An Efficient and Robust Particle Swarm Optimization based Collision Avoidance Scheme for Autonomous Vehicles

Sania Khadim[1], Shafaq Murtaza[2]
Mirpur University of Science & Technology
Pakistan
sania_khadim@hotmail.com
shafaq.csit@must.edu.pk

Faisal Riaz
Iqra University, Islamabad
Pakistan
fazi_ajku@yahoo.com

ABSTRACT: *This research means to present a novel collision avoidance strategy for autonomous road vehicles utilizing a metaheuristic approach named 'Particle Swarm Optimization'. In exceedingly dynamic street environment, changes happen most of the time and in order to adapt to these changes vulnerability an exceptionally vigorous and capable calculation is required. Thus, PSO based plan won for being computationally sparing. The proposed PSO based impact evasion plan has been executed utilizing a swarm of 30 particles. PSO gives back an improved choice which can get away from the mischance situation with the exactness of more prominent than 93%. The presented exactness is in correlation with a perfect arrangement which has been acquired from earlier learning of the predetermined area. The strength and productivity of exhibited plan are stamped after its correlation with another collision avoidance technique based on 'Genetic Algorithm' which is also a biologically inspired optimization algorithm. The two plans were looked at on the premise of the quality of solutions being produced and required computation time. Re-enactment results have demonstrated the value of PSO construct approach in the light of two presented measurements.*

Keywords: Particle Swarm Optimization, Autonomous Vechicles, Genetic Algorithm, Optimization Algorithms

Received: 1 July 2016, Revised 4 August 2016, Accepted 10 August 2016

© 2016 DLINE. All Rights Reserved

1. Introduction

Today's quick pace transportation framework includes a serious life danger. Street crushes are well-thoroughly considered to be one of the main worries of loss of lives. As indicated by [1], World Health Organization has introduced a figure of around 1:2 million street people who lose their lives in various accidents and around 50 million get harmed each year. These street mischances result in life misfortune as well as cause different physical inabilities and financial loss [2]. The issues of street losses turn out to be more severe when it comes to under-developed nations. In just Pakistan, consistently a figure of nearly 7000 gives death toll of Pakistanis activated by street crashes [3]. Foundations of this calamity lie on the ground of the human driver.

The humanoid deficiency lies behind countless disasters. It has been observed in the Guangdong area of China that the majority of the street crashes happen because of over speeding and alcoholic status of the human drivers [4]. The mental and physical

effort of the driver may likewise lead towards the catastrophe [5]. Occupied drivers put their and other street commuters' lives at risk. In [6], authors have observed the results of messaging whilst driving and demonstrated that diversion brought on by messaging expands the odds of crash. From past decade or so a few writing and studies have recognized the key issues behind the street disaster and consequently opened new research skylines for the up and coming researchers. There exists a concurrence on either diminishing human intercession in crisis circumstances through driving assistants [7] or absolutely kill the human endeavors and make vehicles self-sufficient.

High computational capacity and exactness of today's processors are sufficiently fit to adapt to the exceptionally dynamic street environment and deliver opportune and precise results in any emergency situation. Accordingly, the term self-driving autos appeared in [8]. In [9], Google presented their self-directed auto. These self-coordinated autos have demonstrated their proficiency by bringing down the quantity of street crashes [8]. The goal of Intelligent Transport System (ITS) is to outline an efficient collision evasion methodology for autonomous vehicles [10].

A processor with high speed is of no worth if the driving calculation is not sufficiently powerful. A few procedures have been proposed and still are the matter of thought, these techniques mean to keep away from the impact by creating an on time response to the dynamic environment. A broad writing audit has demonstrated an exploration hole in the field of computational Intelligence (CI) field, which is as a rule less investigated in this regard. To the extent CI is concerned, just the Genetic Algorithm (GA) has been adopted by Faisal et. al in [11] to propose an impact shirking plan. As indicated by the outcome introduced in [11], GA being a developmental methodology has given great results which are meeting the Quality of Services (QoS) necessity. Be that as it may, creators have not given the computational many-sided quality of the proposed plan. This exploration additionally contributes its endeavors by actualizing the displayed GA keeping in mind the end goal to register the calculation time. In the event that we discuss in CI connection, there exists another algorithm called Particle Swarm Optimization (PSO), which has been discovered computationally modest than GA [12].

In the wake of processing and examining GA results, we have proposed a much quicker PSO based crash shirking plan. The structure of the parameters and fitness function have been embraced from [11], keeping in mind the end goal to show a reasonable correlation between the two plans. Two measurements i.e. Quality of solution and calculation time have been utilized to quantify the effectiveness of introduced scheme. Reenactment results have demonstrated that our proposed PSO based plan is by normal 14 times speedier than GA-based plan.

Rest of the paper has been composed as follows: Section 2 is demonstrating the received approach; section 3 speaks to an audit of related writing. In section 4, subtle elements of the proposed plan have been given. Usage of PSO based plan has been given in section 5. Section 6 means the extensive correlation amongst GA and PSO based crash evasion plans and 7 concludes the entire research.

2. Methodology

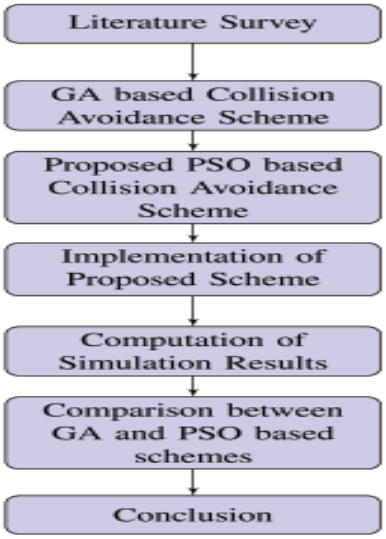

Figure 1. Proposed Methodology

The embraced system for the proposed issue is that first the artistic confirmation has been gathered to discover the examination crevice. We discovered just a Genetic Algorithm (GA) based crash shirking plan. After point of interest investigation of GA based plan, we think of our PSO based crash shirking plan. In the wake of defining the issue as indicated by PSO calculation, we executed the detailed issue in C#. Execution results are computed and later contrasted with GA based scheme. Comparison is performed in light of the premise of two exhibited measurements. At last, finishing up lines are drawn for future investigation in this issue area. Figure 1 is demonstrating the strides required in proposed approach.

3. Literature Review

The present day pattern of making self-governing machines has revealed the concerning examiners in finding the most ideal methodologies, which will guarantee the right usefulness of self-coordinated machines. Prior exploration in the zone of astute transportation framework was centered on to diminish the human intercession in a crisis circumstance. For this, a few approaches are embraced, one of them is Vehicle-to-Vehicle correspondence (V2V) [13]. However, human blunders amid driving may make unfavorable circumstances [14], that is the reason that most recent exploration pattern is putting great endeavors to make vehicles driverless. In such manner, a few collision avoidance schemes have been created. Location of obstructions is of most extreme significance for self-governing vehicles to maintain a strategic distance from obstructions and safe driving. For instance, to recognize the impediments (humans, animals or some other obstacle) around the way of self-sufficient vehicles, researchers have displayed a low thrown sensor, fisheye cameras plan on a European V-Charge task's model given in [15]. For 360 degree picture location and following, two long range stereo cameras are set at the front and back, while four tangible cameras are put at right, left, front and back of the vehicles. Creators have proposed a complete strategy along discrete strides for obstruction recognition, snag discovery begins with the picture catching then unwrapping it and afterward ordering the tube shaped picture utilizing Kalman channel. Another case is in [16], in which developers have secured a point by point portrayal of 360 degree defending around the vehicle from civilian and military viewpoint. A test bed vehicle was get ready to check the proposed obstruction recognition and following system. Test bed vehicle contains movement and video sensors, Ladar, a light strip range discoverer, five Pentium processors and numerous different sensors and actuators. As indicated by proposed plan, video sensors catches the pictures of impediments, after that picture unwrapping is performed then it utilizes Kalman channels to track the hindrances from caught pictures. Faisal et. al in [11] has displayed a GA based collision avoidance plan , which commutes the separation from front, rare, left and right hurdles. Then on the basis of current scenarios an ideal solution is produced which is provided to GA for the generation of an optimized result.

The use of PSO to introduce autonomy in the agents has been explored by few researchers. Some have been presented here to elaborate the significance of targeted algorithm in different problem domains. In [17], Somaiyeh has tried to implement the PSO to locally plan a route for underwater vehicles and then extending the guided area by global route planning. The discrete nature of path planning task has been effectively solved through a heuristic and continuous nature approach, which is robust and efficient. To plan a collision-free path for four-dimensional trajectory, Alejo et.al has presented a system in [18]. This system works with multiple algorithms. First, minimum bounding box based algorithm identifies the possible encounter between unmanned aerial vehicles, then a simple trajectory planning algorithm finds any non-optimal solution which is then finally optimized by PSO. Further, the authors have provided an improvement in previous mentioned scheme by reducing the dimensions of trajectory and sensibly selecting the maneuver to be taken in [19]. Another problem solved by PSO is the source seeking problem given in [20]. Source seeking problem has been solved through the deployment of a swarm of autonomous vehicles by introducing a planner for the swarm. Mobile agents have been represented as the particles in swarms and their positions are updated by PSO strategy. In [21], three white space optimization schemes have been given by the authors. First one is GA based, second one is the improvement in first scheme called Memory Enabled GA based and third one is the PSO based. According to presented results, PSO provides more robust white space optimization than GA. While Memory Enabled GA proven out to be the fastest of all.

The combination of collision avoidance schemes and PSO have shown us a novel way of planning an effective strategy that will ultimately arise the mankind's expectations from autonomous vehicles.

4. Proposed Scheme

The reason for this exploration is to propose such a crash shirking plan, which will create quick and upgraded results on account of any crisis in a very eccentric street environment. To check the proficiency of the introduced algorithm, we initially created a mishap driving situation by figuring the separation from the front, rare, left and right vehicles. By looking at the mischance

driving situation, a perfect choice in light of field studies is created, which will get away from the impacting circumstance effectively. The accomplishment of PSO based plan is to give a choice which is either equivalent or near the perfect choice.

4.1 Particle Swarm Optimization based strategy

PSO, a relatively new algorithm in CI field, was introduced in mid-1990's by Kennedy and Eberhart [22]. As the name implies it consists of a swarm of different particles, each particle represents a potential solution. This algorithm basically consists of three steps. Firstly, it initializes a swarm of particles, their positions, and initial velocities. Secondly, evaluates the fitness of each particle and thirdly updates the velocity and position of each particle in the solution space, in order to find an optimized solution.

4.2 Position structure of each particle

As mentioned earlier, the position of each particle holds a potential solution, so it is very important to carefully formulate the design space of particles' positions. In our problem domain of autonomous vehicles, the design space has been adopted from [11]. Each particle's position is represented by four dimensions i.e. Speed, Brake, Inner tire Angel, and Time To Avoidance (TTA). After each iteration, all particles update their velocities and positions by following their own (local) and social (global) experience.

4.3 Fitness Function

The question of how fit is the particle's current position in the solution space is answered by a function called 'Fitness Function'. Higher the value of fitness, more fit is the particle. In every iteration, each particle ends up with a personal best local solution. Among these particles, the highest fitted particle in the swarm holds the global best solution. In next iteration, all particles are updated in a manner that they try to improve their personal best by following the global best solution. As our particle position is represented by four dimensions. So, for every dimension following formulas have been used to find the fitness. In the first step difference of previously mentioned dimensions is calculated from the ideal decision's dimensions.

$$Difference\ (D) = Ideal\ Decision - Current\ Position \tag{1}$$

As the difference has been computed now fitness can be found by using equation given as under.

$$Fitness(F) = x * (\frac{D}{P}) \tag{2}$$

Here 'x' is the assigned weight i.e. the weight that any particular dimension holds in the particle's position. In our simulation, we have set x equals to 25. 'P' is any randomly selected number having values within the upper and lower limits of the dimension. If the difference is greater than 'P' then fitness will be given as.

$$Fitness(F) = x \tag{3}$$

4.4 Velocity and Position update

After calculating the fitness value of each particle, its velocity is updated by the following equation.

$$v^i_{k+1} = wv^i_k + c1r1\ (p^i_k - x^i_k) + c2r2(p^g_k - x^i_k) \tag{4}$$

Here 'w' represents the inertia factor that tries to balance between the explored and unexplored solution space. The position update is then given by the equation given below.

$$x^i_{k+1} = x^i_k + v^i_{k+1} \tag{5}$$

The basic PSO algorithm elaborated in [22] with "inertia weight" factor 'w' has been implemented in this study.

4.5 Proposed Algorithm

The PSO based collision avoidance strategy for autonomous vehicles can be summarized by following steps.

The whole process starts with the generation of accident leading scenario. After this, an ideal decision with specified QoS (Quality of Service) values of each decision parameter is computed. PSO handles the rest of the processing. After initializing the

swarm of 30 particles, fitness of each particle is found. PSO algorithm keeps on iterating until an optimized particle is found with a global best position in the solution space.

5. Implementation of Proposed Algorithm

The suggested scheme is implemented in *C#*.Net(2013), in order to test its efficiency. PSO parameters $c1$ and $c2$, the acceleration coefficients, are set after extensive testing Depending on problem domain, these parameters may take different values for fast convergence. In our specific problem, $c1$ and $c2$ are set to 0:7 and 0:9 respectively. Literature shows that for inertia weight-w 0:5 value proves out to be a better choice to go with [22]. Every dimension of the particle is initialized randomly from a specified range of values given in Table 1. Parameter 'Brake' is a binary variable which holds either 0 or 1 value. Rest of the parameters' information is given as under.

Algorithm 1 PSO based collision Avoidance Algorithm

1: **procedure** PSO_COLLISION_AVOIDANCE

2: Compute front_distance,rare_distance, left_distance, right_distance.

3: if (front distance >rare_distance)

4: {*Speed*= *Speed* + 20

5: Break= 0 }

6: else {

7: if (rare_distance >20)

8: {Speed= Speed - 20

9: Break= 1

10: } }

11: else

12: {*Speed* = *Speed* - 10

13: Break= 0 }

14: if(left_distance > right_distance)

15: {*Angle*= *Angle* + 20

16: *TTA*= *TTA* + 10 }

17: else

18: {*Angle*= *Angle* - 20

19: *TTA*= *TTA* -10 }

20: Initialize Swarm of 30 particles

21: **for** each iteration *i* **do**

22: Calculate the fitness of each dimension

23: Calculate the total fitness of each particle

24: Update Velocity

25: Update Position

26: **end for**

27: Return Optimized Decision

28: **end procedure**

The main simulator screenshot is shown in Figure 2, which gives a glimpse of implementation of GA and PSO algorithms fordifferent collision avoidance scenarios.

Parameters	Values	Parameters	Values
Swarm size	30	No of iterations	50
Front_distance	rand(2-25)	Rare_distance	rand(2-25)
Left_distance	rand(2-25)	Right_distance	rand(2-25)
Speed	rand(60-120)	Angle	rand(60 - 120)
Break	0-1	TTA	20% - 90%
Inertia 'w'	0.5	c1 & c2	1.5
r1	0.7	r2	0.9
Initial Velocity	0.5	x	25

Table 1. Simulation Parameters

6. Simulation Results

Sr.#	Ideal Decision	PSO Returned Decision	Fitness
1	(120; 0; 81; 79)	(120 4; 0 0; 81 5; 79 5)	95:2%
2	(107; 0; 86; 83)	(107 2; 0 0; 86 6; 83 4:5)	96%
3	(97; 0; 109; 72)	(97 3; 0 0; 109 2; 72 4)	98%
4	(68; 1; 71; 21)	(68 5:5; 1 0; 71 4; 21 3:5)	97%
5	(115; 0; 82; 81)	(115 3:5; 0 0; 82 5:5; 81 3)	96%
6	(85; 0; 89; 50)	(85 3; 0 0; 89 2; 50 2)	99%
7	(120; 0; 60; 20)	(120 4:5; 0 0; 60 5:5; 20 4:5)	92%
8	(76; 1; 97; 30)	(76 5; 1 0; 97 4:0; 30 2:5)	95%
9	(86; 0; 87; 53)	(86 5; 0 0; 87 5:0; 52 3:0)	97%
10	(56; 0; 120; 52)	(56 4; 0 0; 120 4; 52 4:0)	97%
11	(93; 1; 89; 66)	(93 5:5; 1 0; 89 6:5; 66 3)	96%
12	(89; 0; 77; 46)	(89 2; 0 0; 77 3:0; 46 3)	98%
13	(75; 1; 60; 46)	(75 4; 1 0; 60 7:6; 46 3)	96%
14	(102; 0; 71; 20)	(102 2; 0 0; 71 6:5; 20 4:5)	93%
15	(71; 0; 120; 67)	(71 4; 0 0; 120 4:5; 67 5)	96%
16	(120; 0; 97; 59)	(120 2; 0 0; 97 6:5; 59 3)	95%
17	(85; 1; 89; 50)	(85 3; 1 0; 89 2; 50 2:5)	99%
18	(68; 0; 62; 45)	(68 2; 0 0; 62 5:5; 45 5)	95%
19	(67; 1; 120; 67)	(67 2; 1 0; 120 2:5; 67 3)	97%
20	(67; 0; 90; 65)	(67 4; 0 0; 90 8; 65 6:5)	96%

Table 2. Simulation Results

In order to test the proposed algorithm, several accident scenarios are generated and the corresponding quality of the solution is measured by computing the total fitness of the PSO returned decision as compared to the ideal decision which sets its QoS parameters after judging the accidental situation. Every test is run fifteen times and then average values for the results are presented for interpretation and comparison purposes. Table 2 is presenting the twenty test cases. Ideal solutions corresponding to each test case containing values for our four decision parameters i.e. (Speed, Brake, Angle, TTA) are shown in second column. The values for PSO computed optimized solution along with their standard deviations are given in third and relative fitness in the fourth column of Table 2.

Results clearly show that in all twenty test cases, PSO has generated results with fitness values greater than 93% in average. Such a good fitness values are ensuring the high quality of the produced solutions.

7. Comparison Between Gaand Pso Based Collision Avoidance Schemes

The simulator screen-shot is representing the fact that we have not only implemented our proposed collision avoidance scheme but a GA based scheme of [11] has also been implemented for comparison purpose. Providing both schemes with similar scenarios, it has been noted that our proposed PSO scheme has not only produced high-quality solutions in average than GA but also is a computationally very robust strategy. We have defined two performance metrics for comparison. First is the quality of solution being produced and second is the computation time taken by each scheme. Comparison results based on each metric are presented in the following subsections separately.

7.1 One of the utmost important measures to judge the performance
of any strategy is to measure the accuracy of results. If results are accurate enough that they can be relied on, then under consideration strategy is said to be an effective one. We have adopted the fitness function formula given in section 4 to measure the quality. A comparison graph as depicted by Figure 3 is drawn between two schemes based on earlier mentioned twenty test cases given in Table 2.

Figure 2. Simulation Screen Shot

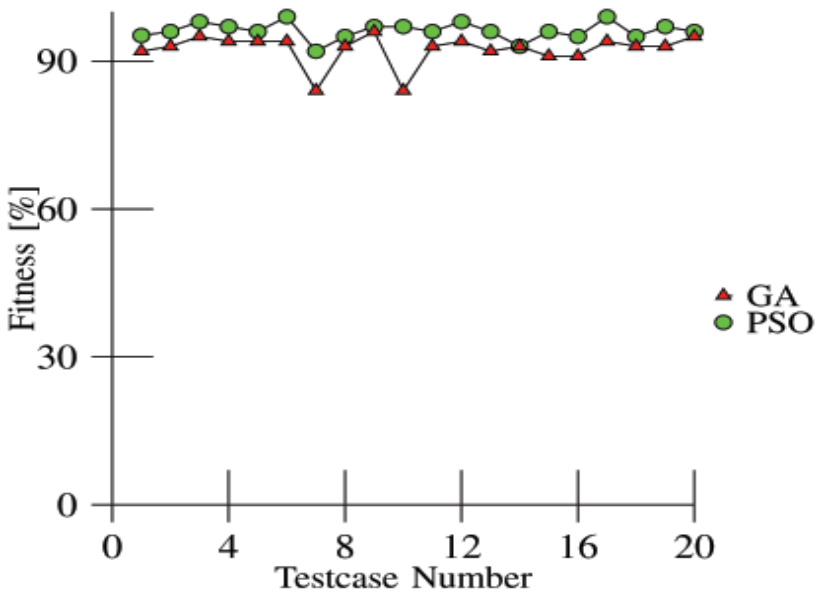

Figure 3. Comparison between PSO & GA Schemes

The comparison graph is reflecting the better solution quality of PSO generated results than GA. While implementing different scenarios,it has been observed that on average PSO returns more reliable solutions.

7.2 Computation Time of two Schemes

Sr.#	Ideal Decision	Time Taken by PSO	Time Taken by GA
1	(120; 0; 81; 79)	13*ms*	168*ms*
2	(107; 0; 86; 83)	16*ms*	206*ms*
3	(97; 0; 109; 72)	18*ms*	207*ms*
4	(68; 1; 71; 21)	13ms	206*ms*
5	(115; 0; 82; 81)	12*ms*	207*ms*
6	(85; 0; 89; 50)	12*ms*	207*ms*
7	(120; 0; 60; 20)	17*ms*	206*ms*
8	(76; 1; 97; 30)	15*ms*	206*ms*
9	(86; 0; 87; 53)	16*ms*	207*ms*
10	(56; 0; 120; 52)	16*ms*	207*ms*

Table 3. Computational Time Taken By Both Schemes

After analyzing the two algorithms and reviewing the various literature about CI field, it has been found that PSO is a computationally rational biological inspired strategy than GA [23]. To check this fact, we processed both schemes using Intel Core i3 processor with 4GB Random Access Memory. In order to get average processing time, each test was run fifteen times and in each run, computational time is computed. Some of the test cases results have been shown in Table 3.

The average computed time for both schemes has been depicted graphically in Figure 4 to show a visible difference. High rising

red bar is showing the slow convergence time taken by GA whereas the green bar is clearly declaring the PSO as a scheme with fast convergence time and hence in average 14 times faster than GA.

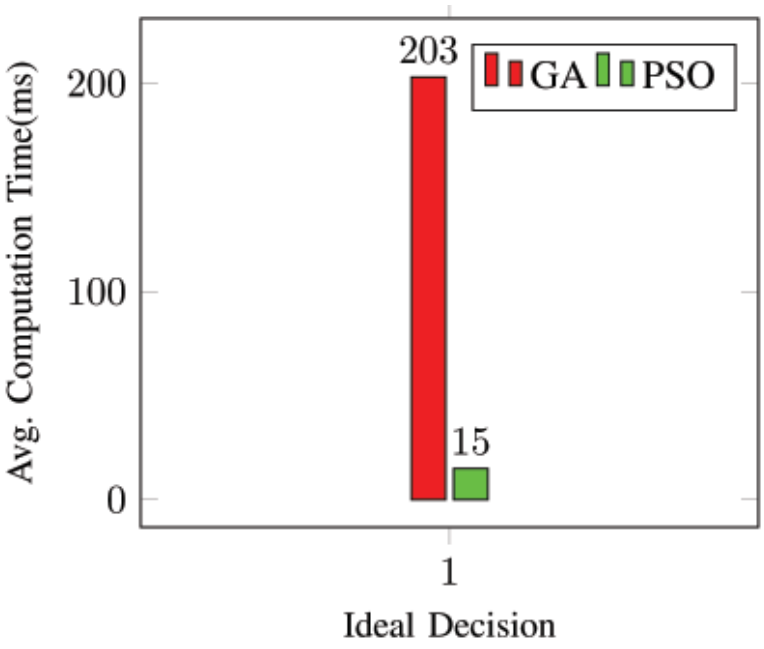

Figure 4. Average Computationa Time

8. Conclusion

This research is focused on the implementation of CI based optimization algorithms to select a collision avoiding decision for highly dynamic road environment. We have presented a PSO based collision avoidance scheme. The proposed algorithm has not only been implemented but also compared with a GA based collision avoidance scheme. PSO stood out in the comparison and not only produced optimum solutions but also provided a 14 times faster processing time. The proposed scheme's efficiency is leaving an open research question for current age scholars that- Why these CI based algorithms have been less explored in Vehicular Ad-hoc Network (VANET) if their proficiency is sufficiently higher to be depended on?

References

[1] Patton, G. C., Coffey, C., Sawyer, S. M., Viner, R. M., Haller, D. M., Bose, K. , Vos, T., Ferguson, J., Mathers, C. D. (2009). Global patterns of mortality in young people: a systematic analysis of population health data, *The Lancet*, 374 (9693) 881–892.

[2] Kazmi J. H., Zubair, S. (2014). Estimation of vehicle damage cost involved in road traffic accidents in karachi: a geospatial perspective,*In*: Procedia engineering, 77, p. 70–78.

[3] Ahmed, A. (2007). Road safety in pakistan, National Road Safety Secretariat, Ministry of Communications, Government of Pakistan, 2007.

[4] Zhang, G., Yau, K. K., and Gong, X., (2014) Traffic violations in guangdong province of china: Speeding and drunk driving, *Accident Analysis & Prevention*, V. 64, p. 30–40.

[5] Sigari, M.-H., Pourshahabi, M.-R., Soryani, M., Fathy, M. (2014). A review on driver face monitoring systems for fatigue and distraction detection, *International Journal of Advanced Science and Technology*, 64, 73–100.

[6] Rumschlag, G., Palumbo, T., Martin, A., Head, D., George, R., Commissaris, R. L. (2015). The effects of texting on driving performance in a driving simulator: The influence of driver age, *Accident Analysis & Prevention*, 74, 145–149.

[7] Hong, S., Doi, S., Min, B. (2014). Study of safety driving assistant system using audio-visual alert, *International Journal of*

Vehicle Safety, 7 (1) 54–69.

[8] Sanchez, D., (2015) . Collective technologies: autonomous vehicles, Australlian Council of Learned Academies, Tech. Rep.

[9] Markoff, J. (2010). Google cars drive themselves, in traffic, The New York Times, 10 (A1) p. 9.

[10] Sitavancova Z., Hajek, M. (2010). Intelligent transport systems: thematic research summary, The National Academies of Sciences, Engineering, and Medicine, Tech. Rep.

[11] Riaz, F., Niazi, M. A., Sajid, M., Amin, S., Ratyal, N. I., Butt, F. (2015). An efficient collision avoidance scheme for autonomous vehicles using genetic algorithm, *J. Appl. Environ. Biol. Sci*, 5 (8) 70–76.

[12] Engelbrecht, A. P. (2007). Computational intelligence: An introduction, J. wiley & Sons Ltd, Ed. John Wiley & Sons.

[13] Rigas, G., Bougia, P., Fotiadis, D. I., Katsis, C. D., Koutlas, A. (2008). Iway: Towards highway vehicle-2-vehicle communication and driver support, *In*: Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on. IEEE, 2008, p. 3376–3381.

[14] Riaz, F., Shah, S. I., Raees, M., Shafi, I., Iqbal, A. (2013). Lateral pre-crash sensing and avoidance in emotion enabled cognitive agent based vehicle- 2-vehicle communication system, *International Journal of Communication Networks and Information Security,* 5 (2) 127.

[15] Bertozzi, M., Castangia, L., Cattani, S., Prioletti, A., Versari, P. (2015). 360 detection and tracking algorithm of both pedestrian and vehicle using fisheye images, *In*: Intelligent Vehicles Symposium (IV), 2015 IEEE. IEEE, 2015, p. 132–137.

[16] Aufr'ere, R., Gowdy, J., Mertz, C., Thorpe, C., Wang, C.-C., Yata, T. (2003). Perception for collision avoidance and autonomous driving, *Mechatronics*, 13 (10) 1149–1161.

[17] Zadeh, S.M., Powers, D.M., Yazdani, A. (2016). Toward efficient task assignment and motion planning for large scale underwater mission.

[18] Alejo, D., Cobano, J., Heredia, G., Ollero, A. (2013). Particle swarm+ optimization for collision-free 4d trajectory planning in unmanned aerial vehicles, *In*: Unmanned Aircraft Systems (ICUAS), 2013 International Conference on. IEEE, 2013, p. 298–307.

[19] Alejo, D., Cobano, J., Heredia, G., Ollero, A. (2015). Collision-free trajectory planning based on maneuver selectionparticle swarm optimization, *In*: Unmanned Aircraft Systems (ICUAS), 2015 International Conference on. IEEE, 2015, p. 72–81.

[20] Zou, R., Kalivarapu, V., Winer, E., Oliver, J., Bhattacharya, S. (2015). Particle swarm optimization-based source seeking, *Automation Science and Engineering, IEEE Transactions on*, 12 (3) 865–875.

[21] Riaz, F., Shafi, I., Jabbar, S., Khalid, S., Rho, S. (2015). A novel white space optimization scheme using memory enabled genetic algorithm in cognitive vehicular communication, Wireless Personal Communications, p. 1–23.

[22] Kennedy, J. (2011). Particle swarm optimization, in Encyclopedia of machine learning. Springer, 2011, p. 760–766.

[23] Hassan, R., Cohanim, B., De Weck, O., Venter, G. (2005). A comparison of particle swarm optimization and the genetic algorithm, *In*: Proceedings of the 1st AIAA multidisciplinary design optimization specialist conference, 2005, p. 18–21.