



Self-organisation in traffic signal control algorithms under light traffic conditions

SJ Movius*

JH van Vuuren†

Received: 21 September 2018; Revised: 26 February 2019; Accepted: 15 March 2019

Abstract

Fixed-time control and vehicle-actuated control are two distinct types of traffic signal control. The latter control method involves switching traffic signals based on detected traffic flows and thus offers more flexibility (appropriate for lighter traffic conditions) than the former, which relies solely on cyclic, predetermined signal phases that are better suited for heavier traffic conditions. The notion of self-organisation has relatively recently been proposed as an alternative approach towards improving traffic signal control, particularly under light traffic conditions, due to its flexible nature and its potential to result in emergent behaviour. The effectiveness of five existing self-organising traffic signal control strategies from the literature and a fixed-control strategy are compared in this paper within a newly designed agent-based, microscopic traffic simulation model. Various shortcomings of three of these algorithms are identified and algorithmic improvements are suggested to remedy these deficiencies. The relative performance improvements resulting from these algorithmic modifications are then quantified by their implementation in the aforementioned traffic simulation model. Finally, a new self-organising algorithm is proposed that is particularly effective under lighter traffic conditions.

Key words: Self-organisation, traffic signal control, microscopic traffic simulation, vehicle actuation.

1 Introduction

As a result of the economic, environmental and social consequences of traffic congestion, improved optimisation of traffic signal timings has been suggested as an appropriate approach towards alleviating this problem [3, 4, 11]. Fixed-time control and vehicle-actuated

*Stellenbosch Unit for Operations Research in Engineering, Department of Industrial Engineering, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, email: sjmovius@gmail.com

†(Fellow of the Operations Research Society of South Africa) Corresponding author: Stellenbosch Unit for Operations Research in Engineering, Department of Industrial Engineering, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, email: vuuren@sun.ac.za

control are two main classes of traffic signal control strategies [10], with the latter involving a more flexible control approach in which traffic signals switch due to the detected presence of vehicles, which is particularly suitable for use under light traffic conditions. A fixed-time control strategy, on the other hand, makes use of predetermined fixed signal cycle times and is better suited for heavy traffic conditions [12].

Self-organisation has been proposed as a promising alternative approach toward traffic signal control, and occurs when a system achieves an increase in order, or structure, without any form of external control [2, 6, 15, 16]. Emergence occurs once a system exhibits a novel behavioural outcome on a macro-level that stems directly from micro-level interactions, although this novel behaviour is not reducible to individual elements at the micro-level [2, 5, 13, 16]. Self-organisation has the potential to lead to an emergence of coordination between intersections in a traffic network.

In this paper, five existing self-organising traffic signal control algorithms and a fixed-time control strategy are described and compared statistically with respect to a number of *performance measure indicators* (PMIs) within a newly designed microscopic traffic simulation model in which left-hand side driving is assumed. These algorithms include an algorithm by Gershenson and Rosenblueth [4], an algorithm by Lämmer and Helbing [11], three algorithms by Einhorn [3] and finally a standard fixed-time control strategy. The purpose of this comparison is to determine which traffic signal control algorithms are superior under different traffic flow densities. It is shown that a selection of the self-organising algorithms perform better under light traffic conditions, while fixed-time control achieves better results under heavy traffic conditions, as anticipated. A similar experiment was carried out in a PhD dissertation by Einhorn [3]. The aim of this paper is three-fold: First, to test the reproducibility of the results obtained by Einhorn, secondly to suggest improvements to a subset of the aforementioned algorithms and to quantify statistically the degree of improvement thus achieved, and finally to propose and test the efficiency of a new self-organising algorithm that operates by clustering vehicles into platoons and switches signals so as not to separate the vehicles of a platoon.

The paper is organised as follows. In §2, the logic of the five existing self-organising algorithms and that of the fixed-time control algorithm are described. The test bed simulation model is then described in §3. This description includes specifications of entities in the modelling framework, the output generated by the model, the statistical analyses performed in respect of the model output data, and the model validation process followed. Simulation results obtained for the six algorithms are presented in §4 under both light and heavy traffic conditions. This is followed by a number of suggested improvements for three of the five self-organising algorithms in §5. Simulation results obtained for the improved algorithms are next compared to those of their original counterparts in §6, again under both light and heavy traffic conditions. A novel self-organising algorithm is finally proposed in §7, after which all seven algorithms (including the new algorithm and improved algorithmic versions) are compared in §8. The paper closes in §9 with a summary of the findings and suggestions for future work in §10.

2 Algorithmic implementations

The logic of each of the five existing aforementioned self-organising traffic signal control algorithms and that of the fixed-time control strategy are described in this section, addressing the authors' interpretation of certain unclear aspects of some of the algorithms. These algorithms include a fixed-time algorithm (referred to here as *Fixed*) [21], a rule-based algorithm proposed by Gershenson and Rosenblueth [4] (referred to here as *Gersh*), an algorithm by Lämmer and Helbing [11] (referred to here as *LH*) that was inspired by the natural occurrence of self-organisation found in oscillations of pedestrian flows through narrow bottlenecks, as well as three algorithms proposed by Einhorn [3]. The first of the latter three algorithms is a predictive algorithm inspired by the virtual costs associated with inventory control (referred to as the *I-TSCA*), while the second algorithm was inspired by the pull and push forces present in the chemical process of osmosis (referred to as the *O-TSCA*). The final algorithm is a hybrid of the first two algorithms proposed by Einhorn [3] (referred to here as *Hybrid*).

The signal cycle in each of these algorithms (with the exception of Fixed) consists of a total of 12 phases, including four green phases, each followed by an amber and all-red phase. The ordering of these phases in the context of a gridded urban street network a gridded road network is as follows: The first green phase awards green time to all vehicles travelling in the vertical direction (south to north and north to south), while the vertical right-turning vehicles may only turn on a permitted basis. This phase is followed by an amber phase (which is 3 seconds in length) and an all-red phase (which is 2 seconds in length), and these are collectively referred to as the setup time. The fourth phase is a protected right-turning phase for vehicles travelling in the vertical direction and is skipped if there are fewer than four vertical right-turning vehicles. Once again, this phase is followed by an amber and all-red phase. The final six phases are defined similarly for vehicles travelling in the horizontal direction (east to west and west to east).

2.1 A fixed-time control strategy

If a number of intersections are located relatively near to one another, it is desirable to coordinate them in such a manner that vehicles receive green signals just as they reach consecutive intersections when travelling through a transportation network at some desired speed. This is done through the determination of a suitable cycle length, an offset time and green times for various vehicle movements through the intersections.

In this paper, signalling at all intersections according to Fixed consists of a cycle of length C , measured in seconds. The value of C is calculated by utilising a formula developed by Webster [21] with the aim of minimising vehicle delays in the context of random vehicle arrivals. This formula is given by

$$C = \frac{1.5L + 5}{1.0 - Y}, \quad (1)$$

where L is the lost time per cycle (the sum of the setup times in one cycle) and Y is the sum total of the critical lane volumes divided by the saturation flow for each phase. Exclusive right-turning phases are not incorporated as the Highway Capacity Manual [7]

states that this is only recommended if the turn volume is over 100 vehicles per hour. The cycle therefore comprises two green phases and two setup times, resulting in a lost time per cycle of $L = 10$ seconds, while the saturation flow is taken as 1 800 vehicles per hour. Light traffic conditions result in a flow of 600 vehicles per hour, and thus the cycle length for light traffic conditions is $L = 30$ seconds. After subtracting the setup time from this value, the green times for each of the green phases are calculated to be 10 seconds each (an equal demand is assumed in both directions). The traffic flow under heavy traffic conditions is 1 200 vehicles per hour, resulting in a cycle length of $L = 60$ seconds. Once the setup time is subtracted, the two green phases each last 25 seconds in this case.

The distance between neighbouring intersections in the underlying gridded road network is assumed to be 385 metres, while the average speed of vehicles is 16.67m/s. The offset is calculated by obtaining the time in seconds required by a vehicle to travel from one intersection to the next. This is calculated by dividing the distance of 385 metres by the speed of 16.67m/s to obtain a time of 23 seconds. An additional two seconds are added to account for reaching the desired speed if there is a vehicle queue or a slower vehicle travelling slower than 60km/h, resulting in an offset time of 25 seconds. The intersections are coordinated in such a way that a wave of uninterrupted traffic flows in the west to east direction and another in the north to south direction.

Since turning is not allowed in [4], the right-turning lane is ignored when determining whether there are approaching vehicles within close proximity of the intersection.

2.2 The algorithm of Lämmer and Helbing

The self-organising traffic control algorithm proposed by Lämmer and Helbing [11] makes use of a stabilisation strategy and an optimisation strategy. While each of these strategies performs poorly under high traffic flow densities in isolation, they achieve a far better performance when they are combined appropriately.

Lämmer and Helbing modelled traffic flow according to fluid-dynamic principles, considering vehicle flow rates rather than individual vehicle velocities. It is therefore assumed in their model that all vehicles travel at a constant speed. The traffic environment is described in terms of the length L_i , the speed limit V_i and the saturation flow rate Q_i^{max} of an approach road i incident with an intersection. The necessary traffic dynamics are defined by an arrival rate $Q_i^{arr}(t) \leq Q^{max}$ as well as a departure rate $Q_i^{dep}(t) \leq Q^{max}$ at time t . These quantities represent the numbers of vehicles per unit time that enter and exit the intersection from approach road i , respectively. Using these flow rates, the accumulated number $N_i^{exp}(t)$ of vehicles expected to travel through the intersection under free-flowing traffic conditions at time t is given by

$$N_i^{exp}(t) = \int_{-\infty}^t Q_i^{arr}(t' - L_i/V_i) dt',$$

where L_i/V_i is the time taken to travel the distance L_i along approach road i at a speed V_i in free-flowing traffic. Due to traffic congestion, the number of vehicles that have actually departed from approach road i at time t is

$$N_i^{dep}(t) = \int_{-\infty}^t Q_i^{dep}(t') dt' \leq N_i^{exp}(t).$$

Therefore, the difference between $N_i^{exp}(t)$ and $N_i^{dep}(t)$ is the number of delayed vehicles $n_i(t)$, known as the *queue length*. The setup time is denoted by τ^0 and the remaining setup time experienced by vehicles is denoted by $\tau(t)$, where $0 \leq \tau(t) \leq \tau^0$.

The service process can be partitioned into three consecutive phases: A setup time, the time required to clear the vehicle queue and an extended green time that follows after the queue has dissipated. The optimisation strategy implemented by Lämmer and Helbing requires a forecast of the amount of green time $\hat{g}_i(t)$ from the current point in time t that is necessary to clear the queue present along approach road i . The value of $\hat{g}_i(t)$ depends on both the number of queued vehicles $n_i(t)$ and the number of vehicles still joining the queue during a period of length $\tau_i(t)$. The total time taken for the queue to dissipate is $t + \tau_i(t) + \hat{g}_i(t)$, at which point the total number of vehicles that have arrived at the intersection is equal to the number of vehicles that have departed from the intersection. This gives rise to the conservation of flow law

$$N_i^{dep}(t) + \hat{g}_i(t)Q^{max} = N_i^{exp}(t + \tau(t) + \hat{g}_i). \quad (2)$$

In this equation, $\hat{g}_i(t)$ is the largest possible solution. The second term on the left-hand side of (2) is the number of vehicles travelling through the intersection at the maximum flow rate, which can also be written as $n_i(t)$. This value includes all vehicles that are already waiting in the queue, joining the queue during the setup time, or clearing or arriving in a platoon at the moment the queue has been cleared.

A “pressure” of priority $\pi_i(t)$ is associated with the flow of traffic along approach road i , and green time is awarded to the traffic flow along approach road i if π_i is the largest priority value. The goal is to derive a formula for the priority index π_i in such a way that the signal switching rule acts to minimise the total waiting time of vehicles in both directions of approach road i . In order to determine whether it is more beneficial to continue a service phase or to service a different flow direction, it is essential that the predicted total waiting time $\hat{w}_i(t)$ of all vehicles from the stop line to the end of the clearing state, be forecast. The predicted total waiting time can be expressed as

$$\hat{w}_i(t) = w_i(t) + A_i(t) + B_i(t), \quad (3)$$

where, $w_i(t)$ is the waiting time of all vehicles on approach road i up to time t , $A_i(t)$ is the waiting time experienced by vehicles during the setup time, and $B_i(t)$ is the waiting time incurred by vehicles while the queue is being cleared. It follows from (3) that

$$\frac{d\hat{w}_i}{dt} = \frac{dw_i}{dt} + \frac{dA_i}{dt} + \frac{dB_i}{dt}. \quad (4)$$

As explained in [11], this expression may be simplified to

$$\frac{d\hat{w}_i}{dt} = \begin{cases} \hat{n}_i(t), & \text{if approach road } i \text{ is not being serviced,} \\ 0, & \text{during the entire service process of approach road } i. \end{cases} \quad (5)$$

The optimisation approach adopted in the algorithm of Lämmer and Helbing [11] is limited to the expected number of vehicles queued at or approaching the intersection along approach road i and it is assumed that there are only two competing directions. The

remaining setup times τ_1 and τ_2 , the expected numbers of vehicles \hat{n}_1 and \hat{n}_2 , and the required green times \hat{g}_1 and \hat{g}_2 are considered in the optimisation approach. It is furthermore assumed that traffic flow 1 is currently receiving green time. Therefore, there are two options:

1. The algorithm continues to serve traffic flow 1 before switching to traffic flow 2, or
2. the algorithm switches to traffic flow 2 immediately and will return to traffic flow 1 later, at the cost of an additional setup time.

In order to decide which option minimises the total waiting time, the total increase in waiting time for each option is calculated. If the first option is chosen, then traffic flow 1 is served for a total of $\tau_1 + \hat{g}_1$ seconds. According to (5), the waiting time experienced by traffic flow 2 increases at a rate of \hat{n}_2 , while the waiting time of traffic flow 1 remains constant. Therefore, the total increase in waiting time as a result of selecting the first option, would be $(\tau_1 + \hat{g}_1)\hat{n}_2$. Alternatively, if the second option were to be chosen, an additional increase of $\Delta\hat{w}_1$ would be present in the waiting time, which represents the additional setup time required to switch back to traffic flow 1 at a later stage. Traffic flow 2 is served for $\tau_2 + \hat{g}_2$ seconds, while the waiting time of traffic flow 2 increases at a rate of \hat{n}_1 . Thus, the total waiting time experienced by vehicles is $\Delta\hat{w}_1 + (\tau_2 + \hat{g}_2)\hat{n}_1$. It is therefore beneficial to continue serving traffic flow 1 as long as

$$(\tau_1 + \hat{g}_1)\hat{n}_2 < \Delta\hat{w}_1 + (\tau_2 + \hat{g}_2)\hat{n}_1. \quad (6)$$

This inequality leads to the definitions of the priority indices π_1 and π_2 which are obtained by rewriting (6) as

$$\pi_1 := \frac{\hat{n}_1}{\tau_1 + \hat{g}_1} > \frac{\hat{n}_2}{\Delta\hat{w}_1/\hat{n}_1 + \tau_2 + \hat{g}_2} =: \pi_2. \quad (7)$$

The first priority index π_1 in (7) depends solely on variables associated with traffic flow 1, whereas the second priority index π_2 exhibits the same dependence on variables related to traffic flow 2, but with an additional term $\Delta\hat{w}_1/\hat{n}_1$. Hence, the penalty for terminating the current service is given by $\Delta\hat{w}_\sigma/\hat{n}_\sigma$, where σ represents the index of the current flow receiving service and lies in the interval $(0, \tau_\sigma^0)$. Since the penalty is only applied when switching service, the general penalty term $\tau_{i,\sigma}^{pen}$ can be written as

$$\tau_{i,\sigma}^{pen} = \begin{cases} \Delta\hat{w}_\sigma/\hat{n}_\sigma & \text{if } i \neq \sigma, \\ 0 & \text{if } i = \sigma. \end{cases} \quad (8)$$

Using (7) and (8), the priority index π_i associated with traffic flow i can therefore be expressed as

$$\pi_i = \frac{\hat{n}_i}{\tau_{i,\sigma}^{pen} + \tau_i + \hat{g}_i}. \quad (9)$$

In (9), π_i is related to the number of vehicles \hat{n}_i that are expected to proceed through the intersection during a period of $\tau_i + \hat{g}_i$ seconds. The approach associated with the highest priority index value is awarded green time. Local optimisation at each intersection does not necessarily lead to global optimisation of the entire system, as instability in the form of growing queue lengths may occur. In order to combat this instability, a stabilisation strategy is also applied in the algorithm of Lämmer and Helbing.

A traffic control strategy is considered stable if vehicle queues remain bounded at all times. Instability in traffic control occurs when signals switch either too frequently, or not frequently enough, resulting in growing vehicle queues. According to Lämmer and Helbing, if traffic conditions arise such that vehicle queues remain bounded for a fixed-time control, but not for a local optimisation control strategy, then the fixed-time control is potentially superior to the locally optimised control, and the latter is considered unstable.

Maintaining stability in a system is less like a scheduling problem and more a problem of allocating the appropriate amount of green time to minimise total vehicle delay [11]. In order to maintain stability during local optimisation at intersections, a supervisory mechanism was therefore introduced by Lämmer and Helbing. The purpose of this mechanism is to assess the current traffic conditions and ensure that green times are neither too short nor too long in order to avoid growing queue lengths. The following stabilisation rule was suggested by Lämmer and Helbing: Let Ω be the set of traffic flows chosen by the supervisory mechanism that require imminent service so as to prevent instability. It is assumed that the traffic flow along approach road i joins the set Ω once the capacity of approach road i has reached a critical number of vehicles, denoted by n_i^{crit} . The parameter n_i^{crit} is specified such that the following two safety requirements are met:

1. Each traffic flow must be served once, on average, within a desired interval of length Z , and
2. each traffic flow must be served at least once within a maximum service interval of length Z_{max} , such that $Z_{max} \geq Z$.

In the above requirements, Z is the cycle time of an associated stable, fixed-time control programme and Z_{max} is the maximum red time that a traffic flow can experience. The anticipated service interval $z_i(t)$ of traffic flow i consists of the setup time, the preceding red time and the anticipated green time, *i.e.* $z_i(t) = \tau_i^0 + r_i + \hat{g}_i$. Assuming that the average arrival rate of vehicles, cycle lengths and green times of a stable, fixed-time control programme are given, a threshold function for n_i^{crit} can be derived as

$$n_i^{crit}(z_i(t)) = Q_i^{arr}(t)Z \frac{Z_{max} - z_i(t)}{Z_{max} - Z}. \quad (10)$$

This function meets the first requirement above, since when $z_i(t) = Z$ (the anticipated service interval has the same length as the desired service interval), the number of vehicles expected to arrive is $Q_i^{arr}(t)Z$. The second requirement is also met, because when $z_i(t) \geq Z_{max}$ (the anticipated service interval is at least as large as the maximum red time), $n_i^{crit}(z_i(t)) \leq 0$. Once an approach has been classified as critical (*i.e.* if $n_i(t) \geq n_i^{crit}(z_i(t))$), the approach is added to the set Ω . The first element in Ω is the first to receive service and continues to receive service until the queue has dissipated, or until the traffic flow receives a green time duration that would be awarded by a fixed-time control programme, which is equivalent to $\frac{Q_i^{arr}(t)}{Q_i^{max}}Z$. While Ω is nonempty, the stabilisation strategy is implemented and the necessary flow receives green time. When Ω is empty, however, the optimisation strategy described above is followed according to the priority indices π_1 and π_2 .

In summary, the index of the road section receiving service according to the two strategies described above is given by

$$\sigma = \begin{cases} \text{head of } \Omega, & \text{if } \Omega \neq \emptyset, \\ \max_i \pi_i, & \text{otherwise.} \end{cases} \quad (11)$$

When $\Omega = \emptyset$, the optimisation strategy is implemented, which aims to minimise vehicle delay by serving the approaching vehicles as fast as possible. When, on the other hand, $\Omega \neq \emptyset$, the stabilisation strategy takes over in order to avoid the situation where queues grow too long and exceed the threshold n_i^{crit} . Therefore, the optimisation strategy has full control of the system, unless it becomes unstable, in which case the stabilisation strategy is implemented in order to regain stability and allow the optimisation strategy once again to commence.

2.3 The O-TSCA of Einhorn

Whereas the functioning of the algorithm of Lämmer and Helbing [11] was based on entire approach roads incident with intersections, the working of the O-TSCA of Einhorn [3] is based on a disaggregation of the traffic flows along each approach road into its individual lanes. Each such approach lane is, furthermore, associated with a unique exit lane, which is the lane a vehicle in the corresponding entry lane would enter upon passing through the intersection.

Let α_i be the length of approach lane i along a road incident with an intersection and let $\hat{\ell}_j$ denote the effective length of vehicle j travelling along that lane, which is determined by summing the physical length of the vehicle and a safety gap that is always maintained between stationary vehicles. The demand $\delta_i(t)$ associated with approach lane i measures the amount of space occupied along the lane and is calculated by determining the number of approaching vehicles along lane i and summing over their effective lengths, *i.e.* $\delta_i(t) = \sum_{j \in C_i(t)} \hat{\ell}_j$, where $C_i(t)$ is the set of all vehicles along approach lane i at time t . The availability $\omega_i(t)$ associated with approach lane i represents the effective space along the corresponding exit lane i' that is available for vehicles to occupy, *i.e.* $\omega_i(t) = \alpha_{i'} - \delta_{i'}(t)$. The pressure $\pi_i(t)$ exerted by approach lane i at the intersection at time t is expressed as the sum of demand and availability associated with lane i at time t . In other words,

$$\tilde{\pi}_i(t) = \delta_i(t) + \omega_i(t). \quad (12)$$

This pressure is used in the O-TSCA to determine which intersection approach should receive service. The O-TSCA also calculates a throughput for each lane, which is the sum of the effective lengths of the vehicles that have travelled through the intersection. The throughput, denoted by $\theta_i(t)$, provides an indication of the total space occupied by these vehicles.

In order to determine the pressure of a phase m over all the approach lanes that receive service during the phase, the pressures $\tilde{\pi}_i(t)$ are summed over all the approach lanes i receiving service during phase m . The total pressure for phase m is therefore given by $\Pi_m(t) = \sum_{i \in U_m} \tilde{\pi}_i(t)$, where U_m is the set of all approach lanes receiving service during phase m . The total pressure for each phase is calculated and the phase with the largest

pressure is awarded service. If phase m is selected for service at time t^* , the following two variables are stored by the O-TSCA: First, the *demand*, given by $\Delta_m = \sum_{i \in U_m} \delta_i(t^*)$, and secondly, the *availability*, given by $\Omega_m = \sum_{i \in U_m} \omega_i(t^*)$. Phase m continues to receive service until

$$\sum_{i \in U_m} \theta_i(t) \geq \Delta_m, \quad (13)$$

or

$$\sum_{i \in U_m} \theta_i(t) \geq \Omega_m. \quad (14)$$

Inequality (13) ensures that phase m receives service until the total throughput of lanes served during phase m is equal to or larger than the total demand of all lanes served during phase m . In other words, it ensures that all vehicles initially requiring service, are served. Inequality (14) requires that phase m receives service until the total throughput of the lanes served during phase m is equal to or larger than the total availability of the lanes served during phase m . In other words, this constraint ensures that the space required by vehicles receiving service along approach lane i does not exceed the amount of available space along the corresponding exit lane i' .

Once condition (13) or condition (14) holds, the O-TSCA recalculates the pressures for each phase and once again awards service to the phase with the largest pressure, which may or may not be the same phase currently receiving service. If, however, service is awarded to a different phase, the throughput $\theta_i(t)$ is set to zero for all $i \in U_m$, where m denotes the phase currently receiving service.

It was noted during the implementation of the O-TSCA that no indication was given in [3] as to the length of approach road considered during the calculation of phase pressures. Einhorn [3] was therefore contacted and it was confirmed that the approach road lengths from the intersection considered in [3] was 275 metres.

2.4 The I-TSCA of Einhorn

The I-TSCA makes use of three sets $C_i(t)$, $Q_i(t)$ and $S_i(t)$. First, $C_i(t)$ is the set of all vehicles along approach lane i at time t , while $Q_i(t)$ is the set of all queued vehicles at time t together with those vehicles that will become queued either behind an existing queue or behind a stop line along approach lane i during a red signal. Finally, $S_i(t)$ is the set of all stationary, queued vehicles along approach lane i at time t . Therefore, $S_i(t) \subseteq Q_i(t) \subseteq C_i(t)$. The position of vehicle j in a queue along approach lane i at time t is denoted by $\epsilon_{ij}(t)$; this value indicates how far the vehicle is from the intersection ahead. A queue position function $\rho_i(t)$ is employed which provides an indication of how far a vehicle queue reaches upstream along approach lane i at time t . The associated vehicle stopping point in the queue of vehicle j at time t is denoted by $\mu_j = \rho_i(t)$, as depicted in Figure 1. Using these values, the distance between the position of vehicle j at time t and its future stopping point $\rho_i(t)$ is denoted by $d_{j,\rho_i(t)}$. Once a green signal is displayed at an intersection, queued vehicles are assumed to depart from their positions in approach lane i at a constant rate of η_i vehicles per second, where η_i is the maximum flow rate of Q_i^{max} vehicles per second.

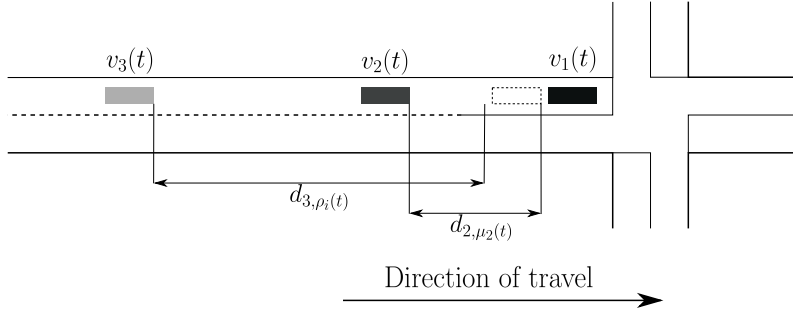


Figure 1: A visual representation of the various variables in the I-TSCA of Einhorn [3]. Assuming the traffic signal ahead of the vehicles is not green and $d_{3,\mu_3(t)}/v_3(t) < \sum_{p \in B \setminus \{m\}} (\chi_p(t) + \mathcal{T}_p(t)) + Q_i(t)/\eta_i$, it follows that $C_i(t) = \{v_1(t), v_2(t), v_3(t)\}$, $Q_i(t) = \{v_1(t), v_2(t), v_3(t)\}$ and $S_i(t) = \{v_1(t)\}$.

Each signal cycle is assumed to consist of a set of B green phases as well as the associated B setup times. The green time allocated during a phase $m \in B$ is denoted by χ_m^* , while $\chi_m(t)$ represents the remaining green time of the phase, where $0 \leq \chi_m(t) \leq \chi_m^*$. Similarly, the remaining amount of the total setup time \mathcal{T}_m^* associated with phase m remaining at time t is denoted by $\mathcal{T}_m(t)$, where $0 \leq \mathcal{T}_m(t) \leq \mathcal{T}_m^*$.

Using these variables, it is possible to determine whether an approaching vehicle will become queued, and therefore delayed, or travel through the intersection without stopping. If vehicle j is travelling at a speed v_j along approach lane i , the time taken for the vehicle to reach the intersection can be determined so as to predict whether it will become queued or not, and if so, the position and stopping point of the vehicle in the queue may be estimated. If the signal is red, this is achieved by calculating the remaining red time, together with the remaining green time required to clear the queue. If, however, the signal is green, then only the green time taken to clear the queue is required. If the time taken for a vehicle to reach the intersection is more than the remaining green time, it will become queued and is thus inserted in $Q_i(t)$.

The above decision may be written mathematically as follows: If the current signal is not green and

$$d_{j,\rho_i(t)}/v_j < \sum_{p \in B \setminus \{m\}} (\chi_p(t) + \mathcal{T}_p(t)) + |Q_i(t)|/\eta_i,$$

then vehicle j is inserted in $Q_i(t)$. If the current signal is green, on the other hand, and $d_{j,\rho_i(t)}/v_j < |Q_i(t)|/\eta_i$, then vehicle j is again inserted in $Q_i(t)$. If the queue is empty, *i.e.* $Q_i(t) = \emptyset$, then vehicle j will become queued in the following two cases: First, if the current signal is green and $d_{j,\rho_i(t)}/v_j > \chi_m(t)$ or, secondly, if the signal is not green and

$$d_{j,\rho_i(t)}/v_j < \sum_{p \in B \setminus \{m\}} (\chi_p(t) + \mathcal{T}_p(t)).$$

The I-TSCA functions by iterating through three main steps. The first of these steps involves calculation of the required green time $\gamma_i(t)$ for approach lane i to clear all the queued vehicles and vehicles that will become queued before the lane receives a green

signal, *i.e.* all vehicles in the set $Q_i(t)$. If $Q_i(t) = \emptyset$, then $\gamma_i(t)$ is the time taken for the first vehicle to pass through the intersection. The required green time is therefore given by

$$\gamma_i(t) = \begin{cases} d_{1,\rho_i}(t)/v_1(t), & \text{if } Q_i(t) = \emptyset \text{ and } C_i(t) \neq \emptyset, \\ |Q_i(t)|/\eta_i, & \text{if } Q_i(t) \neq \emptyset, \\ \infty, & \text{if } C_i(t) = \emptyset. \end{cases} \quad (15)$$

The first condition in (15) is the case in which there are no queued vehicles, but there is at least one vehicle approaching the intersection along approach lane i , and the required green time is the time it will take for that first vehicle to pass through the intersection. The purpose of this condition is to ensure that intersection usage is maximised. The second condition in (15) is the case in which there is at least one queued vehicle, and so the time required to clear the queue is the number of vehicles in the queue, divided by the number of vehicles that depart from the queue per second. The minimum required green time over all the lanes of phase m is chosen, *i.e.* $\Gamma_m(t) = \min_{i \in U_m} \gamma_i(t)$. The purpose of this condition is to ensure that if one approach lane has a very large required green time, it does not prevent other lanes requiring shorter green times from receiving service. The rationale behind the third condition in (15) is that if there are no approaching vehicles along an approach lane, then the lane must not receive service.

The second step of the I-TSCA involves calculation of vehicle delays associated with awarding service to a certain phase. Since a vehicle experiences a delay if it becomes queued, it is necessary to predict the length of time over which the vehicle will be queued. Because this value depends on whether approach lane i is receiving service or not, a binary parameter $\kappa_i(t)$ is introduced, where $\kappa_i(t) = 0$ represents no service, while $\kappa_i(t) = 1$ indicates that approach lane i is receiving service. Therefore, the delay time experienced by a vehicle j along approach lane i during phase m is

$$\phi_{ij}^m(t) = \begin{cases} \mathcal{T}_m(t) + \frac{\epsilon_{ij}(t)}{\eta_i} - \frac{d_{j,\mu_j}(t)}{v_j(t)}, & \text{if } \kappa_i(t) = 1, \\ \sum_{p \in B} \mathcal{T}_p(t) + \sum_{p \in B \setminus \{m\}} \chi_p(t) - \frac{d_{j,\mu_j}(t)}{v_j(t)} + \frac{\epsilon_{ij}(t)}{\eta_i}, & \text{if } \kappa_i(t) = 0. \end{cases} \quad (16)$$

If vehicle j is not queued along approach lane i , the vehicle will not experience a delay and so $\phi_{ij}^m(t) = 0$. The first condition in (16) represents the various times associated with receiving service. The sum of the first two terms represents the time taken for the intersection to be cleared, while the last term represents the time taken for the vehicle to reach the intersection. The second condition in (16) contains four terms, the first being the sum of the remaining setup times of the various phases. The second term represents the remaining green time of the phases preceding the green signal. The third and fourth terms are the same as described in the first line of (16).

Let I be the set of all the approach lanes to the intersection. The total cost $\Phi_m(t)$ of assigning phase m service is determined by summing the delay terms incurred by all vehicles across the approach lanes in I . Therefore, $\Phi_m(t) = \sum_{i \in I} \sum_{j \in C_i(t)} \phi_{ij}^m(t)$. The different ‘‘inventory control’’ costs associated with the delay time are as follows: A setup cost is calculated by summing the setup time $\mathcal{T}_m(t)$ for all delayed vehicles. A holding cost is calculated by summing all delay times incurred by vehicles that are not currently receiving service, and a stock-out cost is the sum of the delay times experienced by vehicles whose service is terminated before they can travel through the intersection.

The third and final step of the I-TSCA involves assigning a service time of length $\Gamma_m(t)$ to a specific phase m such that $\Phi_m(t)$ is a minimum. Once service has been assigned, the I-TSCA continues to recalculate the required green time and total cost, and assigns green time once the current service time $\Gamma_m(t)$ has elapsed. It is possible that after the service time of length $\Gamma_m(t)$ elapses at time t' , phase m is once again selected for service, in which case service is not terminated, but is extended for another $\Gamma_m(t')$ time units.

2.5 The Hybrid algorithm of Einhorn

The Hybrid algorithm of Einhorn [3] functions by utilising the contrasting behaviours of the O-TSCA and I-TSCA. The I-TSCA assigns specific green times in advance, while the O-TSCA assigns an infinite green time and switches signals once certain conditions are met. As a result, the I-TSCA and the O-TSCA are implemented slightly differently in the Hybrid algorithm. These algorithms are executed concurrently with an *Intersection Utilisation Maximisation Supervisory Mechanism* (IUMSM) based on vehicle proximity to the intersection. If there is at least one vehicle within the vehicle's safe following distance of the intersection approach currently receiving service, a proximity variable Ξ is set to a value of 1, otherwise it is assigned a value of 0.

Once a phase has been selected for service, both the I-TSCA and the O-TSCA are executed concurrently, while the amount of green time is decided by the I-TSCA. Since the I-TSCA is associated with faster signal changing, Hybrid waits for the I-TSCA to request a signal change. Once the I-TSCA has requested the signal change (*i.e.* an extended green time had been calculated and was subsequently rejected by the algorithm), the I-TSCA allows the extended green time (which was rejected) to begin, until either the IUMSM or the O-TSCA also requests a signal change, at which point in time the signals change.

3 The simulation model test bed

This section contains a discussion on various aspects of the newly designed microscopic traffic simulation model employed in this paper as an algorithmic test bed. The discussion includes general specifications of the gridded road network considered as well as details of the output generated by the model. The statistical method of comparison performed in respect of the output data is also described, and the methodology followed to validate the model is discussed.

3.1 General specifications of the modelling framework

The road network topology for which traffic flow simulation replications are reported in this paper is a 3×4 grid of equally spaced intersections created in Anylogic 7.3.5 and making use of the Anylogic Road Traffic Library included in the software. The model was validated by making use of real vehicle counts and signal timings recorded at a signalised intersection, as described by Van der Merwe [19], ensuring that the output of the model generated similar results to those reported in [19]. A section of the road network topology is shown in Figure 2. Vehicle arrival rates are classified as conforming to either light

traffic conditions or heavy traffic conditions. Light traffic conditions refer to vehicle arrivals according to a Poisson distribution at a mean rate of 10 vehicles minute for each road entry into the network. Since there are 14 points of entry into the network, this is equivalent to a mean total arrival rate of 140 vehicles per minute into the road network. Heavy traffic conditions similarly involve mean arrival rates of 20 vehicles per minute, which is equivalent to a mean total arrival rate of 280 vehicles into the road network per minute.

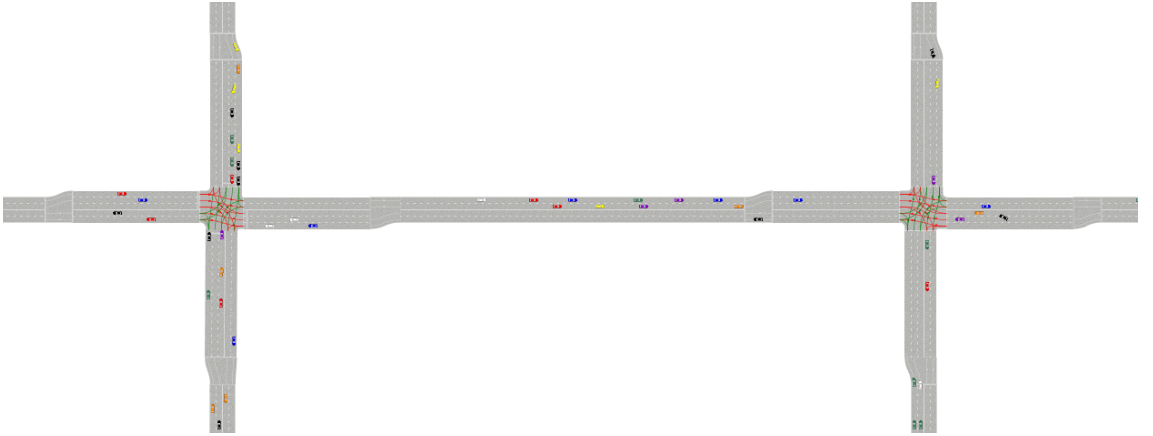


Figure 2: A screen-shot of two neighbouring signalised intersections in a gridded transportation network from the graphical user interface of the simulation model. Each intersection approach consists of four 95 metre three-lane approaches and four two-lane exit roads.

Vehicles that travel through the road network are permitted to make at most one turn throughout their respective routes. Vehicles that enter the network from a westerly or an easterly direction have a total of nine possible destinations if the network is orientated as shown in Figure 3(a). These include four possible left turns, four possible right turns or else travelling straight through the network. If a vehicle is generated to enter from a northerly or southerly direction, it has seven possible destinations, including the three possible left turns, the three possible right turns, or travelling straight through the network without making any turns. An example of this logic is shown in Figure 3(b). All vehicles perform left or right turns at each intersection with a probability of 5%, while the majority of vehicles do not perform turns.

All vehicles in the model have a length of five metres, and distances between consecutive stationary vehicles are determined by a uniform distribution between one and three metres (Paramics [14] makes use of a default value of two metres). Acceleration and deceleration rates are taken as the values recommended by Anylogic [1], namely 1.8 m/s and 4.2 m/s, respectively. These values are in line with what other simulation software packages suggest for default values. SUMO [18], for example, suggests a default deceleration value of 4.5 m/s, while Paramics [14] suggests an acceleration value of 2.0 m/s and a deceleration value of 3.5 m/s. Vehicle following distances are based on the individual vehicle speeds and deceleration rates, allowing a large enough gap to ensure that if the front vehicle begins decelerating, the following vehicle has enough time to react without colliding with the vehicle ahead. A vehicle changes lanes if a slower vehicle is detected ahead of it, or if

the vehicle has an upcoming turn and is not in the correct lane, while preferred vehicle speeds are determined according to a uniform distribution within the range 47–72 km/h.

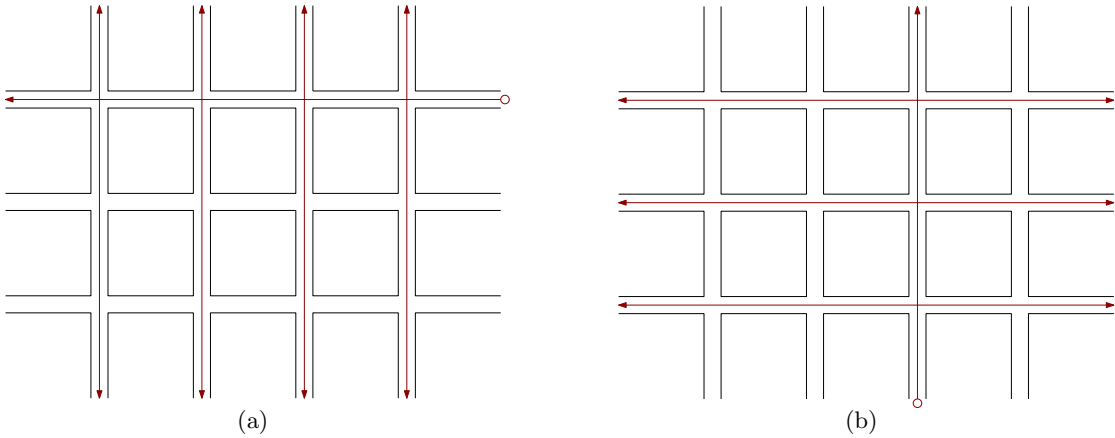


Figure 3: The possible destinations of a vehicle entering the network at the circle from the east are shown in (a), while the possible destinations of a vehicle entering the network from the south are shown in (b). Each vehicle may make at most one turn throughout its entire route through the network.

3.2 Model output

The algorithms of §2 (with the exception of Fixed) all assume the presence of radar vehicle detection equipment mounted at each intersection (we assume the *SmartSensor Advance Extended Range* radar detection unit [20] as the mode of detection used in the simulation model) which may detect the speed at which vehicles are approaching an intersection as well as their distances from the intersection from up to 275 metres away, in each direction. These data are used in signal switching decisions and in calculating certain output data emanating from the simulation model. Six PMI values are recorded for each algorithm (after a simulation warm-up period equivalent to 1 800 seconds has elapsed) and these are used to compare the relative performance of the algorithms.

The first two PMIs are the mean delay time and the normalised mean delay time experienced by vehicles in the road network. Since the route each vehicle takes is known *a priori*, the delay time may be calculated by subtracting the ideal travel time of the vehicle (the route distance divided by the preferred speed of the vehicle) from the actual time spent by the vehicle in the network. The normalised mean delay time is calculated by dividing the mean delay time by the distance travelled by the vehicle in order to take into account the distance travelled by the vehicle, since a vehicle that travels farther is more likely to experience a larger delay than a vehicle travelling over a short distance. If, for example, a vehicle achieves a normalised mean delay value of 2.0, this indicates that the vehicle has spent twice as long in the network than it would have, had it travelled unimpeded at its preferred speed the entire way.

The next two PMIs are the mean number of stops and the normalised mean number of stops experienced by vehicles in the road network. The normalised mean number of stops is

calculated by dividing the mean number of stops of a vehicle by the number of intersections encountered by the vehicle during its entire route, since a vehicle travelling through more intersections is more likely to experience a larger number of stops. A normalised mean number of stops value of 0.5 indicates, for example, that a vehicle is expected to have stopped at 50% of the intersections it encountered.

The final two PMIs are the mean time vehicles spend travelling slowly (under 10 km/h) and the normalised mean time vehicles spend travelling slowly (under 10 km/h). The latter PMI is calculated by dividing the mean time a vehicle spends travelling under 10km/h by the total time the vehicle spends in the system, in order to capture the average portion of time a vehicle spends travelling unacceptably slowly.

3.3 Statistical comparison methodology

Once all the PMIs have been recorded over the course of 30 simulation replications, an ANOVA test [8] is carried out in respect of the mean PMI values in order to test whether a statistical difference exists between at least two of the algorithms. If a statistical difference is detected, a Levene test [17] is performed to test whether the variances between the PMI values of the algorithms are significantly different at a 95% level of confidence. If this is not the case, Fisher's LSD *post-hoc* test [22] (which assumes homogeneity of variances) is carried out in order to isolate the statistical differences. If, on the other hand, the variances are significantly different, a Games-Howell *post-hoc* test [9] (which does not require that the variances be approximately equal) is carried out for this purpose.

A similar comparison experiment was carried out by Einhorn [3], although different statistical tests and a different simulation model was used. Furthermore, additional PMIs are considered in the comparison of this paper. Initially, the objective was to corroborate the results obtained by Einhorn with a view to verify both models, but once the algorithms had been implemented and run in the simulation model, areas of improvement became apparent in some of the algorithms and these improvements were subsequently implemented. While the actual values obtained for the PMIs cannot be compared directly to those reported by Einhorn due to the different simulation models in which the algorithms were tested, the relative performances of the algorithms are compared to those reported by Einhorn. It is noted, however, that both the results reported here and those reported by Einhorn were obtained based on simulation replications in the context of a 3×4 grid of equally spaced intersections, with identical mean arrival rates from all directions, similar vehicle speeds and turning probabilities. Due to these similarities, the results in the simulation experiments were expected to be rather similar to those reported by Einhorn.

3.4 Simulation model validation

A popular method of simulation model validation entails comparing the simulated output of the model to the output of a real system. This validation approach was adopted, using real data collected for a previous study by Van der Merwe [19] at an isolated traffic intersection in Stellenbosch, a university town in the South African Western Cape. An aerial view of the relevant intersection is shown in Figure 4. The intersection has four approaches, and consists of twelve approach lanes and six exit lanes. The legal manoeuvres

for each lane are as follows: Left-most lanes permit both turning and travelling straight for the horizontal direction (AT and R44), yet only left-turning is permitted for the left lane in the vertical directions (Bird and N1). The centre lane only permits travelling straight for all approaches, while the right-most lane only permits right-turning for all approaches. The first green phase allocates all vehicles travelling from the R44 direction a green signal, and the second green phase assigns all the vehicles travelling along the horizontal direction a green signal (R44 and AT), while right-turning occurs on a permissive basis. The third green phase provides the entire vertical direction with green time, while right-turning vehicles do so on a permissive basis once again. The fourth and final green phase allows protected right-turn phases for vehicles travelling from Bird and N1, while left-turning vehicles from AT and R44 are also permitted to complete their turns.

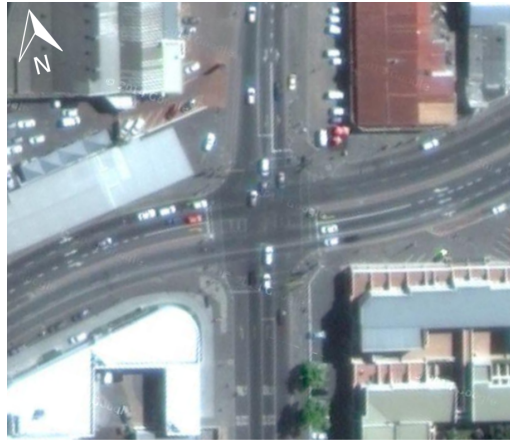


Figure 4: An isolated intersection in Stellenbosch where Adam Tas and Bird Streets intersect. The road approaching from the left is referred to as Adam Tas (AT), the road approaching from the right is referred to as R44, the road approaching from the bottom is referred to as Bird, and the road approaching from the top is referred to as N1.

The number of vehicles passing through the intersection and their associated manoeuvres were recorded by Van der Merwe [19] in fifteen-minute intervals from 06:30 to 18:00 on a Tuesday during school and university term time in order to capture the standard traffic conditions at the signalised intersection. For the purpose of validating the model described in §3.1, the vehicles passing through the intersection were aggregated into eleven one-hour periods and one half-hour period (as were their associated manoeuvre probabilities). Similarly, the lengths of the green time phases recorded by Van der Merwe were aggregated into morning, midday and afternoon green times.

The average hourly arrival rates, as well as turning probabilities for each approach, were taken as input data for the simulation model. Each simulation replication was executed for the equivalent of eleven and a half simulation hours, recording output data for each hour, and 30 of these replications were carried out. The average output results of the 30 replications were compared against the actual known values and the absolute errors were recorded. The total simulated number of vehicles passing through the intersection was found to deviate by no more than 2.6% from the actual value (see Table 1), indicating that

the model calibrated as described in §3.1 accurately represents the real-world system to which it was compared.

Total percentage error after each hour											
1	2	3	4	5	6	7	8	9	10	11	11.5
2.1	2.6	2.2	2.0	1.9	1.8	1.8	1.7	1.7	1.7	2.0	2.2

Table 1: *The mean percentage error after each simulation hour.*

4 Simulation results for the existing algorithms

The simulation results obtained when testing the algorithms described in §2 with the traffic simulation model test bed of §3 are reported in this section for both light and heavy traffic conditions. The results are presented by means of box plots and accompanying interpretations.

4.1 Simulation results under light traffic conditions

For each of the PMIs, a similar trend emerged with respect to the order of the comparative performance of the algorithms at a 95% level of confidence under light traffic conditions, which is clear from box plots of these performances shown in Figure 5. Gersh was found to perform the best overall, achieving the most favourable outcome for each of the six PMIs. Fixed and Hybrid performed the second most effectively overall, followed by the LH algorithm. The O-TSCA was the second worst performing algorithm in respect of each PMI, followed by I-TSCA which was consistently the worst performing algorithm in terms of each of the PMIs.

The very short green times allocated by the I-TSCA (as short as 3.33 seconds) is the principal cause of the poor performance of the algorithm, often only allowing the front vehicle row to make it through an intersection before a signal change is initiated. The O-TSCA, on the other hand, performed poorly for a contrasting reason: Allocating excessively long green times. It was clear, upon observing the working of the O-TSCA in the simulation model, that the signals switched too infrequently, often causing vehicles to wait unnecessarily long at intersections. Hybrid was able to improve upon both the I-TSCA and the O-TSCA, and this improvement is attributed to the IUMSM, maximising intersection utilisation by switching signals according to the more suitable algorithm at the current time.

A number of differences were found between the results reported in Figure 5 and those reported by Einhorn [3]. One of the most notable differences is the finding in terms of the best performing algorithm in respect of the mean number of stops. Einhorn reported the O-TSCA to have achieved the smallest number of stops under light traffic conditions, followed by Hybrid, whereas the authors found that the O-TSCA was relatively ineffective at preventing vehicle stops. In terms of mean delay time, it was further reported by Einhorn that Hybrid was the most effective algorithm, followed by Gersh, while LH was the worst performing algorithm. In contrast, we found that Gersh is the best performing algorithm under light traffic conditions, followed by LH. It was also reported by Einhorn

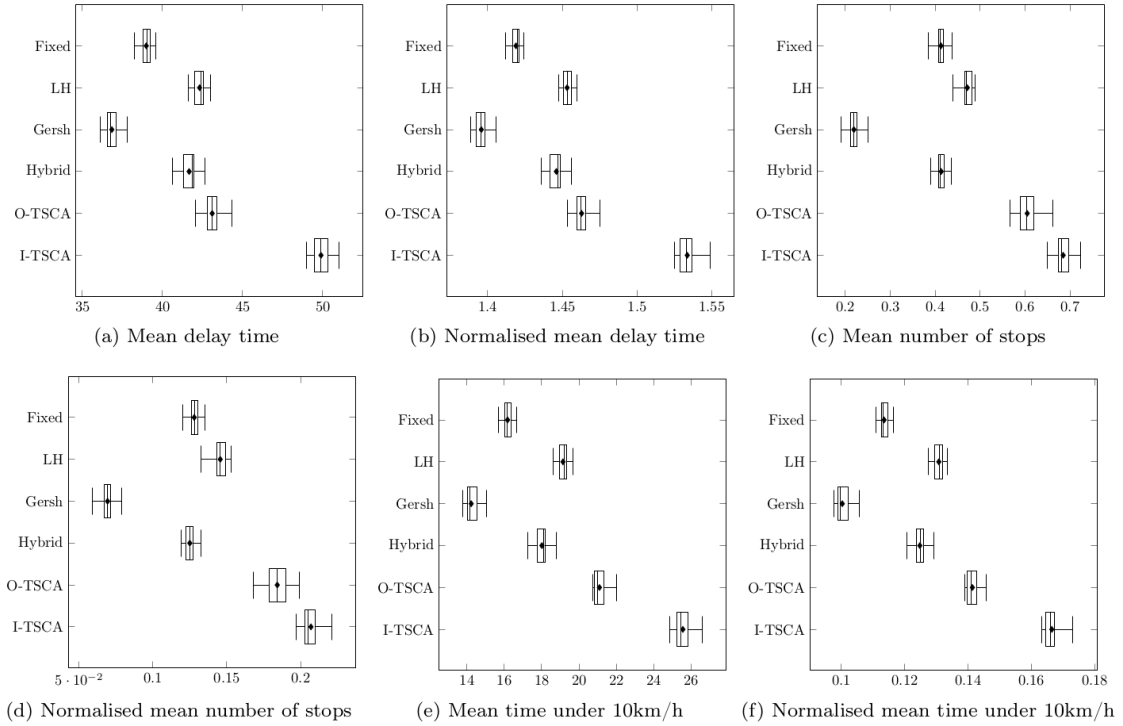


Figure 5: PMI results for the self-organising algorithms of §2 under light traffic conditions.

that Hybrid was the best performing algorithm in terms of normalised mean delay time, significantly outperforming the second best algorithm, Gersh. The opposite is visible in Figure 5. Finally, LH was the worst performing algorithm overall according to Einhorn, again contradicting the results of Figure 5, which indicate that the I-TSCA performed the worst overall under light traffic conditions.

4.2 Simulation results under heavy traffic conditions

Interestingly, it was found that under heavy traffic conditions the performance of Gersh worsened dramatically in comparison with those of the other five algorithms, which all worsened by a similar margin. The variance in the PMIs associated with Gersh were also significantly larger than it was under light traffic conditions, indicating that there are inconsistencies in the performance of this algorithm under heavy traffic conditions.

In terms of the mean number of stops and the normalised mean number of stops, Gersh went from being the best performing algorithm under light traffic conditions, to being the overall worst under heavy traffic conditions. LH was the superior algorithm in terms of both of these PMIs under heavy traffic conditions, achieving a mean number of stops value of 1.637 (see Figure 6(c)) and a normalised mean number of stops value of 0.516 (see Figure 6(d)), thus outperforming Hybrid significantly (achieving corresponding values of 1.936 and 0.599, respectively). Fixed and LH were the best performing algorithms over all the PMIs. The O-TSCA and Hybrid only differed significantly from each other with respect to mean delay time and normalised mean delay time at a statistical significance of

5%, while they performed significantly worse than LH and better than Gersh in terms of all PMIs at this level of significance.

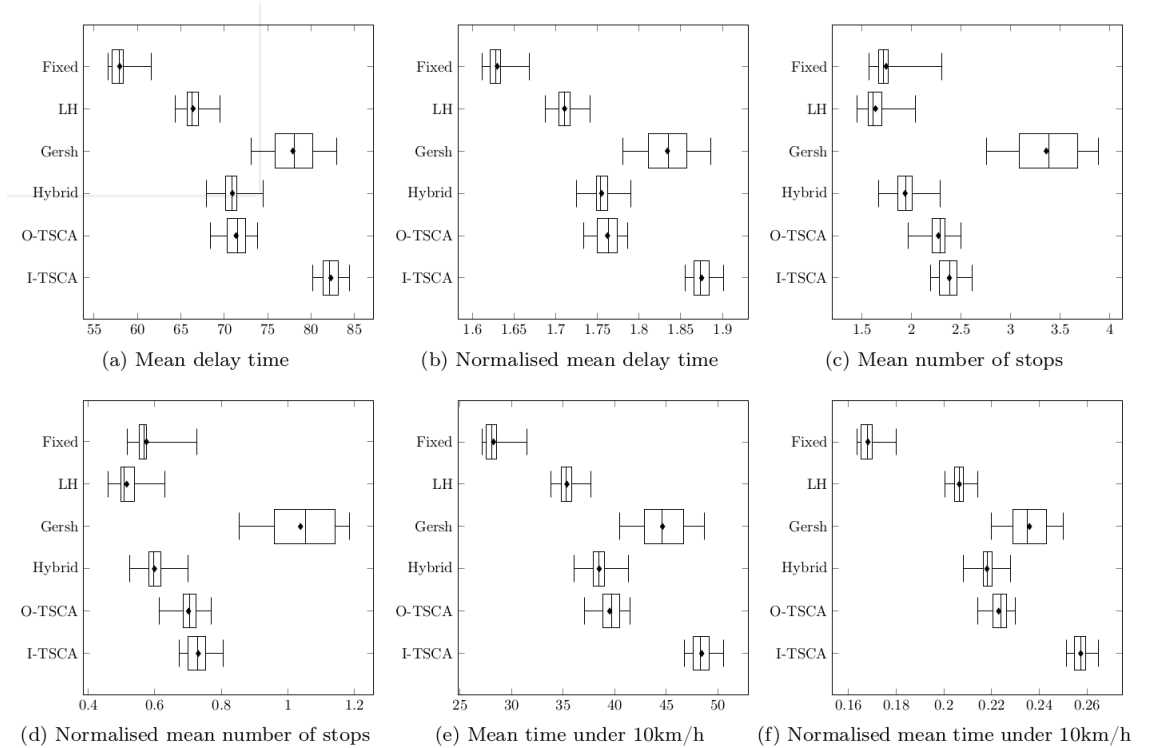


Figure 6: PMI results for the self-organising algorithms of §2 under heavy traffic conditions.

When the aforementioned results are compared to those reported by Einhorn [3] for heavy traffic conditions, contradictions were once again found. In terms of mean delay time and normalised mean delay time, Einhorn found that the O-TSCA was the best performing algorithm, followed by Gersh in both cases. This stands in contrast to the findings reported in Figure 6 in which LH is the best performing algorithm and Gersh is the second worst performing algorithm. Einhorn also reported that the O-TSCA was once again the most efficient at preventing vehicle stops, followed by Hybrid. This is contrary to what is reported in Figure 6.

4.3 Simulation result differences

There are a number of possible causes for the surprising differences between the results in Figures 5–6 and those reported by Einhorn [3], the obvious reason being the use of different simulation models. The behaviours of the algorithms described by Einhorn were in some cases different to what was observed in our simulations. For instance, it was reported by Einhorn that the I-TSCA switched signals frequently (which was the purported reason for its success under light traffic conditions). While it does the same in the authors’ simulation model, signals are switched so frequently that often only one vehicle makes it through an intersection at a time. It is speculated that perhaps the simulation model of Einhorn contained vehicles with faster acceleration rates, allowing more vehicles through

intersections per unit of time. Supporting evidence for this speculation is offered by the results returned by the fixed control scheme employed by Einhorn. In his fixed control scheme, a value of two seconds was implemented under light traffic flows for each green phase, implying that two seconds would be sufficient to allow at least one vehicle to make it through the intersection in the simulation model of Einhorn, while a minimum of 3.33 seconds is necessary for one initially stationary vehicle to make it through the intersection in our carefully calibrated and validated model. In our simulation model, acceleration and deceleration rates were taken as the realistic values recommended within the Anylogic simulation software environment [1], as described in §3.1 and corroborated in [14, 18]. While the acceleration and deceleration rates implemented in the model of Einhorn are not known exactly, it is known for certain that those acceleration rates were substantially larger than those employed in our model.

5 Suggested improvements of the algorithms by Einhorn

Algorithmic shortcomings of three of the algorithms of §2 became clear after observing numerous simulation replications during the performance comparison experiment described in the previous section. These shortcomings may be rectified as described in this section.

5.1 The improved O-TSCA

While it is stated in [3] that the O-TSCA is free of parameters, it does, in fact, indirectly make use of a parameter. The algorithm operates according to the detected demand and availability along approach and exit lanes, respectively. The effective “sight” of the algorithm extends 275 metres down each roadway connected to the intersection, as this is the maximum distance over which the assumed mode of detection functions (as described in §3.2). Although the O-TSCA makes use of this maximum distance, it is not necessarily the most suitable distance. It is therefore recommended that this parameter be adjusted according to the road network and current traffic conditions.

One of the characteristics of the O-TSCA is that it does not change signals until every initially detected vehicle along the approach receiving service has successfully travelled through the intersection. While this may be seen as an advantage of the algorithm, it becomes problematic under light traffic conditions when there may be large distances between consecutive approaching vehicles in which case this behaviour may lead to an ineffective use of green time.

There may also be room for improvement of the algorithm in terms of not switching signals if there is a vehicle in very close proximity of the intersection. Since the O-TSCA considers changing signals once the initially detected demand has passed through the intersection, it may happen, as signals are changed, that a vehicle from an approach receiving service that is very close to the intersection is forced to stop. The delay time of this vehicle will be the sum of at least two setup times and the green time received by opposing directions. If, on the other hand, the service were to be extended by a few seconds, the vehicle would

not experience additional delay and the other vehicles will each only experience an extra few seconds' delay time.

The original O-TSCA was altered to accommodate the two problematic situations described above. The demand and availability detection length was taken to be 110 metres (the length of the three-lane approach together with the length of the lane merge), while the proximity of vehicles to the intersection that prevent a signal change was varied from 0 to 20 metres. A distance of 10 metres was ultimately chosen for this parameter as it yielded the best results for both light and heavy traffic conditions.

5.2 The improved I-TSCA

In contrast to the O-TSCA, the disadvantage of the I-TSCA lies in its switching signals too often, rather than too infrequently. It was found that the majority of the time the initially required green time (before green time extensions) was 3.33 seconds, as this is the time required for a stationary vehicle at the intersection to travel through it. Once this time has elapsed, the I-TSCA calculates a potential extended green time based on the distance between the closest vehicle and the intersection. If the extension is granted, the length of this extension is the minimum time necessary for a vehicle to cross the intersection. This vehicle is often very close to the intersection and does not, in fact, require an extension because if it were travelling at speed, it would make it through the intersection during the amber phase. This is an example of an unnecessary extension. There are, however, also cases where an extension would be preferable, yet is not granted. This is caused when delays are not accurately predicted, which is unavoidable.

The first change recommended for the I-TSCA is applying a minimum green time of 7 seconds to the algorithm [10] in order to prevent it from switching signals after only 3.33 seconds as it commonly does.

Another recommended alteration of the I-TSCA involves the calculation of the extended green time. The algorithm considers the shortest time required by a vehicle to cross the intersection and calculates a cost based on that time. This means that if a green-time extension is allowed, it will only be long enough to ensure that the single closest vehicle makes it through the intersection, resulting in an increment of the delay time of all vehicles queued in the opposing direction, corresponding to the green time extension that was awarded.

Rather than basing this calculation on the vehicle closest to the intersection, it is recommended that vehicles further away are considered as well. In particular, vehicles on the three-lane approach road section to the intersection that are furthest from the intersection should also be considered. In this case, required green times should be calculated based on vehicles that are up to 95 metres (the length of the three-lane approach road) away from the intersection.

An example of the I-TSCA switching signals prematurely is shown in Figure 7. The variables *H3greenActual_Inv* and *V3greenActual_Inv* indicate the most recent horizontal and vertical green times employed at the intersection. The horizontal direction only received 3.33 seconds of green time before the signal was terminated and it is clear from the figure that there are a number of other vehicles requiring service that are forced to wait for the

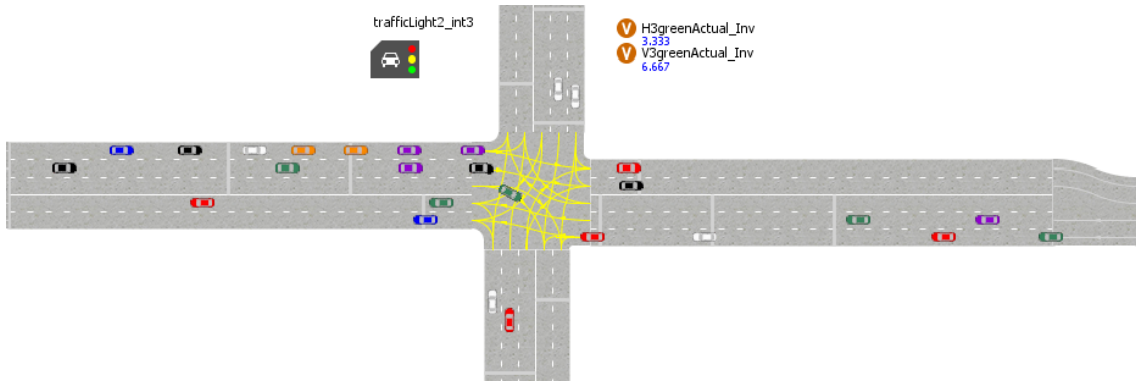


Figure 7: An example of ineffective switching of signals by the I-TSCA in the simulation model visualisation.

following cycle(s). These recommendations were incorporated into the I-TSCA in order to arrive at an alternative version of the algorithm.

5.3 The improved Hybrid algorithm

Since Hybrid is a combination of the two aforementioned algorithms, no additional changes were implemented for this algorithm, other than the indirect changes that occur as a result of the alterations made in the I-TSCA and the O-TSCA, described in §5.1 and §5.2, respectively.

In §4 it was found that Hybrid outperformed or at least matched the performance of both the I-TSCA and the O-TSCA, although its performance is reliant on the performance of these two algorithms. It is therefore expected that if both the I-TSCA and O-TSCA are improved, the Hybrid algorithm should also achieve an improved performance.

6 Simulation results for altered algorithms

The changes suggested in the previous section were implemented, and the relative performances of the original three algorithms of Einhorn [3] and their altered counterparts described in §5.1–§5.3 are compared in this section. The new algorithmic versions are distinguished from their original counterparts by adding an “(n)” to their original acronyms.

6.1 Results under light traffic conditions

A significant improvement in the performance of each of the three self-organising algorithms of Einhorn [3] is evident in Figure 8 as a result of implementing the algorithmic changes suggested in §5. The I-TSCA(n) achieved the largest improvement over all six of the PMIs, including a mean delay time reduction of 13.08 seconds (see Figure 8(a)) and more than halving the mean number of stops from 0.685 to 0.277 (see Figure 8(c)). The mean time spent by vehicles travelling under 10 km/h also improved dramatically from 25.56 seconds to 14.34 seconds, an improvement of almost 44%.

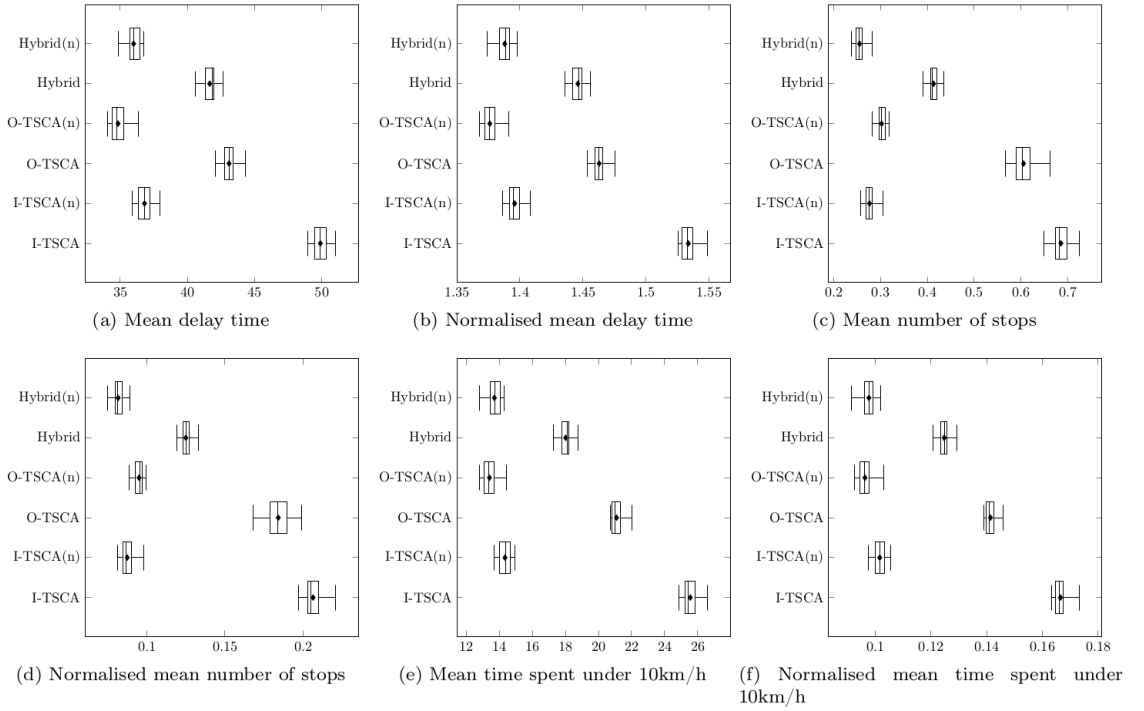


Figure 8: PMI results for the improved self-organising algorithms under light traffic conditions.

The I-TSCA(n) was found to switch signals less often than the I-TSCA (due to the minimum green time constraint), yet it still switched signals relatively frequently when necessary, thus maintaining a considerable amount of flexibility. Green time extensions were granted more frequently as a result of the newly required green time calculation in the extension policy, allowing more vehicles to pass through the intersection during a single signal phase, on average. It was also found that when a platoon of vehicles was within the three-lane approach of an intersection, the I-TSCA(n) was likely to grant a green time extension in order to facilitate their uninterrupted passage through the intersection if there was not a large number of closely approaching vehicles from the opposing direction.

The O-TSCA(n) also achieved considerable improvements with respect to all six of the PMIs. The mean delay time of the O-TSCA was improved from 43.11 seconds to 34.85 seconds (see Figure 8(a)) and its normalised delay from 1.46 to 1.38 (see Figure 8(b)). The mean number of stops achieved by the O-TSCA(n) halved and the normalised number of stops decreased by over 48% when compared to the corresponding results of the O-TSCA. The mean time spent travelling under 10 km/h improved by 7.70 seconds, from a value of 21.10 to a value of 13.40 (see Figure 8(e)).

Hybrid(n) achieved the smallest improvement comparatively, although it is still a considerable improvement. This is not surprising, as Hybrid was already the superior algorithm out of the three algorithms proposed by Einhorn before any algorithmic alterations were made. It would therefore be more difficult to improve upon the performance of this algorithm as much as on those of the I-TSCA(n) and the O-TSCA(n). Hybrid(n) achieved a mean delay of 36.01 seconds, a 5.66 second improvement over Hybrid (see Figure 8(a)).

In terms of the mean number of stops and the mean time spent travelling under 10 km/h, the results of Hybrid were improved upon by 38% and 24%, respectively.

Before any changes were made to Hybrid, it utilised the contrasting elements of the green time allocation by the I-TSCA and the O-TSCA green time allocation to outperform each of the individual algorithms. After the changes to the I-TSCA and O-TSCA were implemented, the green times of the I-TSCA(n) were longer than before, while the green times of the O-TSCA(n) were shorter than before, resulting in the situation where Hybrid(n) does not follow the same approach any more.

6.2 Results under heavy traffic conditions

As was the case under light traffic conditions, all three algorithms proposed by Einhorn [3] were improved with respect to the six PMIs under heavy traffic conditions as a result of the algorithmic changes recommended in §5, as may be seen in Figure 9.

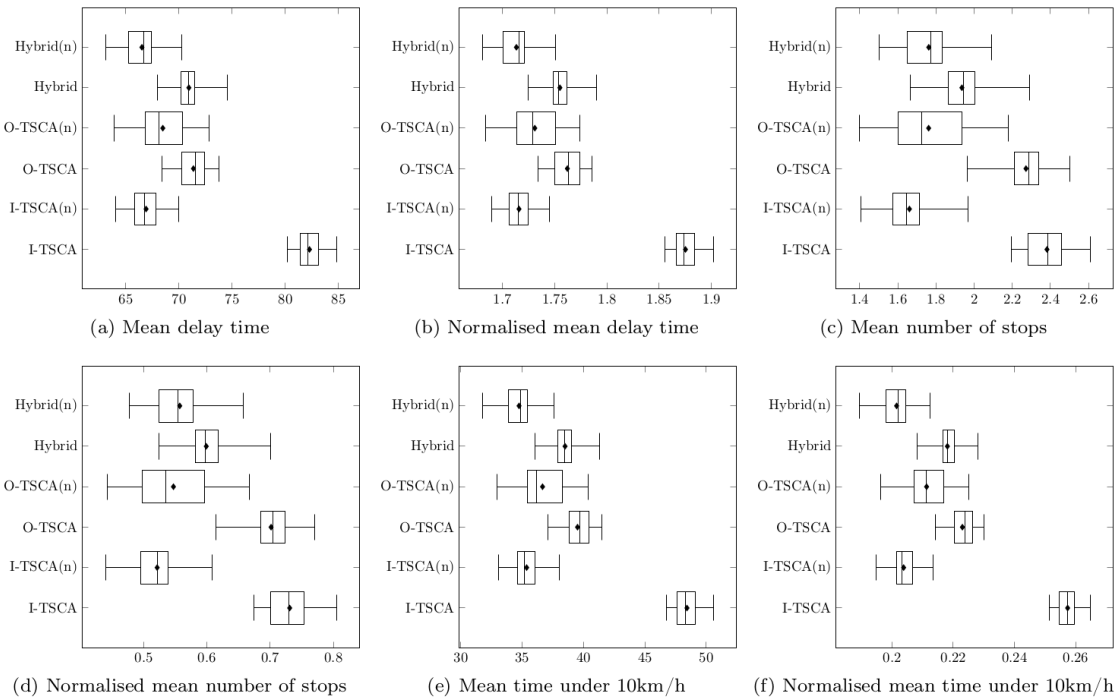


Figure 9: PMI results for the improved self-organising algorithms under heavy traffic conditions.

The I-TSCA(n) once again achieved the largest overall improvement, including a 15.36 second reduction in mean delay time from a value of 82.28 seconds to a mean value of 66.92 seconds (see Figure 9(a)) as well as a 30% reduction in the mean number of stops made (see Figure 9(c)). A 13.06 second improvement of the results of the I-TSCA(n) was observed in terms of the mean time spent travelling under 10 km/h over the corresponding results of the I-TSCA, achieving a value of 35.38 seconds (see Figure 9(e)) — an indication that the average time a vehicle spends travelling unacceptably slowly is reduced by 64%. The extremely fast switching of the I-TSCA was the principal cause for the poor performance

it exhibited before implementation of the changes recommended in §5, particularly under heavy traffic conditions. The longer green times resulting from the changes are the reason for the large improvement in the algorithm's performance.

The O-TSCA(n) achieved a smaller improvement over the O-TSCA under heavy traffic conditions than it did under light traffic conditions. The reason for this is that the O-TSCA is, in principle, better suited for heavy traffic conditions (due to its longer green times). The change resulting from considering a smaller section of road, therefore, did not yield such a large performance improvement, as the longer green times are better suited to heavy traffic conditions. The largest improvement of the O-TSCA(n) was in respect of the mean number of stops and normalised mean number of stops, improving in both these PMIs by just over 22% (see Figures 9(c) and 9(d)) when compared with the corresponding results of the O-TSCA. Improvements in the mean delay and the time spent travelling under 10 km/h were of a smaller margin, yet still significant at a 95% confidence level, obtaining improvements of 2.87 and 2.85 seconds, respectively (see Figures 9(a) and 9(e)).

The performance of Hybrid(n) was again improved the least out of the three algorithms proposed by Einhorn, although the improvement is still significant at a 95% level of confidence. The mean delay and the time spent travelling under 10 km/h improved by 4.42 and 3.74 seconds, respectively, while the mean number of stops was reduced by just over 9% in comparison with the original Hybrid.

7 A new algorithm based on vehicle platoons

Our newly proposed *Vehicle Platoon Traffic Signal Control Algorithm* (VP-TSCA) is a self-organising, adaptive algorithm that clusters vehicles along an intersection approach into a number of groups called *platoons*. The algorithm attempts to switch traffic signals so as not to separate platoons of vehicles travelling through an intersection when the distance between consecutive vehicles is less than a certain threshold. This algorithm is not predictive, but instead makes use of local, real-time traffic information pertaining to the relevant intersection.

The *level of traffic congestion* on an approach road is defined as the proportion of the approach road length actually occupied by vehicles. This level of traffic congestion is denoted by $x \in [0, 1]$. A so-called *within-platoon threshold distance*, denoted by D , is defined as the largest possible distance between two vehicles which admits platoon formation or clustering in which the two vehicles occupy the same platoon or cluster. The variables x and D are expected to be related to one another in an inversely proportional manner, because:

1. Lighter traffic conditions should result in the adoption of large within-platoon threshold distances as vehicles are more sparsely located along an approach road under such conditions, in which case it should be advantageous to cluster them together in order to promote the formation of a platoon.
2. Heavier traffic conditions should result in the adoption of small within-platoon threshold distances in order to avoid the formation of extremely large platoons which may, in turn, give rise to excessively long waiting times for conflicting traffic streams.

Perhaps the simplest functional form for such an inversely proportional relationship is

$$D = \frac{a}{x} + b, \quad (17)$$

where $a > 0$ is a constant of proportionality and $b \geq 0$ is an offset constant. Assuming this relationship, it follows that $D \rightarrow \infty$ as $x \rightarrow 0$. This would seem reasonable, since enforcing a finite upper bound on the platoon formation distance between vehicles seems unnecessary in the case where there are virtually no vehicles present on an approach road, as they should all form part of the same platoon. If, on the other hand, $x \rightarrow 1$, then it follows that $D \rightarrow a + b$. The purpose of the offset constant b is to facilitate the imposition of a finite lower bound $a + b$ on the within-platoon threshold distance under heavy traffic conditions which is independent of the rate a of the proportional decrease in D as x increases. Extensive numerical simulation experiments suggested that good values of the constants in (17) are $a = 2$ metres and $b = 10$ metres for the gridded road network considered in this paper. Thus, the relationship between the within-clustering threshold distance and level of traffic congestion becomes

$$D = \frac{2}{x} + 10. \quad (18)$$

The VP-TSCA comprises three main components. The first is a spillback prevention mechanism which ensures that vehicle queues never back up into neighbouring intersections. This is enforced by monitoring slow-moving vehicles along the intersection exit lanes that are within close proximity of the intersection. Let these vehicles belong to the set E_m , where m is the phase that is currently being served.

The second component of the algorithm involves determination of the level of traffic congestion and thus the within-platoon threshold distance. The level of traffic congestion for a signal phase m is denoted by x_m and is calculated by dividing the sum of the effective approach road lengths occupied by vehicles along approaches served during phase m , by the total length L_m of the approach road. Thus,

$$x_m = \sum_{j \in C_m} \hat{\ell}_j / L_m, \quad (19)$$

where C_m denotes the ordered set of approaching vehicles and $\hat{\ell}_j$ denotes the length of an approaching vehicle j . Once the road saturation x_m has been determined, the within-platoon threshold distance may then be calculated according to (18).

The third component of the algorithm is concerned with partitioning vehicles into platoons determined by the aforementioned within-platoon threshold distance during a phase m . At the start of phase m , a set P_m is defined to contain all the vehicles that form the initial platoon to be served.

8 Simulation results for new and improved algorithms

The performances of I-TSCA(n), O-TSCA(n) and Hybrid(n) of §5 are compared in this section with those of Fixed, Gersh, LH, and the novel VP-TSCA.

8.1 Simulation results under light traffic conditions

The O-TSCA(n) significantly outperformed the other six algorithms in terms of mean delay and normalised mean delay under light traffic conditions, obtaining values of 34.85 seconds and 1.376, respectively, as shown in Figures 10(a) and 10(b). The next best performing algorithm in respect of both these PMIs was Hybrid(n), which obtained corresponding values of 36.01 seconds and 1.388, respectively, followed by the I-TSCA(n) and Gersh, which did not differ significantly from one another in terms of these PMIs. The VP-TSCA achieved corresponding values of 37.69 and 1.405, which are respectively 11% and 3% improvements over LH which was statistically the worst performing algorithm in respect of these two PMIs. Fixed achieved a relatively average result of 38.99 seconds mean delay time, with a normalised value of 1.419.

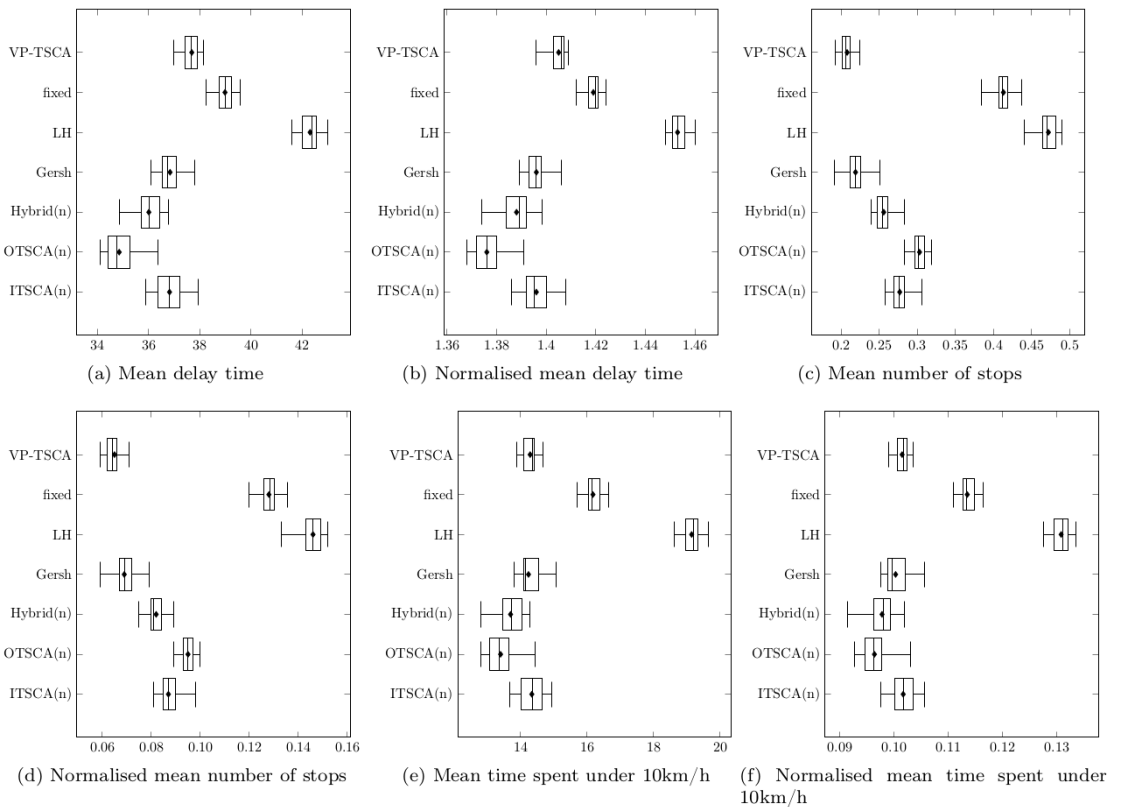


Figure 10: PMI results for seven self-organising algorithms under light traffic conditions.

Interestingly, the O-TSCA(n) was less effective in the prevention of vehicle stops than the other algorithms, with the exception of Fixed and LH, as may be seen in Figures 10(c) and 10(d). The VP-TSCA returned the most favourable results in this respect, significantly outperforming all other algorithms at a 95% level of confidence and achieving values of 0.208 and 0.065, respectively. This indicates that vehicles under the control of the VP-TSCA stop an average of 0.208 times throughout their journey and stop at 6.5% of the intersections that they encounter, suggesting that effective signal switching coordination was achieved at adjacent intersections. Gersh was the second best performing algorithm

in terms of the mean number of stops and the normalised mean number of stops made by vehicles, achieving values of 0.219 and 0.069, respectively, which amounts to 14.45% and 15.85% reductions of the results achieved by Hybrid(n) — the next best performing algorithm. Fixed and LH once again performed poorly with respect to these two PMIs as they failed to achieve good coordination between consecutive intersections.

The O-TSCA(n) and Hybrid(n) outperformed the other five algorithms in terms of the mean time a vehicle spends travelling under 10 km/h and the normalised mean time a vehicle spends travelling under 10 km/h, while they did not differ significantly from one another in terms of these PMIs (see Figures 10(e) and 10(f)). The VP-TSCA achieved a relatively average result in terms of mean time spent travelling under 10 km/h, returning a value of 14.29. The VP-TSCA obtained a value of 0.1015 in terms of normalised mean time spent under 10 km/h, outperforming both Fixed and LH, but not differing from the I-TSCA(n) at a 5% level of significance.

There is not an obviously superior algorithm in the scenario of light traffic as there was not a single algorithm that performed the best over all six PMIs, and the PMIs cannot be ranked according to importance as the relative importance of the PMIs is subjective to road users. It can, however, be claimed that Fixed and LH were statistically the two worst performing algorithms under light traffic conditions. The other five algorithms performed comparatively well, but the VP-TSCA and O-TSCA(n) were the best performing algorithms in respect of certain PMIs under light traffic conditions.

8.2 Simulation results under heavy traffic conditions

Fixed achieved the best results in terms of mean delay under heavy traffic conditions, significantly outperforming the VP-TSCA, LH, Hybrid(n), the I-TSCA(n), the O-TSCA(n) and Gersh by margins of 14%, 15%, 15%, 16%, 18% and 34%, respectively, as may be seen in Figures 11(a) and 11(b). The VP-TSCA, LH, Hybrid(n) and the I-TSCA(n) were found to perform similarly in respect of these PMIs at a 5% level of significance and all significantly outperformed O-TSCA(n) and Gersh. Gersh is very clearly the worst performing algorithm in respect of mean delay time and normalised mean delay time, as shown in Figures 11(a) and 11(b), obtaining corresponding values of 77.90 seconds and 1.834, respectively. The VP-TSCA did not perform as well in terms of mean number of stops or normalised mean number of stops as it did under light traffic conditions. It achieved the second worst value in both these PMIs, improving only upon Gersh, and obtaining values of 1.862 and 0.584, respectively (see Figures 11(c) and 11(d)). All other algorithms, however, outperformed Gersh significantly, with Gersh obtaining a value of 3.359 — double that of any of the other algorithms. LH and the I-TSCA(n) obtained the best results in terms of these PMIs, significantly outperforming all other algorithms at a 95% level of confidence.

Fixed achieved the smallest value for the mean time spent travelling under 10 km/h as well as for the normalised mean time spent travelling under 10 km/h, obtaining values of 28.26 and 0.1682, respectively (see Figures 11(e) and 11(f)), significantly outperforming all the other algorithms. The VP-TSCA was the next best performing algorithm in respect of these PMIs, outperforming all algorithms other than Fixed. The O-TSCA(n) performed significantly worse, outperforming Gersh only and obtaining a normalised value of 0.2113

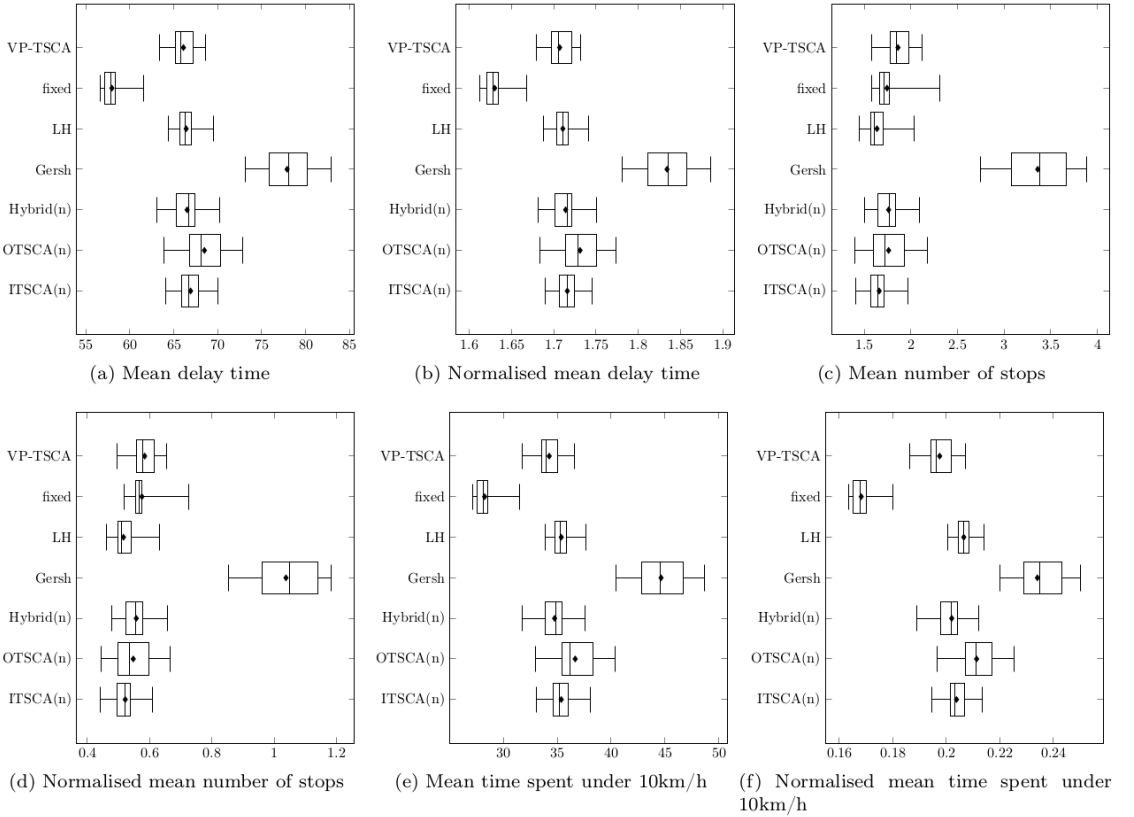


Figure 11: PMI results for seven self-organising algorithms under heavy traffic conditions.

which means that vehicles under the control of this algorithm spent 21.13% of their total travel time travelling under 10 km/h. Gersh is considered the worst performing algorithm in respect of both the PMIs associated with vehicles travelling under 10 km/h.

It is concluded that Gersh is the overall worst performing algorithm under heavy traffic conditions, obtaining the largest values across all six PMIs, as may be seen in Figure 11. This is as a result of the long green times that Gersh employs. Fixed and LH are statistically the best performing algorithms under heavy traffic conditions, obtaining the best PMI values overall.

9 Conclusion

From the results reported in §5 it is clear that the O-TSCA and Hybrid were outperformed by Gersh under light traffic conditions in respect of every PMI, and were outperformed by Fixed and LH under heavy traffic conditions in respect of every PMI. These findings do not corroborate the conclusions drawn by Einhorn [3], who found that the O-TSCA achieved the most desirable PMI values in a 3×4 grid topology, followed by Hybrid. A possible reason for this may be the different simulation models that were used as test beds, although the authors' model was purposefully designed to resemble the model by Einhorn

as closely as possible. We have confidence in our model, however, due to the validation procedure to which it was subjected, as described in §3.4.

Once the algorithms had been implemented in our simulation model, certain shortcomings of the three algorithms of Einhorn became apparent. These shortcomings were addressed and the algorithms were modified within the simulation framework. The results obtained by these modified algorithms were compared with those of their original counterparts and it was found they had been improved upon by a large margin at a 95% level of confidence.

Finally, a new self-organising algorithm was proposed in §7, called the VP-TSCA. The performances of this algorithm and those of the three improved algorithms were then compared with the performances of LH and Gersh. The VP-TSCA and the O-TSCA(n) were found to achieve the best results comparatively, in terms of good PMI values under light traffic conditions, while under heavy traffic conditions Fixed was found to be the best performing algorithm, as expected. Under light traffic conditions it is therefore suggested that the VP-TSCA or O-TSCA(n) be considered for implementation, while Fixed should be considered for implementation in areas of heavy traffic congestion.

10 Future work

All the algorithms considered in this paper have only been tested within the context of a 3×4 grid of intersections. It is, however, suggested that larger grids also be considered in order to test how well each of the five self-organising algorithms scale.

The results reported in this study may be validated by implementing the same experiments in another simulation software model in order to test whether similar results are obtained.

The gridded transportation network considered in this paper is a hypothetical road network. It is therefore suggested that the algorithms be compared in a simulation model of an existing road network, together with real vehicle flow rates experienced in that network, so that the algorithms may be compared in a real-world context.

References

- [1] ANYLOGIC, 2016, *Multimethod simulation software*, [Online], [Cited September 20th, 2018], Available from <http://www.anylogic.com>.
- [2] DE WOLF T, SAMAAY G & HOLVOET T, 2005, *Engineering self-organising emergent systems with simulation-based scientific analysis*, Proceedings of the 3rd International Workshop on Engineering Self-Organising Applications, New York (NY), pp. 146–160.
- [3] EINHORN MD, 2015, *Self-organising traffic control inspired by inventory theory and the process of osmosis*, PhD dissertation, Stellenbosch University, Stellenbosch.
- [4] GERSHENSON C & ROSENBLUETH DA, 2012, *Adaptive self-organization vs static optimization: A qualitative comparison in traffic light coordination*, *Kybernetes*, **41(3)**, pp. 386–403.
- [5] HEYLIGHEN F, 1989, *Self-organization, emergence and the architecture of complexity*, Proceedings of the 1st European Conference on System Science, Stuttgart, pp. 23–32.
- [6] HEYLIGHEN F, 2001, *The science of self-organization and adaptivity*, *The Encyclopedia of Life Support Systems*, **5(3)**, pp. 253–280.

- [7] *Highway capacity manual*, 2000, Transportation and Research Board, Washington (DC).
- [8] HILTON A & ARMSTRONG R, 2010, *One-way analysis of variance*, John Wiley & Sons, Inc., Hoboken (NJ).
- [9] HOWELL JF & GAMES PA, 1974, *The effects of variance heterogeneity on simultaneous multiple-comparison procedures with equal sample size*, British Journal of Mathematical and Statistical Psychology, **27**(1), pp. 72–81.
- [10] KOONCE P, RODEGERDTS L, LEE K, QUAYLE S, BEAIRD S, BRAUD C, BONNESON J, TARNOFF P & URBANIK T, 2008, *Traffic signal timing manual*, (Unpublished) Technical Report, Transportation and Research Board, Portland (OR).
- [11] LÄMMER S & HELBING D, 2008, *Self-control of traffic lights and vehicle flows in urban road networks*, Journal of Statistical Mechanics: Theory and Experiment, **2008**(4), pp. 1–30.
- [12] MILLER AJ, 1963, *Settings for fixed-cycle traffic signals*, Journal of the Operational Research Society, **14**(4), pp. 373–386.
- [13] ODELL J, 2002, *Agents and complex systems*, Journal of Object Technology, **1**(2), pp. 35–45.
- [14] DEPARTMENT OF TRANSPORTATION, STATE OF WISCONSIN, *Paramics Calibration Settings*, [Online], [Cited February 21st, 2019], Available from <https://wisconsin.gov/dtsdManuals/traffic-ops/manuals-and-standards/teops/16-20att6.2.pdf>.
- [15] SERUGENDO GDM, GLEIZES MP & KARAGEORGOS A, 2006, *Self-organisation and emergence in MAS: An overview*, Informatica, **30**(1), pp. 45–54.
- [16] SHALIZI CR, 2001, *Causal architecture, complexity and self-organization in the time series and cellular automata*, PhD dissertation, University of Wisconsin, Madison (WI).
- [17] SCHULTZ BB, 1985, *Levene’s test for relative variation*, Systematic Biology, **34**(4), pp. 449–456.
- [18] SYSTEMS IT, 2016, *SUMO — Simulation of Urban Mobility*, [Online], [Cited September 20th, 2018], Available from <http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931/>.
- [19] VAN DER MERWE E, 2009, *Improving the flow of an isolated traffic system*, MA thesis, Stellenbosch University, Stellenbosch.
- [20] WAVETRONIX, 2016, *Wavetronix SmartSensor Advance*, [Online], [Cited March 23rd, 2016], Available from <http://www.wavetronix.com/en/products/smartsensor/advance>.
- [21] WEBSTER FV, 1958, *Traffic signal settings*, (Unpublished) Technical Report, Road Research Laboratory, London.
- [22] WILLIAMS LJ & ABDI H, 2010, *Fisher’s least significant difference (LSD) test*, Encyclopedia of Research Design, pp. 1–5.