RESEARCH ARTICLE

A Hybrid GASAPSO Algorithm for Traffic Signal Delay Minimization

Tanya Garg[1,*] **and Gurjinder Kaur**[2]

[1] Thapar Institute of Engineering and Technology, Patiala 147001, India
[2] Sant Longowal Institute of Engineering and Technology, Sangrur, Punjab 148106, India

Abstract

Traffic management is a crucial issue due to the massive increase in vehicles on the road. In order to improve traffic flow, reduce congestion, and ensure commuter safety at junctions, traffic light regulation is essential. This research aims to minimize delay and accident risk while maximizing vehicle throughput through intersections. In this paper, we introduced a new hybrid approach by combining various algorithms, which include a genetic algorithm (GA), Simulated Annealing (SA), and Particle Swarm Optimization (PSO) to reduce traffic delay at traffic lights. The Webster delay function has been used as a fitness function to evaluate the algorithmic performance. Our hybrid algorithm efficiently explores the solution space to pinpoint the ideal signal timings through extensive simulations and experiments, resulting in shorter vehicle wait times at traffic lights. Our proposed algorithm, GASAPSO, successfully reduced the traffic delay to 186.5 seconds at 250 iterations. Moreover, GASAPSO has shown the highest accuracy over other algorithms. PSO, GA, and SA work together to converge to superior solutions, increasing the overall effectiveness of the traffic control system. Finally, the performance of our proposed algorithm has been compared with the existing hybrid algorithms in the same context to validate our approach. The proposed algorithm has also been validated on real time traffic data for the Patiala City.

Keywords: traffic management, traffic signal, Webster delay, GASAPSO, delay minimization.

1 Introduction

Control over traffic on roads is a severe issue for several reasons. The threats to people's lives due to road congestion, which in turn is the cause of accidents, can be reduced by providing a suitable approach for deploying traffic management systems on roads. Traffic signals, roundabouts, and speed bumps are a few examples of how to manage traffic flow and lower the risk of collisions. Enhancing traffic flow and reducing the number of vehicles on the road can also aid in easing traffic congestion [1, 2]. Effective traffic management on roadways is crucial to balance the demand for transportation and the general public's safety, health, and welfare. Researchers have offered a variety of methods to improve traffic signals for traffic management, drawing on evolutionary algorithms and metaheuristic techniques [3–5]. According to studies [6, 7], GAs may successfully adjust the timing

Submitted: 23 October 2025
Revised: 14 November 2025
Accepted: 20 January 2026
Published: 06 April 2026

Vol. 2, **No.** 3, 2026.
10.62762/TMI.2025.542785

***Corresponding author:**
Tanya Garg
tanyagarg5023@gmail.com, tanya.garg@thapar.edu

Citation
Garg, T., & Kaur, G. (2026). A Hybrid GASAPSO Algorithm for Traffic Signal Delay Minimization. *ICCK Transactions on Machine Intelligence*, 2(3), 144–160.

© 2026 ICCK (Institute of Central Computation and Knowledge)

of traffic lights based on various data, including shorter wait times, higher average speeds, and lower emissions.

In order to effectively manage urban traffic, traffic light optimization is essential since it has a direct impact on both the movement of automobiles and people. Over the past few decades, substantial studies have been conducted on optimizing genetic algorithms (GAs) for traffic light management. GA is a type of heuristic optimization that imitates the concepts of genetics and natural selection. The GASAPSO hybrid blends the advantages of Simulated Annealing, Genetic method, and Particle Swarm Optimization to reduce traffic delays.

The main contributions are developing and evaluating the GASAPSO hybrid algorithm for traffic delay optimization. The algorithm seeks to reduce traffic delays and increase overall traffic flow efficiency by utilizing the capabilities of Genetic Algorithm, Simulated Annealing, and Particle Swarm Optimization.

The methodology section clearly states and discusses the algorithm proposed in this research work. All the equations related to the algorithms we combined to generate hybrid algorithms have also been discussed in that section. The experimental setup used to evaluate the effectiveness of the GASAPSO algorithm is described in the implementation section. The choice of performance metrics is described. The implementation specifics, such as the software tools and programming language used, are also addressed. The experimental results of evaluations have been presented in the results and discussion section. It examines and interprets the outcomes of using the GASAPSO algorithm in scenarios of actual traffic.

1.1 Motivation

Traditional traffic control techniques frequently fail to adequately adjust to changing traffic circumstances and optimize signal timings. Therefore, intelligent and flexible methods for reducing traffic delays are required. The goal of our work is to create a new hybrid algorithm that combines the advantages of Simulated Annealing (SA), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA) in order to overcome the difficulties associated with traffic delay optimization. Combining these algorithms into a hybrid algorithm can offer a comprehensive and reliable solution for traffic management systems because each has demonstrated success in other optimization domains. The GASAPSO algorithm has been developed due to its proficiency in negotiating complex solution spaces and adjusting to the ever-changing nature of optimization settings. Our traffic signal optimization task presents a complex set of challenges that can be effectively addressed by a flexible framework that combines elements of particle swarm optimization and genetic algorithms. Simulated annealing, which introduces a stochastic component that helps overcome local optima and improves the algorithm's convergence to globally optimal solutions, also lends a layer of relevance to our methodology. This synergistic combination of simulated annealing and GASAPSO reinforces the algorithm's effectiveness in handling the complexity of our optimization problem.

In the context of traffic signal control, understanding the key parameters that influence traffic flow is essential for improving readability and effectiveness. These parameters include phase length, cycle, offset, and random/uniform delays.

1. **Phase Length:** This refers to the duration of a particular traffic signal phase (e.g., green, yellow, or red) for a specific movement or direction. Properly adjusted phase lengths can optimize the flow of traffic and reduce waiting times.
2. **Cycle:** A traffic signal cycle is the complete sequence of all phases (e.g., green, yellow, red) for all directions at an intersection. The length of the cycle influences how long vehicles may need to wait before receiving a green signal again.
3. **Offset:** Offset is the time difference between the start of green phases at consecutive intersections. Coordinating offsets can help create a "green wave," allowing vehicles to move through multiple intersections without stopping.
4. **Random/Uniform Delays:** These are variations in signal timings. Random delays introduce variability in phase lengths, simulating unpredictable real-world conditions. Uniform delays apply consistent adjustments, useful for testing systematic changes in traffic flow.

By clearly understanding these parameters, we can develop more effective strategies to manage traffic, reduce congestion, and improve overall efficiency.

1.2 Contribution

This research has been done to optimize algorithms to reduce traffic delays. The following points summarize the contribution of our proposed work:

1. It has developed a hybrid approach that combines Simulated Annealing (SA), Genetic approach (GA), and Particle Swarm Optimization (PSO) to reduce traffic delay at traffic lights.
2. The Webster delay function has been used as the fitness function to validate the algorithms in minimizing delay.
3. Adaptively adjust signal timings and enhance traffic control by incorporating dynamic signal timing based on real-time traffic flow data.
4. Comparison graphs showing the higher performance of the hybrid algorithm were presented together with convergence graphs for each method (PSO, GA, and SA).
5. Extensive simulations used to validate the algorithm showed how effectively it reduced traffic delays, enhanced traffic flow, and improved road safety.

The construction of fitness functions, the use of numerous algorithms, performance evaluation, and convergence analysis all offer critical new perspectives on the usefulness of the suggested framework.

1.3 Organization

The beginning of the study gives a thorough review of the significance of traffic delay minimization in traffic management systems. In order to enhance traffic flow, decrease delays, and improve road safety, it emphasizes the difficulties and limitations of conventional procedures and the necessity of intelligent and adaptive approaches. It focuses on developing a novel hybrid algorithm called GASAPSO (Genetic Algorithm, Simulated Annealing, and Particle Swarm Optimization) and outlines the purpose of the work.

Initially, we started with the Introduction section. After this section, a literature review was conducted to explore various researchers' research in the same area and get insights for further improvements.

The methodology section clearly states and discusses the algorithm proposed in this research work. All the equations related to the algorithms we combined to generate hybrid algorithms have also been discussed in that section. The experimental setup used to evaluate the effectiveness of the GASAPSO algorithm is described in the implementation section. The choice of performance metrics is described. The implementation specifics, such as the software tools and programming language used, are also addressed. The experimental results of evaluations have been presented in the results and discussion section. It examines and interprets the outcomes of using the GASAPSO algorithm in scenarios of actual traffic.

The primary conclusions of the work done have been outlined in the conclusion section, along with their consequences. It highlights the improvements of a hybrid algorithm for traffic delay optimization and its potential for use in the real world. The limitations of this work have been acknowledged, and further research directions have been proposed in the last section.

2 Related Work

Several researchers have proposed various approaches to optimize traffic signal control and improve traffic flow. Starting with Wen [1], a dynamic automatic traffic signal control expert system for multiple intersections was developed, demonstrating improved waiting times for vehicles. Similarly, Turky et al. [2] employed a genetic algorithm for traffic light control incorporating pedestrian crossing constraints, though both works focused primarily on vehicles coming from a single direction. In a related effort toward more interpretable signal control, Liu et al. [8] proposed an explainable traffic signal control approach for urban networks using genetic programming (GP). Their method automatically evolves interpretable signal control policies and evaluates performance across multiple urban intersection configurations, demonstrating competitive results compared to conventional signal control strategies. Nevertheless, a limitation of their study is that the evolved GP-based controllers may lack straightforward transferability to highly heterogeneous or large-scale urban networks without retraining.

In a different vein, Dezani et al. [9] proposed a genetic algorithm based on the Petri net model of urban traffic flow, which captures the movement of vehicles through road networks and intersections. They aimed to optimize overall traffic flow and reduce travel time by simulating different routes and evaluating their performance. Wijaya et al. [10] conducted a case study on the Toroku road network to optimize traffic signal control parameters using particle swarm optimization (PSO). Their research aimed to optimize parameters such as offset, cycles, and split time using a multi-element genetic algorithm. The results demonstrated the superiority of PSO over

other algorithms. However, it is essential to note that this study was limited to a specific road network, reducing its generalizability to other scenarios.

Zhao et al. [11] proposed an approach for specific roads in the region of China for combining phases of traffic lights using the Particle Swarm Optimization Algorithm in which they calculated the number of stops in the traveler's route. The approach was to decrease the number of stops at the intersection of roads, which they successfully reduced. This phase combination technique has also reduced the delay and carbon monoxide emissions. The limitation of their proposed approach is that the sequence of phase changes has not been considered, which is why the performance could have been optimized to the required extent. Verma et al. [12] proposed a modified Webster delay function to optimize the traffic signal delay for Bangalore City. The proposed approach, Probably First-In-Probably-First-Out (PFIPFO) queuing vehicles for signalized intersections, performed better in India for specific junctions in Bangalore city over the First-in-First-Out queuing approach. This work was also limited to the city road network and not generalizable to other urban road networks. Zhao et al. [13] proposed a multi-objective signal timing optimization model for road intersections using an improved particle swarm optimization (PSO) algorithm. The model minimized average vehicle delay, slow traffic delay, and vehicle stopping times simultaneously. The experimental results demonstrated that the improved PSO effectively enhanced intersection operational efficiency compared with conventional fixed-time signal plans. Cruz-Piris et al. [14] similarly employed a genetic algorithm to automate the optimization of intersection signal timings, demonstrating reduced vehicle delays and improved traffic flow across multiple intersection configurations. Ferrer et al. [15] performed a comparison of various algorithms for traffic signal optimization for real-world road scenarios. The comparative analysis of five algorithms, Iterated Racing for Automatic Algorithm (IRACE), a Genetic Algorithm (GA), a Differential Evolution (DE) and Particle Swarm Optimization (PSO) has been done for the same problem of traffic signal optimization, and the IRACE algorithm performed best out of all these five algorithms. The work was limited to city scenarios only and unsuitable for different traffic scenarios. Chen et al. [16] proposed a GAPSO approach that integrates genetic algorithms with particle swarm optimization to obtain optimized traffic signal timing plans at isolated intersections with multiple objectives, demonstrating the effectiveness of hybrid GA-PSO strategies for signal control. Liang et al. [17] proposed a deep reinforcement learning (DRL) network for traffic light cycle control, in which a convolutional neural network (CNN) was employed to extract traffic state features and generate signal control decisions. The approach was validated using the Simulation of Urban Mobility (SUMO) tool and demonstrated improved performance over conventional fixed-time control strategies. Iskandarani [18] proposed an improved genetic algorithm combined with a balancing technique (GABT) to optimize traffic signal parameters, though this approach was limited to a single intersection only.

Bhattacharyya et al. [19] conducted a systematic review of evolutionary and swarm intelligence approaches for traffic signal control optimization. The review covers a wide range of metaheuristic methods, including genetic algorithms, particle swarm optimization, ant colony optimization, and differential evolution, evaluating their effectiveness in minimizing traffic delays and improving signal timing at urban intersections. Celtek et al. [20] proposed Swarm Optimization methods for real-time traffic signal control using the dataset available from the Kilis city of Turkey using the Simulation of Urban Mobility Tool (SUMO). The comparison has been done with the collected initial data in numerical form. The social learning PSO algorithm has optimized the traffic signal control parameters.

Liu et al. [21] proposed a switching-based dynamic programming approach for signal optimization to manage road traffic. The switching means when to change the phase of the lights from red to green and so on. The simulation must be carried out in a real-time scenario using the SUMO tool. The proposed approach was not well suited to real road networks; it was a theoretical approach. Dimitrov proposed optimizing the phase length and cycle of two isolated traffic light intersections. The average weighted delay has been calculated and evaluated to measure the performance of the proposed approach [22].

Tsai et al. [23] proposed a heuristic hybrid algorithm for traffic signal scheduling. The grey wolf and grasshopper optimization (GWGO) algorithm was created to take advantage of the strengths of both techniques. The basic idea was to utilize the grasshopper optimization to dynamically change the convergence speed of the search algorithm

while employing the grey wolf optimization to prevent slipping into the local optimum too rapidly. Similarly, Dell'Orco et al. [24] applied an artificial bee colony (ABC)-based algorithm to optimize traffic signal timings, leveraging swarm-based foraging behavior to efficiently explore the signal timing solution space and reduce vehicle delays at signalized intersections. Zhang et al. [25] proposed a traffic signal timing optimization method based on a modified Webster delay function. Using the signal cycle length and the proportion of green light duration as independent variables, the method was formulated as a single-objective optimization problem and solved with an improved genetic algorithm, achieving approximately 15.64% reduction in average intersection delay. Furthermore, Lin et al. [26] proposed a hybrid approach that integrates fuzzy control theory with an adaptive sequencing mutation multi-objective differential evolution algorithm (FASM-MDEA) to optimize traffic signal timings. Their method effectively handles the nonlinear and stochastic characteristics of traffic flow, providing another perspective on combining fuzzy logic with evolutionary algorithms for signal optimization. In addition, recent studies have explored artificial intelligence techniques for delay estimation under heterogeneous traffic conditions. Ranpura et al. [27] developed multiple machine learning models, including Support Vector Regression (SVR), K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF), and Artificial Neural Network (ANN), to estimate vehicle control delay at isolated signalized intersections. Their work is particularly relevant for heterogeneous traffic scenarios similar to those observed in Indian cities.

Zhang and Su [28] proposed an optimized traffic light control approach for heterogeneous traffic scenarios for the Singapore road network using the Visual Simulation Tool (VISSIM). Linear and non-linear models have been used for signalized and non-signalized intersections separately. This work has not considered the primary parameters cycles, splits, and offsets. This approach did not consider different types of vehicles in the heterogeneous traffic flow which was the limitation of this approach [29]. In contrast, signal timing optimization approaches that account for specific lane configurations, such as contraflow left-turn lanes, have demonstrated the importance of incorporating detailed geometric and vehicle-type parameters into delay-based optimization models [30]. Liang et al. [31] proposed a decentralized arterial traffic signal optimization framework that leverages connected vehicle information to dynamically adjust signal timings. The approach coordinates signals across multiple intersections to minimize vehicle delays on arterial roads, demonstrating improved performance over fixed-time control strategies. However, the method's effectiveness depends on the penetration rate of connected vehicles, limiting its applicability in areas with low connectivity infrastructure.

Zhang et al. [35] proposed a multi-objective optimization approach for traffic signal timing at typical urban junctions using genetic algorithms. The method simultaneously optimizes traffic capacity, average delay, and vehicle emissions across multiple signal phases. Experimental results demonstrated that the GA-based approach effectively balances these competing objectives and outperforms conventional fixed-time signal plans in terms of overall intersection efficiency. Lee et al. [36] proposed a real-time optimization framework for adaptive traffic signal control using genetic algorithms. The method dynamically adjusts signal timings based on current traffic conditions to minimize overall vehicle delay. The proposed GA-based approach was validated through simulation and demonstrated significant reductions in average intersection delay compared with fixed-time signal plans, highlighting the advantage of real-time adaptive control in managing varying traffic demand.

Mao et al. [37] proposed a boosted genetic algorithm framework that integrates machine learning with GA for traffic signal timing optimization at signalized urban intersections. The method was specifically designed to handle non-recurrent traffic incidents and minimize total network travel time. The results demonstrated that the hybrid machine learning–GA approach outperformed standalone GA in both convergence speed and solution quality. Desmira et al. [38] designed a smart traffic light system using a fuzzy logic-based microcontroller. Ultrasonic and infrared sensors provide real-time traffic density inputs to activate green signals and relieve road congestion, demonstrating how sensor-driven fuzzy control can reduce delays without major infrastructure changes. Liu and Ding [39] proposed a distributed deep reinforcement learning method for traffic light control under varied traffic scenarios, demonstrating improved performance over centralized control approaches. Li and Schonfeld [40] developed a hybrid simulated annealing and genetic algorithm for

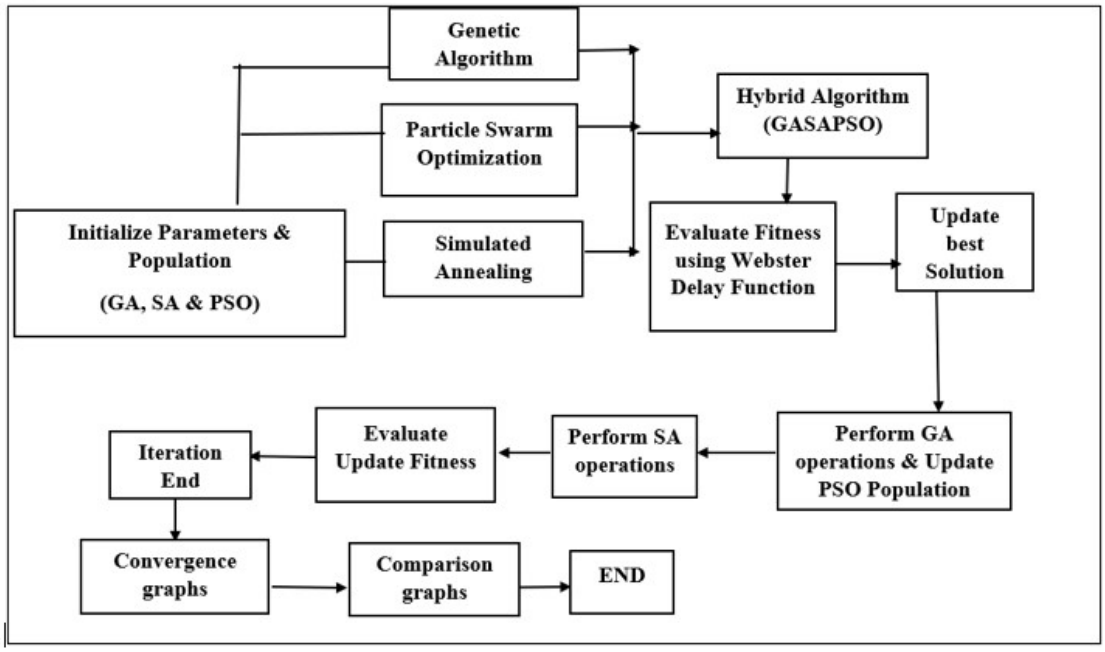


Figure 1. Workflow of proposed approach.

Table 1. State of Art for the delay minimization at traffic signal.

Authors/Year	Technique Used	Description of Algorithm	Application/Domain
Lee et al. [32]	Genetic Algorithm	A genetic algorithm was employed to determine optimal time-of-day break points for coordinated actuated traffic signal systems to minimize total delay.	Intelligent Transportation Systems
Chen et al. [16]	GAPSO	Genetic algorithm (GA) is integrated with the particle swarm optimization (PSO) algorithm to obtain optimized traffic signal timing plan.	Traffic Signal Control
Liang et al. [31]	Decentralized Arterial Signal Optimization	A decentralized optimization method using connected vehicle information optimized arterial traffic signal timings and minimized vehicle delays.	Intersection Management
Jahangiri et al. [33]	Simulated Annealing	A simulated annealing algorithm was employed to optimize traffic signal timing for emergency evacuation and minimize overall network delay.	Traffic Network Optimization
Hu and Chen [34]	Greedy Randomized Tabu Search	A greedy randomized tabu search algorithm optimized traffic signal timings at network level and minimized delays.	Sustainable Transportation
Dell'Orco et al. [29]	Harmony Search	A harmony search algorithm was applied to optimize signal timings and reduce vehicle delays.	Adaptive Traffic Control
Shaikh et al. [7]	Swarm Intelligence and Evolutionary Algorithms (Survey)	A comprehensive review of swarm intelligence and evolutionary algorithms for solving the traffic signal control problem, covering PSO, GA, and bee-inspired methods.	Traffic Signal Control (Survey)

optimizing arterial signal timings under oversaturated traffic conditions, showing that the hybrid approach outperforms either method applied alone. Tran et al. [41] proposed constrained optimization solutions for traffic signal timing at isolated urban intersections, accounting for environmental impacts and the needs of non-motorized vehicles. A metaheuristic is an advanced approach to problem-solving that directs and synchronizes the investigation of possible resolutions. Zhang et al. [42] proposed a hybrid constraint strategy combined with the NSGA-III framework (HCNSGA-III) to address the multi-objective optimization of traffic signal timing at road intersections. The approach simultaneously optimizes vehicle delay, intersection capacity, and vehicle emissions under various traffic demand scenarios, demonstrating that the hybrid multi-objective evolutionary strategy outperforms conventional fixed-time signal plans.

We have observed the progression in traffic signal delay minimization techniques. The researchers explored different aspects such as multiple intersections, GAs, Petri net models, and PSO to improve traffic flow and reduce congestion. However, there is still room for improvement, particularly in considering crucial parameters, encompassing traffic from both directions

and evaluating the performance in various traffic scenarios.

The work carried out by various researchers on delay minimization at traffic lights has been summarized in Table 1.

3 Proposed Approach

Our proposed approach is focused on optimizing traffic delays in traffic management systems, specifically at traffic lights and road intersections, using a hybrid algorithm that combines Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Simulated Annealing (SA). The decision to use PSO, GA, and SA in the hybrid algorithm was based on their specific characteristics and capabilities that make them suitable for traffic delay minimization in traffic management systems. Figure 1 represents the block diagram for the proposed methodology. Beginning with parameter and population initialization, the workflow proceeds through fitness evaluation using the Webster Delay function, best solution update, GA operations, PSO population update, and SA operations. After updated fitness evaluation and convergence checking, the algorithm generates convergence and comparison graphs before terminating.

3.1 Experimental Setup and Workflow

Extensive experiments have been conducted to assess the performance of the hybrid algorithm. The objective is to minimize waiting times for individuals by selecting the best solutions based on the Webster delay function, which serves as the fitness function. The proposed algorithm has been validated by performing a comparative analysis with the three algorithms explicitly used for optimization problems. The MATLAB code has been designed for traffic delay minimization using GA, and the same procedure has been repeated for the other three algorithms.

3.2 Parameter Initialization

The parameters used and initialized in the experimental work have been summarized in Table 2. Our hybrid algorithm demonstrates superior performance in minimizing traffic delays, highlighting its effectiveness in traffic management.

3.3 Algorithm Implementation Procedure

The proposed hybrid algorithm's implementation procedure has been outlined in Figure 2. A Hybrid algorithm for traffic light timing minimization uses the three algorithms to change the timing of traffic lights to improve road traffic and minimize the congestion on roads in urban areas.

Table 2. Parameters initialization.

Parameter	Values
No. of Traffic Lights	[4]
No. of Intersections	[2]
Effective green time (ge) (seconds)	[20, 60]
Cycle Length (seconds)	[60,180]
Road Capacity NSWE (random)	[18, 20, 25, 28]
Selection Pressure (beta)	[8]
Population	100
No. of Iterations	1000
Algorithms used	[Genetic, Simulated Annealing, Particle Swarm Optimization]
Crossover Rate	0.6
Mutation Rate	0.2
Temperature	300
Cooling Factor	0.85

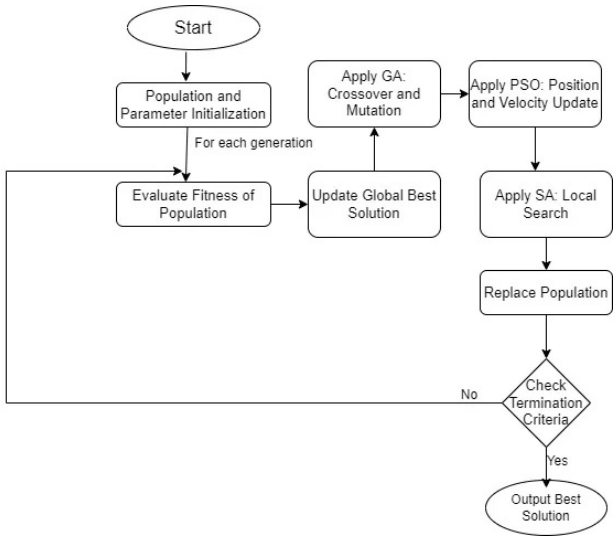


Figure 2. GASAPSO (hybrid algorithm) workflow.

The algorithm works by creating a population of possible solutions (in this case, traffic light timing plans) and then evaluating the fitness of each solution based on specific criteria (such as traffic flow and delay). The best-performing solutions are then combined to develop the solution for the next generation, and the same process is done until an optimal solution is found. This approach is beneficial for problems with many variables and a high degree of complexity, as it allows for a more efficient search for a solution than traditional methods. A multi-phase optimization process for traffic delay minimization is used to create GASAPSO. Using a GA, a population of viable solutions was first created. Traffic signal timings are then encoded, and the Webster delay function has been used to evaluate each candidate's

fitness. Then, an iterative optimization procedure is carried out, utilizing the Simulated Annealing (SA) technique to explore solution spaces and avoid local optima probabilistically. Using genetic operators, the GA phase continues to improve solutions. Particle Swarm Optimization (PSO) is used in tandem, using particles to explore the solution space using past and present global best solutions. Integrating GA, SA, and PSO data is coordinated to identify the best option for reducing traffic delay, resulting in a well-rounded and cooperative traffic optimization strategy. The optimization of parameter tuning and integration procedures are essential for improving the effectiveness of the hybrid algorithm.

3.4 Description of Algorithms

Here in this section, the algorithms that have been used to implement the hybrid (GASAPSO) Algorithm have been discussed:

3.4.1 Particle Swarm Optimization (PSO)

PSO is a population-based optimization method that draws inspiration from the social behavior of fish schools and bird flocks. It is renowned for its capacity for rapid solution space exploration and global optimum discovery. PSO is particularly good at managing continuous search areas and may arrive at solid conclusions quickly. By modifying the particle locations and velocities, PSO can assist in determining the best traffic signal timings in the context of traffic delay minimization.

$$x_i(t+1) = x_i(t) + v_i(t) \tag{1}$$

In this equation, $x_i(t+1)$ represents the updated position of particle i at Time $t+1$, $x_i(t)$ is the current position of particle i at time t, and $v_i(t)$ denotes the velocity of particle i at time t. The equation describes the iterative process of updating the particle's position based on its current position and velocity.

Particle Velocity Update Equation

$$\begin{aligned} v_i(t+1) = w \cdot v_i(t) + c_1 \cdot \text{rand}(t) \cdot (\text{pbest}_i(t) - x_i(t)) \\ + c_2 \cdot \text{rand}(0) \cdot (\text{gbest}(t) - x_i(t)) \end{aligned} \tag{2}$$

In this equation, $v_i(t+1)$ represents the updated velocity of particle i at Time $t+1$, $v_i(t)$ is the current velocity of particle i at time t, $\text{pbest}_i(t)$ denotes the personal best position of particle i at time t, $x_i(t)$ is the current position of particle i at time t, $\text{gbest}(t)$ represents the global best position at time t, w is the inertia weight, c_1 and c_2 are acceleration coefficients, and $\text{rand}(0)$ generates a random value between 0 and 1. This equation describes the update of particle velocity in a particle swarm optimization algorithm.

3.4.2 Simulated Annealing

The metallurgical annealing process served as the inspiration for SA, a probabilistic optimization technique. In order to leave local optima and explore the solution space, it regulates the acceptance of poor solutions using a temperature parameter. SA is beneficial when dealing with challenging or intricate environments where conventional optimization techniques could become stuck. SA can offer a method for exploring various signal timing configurations in traffic delay minimization and gradually converge to give better results.

Acceptance Probability Equation

$$P = \exp\left(\frac{(E_{\text{new}} - E_{\text{current}})}{T}\right) \tag{3}$$

The acceptance probability P is determined based on the energy differential $(E_{\text{new}} - E_{\text{current}})$ between the new solution and the current solution, as well as the temperature T. The solution space can be explored because of the higher acceptance probability that results from more considerable energy differences and temperatures.

Temperature Update Equation

$$T = T \times \text{cooling_factor} \tag{4}$$

In this equation, T represents the temperature at a given iteration, and cooling_factor is the factor by which the temperature is reduced. This equation is often used in simulated annealing algorithms to decrease the temperature as the algorithm progresses, mimicking the annealing process in metallurgy. The temperature is updated by multiplying it with a cooling factor at each iteration. The cooling factor determines how fast the temperature decreases, controlling the balance between exploration and exploitation.

3.4.3 Genetic Algorithm (GA)

GA is a population-based optimization algorithm inspired by natural selection and genetic evolution. It uses genetic operators such as selection, crossover,

and mutation to evolve a population towards better solutions iteratively. GA is suitable for problems with discrete search spaces and can handle multiple objectives. In the case of traffic delay minimization, GA can be used to evolve the population of signal timing solutions and improve their fitness over time. Here are the steps used for the GA Implementation. The various symbols, along with their meanings used for the proposed approach have been described in Table 3.

Table 3. Terminology and symbols used.

Symbol	Meaning
TDi	Traffic Delay (seconds)
C	Cycle Length (seconds)
ge	Effective green light time
x	Arrival flow rate (vehicles/second)
c	capacity
v	Vehicle arrival rate
X	(v/c) ratio (flow to capacity ratio)
s	Saturation flow rate
V	Number of vehicles

i. Initialize the population, lower and upper bound of adequate green light (ge), and number of vehicles from all four directions (North, South, West, and East). A population of N individuals will be generated using GA. The population has been initialized as shown in equation (1) below:

$$pop(i).\text{GreenNSWE} = \text{rand}([\text{greenMin},\ \text{greenMax}],\ \text{VarSize}) \tag{5}$$

where $i = 1, 2, \ldots, N$ and green Min and green Max will be randomly chosen according to defined lower and upper bounds. The crossover criterion followed is as shown in the equation below:

$$i = \text{rand}([1\ 3]) \tag{6}$$

Here, the `rand()` function in MATLAB generates the n-by-n co-distributed matrix for the random integers distributed uniformly as per the defined range.

ii. Determine the selection probability: The fitness function for calculating total delay was calculated using the Webster Delay function, which considers uniform and random delays. The delay for each individual generated has been calculated and evaluated against the stopping criteria. Finally, the individuals with higher fitness function values have been used to reproduce more individuals. So, selection probability has been calculated using the formula as shown in the equation at iteration g:

$$P = \exp\left(-\beta \cdot \frac{\text{TotalDelay}}{\text{WorstDelay}}\right) \tag{7}$$

$$P = P / \sum P \tag{8}$$

iii. Perform mutations of the selected particles: The mutation for each particle generated at different iterations after crossover suffers from diversity loss. A mutation rate of 0.2 has been set for the mutations, which is considered appropriate for the GA in this study. Initially, an empty matrix was created to contain mutants. The mutation has been done at k steps as per the equation 9 below:

$$popm(k).\text{TotalDelay}(j) = \text{FitnessFunction}(\text{currentcycle}, popm(k).\text{GreenNSWE}(j), \text{RoadCongestionNSWE}(j), \text{RoadcapacityNSWE}(j)) \tag{9}$$

In this equation, for a given index k in the population ($popm$), and a specific index j, the total delay TotalDelay(j) is calculated using the `FitnessFunction`. The arguments passed to the function are `current_cycle`, `popm. GreenNSWE`, `popm(k)`, `GreenNSWE(j)`, `RoadCongestionNSWE(j)`, and `RoadcapacityNSWE`. `RoadcapacityNSWE(j)` represents the current cycle, green light timings, road congestion, and road capacity. The equation is likely part of an optimization process where the fitness function has been used to evaluate the performance of a solution in a multi-objective or constrained optimization problem.

3.5 Fitness Function

Our proposed approach used the Webster Delay Function as the fitness function. Both uniform and random delays have been considered to calculate the total delay using this objective function. The Webster Delay function calculates the delay experienced by vehicles traveling through a specific area or network, and it typically considers factors such as travel time, vehicle density, and speed. The specific form of the function will depend on the context in which it is being used. However, in general, the function will be used to calculate the total delay experienced by vehicles over a specific period or distance. The Webster delay function is commonly used in transportation engineering and

traffic management to evaluate the performance of different traffic control strategies and to optimize traffic flow.

$$\text{FitnessFunction} = @(C, ge, x, c)\text{TDi}(C, ge, x, c) \quad (10)$$

$$\begin{aligned} &pop(i).\text{TotalDelay}(j) \\ &\quad = \text{FitnessFunction}(\\ &\quad\quad \text{current_cycle}, \\ &\quad\quad pop(i).\text{GreenNSWE}(j), \\ &\quad\quad \text{RoadCongestionNSWE}(j), \\ &\quad\quad \text{RoadcapacityNSWE}(j)) \end{aligned} \quad (11)$$

$$\text{TotalDelay} = (d1_i + d2_i) \quad (12)$$

where $d1_i$ is the uniform delay, and $d2_i$ is the random delay calculated using the Webster Delay function.

4 Results and Discussions

The results demonstrate the effectiveness of the hybrid algorithm as it performed better than the individual algorithms in terms of minimizing traffic delays, showcasing its effectiveness in traffic management.

4.1 Comparison of the proposed algorithm based on the number of iterations

Figure 3 (a), (b), (c) and (d) clearly show the performance of the Genetic, Simulated Annealing, PSO, and Proposed hybrid (GASAPSO) algorithms for the given number of iterations, minimization of traffic delay as the number of iterations increases.

Figure 3 (a) presents the traffic delay obtained using the Genetic Algorithm, with a peak delay value of 189 seconds. Figure 3 (b) presents the traffic delay obtained using the Simulated Annealing Algorithm, with a peak delay value of 190.5 seconds. Figure 3 (c) presents the traffic delay obtained using the Particle Swarm Optimization Algorithm, with a peak delay value of 190 seconds. Figure 3 (d) presents the traffic delay obtained using our proposed hybrid algorithm, GASAPSO, which achieves the lowest peak delay value of 186.5 seconds. This result demonstrates that our proposed algorithm successfully minimizes traffic delay and outperforms the other algorithms. Therefore, we conclude that GASAPSO is effective in reducing traffic delay.

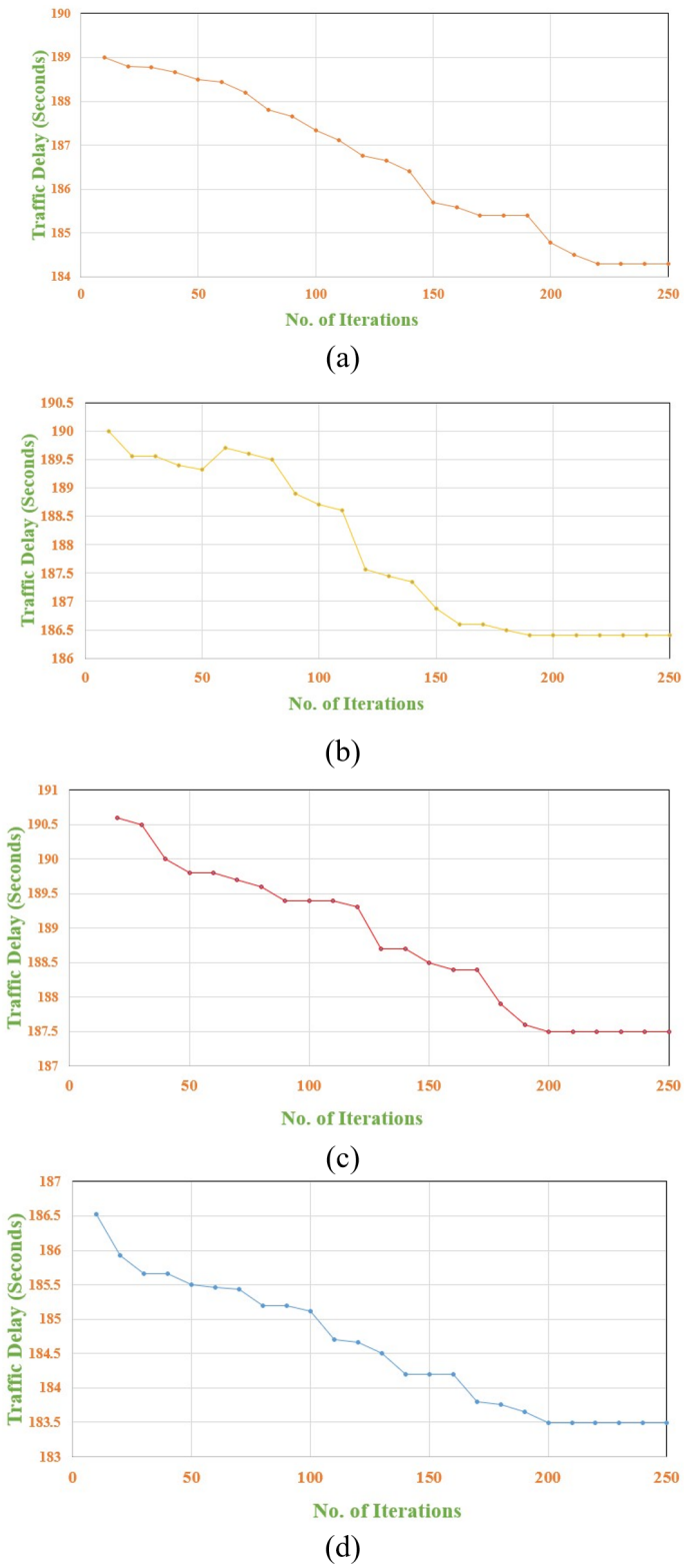


Figure 3. (a) Traffic Delay using Genetic Algorithm; (b) Traffic Delay using Simulated Annealing Algorithm; (c) Traffic Delay using PSO; (d) Traffic Delay using GASAPSO.

4.2 Convergence Rate of GASAPSO

The graph in Figure 4 shows the fitness value, representing the minimization of traffic delay using the Webster delay function, for the GASAPSO hybrid

algorithm and the GA, SA and PSO algorithms.

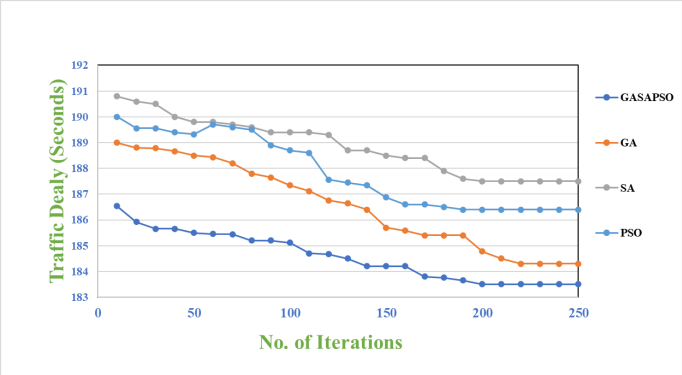


Figure 4. Delay using GASAPSO.

Our proposed algorithm achieves the minimum traffic delay at 250 iterations, while other algorithms exhibit higher fitness function values, indicating greater traffic delay. It can be observed that our hybrid algorithm performed better than the other algorithms, as it successfully minimized the traffic delay. The Table 4 shows the comparison of convergence rate of all the algorithms with the minimum value of delay they calculated.

Table 4. Convergence rate of algorithms.

Algorithms	Minimum Traffic Delay (Seconds)	Convergence Rate
GA	184.3	0.13
SA	187.5	0.14
PSO	186.4	0.16
GASAPSO	183.5	0.20

Figure 5 shows the convergence rate of all the algorithms at 250 iterations and it can be observed that our proposed hybrid algorithm GASAPSO has the highest convergence rate as compared to other algorithms. This high convergence rate of our proposed algorithm indicates that GASAPSO is extremely efficient as it achieves an ideal or near-optimal solution faster than other algorithms.

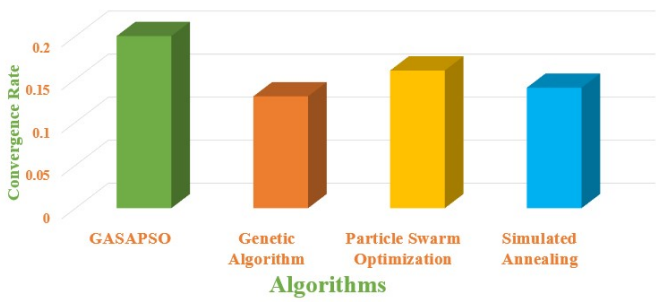


Figure 5. GASAPSO comparison with other algorithms.

As a result, the method produces answers more quickly and at a lower computing cost. It is also appropriate for real-time applications where judgments must be made quickly, like traffic control. Fast convergence rates shouldn't, however, lead to premature convergence, in which case the algorithm might accept less-than-ideal answers without thoroughly examining the solution space.

4.3 GASAPSO (Proposed Algorithm) Validation with Real Time Traffic Data

Finally, the real time traffic data has been collected for one of the famous and busiest city Patiala for the validation of our proposed algorithm. The proposed algorithm GASAPSO has been validated on the collected real time traffic data of the area near new bus stand Patiala. This area is highly congested and vulnerable to accidents. We collected the morning peak hour traffic data coming from four directions. One from Rajpura side, one from Arsh Nagar, one from Bus Stand and one from Chaura.

4.3.1 Validation Parameters

Three key parameters were used to validate the collected real-time traffic data using GASAPSO. The first is **cycle time** (in seconds), which refers to the total duration of one complete signal cycle across all phases (green, yellow, and red) for all directions; an optimized cycle time directly reduces overall intersection delays and can be validated by comparing GASAPSO outputs against real-world observations or the Webster Delay Function. The second parameter is **effective green light time** (in seconds), defined as the duration during which vehicles are permitted to pass through the intersection under a green signal; optimizing this parameter reduces congestion and waiting times, and its improvement can be quantified by comparing GASAPSO-derived timings against conventional fixed-time plans. The third is **average traffic delay** (in seconds per vehicle), which measures the mean time vehicles are held up by signals or congestion at an intersection; this is estimated using the Webster Delay Formula, and the reduction in delay before and after applying GASAPSO serves as the primary indicator of the algorithm's optimization effectiveness.

4.3.2 Validation on Real Time Traffic Data

Peak Hour Volume refers to the highest volume of traffic observed on a road or intersection during a specific peak hour period, usually during the morning or evening rush hours. This metric is essential for

infrastructure planning and traffic management, as it represents the highest demand on the road network during its peak periods. Peak hour volumes are typically used to assess the capacity of roads and intersections and to design traffic control measures.

The Table 5 in this section discusses the real time traffic count used to validate our proposed GASAPSO algorithm.

Table 5. Real time traffic data (Patiala).

Signal Phase	Peak Hour Volume	Demand Volume	Cycle Time (Seconds)	Green Light Time (Seconds)
Rajpura	1702	1619	45	35
Bus Stand	1911	1666	43	34
Arsh Nagar	780	715	23	15
Chaura	603	600	31	24

Demand Volume, on the other hand, represents the volume of traffic that would like to use a particular road or intersection. It is a broader measure that can be averaged over a longer time period and takes into account factors like land use, population density, and travel patterns. Demand volume is used to estimate future traffic needs and to plan for changes in transportation infrastructure.

4.3.3 *Traffic Delay Minimization using GASAPSO*

The average traffic delay has been calculated for the traffic coming from all the four directions in minutes using the Webster Delay Function and after that our proposed algorithm was applied on the same traffic data to minimize the traffic delay. It was successfully reduced on applying GASAPSO on the collected traffic data. The results are illustrated in Figure 6 and Table 6.

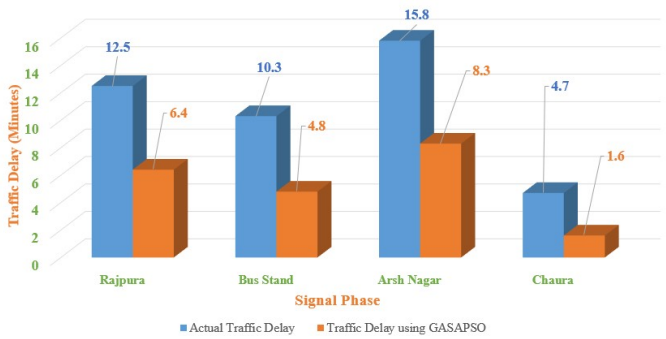


Figure 6. Traffic delay minimization using GASAPSO.

The results in the Table 6 show that the GASAPSO algorithm effectively reduces traffic delays across all signal phases. The optimization not only improves the efficiency of green light times but also significantly cuts down on average traffic delays. The results demonstrate that the GASAPSO algorithm substantially improves traffic signal efficiency across different phases. By optimizing green light times, GASAPSO reduces average traffic delays significantly: 6.1 minutes at Rajpura, 5.5 minutes at Bus Stand, 7.5 minutes at Arsh Nagar, and 3.1 minutes at Chaura.

These reductions highlight the algorithm's effectiveness in enhancing traffic flow and reducing congestion, even with varying green light durations. The consistent decrease in delays across all signal phases underscores GASAPSO's capability to adapt signal timings to better manage traffic, leading to notable improvements in overall traffic efficiency. Figure 6 above shows the minimized traffic delay for the traffic collected for the new bus stand area, Patiala. The results show that the traffic delay is minimized to approximately 47–66% by applying GASAPSO on real time traffic data which proves that our proposed algorithm GASAPSO is valid and applicable on real time scenarios.

4.3.4 *Traffic Delay Minimization Percentage*

Figure 7 shows the percentage of traffic delay minimization after applying GASAPSO on actual collected real time traffic data. The graph describes that for the traffic towards area of Bus stand, Patiala, average traffic delay reduced by 53.30 percent which is a huge reduction by applying GASAPSO on the collected traffic data. Also, for other traffic cases, average traffic delay significantly reduced.

4.3.5 *Comparison of Proposed Algorithm with State-of-the-Art Techniques*

In this section, our proposed hybrid algorithms have been compared against the existing hybrid algorithms proposed by various researchers to minimize traffic delay at traffic lights and intersections, as summarized in Table 7. The table shows that our proposed algorithm performed better than the existing algorithms as the maximum delay shown by our proposed algorithm is 186.5 seconds, less than the delay value calculated by the existing algorithms.

The proposed hybrid algorithm, GASAPSO, designed to minimize traffic delays at junctions, is compared to the state-of-the-art algorithms in the same field to see its effectiveness. A thorough comparison of essential parameters, including robustness against changing conditions, scalability, adaptability to traffic patterns, computational efficiency, and delay reduction percentages, is provided in the table. The table compares the proposed algorithm to other currently

Table 6. GASAPSO validation results.

Signal Phase	Rajpura	Bus Stand	Arsh Nagar	Chaura
Optimized Green Light Time	40 seconds	38 seconds	18 seconds	26 seconds
Traffic Delay without GASAPSO (Minutes)	12.5	10.3	15.8	4.7
Traffic Delay calculated using GASAPSO (Minutes)	6.4	4.8	8.3	1.6

Table 7. Comparison of the proposed algorithm with state-of-the-art techniques.

Year	Author (s)	Algorithm Used	Parameter	Maximum Delay (Seconds)
2024	Proposed Hybrid algorithm	GASAPSO (Genetic, Simulated Annealing and Particle Swarm Optimization)	Traffic Delay	186.5 seconds
2024	Akopov and Beklaryan [43]	BORCGA and BOPSO (Biobjective Real-Coded Genetic Algorithm and Biobjective PSO)	Traffic Delay	Delay not calculated
2020	Jian [44]	Multi-objective PSO for traffic signal control at isolated intersections	Traffic Delay	Not directly reported
2021	Zhang et al. [25]	Modified Webster Function combined with an improved genetic algorithm to optimize traffic signal timings and minimize intersection delay	Traffic Delay	Delay reduced by ∼15.64%
2022	Zhang et al. [42]	HCNSGA –III (Hybrid Constraint Strategy and NSGA-III framework)	Traffic Delay	660 seconds
2020	Iskandarani [18]	GABT (Genetic algorithm and Balancing technique)	Traffic Delay	215 seconds
2021	Zhang and Su [28]	Linear and non-linear optimization models for traffic light control in heterogeneous traffic systems	Traffic Delay	200 seconds

used methods, such as machine learning-based and conventional methods. This allows for an analysis of each methodology's relative performance and application. This comparison can be used by various researchers involved in urban planning and traffic management to determine the most viable approaches to improving traffic flow and easing congestion at signalized crossings. We discovered significant areas for improvement and difficulties in the state-of-the-art methods by combining knowledge from these foundational publications. This opened the door to creating our groundbreaking proposed algorithm. This method is a paradigm change in traffic management, combining the best aspects of PSO, GA and SA to produce an innovative solution.

Figure 7 shows the performance comparison using delay calculated by the existing algorithms and our proposed algorithm. It is clear from the figure above that our proposed delay successfully minimized the traffic signal delay, and it is the minimum delay compared with the delay shown by the other algorithms.

4.4 Time and Space Complexity of GASAPSO

The GASAPSO algorithm, which combines Genetic Algorithm (GA), Simulated Annealing (SA), and Particle Swarm Optimization (PSO), has both time and space complexities driven by the population size N and the number of generations G.

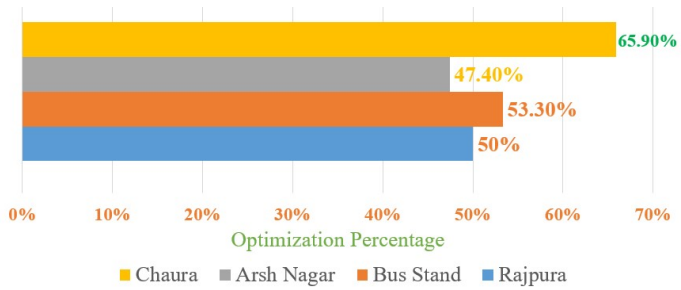


Figure 7. Performance Comparison of GASAPSO with existing algorithms.

4.4.1 *Time Complexity*

The time complexity of the algorithm is determined by the operations involved at each generation and over the total number of generations.

Initialization:

The initialization phase establishes the starting conditions for all three algorithmic components. A population of N individuals is first generated, with each individual representing a candidate traffic signal timing plan encoded as a set of green light durations for all directions; this population initialization step requires $O(N)$ time. Simultaneously, the PSO component initializes a velocity vector for each of the N particles, which governs the rate and direction of change in each particle's position during subsequent optimization iterations; this velocity initialization also contributes $O(N)$ to the overall setup cost.

Together, these two initialization steps ensure that both the evolutionary search space of the GA and the swarm-based exploration of PSO are properly configured before the main optimization loop begins, with a combined initialization complexity of $O(N)$.

Main Optimization Loop (for G generations):
During each generation of the main optimization loop, the algorithm performs a sequence of coordinated operations across all three components. First, fitness evaluation is applied to all N individuals, with each evaluation requiring $O(1)$ time, resulting in an overall cost of $O(N)$ per generation. The Genetic Algorithm phase then applies crossover to combine pairs of individuals and generate offspring, followed by mutation to introduce diversity into the population; both operations are applied across all N individuals and each contributes $O(N)$ to the per-generation cost. Concurrently, the PSO component updates the velocity of each particle in $O(1)$ time and subsequently updates its position, again yielding $O(N)$ for each operation across the full population of N particles. The Simulated Annealing component applies a local search adjustment to each individual, introducing a stochastic perturbation that helps the algorithm escape local optima, at a cost of $O(N)$ per generation. Finally, population replacement consolidates the updated solutions at the end of each generation, requiring $O(N)$ time. Since all per-generation operations scale linearly with population size N, the total time complexity per generation is $O(N)$, and the overall complexity across G generations is $O(G \times N)$.

Overall Time Complexity:
The total time complexity per generation is $O(N)$, and for G generations, the overall time complexity is:

$$O(G \times N) \tag{13}$$

where G is the number of generations, N is the population size.

4.4.2 Space Complexity

The space complexity of the GASAPSO algorithm is primarily driven by the storage requirements for the population, velocity vectors (for PSO), and potentially additional variables for fitness and positions.

The space complexity of GASAPSO is determined by three categories of stored data. The dominant contribution comes from population storage: the algorithm maintains N individuals in memory, each represented by d parameters (corresponding to the dimensionality of the signal timing problem), resulting in a storage cost of $O(N \times d)$. The PSO component introduces two additional storage requirements of the same order: a velocity vector of length d is maintained for each of the N particles to guide its search trajectory, and a position vector of equal length is stored to record each particle's current location in the solution space, each contributing $O(N \times d)$. By contrast, auxiliary variables including fitness values, the global best solution, and the temperature parameter used by the Simulated Annealing component occupy only $O(1)$ or $O(d)$ space, which is negligible relative to the population-level storage. Consequently, the overall space complexity of GASAPSO is $O(N \times d)$, reflecting a linear scaling with both population size and problem dimensionality.

Overall Space Complexity:
The total space complexity is:

$$O(N \times d) \tag{14}$$

where N is the population size, d is the number of parameters (dimensions) for each individual.

These complexities reflect a linear scaling of the algorithm, which is computationally efficient given the combination of GA, PSO, and SA techniques.

4.4.3 Complexity Analysis

The complexity analysis reveals three important properties of GASAPSO. First, the algorithm achieves a favorable balance between time and space complexity, as both grow linearly with population size N and problem dimensionality d, ensuring that the algorithm remains scalable and computationally tractable across a wide range of problem sizes. Second, this linear scaling makes GASAPSO well-suited for real-world traffic signal optimization, where large-scale intersection networks and multi-dimensional signal timing variables are common; the algorithm can handle moderate-to-large datasets without incurring prohibitive computational or memory costs. Third, despite integrating three distinct optimization techniques—GA, PSO, and SA—GASAPSO maintains low overall computational cost, as the per-generation operations of all three components each scale as $O(N)$, avoiding the multiplicative overhead that might otherwise arise from combining multiple metaheuristics. Taken together, these properties confirm that GASAPSO is both a practically efficient and theoretically sound

algorithm for complex, real-time traffic management applications.

4.5 Computational Resource Consumption of GASAPSO

The GASAPSO algorithm is generally efficient in both CPU and memory consumption for moderate problem sizes. However, for large-scale problems with high dimensionality and a large population, the resource consumption can become significant. Careful tuning of parameters like population size, number of generations, and dimensionality is crucial to ensure that the algorithm remains computationally feasible on the available hardware.

4.5.1 Considerations for Resource Optimization:

Several practical measures can be adopted to optimize the computational resource consumption of GASAPSO. Selecting an appropriate population size N is fundamental, as an overly large population increases both time and memory costs without proportional gains in solution quality, while too small a population risks premature convergence; a carefully tuned N therefore strikes the optimal balance between exploration breadth and resource efficiency. Similarly, limiting the number of generations G based on convergence criteria—such as terminating when the improvement in fitness falls below a predefined threshold—prevents unnecessary iterations and reduces total CPU time without sacrificing solution quality. Where hardware permits, parallelization offers a further reduction in wall-clock time: since fitness evaluations and the per-individual operations of GA, PSO, and SA are largely independent across the population, they can be distributed across multiple processor cores or threads, achieving near-linear speedup with respect to the number of available cores. Finally, the use of efficient data structures for storing population individuals, velocity vectors, and fitness values—such as contiguous arrays rather than linked structures—minimizes memory overhead and improves cache locality, which is particularly beneficial when N and d are large.

5 Conclusion and Future Scope

The GASAPSO algorithm has been proposed in this research work for traffic signal delay minimization to manage traffic on highly congested roads for multiple intersections and multiple traffic lights. These contributions generally advance the understanding and use of optimization approaches for reducing traffic delays, which may improve traffic management tactics in real-time traffic scenarios. GASAPSO contributes by eliminating delays and increasing the overall efficiency of urban transportation networks by optimizing traffic signal timings, reducing congestion, and improving overall traffic flow through the synergistic combination of genetic algorithms, simulated annealing, and particle swarm optimization. The suggested approach can be helpful for urban traffic control to decrease traffic jams on roads and reduce waiting times at traffic lights across multiple junctions. The delay has been minimized to a large extent with the proposed hybrid algorithm, but further improvements are required to optimize the other traffic signal parameters. In the future, this research work can be used to implement adaptive traffic signals to automatically adjust their timings based on actual traffic and weather conditions.

Furthermore, in larger-scale applications, the scalability of GASAPSO may provide difficulties. When dealing with complex problem areas that require a high level of investigation, the algorithm's performance may suffer. Furthermore, the technique may not be suitable for problems with rigorous deterministic requirements due to its dependence on heuristics and probabilistic mechanisms, especially during the SA phase.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Wen, W. (2008). A dynamic and automatic traffic light control expert system for solving the road congestion problem. *Expert systems with Applications*, *34*(4), 2370-2381. [CrossRef]

[2] Turky, A. M., Ahmad, M. S., Yusoff, M. Z. M., & Hammad, B. T. (2009, July). Using genetic algorithm

for traffic light control system with a pedestrian crossing. In *International Conference on Rough Sets and Knowledge Technology* (pp. 512-519). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[3] Slowik, A., & Kwasnicka, H. (2020). Evolutionary algorithms and their applications to engineering problems. *Neural Computing and Applications, 32*(16), 12363-12379. [CrossRef]

[4] Qadri, S. S. S. M., Gökçe, M. A., & Öner, E. (2020). State-of-art review of traffic signal control methods: challenges and opportunities. *European transport research review, 12*(1), 55. [CrossRef]

[5] Sofronova, E. (2021). Evolutionary computations for traffic signals optimization. *Procedia Computer Science, 186*, 802-811. [CrossRef]

[6] Wang, H., Hu, P., & Wang, H. (2021). A genetic timing scheduling model for urban traffic signal control. *Information Sciences, 576*, 475–483. [CrossRef]

[7] Shaikh, P. W., El-Abd, M., Khanafer, M., & Gao, K. (2020). A review on swarm intelligence and evolutionary algorithms for solving the traffic signal control problem. *IEEE Transactions on Intelligent Transportation Systems, 23*(1), 48–63. [CrossRef]

[8] Liu, W. L., Zhong, J., Liang, P., Guo, J., Zhao, H., & Zhang, J. (2024). Towards explainable traffic signal control for urban networks through genetic programming. *Swarm and Evolutionary Computation, 88*, 101588. [CrossRef]

[9] Dezani, H., Marranghello, N., & Damiani, F. (2014). Genetic algorithm-based traffic lights timing optimization and routes definition using petri net model of urban traffic flow. *IFAC Proceedings Volumes, 47*(3), 11326–11331. [CrossRef]

[10] Wijaya, I. G. P. S., Uchimura, K., & Koutaki, G. (2015). Traffic light signal parameters optimization using particle swarm optimization. In *2015 International Seminar on Intelligent Technology and Its Applications* (*ISITIA*) (pp. 11–16). IEEE. [CrossRef]

[11] Zhao, Y., Fang, W., & Qing, C. (2016). A PSO based signal timing optimization approach of phase combination. In *2016 International Conference on Civil, Transportation and Environment* (pp. 314–319). Atlantis Press. [CrossRef]

[12] Verma, A., Nagaraja, G., Anusha, C. S., & Mayakuntla, S. K. (2018). Traffic signal timing optimization for heterogeneous traffic conditions using modified Webster's delay model. *Transportation in Developing Economies, 4*(2), 13. [CrossRef]

[13] Zhao, H., Han, G., & Niu, X. (2020). The signal control optimization of road intersections with slow traffic based on improved PSO. *Mobile Networks and Applications, 25*(2), 623-631. [CrossRef]

[14] Cruz-Piris, L., Lopez-Carmona, M. A., & Marsa-Maestre, I. (2019). Automated optimization of intersections using a genetic algorithm. *IEEE Access, 7*, 15452-15468. [CrossRef]

[15] Ferrer, J., Lopez-Ibanez, M., & Alba, E. (2019). Reliable simulation-optimization of traffic lights in a real-world city. *Applied Soft Computing, 78*, 697-711. [CrossRef]

[16] Chen, Y., Qiao, F., Guo, L., & Liu, T. (2020, August). GAPSO-based traffic signal control in isolated intersection with multiple objectives. In *International Symposium on Simulation and Process Modelling* (pp. 263-275). Singapore: Springer Singapore. [CrossRef]

[17] Liang, X., Du, X., Wang, G., & Han, Z. (2019). A deep reinforcement learning network for traffic light cycle control. *IEEE Transactions on Vehicular Technology, 68*(2), 1243–1253. [CrossRef]

[18] Iskandarani, M. Z. (2020). Optimizing genetic algorithm performance for effective traffic lights control using balancing technique (GABT). *International Journal of Advanced Computer Science and Applications, 11*(3). [CrossRef]

[19] Bhattacharyya, R., Gupta, S., Marisha, Kumar, A., Mishra, D., & Gupta, M. (2026). A systematic review of evolutionary and swarm intelligence approaches for traffic signal control optimization. *Artificial Intelligence Review*. [CrossRef]

[20] Celtek, S. A., Durdu, A., & Ali, M. E. M. (2020). Real-time traffic signal control with swarm optimization methods. *Measurement, 166*, p.108206. [CrossRef]

[21] Liu, D., Yu, W., Baldi, S., Cao, J., & Huang, W. (2019). A switching-based adaptive dynamic programming method to optimal traffic signaling. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 50*(11), 4160–4170. [CrossRef]

[22] Dimitrov, S. (2020). Optimal control of traffic lights in urban area. In *2020 International Conference Automatics and Informatics* (*ICAI*) (pp. 1–6). IEEE. [CrossRef]

[23] Tsai, C. W., Teng, T. C., Liao, J. T., & Chiang, M. C. (2021). An effective hybrid-heuristic algorithm for urban traffic light scheduling. *Neural Computing and Applications, 33*(24), 17535-17549. [CrossRef]

[24] Dell'Orco, M., Başkan, Ö., & Marinelli, M. (2013, November). Artificial bee colony-based algorithm for optimising traffic signal timings. In *Soft Computing in Industrial Applications: Proceedings of the 17th Online World Conference on Soft Computing in Industrial Applications* (pp. 327-337). Cham: Springer International Publishing. [CrossRef]

[25] Zhang, H., Yuan, H., Chen, Y., Yu, W., Wang, C., Wang, J., & Gao, Y. (2021). Traffic light optimization based on modified webster function. *Journal of Advanced Transportation, 2021*(1), 3328202. [CrossRef]

[26] Lin, H., Han, Y., Cai, W., & Jin, B. (2022). Traffic signal optimization based on fuzzy control and differential evolution algorithm. *IEEE Transactions on Intelligent Transportation Systems, 24*(8), 8555-8566. [CrossRef]

[27] Ranpura, P., Shukla, V., & Gujar, R. (2024). Estimation of vehicle control delay using artificial intelligence techniques for heterogeneous traffic conditions. *Expert*

Systems with Applications, 246, 123206. [CrossRef]

[28] Zhang, Y., & Su, R. (2021). An optimization model and traffic light control scheme for heterogeneous traffic systems. *Transportation research part C: emerging technologies, 124*, 102911. [CrossRef]

[29] Dell'Orco, M., Baskan, O., & Marinelli, M. (2013). A Harmony Search Algorithm approach for optimizing traffic signal timings. *PROMET-Traffic&Transportation, 25*(4), 349-358. [CrossRef]

[30] Feng, B., Zheng, M., & Liu, Y. (2023). Optimization of signal timing for the contraflow left-turn lane at signalized intersections based on delay analysis. *Sustainability, 15*(8), 6477. [CrossRef]

[31] Liang, X., Guler, S. I., & Gayah, V. V. (2023). Decentralized arterial traffic signal optimization with connected vehicle information. *Journal of Intelligent Transportation Systems, 27*(2), 145-160. [CrossRef]

[32] Lee, J., Kim, J., & Park, B. (2011). A genetic algorithm-based procedure for determining optimal time-of-day break points for coordinated actuated traffic signal systems. *KSCE Journal of Civil Engineering, 15*(1), 197-203. [CrossRef]

[33] Jahangiri, A., Afandizadeh, S., & Kalantari, N. (2011). The optimization of traffic signal timing for emergency evacuation using the simulated annealing algorithm. *Transport, 26*(2), 133–140. [CrossRef]

[34] Hu, T. Y., & Chen, L. W. (2012). Traffic signal optimization with greedy randomized tabu search algorithm. *Journal of transportation engineering, 138*(8), 1040-1050. [CrossRef]

[35] Zhang, Z., Zhu, H., Zhang, W., Cai, Z., Zhu, L., & Li, Z. (2023). Multi-Objective Optimization of Traffic Signal Timing at Typical Junctions Based on Genetic Algorithms. *Computer Systems Science & Engineering, 47*(2). [CrossRef]

[36] Lee, J., Abdulhai, B., Shalaby, A., & Chung, E. H. (2005). Real-time optimization for adaptive traffic signal control using genetic algorithms. *Journal of Intelligent Transportation Systems, 9*(3), 111-122. [CrossRef]

[37] Mao, T., Mihăită, A. S., Chen, F., & Vu, H. L. (2021). Boosted genetic algorithm using machine learning for traffic control optimization. *IEEE transactions on intelligent transportation systems, 23*(7), 7112-7141. [CrossRef]

[38] Desmira, D., Abi Hamid, M., Bakar, N. A., Nurtanto, M., & Sunardi, S. (2022). A smart traffic light using a microcontroller based on the fuzzy logic. *IAES International Journal of Artificial Intelligence, 11*(3), p.809. [CrossRef]

[39] Liu, B., & Ding, Z. (2022). A distributed deep reinforcement learning method for traffic light control. *Neurocomputing, 490*, 390–399. [CrossRef]

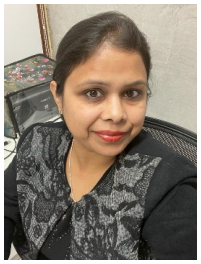
[40] Li, Z., & Schonfeld, P. (2015). Hybrid simulated annealing and genetic algorithm for optimizing arterial signal timings under oversaturated traffic conditions. *Journal of advanced transportation, 49*(1), 153-170. [CrossRef]

[41] Tran, Q. H., Do, V. M., & Dinh, T. H. (2022). Traffic signal timing optimization for isolated urban intersections considering environmental problems and non-motorized vehicles by using constrained optimization solutions. *Innovative Infrastructure Solutions, 7*(5), 299. [CrossRef]

[42] Zhang, X., Fan, X., Yu, S., Shan, A., Fan, S., Xiao, Y., & Dang, F. (2022). Intersection signal timing optimization: A multi-objective evolutionary algorithm. *Sustainability, 14*(3), 1506. [CrossRef]

[43] Akopov, A. S., & Beklaryan, L. A. (2024). Traffic improvement in Manhattan road networks with the use of parallel hybrid biobjective genetic algorithm. *IEEE Access, 12*, 19532-19552. [CrossRef]

[44] Jian, L. (2020). Multi-objective optimisation of traffic signal control based on particle swarm optimisation. *International Journal of Grid and Utility Computing, 11*(4), 547-553. [CrossRef]



Tanya Garg is currently working as Assistant Professor at Thapar Institute of Engineering & Technology. She obtained her Ph.D. from SLIET Longowal in 2024. She obtained her Masters in Computer Science & Technology from Central University of Punjab, Bathinda in year 2014 and B.E in CSE from S.L.I.E.T, Longowal in year 2012. She is a member of various professional bodies. Her current area of research and interest includes Machine Learning, Big Data & Data Mining. (Email: tanya.garg@thapar.edu)

Gurjinder Kaur is an accomplished academician and researcher currently serving as an Associate Professor in the Department of Computer Science and Engineering at Sant Longowal Institute of Engineering and Technology (SLIET), Longowal. She holds a Bachelor of Engineering degree from Kyrgyz Technical University, Bishkek, an M.S. from Birla Institute of Technology and Science (BITS), Pilani, and a Ph.D. from SLIET, Longowal. With a strong academic background and extensive teaching experience, Dr. Kaur has been actively involved in guiding students and contributing to research in emerging areas of computer science. Her academic pursuits are complemented by her commitment to fostering innovation, analytical thinking, and interdisciplinary learning among students. She continues to engage in research collaborations and academic initiatives aimed at advancing knowledge and technological development in the field of computer science and engineering. (Email: gurjinder13@yahoo.com)