addition, AS-gamma is different from the algorithm proposed by Souza and Lima in 2019 (TIACA). TIACA used a tabu-based memory for robots that enabled them to store their previous positions in the memory. When robots wanted to decide for their next movements, they searched their memory to avoid moving to recently visited positions.

3 Methodology

The process of finding high-quality food resources by real ants inspires ACO. The goal of real ants is to find the shortest route to a food resource from their nest. Each ant forms a solution for a problem (Dorigo and Stützle 2004). In route finding problem, each ant is a sequence of nodes from an origin to a destination. After generating ants, pheromone amounts of the edges they traversed, are updated based on the quality of each ant regarding an objective function. Different ACO algorithms have been developed by researchers such as Ant Colony System (ACS), Max–Min Ant System (MMAS) and Ant System (AS) (Rizzoli et al. 2007). AS is a simple implementation of ACO algorithms which was developed in 1996 by Dorigo et al. (Dorigo et al. 1996). The process of constructing ants is based on the calculated probability of Eq. (1) from node i to its neighbour j (Abolhoseini and Sadeghi Niaraki 2017).

$$P_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{h \notin tabu_k} (\tau_{ij})^\alpha (\eta_{ij})^\beta} & \text{if } j \notin tabu_k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

In the above equation, $tabu_k$ is the set of visited nodes by k th ant. τ_{ij} is the amount of pheromone value on the edge between node i and j . α determines the influence of

pheromone value on selecting the next node. η_{ij} is the desirability of choosing the edge between node i and j , and typically, it is equal to the reverse of the edge’s cost. β controls the influence of η_{ij} . Equation (1) is the basis of generating ants in AS algorithms. In Sect. 3.1, the modification of this equation is explained to consider *in-memory pheromone* trail in generating ants in AS-gamma.

After constructing the solutions (ants), the pheromone value of all edges must be evaporated (decrease) to reduce the probability of them being chosen by other ants. Then, the pheromone value of the traversed path by the best ant must be increased due to its suitability to increase the probability of selecting the path by other ones. These two tasks are applied through Eq. (2) (Abolhoseini and Sadeghi Niaraki 2017).

$$\tau_{ij}^{new} = (1 - \rho)\tau_{ij}^{old} + \Delta\tau_{ij}^{best} \quad (2)$$

$\rho \in [0, 1)$ is called pheromone evaporation rate and it has a constant value. Increasing the value of pheromone amount on each link, traversed by the best ant (τ_{ij}^{best}), is calculated through Eq. (3). In Eq. (3), Q is constant and f_k is the value of the objective function for the best ant in that iteration (best-of-generation).

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{f_k} & \text{if the best ant traversed link}(i,j) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

AS iterative process stops when one of the predefined conditions is satisfied. Specific number of iterations, running time or constant value of pheromones after a certain number of iterations, can be possible stopping criteria.

3.1 AS-gamma

Treating DOP can be easily done by a complete restart at each state of an environment. The pheromone trails of AS can be re-initialized with an equal amount whenever a change is detected. By a complete restart, the diversity is highly increased but all the knowledge that gained through iterations in previous states is lost (Mavrouniotis and Yang 2013a, b). As it is evident, this strategy is not efficient.

In this paper, a new memory-based scheme is presented to use the knowledge of previous states of problems in dynamic environments. In this new scheme, the pheromone trails of the last state are transferred to the new state as *in-memory pheromone* trails and an additional parameter is added to Eq. (1) to determine the effect of the *in-memory pheromone* trails in generating new ants. In Eq. (4), parameter γ_{ij} determines the amount of *in-memory pheromone* on the edge between node i and j . δ determines the influence of *in-memory pheromone* trail on selecting the next node.

$$P_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta (\gamma_{ij})^\delta}{\sum_{h \notin \text{tabu}_k} (\tau_{ij})^\alpha (\eta_{ij})^\beta (\gamma_{ij})^\delta} & \text{if } j \notin \text{tabu}_k \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

This probability computation equation can lead to use the knowledge from the last state of an environment. Higher values of δ will result in premature convergence to the solution of the previous state, whilst lower values of δ do not give the proper influence from the previous knowledge. As a consequence, this parameter should be set to an optimal value.

In this paper, we tried to add the previous experience effects to the process of generating ants by adding a new parameter (γ_{ij}) to the Eq. (1). The flowchart of AS-gamma is depicted in Fig. 2. In the followings, each step of the algorithm has been explained (Fig. 2).

In step 1, optimization begins in a dynamic environment with no memory at the first state. Ants are generated based on probability calculation of Eq. (1) in the first state, because there is no memory. In the subsequent states, the algorithm uses Eq. (4) to consider the effect of the *in-memory pheromones* stored in step 7 in the optimization process.

In step 2, fitness is calculated for each generated ant in step 1, based on the objective function. In our case, the objective function is the total travel time of the path in which ants completed their trip. In this study, it is assumed that the travel time for each link is observed, processed, and provided to the algorithm by the traffic control organizations.

In step 3, the best ant amongst the population is selected (based on the calculated fitness in step 2) to update the

pheromone values on the network. This update is based on Eq. (2).

In step 4, the optimization stop criteria are checked. If the stop criteria are met, the optimization will stop and the algorithm will go to step 5. If the stopping criteria are not met, the optimization continues, the algorithm returns to step 1 and a new population is generated.

In step 5, 6 and 7, the algorithm gets the pheromone values on the network, normalizes them and stores them in its memory. Pheromone values are high after an optimization is completed, so it cannot be used in the next state of the problem without normalization. Normalization prevents premature convergence. Normalizing the pheromone values in the memory is done using Eq. (5). ζ is the inflation factor and it is needed to better guide ants toward the best solution of the previous state. High amounts of ζ will cause premature convergence and low values of this factor will cause no use of the previous knowledge. So, this factor must be tuned (Sect. 4.2).

$$\gamma_{ij} = \frac{\gamma_{ij} - \min \gamma_{ij}}{\max \gamma_{ij} - \min \gamma_{ij}} \times \zeta \quad (5)$$

After the 7 steps explained above, the optimization process will be resumed if there is another change in the environment (transportation network). As mentioned before, after the first state, next ones starts with generating ants based on the probability calculation of Eq. (4). After convergence, pheromone trails are stored in memory for the next state. This process will continue until no change in the environment is detected.

3.2 Performance measurements

There are different measurements to evaluate the behaviour of DOP algorithms. *Collective mean fitness* (Morrison 2003), *best for change* (Mavrovouniotis et al. 2017a, b; Mavrovouniotis et al. 2014) and *modified offline performance* (Mavrovouniotis and Yang 2015; Branke 2012) are the examples of such measures. The proximity of the values to the optimal value is also investigated. Therefore, a global optimum value is required to be defined during the dynamic changes. *Offline error* (Mavrovouniotis et al. 2015), *average score* (Li et al. 2008), *accuracy* (Schaefer et al. 2002) and measurements based on the distance between the solution found by the algorithm and the global optimum are the examples of such measures. In this research, an investigation of the best-so-far solution cost after a dynamic change is made to evaluate the performance of 10 independent executions for each scenario. For a better comparison, all the algorithms performed by the same number of iterations and for each iteration the same number of ants are generated. The overall offline performance is defined in Eq. (6) (Jin and

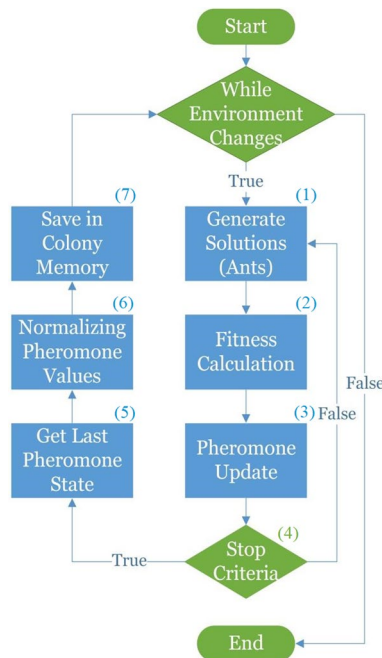


Fig. 2 Flowchart of the AS-gamma

Branke 2005). I is the total number of iterations, E is the number of independent executions and P_{ij}^* is the best-so-far solution since the last change in the environment. The main purpose of this optimization is to minimize the cost. Therefore, lower values of overall offline performance mean that the algorithm performs better.

$$\overline{P_{OFF}} = \frac{1}{I} \sum_{i=1}^I \left[\frac{1}{E} \sum_{j=0}^E P_{ij}^* \right] \tag{6}$$

Accuracy is another measure for evaluating the performance of swarm intelligence algorithms in dynamic environments. Accuracy stands for the average of the difference between the value of the current best individual in the population and the optimum value, just before a change in the environment (Eq. 7) (Trojanowski and Michalewicz 1999). τ is the number of iterations in one state of the problem. $err_{i,\tau-1}$ is the difference between the value of the best ant in a state of the problem and its optimum solution, just before a change in the environment. Also, K is the number of changes in the environment.

$$Acc = \frac{1}{E} \left[\frac{1}{K} \sum_{i=1}^I [err_{i,\tau-1}] \right] \tag{7}$$

For computing the Acc , the global optimum value must be known. To estimate the exact value of the global optimum, we have used Dijkstra algorithm to compute the fastest route between origin and destination nodes on the graph for each state of the problem. Solutions must be close to the global optimum value, so lower values of Acc means the algorithm performs better. The cost of each link is the travel time of traversing the link, which can be obtained by different providers. Collecting traffic information and dealing with their uncertainties are out of the scope of this study.

4 Results and discussion

This section is devoted to experimental evaluations of AS-gamma algorithm on three simulated scenarios. The adopted dataset will be described in detail in Sect. 4.1; Sect. 4.2 will document the parameter tuning phase for AS-gamma; and in Sect. 4.3 the numerical results achieved by the algorithm will be presented. The algorithms have been coded in JAVA, and all the tests have been carried out on a 3.4 GHz Intel Core(TM) i7 PC with 16 GB of RAM.

4.1 Experimental setup

For evaluating the performance of AS-gamma in solving dynamic route finding, a part of transportation network in district 1 of Shiraz, Fars located in south of Iran was selected (Fig. 3). This part of Shiraz downtown is very crowded due to its shopping centres, malls and cinemas. The infrastructure of the streets in this area is also significantly poor and causes heavy congestions in rush hours. A dynamic routing service can significantly help drivers to avoid congested and crowded streets. This part of Shiraz streets consists of 554 nodes and 1329 arcs.

Three scenarios are designed to test AS-gamma. In all of them, it is assumed that a driver wants to travel from node 0 and to node 500. These scenarios are implemented in SUMO software. SUMO is an open source, microscopic and continuous road traffic simulation package for handling large road networks (Blokpoel et al. 2016). We use this software to model a dynamic transportation environment and test AS-gamma. Each scenario has a different set of traffic parameters. Low traffic, moderate traffic and high traffic environments are generated based on the parameters in Table 1. *Count* parameter defines the number of vehicles generated per hour and lane-kilometre

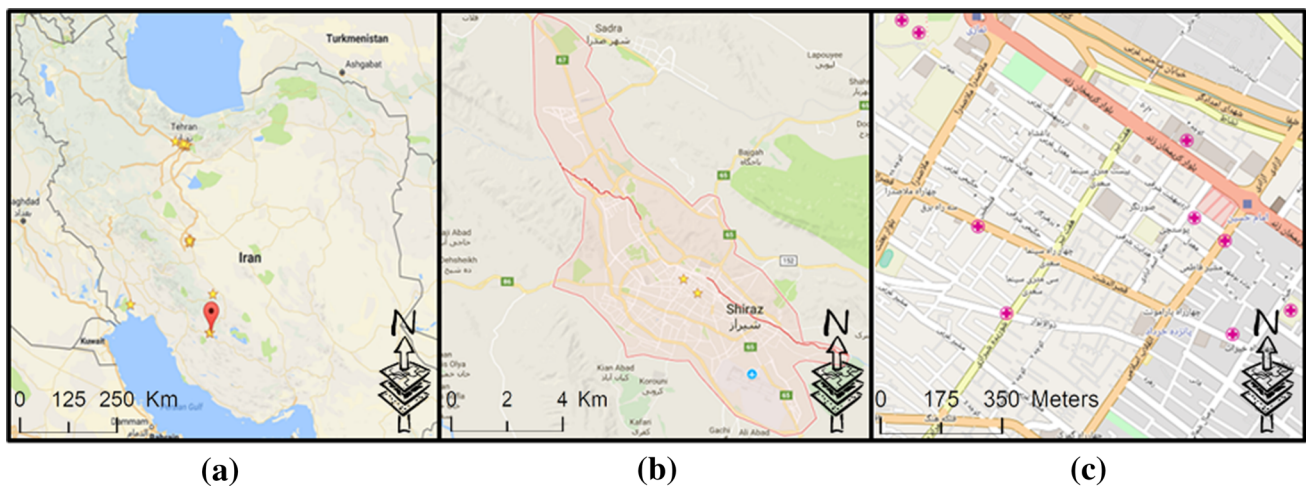


Fig. 3 The study area a Map of Iran b Shiraz City c Part of Shiraz transportation network

Table 1 Scenario Parameters in SUMO software

| | Count of vehicles per hour and lane-kilometre | | |
|------------------------|---|-----|------------|
| | Passenger | Bus | Motorcycle |
| Low | 200 | 8 | 20 |
| Moderate | 300 | 12 | 30 |
| High | 600 | 16 | 60 |
| Through traffic factor | 10 | 5 | 10 |

Table 2 Tested values for tuning the parameters

| Parameter | Tested values |
|-----------|--|
| δ | [5, 4, 3, 2, 1] |
| ζ | [4, 3, 2, 1, 0.5] |
| Q | [0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1] |
| ρ | [0.006, 0.005, 0.004, 0.003, 0.002, 0.001] |

in a simulation environment. *Through traffic factor* determines how many times a lane at the boundary of the environment can be selected as origin or destination. It should be noted that we have considered 10 states for each scenario.

4.2 Parameter setting

Most of the parameters are those of the AS algorithm uses to tackle the static route finding. In general, tuning parameters is achieved through experiments (Eiben et al. 1999) but the common parameters for all ACO algorithms are set to typical values (i.e. $\alpha=1, \beta=5$) and the number of ants are set to 50 (Mavrovouniotis et al. 2017a, b). First, Q and ρ are tuned by considering a static state and repeat 10 tests for each of the values in Table 2. Then δ and ζ are set. Setting δ and ζ are important in determining the influence of *in-memory pheromone trail* on selecting the next node. δ and ζ must be tuned on an environment with states being changed over time. In this study, δ and ζ are tuned by testing different values for 10 independent executions for all the states of the problem. At each time slice, the difference between the optimized solution and the global optimum is computed and averaged for all the states of the environment. Afterwards, the averages of these values are compared. Tested values for each parameter are reported in Table 2.

Table 3 Final tuned values for AS-gamma

| Params | Pop | Iter | α | β | ρ | Q | δ | ζ |
|--------|-----|------|----------|---------|--------|-----|----------|---------|
| Value | 50 | 100 | 1 | 5 | 0.004 | 0.2 | 1 | 2 |

Table 4 Offline performance and accuracy of the algorithms in three different scenario

| | Low traffic | Moderate traffic | High traffic |
|---------------------|--------------------|--------------------|--------------------|
| Offline performance | | | |
| AS-gamma | 18.08242103 | 17.91360833 | 17.79635626 |
| AS-Restart | 19.08879231 | 19.24601667 | 19.17348884 |
| AS-MON | 18.28530973 | 18.40431241 | 18.00316454 |
| AS-IMG | 18.230022 | 17.40097389 | 19.6698739 |
| Accuracy | | | |
| AS-gamma | 0.461287 | 0.171560 | 0.280050 |
| AS-Restart | 0.903035 | 0.957489 | 0.950472 |
| AS-MON | 1.125278 | 1.246533 | 0.630100 |
| AS-IMG | 0.681773 | 0.150935 | 0.490022 |

*All the values are unit free due to data normalization
The numbers in bold are the best results among four algorithms

Final values for all the parameters in AS-gamma are presented in Table 3. The algorithm which was designed by Montemanni (AS-MON) is one of the algorithms we used for comparison. AS-MON applies the pheromone conservation procedure that introduces a new parameter (γ_r). This parameter further needs to be tuned (Montemanni et al. 2005). For tuning this parameter, the procedure explained in the above-mentioned study is used. We have tested 1, 0.5, 0.3 and 0.1 values and ran each value for 10 times. Results indicated that 0.1 was the proper value for this problem.

4.3 Experimental results

Three environments with low, moderate and high traffic conditions are defined and tested in this section. Global optimum values are calculated using Dijkstra algorithm for each state of the environment in each scenario. Thus, the global optimum is known in each state of the scenarios during the dynamic changes.

The results of the offline performance and the accuracy of the algorithms are presented in Table 4. Moreover, to better understand the dynamic performance of the proposed algorithms, the offline performances of the algorithms are plotted in Fig. 4 for all the scenarios.

The smaller the measured values are, the better the results are. It is evident that AS-gamma outperforms other three algorithms in low and high traffic condition scenarios in offline performance and accuracy. In particular, lower values of accuracy means that the obtained solution is closer to the optimum just before the environment is changed. In AS-gamma, when

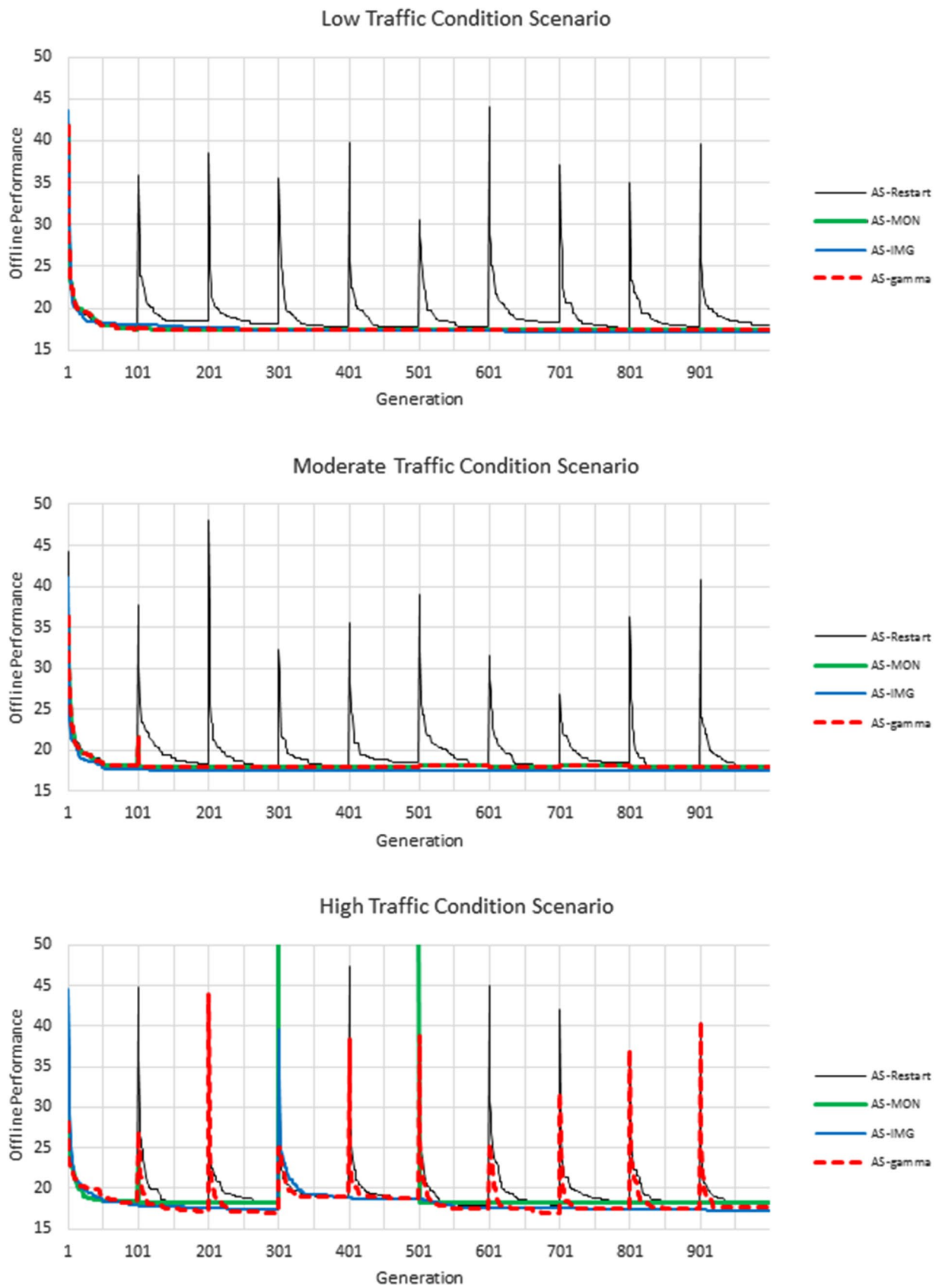


Fig. 4 Offline performance of the AS-gamma, AS-MON, AS-IMG and AS-Restart algorithms for the scenarios with low, moderate and high traffic condition

the state of the environment changes, ants do not lose their optimality due to the *in-memory pheromone* trails. In addition, they have enough time to search for the new optimum close to the best solution of the previous state. This process is completely missing in the AS-restart. In comparison, AS-MON uses the pheromone conservation procedure to keep the knowledge of previous states in new ones. Based on the accuracy of AS-MON and AS-gamma, transferring previous pheromone trails to new states and considering an independent variable for them, helps achieving more accurate results. In AS-MON, pheromone trails are transferred to new states and added to typical pheromone trails (τ_{ij}). AS-IMG uses immigrant ants to transfer knowledge from previous states to new ones. Although AS-IMG performs better than AS-gamma in the scenario with moderate traffic condition, it cannot perform better in the scenarios with low and high traffic condition. This comparison shows that AS-gamma reaches more accurate solutions than AS-IMG in more complex environments.

Offline performance indicates the average of the best solutions at each time step. When algorithms reach the optimum solution in earlier iterations, their offline performances are reduced. This conveys that AS-gamma converges faster and closer to optimum solutions than AS-IMG, AS-MON and AS-restart in scenarios with low and high traffic condition. In addition, higher offline performance can be due to facing stagnation in the algorithms. By looking at Fig. 4 it can be seen that AS-MON and AS-IMG faces stagnation in these scenarios. The better accuracy of AS-gamma is a result of the *in-memory pheromone* trails effect on generating new populations (4). The pheromone conservation procedure method presented in AS-MON causes stagnation (Fig. 4) which leads to a higher offline performance measures. AS-IMG results are similar to AS-MON algorithm. It cannot prevent stagnation especially in heavy traffic condition. This comparison shows that AS-gamma faces less stagnation and converges faster to optimum solutions due to lower values of offline performance and accuracy. If algorithms face stagnation, they cannot find an alternative for congested routes ahead and as a result, offline performance rises. Also, if they converge slowly, the offline performance rises.

Based on the results reported in Table 4, comparing AS-Gamma with AS-MON, AS-gamma offline performance is better by an average of 1.6% and the accuracy of the solutions by 66.7%. To obtain these numbers, the smallest offline performance of the compared algorithms is divided by the larger number in each scenario. Afterwards, the averages of these numbers are calculated in three scenarios. AS-gamma offline performance is 4.3% better than AS-IMG. Moreover, the accuracy of AS-gamma is 42.9% better than AS-IMG. By comparing AS-Gamma and AS-Restart, it was apparent that AS-Gamma offline performance is better by an average of 6.5% and the accuracy by an average of 67.1%.

Wilcoxon rank sum test with a significance level of 0.05 is applied to check the results in statistic (Table 5). Table 5 demonstrates that although the AS-gamma algorithm performs similar to the AS-MON algorithm in the scenarios with low and moderate traffic condition, it performs significantly better in high traffic condition. AS-gamma also performs better than the AS-IMG in scenarios with low and high traffic condition. AS-IMG performs better than AS-gamma in the scenario with moderate traffic condition. Compared to AS-Restart, AS-gamma performs better in low and high traffic condition and performs similar to AS-Restart in the moderate scenario. Except one of the comparisons, AS-gamma performed better and similar to existing algorithms. It seems that the AS-gamma performs better in facing congestions, and can save more travel time due to the results obtained in the high traffic condition scenario.

Offline performance graphs of the algorithms in different scenarios are drawn in Fig. 4. In all the scenario, the effect of not using the knowledge gained in previous states of the environment can be clearly observed in the AS-Restart algorithm graph. At the beginning of each state, AS-Restart has to start from the beginning and converge to the optimum. In the scenario with low traffic condition, other three algorithms (AS-gamma, AS-MON, and AS-IMG) perform almost similarly. Referring to Table 5, it can be seen that statistically, AS-gamma performs similar to AS-MON and better than AS-IMG in this scenario. In the scenario with moderate traffic condition, AS-gamma performs similar to AS-MON. It can be seen that in this scenario there is

Table 5 Statistical results of AS-gamma, AS-IMG, AS-MON and AS-Restart on three scenarios with different traffic condition over 10 runs

| Algorithm Scenarios | AS-gamma | | AS-MON | | AS-IMG | | AS-Restart | |
|---------------------|--------------|------|--------------|----------|--------------|----------|------------|----------|
| | Ave | Std | Ave | Std | Ave | Std | Ave | Std |
| Low traffic | 17.54 | 0.21 | 17.47 | 0.13 (~) | 17.70 | 0.35 (+) | 18.06 | 0.15 (+) |
| Moderate traffic | 17.71 | 0.35 | 17.86 | 0.06 (~) | 17.40 | 0.17 (-) | 17.95 | 0.24 (~) |
| High traffic | 17.36 | 0.38 | 17.73 | 0.66 (+) | 17.69 | 0.59 (+) | 18.19 | 0.44 (+) |

‘Ave’ and ‘Std’ denotes the average and standard deviation of best solution over 10 runs and 10 states of the scenarios. ‘+’, ‘~’ and ‘-’ indicate that the performance of AS-gamma is significantly better, similar and worse than that of compared algorithms

The numbers in bold are the best results among four algorithms



Fig. 5 Traffic map of two consequent states of the simulated network. a Before b After traffic congestion on one of the streets on the main route c shows the congested street



Fig. 6 Optimal routes obtained from the algorithm a, b AS-gamma c, d AS-MON e, f AS-IMG g, h AS-Restart

a major change in the environment between the first and second states. Because of this change, the diversity of AS-gamma and AS-MON are increased and their graphs rises in the 101st generation. After this event, they converge rapidly to the optimum. The increased diversity after change is not happened in AS-IMG and caused lower offline performance and accuracy in this scenario (Table 4). Referring to Table 5, this result is consistent with the statistical results. In the scenario with high traffic condition, it can be observed that there are several major changes in the environment that caused increase in diversity in algorithms. By looking at the graph of the AS-Restart algorithm, it can be seen that the gradient of the graph of this algorithm is lower than other algorithms and converges more slowly to the optimum. Furthermore, it is unable to find the optimum in several states. The diversity rises after each change for the AS-MON algorithm, but it faces stagnation and converges rapidly to non-optimal solutions. There is only one increase in the diversity of AS-IMG. After this rise, the gradient of the graph is low. The procedure proposed in AS-IMG algorithm, does not help the algorithm to converge faster to the optimum. In addition, it can be observed that, the AS-IMG cannot find the optimal solution in several states of the problem. This happens because AS-IM faces stagnation in this scenario. This causes the higher values of *accuracy* and *offline performance* measures for this algorithm compared to the AS-gamma in Table 4. The diversity is increased after each change in the environment in the AS-gamma algorithm and it converges rapidly to the optimum value after each change. This algorithm is able to reach the optimum solutions in most of the states of the problem. As a result, it has smaller values of *accuracy* and *offline performance* measures compared to other algorithms (Table 4). Statistical results of Table 5 indicate the superiority of the AS-gamma algorithm for complex environments.

The average running time of the algorithms in each state of the transportation network is 12 s for the AS-gamma algorithm, 16 s for the AS-IMG algorithm, 2 s for the AS-MON algorithm, and 20 s for AS-Restart algorithm. The runtime of AS-MON algorithm is shorter than the other algorithms because the solutions are generated and converged faster. It should be noted that the rapid generation of ants and faster convergence caused stagnation and premature convergence in AS-MON. On the other hand, AS-Restart algorithm needed more time to construct ants and to converge to the optimum, because in each state of the transportation network, AS-Restart ignores the prior knowledge of ants. AS-IMG needs some time to transfer immigrant from previous states to new ones. These ants cannot help generating ants faster in early iterations. Therefore, it takes more time for AS-IMG to complete generating ants and to converge to optimums. The AS-gamma algorithm is able to perform better than AS-IMG and AS-Restart algorithms at runtime

and could converge faster to the optimum solutions. Existing *in-memory pheromone* trails helps generating ants faster. However, to avoid stagnation, its execution time was longer than AS-MON algorithm execution time.

Traffic map of the simulated environment is presented in Fig. 5. As it can be observed, one of the streets on the path of the hypothetical vehicle gets congested. In these traffic maps, green is used to represent streets with low traffic condition, orange for moderate conditions, light red for high traffic condition and dark red for congested streets.

The results of each of the methods tested in the experiments are presented in Fig. 6. Optimal routes obtained by AS-gamma are presented in Fig. 6a and b for before and after the congestion. As evident, algorithm finds a detour to avoid the congested street. Figure 6c and d represent the routes obtained by AS-MON before and after the congestion. As the figures show, AS-MON is not able to prevent stagnation and converges prematurely. As a result, AS-MON could not find a detour and prevent the congested street. Figure 6e and f illustrate the solutions of AS-IMG. As it can be seen, AS-IMG finds a detour, which is not the optimal solution to avoid the congested street. Figure 6g and h illustrate the routes obtained by the AS-Restart algorithm. As it was mentioned in previous sections, AS-Restart doesn't utilize the knowledge of ants from the previous states, so algorithm acts independent in each state and it has the least stagnation amongst other algorithms. As shown in the figures, algorithm misses the optimal route before the congestion (Fig. 6e) but due to its independence from previous results, it can find the optimal route after the congestion (Figs. 6f).

5 Conclusion

Dynamic routing is essential in today's crowded and congested cities. One of the best methods to solve dynamic routing problems is to model it as a DOP and solve it by the ant algorithms. Due to stagnation, premature convergence, and low convergence rate in dynamic ant algorithms, researchers are implementing different approaches to improve these parameters. In this paper, we have presented *in-memory pheromone trails* in the framework of AS algorithm to solve dynamic routing problems by keeping the pheromones in the memory of the colony and use them in generation of ants whenever a change occurs. An algorithm based on *in-memory pheromone trails* is developed (AS-gamma) to solve dynamic routing problem. Diversity is increased after a change in a DOP so that the algorithm can prevent stagnation and premature convergence.

By comparing AS-gamma with the traditional AS-Restart, the pheromone conservation procedure introduced by Montemanni (AS-MON), and the immigrants scheme

developed by Mavrovouniotis (AS-IMG), in three different scenarios with low, moderate and high traffic conditions on a real-world transportation network, it can be observed that *in-memory pheromone trail* is effective and the AS-gamma algorithm has a better overall performance than other compared algorithms in solving dynamic routing problem.

This paper has illustrated the suitability of in-memory pheromone trail in solving DVRPs. The presented method in this paper uses the normalized pheromone trails from previous state to create an initial population of ants for optimizing a new state. This procedure transfers knowledge from a previous population to a new one. Results indicate that the AS-gamma faces less stagnation and premature convergence. In addition, convergence rate of AS-gamma was higher compared to AS-Restart. Although convergence rate of AS-gamma is lower than AS-MON and AS-IMG, it prevents facing stagnation and premature convergence. As the research has demonstrated, AS-gamma can be a better solution to the dynamic routing problem in crowded and congested urban networks.

One drawback of this approach is the addition of a parameter to the classic ACO algorithm, which can cause more difficulties and more computation in tuning the parameters. For the future work, it is suggested that AS-gamma be implemented for multi-objective problems.

References

- Abolhoseini S, Sadeghi Niaraki A (2016) Survey on certain and heuristic route finding algorithms in GIS (in Persian). *GEJ* 7(4):49–65
- Abolhoseini S, Sadeghi Niaraki A (2017) Dynamic multi-objective navigation in urban transportation network using ant colony optimization. *Int J Transp Eng (IJTE)* 6(1):49–64
- Aghdam AH, Alesheikh AA (2018) Predicting the future location of cars on urban street network by chaining spatial web services. *IET Intel Transp Syst* 12(8):793–800
- Blokpoel R et al (2016) SUMO 2016—traffic, mobility, and logistics. DLR, Berlin
- Branke J (2012) *Evolutionary optimization in dynamic environments illustrated*. Springer, New York
- Chowdhury S et al (2019) A modified ant colony optimization algorithm to solve a dynamic traveling salesman problem: a case study with drones for wildlife surveillance. *J Comput Des Eng* 6(3):368–386
- Colorni A, Dorigo M, Maniezzo V (1992) *Distributed optimization by ant colonies*. MIT Press, Toward a practice of autonomous systems. In proceedings of the First European Conference on Artificial Life
- de Almeida JPLS, Nakashima RT, Neves-Jr F, Arrudal LVR (2019) Bio-inspired on-line path planner for cooperative exploration of unknown environment by a Multi-Robot System. *Robot Auton Syst* 112:32–48
- Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern B* 26(1):29–41
- Dorigo M, Stützle T (2004) *Ant colony optimization, illustrated edn*. MIT Press, Cambridge
- Dorigo M, Stützle T (2019) Ant colony optimization: overview and recent advances. In: Glover FW, Kochenberger GA (eds) *Handbook of metaheuristics*. Springer, Cham
- Eiben ÁE, Hinterding R, Michalewicz Z (1999) Parameter control in evolutionary algorithms. *IEEE Trans Evol Comput* 3(2):124–141
- Euchi J, Yassine A, Chabchoub AH (2015) The dynamic vehicle routing problem: solution with hybrid metaheuristic approach. *Swarm Evol Comput* 21:41–53
- Felicia A, Lakshmanan L (2016) Accident avoidance and privacy preserving navigation system in vehicular network. *Int J Eng Sci* 6(3):2266–2270
- Gao S et al (2016) Ant colony optimization with clustering for solving the dynamic location routing problem. *Appl Math Comput* 285:149–173
- Garcia MP et al (2009) Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Appl Soft Comput* 9(3):1102–1110
- Hwang I, Jang YJ (2019) Q (λ) learning-based dynamic route guidance algorithm for overhead hoist transport systems in semiconductor fabs. *Int J Prod Res* 58:1–23
- Jin Y, Branke J (2005) Evolutionary optimization in uncertain environments—a survey. *IEEE Trans Evol Comput* 9(3):303–317
- Jin Y et al (2013) A framework for finding robust optimal solutions over time. *Memet Comput* 5(1):3–18
- Li C et al (2008) Benchmark generator for CEC 2009 competition on dynamic optimization
- Lima DA, Tinoco CR, Oliveira GM (2016) A cellular automata model with repulsive pheromone for swarm robotics in surveillance. In *International Conference on Cellular Automata*. Springer, Cham
- Liu J-L (2005) Rank-based ant colony optimization applied to dynamic traveling salesman problems. *Eng Optim* 37(8):831–847
- Manusov V, Matrenin P, Khasanzoda N (2018) Swarm algorithms in dynamic optimization problem of reactive power compensation units control. *Int J Electr Comput Eng* 9:2088–8708
- Mavrovouniotis M, Li C, Yang S (2017a) A survey of swarm intelligence for dynamic optimization: algorithms and applications. *Swarm Evol Comput* 33:1–17
- Mavrovouniotis M, Müller FM, Yang S (2015) An ant colony optimization based memetic algorithm for the dynamic travelling salesman problem. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. ACM
- Mavrovouniotis M, Müller FM, Yang S (2017b) Ant colony optimization with local search for the dynamic travelling salesman problems. *IEEE Trans Cybern* 47(7):1743–1756
- Mavrovouniotis M, Yang S (2011) Memory-based immigrants for ant colony optimization in changing environments. In *European Conference on the Applications of Evolutionary Computation*. Berlin, Heidelberg
- Mavrovouniotis M, Yang S (2012) Ant colony optimization with memory-based immigrants for the dynamic vehicle routing problem. In *2012 IEEE Congress on Evolutionary Computation (CEC)*. IEEE
- Mavrovouniotis M, Yang S (2013) Adapting the pheromone evaporation rate in dynamic routing problems. In *European Conference on the applications of evolutionary computation*. Springer, Berlin, Heidelberg, pp 606–615
- Mavrovouniotis M, Yang S (2013) Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors. *Appl Soft Comput* 13(10):4023–4037
- Mavrovouniotis M, Yang S (2014) Ant colony optimization with self-adaptive evaporation rate in dynamic environments. *IEEE*
- Mavrovouniotis M, Yang S (2014) Interactive and non-interactive hybrid immigrants schemes for ant algorithms in dynamic environments. In *2014 IEEE Congress on evolutionary computation (CEC)*. IEEE

- Mavrouniotis M, Yang S (2015) Ant algorithms with immigrants schemes for the dynamic vehicle routing problem. *Inf Sci* 294:456–477
- Mavrouniotis M, Yang S, Yao X (2014) Multi-colony ant algorithms for the dynamic travelling salesman problem. In *Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*. IEEE
- Montemanni R, Gambardella LM, Rizzoli AE, Donati AV (2005) Ant colony system for a dynamic vehicle routing problem. *Journal Comb Optim* 10(4):327–343
- Morrison R (2003) Performance measurement in dynamic environments. In *Proceedings of the 2003 Genetic and Evolutionary Computation Conference*. Chicago
- Prakasam A, Savarimuthu N (2019) Novel local restart strategies with hyper-populated ant colonies for dynamic optimization problems. *Neural Comput Appl* 31(1):63–76
- Rizzoli A, Montemanni R, Lucibello E (2007) Ant colony optimization for real-world vehicle routing problems. *Swarm Intell* 1(2):135–151
- Schaefer R, Cotta C, Kolodziej J, Rudolph G (2002) Parallel problem solving from nature-PPSN X. *Lect Notes Comput Sci* 6238:64–73
- Sharif M, Sadeghi-Niaraki A (2017) Ubiquitous sensor network simulation and emulation environments: a survey. *J Netw Comput Appl* 93:150
- Singh P et al (2015) Dynamic shortest route finder using pgRouting for emergency management. *Appl Geomat* 7(4):255–262
- Skinderowicz R (2016) Ant colony system with a restart procedure for TSP. In *Proceedings of the International Conference on computational collective intelligence*. Springer, Cham
- Souza NLB, Lima DA (2019) Tabu search for the surveillance task optimization of a robot controlled by two-dimensional stochastic cellular automata ants model. In *2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE)*. IEEE
- Trojanowski K, Michalewicz Z (1999) Searching for optima in non-stationary environments. In *Proceedings of the 1999 Congress on evolutionary computation, CEC 99*. IEEE
- Xiang X, Qiu J, Xiao J, Zhang X (2020) Demand coverage diversity based ant colony optimization for dynamic vehicle routing problems. *Eng Appl Artif Intell* 91:103582
- Xu H, Pu P, Duan F (2018) Dynamic vehicle routing problems with enhanced ant colony optimization. *Discret Dyn Nat Soc* 2018:13
- Yang Z, Emmerich M, Bäck T (2015) Ant based solver for dynamic vehicle routing problem with time windows and multiple priorities. IEEE
- Zhang S, Zhang Y (2018) A hybrid genetic and ant colony algorithm for finding the shortest path in dynamic traffic networks. *Autom Control Comput Sci* 52(1):67–76

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.