Article

Nature-Inspired Cloud–Crowd Computing for Intelligent Transportation System

Vandana Singh, Sudip Kumar Sahana * and Vandana Bhattacharjee

Birla Institute of Technology Mesra, Ranchi 835215, India
* Correspondence: sudipsahana@bitmesra.ac.in

Abstract: Nowadays, it is crucial to have effective road traffic signal timing, especially in an ideal traffic light cycle. This problem can be resolved with modern technologies such as artificial intelligence, cloud and crowd computing. We hereby present a functional model named Cloud–Crowd Computing-based Intelligent Transportation System (CCCITS). This model aims to organize traffic by changing the phase of traffic lights in real-time based on road conditions and incidental crowd-sourcing sentiment. Crowd computing is responsible for fine-tuning the system with feedback. In contrast, the cloud is responsible for the computation, which can use AI to secure efficient and effective paths for users. As a result of its installation, traffic management becomes more efficient, and traffic lights change dynamically depending on the traffic volume at the junction. The cloud medium collects updates about mishaps through the crowd computing system and incorporates updates to refine the model. It is observed that nature-inspired algorithms are very useful in solving complex transportation problems and can deal with NP-hard situations efficiently. To establish the feasibility of CCCITS, the SUMO simulation environment was used with nature-inspired algorithms (NIA), namely, Particle Swarm Optimization (PSO), Ant Colony Optimization and Genetic Algorithm (GA), and found satisfactory results.

Keywords: cloud computing; crowd computing; traffic signal control; nature-inspired algorithm

1. Introduction

The smart city concept, which is about making intelligent decisions following a comprehensive calculation, has taken off in today's widely digitized world. Moreover, such a framework is also marked with a collection of information mechanisms and communication technologies everywhere, and ITS (Intelligent Transportation System) has been one of the technologies used to control traffic [1]. An increase in the number of vehicles causes numerous problems, which are thus flattened by specific policies and rules identified under ITS. Congestion is a major road concern, which branches out into other issues such as fuel consumption and air pollution. This factor also makes it hard to implement public safety schemes in transportation [2] smoothly. In addition to these factors, the chances of road accidents are always high amid such chaotic situations, and the lives of those driving along (bikers, motorists and even pedestrians) are always at risk [3].

An urban traffic signal control system is needed to prevent driving delays caused by trunk line congestion, exhaust emissions from frequent vehicle starts and stops, and fuel waste from long idling times. Traffic control research has always focused on maximizing intersection capacity and reducing vehicle delay.

The problem can be solved by optimizing techniques, crowd and cloud computing. As regards verifying safety, the cloud platform provides accessible and timely information. For example, vehicle drivers get quick alerts about the road conditions and find out if such are clear or if some unpleasant event has taken place somewhere, due to which they need to opt for some safer route to reach their destination safely. Another solution may be using fog computing, through which authorities are informed about an accident or any

emergency and rescue efforts, such as ambulances and firefighters, are needed desperately. However, in the case of crowd computing, as mobile devices have become popular and are seen everywhere, such communication equipment can be turned into a mobile sensor which can help unlock real-time information concerning the vehicle's travel time, speed and usual traffic conditions. Such methodology is already in place in navigation systems such as Google's, which utilizes raw inputs gathered from users, providing real-time traffic information (crisp and reliable) to those desperately looking for traffic updates. This category of data is called "crowdsourced data", which are used to decipher mobility behaviour [4] and degree of congestion. Further, such type of data, sourced from the crowd, is pretty accurate and is available to users efficiently and at a pretty economical cost. Comparatively, physical sensors still depend upon traffic data and involve huge installation and maintenance costs. Even though tremendous advancements have been made in data availability, recent work about queue length information still shows a great leaning towards adaptive signal control.

Currently, research efforts being undertaken in the area of autonomous traffic signals control are concentrated in two primary areas. Firstly, automatic signal control adaptation models are designed to modify signal cycle program duration during the peak hour when vehicles in line demand it. The functioning of these technologies is strongly linked to the sensor system and the computation of real-time traffic flow [5]. Although these techniques have proven effective in several ci, real-time traffic network management has a very high functioning cost. Traffic flow patterns in the real world tend to repeat themselves (peak hours, holidays, etc.) [6].

In this paper, we have proposed nature-inspired algorithms, such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO), to provide a better flow of the vehicle on the road. A traffic signal cycle pattern will be found using SUMO, a traffic simulator. SUMO provides traffic cycle functionality for the instance's traffic signals. In this study, we have considered the road traffic network of Ranchi, Jharkhand (India). The current studies worldwide consider the following points to develop the CCCITS model.

- We offer an NIA method for generating efficient cycle schedules in a realistic urban environment (SUMO simulation network).
- Data from such a simulated network are stored in the cloud.
- Crowdsourcing and social media streams help update the cloud data.
- In case of any major event (accident, fire, etc.) that disrupts traffic flow, crowd data are provided, and updated information is stored in the cloud.
- Those about to take this affected route may change their direction after getting useful updates and can take another regular route to their destination.

In Figure 1, CCCITS model is presented. In this model, we have offered an NIA method for generating efficient cycle schedules in a realistic urban environment. First, we have coded and implemented nature-inspired algorithms (GA, PSO, ACO) on the SUMO simulation environment as a real-time road dataset. Secondly, the cloud and crowd computing frameworks are combined to dynamically update the network information for choosing an optimal path. Initially, the road network is stored on the cloud. In the cloud framework, road-related data are used to identify the best route with the help of crowd data. As the crowd gains access to cloud-based data, people can update this information in the cloud regarding any mishappening, road accidents, etc. Thus, the users of a particular road will be warned to pick a clear and safer route.

The following is a description of how this research work is structured. A survey of research work in the literature is discussed in Section 2—problem formulation and technology in Section 3. The optimization methodology is given in Section 4. The experimental approach and findings are presented in Section 5, and Section 6 contains the conclusions and recommendations for future work.
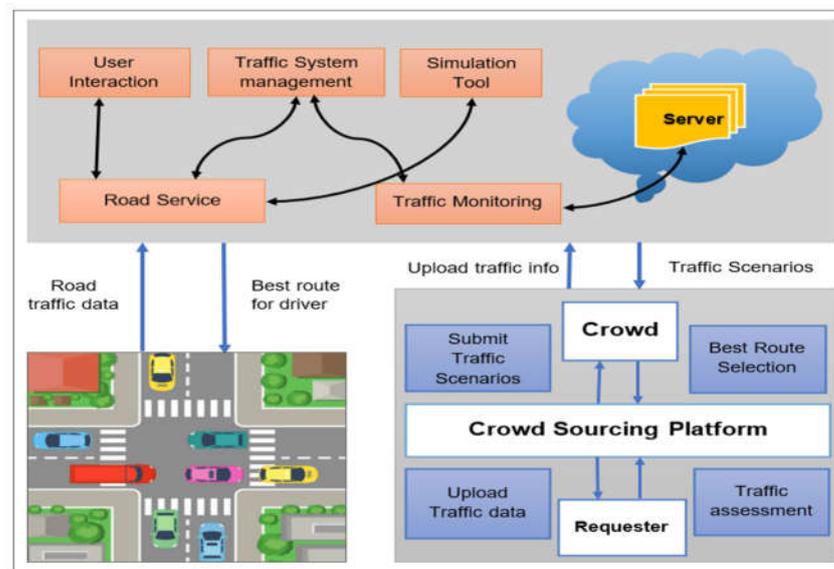
Figure 1. Cloud–crowd computing-based intelligent transportation system.

2. Literature Survey

The development of intelligent transportation systems (ITS) has led to the development of a wide range of implementations for safer and more convenient travel, owing to the creation of smart vehicles. On the other hand, the growing number of automobiles on roadways causes more and more congestion. Given this situation, a lot of research has concentrated on finding solutions or ways to reduce this problem. Congested cities are at crossroads. Signals are a systematic method for managing traffic at intersections to prioritize traffic movements and maintain an effective and safe flow of traffic. The pursuit of phase duration, cycle duration, and green times that would reduce interruption and optimize data has been pushed by growing congestion since traffic signals were first introduced to control conflicts at traffic junctions.

Congested traffic, characterized by frequent speed fluctuations and stops at signalized intersections, consumes most fuel on urban arterials. Traffic control strategies have been suggested to reduce urban fuel consumption and emissions, including the combination of an ecology-based performance index and an evolutionary method for optimizing traffic signal settings. Finding the best traffic light control parameters on a street network is an optimization problem [7]. As smart cities develop, the traffic signal must be improved to overcome the congestion on the traffic signal. Despite relentless efforts to improve traffic flow, road traffic is controlled by very old traffic signals (3-color signals). Traditional traffic signals have many problems, including inefficient time management at intersections and no way to prioritize emergency vehicles. New technologies, such as Vehicular Ad-hoc Networks (VANET) and the Internet of Vehicles (IoV), enable wireless vehicle communication [8] and are able to control traffic. Unlike previously, no such model and design could detect road traffic and provide smart solutions. Standard layout lacks the dynamic time intervals, learning, emergency priority management, and intelligent functionality of modern designs [9].

Traffic signal coordination is a multi-objective optimization problem. This research strategy can optimize a new traffic trunk coordinated control system, average delay, queue length, and total delay at intersections [10]. A predictive control model controls inlet traffic flows via network gates and traffic lights to control traffic congestion. Centralized and decentralized schemes are compared [11]. In a centralized system, one unit controls the network. This scheme is optimal, but it may not meet real-time computation needs in large networks. In a decentralized system, each intersection has its control unit that sends local data to an aggregator. The aggregator receives local information from all network control units, and the emergency vehicle augments it and shares it with the control units. Since

decision-making in a decentralized scheme is local and the aggregator must complete the above tasks during a traffic cycle, the decentralized system is suitable for large networks despite providing a suboptimal solution. In the future, the investigation can be expanded from different aspects, including the energy consumption and emission of vehicles, taking into account the publications [12–14].

The methods of controlling a traffic light in real-time are those that modify the signal to the current conditions on the road. In [15], the authors proposed a PSO-based traffic light cycle program to solve the traffic control problem. Real-world traffic data from Argentina and Spain are used to test the proposed approaches. Performance is evaluated in real-time using the SUMO traffic simulator. Compared to data from real-world traffic measurements, the proposed methods increase vehicle departures and journey duration. For the traffic signal intersection control problem, PSO with a linearly decreasing weight technique has been applied [16]. This study aims to develop a finite interval model that minimizes spoilage time. SUMO traffic simulator's traffic simulation was framed by a real-time simulation of Dalian, China. At intersections, a multi-objective-based PSO was developed [17] to maximize the amount of delay and emissions experienced by each individual while simultaneously increasing capacity. Every individual particle carries both the signal cycle and the green times. In Jinzhou, China, a four-way junction served as the testing ground for the program. The created Pareto-front solutions had a higher degree of uniformity and diversity compared to their counterparts that NSGAII produced. Furthermore, the proposed approach beat NSGA-II in terms of computing time while running on the same hardware.

To effectively schedule the phases of each isolated traffic light, an intelligent traffic light controlling algorithm (ITLC) [18] utilizes VANET communication to obtain real-time data. The real-time traffic aspects of every flow that crosses the road connection, including the number of moving vehicles, the speed of the traffic, and the volume of each traffic flow. In addition to ITLC, the authors considered arterial traffic lights (ATL), which manage traffic light signaling to control traffic at junctions. This algorithm finds nearby traffic lights as well as real-time road traffic aspects to design and implement in red-green light phases. an optimal traffic signal configuration schedule maximizes network traffic flow. The signal timing control problem involves determining optimal phase sequences and durations. To solve the traffic problem, geometric information describing target network intersections, traffic information, including traffic demand and vehicle turning movements, and traffic signal component limits are considered [19].

In [20], the paper proposed an intelligent context-aware negotiation protocol (ICANP) where vehicles can reduce speed with the traffic light to enter the assigned green phase. ICANP uses a genetic algorithm (GA) to reduce the crossing time. The ICAPP considers the vehicle. When a priority vehicle approaches a signalized intersection, the ICAPP suspends the green-light period for negotiating vehicles, allowing the first concerned vehicle to proceed without waiting time. In [21], Younes introduced a context-aware traffic light scheduling (CATLS) algorithm. By taking into account the traffic flow's characteristics, CATLS schedules each traffic light's phases. The green phase of the traffic light can be interrupted at any time so that emergency vehicles can cross the intersection. Due to the high cost and limited availability of delay data, most adaptive traffic signal systems have historically relied on volume and queue length data. SCOOT [22], SCATS [23], OPAC [24], and RHODES [25] are notable examples of this kind of system. These adaptive traffic signal systems (ATSS) and other commonly used ones use physical sensors such as loop detectors, radar sensors, or nearly real-time traffic videos. The queue length and volume count employed for adaptive signal systems at the network level by more modern and innovative algorithms, such as max-pressure [26], also depend on physical sensors. In the past, it was considerably more affordable to collect volume and queue data at intersections than delay data. Because of this, traffic signals have not produced or used delay data; however, it is randomly taken in most optimization algorithms implementations. In [27], research introduces an adaptive traffic signaling scheme based on road traffic density to

improve traffic signal control and management. Live video has been fed into a deep Q network to generate adaptive phase timings and introduced per car unit (PCU) as a novel input to represent each vehicle type's traffic effect. Real-time data prove that the proposed scheme increases traffic speed by 5.597 km/h. The proposed scheme increases mean speed by 175.71% compared to static methods.

Current work on ATSS has mainly relied on queue length knowledge despite improvements in data availability. Changing network back-pressure control [28] and controlling network back-pressure [29] has been a recent research focus. Despite the decentralized control approach, Max-pressure management has maximum analytical stability. A traffic network with stochastic demand and a specific signal control policy is stable unless the average vehicle number remains fixed over time. An unstable network suggests that there needs to be more capacity to meet the moderate demand, which is partly driven by the timing of the signals: limited queue buffers and dynamic method out-performs. Except [30], prior work only used queue length information to control movements [31].

3. Problem Formulation and Technology

This section provides an overview of the problem formation and technology used to control Road Traffic Signal Control (RTSC) with the help of the microscopic traffic simulator SUMO.

3.1. Nature-Inspired Algorithm (NIA)

Evolutionary Algorithm (EA) and Swarm Intelligence (SI) are the two types of nature-inspired algorithms, as indicated in the introduction. Both forms of algorithms, however, share a similar structure for solving a problem. Initially, the population is seeded with random or simple greedy heuristics, such as dispatch rules, to get things started. This means that the values of all the decision variables must fall within their designated search ranges or domains. The problem's objective function is used to evaluate the first solutions. Once this iterative procedure is complete, a new set of keys can be generated. EA uses techniques including selection, crossover, and mutation, all of which are influenced by Darwin's theory of evolution [32]. Such mechanisms are, nevertheless, derived from the motion equations of the living organism that inspired the algorithm in SI. The PSO and ACO pheromone updated rules use the ideas of locations and velocities. The new population members are examined to replace the old ones with established rules. Additionally, the population's objective values are also updated. Various local search or randomization tactics can be applied during the search to encourage exploitation or exploratory behaviour. A stopping requirement must be met before the iterative procedure may be completed (for example, a set number of iterations, a set number of function evaluations, a set acceptable error threshold, a set time limit, etc.). Finally, the method returns the optimal solutions along with their objective values. Figure 2 illustrates a flowchart for a nature-inspired algorithm.

3.2. SUMO Simulation

An open-source, highly portable and microscopic traffic simulation problem for extensive traffic networks is provided by SUMO [33]. SUMO requires several input files containing data on the traffic to be simulated. A network (.network.XML) includes information on the map's structure, such as nodes, connections, edges and nodes. A sequence of scripts included in the SUMO package may be used to import the network from well-known digital maps, such as OpenStreetMap [34], and transform it into a SUMO network.

This section explains our proposed optimization solution for the best cycle of a traffic signal generation program. It is responsible for encoding the key, performing the fitness function, and minimizing departed vehicle timing.
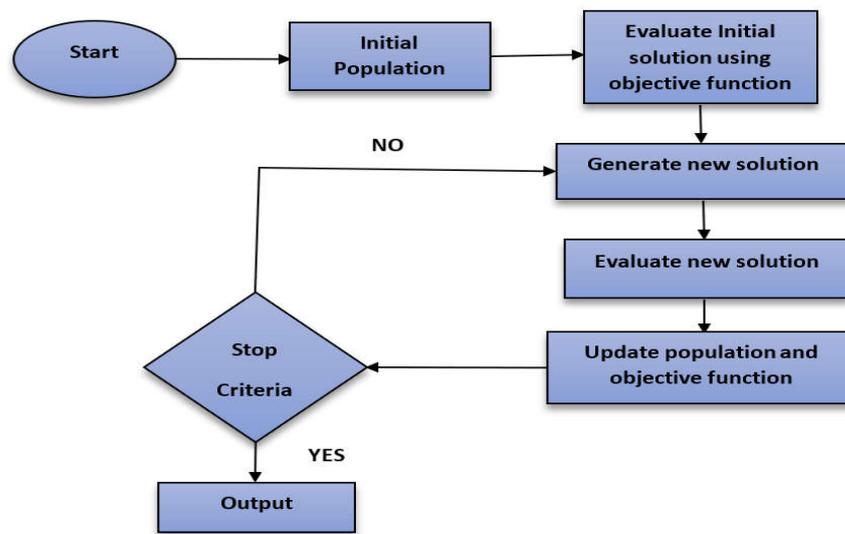
Figure 2. A flowchart of a nature-inspired algorithm.

3.3. Traffic Simulations and Their Solution

Each vehicle indicates a phase duration of one state of the traffic lights involved in a given intersection following the pattern of programming cycles utilized by SUMO. This information is then used to record the condition of the traffic lights in a real scenario. This is accomplished with the assistance of a traffic sensor. Following one another, the phase durations of the TL-logic components are inserted into the solution vector. This results in a straightforward array of numbers that maps the traffic light staging, as seen in Figure 3. The explanations are as follows: first, with integer values, the SUMO simulator depicts the discrete sequence of time steps (seconds) that composes the entire simulation technique. This sequence is what makes up the simulation. The presentation of information about the most effective traffic signals makes use of integer numbers, which is also a function of fitness. The optimization strategy for the Traffic light cycle program describes in Figure 4.
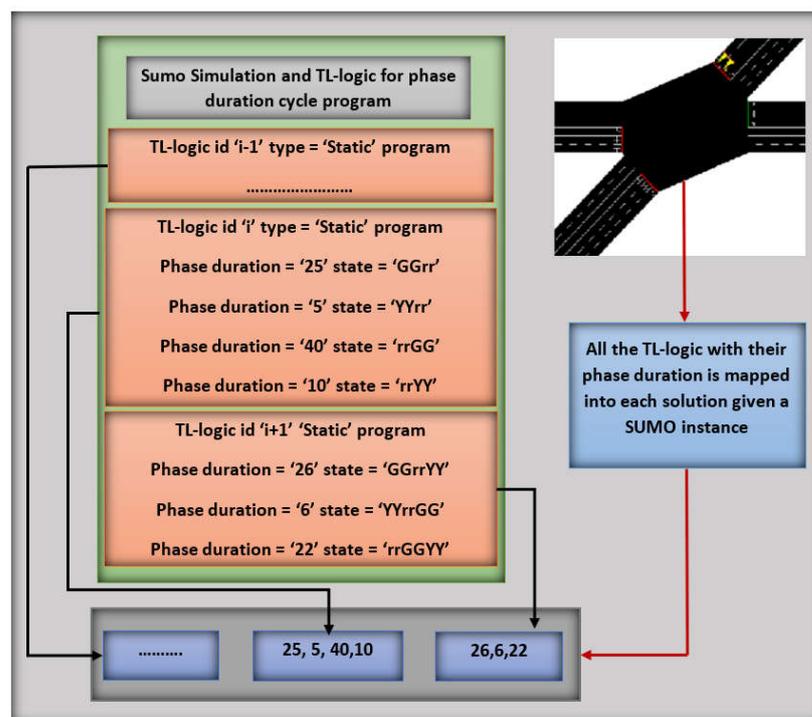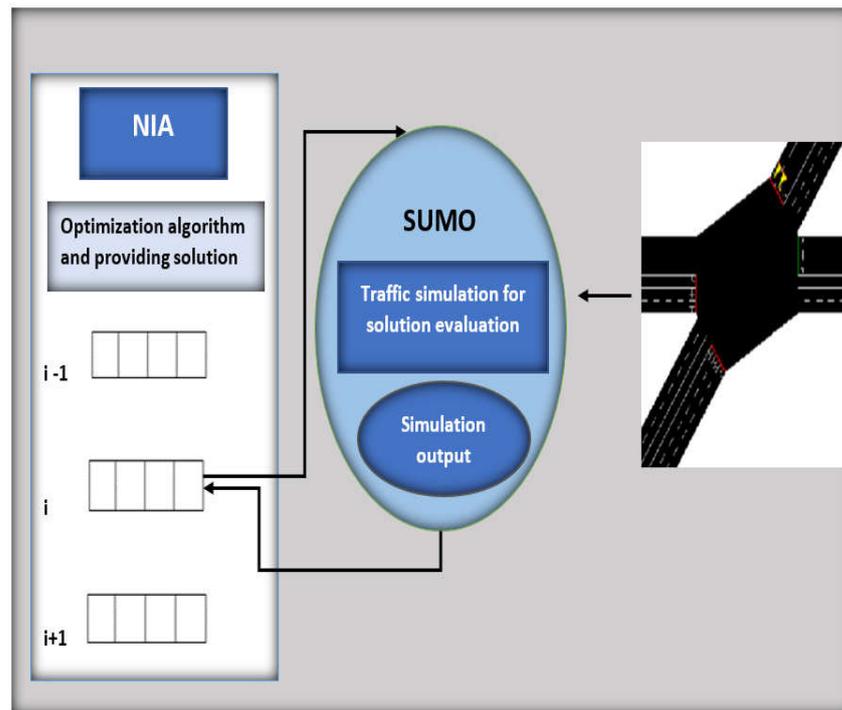

Figure 3. Traffic light junction using NIA.

Figure 4. Optimization strategy for traffic light cycle program configuration.

3.4. Fitness Function

The fitness function is defined as a function that takes a candidate solution to the problem as input and gives as output how to fit it for the problem. When analyzing each solution, the information is incorporated into the explanation, which generates the cycle program of the traffic light based on the occurrences that took place throughout the simulation by applying the formula below:

$$\text{Fitness} = \frac{\left(\sum_{a=0}^{A} Ov + \left(\sum_{a=0}^{A+C} W_v\right) + C \cdot S_t\right)}{A^2 + Cr} \tag{1}$$

The main goal is to increase the number of vehicles arriving at their destination *(A)* and simulation time *(S_t)* while decreasing the number of vehicles that do not arrive and continue to circulate *(C)*. Another goal is to reduce the overall duration of the vehicle's journey *(O_v)*, and W_v is the stop and waiting time of the vehicle. During the simulation, it is evident that the overall duration refers to how long it takes for the various vehicles to reach their final road destinations. Vehicles with unfinished journeys *(C_r)*, time spent in the simulation time *(S_t)*, and vehicles that arrive at their destination *(A^2)*. The number of vehicles that arrive at their destination is included in this formulation. To obtain the usually required color lights in phase length, states with more green lights on high-traffic roads should be preferred over states with more red signals on low-traffic streets. In each tl-logic stage state, the color calculation is done. Traffic signals light in green *(G)* and red *(R)* in phase state h and tl-logic are counted in G_{lm} and R_{lm}, respectively.

$$Cr = \frac{\left(\sum_{l=0}^{tl}\left(\sum_{m=0}^{ph} Slm\right) R_{lm}\right)}{G_{lm}} \tag{2}$$

4. Methodology

In this research, we have used two layered frameworks based on a nature-inspired algorithm and cloud and crowd computing. In layer 1, out of the NIA, we have implemented GA, PSO, and ACO for real-time road datasets using the SUMO environment.

Figure 5 describes the Two-layered architecture based on cloud-crowd based intelligent transportation system.
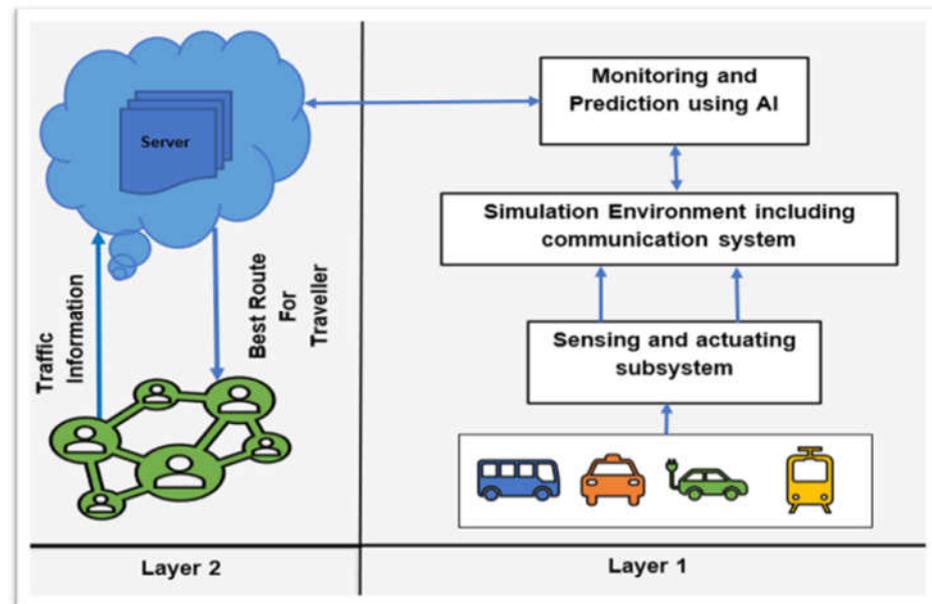
Figure 5. Two-layered architecture based on the cloud–crowd-based intelligent transportation system.

In layer 2, the combination of cloud and crowd computing framework is used. The real road network data are mounted on the cloud to initialize the road network. The SUMO simulation model predicts the tentative route for the source–destination pair based on the road network data. The users upload crowd-based information to update the route information, and finally, a decision is taken to refine the route with the combination of cloud and crowd computing. People in the crowd can update the information in the cloud regarding any mishappening or road accidents. Other users using that road or who would use that road network will be alerted beforehand to opt for a different, clear and safer route. Actual traffic data are used to build a network simulation model, and refinement is done through crowd-based information (through GPS). The intelligent NIA-based algorithm predicts the suitable path for the users with the synergy of cloud and crowd-based computational architecture.

5. Experimental Approach

The experimental approach used to evaluate the performance of our optimization problem is described in this section. First, we will go over the traffic light scenarios created expressly for this research work. The details of the implementation and parameters are discussed below.

5.1. Dataset

We created one scenario by extracting information from a real digital map of Ranchi, Jharkhand, India, as we are interested in constructing an optimization solver capable of coping with close-to-reality and generic metropolitan environments, such as traffic rules, traffic lights, road directions, streets, intersections.

5.2. Experimental Setup

A framework based on Python is employed to apply the nature-based method to the traffic optimization problem (GA, PSO, ACO). Executing the traffic simulator SUMO version 1.12 for Windows brings the simulation process to a successful conclusion. We carried out 30 separate iterations of our model for each unique case instance. The population

size was determined to be 100 particles, and there were 30 iteration steps in each run and instance. As a consequence, 30 different solutions were examined.

5.3. Results and Analysis

　　NIA such as GA, ACO and PSO algorithms have been used in traffic signal optimization techniques for creating successful traffic light programs for an area in Ranchi, Jharkhand (India). A simulation run on the same scenario using a conventional algorithm (Dijkstra) has been done to evaluate the performance comparison with NIA algorithms. In addition, an instance is taken from real road traffic to compare the performance of the considered NIA algorithms with real-time road traffic scenarios. Depending on the number of vehicles on the road junction and the number of traffic signals operational, a total of 30 distinct numerical iterations have been generated. Several analyses have been carried out from various perspectives on road junctions. It is observed that a 24% improvement in the flow of vehicles can be achieved using the proposed PSO, and 22% and 17% of improvements in the flow of vehicles have been achieved using ACO and GA, respectively. The performance measure considered here is the traffic flow determined on the same considered path. It indicates the improvements in congestion and waiting time. Performance and analysis at each iteration have been described in Figure 6, flow of vehicle, fitness and simulation values are described in Figures 7 and 8, respectively.
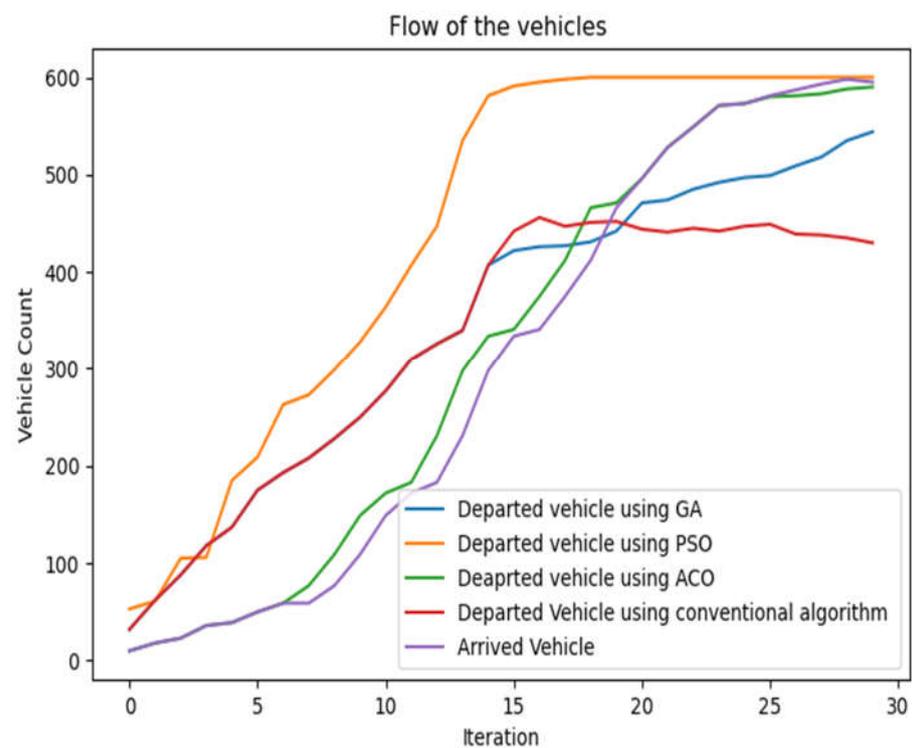
Figure 6. Flow of the vehicle using NIA's algorithm.

　　Iteration values are listed below in Table 1 concerning the simulation time (ST), vehicle count (VC), and fitness value (FV). The comparative results show that the PSO algorithm outperforms the ACO and GA algorithms. An increase in the vehicle route denotes a more significant number of vehicle inflow, which is intended for efficient transportation. PSO reaches its maximum vehicle count at the 19th iteration, whereas ACO and GA reach a maximum of 590 and 544 until the maximum iteration is considered the 30th iteration. Fitness value indicates how fit the system is in terms of flow at any instance, and the value is expected to be lower. The fitness value is improved in all three cases, and PSO and ACO exhibit comparative FV. It is also observed that PSO is better in terms of consumption than ACO and GA.
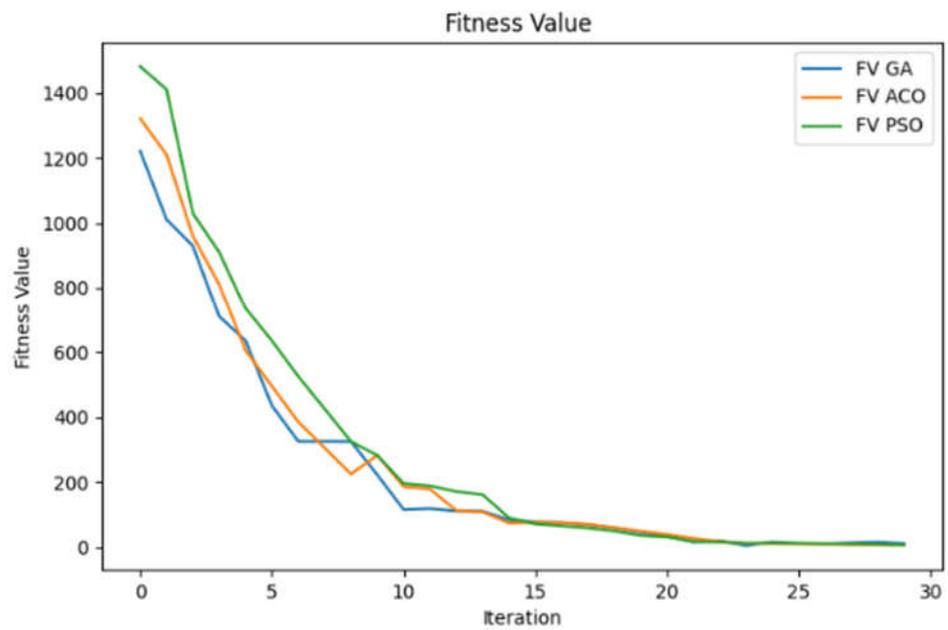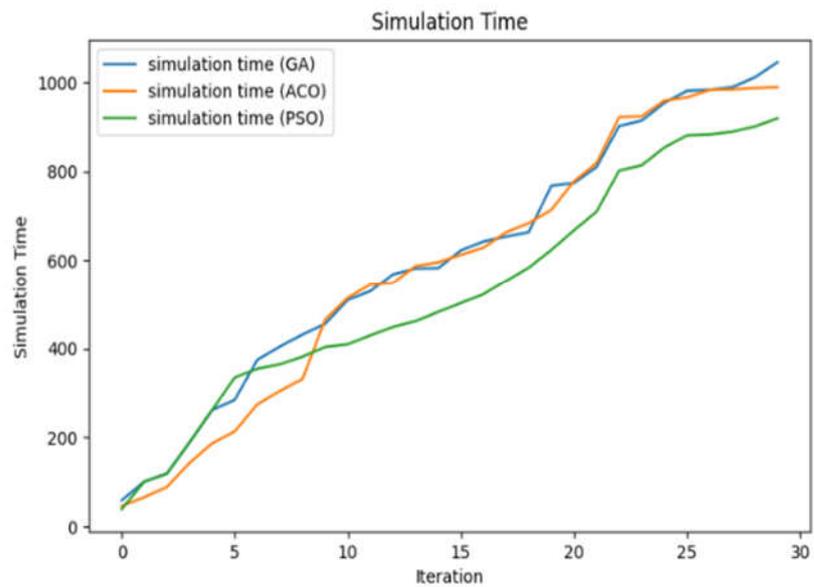
Figure 7. Fitness value at each iteration.

Figure 8. Simulation time at each iteration.

Table 1. Comparison of various listed NIA algorithms.

Iteration	PSO			ACO			GA		
	ST	VC	FV	ST	VC	FT	ST	VC	FV
1	39	53	1480.99	46	10	1320.92	59	32	1220.99
2	101	61	1410.07	66	18	1210.17	101	62	1010.07
3	119	105	1029.05	89	23	959.05	119	88	929.05
4	189	106	910.12	143	36	810.11	189	118	710.12
5	262	185	735.17	187	39	605.17	262	137	635.14
6	385	209	635.01	214	50	495.12	285	175	435.02
7	355	263	525.11	275	59	385.19	375	193	325.11
8	365	273	425.18	305	77	224.82	405	208	325.13

Table 1. *Cont.*

Iteration	PSO			ACO			GA		
	ST	VC	FV	ST	VC	FT	ST	VC	FV
9	382	299	324.72	332	109	282.14	432	228	324.72
10	404	328	282.11	466	149	185.95	456	250	222,11
11	410	364	195.45	515	172	178.34	510	277	115.45
12	430	407	188.34	545	183	112.28	530	310	118.34
13	449	447	170.88	549	231	108.08	569	326	110.88
14	462	535	160.87	588	298	73.89	582	340	110.07
15	483	581	89.69	596	334	76.17	583	407	83.69
16	503	591	71.17	613	341	75.03	643	422	77.37
17	523	595	65.03	629	375	69.98	654	426	75.03
18	654	598	58.78	664	412	59.77	674	427	68.78
19	584	600	49.08	684	466	49.18	724	431	59.02
20	624	600	35.28	714	471	38.69	768	442	45.78
21	668	600	31.11	778	496	26.72	774	471	33.59
22	710	600	17.05	819	528	17.49	810	474	16.02
23	802	600	15.39	922	549	11.03	902	485	19.69
24	814	600	11.08	924	571	10.48	914	492	4.03
25	854	600	12.78	959	573	9.58	954	497	15.48
26	881	600	11.78	966	580	8.62	981	499	10.98
27	883	600	10.92	983	581	7.54	983	509	10.02
28	889	600	9.64	984	583	8.68	989	518	13.24
29	901	600	8.68	987	588	6.38	1011	535	15.48
30	919	600	6.57	989	590	5.78	1045	544	10.98

6. Conclusions

This paper presents an optimization strategy for developing a traffic light program based on the nature-inspired algorithm. We investigated the flow of traffic in Ranchi, Jharkhand (India). The road traffic data are mounted in the CCCITS model depending on the number of vehicles that are present at a road junction as well as the number of traffic signals that are currently active. In order to transport the active vehicle in the considered network most effectively, a model that is based on cloud and crowd computing is combined with CCCITS. The paper's novelty is the consideration of NIA for routing purposes and feedback using crowd computing through the interface with the cloud. Several different junctions of the road have been analyzed from a variety of perspectives in order to carry out a number of calculations to determine the vehicle counts and fitness values. It has been observed that PSO can achieve approximately 24% of improvement, ACO can achieve approximately 22% of improvement, and GA can achieve approximately 17% of improvement. Our proposed CCCITS model is well-suited for urban areas and metropolitan cities to ensure the efficient flow of vehicles and transportation.

References

1. Hurtado-Gómez, J.; Romo, J.D.; Salazar-Cabrera, R.; Pachón de la Cruz, Á.; Madrid Molina, J.M. Traffic Signal Control System Based on Intelligent Transportation System and Reinforcement Learning. *Electronics* **2021**, *10*, 2363. [CrossRef]
2. Shawe-Taylor, J.; Bie, T.D.; Cristianini, N. Data mining, data fusion and information management. *IEE Proc-Intell. Transp. Syst.* **2006**. [CrossRef]
3. Zhang, J.; Wang, F.-Y.; Wang, K.; Lin, W.-H.; Xu, X.; Chen, C. Data-Driven Intelligent Transportation Systems: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 1624–1639. [CrossRef]
4. Nair, D.J.; Gilles, F.; Chand, S.; Saxena, N.; Dixit, V. Characterizing multicity urban traffic conditions using crowdsourced data. *PLoS ONE* **2019**, *14*, e0212845.
5. Srivastava, S.; Sahana, S.K. Application of Bat Algorithm for Transport Network Design Problem. *Appl. Comput. Intell. Soft Comput.* **2019**, *2019*, 1–12. [CrossRef]
6. Dobrota, N.; Stevanovic, A.; Mitrovic, N. Development of Assessment Tool and Overview of Adaptive Traffic Control Deployments in the U.S. *Transp. Res. Rec. J. Transp. Res. Board* **2020**, *2674*, 464–480. [CrossRef]
7. Garcia-Nieto, J.; Olivera, A.C.; Alba, E. Optimal Cycle Program of Traffic Lights with Particle Swarm Optimization. *IEEE Trans. Evol. Comput.* **2013**, *17*, 823–839. [CrossRef]
8. Jovanović, A.; Stevanović, A.; Dobrota, N.; Teodorović, D. Ecology based network traffic control: A bee colony optimization approach. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105262. [CrossRef]
9. Mohamed, S.A.E.; AlShalfan, K.A. Intelligent Traffic Management System Based on the Internet of Vehicles (IoV). *J. Adv. Transp.* **2021**, *2021*, 1–23. [CrossRef]
10. Shi, Y.; Qi, Y.; Lv, L.; Liang, D. A Particle Swarm Optimisation with Linearly Decreasing Weight for Real-Time Traffic Signal Control. *Machines* **2021**, *9*, 280. [CrossRef]
11. Mohamed, N.E.; Radwan, I.I. Traffic light control design approaches: A systematic literature review. *Int. J. Electr. Comput. Eng.* **2022**, *12*, 5355–5363. [CrossRef]
12. Nagy, J.; Lakatos, I. Possibilities of Using Online Vehicle Diagnostics in the Future. In *Vehicle and Automotive Engineering 4. VAE 2022. Lecture Notes in Mechanical Engineering*; Jármai, K., Cservenák, Á., Eds.; Springer: Cham, Switzerland, 2022. [CrossRef]
13. Lakatos, I.; Szauter, F.; Pup, D.; Nagy, A. Alternative Propulsion Buses in the Metropolitan Public Transport. In *Vehicle and Automotive Engineering 3. VAE 2020. Lecture Notes in Mechanical Engineering*; Jármai, K., Voith, K., Eds.; Springer: Singapore, 2021. [CrossRef]
14. Péter, T.; Szauter, F.; Rózsás, Z.; Lakatos, I. Integrated application of network traffic and intelligent driver models in the test laboratory analysis of autonomous vehicles and electric vehicles. *Int. J. Heavy Veh. Syst.* **2020**, *27*, 227–245. [CrossRef]
15. Lin, H.; Han, Y.; Cai, W.; Jin, B. Traffic Signal Optimization Based on Fuzzy Control and Differential Evolution Algorithm. *IEEE Trans. Intell. Transp. Syst.* **2022**, 1–12. [CrossRef]
16. Hosseinzadeh, M.; Sinopoli, B.; Kolmanovsky, I.; Baruah, S. MPC-Based Emergency Vehicle-Centered Multi-Intersection Traffic Control. *IEEE Trans. Control Syst. Technol.* **2022**, 1–13. [CrossRef]
17. Jia, H.; Lin, Y.; Luo, Q.; Li, Y.; Miao, H. Multi-objective optimization of urban road intersection signal timing based on particle swarm optimization algorithm. *Adv. Mech. Eng.* **2019**, *11*, 1687814019842498. [CrossRef]
18. Le, T.; Kovács, P.; Walton, N.; Vu, H.L.; Andrew, L.L.; Hoogendoorn, S.S. Decentralized signal control for urban road networks. *Transp. Res. Part C Emerg. Technol.* **2015**, *58*, 431–450. [CrossRef]
19. Eom, M.; Kim, B.-I. The traffic signal control problem for intersections: A review. *Eur. Transp. Res. Rev.* **2020**, *12*, 1–20. [CrossRef]
20. Le, T.; Vu, H.L.; Walton, N.; Hoogendoorn, S.P.; Kovács, P.; Queija, R.N. Utility optimization framework for a distributed traffic control of urban road networks. *Transp. Res. Part B Methodol.* **2017**, *105*, 539–558. [CrossRef]
21. Younes, M.B.; Boukerche, A. Intelligent Traffic Light Controlling Algorithms Using Vehicular Networks. *IEEE Trans. Veh. Technol.* **2015**, *65*, 5887–5899. [CrossRef]
22. Hunt, P.B.; Robertson, D.I.; Bretherton, R.D.; Winton, R.I. *SCOOT—A Traffic Responsive Method of Coordinating Signals*; No. LR 1014 Monograph; Transport and Road Research Laboratory (TRRL): Berkshire, UK, 1981.
23. Sims, A.G. The Sydney coordinated adaptive traffic system. In *Engineering Foundation Conference on Research Directions in Computer Control of Urban Traffic Systems*; SCAT: Pacific Grove, CA, USA, 1979.
24. Gartner, N.H. *OPAC: A Demand-Responsive Strategy for Traffic Signal Control (No. 906)*; Transport and Road Research Laboratory (TRRL): Berkshire, UK, 1983.
25. Mirchandani, P.; Head, L. A real-time traffic signal control system: Architecture, algorithms, and analysis. *Transp. Res. Part C Emerg. Technol.* **2001**, *9*, 415–432. [CrossRef]
26. Tassiulas, L.; Ephremides, A. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. In Proceedings of the 29th IEEE Conference on Decision and Control, Honolulu, HI, USA, 5–7 December 1990; pp. 2130–2132.
27. Weerasundara WA, G.; Udugahapattuwa DP, D.; Munasingha, T.D.; Gunathilake WD, K.; Dampage, E. An Adaptive and Coordinated Traffic Signal Scheme for Greener Transport 4.0. In Proceedings of the International Conference on Electrical and Electronics Engineering, Virtual, 27–18 August 2022; pp. 206–217.

28. Wongpiromsarn, T.; Uthaicharoenpong, T.; Wang, Y.; Frazzoli, E.; Wang, D. Distributed traffic signal control for maximum network throughput. In Proceedings of the 2012 15th International IEEE Conference on Intelligent Transportation Systems, Anchorage, AK, USA, 16–19 September 2012; pp. 588–595.

29. Gregoire, J.; Samaranayake, S.; Frazzoli, E. Back-pressure traffic signal control with partial routing control. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016; pp. 6753–6758.

30. Wu, J.; Ghosal, D.; Zhang, M.; Chuah, C.-N. Delay-Based Traffic Signal Control for Throughput Optimality and Fairness at an Isolated Intersection. *IEEE Trans. Veh. Technol.* **2017**, *67*, 896–909. [CrossRef]

31. Dobrota, N.; Stevanovic, A.; Mitrovic, N. Modifying Signal Retiming Procedures and Policies by Utilizing High-Fidelity Modeling with Medium-Resolution Traffic Data. *Transp. Res. Rec. J. Transp. Res. Board* **2021**, *2676*, 660–684. [CrossRef]

32. Srivastava, S.; Sahana, S.K. A survey on traffic optimization problem using biologically inspired techniques. *Nat. Comput.* **2019**, *19*, 647–661. [CrossRef]

33. Krajzewicz, D.; Bonert, M.; Wagner, P. *The Open Source Traffic Simulation Package SUMO*; RoboCup: Bordeaux, France, 2006.

34. Haklay, M.; Weber, P. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Comput.* **2008**, *7*, 12–18. [CrossRef]