

## Master-Slave Asynchronous Evolutionary Hybrid Algorithm and its Application in VANETs Routing Optimization

Said Mohamed Said

*Information Engineering Department*

*University of the Ryukyus*

*1 Senbaru, Nishihara, Okinawa 903-0213 JAPAN*

*saidy87@hotmail.com*

Morikazu Nakamura

*Information Engineering Department*

*University of the Ryukyus*

*1 Senbaru, Nishihara, Okinawa 903-0213 JAPAN*

*morikazu@ie.u-ryukyu.ac.jp*

**Abstract**—Hybrid algorithms incorporated with parallel processing techniques are very powerful tools for efficiently solving very complex optimization problems. We present asynchronous parallel computer architecture adaptation based on hybridization of Genetic Algorithms (GAs) and Estimation of Distribution Algorithms (EDAs). In this master-slave formulation, slaves perform evolutionary computation independently using GAs, while master supervises and controls the searching process. Master's role is to probabilistically study the characteristics of solution space and directs the slaves on good searching spots. This study reports some few findings on the ability of our hybrid algorithm to solve some instances of BQP problem as well as AODV routing optimization in VANETs. For both problems our hybrid algorithm has obtained best results in terms of quality of solutions as well as computational speed.

**Keywords**—Hybrid algorithms, Estimation of Distribution Algorithm, Genetic Algorithms, Synchronous algorithms, Asynchronous algorithms, Parallel processing, Master-Slave, K-means clustering, BQP, VANETs, AODV, NS-2;

### I. INTRODUCTION

An enormous increase in number of vehicles in our cities in recent years has not only made driving as challenging and dangerous but also increased daily drive-thru communication demands as we spend more time on roads. In dealing with this issue, leading car manufacturers and government agencies decided to work on projects which aim at helping drivers to anticipate hazardous events or enable them to reduce traffic. With such efforts Wireless Access for Vehicular Environment (WAVE) was established which can provide vehicle-to-vehicle and vehicle-to-roadside wireless communication. Different bodies such as (C2CCC, VII, CALM) and projects (VICS [1], CarTALK 2000 [2], NOW, CarNet [3], etc.) involved in this field with objectives of improving road safety, providing traffic management solutions as well as on board entertainment applications and connectivity.

WAVE equipped vehicles, roadside infrastructures and personal road user's devices (e.g. smartphones) form a volatile type of network called Vehicular Ad-hoc Networks (VANETs). VANETs are mainly characterised by their constrained but highly variable topology due to high mobility of the nodes and by the WiFi coverage limitations that depends on the surrounding conditions (buildings, other vehicles, etc.). For this reason, and the fact that there is no any central manager entity, routing in VANETs

becomes critical and very difficult task. Therefore many researchers are working on whether to create new protocols for VANETs [4] or improving the existing ones [5], [6].

One of the possible attempts to improve performance of these routing protocols is by setting the best appropriate configuration of the parameters that are used in several operations. However, the attempt is not an easy task, due to the fact that the number of possible configurations is quite large (ie, many parameters are involved and with large solution space).

In modern computing metaheuristics have been used with great success to solve several optimisation problems [5]. They have been similarly involved in VANETs to find optimal parameter configuration for routing protocols. Several researchers have recently been involved in this particular type of optimization [7],[8],[9],[10]. The outcome has also been of great success in terms of performance improvement although limited on how fitness evaluations are performed. In VANETs the employed algorithm has to evaluate the fitness of proposed solutions by performing analysis of Quality of Service (QoS) with possible parameter configurations using simulations. Since these simulations take considerable amount of time to finish, metaheuristics have to consider to scale down VANETs simulations with reduced number of iterative steps and limited population sizes.

Some routing protocols are mentioned in [11] and [12] that have either been specifically designed or used for VANETs routing. The most common and basic being Ad-hoc on Demand Distance Vector (AODV) and Optimized Link State Routing (OLSR) protocols.

In this particular study, we have used asynchronous parallel hybrid approach of GAs and EDAs to solve a well known BQP optimization problem, and employ the algorithm to search for optimal configuration of AODV routing protocol for VANETs. The method uses the same master-slave approach with strategies to control and explore the solution space as explained in our previous work in [13]. The algorithm is set for both VANETs simulations and QoS fitness evaluations to be performed in parallel to increase algorithm's computational speed.

Among the algorithms compared in experiments, our algorithm has shown not only the highest capability to solve BQP problems and improve QoS of AODV rout-

ing parameters, but also managed to cut off the overall computation time significantly.

## II. BQP OPTIMIZATION

The Binary Quadratic Programming is a problem of maximising quadratic objective by choosing suitable binary variables. Mathematically it can be represented as;

$$\text{maximize} \rightarrow f(x) = \sum_{i=1}^n \sum_{j=1}^n [q_{ij}] x_i x_j \quad (1)$$

$$\text{s.t} \rightarrow x_i \in [0, 1], i = 1, \dots, n \quad (2)$$

Where by;  $n$  is the problem size,  $[q_{ij}]$  is  $n$  by  $n$  symmetric matrix.

$$x_i = \begin{cases} 1, & \text{if variable } i = 1, \dots, n \text{ is chosen} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

The BQP is known to be NP-hard and has a large number of applications, for example in capital budgeting and financial analysis problems, CAD problems, traffic message management problems, machine scheduling and molecular conformation [14].

## III. VEHICULAR AD-HOC NETWORKS (VANETs)

VANET is a digital communication between vehicles, road side units such as antennas or road user's hand held devices. Every communicating node in a VANET serves as a wireless router to allow connectivity of approximately 100 to 300 meters distance, resulting into a wide range network. The topology of this network is undefined since the vehicles are very mobile and nodes can join or leave the network in an unspecified rate and fashion. Several potential VANETs applications which do exist are mainly for road safety, traffic control as well and on board entertainment.

To perform optimization on routing protocol used in VANETs, we need to set proper values of parameters that govern the operation of the routing protocol. The aim is to try to reach the AODV configurations that will eventually improve overall network performance in VANETs better than or similar to found parameterizations in a reasonable amount of computation time, taking as search space all feasible AODV settings. Hence we use the same values of solution vectors as shown in Table I. In this work we use our optimization algorithm to evaluate the fitness of proposed solutions by performing analysis of QoS with possible AODV parameter configurations. We transform the definition of QoS by considering three commonly used AODV performance metrics to evaluate tentative AODV configurations.

- i *Packet Delivery Ratio (PDR)*. Ratio of total data packets delivered at destinations to those sent from the sources.
- ii *Normalized Routing Load (NRL)*. The number of routing packets transmitted (including forwarded packets) per data packets delivered at the destinations.

Table I  
AODV PARAMETERS AND THE STANDARD RFC 3561 VALUES

Parameter	RFC value	Data Type	Range
HELLO_INTERVAL	1.0 s	$\mathbb{R}$	[1.0, 20.0]
ACTIVE_ROUTE_TIMEOUT	3.0 s	$\mathbb{R}$	[1.0, 20.0]
MY_ROUTE_TIMEOUT	6.0 s	$\mathbb{R}$	[1.0, 40.0]
NODE_TRAVERSAL_TIME	0.04 s	$\mathbb{R}$	[0.01, 15.0]
MAX_RREQ_TIMEOUT	10.0 s	$\mathbb{R}$	[1.0, 100.0]
NET_DIAMETER	35	$\mathbb{Z}$	[3, 100]
ALLOWED_HELLO_LOSS	2	$\mathbb{Z}$	[0, 20]
REQ_RETRIES	2	$\mathbb{Z}$	[0, 20]
TTL_START	1	$\mathbb{Z}$	[1, 40]
TTL_INCREMENT	2	$\mathbb{Z}$	[1, 20]
TTL_THRESHOLD	7	$\mathbb{Z}$	[1, 60]

- iii *End-to-End Delay (E2ED)*. Refers to the time taken for a packet to be transmitted across a network from source to destination.

Hence the objective of our optimization algorithm is searching for optimal AODV parameter values that will improve performance of VANETs by minimising *NRL* and average *E2ED*, while maximising *PDR*.

The fitness function inherited from the research in [7] is a minimization function shown in equation 4 below.

$$f = (-PDR.w_1) + (NRL.w_2) + (E2ED.w_3) + \Delta \quad (4)$$

Constants  $w_1, w_2$  and  $w_3$  valued as 0.6, 0.35 and 0.25 respectively are the weights corresponding to *PDR*, *NRL* and *E2ED* contributions. In the equation, normalizing offset  $\Delta$  is used to keep the fitness value in the interval [0, 1].

## IV. MASTER-SLAVE ASYNCHRONOUS EVOLUTIONARY HYBRID ALGORITHM

Master-slave architecture is used to accommodate a strategic asynchronous hybrid algorithm. The main searching algorithm used is isolated island model parallel GAs executed by slaves. The searching is under the supervision of master process which direct slaves on how to initialise their population members. The master performs this task by employing the well known EDA algorithm that mainly uses information contained in past generation solutions using probabilistic estimation methods to learn characteristics of solution space under consideration.

In every phase slaves return their optimal solutions to the master's database (*DB*) and master classifies these solutions into different groups using  $K$  means clustering algorithm. From within the clusters of solutions, master estimates probability distribution using special formula to form probabilistic vectors. This role of master to iteratively formulate probability distribution vectors is an EDA part of our algorithm. On receiving the vectors from master, the slaves use them to sample initial population in their next searching.

For effective search space exploration, we have characteristically defined four phases of searching that are strategically arranged. The phases are defined by characteristics of probabilistic vectors returned by slaves. Serial arrangement of these phases in top down fashion (Strategy) is also very important aspect to determine quality of global optimal solution as well as overall computation

time. Suppose  $\mu_j$  is probabilistic vector defined by master for cluster  $j$ , the following explanation briefly explains four phases;

- **Wide-Range Search:** The first phase of searching where master allow the slaves to randomly initialise their solutions. Its purpose is to include a much wider region of solution space.
- **Outside Clusters Search:** After clustering, the vectors  $\mu_j$  are formed for every cluster  $j$ , but their complements  $\mu_j^c$  are sent to slaves. The aim here is to cover the regions not included in the clusters to avoid immature convergence.
- **Cumulative Clustering:** Here more solutions are added to the  $DB$  by slaves as searching continues. And for every attempt (iteration), new vectors are formed by clustering and estimation. Now slaves are given the real  $\mu_j$  to rely on their initialisations. Different regions of search space are explored.
- **Focusing:** Here the most effective area of solution space is determined and all slaves are given the same  $\mu_j$  to concentrate on their final search.

Figures 1 and 2 respectively show the pseudo-codes for slaves and master  $P$  is GA population and  $T$  is master's strategy. For BQP optimization, local search is performed between step 3 and 4 of Figure 1 to enhance the GA capability.

```

1: for  $i = 0 \dots \text{MAXGENERATIONS}$  do
2:   Recieve( $\mu_j$ ) from Master;
3:    $P \leftarrow \text{GenerateInitialPopulation}(\mu_j)$ ;
4:   Evaluate( $P$ );
5:   while termination condition is not met do
6:      $P' \leftarrow \text{Recombine}(P)$ ;
7:      $P'' \leftarrow \text{Mutate}(P')$ ;
8:     Evaluate( $P''$ );
9:      $P \leftarrow \text{Select}(P, P'')$ ;
10:  end while
11:  Send the best solutions to Master;
12: end for

```

Figure 1. Pseudo code for Slaves

In order to reduce the waiting times between parallel processes, the algorithm executes all processes in an asynchronous fashion. That is slaves start and finish their searching in every iteration independently while master does not wait for all slaves to finish in order to start estimation process. The details of our asynchronous algorithm are well explained in [13].

#### A. EDA Probability Vector Estimation

Many EDA probabilistic estimation models are explained in [15]. In our work in [16], the simplest UMDAC algorithm was used for probabilistic vectors estimation. After clustering, the master calculates the mean probabilistic vector for each cluster using UMDAC as the Gaussian Mixture Model without variables dependency.

```

1: Initialize( $DB$ );
2: for  $j = 0 \dots k - 1$  do
3:    $T \leftarrow \text{WRS}$ ;
4:    $\mu_j \leftarrow \text{GenerateProbabilityVector}(T, DB)$ ;
5:   Send  $\mu_j$  to Slave  $j$ ;
6: end for
7: while termination condition is not met do
8:   ReceiveSolutionFromSlaves( $DB, j$ );
9:   if EnoughSolutions then
10:    EstimationOfDistribution( $DB$ );
11:     $T \leftarrow \text{Strategy}()$ ;
12:    Perform parallel  $K$  means clustering( $DB$ );
13:     $\mu_j \leftarrow \text{GenerateProbabilityVector}(T, DB)$ ;
14:    Send  $\mu_j$  to Slave  $j$ ;
15:   end if
16: end while

```

Figure 2. Pseudo code for Master

Let us denote the mean vector by;

$$\mu_j^l = (\mu_{1,j}^l, \mu_{2,j}^l, \dots, \mu_{n,j}^l), \quad (5)$$

$$\mu_{i,j}^l = \frac{1}{m_j^l} \sum_{y=1}^{m_j^l} x_{i,j}^{l,y}, \quad (6)$$

Where  $m_j^l$  is the total number of solutions in the  $j$ -th cluster and  $x_{i,j}^{l,y}$  is the  $i$ -th component of the  $y$ -th solution in the  $j$ -th cluster at the  $l$ -th iteration.

We proposed the improvements of above equation in [13], by introducing information about best and worst solutions in calculating the vectors inspired from machine learning algorithms. Hence  $\hat{\mu}_j^l$  in equation 7 below is used as our estimation vector instead of  $\mu_j^l$  equation 6.

$$\hat{\mu}_{i,j}^l = (1 - \alpha) \cdot \mu_{i,j}^l + \alpha \cdot (X^{best,1} + X^{best,2} - X^{worst}), \quad (7)$$

Where  $\alpha$  is known as learning rate,  $X^{best,1}$  and  $X^{best,2}$  are cluster's best and second best solutions respectively, while  $X^{worst}$  is cluster's worst solution.

#### B. Implementation in VANETs Optimization

Figure 3 shows diagrammatically the slaves part of our algorithm implementing parallel GAs with parallel chromosome's fitness evaluations through simulations. Like standard parallel GA (pGA) in each iteration, every slave starts by generating initial solutions (it can be either random initialisation or by sampling using probabilistic vectors, depending on searching phase).

Very crucial and difficult part as far as VANETs QoS optimization is concerned is during the chromosome's fitness evaluation, since it involves time consuming VANETs simulations. In every slave, the simulations are performed in parallel using NS-2 network simulator. For every simulation, the corresponding trace file is generated which contains all network information necessary for network QoS metrics calculations.

The awk scripts with necessary network QoS definitions coded to operate specifically on these trace files are executed in parallel for every slave to calculate QoS metrics for every chromosome. Upon completion of the above explained long process, every slave proceeds with the remaining steps of GAs as explained by pseudocode in Figure 1.

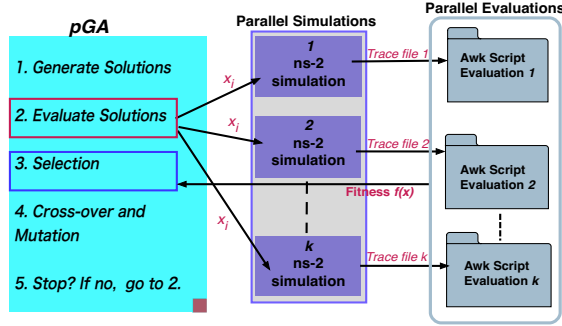


Figure 3. Slaves Implementation in VANETs

## V. PARALLEL PROCESSING

In this research have used shared memory parallel processing technique using multi-core processing unit. The used hybrid approach includes parallelism in both master and slaves.

Master's search space control is initialized by dynamic  $K$  means clustering explained in [17] to optimize the results of clustering algorithm by increasing inter-cluster distance and decreasing intra-cluster distance. Other approaches for better clustering results are well explained in [18][19]. Therefore we have performed clustering using different  $K$  values, each running in an independent parallel process. Only  $K$  value with optimal results was used.

All slaves need to initialize their population for GAs at the beginning of every iteration. Under the guidance of master process, the searching is performed independently in parallel by all slaves.

## VI. EXPERIMENTAL EVALUATION

The implementation of our algorithm was done using C programming language. POSIX Threads (Pthreads) library was used for parallelization.

Experiments were done using Mac Pro 5.1 computer with  $2 \times 2.66\text{GHz}$  6-core Intel Xeon processors and 12GB of main memory. For both BQP and VANET, every slave initializes GA's population only once in each phase during these experiments, hence we had 4 iterations for one execution.

### A. BQP Experiments

Comparison experiments for BQP involved two algorithms;

- Parallel Genetic Algorithm (pGA)
- Our new Enhanced Asynchronous Hybrid Approach of GA and EDA (eAH) used in [13].

1) *Parameter Settings*: 12 different sets of BQP instances, four for each of sizes 100, 500 and 1000 were used in experiments. For both algorithms, we set number of parallel threads as four, with one thread doing both master and slave tasks.

Each slave's GA was running population size of 60 and iterated 30, 30, and 40 generations for problems of sizes 100, 500 and 1000 respectively. The GA's crossover and mutation probabilities were set as 1 and 0.03 respectively.

Each algorithm was given fairly equal number of evaluations ( $nEval$ ) to solve problems which is calculated according to the following equation;

$$nEval = nThreads \times Popsiz e \times Gen \times Iter \quad (8)$$

Where  $nThreads$  is number of parallel threads (slaves) used in algorithms,  $Gen$  is number of GA generations and  $Iter$  is number of times population members are being initialised (iterations).

2) *Results*: The experimental results obtained from two algorithms stated above are presented and analysed in this section. Since both algorithms were able to reach target values in all problem sets used, we present the comparison results based on percentage of successful evaluations ( $pSucc$ ) for each algorithm given by the following formula;

$$pSucc = \frac{\text{number of successful evaluations}}{\text{Total number of evaluations (nEval)}} \times 100 \quad (9)$$

Figures 4 and 5 confirm the outstanding capability of eAH over pGA in terms of obtaining more evaluations which could reach the target values. This is because, pGA was not strong enough to cover very large search space frequently, even with the help of Local Search. eAH was able to locate the target solutions more frequently even when the problem size was large.

In Figure 6, average algorithms computation time (in seconds) is compared. The time was calculated by considering the average computation time in 10 executions for each problem instance. The observation shows that even with small size problem instances, eAH used smaller amount of time to finish its executions. The difference becomes increasingly large from 500 towards 1000 problem sizes.

### B. VANETs Experiments

The experiments performed in this portion of our work, focused on performance comparison of two algorithms;

- Our Asynchronous Hybrid Approach of GAs and EDA (AH) used in [20].
- Our new Enhanced Asynchronous Hybrid Approach of GAs and EDA (eAH) used in [13].

We conducted our experiments using real VANETs simulation scenario which covers a  $582m \times 682m$  of real area of Malaga. Scenario and movement pattern files were downloaded from [21]. The scenario considered road traffic density as 45 vehicles and constant bit rates (CBR) as 64kbps. Table II presents the network propagation model

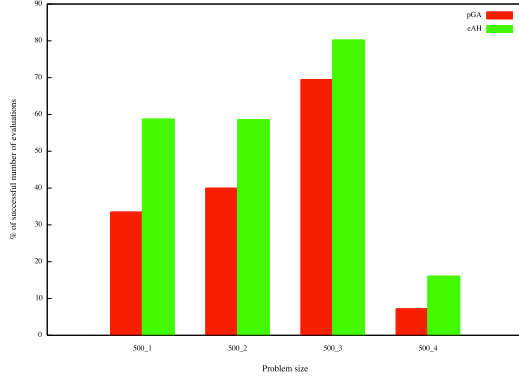


Figure 4. Percentage of successful evaluations in problem instances of size 500

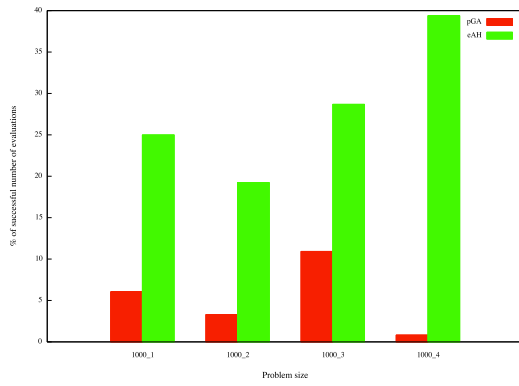


Figure 5. Percentage of successful evaluations in problem instances of size 1000

and network layers used in simulations. All Network Simulations were done using NS-2 simulator, implementing AODV routing protocol.

1) *Parameter Settings:* For GAs, probability of crossover and mutation used was 1 and 0.1 respectively. Due to time consuming nature of VANETs simulations,

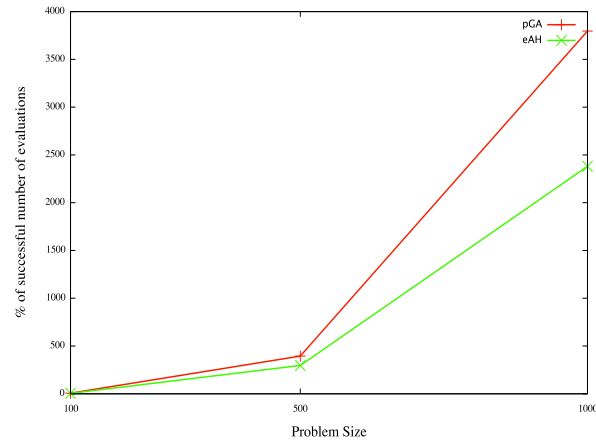


Figure 6. Average algorithms computation time with different problem sizes

Table II  
NETWORK CONFIGURATIONS WITH NS-2

Propagation Model	Nakagami
PHY/MAC Layer	IEEE 802.11p
Network Layer	AODV
Transport Layer	UDP
CBR Data Rate	64/128/256 kbps
Simulation Time	180 s

we limited number of generations for GA to 10. In eAH, the value of  $\alpha$  was set to 0.009. For both algorithms, we did the experiments by varying number of slaves as 2, 3, 4 and 5.

2) *Results:* Table III displays minimum (MIN), maximum (MAX), average (AVG), and Standard Deviation (STDEV) of fitness values for eAH and AH algorithms. The values are recorded by running every algorithm 5 times (smaller number of trials is due to nature of network simulations being very much time consuming). Observation shows that eAH found better AVG and MIN values. However slight difference is observed between two algorithms. The comparison when varying number of slaves, shows no significant trend.

Table III  
SOLUTION QUALITY

	eAH	AH
MIN	<b>8.227100E-03</b>	9.298830E-03
MAX	9.743050E-03	1.241070E-02
AVG	<b>9.040986E-03</b>	1.073452E-02
STDEV	7.041978E-04	1.439495E-03

Figure 7 presents overall computation time using eAH algorithm for different number of slaves. The trend shown by the Figure confirms that parallel processing was able to achieve a linear speedup ratio.

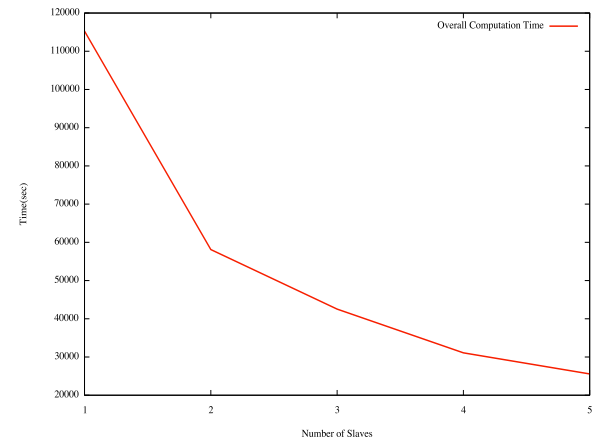


Figure 7. Computation time with number of parallel processes in eAH

Table IV shows QoS metrics validation results using global minimum solutions shown in Table III above, compared to values obtained in AODV RFC standard settings. The values suggest that eAH defeats both RFC

Table IV  
VALIDATION RESULTS

QoS Metric	RFC	AH	eAH
<i>PDR</i>	0.999776	0.999874	0.999968
<i>E2ED(ms)</i>	13.5011	1.88545	1.46504
<i>NRL</i>	0.052	0.037	0.032

standard and AH in maximising *PDR*, while minimising *NRL* and *E2ED*.

Further experiments are still being conducted using different data rates and different number of nodes (traffic density).

## VII. CONCLUSION

In this paper, we have presented the combination of hybridization and parallel processing techniques to solve some difficult optimization problems. The algorithm is formulated in master-slave topology in which a supervised iterative searching process is performed by slaves within strategically defined phases. Principles of EDAs with the help of *K*-means clustering algorithm govern the master's control process. The algorithm's performance was tested using benchmark instances of BQP problem and AODV routing optimization in VANETs. The results proved that our new approach performed better in both test problems.

## REFERENCES

- [1] S. Yamada, *The strategy and deployment for VICS*, IEEE Communication, 34(10), pp. 94–97. doi:10.1109/35.544328, 1996.
- [2] D. Reichardt, M. Miglietta, L. Moretti, P. Morsink, and W. Schulz, *CarTALK 2000 safe and comfortable driving based upon inter-vehicle-communication*, In Proceedings of the IEEE IV, 02, pp. 545–550, June 2002.
- [3] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. Decouto, *CarNet: A scalable ad hoc wireless network system*, In Proceedings of the 9th ACM SIGOPS European Workshop. Kolding, Denmark, September 2000.
- [4] K. C. Lee, U. Lee, and M. Gerla, *Survey of Routing Protocols in Vehicular Ad Hoc Networks*, Eds. IGI Global, 2009, ch. 8, pp. 149–70.
- [5] J. Garcia-Nieto and E. Alba, *Automatic parameter tuning with metaheuristics of the AODV routing protocol for vehicular adhoc networks*, in EvoApplications (2), ser. LNCS, vol. 6025, Springer, pp. 21–30, 2010.
- [6] J. Toutouh, J. Garcia-Nieto, and E. Alba, *Intelligent olsr routing protocol optimization for vanets*, Vehicular Technology, IEEE Transactions on, vol. 61, no. 4, pp. 1884–1894, 2012.
- [7] J. Toutouh, and E. Alba, *Parallel Swarm Intelligence for VANETs Optimization*, Seventh International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pp. 285–290, 2012.
- [8] E.G. Talbi, *Metaheuristics: From Design to Implementation* Wiley Publishing, 2009.
- [9] J. Garcia-Nieto, J. Toutouh, and E. Alba, *Automatic tuning of communication protocols for vehicular ad hoc networks using metaheuristics* Eng. Appl. Artif. Intell., vol. 23, no. 5, pp. 795 – 805, 2010.
- [10] J. Toutouh, S. Nesmachnow, and E. Alba, *Fast energy-aware olsr routing in vanets by means of a parallel evolutionary algorithm* Cluster Computing, pp. 1–16, 2012.
- [11] R.K. Chauhan and A. Dahiya, *AODV Extension using Ant Colony Optimization for Scalable Routing in VANETs*, Journal of Emerging Trends in Computing and Information Sciences, ISSN 2079–8407, Vol. 3, No. 2, pp. 241–244, 2012.
- [12] S. A. Ade and P. A. Tijare, *Performance Comparison of AODV, DSDV, OLSR and DSR Routing Protocols in Mobile Ad Hoc Networks*, International Journal of Information Technology and Knowledge Management, Vol. 2 No. 2. pp. 545–548, 2010.
- [13] S. M. Said and M. Nakamura, *Asynchronous Parallel Algorithms for Strategic Hybrid Searching Based on a Mixture Gaussian Model*, International Journal of Innovative Computing, Information and Control (IJICIC), vol.10, No.2, pp.459–479, 2014.
- [14] P. Merz and B. Freisleben, *Genetic Algorithms for binary quadratic programming problem*, In proceedings of the 1999 Genetic and Evolutionary Computation Conference, vol. 1, pp. 417–424, 1999.
- [15] P. Larrañaga and J. A. Lozano, *Distribution Algorithms*, Kluwer Academic Publishers, 2002.
- [16] S. M. Said and M. Nakamura, *A hierarchical Hybrid Evolutionary Algorithm for Continuous Function Optimization*, International Journal of Next Generation Computing, vol.3, pp.13 – 28, 2012.
- [17] Ray and R. H. Turi, *Determination of number of clusters in K-means clustering and application in colour image segmentation*, (invited paper) in N R Pal, A K De and J Das (eds), Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques (ICAPRDT'99), Calcutta, India, Narosa Publishing House, New Delhi, India, ISBN: 81-7319-347-9, pp 137-143, 27-29 December, 1999.
- [18] J.C. Bezdek, and N.R. Pal, *Some new indexes of cluster validity*, IEEE Trans. Syst. Man. Cybern., vol. 28, pp. 301–315, 1998.
- [19] G. W. Milligan, *Clustering validation: Results and implications for applied analyses*, In P. Arabie, L.J. Hubert and G. De Soete (Eds.), Clustering and Classification, Singapore: World Scientific, pp 341–375, 1996.
- [20] S. M. Said and M. Nakamura, *Asynchronous Strategy of Parallel Hybrid Approach of GA and EDA for Function Optimization*, Proceedings Third International Conference on Networking and Computing ICNC'12, WCOP3, pp.420 – 428, 2012.
- [21] J. Toutouh and E. Alba, *Real VANETs Simulation Scenarios*, <http://neo.lcc.uma.es/staff/jamal/vanet/?q=content/download-simulations>, 2011.