



Smart Solution for STSP Semantic Traveling Salesman Problem via Hybrid Ant Colony System with Genetic Algorithm

Eman K. Elsayed^{1*}Asmaa Hekal Omar¹Khadija ElAbd Elsayed¹¹*Faculty of Science, Al-Azhar University (Girls Branch), Cairo, Egypt*

* Corresponding author's Email: emankaran10@azhar.edu.eg

Abstract: Travelling Salesman Problem (TSP) is one of the main and famous problems in finding the shortest path. But life is not ideal, so in this paper, we proposed design a Semantic Travelling Salesman Problems (STSP). The STSP enhances nodes by suitable Ontologies according to a real problem to be more flexible. Also, we proposed a hybrid model (ACSGA) combining the ant colony system (ACS) with the genetic algorithm (GA). That is to speed-up convergence and finds an effective path of a Semantic Travelling Salesman Problem STSP. The performance of the proposed hybrid model ACSGA was faster by 66% than other standard algorithms. And the distance decreases by 0.72%. We apply the proposed solution to the real problem with challenges as traffic overcrowding and weather changeable. So, the proposed STSP in this paper used traffics and weather Ontology as a case study. The proposed method modified the results according to time which decreases by 2.3%.

Keywords: Travelling salesman problem (TSP), Genetic algorithm (GA), Ant colony system (ACS), Ontology.

1. Introduction

Traveling Salesman Problem (TSP) is one of combinatorial optimization and NP-hard problems. It is also one of the optimizations of the most greatly studied problem in recent years.

TSP is to find the shortest possible tour to visits each city once in a list of cities and back to the starting city Travelling Salesman Problem (TSP) and the tour must be closed. Many methods have been used to solve it, such as Nearest Neighbourhood Search (NNS), Tabu Search (TS), Simulated Annealing (SA) [1], Genetic Algorithm (GA) [2], Neural Networks (NN) [3], and Ant Colony System (ACS) [4].

The genetic algorithm is an optimization algorithm based on the evolutionary ideas of natural selection and can search on the range space globally to solve complex combinatorial optimization problems. GA produces the initial population randomly and have the operator's crossover and mutation of genetic produce generation that is also random [5].

But the algorithm gives results with low search efficiency and slow convergence speed of the algorithm.

Ant colony optimization is one of the most successful systems and is the choice of most common researches now. ACO algorithms have attracted the interest of researchers on account of their efficiency, effectiveness, and simplicity in solving complex optimization problems such as TSP [6].

Ant Colony Optimization has an efficient method for the shortest path for traveling and has a low cost.

Artificial researcher Marco Derige described in 1993 a method for producing "best solutions" of TSP in an indicative manner using ACS (ant colony system). Models showed in real ants to find short paths between food sources and their nest, then developed the result from the priority of each ant to follow the pheromones of the other ant trail. In order to improve the ability of the global solving and speed of convergence, avert falling into the ideal local solution. Xue Yang in reference [7] applied the ant colony optimization algorithm on the Traveling Salesman Problem to update the pheromone

information strategy, parameter selection, and optimized local search strategy. The researchers in reference [8] analysed traveling salesman problem with new parameters using Ant Colony Optimization in improved ACO of parameter constants like exponential weight, heuristic exponential weight, and evaporation constant. Although cost/distance travelled remains approximately same the execution time will and the number of iterations to get optimal result reduced safely. Dorian Gaertner et al. in [9] used three parameters in ACS by three optional settings and analyses. They devised a Genetically Modified Ant Colony System (GMACS), which combines GA and ACS by the fitness function, to produce improved solutions. Yet it did not denote relationship ant number with parameter set. Finding the best set of parameters for ACS was an interesting approach and was a task in practice hence the performance of ACS depends on the settings of their parameters that are normally well known.

In reference [10] researchers initialized the parameters of the ACS algorithm (α , β) which is a relatively important parameters in the pathway of identification (ρ); is the pheromone evaporation coefficient. The parameter values ρ needs to change when we apply the algorithm in different state functions.

Kovářík and Skrbek in [11] described an approach that divides an ant colony into groups of ants using different parameter settings. They adapted the number of ants in each group depending on improving the quality of the solution obtained by each group; however, they did not provide the details of the adaptation strategy. The analysis of experimental results for adapting the values of β indicated that better initial performance is obtained with high values of β while lower-end values are preferred to run.

Hao et al. [12] and Cai et al. proposed a variety of ACS for the TSP. In their implementations, a different value of the parameter ρ is associated with each ant depending on the quality of its solution. MAX-MIN Ant SYSTEM (MMAS) is currently one of the best performing ACO algorithms for the TSP. It achieves the best solutions to add pheromone during the pheromone trail update. The use of only one solution for the pheromone update is the most important means of search use in MMAS. By this choice, solution elements that frequently occur in the best-found solutions get large support [13].

Khichane et al. Studied a self-adaptive mechanism for setting α and implementation parameters to apply (MMAS) and its applications [14]. However, differently from previous works, do not specify parameter settings at

the individual ant level for each iteration that specifies a common parameter for the entire ant colony. Hence, we proposed parameters $\alpha = 1, \beta = 10, \rho = 0.5$, to execute MMACS.

Ontology has become very important as we currently lack the shared knowledge-rich in semantics represented in a machine-comprehensible form. Ontology can then be applied to problem-solving, in computer science; Ontology represents knowledge as a set of concepts in a certain domain and relationships between pairs of concepts [15]. It plays a main role in solving the problem of semantic heterogeneity of different data sources and contributes to improving system interoperation [16]. Ontology's installation supplies a semantic to express the semantic information of traffic data, the using of Ontology adds semantics to this information, as a result, information is easily obtained.

Ontology has an advantage which is sharing the common concepts of information structure between people or software agents [17,18].

GA used to solve the Travelling Salesman problem and determine the optimal route on Google map [19].

Lin et al presented an improved hybrid genetic algorithm (HGA) to solve the traveling salesman problem in which the crossover operator is enhanced with a local search. The proposed HGA improve the evolutionary process in terms of the convergence rate and the solution quality to efficiently [20].

In reference [21], the searchers developed the genetic Algorithm for solving the Traveling Salesman Problem to find the optimal route for the Istanbul Electricity Tram and Tunnel Operations (IETT). This method speeded up the process and reduced the time.

Nested Hybrid Discrete Evolutionary Model can achieve better results from independent ACO and GA applications of the problem of improving traffic signal timing by reducing waiting time at each intersection and enhancing on-road mobility by restricting congestion [22]. As a result, the traveling salesman problem is a public problem and faces many difficulties issues related to traffic analysis for transportation decision making. We use a knowledge base of traffic information based on the Ontology which can be used for describing the meaning of traffic data in formalization.

Ontology can be used to expand semantic knowledge of the concepts attached to the traffic domain. This knowledge is expressed in terms of type, classification, constraints, and properties. Ontology not only enhances semantics but also generates a relation between each of these semantics [23, 24].

Although the genetic algorithm can search and solve complex combinatorial optimization problems, when the number of nodes is very big, the best solution cannot be found in finite iteration. That makes GA is an inefficient solution in case of many nodes.

ACS has a restriction on convergence speed for the shortest path. GA does not guarantee the best solution, but it generally provides attainable approximations. GA is a strong global optimization tool and produces the initial population randomly. And direct use of GA in the initial stage may cause falling into local optimal solutions of TSP. Where it results in low search efficiency and slow converging velocity of the algorithm. ACS is an effective and competent algorithm to provide starting solutions. To increase the convergence speed, we are hybridizing ACS with GA [25].

Max-Min Ant System (MMAS) improves performance by updating the Pheromone trail. So; we proposed to take advantage of the ACS and max-min ant system (MMAS) [26] to GA via the ACSGA hybrid model. That is to improve convergence speed and sequence quality on many points in TSP.

In this paper, we ameliorate also the information in each node to be dynamic knowledge. That is via designing traffic Ontology and merging it with nodes; the knowledge-based in traffic Ontology may change according to the real problem.

So, this paper proposes to enhance TSP to be semantic via Ontology (STSP) and hybrid Ant Colony System (ACS) with Genetic Algorithm (GA) to get the shortest path with high accuracy.

The performance of the proposed hybrid model ACSGA was faster than other standard algorithms. the proposed STSP in this paper used traffics and weather Ontology and modified the results according to time.

2. Genetic algorithm

The genetic algorithm (GA) was officially presented in the 1970s by John Holland at the University of Michigan. GA is a kind of evolutionary algorithms, which simulates natural selection [27]. GA uses the operations of selection, crossover, and mutation to generate the next generation, and they search for an optimal solution from a set of points. Moreover, GA is used to optimize parameters to speed up the prediction process [28].

GA has been applied to solve different kinds of difficult optimization problems with the simulation of natural selection and genetic evolution as TSP [27,29].

Although Genetic Algorithms have a fast-global searching ability, when the number of the cities is very big, the search capacity will decrease gradually. Then the best solution cannot be found in finite iteration, which leads to an inefficient solution.

In the following subsections, we will present details of how GA components as a crossover function, a fitness function, and a mutation applied on TSP. That is to clear the proposed hybrid solution between GA and ACS on STSP.

2.1 Fitness function

The fitness function is the essential component of GA which evaluates the chromosomes .and fitness function evaluation is incorporated to assigns a value to each organism, noted as $f(x)$. then compute the fitness value $f(x)$ by using Eq. (1)

$$f(x) = D_{max} - D_i \quad (1)$$

Where D_i is the total Euclidean distance and D_{max} is the Euclidean distance, it should be the longest organisms in the chosen population. Then a fitness function that tells us how good each route is (in our case, how short the distance is). The GAs are used for the maximization problem. For the maximization problem, the fitness function is the same as the objective function. But, for a minimization problem, one way of defining a ‘fitness function is as

$$F(x) = 1/f(x) \quad (2)$$

2.2 Crossover

Crossover operator plays a vital role in GA that creating new offspring from two parents having the properties of both the parents [30]. So many crossover operators have been proposed for the TSP [31].

2.3 Mutation

The mutation serves a significant function in GA, because it helps to avoid local convergence by introducing novel routes that will allow us to explore other parts of the solution space. Almost like a crossover, the TSP features a special consideration when it involves mutation. Mutation would simply mean assigning a minimum probability of a gene changing from 0 to 1, or the other way around.

3. Ant colony optimization (ACO)

Ants put Pheromone, a chemical material when they move on the ground. Their movement is adjusted

by the concentration of Pheromone proportionally. they can find the shortest path from the nest to the food sources. The ACO is successfully implemented in finding good solutions to difficult discrete optimization problems [16,32] such as used to solve the Travelling Salesman Problem (TSP) [33].

Research has shown that ACO performance can be improved by making better use of the best solutions found with an effective mechanism to avoid stagnation in early research and analysis of search space and this achieves the best performance of the ACO algorithm [34].

We can say there are three forms of ACO:

3.1 Ant system (AS)

The ant system (AS) is the first ACO algorithms and was first applied to the TSP. where there are n cities and m ants. Each ant initially exists on a different city node and the pheromone intensity on the edges between every two cities are the same. Although AS was useful for finding good or optimal solutions for small TSP, However, it is not practical for large problems at the required time. Ant colony system [15] and max-min ant system [12] are two models improved AS algorithms.

The ant system (AS) is the first ACO algorithms and was first applied to the TSP. where there are n cities and m ants. Each ant initially exists on a different city node and the pheromone intensity on the edges between every two cities are the same. Although AS was useful for finding good or optimal solutions for small TSP, However, it is not practical for large problems at the required time. Ant colony system [15] and max-min ant system [12] are two models improved AS algorithms.

3.2 Ant colony system (ACS)

Artificial researcher Marco Dorigo described in 1993 a method of heuristically generating "best solutions" to the TSP using a colony called ACS (Ant Colony System) which is a type of ACO, where Ant Colony System (ACS) That's the best efficacy in finding the shortest path [33]. It contains a group of ants in random cities destined to move to other cities. Ants can make local and global updates of the pheromone pathway. Pheromone Path is responsible for choosing the next city to move and every movement creates a path with a stronger Pheromone Path. This process is repeated until an acceptable shorter tour is found, or all ants converge in the same path. Ants move from one place to another according to the move rule. The probability of ant k moves from r to next node s by following transmission rule in Eq. (3):

$$PI_k(r,s) = \begin{cases} \frac{\tau(r,s) \cdot [\eta(r,s)]^\beta}{\sum_{\tau(r,u)} [\eta(r,u)]^\beta} s \in J_k(r), U \in J_k(r), \\ 0 & otherwise \end{cases} \quad (3)$$

$\eta = 1/\delta$ is the inversion of the distance $\delta(r, s)$, τ is the amount pheromone, $J_k(r)$ is the set of not visited node by ant k , the β and α are parameters, β which determines the importance of pheromone versus distance ($\beta > 0$). Ant Colony System (ACS) add another rule in which ants after each step update pheromone locally when selecting their next node. his rule is named Pheromone is locally updated by following rule in Eq. (4):

$$\tau(r, s) \leftarrow (1 - \rho) \cdot \tau(r, s) + \rho \cdot \Delta\tau(r, s), \quad (4)$$

where ρ is a parameter of pheromone evaporation [35], and $0 < \rho < 1$, Then It is assumed to be the acceptable value of $\rho = 0.5$. And $\Delta\tau(r, s) = \tau\theta$ where τ is pheromone. Pheromone is globally updated by following rule in Eq. (5)

$$\tau(r, s) \leftarrow (1 - \alpha) \cdot \tau(r, s) + \alpha \cdot \Delta\tau(r, s), \quad (5)$$

Where α is the relative importance parameter of the pheromone a cumulative in the ant selection path during his movement. We choose the value of $\alpha = 1$ that selected with several trials to achieve the globally shortest path. The value of $\Delta\tau(r, s)$ is determined as the value of $\Delta\tau(r, s)$ is determined According to Eq. (6):

$$\Delta\tau(r, s) = \begin{cases} 1/Lgb & \text{if } (r, s) \in \text{global best tour} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Where, Lgb is the length of the globally best tour.

3.3 Max-Min Ant System (MMAS)

Max-min ant system (MMAS) improves its performance throw Pheromone trail updating, where only one single ant is used to update the pheromone trails after each iteration and by using one single ant for the pheromone trail update. The modified updating rule is given by Eq. (7).

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \Delta t^{best} \quad (7)$$

Δt^{best} was proposed in ACS [19]

where $\Delta t^{best} = 1/f(s^{best})$, and $f(s^{best})$ denotes the solution cost of either the iteration best solution (sib)

or the global best solution (*sgb*). By this choice, the solution elements which much occur in the best-found solutions get large support. A hybrid approach of GA using heuristics was discussed by Jayalakshmi [20] to speed up the convergence process in TSP. The authors used multiple instances of local search which led to slow convergence speed for the solution. To validate our proposed Semantic, Enhance Hybrid Ant Colony Optimization and Genetic Algorithm using Ontology (STSP).

ACS has a restriction on convergence speed for the shortest path, to increase the convergence speed researchers get good results by hybridizing ACS with GA. because GA does not provide the best solution, but it generally provides negotiable approximations in a comparable amount of time.

MAX - MIN Ant colony System (MMACS) is the treatment of updating pheromone pathways. First, only one solution is used to deposit the pheromone. This is usually the best solution for repetition or is the best solution yet, that is the best solution since the starting of the algorithm. Second, all pheromone values are restricted by the bounded by the interval $[\tau_{\min}, \tau_{\max}]$.

The pheromone update base used is Eq. (8):

$$\tau_{ij} \leftarrow \max \{ \tau_{\min}, \min \{ \tau_{\max}, (1-\rho) \cdot \tau_{ij} + \Delta\tau_{ij}^{best} \} \}, \quad (8)$$

where $\Delta\tau_{ij}^{best}$ is defined as

$$\begin{cases} F(s^{best}) & \text{if edge } (i, j) \text{ is part of the} \\ & \text{best solution } s \text{ best,} \end{cases} \quad (9)$$

$F(s^{best})$ is the alternate of the cost of the solution considered for the deposit. To increase the convergence speed researchers, get good results by hybridizing ACS with GA. because GA does not provide the best solution, but it generally provides negotiable approximations in a comparable amount of time. When using MMAS with local search for solving the larger TSP, the best strategy appears to be the use of a dynamical mixed strategy which increases the frequency of using [34]. To validate our proposed Semantic, Enhance Hybrid Ant Colony System and Genetic Algorithm using Ontology (STSP).

4. Semantic travelling salesman problems STSP

To solve TSP, we divided the proposed solution into two main parts:

First, we proposed a hybrid model (ACSGA) between Ant Colony system ACS and Genetic Algorithm GA.

Second, we enhanced TSP by Ontology to be STSP.

Finally, we applied the hybrid model (ACSGA) on (STSP).

4.1 The proposed hybrid ant colony system and genetic algorithm (ACSGA)

ACS and GA have been used in our proposed algorithm. To chooses sequential cities from the path and it order cities according to the minimum distance between the cities by searching for all possible arrangements. The proposed hybrid ACSGA algorithm is displayed, begins with ACS that evaluating the start-up solutions that helped reach the optimum solution more quickly the proposed ACSGA algorithm is shown in Algorithm 1.

Algorithm for Hybrid ACS, GA:

Step1: Initialize parameters of ACS (ants and pheromone values) Iteration $i = 1$.

Step2: Generate the initial distribution of pheromones according to optimization solution:

- a) Calculate the transition probability of each ant, move ant according to probability.
- b) Randomly pick the routes based on the updated pheromone trail.
- c) Update pheromone values that increase for better values and reduce for all others (apply local update).
- d) Store and update ant global best tour

Step3: Choose a good tour from the output (ACS/MAX-MIN) of Step2 and perform the following steps.

- a) Use initial population on the chosen tours and produce population.
- b) Calculate fitness and do Selection.
- c) Apply Crossover operators of 2 tours to produce an offspring.
- d) Apply mutation operators on offspring and calculate the fitness of a new generation (new path).

Step4: if Stopping criteria satisfied that cost new tour < cost global best tour **then goes to step 7**

Step5: Iteration $i = i+1$

Step6: Repeat Step 2 to Step 5 until all the ants converge to one path (shortest path) or an appropriate result achieved

Step7: Exit

The proposed hybrid ACS with GA algorithm shown below of the flowchart is shown in Fig. 1.

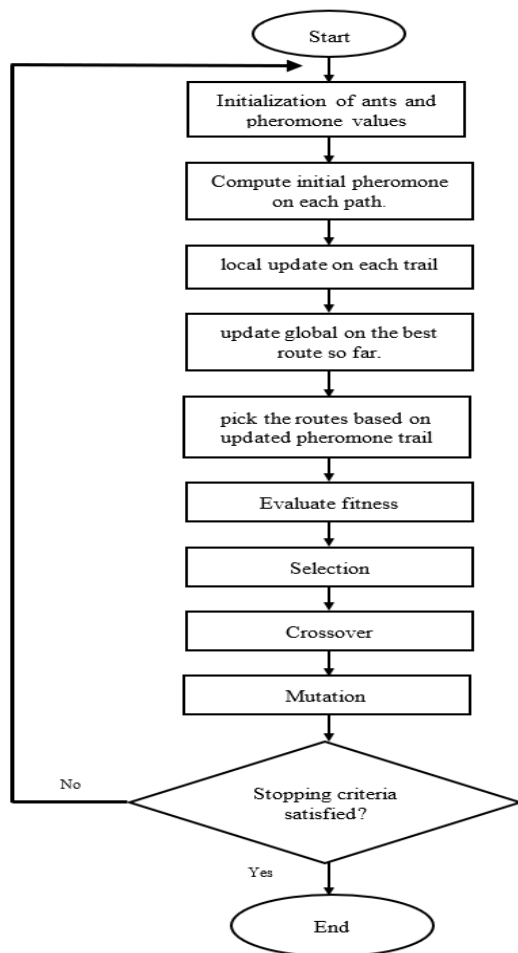


Figure. 1 The flowchart of the proposed model (ACSGA)

In the proposed algorithm all ants are initialized at the starting city and finally return to the same starting city, the traveller is visiting all the cities once in different routes this makes the local update of pheromone on all explored paths and global update of pheromone on the shortest path. The parameters α and β determine the respective weights of the pheromone trail and the distance between two cities.

Next, the evolution of GA begins on the results of ACS. To simulate our “survival of the fittest”, we can make use of Fitness to arrange everyone in the population. Our output will be an ordered list and each associated fitness score starting from the initial population (ACS results), after that the crossover, mutation function of GA. These results lead to more global update causes early discovery of the shortest path and finding the search space locally as well as globally. This algorithm implements until the maximum number of ants or all ants converge in the shortest path. The flowchart is shown in Fig. 1.

Initial pheromone is initialized according to the kind of problem to be solved. In the TSP initial pheromone is supposed as the inverse of the distance between the cities. The ACS is applied with Local

pheromone update to variety in the search space and global pheromone update to early converge on the shortest path. After applying ACS and MMAS, the best paths are chosen based on the length of the path from the available paths, where the shortest path is chosen and then do the process for GA. In the GA module, there are a few options for how to select the parents that will be used to create the next generation. We used roulette wheel selection that is the most common approaches are either fitness proportionate selection to choose chromosomes with more fitness values and perform crossover operation in our implementation. The process stops while the new population is complete, or the maximum number of iterations is performed. The next generation is formed by the selection of larger fitness values.

4.2 The proposed semantic travelling salesman problem (STSP)

Semantic observation helps to plug the obscurity of the natural language when expressing notions and their computational representation in an official language. By telling the computer how the data elements relate and how these relationships can be evaluated automatically, from this it is possible to treat complex processes, and Semantic observations can be used in the decision-making process [5, 8-9]. The semantic representation supports traffic state classification by way of inferred, which provides information that helps the traveller in obtaining the best route and considering time and safety.

In this section, we propose to enhance TSP traveling salesman problem via semantic to be STSP. To explain the idea, we use traffic concepts like time and distance.

So, we create traffic Ontology. That is to provide a traveling salesman with variable information about the traffic occurring in a specific place. That is because the traffic is a useful factor to control TSP.

Ontology is consisting of conceptualization and relationship. Each of these denotes a set of classes and a set of properties. It consists of the name, relationship group, and description of the natural language. The main purpose of building Ontology is to capture the knowledge of urban traffic. We can represent a path that contains one set of attributes for each concept. For example, this may contain attributes like one way or two-way roads, etc. and we represent a set of relationships. We represent a collection of attributes, one for each relationship. Thus, a set of attributes and a set of relationships can be start crossroads and end crossroads. Each axiom in N is a limitation or restriction on the attribute values of a concept trait and relationship or a limitation on

the relationships between concept objects. For example, N could be crossroad must have more than one road. Once you build an Ontology of traffic, it reduces data redundancy and ineffective sharing problems. If the traffic is congested, another road is suggested. The alternate route suggested will present the routes that consist of crowded traffic and normal traffic. The system of traffic will allow the user to pass through the safest and most accurate roads. we use a web mapping service That is being developed by Google. In a show route planning, searching and finding places, etc.

In some research, properties (such as road width that is constant in highways which are the most common roads used by drivers) that have an insignificant effect on the performance of route - finding algorithms are considered; however, in this paper, by defining classes and properties, more important properties such as the x and y coordinates are defined as data properties. Moreover, dividing routes according to distance into long and short routes and dividing routes according to traffic into normal and crowded traffic.

Generally, there are three steps to create the Ontology of TSP (STSP): designing Ontology, saving routes in Ontology, and retrieving routes from Ontology.

In the first step, the schematic view of Ontology is designed using Protégé that contains the name places, And information about routes traffic. Based on proposed hybrid ACSGA that used for route finding, Ontology is created to store the routes of the short distance. This Ontology includes four main classes as shown in Fig. 2.

The classes of the path have subclasses: short and long routes. short routes used to store the routes of TSP. Traffic class includes information about traffic (crowded/normal) traffic from google maps of the routes. traffic class comprises the routes travelled at peak traffic times.

In the step of saving routes in the Ontology, data are classified as relevant classes. The long routes include the routes in which their locations and destinations are not among the routes that are traversed by travelling salesmen. They are obtained from a route - finding algorithm.

In the OWL traffic Ontology of travelling salesman, short routes are defined as classes to improve efficiency and simplicity. The name of places an individual is defined using the ID of the corresponding place. The attributes of the x and y coordinates of these places are defined as the data properties of relevant individuals. Fig. 3 shows part

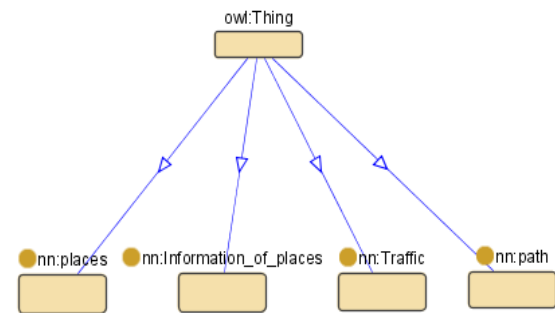


Figure. 2 Traffic ontology

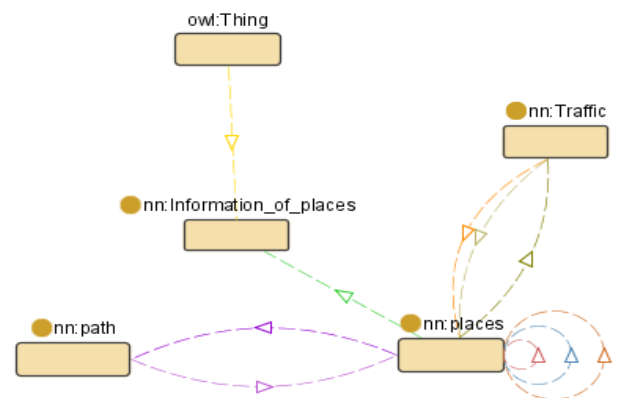


Figure .3 Part of relations of traffic ontology of STSP

of relations for each short route whether as is the route in peak traffic times or non.

If the route related to peak traffic times is crowded. Then it is classified as one of the roads with crowded traffic in the Ontology of STSP.

In the step of retrieving routes from the Ontology of STSP, a traveling salesman specifies the conditions of places by SPARKLE queries around places that want to visit. If the places have normal traffic traveling salesman will visit them firstly, only the short routes are used to create the road. If normal traffic does not exist in the short routes, the long road is used to create the road, but we depend on short distances. If the traffic is congested, another road is suggested.

Finally, we apply the hybrid model (ACSGA) on (STSP). That is convert TSP to be dynamic problem. Where final output is the shortest path in certain time.

To implement the proposed model, we use Google Maps to obtain both traffic and distance between places and locations (latitude and longitude) on the map as destination places of the traveling salesman. After obtaining information about the places intended to be visited through Google maps at different times and storing information in the traffic Ontology.

The main criterion for choosing the best solution is the distance, traffic and time. And with the choice of the shortest path as an ideal solution considers the choice of the least time and the time may vary with the different traffic.

5. Implementation and result discussions

The set of cities are given to the program as inputs and then calculate the distance between each city in matrix form.

The outputs are obtained as iteration in terms of convergence for the shortest path. The iteration in terms of convergence for both simple ACS and the proposed hybrid ACSGA has been studied for the TSP as shown in Fig. 5. For the example we used some towns and cities of “Rijeka Urban Agglomeration” [36] to evaluate the effects of the required number of iterations for the convergence of ants in the shortest path. The edge *i* is considered as the source and destination of the city and initialized with ten ants, the basic ACS and the proposed ACSGA algorithm is executed 10 times respectively. Then use the results that are used to describe the proposed ACSGA requires only one iteration whereas simple ACS requires three iterations to reach the shortest path.

For each iteration the results are shown Fig. 5. In the first iteration in simple ACS 40% of ants chooses the shortest path as compared to hybrid ACSGA where all ants choose the shortest path in the first iteration. In the second iteration of simple ACS only 80% ants choose the shortest path, in the third iteration all ants choose the shortest path and full convergence achieved in the three iteration of the simple ACS. This confirms that ACS has a restriction on convergence speed for the shortest path.

After we obtain the data about each city in terms of its coordinates from “tsplib”, we insert this data into the ACSGA model. The output of this pre-process is a matrix containing the distances between cities. Then the output of ACSGA is the shortest path between cities. We applied ACS and ACSGA on TSP and STSP to evaluate the effects of the required number of times for the convergence of ants in the shortest path.

A simple ACS requires more iterations than the proposed model ACSGA as shown in Fig. 5. To evaluate the proposed model ACSGA, we compared it with the most widely used optimization algorithms (ACS, GA, and simulated annealing (SA)) [37]. The outputs for varying number of cities starting from 10 up to 100 on different algorithms shown in Table 1.

The experimental results of GA gave the best solution in case of a small set of cities (less than 20). But when the number of cities increases to 100, the

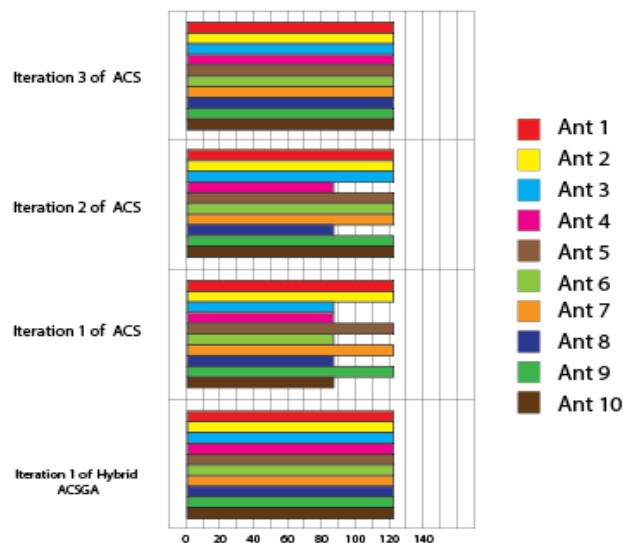


Figure. 5 Displays ant convergence for ACSGA and ACS

Table 1. Displays the tour length outcome of various Algorithms to solve TSP.

No of cities	GA	SA	ACS	ACSGA
10	7.700	7.70043	7.70043	7.70043
20	18.585	18.95	18.56	18.47
30	33.412	25.15	23.79	23.59
40	49.380	29.497	27.25	27.13
50	52.086	31.056	28.28	28.17
60	64.293	33.91	30.07	29.56
70	91.132	35.26	30.79	30.43
80	114.728	42.52	35.57	35.30
90	127.488	47.24	39.12	38.39
100	139.302	50.94	43.11	41.86

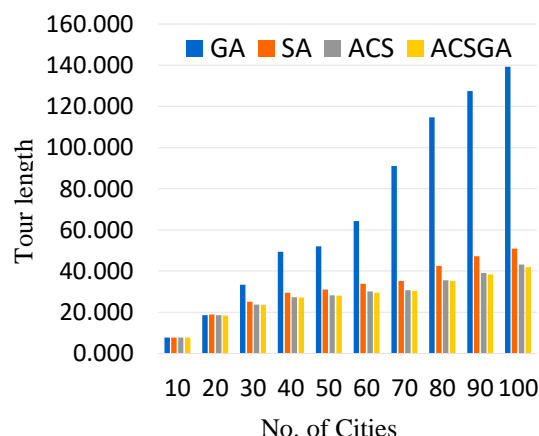


Figure. 6 Comparison between (ACSGA, ACS, SA, and GA) on varying number of cities.

order was (ACSGA, ACS, SA, finally GA) Fig. 6 displays the comparison in the Table 1.

Table 2. Comparison of ACSGA with GA, Hybrid GA, LBSA, GA-MaRI-NICH_LS, ACO-PSO, MN2, STASA, DTSA, ICMPACO), Nested hybrid ACS , LevyACO, Greedy-LevyACO on different TSP standard databases

Problem Name \ Methods	Oliver 30	Eil51	Berlin 52	eil76	eil101
GA	421(423.74)	496	(N/A)	(N/A)	(N/A)
ACS	420 (423.74)	426	(N/A)	560 (562.6)	672.1
Hybrid GA	(N/A)	426 (428.87)	7542 (7544.37)	538 (544.37)	629 (640.975)
LBSA	(N/A)	426	7542	538	629
GA-MaRI-NICH_LS	(N/A)	426	7542	538	629
ACO-PSO	(N/A)	448.55	7663.59	5680.03	700.73
MN2	(N/A)	431.17	7774.24	563.66	664.258
STASA	(N/A)	529.8	7544	554.5	
DTSA	(N/A)	456.5184	7761.6000	588.0623	698.8384
ICMPACO)	(N/A)	429.8817	7548.6		668.23
Nested hybrid ACS	420 (423.74)	426 (429.98)	7542 (7544.37)	538 (545.39)	629 (642.31)
LevyACO	(N/A)	426	7542	538	629
Greedy-LevyACO	(N/A)	426	7542	538	629
Proposed ACSGA	420(423.74)	426(426.87)	7500 (7500.59)	538(545.39)	629(642.03)
Best known solution from TSP Library	420((423.74)	426 (429.98)	7542((7544.37)	538 (545.39)	629 (642.03)
Percentage of decrease	0%	.72%	.58%	0%	.04%
Average			.27%		

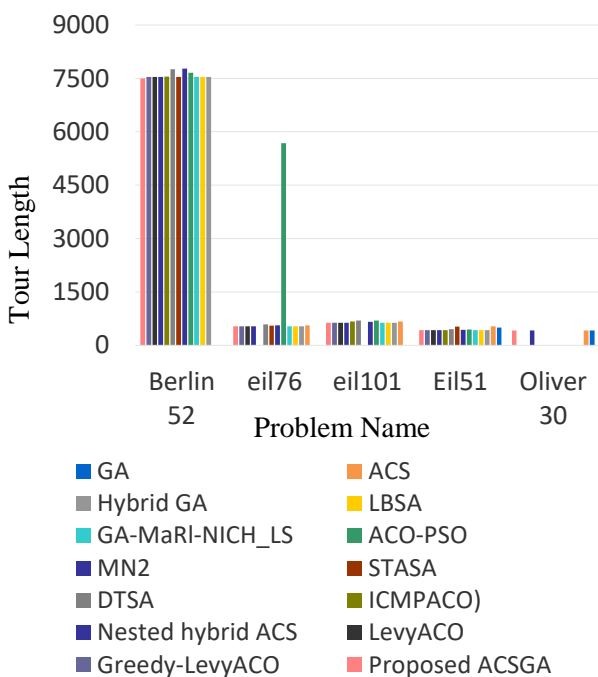


Figure. 7 A comparison of tour length for TSP on different algorithms and ACSGA

Evaluating the performance of ACSGA is by comparing with ACS [38,25], GA [39], Hybrid GA [28], LBSA [40], GA-MaRI-NICH_LS [41], ACO-PSO [42], MN2 [43], STASA [44], DTSA [45], ICMPACO) [46], Nested hybrid ACS [25],

LevyACO [47] and Greedy-LevyACO [48] as shown in the Table 2.

The quality of solutions of proposed ACSGA has been made by calculating the percentage decrease which is defined by (best optimum) = optimum × 100%. Compared to solutions from the best-known solutions of TSPLIB [25, 49-50] For example, TSPLIB data sets were used Shown in the Table 2. A diagram in Fig. 7 displays the comparison of the performance of ACSGA with other algorithms on different standard TSP instances Oliver30, Berlin 52, Eil51, Eil76, and Eil101. The path length achieved by ACSGA is better than the other algorithms.

All the above comparisons proved the high performance of using ACSGA on different TSP databases. But we proposed also enhance the TSP with traffic Ontology to be STSP. In the previous subsection, we will evaluate the performance of STSP.

5.1 Result discussions of STSP

The shortest distance output on TSP is a virtual solution. But in real life, we always face many challenges as weather and traffic. So, we will evaluate applying ACSGA on STSP.

The database we used from the transport network optimization of "Urban agglomeration Rijeka", the standard factor of optimization is the minimum length of the distance travel.

Table 3. Shows the coordinates for towns and cities of Rijeka urban agglomeration

Cities	X	Y
1	45.32704	14.51222
2	45.27464	14.64176
3	45.30716	14.56757
4	45.3376	14.37524
5	45.37567	14.45401
6	45.37156	14.42046
7	45.35688	14.55813
8	45.44641	14.44702
9	45.23921	14.32341

Destination between the nodes (cities, municipalities, distribution centres, ports, stations) as well as road planning. Global positioning of Rijeka It is developing Traffic Direction This means to connect all entities within the transport sector. and the purpose of such connecting is to develop the Rijeka Traffic Direction on the global market [22].

Table 3 shows distances between towns and cities of Rijeka Urban Agglomeration determined by X, Y coordinates.

When we applied ACSGA model directly on data set in Table 3, that has proven his efficacy in solving of standard TSP by Euclidean distance of coordinates we get the total distance 0.88069 for best route (1, 9, 4, 6, 5, 8, 7, 2, 3, 1) with length 126 km. All of that is virtual.

But when use the distance between cities from Table 4 that determined by using Google Maps technology in certain time and using traffic Ontology in STSP.

Where the scenario as the following: We used Google Maps to get information about traffic routes in a certain time to know normal or crowded traffic and insert the real information in the traffic Ontology. Then traffic Ontology classified places to two main classes (crowded and normal).

We got the places that have normal traffic then applied ACSGA on these places to get the shortest path (path 1) without comeback to the start place. Then the Ontology filtered the places that hadn't been visited to visit it at another time. In the second iteration, we applied ACSGA on these places to get the shortest path (path 2) then coming back to the start place in path 1.

From the Table. 5., applying ACSGA on STSP divided path into two paths (crowded/normal path). Although, that led to an increase in the total distance as shown in Fig. 8. That is not only virtual.

Table 4. Displays distances between towns and cities of Rijeka urban agglomeration from google maps

			1	2	3	4	5	6	7	8	9
No	City	Acc	RI	KR	KO	OP	VI	KA	CA	KL	MO
1	Rijeka	RI	0	22	9	14	9	10	9	18	28
2	Kraljevica	KR	22	0	14	37	31	32	17	40	57
3	Kostrena	KO	9	14	0	22	23	24	12	32	43
4	Opatija	OP	14	37	22	0	12	7	22	20	14
5	Viškovo	VI	9	31	23	12	0	5	16	10	26
6	Kastav	KA	11	32	24	7	5	0	18	13	22
7	Čavle	CA	9	17	12	22	16	18	0	22	37
8	Klana	KL	18	40	32	20	10	13	22	0	36
9	Mošćenička Draga	MO	28	57	43	14	26	22	37	36	0

Table 5. Comparison between applying ACSGA on TSP and on STSP

No of places	On TSP	On STSP (Divided in two paths)	Total Dis. In case STSP
10	0.05	.018189+.033673	0.051862
20	0.08	.051861+.044595	0.096456
30	0.12	.079199+.082428	0.161627
40	0.18	.058452+.187103	0.245555
50	0.21	.129147+.194751	0.323898
60	0.23	.120499+.204949	0.325448
70	0.26	.125580+.273731	0.399311
80	0.34	.179863+.364095	0.53958
90	0.46	.197473+.417899	0.615372

To prove the dynamic output in our proposed solution, we applied ACSGA on the cities of Rijeka in the Table. 4. Then we applied the STSP scenario; where, we got the traffic information about the best tour by using google Maps on different days and merged data with traffic Ontology. We calculated the average time for each distance.

Comparison between applying ACSGA on TSP and STSP according to travelling time (min).

That is as shown in Table 7. Where Path1 (9 cities) is shown in Table. 6, the data of Path2 (10 cities) and Path3 (20 cities) is taken from Table 5.

Fig. 9 indicates the travelling time diagrams for corresponding routes of ACSGA (TSP) and (STSP). According to Fig. 9, there are two paths, the travelling time of the (STSP) is shorter than the travelling time observed through ACSGA (TSP); in one path, and the travel times for both methods are similar. Applying ACSGA on STSP divided path into two paths (crowded/normal path) gives path Free from congestion in traffic or from any obstacle to movement, so the speed of reaching the target increases, and since the speed is inversely proportional to time, the higher the speed at reaching the goal, the time decreases. Then we found travelling time of the (STSP) is shorter than ACSGA.

Figure. 8 Comparison between applying ACSGA on TSP and on STSP

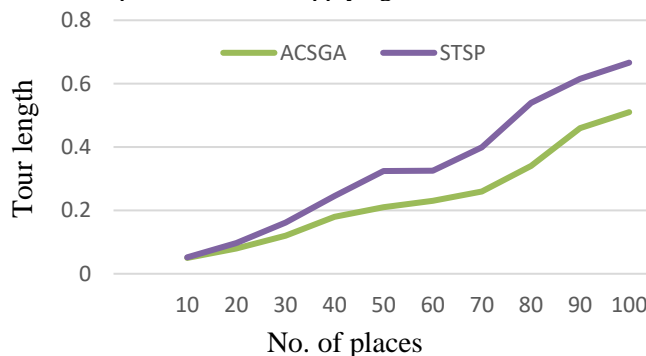


Table 6. Displays the outcomes of applying ACSGA on STSP in different times.

FROM	TO	Dis	Day1 Time (minute)	Day2 Time (minute)	Day3 Time (minute)	Day4 Time (minute)	Day5 Time (minute)	Day6 Time (minute)	Day7 Time (minute)
1	9	28	40	40	46	46	49	49	51
9	4	14	18	20	18	18	18	18	18
4	6	7	10	12	10	10	12	13	13
6	5	5	9	9	9	10	12	12	12
5	8	10	12	12	12	13	12	12	12
8	7	22	35	30	30	35	35	35	35
7	2	17	21	21	21	23	23	23	23
2	3	14	18	18	18	20	20	20	20
3	1	9	17	17	17	17	19	19	21
Total		126	180	179	184	192	200	201	205

Table. 7. Comparison between applying ACSGA and STSP according to travelling time (min)

Paths method of Travel time(min)	Path1 (9cities)	Path2 (10 cities)	Path3 (20cities)
ACSGA	184	101	171
STSP	179	101	164

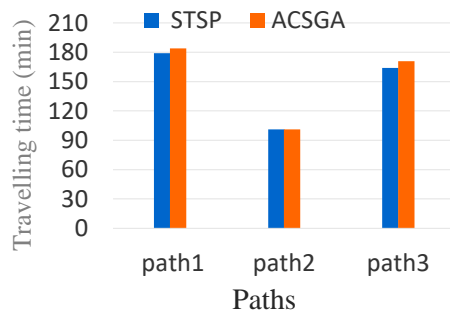


Figure. 9 Optimal travelling time with ACSGA and on STSP

6. Conclusion

In this paper, we proposed a hybrid model between the Ant colony system and the genetic algorithm (ACSGA) by a certain way. That's to solve TSP. We noticed that starting with ACS in the case of hybrid ACSGA be effective to choose and assess the starting solution. ACSGA takes the advantages of the max-min ant system (MMAS) algorithm. We used important Parameters ($\alpha=1$, $\beta=10$, and $\rho=0.5$). We used MMAS to enhance the performance of the ant colony system. this helped to choose the best path. We also used GA to obtain high-quality solutions to optimization problems. That's based on important operators such as crossover, mutation, and selection. the proposed hybrid model ACSGA helped to reach the optimal solution faster 66% than other methods and the distance decreasing by .72%, thus enhancing the results. Also, this paper proposed a semantic enhancement in travelling salesman problem via merging nodes with traffic Ontology or generally any suitable Ontology (STSP). The proposed model ACSGA(STSP) can solve the traffic problems and get the shortest path, and we found time is shorter than Google maps by 2.3%.

In the future, we will enhance Google maps with our proposed model. Also, we will apply STSP in different real cases in the network.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Eman K. Elsayed and Asmaa Hekal Omar, methodology, Eman K. Elsayed, Asmaa Hekal Omar, and Khadija ElAbd Elsayed; software, Khadija ElAbd Elsayed; validation, Eman K. Elsayed, Asmaa Hekal Omar, and Khadija ElAbd Elsayed; formal analysis, Eman K. Elsayed, Asmaa Hekal Omar and Khadija ElAbd Elsayed; investigation, Eman K. Elsayed; resources, Eman K. Elsayed, Asmaa Hekal Omar and Khadija ElAbd Elsayed; data curation, Asmaa Hekal Omar and Khadija ElAbd Elsayed; writing—original draft preparation, Eman K. Elsayed and Asmaa Hekal Omar; writing—review and editing, Eman K. Elsayed; visualization, Eman K. Elsayed and Khadija ElAbd Elsayed supervision, Eman K. Asmaa Hekal Omar, project administration, Eman K. Elsayed, Asmaa Hekal Omar, and Khadija ElAbd Elsayed; funding acquisition, Asmaa Hekal Omar and Khadija ElAbd Elsayed.

References

- [1] Y. Lin, Z. Bian, and X. Liu, "Developing a dynamic neighborhood structure for an adaptive hybrid simulated annealing-tabu search algorithm to solve the symmetrical traveling salesman problem", *Appl. Soft Comput. J.*, Vol. 49, pp 937–52, 2016.
- [2] H. Zhou and M. Song, "An improvement of partheno-genetic algorithm to solve multiple travelling salesmen problem", In: *Proc. of IEEE/ACIS 15th International Conf. on Computer and Information Science (ICIS)*, Okayama, pp. 1-6, 2016.
- [3] K. Leung, H. Jin, and Z. Xu, "An expanding self-organizing neural network for the traveling salesman problem", *Neurocomputing*, Vol. 62, pp. 267-292, 2004.
- [4] Y. Liu, "Solving NP-Hard Problems with Physarum-Based Ant Colony System", in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 14, No. 1, pp. 108-120, 2017.
- [5] E. Alkafaween, "Novel Methods for Enhancing the Performance of Genetic Algorithms", *Mutah University, Mutah*, 2015.
- [6] A. Mohsen, "Annealing Ant Colony Optimization with Mutation Operator for Solving TSP", *Computational Intelligence and Neuroscience*, pp. 1-13, 2016.

- [7] X. Yang and J. Wang, "Application of improved ant colony optimization algorithm on traveling salesman problem", In: *Proc. of Chinese Control and Decision Conference (CCDC)*, Yinchuan, pp. 2156-2160, 2016.
- [8] H. Ali, M. Haris, F. Hadi, S. Ahmadullah, and Y. Shah, "Solving Traveling Salesman Problem through Optimization Techniques Using Genetic Algorithm and Ant Colony Optimization", *Journal of Applied Environmental and Biological Sciences*, Vol. 6, No.4, PP. 55-62, 2016.
- [9] D. Gaertner and K. Clark, "On Optimal Parameters for Ant Colony Optimization Algorithms", In: *Proc. of International Conf. on Artificial Intelligence*, IC-AI, Vol. 1, pp. 83-89, 2005.
- [10] K. Shweta and A. Singh, "An Effect and Analysis of Parameter on Ant Colony Optimization for Solving Travelling Salesman Problem", *International Journal of Computer Science and Mobile Computing*, Vol. 2, pp. 222-229, 2013.
- [11] O. Kovářik and M. Skrbek, "Ant Colony Optimization with Castes", In: Kůrková V., Neruda R., Koutník J. (eds) *Artificial Neural Networks - Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, ICANN, Vol. 5163, 2008.
- [12] Z. Hao, H. Huang, Y. Qin, and R. Cai, "An ACO Algorithm with Adaptive Volatility Rate of Pheromone Trail", *Computational Science-ICCS*, pp. 1167-1170, 2007.
- [13] Z. Cai, H. Huang, Y. Qin, and X. Ma, "Ant colony optimization based on adaptive volatility rate of pheromone trail", *International Journal of Communications, Network and System Sciences*, Vol. 2, No. 8, pp. 792-796, 2009.
- [14] M. Khichane, P. Albert, and C. Solnon, "An ACO-based reactive framework for ant colony optimization: First experiments on constraint satisfaction problems", In: *Proc. of St'utzle T (ed) Learning and Intelligent Optimization, Third International Conf. LION 3, Lecture Notes in Computer Science*, Springer, Heidelberg, Germany, Vol. 5851, pp. 119-133, 2009.
- [15] R. Gupta, R. Gupta, and A. Bishnoi, "ROLES AND USAGES OF RDF, SPARQL, ONTOLOGY TOOLS IN SEMANTIC WEB", *International Journal of Research in Engineering and Technology*, Vol. 3, PP. 86-89, 2014.
- [16] O. Vasilecas, D. Bugaite, and J. Trinkunas, "On Approach for Enterprise Ontology Transformation into Conceptual Model", In: *Proc. of International Conf. On Computer Systems and Technologies*, University of Veliko Tarnovo, Bulgaria, pp. IIIA. 23-1- IIIA. 23-6, 2006.
- [17] T. Gruber, "A translation approach to portable ontology specifications", *Knowledge Acquisition*, Vol. 5, No. 2, pp.199-220, 1993.
- [18] M. Musen, "Dimensions of knowledge sharing and reuse", *Computers and Biomedical Research*, Vol. 25, No. 5, pp. 435-467, 1992.
- [19] L. Helshani, "An android application for google map navigation system, solving the travelling salesman problem. Optimization throught genetic algorithm", In: *Proc. of FIKUSZ '15 Symposium for Young Researchers*, 2015.
- [20] B. Lin, X. Sun, and S. Salous, "Solving travelling salesman problem with an improved hybrid genetic algorithm", *Journal of Computer and Communications*, Vol. 4, pp. 98-106, 2016.
- [21] U. Hacizade and I. Kaya, "GA Based Traveling Salesman Problem Solution and its Application to Transport Routes Optimization", *IFAC-Papers OnLine*, Vol. 51, No. 30, pp. 620-625, 2018.
- [22] S. Srivastava and S. Sahana, "Nested hybrid evolutionary model for traffic signal optimization", *Appl Intell*, Vol. 46, No. 1, pp. 113-123, 2017.
- [23] T. Feng, H. Liang, and W. Jianping, "Ontology Driven Data Sharing Between Microscopic Traffic Simulation and GIS", In: *Proc. of IEEE International Conf. on Computer Simulation and GIS*, Vol. 3, pp. 84-89, 2010.
- [24] E. Elsayed and M. Aly, "Hybrid between Ontology and Quantum Particle Swarm Optimization for Segmenting Noisy Plant Disease Image", *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 5, pp. 299-311, 2019.
- [25] S. Sahana, "Hybrid optimizer for the travelling salesman problem", *Evolutionary Intelligence*, Vol. 12, No. 2, pp. 179-188, 2019.
- [26] J. Yu and C. Wang, "A max-min ant colony system for assembly sequence planning", *Int J Adv Manuf Technol*, Vol. 67, pp. 2819-2835, 2013.
- [27] M. Jafer, M. Khan, S. Rehman, and T. Zia, "Optimizing Broadcasting Scheme for VANETs Using Genetic Algorithm", In: *Proc. of IEEE 41st Conf. on Local Computer Networks Workshops (LCN Workshops)*, 2016.
- [28] Y. Ding, Y. Cai, P. Sun, and B. Chen, "The Use of Combined Neural Networks and Genetic Algorithms for Prediction of River Water

- Quality”, *Journal of Applied Research and Technology*, Vol. 12, No. 3, pp. 493–499, 2014.
- [29] Z. Zheng, Z. Liu, X. Liu, and P. Du, “Genetic algorithm-based image preprocessing for volume rendering optimization”, *IEEE International Symposium on IT in Medicine and Education, Jinan*, pp. 389-393, 2009.
- [30] C. Arunkumar, M. Sooraj, and S. Ramakrishnan, “Finding expressed genes using genetic algorithm and extreme learning machines”, In: *Proc. of 4th International Conf. on Advanced Computing and Communication Systems (ICACCS), Coimbatore*, pp. 1-4, 2017.
- [31] Z. Ahmed, “Genetic Algorithm for the Traveling Salesman Problem Using Sequential Constructive Crossover Operator”, *Journal of Biometrics & Bioinformatics (IJBB)*, Vol. 3, No. 6, pp. 96-105, 2010.
- [32] P. Kora and P. Yadlapalli, “Crossover Operators in Genetic Algorithms: A Review”, *International Journal of Computer Applications*, Vol. 162, No. 10, pp. 34–36, 2017.
- [33] M. Dorigo and L. M. Gambardella, “Ant colony system: a cooperative learning approach to the traveling salesman problem”, in *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 53-66, April 1997.
- [34] T. Stutzle and H. Hoos, “Max–min ant system. Future generation computer systems”, Vol. 16, No. 8, pp.889–914, 2000.
- [35] S. Arora, “Polynomial time approximation schemes for Euclidean TSP and other geometric problems”, In: *Proc. of 37th Conf. on Foundations of Computer Science, Burlington, VT, USA*, pp. 2-11, 1996.
- [36] S. Vukmirović, Z. Čapko, and A. Babić, “The Exhaustive Search Algorithm in the Transport network optimization on the example of Urban Agglomeration Rijeka”, *42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia*, pp. 888-895, 2019.
- [37] F. Valdez, F. Moreno, and P. Melin, “A Comparison of ACO, GA and SA for Solving the TSP Problem”, In: *Castillo O., Melin P. (eds) Hybrid Intelligent Systems in Control, Pattern Recognition and Medicine. Studies in Computational Intelligence*, Springer, Cham, Vol. 827, 2020.
- [38] S. Sahana and A. Jain, “High Performance Ant Colony Optimizer (HPACO) for Travelling Salesman Problem (TSP)”, *Lecture Notes in Computer Science*, pp. 165-172, 2014.
- [39] J. Renders and H. Bersini, “Hybridizing genetic algorithms with hill-climbing methods for global optimization: two possible ways”, In: *Proc. of the First IEEE Conf. On Evolutionary Computation. IEEE World Congress on Computational Intelligence*, Orlando, FL, Vol. 1, pp. 312-317, 1994.
- [40] S. Zhan, J. Lin, Z. Zhang, and Y. Zhong, “List-Based Simulated Annealing Algorithm for Traveling Salesman Problem”, *Computational Intelligence and Neuroscience*, pp. 1-12, 2016.
- [41] M. Alipour, S. Razavi, M. Derakhshi, and M. Balafar, “A hybrid algorithm using a genetic algorithm and multiagent reinforcement learning heuristic to solve the traveling salesman problem”, *Neural Computing and Applications*, Vol. 30, No. 9, pp.2935-2951, 2017.
- [42] H. Qian and T. Su, “Hybrid algorithm based on max and min ant system and particle swarm optimization for solving TSP problem”, In: *Proc. of 33rd Youth Academic Annual Conf. of Chinese Association of Automation (YAC), Nanjing*, pp. 683-687, 2018.
- [43] I. Ahmia and M. Aïder, “A novel metaheuristic optimization algorithm: The monarchy metaheuristic”, *Turkish Journal of Electrical Engineering and Computer Sciences*, Vol. 27, No. 1, pp. 362-376, 2019.
- [44] X. Han, Y. Dong, L. Yue, and Q. Xu, “State Transition Simulated Annealing Algorithm for Discrete-Continuous Optimization Problems”, in *IEEE Access*, Vol. 7, pp. 44391-44403, 2019.
- [45] A. Cinar, S. Korkmaz, and M. Kiran, “A discrete tree-seed algorithm for solving symmetric traveling salesman problem”, *Engineering Science and Technology, an International Journal*, 2019.
- [46] W. Deng, J. Xu, and H. Zhao, “An Improved Ant Colony Optimization Algorithm Based on Hybrid Strategies for Scheduling Problem”, in *IEEE Access*, Vol. 7, pp. 20281-20292, 2019.
- [47] Y. Liu and B. Cao, “A Novel Ant Colony Optimization Algorithm with Levy Flight”, in *IEEE Access*, Vol. 8, pp. 67205-67213, 2020.
- [48] Y. Liu, B. Cao and H. Li, “Improving ant colony optimization algorithm with epsilon greedy and Levy flight”, *Complex Intell. Syst.*, 2020.
- [49] K. Shweta and A. Singh, “An Effect and Analysis of Parameter on Ant Colony Optimization for Solving Travelling Salesman Problem”, Vol. 2, pp. 222-229, 2013.
- [50] G. Reinelt, “TSPLIB-A traveling salesman problem library”, *ORSA J. Comput.*, Vol. 3, No. 4, pp. 376-384, 1991.