Applied Soft Computing Journal 184 (2025) 113782

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Grammatical Evolution for automatic design of actuated traffic signal control plans

Mahmud Keblawi [a,b,*], Tomer Toledo [a]

[a] Transportation Research Institute, Technion—Israel Institute of Technology, Haifa, 32000, Israel
[b] School of Civil Engineering, The University of Queensland, St Lucia, Brisbane, 32000, Queensland, Australia

ARTICLE INFO

Keywords:
Evolutionary algorithms
Optimization
Grammatical evolution
Automatic design
Traffic signal control
Traffic simulation

ABSTRACT

In traffic networks, proper signal control design is essential to ensure a reasonable level of service. Signal control designs are becoming increasingly complex, with numerous settings that must be calibrated and set. This paper introduces a novel approach for handling this complexity by automatically generating optimal actuated signal control plans using Grammatical Evolution (GE). GE has proven its effectiveness in automating the design of different complex systems, such as neural networks and analog electronic circuits. GE's distinctive mapping and representation capabilities make it a powerful candidate for optimizing various systems.

In contrast to traditional optimization methods for actuated signal plans, which focus on specific parameters, such as green times and cycle length, the GE-based approach evolves complete plans, including phases, detector placements, and transit priority strategies. As a result, it eliminates the need for human intervention in the design process, making it more efficient and less time-consuming.

The proposed approach was tested with an application to an isolated intersection in Haifa, Israel. The results showed that the automatically generated signal plan outperformed the existing plan by reducing delay times and queue lengths. Moreover, this method demonstrated its efficiency in generating reliable traffic signal plans under challenging traffic conditions.

1. Introduction

Traffic congestion is a major problem in urban areas. Its direct cost was estimated at around 88 billion dollars in the United States in 2019 [1]. Congestion also adversely affects the environment, life quality, economic activity, and safety. Therefore, congestion mitigation is an important goal for the managers of transportation systems. One way to achieve this is to invest in new or improve existing transportation infrastructure. However, this approach cannot alone cope with the increase in traffic demands due to funding restrictions and insufficient space for additional infrastructure [2]. As a result, more efficient operations of the existing infrastructure are necessary [3].

Traffic signal control is one of the most cost-effective tools to improve the performance of traffic systems [4]. As a result, a lot of effort has been devoted ever since Webster [5] developed the basic principles and theory of traffic signal optimization with the aim to improve existing methods and introduce new ones. Traffic signal control plans have become more complex due to the inclusion of advanced features, such as pedestrian actuation and Transit Signal Priority (TSP), to increase the overall safety and mobility of the system. Consequently, designing and optimizing signal control plans has become an increasingly challenging task [2].

Studies on the optimization of actuated signal control plans have primarily focused on their design parameters. Commercial optimization programs, such as HCM [6], SYNCHRO [7], TRANSYT-7F [8], PASSER II [9], and PASSER V [10], as well as studies [11,12], optimize the basic signal plan parameters (i.e., cycle length, green splits, phase sequence, and offsets between adjacent intersections). Several studies [13–16] have also incorporated additional parameters into the optimization, such as maximum allowable vehicle waiting times and minimum and maximum green times for actuated phases. Stevanovic et al. [17], Stevanovic et al. [18], Balasha and Toledo [2], and Toledo et al. [19] optimized TSP parameters along with the basic design parameters of actuated signal plans that also implement TSP functions. In all the studies mentioned above, incorporating additional signal plan parameters into the optimization significantly improved system performance. Optimization was based on gradient-based, genetic, and other heuristic search algorithms. Traffic simulation models were used to evaluate the performance of candidate solutions. These studies optimized the parameters of plans with predefined structures. However, they did

* Corresponding author at: School of Civil Engineering, The University of Queensland, St Lucia, Brisbane, 32000, Queensland, Australia.
E-mail address: mahmudkeblawi@gmail.com (M. Keblawi).

https://doi.org/10.1016/j.asoc.2025.113782
Received 14 July 2024; Received in revised form 5 August 2025; Accepted 8 August 2025
Available online 26 August 2025

not consider other design components, such as the composition of phases, selection of priority strategies to be implemented, and placement of detectors. They were instead set manually by professional traffic engineers following general engineering guidelines. To the best of our knowledge, methods to fully design actuated signal control plans automatically from scratch have not been proposed in the literature.

This study aims to fill this gap by leveraging Grammatical Evolution (GE) to automatically generate complete actuated signal control plans. While various algorithms such as Genetic Algorithms (GA) [20–23], Differential Evolution (DE) [24–26], and Swarm Intelligence methods like Particle Swarm Optimization (PSO) [27–29] have been applied to traffic signal optimization, as reviewed in [30], these approaches primarily optimize numerical parameters of the signal plan (e.g., cycle length, green splits, red times) but lack the capability to generate complete control plans. In contrast, GE overcomes this limitation by employing a grammar-based approach, enabling the automatic generation of complex control plans suitable for different intersection layouts. The use of grammar provides a simple mechanism for representing complex systems and allows for incorporating problem-specific information while limiting the design space [31]. This focused representation avoids broad, unconstrained search spaces that often reduce the efficiency of other optimization algorithms. Consequently, it becomes more effective and can lead to more creative solutions [32,33].

In recent years, GE [34] has been applied in several domains to design complex systems automatically due to its unique representation and mapping process [35]. For example, Tsoulos et al. [36] used GE to evolve the weights and topology of Artificial Neural Networks (ANNs) with one hidden layer. Ahmadizar et al. [37] applied GE to generate the ANN topology, while weights were optimized using GA [38]. Assunção et al. [39] employed GE to simultaneously evolve topology and weights for ANNs with more than one hidden layer. In addition, this approach was applied to deep learning models. Baldominos et al. [40], Baldominos et al. [31], and Assunção et al. [41] proposed systems for automatically designing convolutional neural networks for different applications (e.g., activity and handwriting recognition). These systems simultaneously generate the convolutional and fully connected layers, activation functions, and other learning parameters. In three studies by Castejón and Carmona [42–44], GE has been successfully used in the design of analog electronic circuits. In these studies, the position and connections of the electronic components (such as resistors, transistors, and capacitors) are determined to meet specific circuit design requirements. In addition, Zhao et al. [33] proposed a fully automated Grammar-based approach for generating optimized robot structures (i.e., physical robot components including wheels, joints, and limbs) and their corresponding controllers to traverse given terrains. These studies demonstrate that GE can be employed successfully to design complex systems automatically. To the best of our knowledge, GE has never been used in the field of traffic signal control.

This paper introduces a fully automatic design system for signal control plans, which uses grammatical evolution to represent and optimize candidate plans. It goes beyond current signal plan optimization methods in that it generates the full control plan, including the incorporation of advanced features, rather than only optimizing its parameters such as green times and cycle length. Thus, it can help reduce the time, cost, and effort typically associated with the design process. The proposed system provides a generic approach suitable for various real-world intersection configurations, allowing effective traffic control based on the intersection's layout and traffic flow patterns.

The previous publication [45] laid the groundwork by presenting the overall structure of the automatic signal control design system, with an emphasis on transportation aspects (e.g., simulation model calibration, characteristics of the resulting signal control plans) and its application in real-world scenarios, including sensitivity analysis tests. Extending from these foundational insights, the present paper takes a deeper dive into GE's application for traffic signal control design. It focuses on the specifics and effectiveness of GE within the automated process of designing optimal traffic signal plans, paying particular attention to the decoding and evolution of candidate traffic signal plans.

The remainder of this paper is organized as follows: The next section briefly presents the basic concepts, the structure, and functionalities of signal control plans. Sections 3 presents the structure of the automatic design system, focusing on the grammatical evolution approach and its application in designing traffic signal plans. A case study and analysis of its results are presented in Sections 4 and 5, respectively. Finally, a conclusion is provided in Section 6.

2. Signal control plans

Traffic signal plans provide repeated sequences of signal indications to road users. Fig. 1 illustrates the basic elements of signal control plans: Phases, green times, inter-green times and cycles. Phases (also referred to as stages) consist of compatible vehicle and pedestrian movements. These phases are organized to operate in a specific sequence. Green times are the time periods allocated to each phase during which the movements within that phase are allowed to cross the intersection. Inter-green periods are the intervals between the end of a phase and the start of the next one, which is used to avoid interference between conflicting movements at the intersection. Finally, the cycle length is the time it takes to complete a full phase sequence [46,47]. It is the sum of the green times given to each phase and the inter-green between phases [48]. Each movement through the intersection must be allocated to at least one phase within a cycle, so that it can be served.

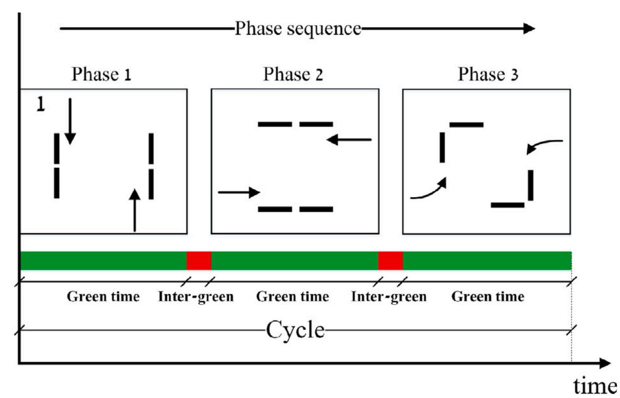


Fig. 1. The basic components of traffic signal control plans: phases, their sequence, timing and inter-green transitions.

The design of signal control plans includes determining the composition of phases, their sequence, green times and cycle lengths that would provide the best performance for the intersection [46]. Signalized intersections are usually operated by one of three traffic control strategies: fixed-time, actuated, or adaptive [49]:

- **Fixed-time or pre-timed control strategies** assume constant traffic demand at all times. Historical traffic demand information is used to determine optimal signal control design based on a fixed signal phase sequence with a fixed-time duration for each phase, regardless of changes in traffic conditions. Therefore, it is a traffic unresponsive signal control scheme. Notably, the vast majority of intersections worldwide operate under this approach due to its relative simplicity and low costs.
- **Actuated control strategies** extend fixed-time control by allowing the modification of green time allocation in real-time by applying simple logic criteria, such as phase skipping or extending the green time for a specific phase up to a predefined maximum time. These modifications occur in response to traffic state information, such as the presence of vehicles at the stop line in a lane, which is commonly obtained from electromagnetic loop detectors or video cameras.

- **Adaptive control strategies** adjust signal settings in real-time by solving an optimization problem utilizing dynamic traffic models to predict the near future traffic conditions, such as vehicle platoons and their movements in the network.

With the rise of machine learning techniques, numerous learning-based approaches have emerged in traffic signal operations, such as reinforcement learning and deep reinforcement learning. These approaches fall under adaptive strategies, as they enable systems to dynamically adjust to real-time traffic conditions [50]. Despite their flexibility and potential performance benefits, the implementation, maintenance, and operational costs of adaptive systems often exceed the available resources of many transportation agencies [51]. Consequently, such methods remain less common in practice compared to actuated systems. Although adaptive methods fall outside the scope of this study, readers interested in further details are referred to review papers [46,50,52,53] which provide in-depth analyses of recent advancements in the field.

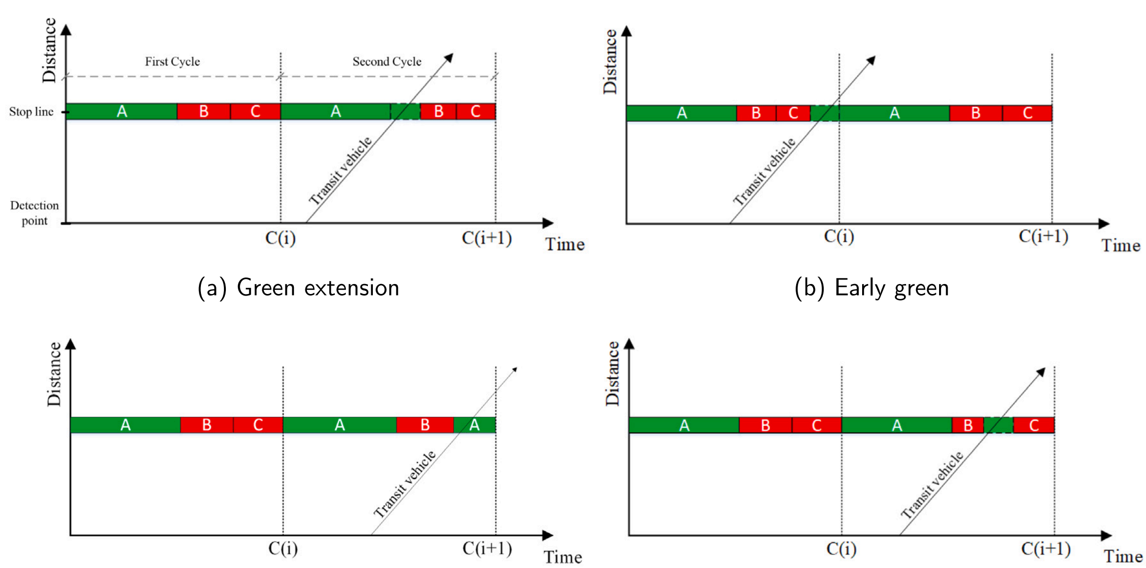
In contrast, actuated traffic signal control presents a practical middle ground between fixed-time and adaptive control strategies, balancing performance improvement with economic feasibility. Actuated control plans may also include functions for TSP, which aim to reduce public transportation travel times [54]. Several active TSP strategies may be used to adjust the control in response to priority requests from transit vehicles. The TSP strategy implemented in each case depends on the current active phase and its elapsed green time when the transit vehicle is expected to arrive at the stop line [55]. Common TSP functions are shown in Fig. 2, which presents a standard cycle followed by a TSP-adjusted one [56]:

1. Green extension is applied when a transit vehicle is expected to arrive at the intersection shortly after the end of the green time for the phase it is served by; in this example, phase A (Fig. 2(a)).
2. Early green is applied when a transit vehicle is expected to arrive at the intersection shortly before the phase that serves it is activated. The green times for one or more of the preceding phases are shortened to allow the TSP phase to start earlier (Fig. 2(b)).
3. Phase skipping involves omitting one or more phases to serve a priority request sooner (Fig. 2(c)).
4. Phase insertion involves adding a special phase into the normal sequence to serve the transit vehicles approaching the intersection. It may also require truncation of the green times for other phases (Fig. 2(d)).

3. Automatic signal control design system

Fig. 3 illustrates the overall structure of the automatic signal control design system. It consists of two main components: a traffic simulation model used to evaluate candidate control plans and an optimization algorithm, specifically Grammatical Evolution, utilized to generate new control plans based on the evaluation results. The inputs to the design system include data on traffic demand for all road users (i.e., pedestrians, passenger cars, and public transportation), the intersection's geometric layout, and a user-defined design constraint, such as the operation strategy (i.e., operating with a fixed or variable cycle length), maximum or minimum cycle length, and maximum allowed waiting times for specific movements. The system's output is the best-evolved control plan obtained by the Grammatical Evolution algorithm. The following points provide detailed information about the design process:

1. Initialization:
    - set the geometric intersection layout, the traffic demands and the design constraints.
    - Pre-process the input data and generate all feasible phases based on the geometric intersection layout. All feasible phases consist of all combinations of non-conflicting vehicle movements. Besides, within each phase, the crosswalks that do not conflict with its vehicle movement are included. Accordingly, pedestrians are allowed to cross the road (i.e., at signalized crosswalks) only when competing movements have red lights.
    - Automatically generate initial feasible signal control plans set.
2. Run the traffic simulation model with the current signal control plan.
3. Calculate performance measure from the simulation output.
4. Generate new solution sets (as numerical vectors) for the next iteration based on the performance of the solutions from the previous iteration, implementing the genetic operators: selection, crossover, and mutation.



(a) Green extension (b) Early green (c) Phase skipping (d) Phase insertion

Fig. 2. Common strategies for transit priority provision: (a) Green extension, (b) Early green, (c) Phase skipping, and (d) Phase insertion.

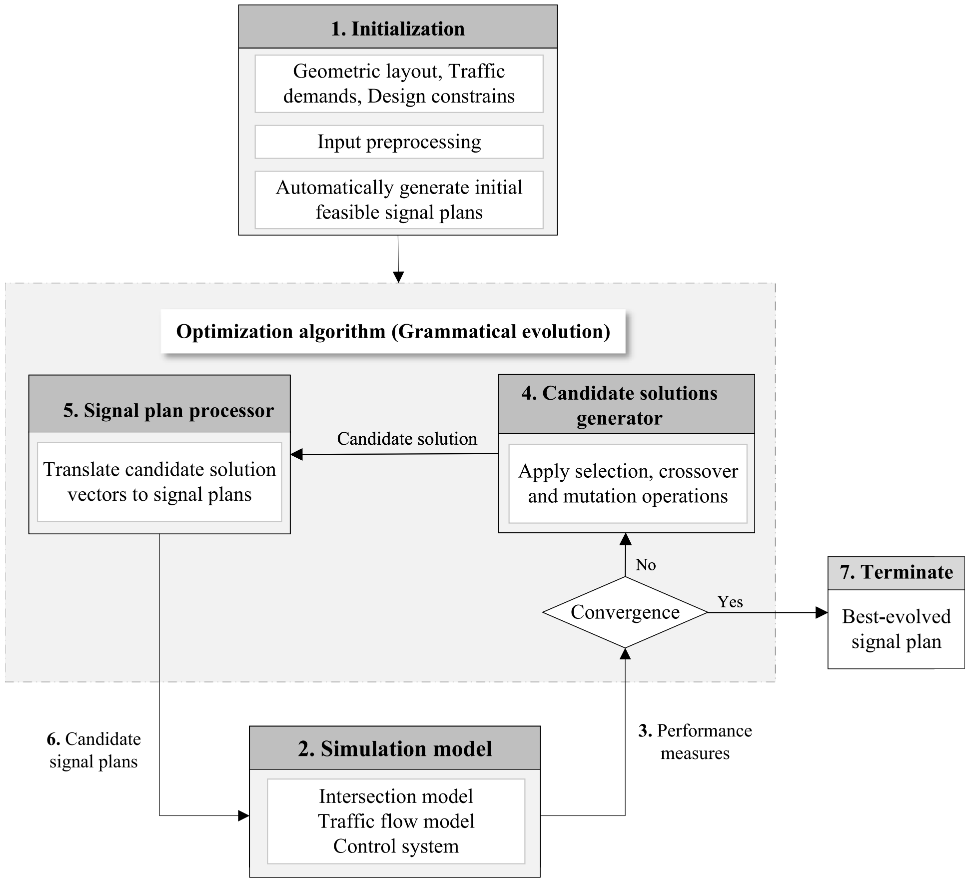


Fig. 3. The overall structure of the automatic signal control design system.

5. Translate the solution vectors into executable signal control plans for application in the simulation model.
6. Set the new plans in simulation model.
7. Repeat steps 2 through 6 until the termination criteria are satisfied, including either reaching the maximum number of generations or observing no improvement in fitness for a defined number of consecutive generations.

The components of the automatic design system are discussed in detail in the following subsections.

3.1. Grammatical evolution (GE)

GE is an evolutionary algorithm classified as a grammar-based form of genetic programming. GE is one of the most widely used grammar-based approaches [57]. It can automatically evolve optimal expressions and programs of arbitrary complexity for different problems [34]. In GE, individuals (candidate solutions) are represented as variable-length numeric arrays called chromosomes. Chromosomes consist of information units called codons, as shown in Fig. 4.

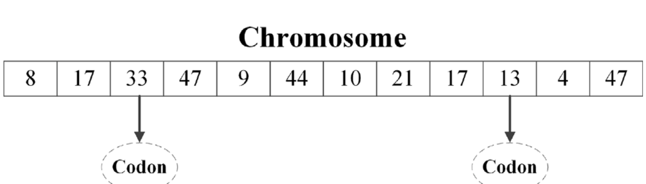


Fig. 4. An example of the structure of a chromosome and its corresponding codons.

The main steps of the GE search process for the optimal solution begin with the random generation of an initial population of feasible individuals. Next, each individual is decoded into the desired format (e.g., a computer program). The decoding of each individual is performed based on a formal grammar set. Typically, this grammar is expressed in BNF format (Backus–Naur Form) [34]. In the third step, the fitness of each individual is evaluated, i.e., the value of the objective function. Next, a new generation is produced by applying genetic operators: selection, crossover, and mutation. Ultimately, after the repeated application of the genetic operators and the fitness evaluations, the optimal solution will be reached when one of the termination criteria is met (e.g., convergence thresholds for the fitness or the maximum number of generations). Note that the GE algorithm is similar to the GA scheme. However, the main difference between GA and GE is the decoding phase, where in GE the decoding is done using a grammar set, while in GA it is done directly [32]. The use of grammars can provide a simple mechanism for representing complex systems [31]. In addition, it allows for the incorporation of problem-specific information while limiting the design space. Consequently, the search processes become more effective, leading to more creative solutions [32,33].

3.1.1. BNF grammar

BNF is a formal representation for encoding grammar as production rules [58]. Generally, BNF consists of four types of elements:

- T - Terminal symbols that appear in a valid language/program sentence, e.g., numbers (1, 2, 3), and operators (+, –, %).
- N - Non-terminal symbols that can be expanded to either terminal or non-terminal symbols.
- S - Start symbol from which the decoding process begins, and it belongs to N.

- R - Production rules for mapping elements from N to T. Each production rule may include a different number of rule choices.

3.1.2. Individual decoding process

The decoding process sequentially reads codons from left to right, using them to determine the value for the next symbol. Symbols are selected by:

$$S_i = (C_i)MOD(N_{ri}) \tag{1}$$

Where S_i is the selected symbol by the value C_i of codon i. MOD is the modulus function. Its result is an integer in the set $[0,1,\ldots,(N_{ri}-1)]$, which represents the index of the chosen value. N_{ri} is the number of choices for the relevant production rule.

The decoding process starts from the start symbol in the BNF grammar, which is a non-terminal symbol. After that, if the selected rule is a non-terminal symbol, another codon is read. This process continues until the final expression is reached, i.e., all non-terminal symbols are mapped. When all codons are read, but there are still non-terminal symbols that are unmapped, then the wrapping mechanism will be used [57], which means that the reading of the codons will start again from the beginning of the chromosome until all non-terminal symbols are mapped.

3.1.3. Developed grammar for traffic signal plan design

The grammar used to decode chromosomes into traffic signal plans is expressed in BNF format. It specified the design production rules, which map codons to components of the signal control plan, as shown in Fig. 5.

The production rule P represents the feasible compositions of basic phases. Basic phases constitute the normal phase sequence for each signal plan. In the proposed system, the compositions of these phases are automatically generated during the initialization step, using the intersection's geometric layout and specifically considering conflicts between competing movements to ensure that all compositions are legal and conflict-free.

However, the production rule T is a subset of P and represents the compositions of insert phases. Insert phases operate outside the normal phase sequence to support only transit priority actions, as depicted in Fig. 2(d). Thus, T includes all compositions in P containing at least one transit movement. This means that P covers all possible combinations of intersection movements, while T focuses on combinations that involve at least one transit movement.

The demand (C) and extension (J) detector symbols expand to determine whether detectors exist (D and E, respectively) or not (null) in the relevant movements. In addition, the extension detector (J) includes a symbol that determines its distance from the stop line (W). Demand detectors are always placed at the stop line to enable the control system to detect the presence of vehicles. If vehicles are detected before the stop line, the relevant phase is activated; if no vehicles are detected, the phase may be skipped. Extension detectors are positioned upstream of the intersection to monitor approaching vehicles, allowing the control system to decide whether to extend the active phase or not.

The parameters minimum duration (L), extension time (U), importance index (K), and extension detector distance (W) are numeric and can have values from 0 to their pre-defined upper bounds $u1$, $u2$, $u3$, and $u4$, respectively. The minimum duration is the least green time granted to the active phase, ensuring sufficient operation even without extension requests. The extension time represents the maximum green duration that can be added to the minimum duration in response to

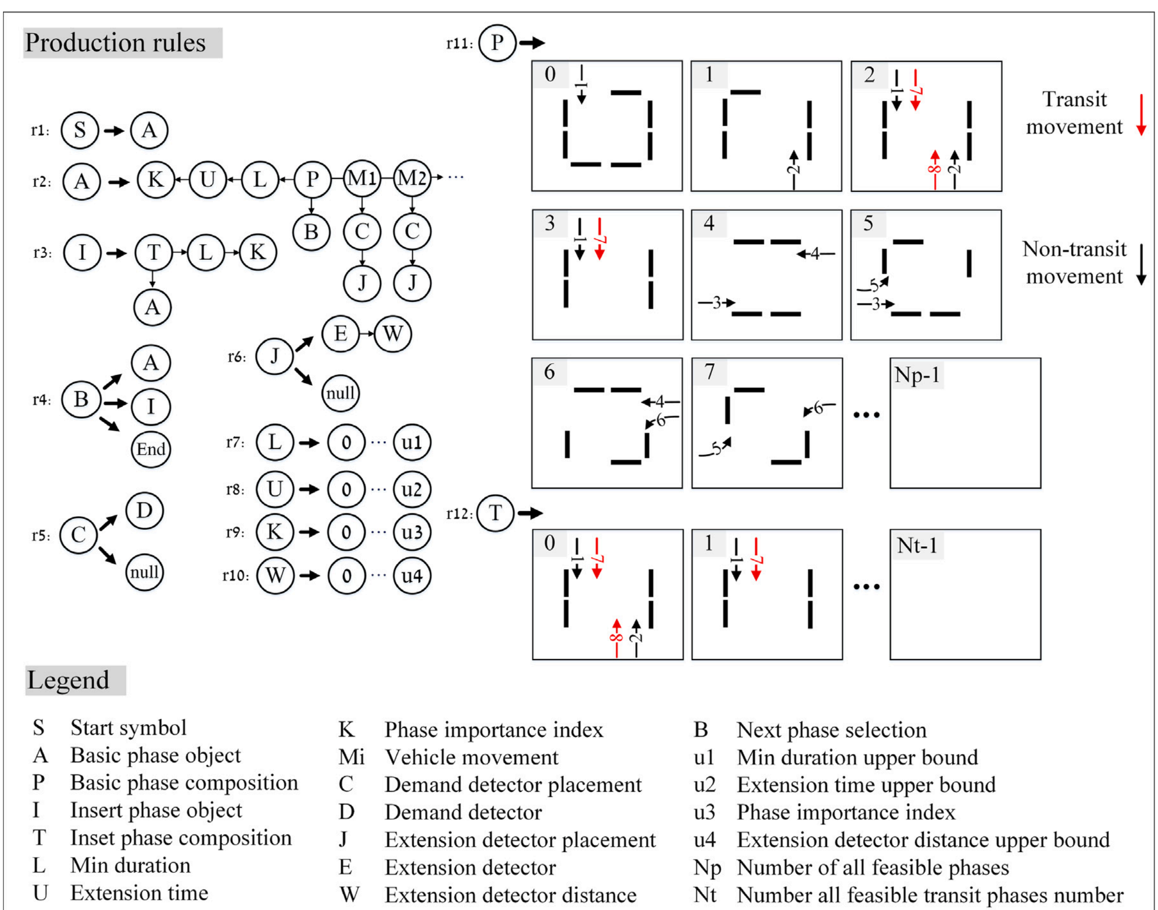


Fig. 5. Detailed representation of the grammar used for decoding traffic signal plans.

continuous requests for extensions. The importance index, a unique feature in the proposed system, assists in determining which phases need to be shortened more when multiple phases need to be shortened due to transit priority actions. This parameter allows the control system to make strategic decisions, avoiding disruption to the most crucial phases. Typically, such prioritization would be embedded within the traffic control logic designed manually by transportation engineers.

The decoding process starts from the start symbol (S), which leads to the first basic phase within the cycle (A). The phase expands to define the phase composition (P), phase minimum duration (L), extension time (U), importance index (K), the next phase (B) in the sequence, existence of demand detectors (C), and the existence and position of extension detectors (J) for the movements that are active in the specific phase ($M1$, $M2$, and so on). The next phase symbol (B) can be expanded to another basic phase (A), an insert phase (I), or to the END symbol, which means that this was the last phase in the cycle and terminates the decoding process. The insert phase component (I) expands to define its composition (T), minimum duration (L), importance index (K), and the next phase (A). Insert phases cannot be the last ones in the cycle and cannot be followed by another insert phase. This is a standard rule of thumb in transportation engineering to ensure smooth operation and avoid creating bottlenecks or safety hazards by having too many priority phases in succession.

This grammar is generic and can be applied to any intersection layout by adjusting the phase compositions. Further, they could be easily adapted for other design requirements, such as using other sensing technologies or defining pedestrian buttons. The decoding process is defined so that each phase (A or I) consumes a fixed-length block of 12 codons, as shown in Fig. 6, even if some of them will not be used for decoding. This would be useful in the chromosome evolution step.

3.1.4. Post-processing stage

After decoding, the generated control plans cannot be directly fed to the simulation model. These plans should undergo a post-processing step, which includes:

- Sum the minimum duration and extension time of each phase to determine its maximum duration.
- Calculate the cycle length. The cycle length can be calculated by the summation of all phases' maximum green times and intergreen times among them:

$$C = \sum_{i=1}^{n} G_i^{\max} + \sum_{j=1}^{n} PT_{j,j+1} \tag{2}$$

  Where C is the cycle length in seconds. n is the number of phases in the cycle. $G_i^{\max}$ is the maximum green time allocated to phase i. $PT_{j,j+1}$ are the safe transition time between phases j and j+1. For the last phase in the cycle ($j = n$), the transition is defined to the first phase in the next cycle ($j + 1 = 1$).
- Install pedestrian pushbuttons in the crosswalk that are served only in the phases that would not be activated unless the vehicles' requests are detected. The purpose of this is to avoid situations in which pedestrians are waiting to cross the road, however, the phase that serves their movement has been skipped because there are no vehicles detected during a certain period of time.
- Generate the signal control-flow diagram for execution in the simulation model. This diagram is generated based on the phase sequence, phase parameters, detectors, and TSP strategies. An illustration of this diagram can be found in the decoding example in Fig. 9.

Table 1
Overview of the relevant grammar parameters.

Parameter	Value
u1	19
u2	39
u3	9
u4	39
Np	30
Nt	12

3.1.5. Screening stage

After the post-processing step, processed control plans are examined for adherence to design constraints with the goal of maximizing optimization efficiency by excluding infeasible designs. In some cases, constraints can be satisfied by modifying control plans that violate them (e.g., maximum cycle length constraint). The design constraints are predetermined and provided to the system during the initialization process. Other constraints, such as pedestrian pushbutton integration or phase sequence logic, are already addressed during the post-processing stage or embedded in the grammar definition. The following are the primary constraints checked at this stage and how they are handled:

- All intersection movements and crosswalks are included in the candidate signal control plan. If this condition is not satisfied, the generated traffic signal plan will be strongly penalized to decrease its chances of being chosen during the parent selection.
- The cycle length (calculated) is less than or equal to the maximum cycle length defined by the user. If this constraint is not satisfied, phases' maximum durations must be decreased by multiplying them by a reduction factor. This factor equals the maximum cycle length divided by the calculated cycle length. In the event that the phase's new maximum duration is less than its minimum duration, it will be increased to be equal to the minimum duration. However, this can lead the cycle length to be greater than the maximum cycle length again, so a new adjusting iteration is required. In cases where the phases' minimum duration is too high and does not allow the cycle length to be less or equal to the maximum cycle length, the traffic signal plan will be strongly penalized.

3.1.6. Decoding example

A simple decoding example using the grammar described in Fig. 5 is presented below. It includes a detailed decoding of the first phase of the traffic control plan presented in this example. The parameter values within the grammar are presented in Table 1. Based on the values in this table, the minimum duration for each phase can range from 0 to 19 ($u1$), the extension time from 0 to 39 ($u2$), the importance index from 0 to 9 ($u3$), and the extension detector distance from 0 to 39 meters ($u4$). These values represent acceptable ranges rather than fixed numbers. While larger numbers within these parameters are possible, limiting them to these acceptable ranges increases the efficiency of the optimization process. Additionally, there are 30 available combinations for basic phases (Np) and 12 for insert phases (Nt), reflecting the number of feasible phase combinations generated based on the specific intersection layout during the initialization process.

Table 2 presents the number of choices for each production rule in the given grammar. The intersection layout in this example consists of

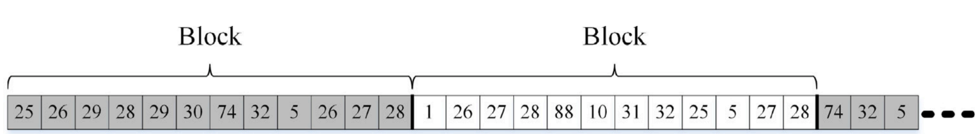


Fig. 6. An example of fixed-length codon blocks.

Table 2
Detailed enumeration of choices available for each non-terminal symbol in the grammar production rules.

Rule number	Non-terminal symbol	Number of rule choices
1	S	1
2	A	1
3	I	1
4	B	3
5	C	2
6	J	2
7	L	20
8	U	40
9	K	10
10	W	40
11	P	30
12	T	12

six non-transit vehicle movements (movements 1 to 6), two transit vehicle movements (movements 7 and 8) that pass through two dedicated lanes, and eight pedestrian crosswalks that cross these movements.

The chromosome used in this example is: **{242, 126, 215, 19, 83, 34, 63, 11, 22, 100, 99, 156, 84, 48, 253, 12, 221, 71, 68, 174, 145, 168, 44, 128, 105, 129, 37, 98, 186, 56, 16, 138, 17, 2}**. The decoding of this individual involves the following steps:

- Starting with the S symbol. This symbol has only one associated rule that is expanded directly to the basic phase component (A).
- The basic phase component also has only one associated rule. Therefore, it is expanded directly, and the result is the symbols that define the first phase: P, L, U, K, $M1$, $M2$, and B.
- P is a non-terminal symbol that has 30 rules ($Np = 30$). By using the first codon, phase 2 in Fig. 5 will be selected (242 MOD 30 = 2) as phase "A" in the plan (Fig. 7).
- Codons 126, 215, and 19 are used to define phase A minimum duration (126 MOD 20 = 6 s), extension time (215 MOD 40 = 15 s), and importance index (59 MOD 10 = 9).
- Phase A includes two non-transit vehicle movements. Therefore, the symbols $M1$ and $M2$ are expanded to define the demand and extension detectors that may exist in movements 1 and 2, respectively. Note that in this application, detectors installed in transit vehicles' approaches (dedicated lanes) were assumed to be fixed and are provided to the system as input. Therefore, they are not included in the grammar.
- The next codon, 83, is used with rule #5 to determine that a demand detector does not exist in movement 1 (D_1) (83 MOD 2 = 1 => Null).
- The codon 34 is applied with rule #6 to determine the existence of an extension detector upstream movement 1 (E_1) (34 MOD 2 = 0 => E_1). The distance of this detector from the stop line is determined by rule #10 and the codon 63 (63 MOD 40 = 23 m).
- Similarly, the demand and extension detectors placed in movement 2 are defined by codons 11 (11 MOD 2 = 1 => Null), 22 (22 MOD 2 = 0 => E_2), and 100 (100 MOD 40 = 20 m), respectively.
- The symbol B expends to determine the type of the next phase (basic, insert, or terminate sequence) By applying rule #4 to the codon 99, a next basic phase is selected. (99 MOD 3 = 0 => A).

The complete design of the signal plan is reached by continuing to decode the chromosome in the same way. As shown in Fig. 8, the resulting design includes four phases. Phases A, B, and C are basic phases, while I is an insert phase. Table 3 presents the parameter values associated with these phases.

In this design, crosswalks unserved in phase A need to have pedestrian pushbuttons because phases B and C (that serve these crosswalks) will not activate without vehicle requests. Conversely, the other intersection crosswalks do not need to have pushbuttons since they are served in phase A, which is activated every cycle regardless of the vehicles' presence (There are no demand detectors used within this phase).

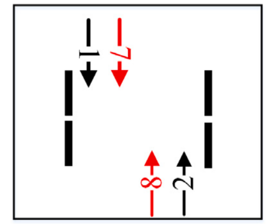


Fig. 7. Phase "A" composition.

The flowchart presented in Fig. 9 shows the control-flow diagram of the yielded design within a single cycle operation. During the dynamic simulation, the control system executes this diagram every time step to determine the signal indication (active phase A, B, I, or C) in the next time step.

3.1.7. Genetic operators

Genetic operators, i.e., selection, crossover, and mutation are applied to evolve the chromosomes and generate signal plans. These operators are described as follows:

- **The selection** operator chooses the chromosomes from the current generation that will be used to generate the next generation. In this implementation, proportionate selection [38] was used, which assigns probability of being selected to solutions based on their fitness values. The probability of selection of a chromosome is given by:

$$p_i = \frac{1/f_i}{\sum_{n=1}^{N} 1/f_n} \tag{3}$$

where f_i is the fitness value of individual i. N is the number of chromosomes in the generation.
- **The crossover** operator generates new chromosomes by mixing two of the selected chromosomes from the current generation. In this study, a two point-block crossover strategy was used, as illustrated in Fig. 10. It selects two random points in each parent chromosome and exchanges the information between the two points to generate the new chromosome. As discussed above, chromosomes are structured in equally sized blocks that correspond to phases. The crossover points were restricted to the boundaries between blocks, so that only complete phases (i.e., movements, parameters, and detectors) are exchanged. Without this restriction, over 99% of the new chromosomes were infeasible. A constraint on the maximum length of a chromosome was also included to avoid the bloat effect [59]. Violation of this constraint leads to trimming the new chromosome to the maximum.
- **Mutation** randomly alters a small part of the selected chromosome. The creep mutation method, which is shown in Fig. 11, is used in this application. It is appropriate for chromosomes that are made of integer codons. It randomly selects a pre-defined fraction of codons and changes their value randomly [59]. Thus, it changes some information of the phases.

As with most heuristic approaches, the proposed method does not guarantee convergence to a globally optimal solution. However, the grammar structure and operator design help maintain feasibility and guide the search toward practical, high-quality configurations.

3.2. Simulation model

The simulation model represents both the movements of individual road users (i.e., passenger car and transit vehicles and pedestrians)

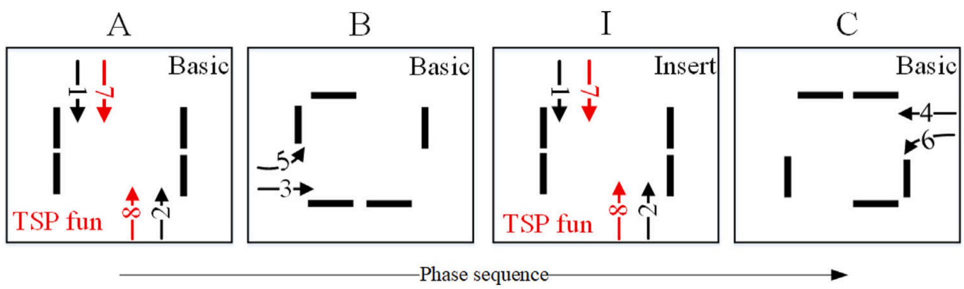


Fig. 8. Generated phases of the traffic signal control plan.

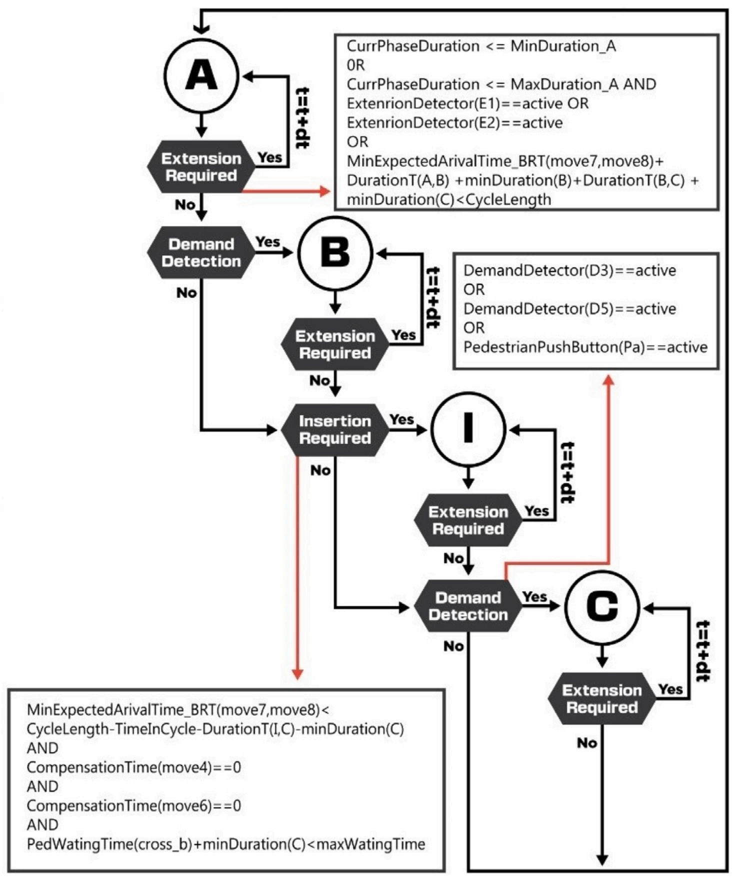


Fig. 9. Control-logic diagram and associated code, demonstrating the decision-making process in traffic signal control.

Table 3
Parameter values for each phase in the traffic signal control plan.

Phase	Min [s]	Max [s]	Importance index	First movement			Second movement		
				Dd	Ed	Ed_dist [m]	Dd	Ed	Ed_dist [m]
A	6	15	9	null	E_1	23	null	E_2	20
B	4	8	3	D_3	null	–	null	E_5	14
I	1	–	4	–	–	–	–	–	–
C	5	9	7	D_4	E_4	16	D_6	E_6	17

and the signal control plans, as shown in Fig. 12. The inputs to the simulation model include intersection geometry and information on the traffic demands of all road users. Within each dynamic simulation stage, the traffic flow model receives the traffic light indication from the signal controller and releases the queues accordingly. It updates the detectors' states and transfers them to the simulated signal plans. After receiving detector states, the controller determines the traffic light indications for the next time step. These indications are transferred back and applied in the traffic flow model. The simulation model can output various measures of performance, such as queue lengths, person delays, and number of vehicle's stops.

In this application, the MESCOP mesoscopic traffic simulation model was used. It has proven to be computationally efficient compared to the microscopic models previously used for similar purposes. Additionally, this class of traffic models maintains the levels of detail required to simulate complex intersection systems. For a detailed description of the MESCOP model, see [2,19].

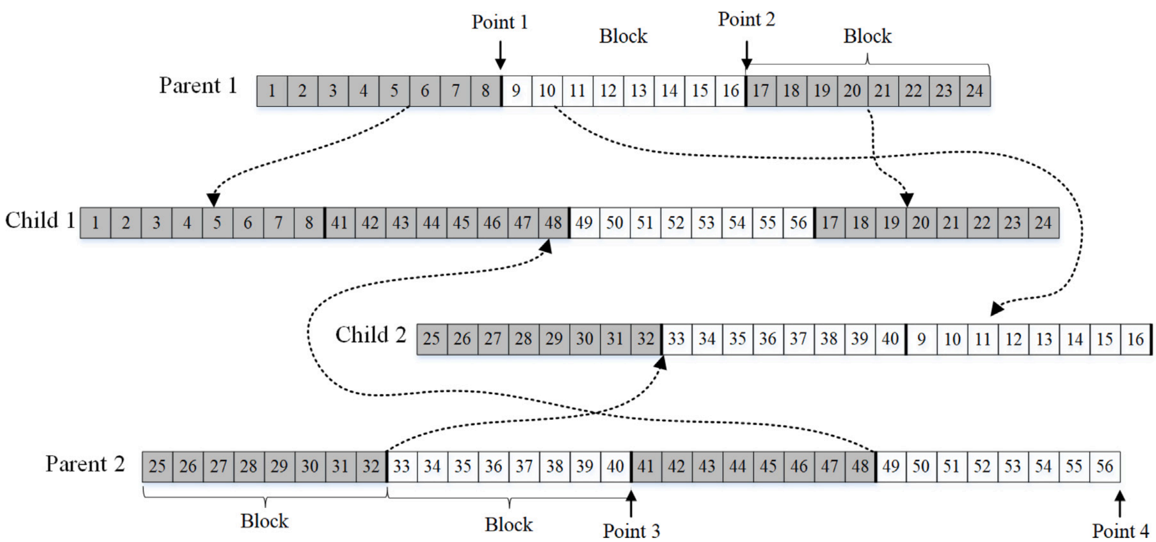


Fig. 10. An example of a two-point block crossover strategy.

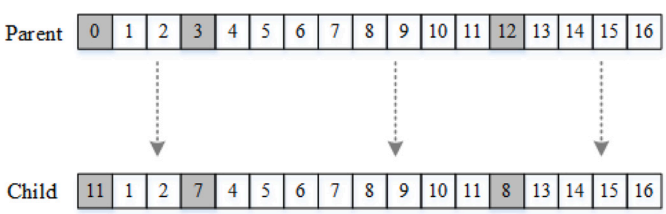


Fig. 11. An example of a creep mutation strategy.

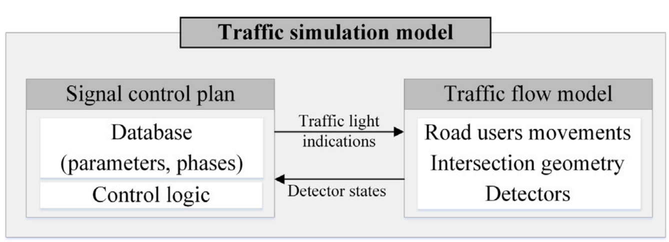


Fig. 12. Simulation setup for evaluating the candidate signal control plans.

3.3. Performance measures

Optimization can incorporate a variety of performance measures, including throughput, average delays, stops, and travel times. Average person delay is considered a valuable performance measure by several studies [18,60–62]. Accordingly, this application strives to minimize the expected value of the average delay in the system. The vehicle's delay is determined by the time it takes to enter and exit the queue. In addition, each vehicle type is assumed to carry a specific number of occupants. The pedestrians' delay refers to the time between their arrival at the crosswalk and their start crossing. Based on multiple simulation runs, the following formula is used to calculate the performance measures for each candidate signal control plan:

$$\min_{\theta} E\left[d\left(\theta\right)\right] = \frac{1}{R}\left[\sum_{r} \frac{\sum_i \sum_n d_{nr}(\theta) w_i \delta_{nri}}{\sum_i \sum_n w_i \delta_{nri}}\right] \tag{4}$$

s.t.

$$\theta^L \leq \theta \leq \theta^U \tag{5}$$

$$g\left(\theta\right) \leq 0 \tag{6}$$

Where, R represents the number of replications used in evaluating each candidate signal control plan. d_{nr} is the delay to vehicle (or pedestrian) n in replication r. w_i is the weight of a vehicle of type i, which captures the number of travelers in the vehicle. δ_{nri} is an indicator variable that takes the value 1 if vehicle n is of type i (e.g., car, bus, pedestrian) in replication r, and 0 otherwise. θ is the vector of decision variables that define the candidate signal control plan, including phase compositions, detector configurations, phase durations, and priority settings. θ^L and θ^U are their lower and upper bounds, respectively. g represents additional constraints that may be imposed on the control plan and its parameters, such as bounds on the minimum and maximum green times. These constraints are integrated into the grammar developed for this application.

The run-time complexity of the proposed grammar-based design process is determined by several components, including fitness evaluation (simulation replications), chromosome decoding, and genetic operations. Let:

- **G**: number of generations
- **P**: population size
- **L**: average chromosome length (fixed at 12 codons × number of signal phases)
- **R**: number of replications per candidate
- $\mathbf{T_{sim}}$: average runtime of one simulation replication
- $\mathbf{T_{dec}}$: average time to decode one chromosome
- $\mathbf{T_{ops}}$: average time to apply genetic operators to one chromosome

Per generation, the algorithm performs the following tasks:

1. **Decoding** all chromosomes: $O(P \cdot T_{dec}) = O(P \cdot L)$, since decoding traverses the grammar once per codon.
2. **Applying genetic operators**: Selection and crossover operate once per chromosome, giving cost $O(P)$. Creep mutation operates at the codon level, leading to a complexity of $O(P \cdot L)$. Thus, the overall cost per generation is $O(P \cdot L)$.
3. **Fitness evaluation** (simulation replications): $O(P \cdot R \cdot T_{sim})$, where each candidate plan is evaluated with R independent replications. The value of T_{sim} depends on several factors, including the number of road users, the number of actions per road user, the number of control actions, the total number of simulation time steps, etc.

Among these components, the simulation time (T_{sim}) is by far the most time-consuming, making it the primary factor that determines the overall run-time complexity.

$$O\left(G \cdot [P \cdot L + P \cdot R \cdot T_{\mathrm{sim}}]\right) \approx O(G \cdot P \cdot R \cdot T_{\mathrm{sim}}) \quad (7)$$

This expression reflects the main computational contributors and the expected scaling behavior of the proposed approach, with simulation time being the dominant factor. For further empirical insights into simulator run-time behavior under varying traffic flow levels, readers are referred to [2].

4. Case study

4.1. Intersection and control

The automatic control design system was applied to an isolated intersection in Haifa, Israel, which is shown in Fig. 13. It includes 12 vehicle movements: Six signalized vehicle movements (1 through 6), four free right-turn movements (9 through 12), and two Bus Rapid Transit (BRT) movements (7, 8), operating on dedicated bus lanes (marked in red). In addition, there are eight signalized pedestrian crosswalks (a through h) and four unsignalized ones (i through l). It currently operates an actuated traffic signal plan that uses presence detectors on all signalized vehicle movements for both extension (E1, E2, E3, E4, E5, E6) and demand (D2, D5) tasks. The extension detectors are located between 6 and 24 meters upstream of the intersection. Demand detectors are placed at the intersection's stop lines. In addition, each BRT lane contains three presence detectors: two upstream detectors (150 to 600 m) to detect approaching BRT vehicles, and one downstream used to cancel priority once BRT vehicles cross the stop line.

The design traffic flows, also shown in Fig. 13 are based on traffic count measurements in the morning peak hour (7:00 to 8:00 a.m.).

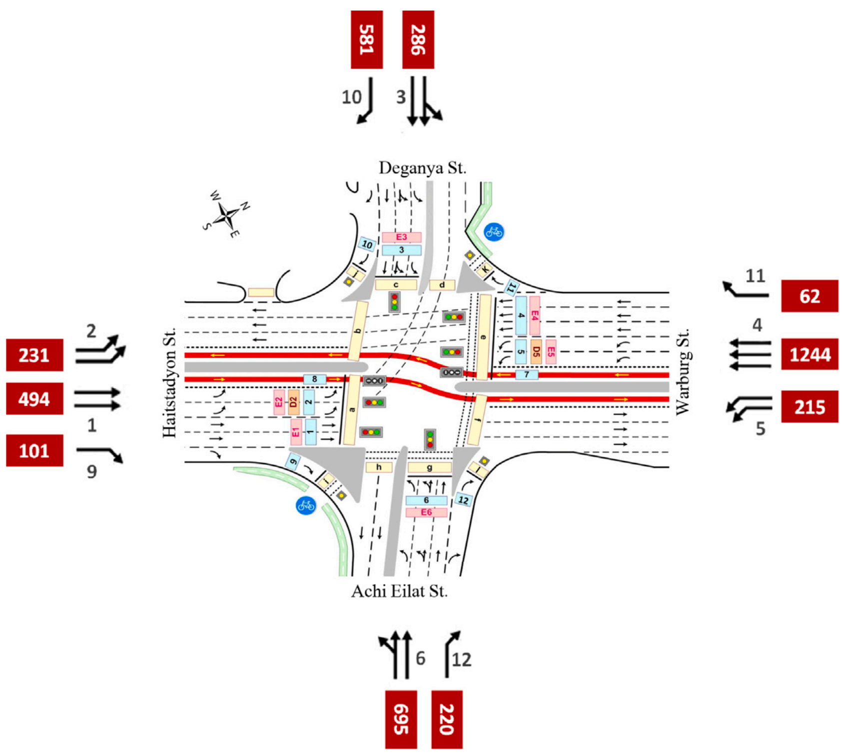


Fig. 13. Detailed layout of the case study intersection, illustrating movement and detector placements.

Fig. 14. Case study intersection modeling by the microscopic simulation model Aimsun.

12 BRT vehicles per hour arrive in each direction (movements 7 and 8). Estimation of pedestrian flow was done using CCTV (closed-circuit television) camera data. The pedestrian flow was set at 25 pedestrians crossing the intersection per hour, from each side.

Although the current study focuses on isolated intersections, the proposed framework can be adjusted to handle coordinated intersections. This can be achieved by simultaneously evolving signal plans for multiple adjacent intersections, using the same grammar-based representation to encode and optimize full signal plans. Such an extension would involve modifying the grammar to allow encoding of multiple intersections within a single chromosome. Similar to the system developed in [19], where the timing parameters of adjacent intersections were simultaneously optimized to improve corridor performance and achieve coordination, the proposed approach would enable the optimization of the full signal plan structure, including timing parameters, phase sequences, detectors placement, and priority rules.

4.2. Experiment

The experiment consisted of two steps: generating two different control plans using the proposed system and evaluating them using an independent traffic simulation model. The second step also incorporated comparing these new plans with the existing plan.

The first new signal plan was developed using the peak hour vehicle flows shown in Fig. 13, and the other one was generated by increasing these flows by 20% to assess the ability of the proposed system to generate signal control plans that maintain robust performance under different traffic conditions, particularly extreme ones. In the optimization process for the two new plans, the objective function value of candidate plans was estimated using ten parallel MESCOP replications.

The performance of the three control plans (existing and the two automatically designed) was then independently evaluated, separate from the optimization process. For this purpose, AIMSUN [63], a microscopic traffic simulation model, was used as the evaluation environment, illustrated in Fig. 14. Twenty simulation replications were employed in the evaluation.

From this, it can be concluded that the optimization process of the new traffic signal plans is carried out offline using the proposed system. However, their operation is online, executing logic functions such as phase extension, termination, skipping, and priority actions based on real-time traffic flow data (in this case, using AIMSUN).

5. Results and discussion

5.1. Automatically designed signal plan

The convergence of the automatic design process is shown in Fig. 15, which graphs the average person delay of the best solution in each generation. The optimal signal plan was obtained in generation 49. The entire optimization process, with 80 generations and 400 chromosomes in each, took approximately 18 h to complete. In comparison, manual design typically requires significantly more time and effort, often representing days' worth of expensive person-hours from experienced design engineers. In addition, the required time can extend well beyond that, depending on the skill level of the designer, intersection complexity, agency coordination, and the inclusion of advanced functions. In contrast, the proposed method incorporates such considerations directly into the optimization problem, such as the inclusion of pedestrian pushbuttons and cycle length constraints.

Compared to the existing plan, the automatically designed plan reduced the average person delay by 25% (from 39.8 to 29.8 s) and the maximum queue length by 35% (from 11.2 to 7.3 vehicles). These improvements were statistically significant ($p-value < 0.001$). Reduced delays are due to a reduction in vehicles (−27.9%, $p-value < 0.001$) and pedestrian (−42.8%, $p-value < 0.001$) delays. This reduction was accompanied by a marginal increase (4.9%, $p-value = 0.502$) in BRT delays. Thus, the improvement in delays for vehicles when using the automatically designed plan mainly result from reducing the adverse consequences of priority activities that affect them.

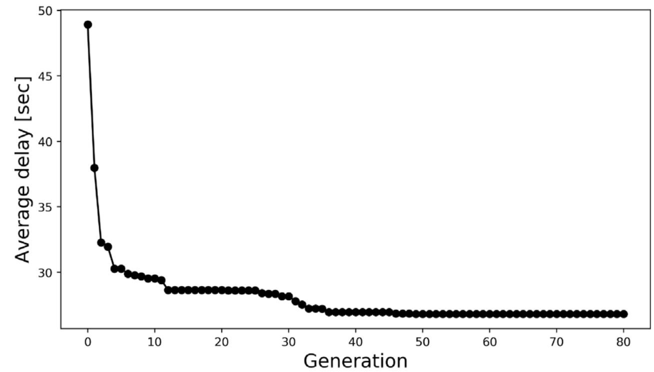


Fig. 15. Convergence properties of GE in the optimization process for designing a new signal plan with base flows.

5.2. Automatically designed signal plan with increased vehicle flows

Fig. 16 graphs the average person delay of the best solution in each generation during the optimization process for designing a new signal plan with increased vehicle flows. The optimal signal plan was obtained in generation 28. The entire optimization process, with 80 generations and 400 chromosomes in each, took approximately 18 h to complete. Accordingly, this is similar to the time taken to generate the signal plan with base flows, demonstrating that increasing the traffic flows does not significantly affect the optimization time, indicating that the proposed system is still efficient under challenging traffic conditions.

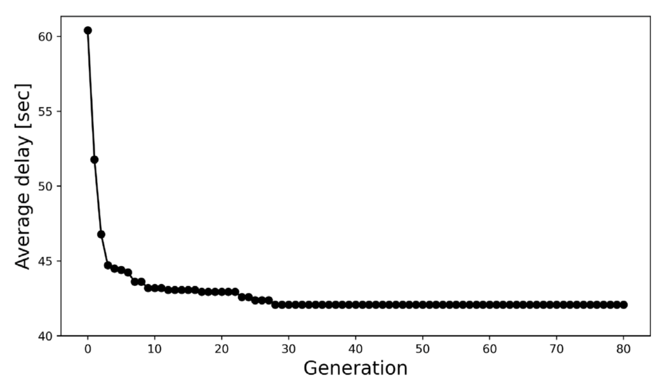


Fig. 16. Convergence properties of GE in the optimization process for designing a new signal plan with increased vehicle flows.

To effectively manage the increased vehicle flows, particularly at the through movement 4, which spills back and blocks the left turn movement 5, as shown in Fig. 17, the phase sequence in this plan was modified compared to the automatically designed plan with base vehicle flows. As illustrated in Fig. 18, this modification included swapping the order of phases A and B to activate phase B first. This

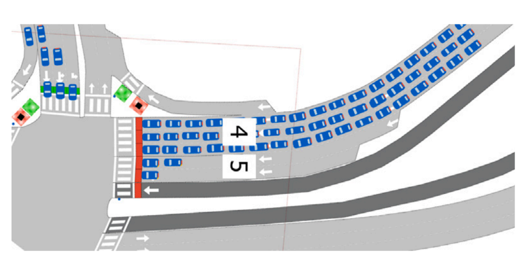


Fig. 17. Visual analysis of queue spillback and its impact on adjacent movements.

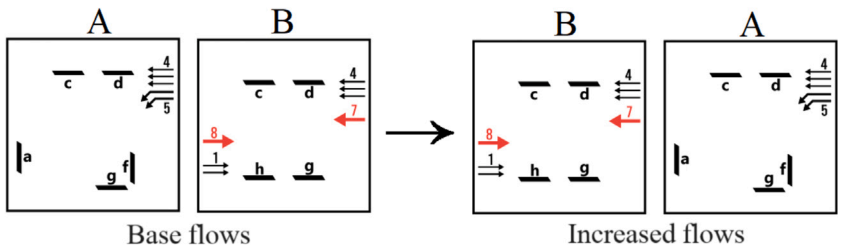


Fig. 18. Reordering phases A and B to mitigate queue spillback issues.

way, movement 4 queues that had accumulated during the red light are cleared first, allowing left-turning vehicles to join the queue of movement 5 in preparation for the next phase.

In the automatically designed plan with increased vehicle flows, the average person delay and the maximum queue length increased by 43% (from 29.8 to 43.6 s, $p - value < 0.001$) and 78% (from 7.3 to 13 vehicles, $p - value < 0.001$), respectively, compared to the plan designed for base flows. However, compared to the existing plan, that was manually designed based on the base flows — the average delays and the maximum queue length increased only by 7% (from 39.8 to 43.6 s, p-value $= 0.011$) and 16% (from 11.2 to 13 vehicles, $p{-}value = 0.024$). This slight increase occurred despite the vehicle flows growing by 20%, of which the vehicle flows accounted for 78% of the total road users. These findings demonstrate the ability of the proposed system to generate automatically effective signal control plans under increasing traffic demands.

6. Conclusion

This paper presented a method for automatic traffic signal plan design. It is based on grammatical decoding and evolution of vector-coded signal plans. The developed grammar provides a generic mechanism to represent a wide range of actuated signal plans for different intersection layouts. Furthermore, the grammar can be easily customized to satisfy various design requirements (e.g., using alternative detection technologies). Within the optimization framework, candidate plans are evaluated using a traffic simulation model. The automated design can generate actuated signal control plans with transit priority without requiring human intervention.

The automated design process was demonstrated with a case study. The results show significant improvements in the average person delay compared to the existing plan. Moreover, the automated design could successfully replace the work of experienced traffic engineers at a lower cost, with less effort, and in lower design time. The results also demonstrate that the proposed method remains effective in terms of performance and running times under challenging traffic conditions. Future research in this direction may expand the application to other intersection configurations or to systems of coordinated intersections, particularly those incorporating transit-priority functions, and investigate techniques for improving running time and overall computational efficiency. In addition, alternative grammar-based optimization methods could be explored to further enhance search efficiency relative to the evolutionary approach used in this study.

CRediT authorship contribution statement

Mahmud Keblawi: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Tomer Toledo:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was partially funded by grants from the Israeli Smart Transportation Research Center and partially by The Ministry of Science and Technology, Israel.

Appendix A. Code availability

The source code used in this study is publicly available at the following GitHub repository:
`github.com/mahmudkeb/GE-Traffic-Signal-Control`

Appendix B. Simulation video materials

To support the findings presented in this study, four videos have been provided and are available at the following link: https://drive.google.com/drive/folders/1JBVqb4oYniK0dm2rIX6uNPrKO_WVynDO?usp=sharing

The videos include:

- A 3D animation of the simulated intersection;
- A video of the existing (manual) signal plan in operation;
- A video of the automatically generated signal plan;
- A video illustrating the application of transit signal priority.

These materials are intended to help visualize the behavior and impact of different signal control strategies discussed in the paper.

Data availability

No data was used for the research described in the article.

References

[1] INRIX, Congestion Costs Each American 97 hours, 1,348 $ A Year, 2019, https://inrix.com/press-releases/scorecard-2018-us/.
[2] T. Balasha, T. Toledo, MESCOP: A mesoscopic traffic simulation model to evaluate and optimize signal control plans, Transp. Res. Rec. 2488 (1) (2015) 1–9.
[3] M. Keblawi, H. Maripini, J. Kim, M. Hickman, Z. Zheng, M. Yildirimoglu, Integrating road network operations planning into real-time traffic management: A conceptual framework, Transportation Research Interdisciplinary Perspectives 32 (2025) 101525.
[4] B. Park, I. Yun, Evaluation of Stochastic Optimization Methods of Traffic Signal Control Settings for Coordinated Actuated Signal Systems, Tech. Rep., 2006.
[5] F.V. Webster, Traffic Signal Settings, Tech. Rep., 1958.
[6] H.C. HCM, HCM2010, Transp. Res. Board Natl. Res. Counc. Wash. DC 1207 (2010).
[7] D. Husch, J. Albeck, Trafficware SYNCHRO 6 User Guide, vol. 11, TrafficWare, Albany, California, 2004.

[8] D. Hale, Traffic Network Study Tool–TRANSYT-7F, United States Version, Mc-Trans Center in the University of Florida, 2005.
[9] E. Chang, C.J. Messer, Arterial signal timing optimization using PASSER II-90-program user's manual, 1991.
[10] N. Chaudhary, C. Chu, PASSER V: Software for Timing Signalized Arterials, Texas Transportation Institute, Texas A&M University System, College Station, 2002.
[11] B. Park, J. Schneeberger, et al., Evaluation of Traffic Signal Timing Optimization Methods Using a Stochastic and Microscopic Simulation Program, Tech. Rep., Virginia Transportation Research Council, 2003.
[12] A. Stevanovic, P.T. Martin, J. Stevanovic, VisSim-based genetic algorithm optimization of signal timings, Transp. Res. Rec. 2035 (1) (2007) 59–68.
[13] J. Branke, P. Goldate, H. Prothmann, Actuated traffic signal optimization using evolutionary algorithms, in: Proceedings of the 6th European Congress and Exhibition on Intelligent Transport Systems and Services, Citeseer, 2007, pp. 203–225.
[14] P. Li, M.M. Abbas, R. Pasupathy, L. Head, Simulation-based optimization of maximum green setting under retrospective approximation framework, Transp. Res. Rec. 2192 (1) (2010) 1–10.
[15] B. Park, J. Lee, Optimization of coordinated–actuated traffic signal system: Stochastic optimization method based on shuffled frog-leaping algorithm, Transp. Res. Rec. 2128 (1) (2009) 76–85.
[16] I. Yun, B. Park, Stochastic optimization for coordinated actuated traffic signal systems, J. Transp. Eng. 138 (7) (2012) 819–829.
[17] J. Stevanovic, A. Stevanovic, P.T. Martin, T. Bauer, Stochastic optimization of traffic control and transit priority settings in VISSIM, Transp. Res. Part C: Emerg. Technol. 16 (3) (2008) 332–349.
[18] A. Stevanovic, J. Stevanovic, C. Kergaye, P. Martin, Traffic control optimization for multi-modal operations in a large-scale urban network, in: 2011 IEEE Forum on Integrated and Sustainable Transportation Systems, IEEE, 2011, pp. 146–151.
[19] T. Toledo, T. Balasha, M. Keblawi, Optimization of actuated traffic signal plans using a mesoscopic traffic simulation, J. Transp. Eng. Part A: Syst. 146 (6) (2020) 04020041.
[20] Z. Li, M. Shahidehpour, S. Bahramirad, A. Khodaei, Optimizing traffic signal settings in smart cities, IEEE Trans. Smart Grid 8 (5) (2016) 2382–2393.
[21] X. Li, J.-Q. Sun, Signal multiobjective optimization for urban traffic network, IEEE Trans. Intell. Transp. Syst. 19 (11) (2018) 3529–3537.
[22] C. Ma, P. Liu, Intersection signal timing optimization considering the travel safety of the elderly, Adv. Mech. Eng. 11 (12) (2019) 1687814019897216.
[23] E. Sofronova, A. Diveev, Signal timing optimization by VarGA: Case study, in: 2024 IEEE Intelligent Vehicles Symposium, IV, IEEE, 2024, pp. 3120–3125.
[24] H. Ceylan, Optimal design of signal controlled road networks using differential evolution optimization algorithm, Math. Probl. Eng. 2013 (1) (2013) 696374.
[25] Z. Cakici, Y.S. Murat, A differential evolution algorithm-based traffic control model for signalized intersections, Adv. Civ. Eng. 2019 (1) (2019) 7360939.
[26] O. Baskan, A multiobjective bilevel programming model for environmentally friendly traffic signal timings, Adv. Civ. Eng. 2019 (1) (2019) 1638618.
[27] J. García-Nieto, E. Alba, A.C. Olivera, Swarm intelligence for traffic light scheduling: Application to real urban areas, Eng. Appl. Artif. Intell. 25 (2) (2012) 274–283.
[28] K. Han, Y. Sun, H. Liu, T.L. Friesz, T. Yao, A bi-level model of dynamic traffic signal control with continuum approximation, Transp. Res. Part C: Emerg. Technol. 55 (2015) 409–431.
[29] P. Jiao, R. Li, Z. Li, Pareto front–based multi-objective real-time traffic signal control model for intersections using particle swarm optimization algorithm, Adv. Mech. Eng. 8 (8) (2016) 1687814016666042.
[30] P.W. Shaikh, M. El-Abd, M. Khanafer, K. Gao, A review on swarm intelligence and evolutionary algorithms for solving the traffic signal control problem, IEEE Trans. Intell. Transp. Syst. 23 (1) (2020) 48–63.
[31] A. Baldominos, Y. Saez, P. Isasi, Evolutionary design of convolutional neural networks for human activity recognition in sensor-rich environments, Sensors 18 (4) (2018) 1288.
[32] J. Colmenar, J. Hidalgo, S. Salcedo-Sanz, Automatic generation of models for energy demand estimation using grammatical evolution, Energy 164 (2018) 183–193.
[33] A. Zhao, J. Xu, M. Konaković-Luković, J. Hughes, A. Spielberg, D. Rus, W. Matusik, Robogrammar: graph grammar for terrain-optimized robot design, ACM Trans. Graph. 39 (6) (2020) 1–16.
[34] M. O'Neill, C. Ryan, Grammatical evolution, IEEE Trans. Evol. Comput. 5 (4) (2001) 349–358.
[35] D. Samarasinghe, M. Barlow, E. Lakshika, K. Kasmarik, Grammar-based autonomous discovery of abstractions for evolution of complex multi-agent behaviours, Swarm Evol. Comput. 73 (2022) 101106.
[36] I. Tsoulos, D. Gavrilis, E. Glavas, Neural network construction and training using grammatical evolution, Neurocomputing 72 (1–3) (2008) 269–277.
[37] F. Ahmadizar, K. Soltanian, F. AkhlaghianTab, I. Tsoulos, Artificial neural network development by means of a novel combination of grammatical evolution and genetic algorithm, Eng. Appl. Artif. Intell. 39 (2015) 1–13.
[38] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1) (1992) 66–73.
[39] F. Assunçao, N. Lourenço, P. Machado, B. Ribeiro, Towards the evolution of multi-layered neural networks: a dynamic structured grammatical evolution approach, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2017, pp. 393–400.
[40] A. Baldominos, Y. Saez, P. Isasi, Evolutionary convolutional neural networks: An application to handwriting recognition, Neurocomputing 283 (2018) 38–52.
[41] F. Assunção, N. Lourenço, P. Machado, B. Ribeiro, DENSER: deep evolutionary network structured representation, Genet. Program. Evolvable Mach. 20 (1) (2019) 5–35.
[42] F. Castejón, E. Carmona, Automatic design of electronic amplifiers using grammatical evolution, 2013, pp. 703–712, Actas de Multiconferencia CAEPIA-13.
[43] F. Castejón, E.J. Carmona, Automatic design of analog electronic circuits using grammatical evolution, Appl. Soft Comput. 62 (2018) 1003–1018.
[44] F. Castejón, E.J. Carmona, Introducing modularity and homology in grammatical evolution to address the analog electronic circuit design problem, IEEE Access 8 (2020) 137275–137292.
[45] M. Keblawi, T. Toledo, Automatic design of optimal actuated traffic signal plans with active transit priority, IEEE Trans. Intell. Transp. Syst. (2023).
[46] M. Eom, B.-I. Kim, The traffic signal control problem for intersections: a review, Eur. Transp. Res. Rev. 12 (1) (2020) 1–20.
[47] T. Urbanik, A. Tanaka, B. Lozner, E. Lindstrom, K. Lee, S. Quayle, S. Beaird, S. Tsoi, P. Ryus, D. Gettman, et al., Signal timing manual, vol. 1, Transportation Research Board Washington, DC, 2015.
[48] C.A.T. Vilarinho, Intelligent Traffic Signal Control (Ph.D. thesis), Universidade do Porto (Portugal), 2019.
[49] Y. Feng, K.L. Head, S. Khoshmagham, M. Zamanipour, A real-time adaptive signal control in a connected vehicle environment, Transp. Res. Part C: Emerg. Technol. 55 (2015) 460–473.
[50] M. Noaeen, A. Naik, L. Goodman, J. Crebo, T. Abrar, Z.S.H. Abad, A.L. Bazzan, B. Far, Reinforcement learning in urban network traffic signal control: A systematic literature review, Expert Syst. Appl. 199 (2022) 116830.
[51] H. Kim, Y. Cheng, G.-L. Chang, An arterial-based transit signal priority control system, Transp. Res. Rec. 2672 (18) (2018) 1–14.
[52] A. Agrahari, M.M. Dhabu, P.S. Deshpande, A. Tiwari, M.A. Baig, A.D. Sawarkar, Artificial intelligence-based adaptive traffic signal control system: A comprehensive review, Electronics 13 (19) (2024) 3875.
[53] P. Michailidis, I. Michailidis, C.R. Lazaridis, E. Kosmatopoulos, Traffic signal control via reinforcement learning: A review on applications and innovations, Infrastructures 10 (5) (2025) 114.
[54] B. Hellinga, F. Yang, J. Hart-Bishop, Estimating signalized intersection delays to transit vehicles: Using archived data from automatic vehicle location and passenger counting systems, Transp. Res. Rec. 2259 (1) (2011) 158–167.
[55] E. Chang, A. Ziliaskopoulos, Data challenges in development of a regional assignment: simulation model to evaluate transit signal priority in chicago, Transp. Res. Rec. 1841 (1) (2003) 12–22.
[56] C. Diakaki, M. Papageorgiou, V. Dinopoulou, I. Papamichail, M. Garyfalia, State-of-the-art and-practice review of public transport priority strategies, IET Intell. Transp. Syst. 9 (4) (2015) 391–406.
[57] R.I. McKay, N.X. Hoai, P.A. Whigham, Y. Shan, M. O'neill, Grammar-based genetic programming: a survey, Genet. Program. Evolvable Mach. 11 (2010) 365–396.
[58] J.W. Backus, F.L. Bauer, J. Green, C. Katz, J. McCarthy, P. Naur, A.J. Perlis, H. Rutishauser, K. Samelson, B. Vauquois, et al., Revised report on the algorithmic language algol 60, Comput. J. 5 (4) (1963) 349–367.
[59] A.E. Eiben, J.E. Smith, et al., Introduction to Evolutionary Computing, vol. 53, Springer, 2003.
[60] Q. He, K.L. Head, J. Ding, PAMSCOD: Platoon-based arterial multi-modal signal control with online data, Transp. Res. Part C: Emerg. Technol. 20 (1) (2012) 164–184.
[61] E. Christofa, I. Papamichail, A. Skabardonis, Person-based traffic responsive signal control optimization, IEEE Trans. Intell. Transp. Syst. 14 (3) (2013) 1278–1289.
[62] M.E.C. Bagdatli, A.S. Dokuz, Vehicle delay estimation at signalized intersections using machine learning algorithms, Transp. Res. Rec. 2675 (9) (2021) 110–126.
[63] Aimsun, Aimsun Next 23 User's Manual, Barcelona, Spain, aimsun next 23.0.0 Edition, 2023, (Accessed 19 July 2023). [Online] URL https://docs.aimsun.com/next/23.0.0/.