

PARTICLE SWARM OPTIMIZATION OF SPEED IN UNPLANNED LANE TRAFFIC

Prasun Ghosal¹, Arijit Chakraborty², Sabyasachee Banerjee²

¹ Department of Information Technology, Bengal Engineering and Science University,
Shibpur, Howrah 711110, WB, India

prasung@gmail.com

² Department of Computer Science and Engineering, Heritage Institute of Technology,
Kolkata, WB, India

{arijitchakraborty.besu, sabyasachee.banerjee}@gmail.com

ABSTRACT

Analysis of Speed Optimization Technique in Traffic is a very promising research problem. Searching for an efficient optimization method to increase the degree of speed optimization and thereby increasing the traffic flow in a lane is a widely concerning issue. However, there has been a limited research effort on the optimization of the lane usage with speed optimization. This paper presents a novel technique to solve the problem optimally using the knowledge base analysis of speeds of vehicles, population of lanes, using partial modification of Swarm Intelligence which, in turn will act as a guide for design of lanes optimally to provide better optimized traffic with less number of transitions of vehicles between lanes..

KEYWORDS

Transition Points, Lane's Population, Lane's average Speed

1. INTRODUCTION

The challenges of the accidental and congested lane design system are to move traffic safely and efficiently, although, highways and motor vehicles are designed to operate safely at their optimum speed. The purpose of our investigation is to create predictive models for different types of speed optimization techniques on lane, based on infrastructural design and traffic intensity. In this paper, the results for all transition points and vehicle's lane transition for speed optimization is discussed.

The Analysis starts with identifying main issues and element of the problem in hand which are as follows.

- Entry zones,
- Transition points, and
- Exit zones.

Most of the traditional approach for tackling the problem in hand is based on deterministic models which can be efficient and more or less accurate at times, but to achieve optimality of solution deterministically, at all time, seems to be far from reality till now.

Apart from that, making the lanes at their optimal average speed at any point of time using previous knowledge and current information is a major highlight presented in this paper.

The overall organization of the paper is as follows. Section 2 discusses about the background of the present work with a description of the related works done so far in this area, and pointing out the drawbacks of the existing solutions. In the next section, the problem formulation and proposed algorithms are represented. First algorithm does not consider the concept of population knowledge base, and second one with the population knowledge base. Experimental results and observations are represented in section 4. Finally, section 5 concludes the paper with possible future directions of work.

2. BACKGROUND AND MOTIVATION

Since we know transportation cost plays a huge role while deciding any products cost and quality. So our basic aim is to optimum the vehicle's speeds as well as optimum the usage of the lanes present in the traffic. So our paper plays an important role for public interest.

2.1. Related Works

The paper proposed by Jake Kononov, Barbara Bailey, and Bryan K. Allery, first explores the relationship between safety and congestion and then examines the relationship between safety and the number of lanes on urban freeways.

The relationship between safety and congestion on urban freeways was explored with the use of safety performance functions [SPF] calibrated for multilane freeways in Colorado, California, Texas.

The Focus of most SPF modeling efforts to date has been on the statistical technique and the underlying probability distributions. The modeling process was informed by the consideration of the traffic operations parameters described by the Highway Capacity Manual. [1]

H Ludvigsen, Danish Road Directorate, DK; J Mertner, COWI A/S, DK, 2006, published, Differentiated speed limits allowing higher speed at certain road sections whilst maintaining the safety standards are presently being applied in Denmark.

The typical odds that higher speed limits will increase the number of accidents must thus be beaten by the project.

That paper presented the methodology and findings of a project carried out by the Danish Road Directorate and COWI aimed at identifying potential sections where the speed limit could be increased from 80 km/h to 90 km/h without jeopardizing road safety and where only minor and cheaper measures are necessary. Thus it described how to systematically assess the road network when the speed limit is to be increased. [2].

C.J. Messer and D.B. Fambro, 1977, presented a new critical lane analysis as a guide for designing signalized intersections to serve rush-hour traffic demands.

Physical design and signalization alternatives are identified, and methods for evaluation are provided. The procedures used to convert traffic volume data for the design year into equivalent turning movement volumes are described, and all volumes are then converted into equivalent through-automobile volumes.

The critical lane analysis technique is applied to the proposed design and signalization plan. The resulting sum of critical lane volumes is then checked against established maximum values for each level of service (A, B, C, D, E) to determine the acceptability of the design. [3].

2.2. Drawbacks of existing solutions

Many traditional speed-optimizing algorithms for lanes were proposed earlier to optimize deterministic problems. But these algorithms didn't show their ability to use their previous knowledge to tackle the inherent randomness in the traffic systems. Therefore, to handle with such random realistic situation and generate some efficient solution, good computational models of the same problem as well as good heuristics are required.

This article is divided into two major sections: -

In first part, simulation algorithm will provide us with no. Of lanes required moving the traffic at optimal speed in each proposed lane.

Second part, deals with knowledge obtained from the first part to make the lane transitions less in number making it nearer towards the desired goal.

3. PROBLEM FORMULATIONS AND PROPOSED ALGORITHM

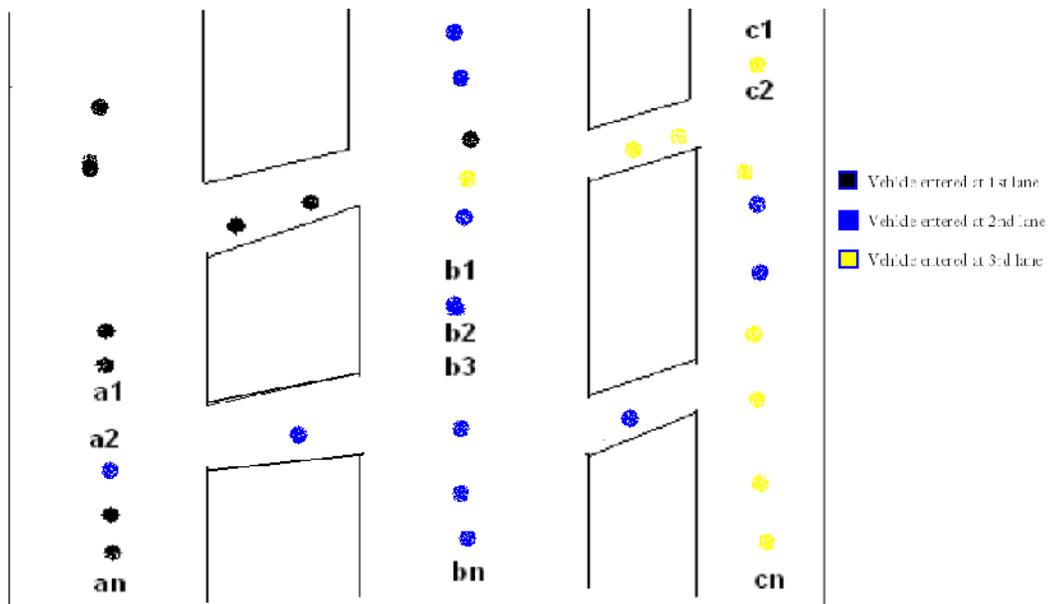


Fig. 1. Vertical lanes are unidirectional and with the property of the three lanes with transition points.

Description of the Figure

Figure 1, Three vertical lanes that are unidirectional, and $A = \{a1, a2... an\}$, $B = \{b1, b2 \dots bn\}$, $C = \{c1, c2, \dots, cn\}$, three lanes. I, II, III are the transition points through which vehicles can overtake its preceding vehicle with lesser speed and then immediately moves to its original lane. i.e. I from lane A to B or B to A and II, III are from B to C or C to B. Here we assume that each

and every lane's car speed is greater than 0 kmph. If speed of any car is less than or equal to 0 kmph then we assume that there may be problem.

Generally in a lane where no vehicle can overtake its preceding low speeding vehicle its vehicle got inherited by the low speeding vehicle which creates lots of problem. Therefore we have first calculated the number of lanes that will be required to move the vehicles in its optimum speed then we have taken the number of lanes as an input of our proposed second algorithm.

3.2. Problem Formulation

Random movement of vehicle in rush hour traffic are required to be frame up in optimal no. lanes with respect to number of transitions between lanes so that each lane have optimal speed.

Bio inspired algorithms like swarm intelligence (i.e. Ant Colony Optimization where ant follows the previous ants pheromone trail for their optimum routing) technique used here with speed of the 'vehicle' acting as a pheromone to solve the problem in hand.

To maintain the optimality of a solution in a heuristic search using population information as a knowledge base is used in the proposed algorithms.

3.3. Proposed Algorithms

3.3.1. Algorithm I

Initial Assumptions

- There will be no change in the speed of the vehicle
- In case of sudden change of speed, accommodate the speed of previous slower vehicle.
- Any vehicle having speed equivalent of 0 is discarded from the initial sample or population

Table 1: Symbolic Interpretation used in algorithms

Symbols used	Meaning
V_i	Velocity of vehicle i
V_j	Velocity of vehicle j
L_i	Lane of the vehicle i
L_j	Lane of the vehicle j
L_1	Lane of the 1st vehicle.
type(i)	Category of Vehicle i
t	Arrival time difference between a high and low speed vehicles
t1	Time interval to overtake a vehicles at lower speed
d	Distance covered by low speed Vehicle
d1	Distance covered by high speeding Vehicle
B_n	Buffer of Lane n
Count	Total no. Vehicle in unplanned traffic
Count1	Total no. Lanes for optimal speed
Count2	Total number Of transition

Details of the proposed Algorithm

The major keynotes and functionality of the proposed algorithm are as follows: -

Step 1 is taking input from sensors, like the current speed of the vehicle, arrival time etc., and, counting the number vehicles the user has entered.

Step 2 is categorizing the vehicles depending on their current speed.

Step 3 is checking total how many numbers of lanes will be required for our sample data in an unplanned zone, and, which vehicle is moving in which lane.

Step 4 is checking total number of transitions i.e. at which point of the lane and from which lane to where the transition will occur.

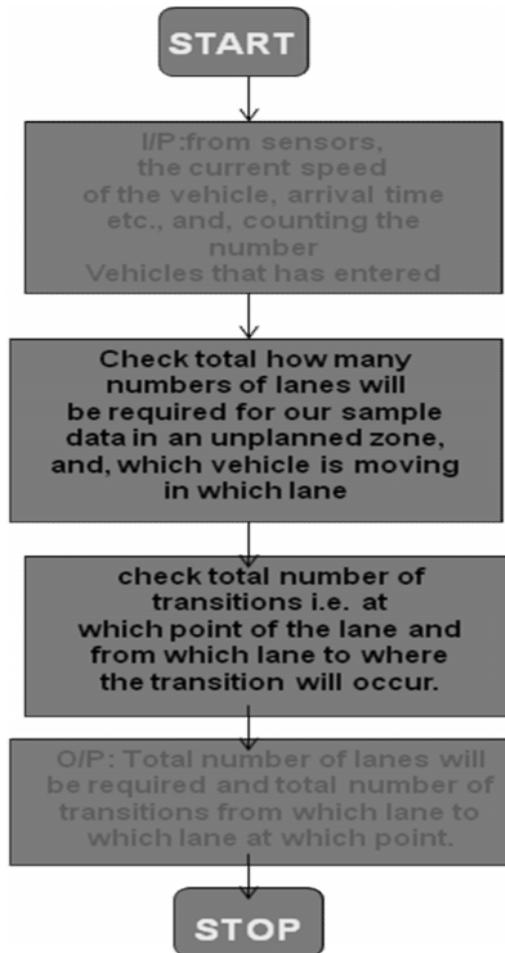


Fig. 2. Pictorial representation of Proposed Algorithm I

Pseudo code (Algorithm I)

Input: Details of vehicles, Current speed of the vehicle, arrival time.

Output: Category of the vehicle, Number of lanes will be required, Number of transitions.

Step 1.1: Set count = 1; /*Used to count the number of vehicles. */

Step 1.2: get_input (); /*Enter Details of vehicles, current speed, arrival time and store it into a record. */

Step 1.3: Continue Step 1.1 until sensor stops to give feedback and Update count = count + 1 for each feedback;

```

Step 2: For 1 <= i <= count for each vehicle
    If 0 < Vi < 11 then categorize Vi as type A
    If 10 < Vi < 31 then categorize Vi as type B
    If 30 < Vi < 46 then categorize Vi as type C
    If 45 < Vi < 51 then categorize Vi as type D
    If 50 < Vi < 101 then categorize Vi as type E
Step 3: Set counter: count1 = 1;
    Set L1 = 1;
    For 2 <= i <= count for each Vehicle
        For 1 <= j <= count1
            Compare the {type(i), type(j)} present in the lane
            If different update count1 = count1 + 1 and
            Li = count1;
            Else
            Li = j;
        End of loop;
    End of loop;
Step 4: Set counter: count2 = count1;
    For 1 <= i <= count - 1 for each Vehicle
        For 2 <= j <= count for each Vehicle
            If type(Vi) = type(j) and Vi < Vj and arrivaltime(Vi) <= arrivaltime(Vj)
            Set t = arrivaltime(Vj) - arrivaltime(Vi);
            Set t1 = 0;
                Begin loop
                Set t1 = t1 + 1;
                Set d = Vi x (t + t1);
                Set d1 = Vj x t1;
                If d1 <= d Set count2 = count2 + 1;
                If Lj = 1 then transition will be to 2 - lane;
                If Lj = count1 then transition is count1 - lane;
                Else
                Transition is either Lj - 1 or Lj + 1;
            End loop;
        End loop;
    End loop;
Step 5: Return Number of lanes required = count1;
    Number of transitions required = count2;
Step 6: End

```

Analysis of the Proposed Algorithm (Algorithm I)

- The above algorithm is implemented on an open unplanned Area.
- The objective will follow linear queue as long as speed/value/cost of proceeding is greater than the immediate next.
- Transition/Cross over are used and they again follow appropriate data structure in order to maintain the preceding step rule.
- Here we assume the lanes are narrow enough to limit the bi-directional approach.
- Here we maintain optimize speed for each lane.

- Here we also maintain the transition points if speed/value/cost of a vehicle is found unable to maintain the normal movement and transition in all the calculated lanes.
- Transition points are recorded with their position and number and it follows appropriate data structure in order to maintain the record.

3.3.2. Algorithm II

Description of the proposed algorithm.

The primary sections of the proposed algorithm and their major functionalities are described below.

- Step 1. Take relevant information from sensors, i.e. the current speed of the vehicle, arrival time etc. and count the number of vehicles the sensor has entered along with that consider number of lanes that are present in the traffic.
- Step 2. Assign lanes to different vehicles having different current speeds at any time instant t in order to categorize them.
- Step 3. Determine whether the current speed of the vehicle is equal to the speeds present in speed buffers of lanes or not.
- Step 4. This step finds the lane, where, the difference between the vehicle's current speed and lane's speed buffer's average speed is minimum and takes the vehicle to the lane, categorizes it same as the lane's other vehicles, increases the population of the lane, and stores the vehicle's current speed in the speed buffer of the lane.
- Step 5. This step is used for checking total numbers of transitions, i.e. at which point of the lane and from which lane to where the transition will occur, thereby calculating the average speed of the lanes.

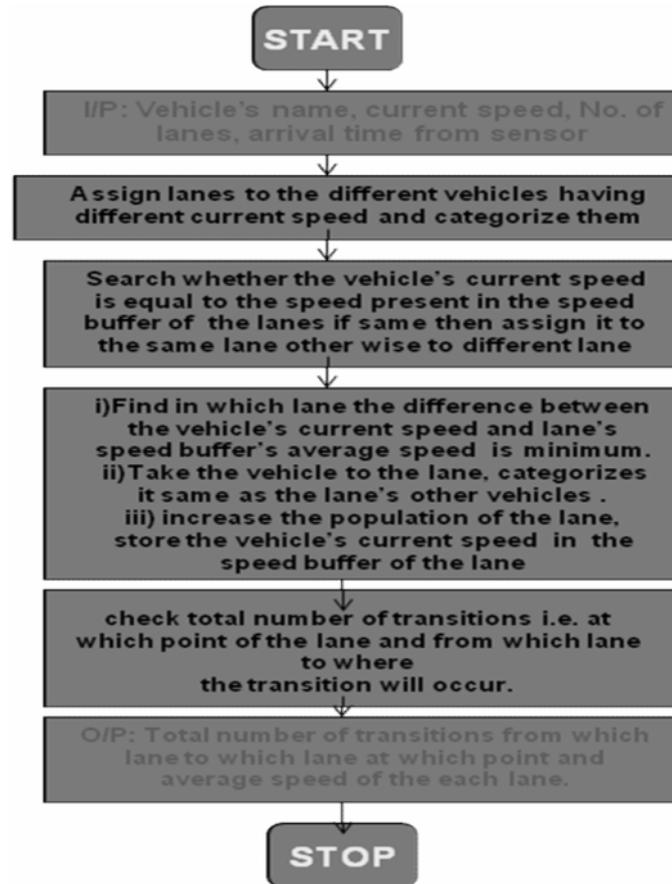


Fig. 3. Pictorial representation of Proposed Algorithm II

Pseudo code (Algorithm II)

INPUT: Vehicle's name, current speed, arrival time.

OUTPUT: Vehicle's Type, Number of transitions.

Step 1.1: Set count=1; /*used to count the number of vehicles*/

Step 1.2: get_input ()/*Enter the inputs when speed of the vehicle is non-zero. */

Step 1.3: Continue Step 1.1 until sensor stops to give feedback.

Step 2: Set type(1)='A', Enter V1 into 1st lane's speed buffer, Set 1st lane's population (count_1) as '1', Set n=2.

For 2 i count

Set a buffer buf=0

Loop1 until lane='0'

Loop2 for 1 j<I for each vehicle

If Vi = Vj

Set buf =1, type(i)=type(j)

Goto Step 3 and send Vi to Step 3 as 'speed1'.

Step 2.1

If buf=1 then end Loop1

If buf=0

Enter Bn=Vi, Set count_1=1, Set type(i)=A++;

End Loop1 /*Bn=n lane speed buffer*/

```

    If lane=0
    Then end Loop1.
    If lane=0
    Then end Loop.
    Store buf2=i+1
Step 3: For 1 i lane_1 for each lane
    If Bi's 1st speed=speed1
    Update count_li++;
    Set Bi, count_1=speed1
    goto step 2.1
Step 4: for buf2 i count
    Set c=1, switch=0.
    Set min=|Vi, Lc|, /*Lc=c lane's average speed*/
    type(i)=1st lane's vehicle type
    For 1 j lane_1
    Set d=|Vi, Lj|
    If d=0
    Set type(i)= type(Lj)
    Update (j) lane's count_1= (j) lane's count_1+1
    Set switch=1
    End Loop
    If d<min
    Then min=d
    Set type(i)= type(Lj )
    Update (j) th lane's count_1= (j) lane's conut_1+1
    Set (j) th lane's speed buffer [count_1] = (i) vehicle's speed (Vi)
    If switch=0
    Update L1, count_1 ++;
Step 5: Set count2 as count2 =1
    For 1 i count-1
        For 2 j count
            If type(i)= type(j) and Vi<Vj and (i) vehicle's arrival time (j) vehicle's
            arrival time
            Set t=(j) vehicle's arrival time - (i) vehicle's arrival time
            Set t1=0
            Begin loop
            Set t1=t1+1
            Set d=Vi*(t+t1)
            Set d1=Vj*t1
            If d1 > d set count2 = count2+1
            If Lj =1 then transition will be to 2-lane
            If Lj =count1 then transition will be to count1-lane
            Else transition will be to Lj -1 or Lj +1
            End loop
        End loop
    End loop
    For 1 m lane_1
    Calculate each lane's average speed from its speed buffer.
Step 6: Return Number of transitions required= count2
Step 7: End.

```

Analysis of algorithm II: The salient points and features of the proposed algorithm may be analyzed as follows.

- The above algorithm is implemented on an open lane area.
- The objective will follow linear queue as long as speed/value/cost of proceeding to greater than the immediate next.
- Transition/Cross over are used and they again follow appropriate data structure in order to maintain the preceding step rule.
- Here we assume the lanes are narrow enough to limit the bidirectional approach.
- Here we also maintain the transition points if speed/value/cost of a vehicle is found unable to maintain the normal movement and transition in all the calculated lanes.
- Transition points are recorded with their position and number and it follows appropriate data structure in order to maintain the record.

4. EXPERIMENTAL RESULTS AND OBSERVATIONS

The optimization of the speed in rush hour traffic with the swarm intelligence approach in an open lane area used the population information as a knowledge base. Primary objective of this approach is to improve the traffic movement in rush hours and to optimize the speed of the vehicles using the concept of transition points between adjacent Lanes.

The above proposed algorithms has been implemented using programming language ANSI C in an open platform , on a Intel Pentium IV processor, 1GB RAM, 80 GB HDD.

Below is the simulated graphical analysis of experimental results thereby obtained.

4.1. Simulated Graphical Analysis of the proposed Algorithms

By implementing the above proposed algorithm and doing the simulation we were able to generate the following graphical results shown in figures 4 and 5 as follows.

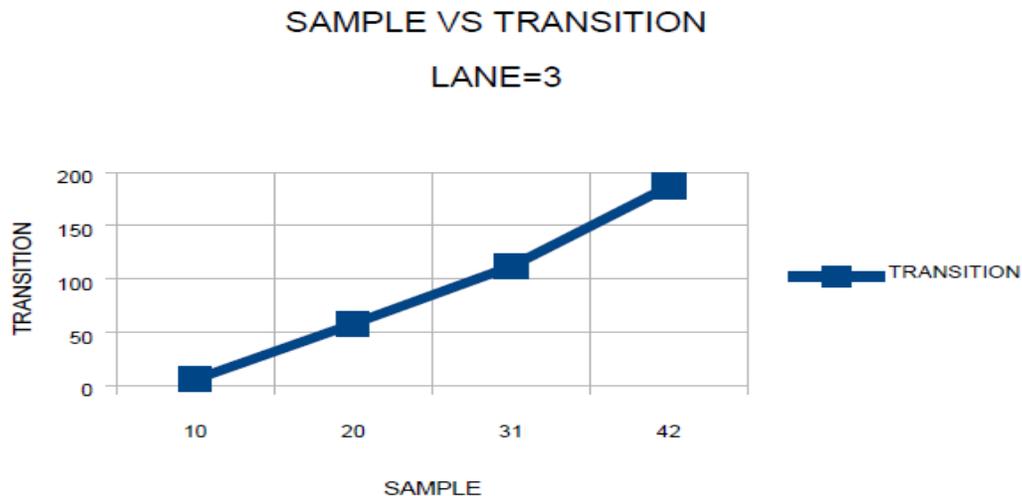


Fig. 4. This figure shows the variation of number of transitions with the number of lanes for a fixed number of samples i.e. 20.

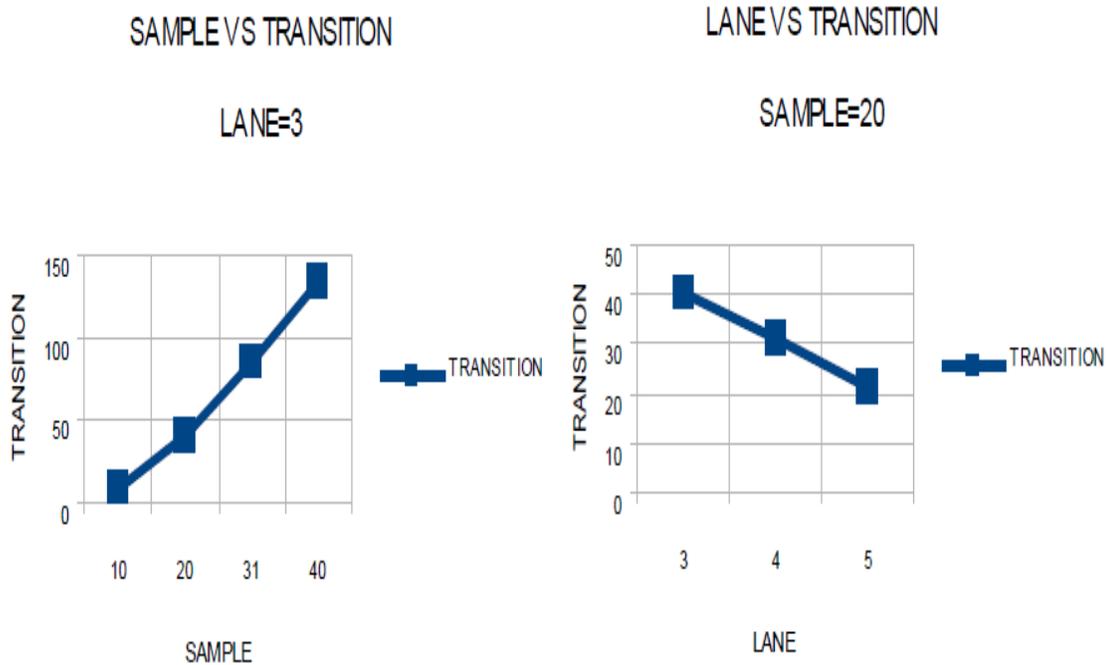


Fig. 5. First figure shows the nature of variation of the number of transitions with the variation of sample size without the population consideration. Second figure shows the same variation with population consideration.

Brief Analysis of Figure 4 and Figure 5

Analysis of the above-simulated results may be interpreted as follows.

- From figure 4 it is clear that as we increase the number of available lanes for a fixed number of samples, the number of transitions is decreasing drastically, which is, very much in conformity with the real life scenario.
- Another important point may be noticed from figure 5. As we are using the population knowledge base, there is a significant improvement in the number of transitions with the result when we were not using population knowledge base under consideration. This shows the effectiveness of our algorithm.

4.2. Time Complexity Analysis of the proposed Algorithm

We found our proposed algorithm has a time complexity of $O(N^2)$, which can be easily observed from the given table below. However we doesn't account all those iterations which doesn't contribute much to the time complexity of the proposed Algorithm

Table2: Time complexity of the number of steps and sub steps

Module Name	Time complexity
Step 1	$O(N)$
Step 2	$O(((N*N)+(L*L))/2)$
Step 3	$O(L)$
Step 4	$O(L*(\lfloor N/L \rfloor)*\log N)$
Step 4.1	$O(\log N)$
Step 5	$O((N-1)*(N-1))$

Here N is the number of cars and L is the number of lanes present in the traffic.

5. CONCLUSIONS

The article presented through this paper mainly emphasize on optimal usage of lanes using population information as knowledge base, but at the cost of transitions, because in real life scenario transitions may be too high, hence our future effort will be certainly in this direction.

In this article amount of time taken to transit between lanes has been considered as negligible. However cumulative sum of transition time between lanes in real world problem contributed much in optimality of the proposed solution.

Bio inspired algorithms (like swarm intelligence) has been used with population information as knowledge base, but partial modification of the stated concept taking weighted average of transition information as well as population information will certainly be taken into consideration during implementation and formulation of algorithms in future, there by optimizing various aspects of traffic movement in real world.

ACKNOWLEDGEMENTS

The authors would like to thank everyone, just everyone!

REFERENCES

- [1] Jake Kononov, Barbara Bailey, and Bryan K. Allery, "The relationship between safety and congestion", Journal of the Transportation Research Board, No. 2083.
- [2] "Differentiated speed limits", European Transport Conference Differentiated speed limits, 2007.
- [3] C.J. Messer and D.B. Fambro, "Critical lane analysis for intersection design", Transportation Research Record No. 644; 1977, pp, 26-35.
- [4] Prasun Ghosal, Arijit Chakraborty, Amitava Das, Tai-Hoon Kim, Debnath Bhattacharyya, "Design of Non-accidental Lane", In Advances in Computational Intelligence, Man-Machine Systems and Cybernetics, pp. 188-192, WSEAS Press, 2010.
- [5] Prasun Ghosal, Arijit Chakraborty, Sabyasachee Banerjee, "Design of Knowledge Based Efficient Speed Optimization Algorithm in Unplanned Traffic", The IUP Journal of Computer Sciences, Vol. VI, No. 1, pp. 23-30, January 2012.

Authors

Dr. Prasun Ghosal is currently associated with Bengal Engineering and Science University as an Assistant Professor. He has completed PhD (2011) from Bengal Engineering and Science University, M.Tech. (2005) as well as B.Tech. (2002) in Radio Physics and Electronics Engineering from Institute of Radio Physics and Electronics, University of Calcutta, India. He is also an Honours Graduate (major in Physics) under University of Calcutta. He has received Young Scientist Research Award for the year 2010-11 from Indian Science Congress Association. He is also recipient of several Best paper awards. His research interests include VLSI Physical Design: Algorithms and Applications, Network on Chip: Architectures and Algorithms, Embedded Systems: Architectures and Applications, Quantum Computing, Circuits, and Cryptography. He has contributed around 10 research articles in several peer reviewed international journals and around 40 in peer reviewed international conferences. Besides a copyright application he has also contributed towards several book chapters. He has carried out several sponsored research projects funded by AICTE, DIT, MCIT, Govt. of India, IEI etc.



Arijit Chakraborty is currently working in Heritage Institute of Technology. He is a post graduate (M.sc) in IT and also done his M. Tech (IT) from Bengal Engineering and Science University, Shibpur, India. His research interests include Soft Computing, Bio Inspired Algorithms.



Sabyasachee Banerjee, received B. Tech. degree in computer science from Government College of Engineering and Textile Technology, Shrirampore, under WBUT, WB, India in 2008 and M. E. in Computer Science from Bengal Engineering and Science University, Shibpur, WB, India in 2011. He is currently with the Department of Computer Science and Engineering, Heritage Institute of Technology, Kolkata, WB, India. His current research interests include Soft Computing, VLSI Physical Design algorithms and optimization.

