SAPIENZA
UNIVERSITÀ DI ROMA

Department of Civil and Environmental Engineering

PhD Thesis
Infrastructure and Transport Engineering
XXXI CYCLE

Forecast Based Traffic Signal Coordination Using Congestion Modelling and Real-time Data

Tutor
Prof. Guido Gentile

Candidate
Pietro Meschini, MEng

I wish to extend my heartfelt gratitude
to my tutor Guido Gentile, who bore with me through this journey,
to my referee Andrea Papola, who provided invaluable feedback,
and to my friends and colleagues at SISTeMA, for their unfaltering support,
without whom I would have never made it this far.

A Maria Chiara,
che ci aveva visto giusto.

p

Abstract

This dissertation focusses on the implementation of a Real-Time Simulation-Based Signal Coordination module for arterial traffic, as proof of concept for the potential of integrating a new generation of advanced heuristic optimisation tools into Real-Time Traffic Management Systems. The endeavour represents an attempt to address a number of shortcomings observed in most currently marketed on-line signal setting solutions and provide better adaptive signal timings. It is *unprecedented* in its use of a Genetic Algorithm coupled with Continuous Dynamic Traffic Assignment as solution evaluation method, only made possible by the recently presented parallelisation strategies for the underlying algorithms.

Within a fully functional traffic modelling and management framework, the optimiser is developed independently, leaving ample space for future adaptations and extensions, while relying on the best available technology to provide it fast and *realistic* solution evaluation based on reliable real-time supply and demand data. The optimiser can in fact operate on high quality network models that are well calibrated and always up-to-date with real-world road conditions; rely on robust, multi-source network wide traffic data, rather than being attached to single detectors; manage area coordination using an external simulation engine, rather than a naïve flow propagation model that overlooks crucial traffic dynamics; and even incorporate real-time traffic forecast to account for transient phenomena in the near *future* to act as a feedback controller.

Results clearly confirm the efficacy of the proposed method, by which it is possible to obtain relevant and consistent corridor performance improvements with respect to widely known arterial bandwidth maximisation techniques under a range of different traffic conditions.
The computational efforts involved are already manageable for realistic real-world applications, and future extensions of the presented approach to more complex problems seem within reach thanks to the load distribution strategies already envisioned and prepared for in the context of this work.

Contents

Introduction

The fundamental role of *traffic signals* is to equitably and efficiently administer the right of way amongst conflicting streams of road users.

Since the first sporadic appearances around the turn of the 20th century, traffic lights have become a ubiquitous feature in the everyday life of all road users, regardless of their preferred mode of transportation: whether they sit behind the wheel of their own car, walk, or let the public service carry them about their business, traffic lights will be regulating their movements and those of others around them (most noticeably, of those in front).

It is therefore natural that traffic signals should garner so much attention: they are perceived (only *sometimes* unfairly) as a major source of delay and frustration to drivers, and the tantalising idea of an intelligent traffic control system often comes to identify, in the general public's fantasies, with the very notion of an *Intelligent Transport System*.

In fact, a history of case studies shows that wherever public money has been invested into the development and maintenance of a signalisation system tailored to the transportation needs of a community, the returns have invariably surpassed expenditures by far in terms of direct fuel and time savings distributed among road users, and indirectly reduced health and safety costs for the community at large [Koonce, Rodegerdts, Lee, Quayle, Beaird, Braud, Bonneson, Tarnoff, and Urbanik, 2008].

The most effective way to alleviate urban traffic congestion by *orders of magnitude* is to provide viable alternatives to the private car and promote a modal shift to more sustainable forms of collective or personal transport: however, carefully planned signalisation allows a more efficient use of the existing road infrastructure, minimising the stress suffered by drivers as well as the risk of accidents, favouring public transport and improving air quality, with a positive impact on virtually every aspect of life in a modern city.

About Notation

A quick glossary of the relevant variables is provided below, alongside the units of each dimensional quantity.

For a leaner presentation of the model, subscripts referring to topological elements may be dropped to simplify notation.

Network Topology

$i, j \in \mathrm{N}$		nodes (junctions)
$a, b \in \mathrm{A} \subseteq \mathrm{N} \times \mathrm{N}$		arcs (lane groups)
ℓ_a	[m]	length of arc a
$(\mathrm{N}_a^- \,,\, \mathrm{N}_a^+) = a$		tail and head nodes of arc a
$\mathrm{A}_i^+ = \{a \in \mathrm{A} \,\vert\, \mathrm{N}_a^- = i\}$		forward star of node i (outgoing arcs)
$\mathrm{A}_i^- = \{a \in \mathrm{A} \,\vert\, \mathrm{N}_a^+ = i\}$		backward star of node i (incoming arcs)
$y, z \in \mathrm{Y}$		manoeuvres

Signal Phases

$p, q \in \mathrm{P}_j$	signal phases at junction j
$\mathrm{A}_p \subseteq \mathrm{A}_j^-$	lane groups open during phase p

Signal Timing

t_j^C	[s]	cycle time at junction j
t_j^L	[s]	time lost per cycle at junction j
t_j^O	[s]	offset of junction j
t_p	[s]	nominal duration of phase p
t_a^g		initial instant of the green phase for arc a
g_a	[s]	effective green duration for arc a
$G_a = [t_a^g \,, t_a^g + g_a]$		span of the green phase for arc a
$\gamma_a = \dfrac{g_a}{t_j^C}$	[%]	effective green share of arc a

Demand and Supply

q_a	[veh/s]	demand flow on arc a
$\hat{q}_a$	[veh/s]	saturation flow of arc a
$\phi_a = \dfrac{q_a}{\hat{q}_a}$		flow ratio on arc a
$\chi_a = \dfrac{\phi_a}{\gamma_a}$		saturation on arc a

Simulation Parameters

$\tau \in \mathrm{T}$		intervals of the simulation window
Δt^{τ}	s	duration of interval τ

Simulation Results

$n^{F}_{a,t}$	[veh]	vehicles that entered arc a before time t
$n^{G}_{a,t}$	[veh]	spaces that reached N^{-}_{a} before time t
$n^{H}_{a,t}$	[veh]	vehicles that reached N^{+}_{a} before time t
$n^{E}_{a,t}$	[veh]	vehicles that left arc a before time t
$n_{a,t}$	[veh]	vehicles on arc a at time t
$t^{t}_{a,t}$	[s]	travel time for arc a entering at time t
$n^{Q}_{a,t}$	[veh]	vehicles in the vertical queue at N^{+}_{a} at time t

Performance Indicators

t^{Q}_{a}	s	queue clearance time on arc a (per cycle)
ω^{Q}_{a}	[%]	queue length relative to total length of arc a
ω^{D}_{a}	s	average delay of arc a
ω^{S}_{a}		share of q_a stopping at or before N^{+}_{a}
ω^{n}_{a}, ω^{n}_{C}	[veh]	total inflow to arc a or all sections of corridor C
ω^{t}_{a}, ω^{t}_{C}	[s]	user time spent on arc a or corridor C

Chapter 1

Signalisation of Urban Networks

The present work concerns the regulation of urban traffic by means of traffic signals.

The *lights*, which are nowadays a ubiquitous feature of the urban landscape, first appeared in 1868 outside the British House of Commons in Victorian London, where the horse drawn carriage traffic was becoming an insurmountable barrier posing a serious threat to pedestrians. Since then, and especially as motor cars were introduced, traffic regulation proved indispensable to administer the right of way among competing traffic flows and safeguard the more vulnerable users of the urban road environment.

This chapter introduces the formal representation of the signalised road network used for all practical purposes in this dissertation. It builds upon the definition of the network itself to describe the way it interacts with its users, modelling the problems that traffic signals need to tackle and the ways in which they might do so. Finally, the most relevant signal planning approaches based on the paradigm just outlined are illustrated, as they form the basis for the adaptive signalisation strategies to which the present work aims to contribute.

1.1 The Urban Network

In the context of transport modelling and planning, a transportation network is represented as a *directed graph* in the mathematical sense, with the *vertices* representing locations and the *edges* connections that a user may travel between them. The term *connection* is used loosely on purpose here, since in general these need not be *roads* but may be transit lines, footpaths, railways etc. each with complex properties which determine its *cost*, or even accessibility, to a given class of users.

In its extremely simplified acceptation of *road network* which will serve the purposes of the present work, a transportation network may be reduced to an ordered pair (N, A) where

- N is the set of vertices of the graph, called *nodes*, representing junctions and road ends;
- A is the set of directed edges between them, called *arcs*, along which the users move.

This allows to encapsulate both the network topology and the properties of individual roads, which determine the way in which the users will interact with them: the choice of a path between two nodes depends on the perceived cost of each alternative as determined by a combination of its properties, e.g. length, toll, number of lanes, pleasantness; the same properties, albeit through conceptually different mechanisms, determines how the users will be able to move along the chosen path.

1.2 Anatomy of a Signal Plan

The following section briefly illustrates the main features of a signal plan devised for urban traffic regulation. This term encompasses all timings and schedules behind the delicate clockwork of traffic signals, from the elements that constitute a single signal program at one of the many junctions of the network, to the succession of network-wide program changes designed to meet the daily evolution of traffic demand and the propagation of vehicle flows. The features presented in this section fully define what is commonly called a pre-timed plan, and as such do not describe any real-time actuation or decision making logic. They are themselves, however, the decision variables of most optimisation methods and adaptive strategies, and it is crucial to understand their significance in order to appreciate the diversity of setting and control approaches illustrated in more detail throughout this chapter.

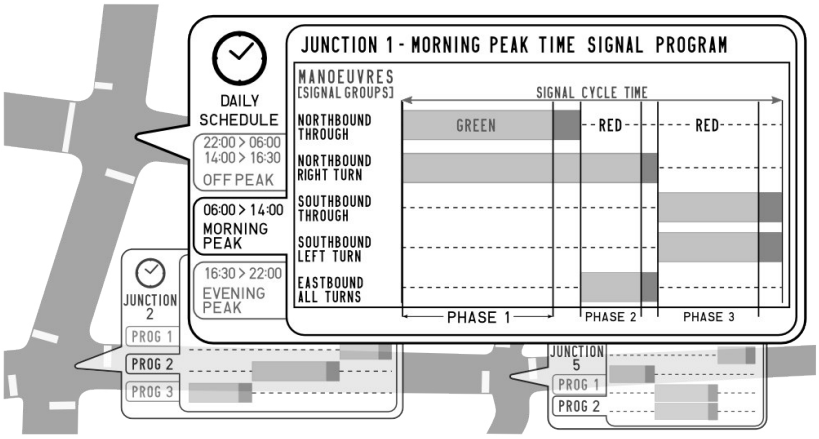

FIGURE 1.1 – Elements of a network-wide signal plan: a daily schedule specifies the signal programs running at each intersection. The sequence and duration of signal phases repeats over the course of every signal cycle as specified by the different signal programs, administering junction capacity amongst the expected traffic flows. During each phase, a set of compatible manoeuvres is allowed through while the others remain closed.

Signal Phases

Traffic signals exist mainly to separate conflicting traffic flows competing for the right of way at a road intersection. The natural way of doing so is to bundle compatible (e.g. non-secant) manoeuvres which may be safely performed simultaneously into signal phases, so that the corresponding flows may be allowed through the junction in turn. Phases are the fundamental blocks of a signal program, and are usually repeated in the same order at every signal cycle, although some signalisation systems provide phase skipping, usually as part of their public transport prioritisation strategy. Manoeuvres may pertain to different modes of transport, meaning that cars, trams and pedestrians are taken into joint consideration and can be given the right of way during the same signal phase.

Consider a junction, i.e. a network node $j \in \mathrm{N}$ where it is possible to perform a given set of manoeuvres Y_j . The generic manoeuvre $y \in \mathrm{Y}_j$ may be:

- a turn, from an arc $a \in \mathrm{A}_j^-$ of the node's backward star, to a forward star arc $b \in \mathrm{A}_j^+$;
- a tram crossing or similar transport system specific operation;

- a pedestrian crossing affecting one or more arcs either entering or leaving the junction.

In order to present a straightforward definition of *manoeuvres* in relation to junction layout and signalisation, the focus will henceforth be on the movement of private vehicles only, unless otherwise specified. It shall be clear that the principles of manoeuvre compatibility illustrated in this manner in Figure 1.2 may be easily generalised to different and etherogeneous modes of transport, such as public transport, pedestrians and bicycles.

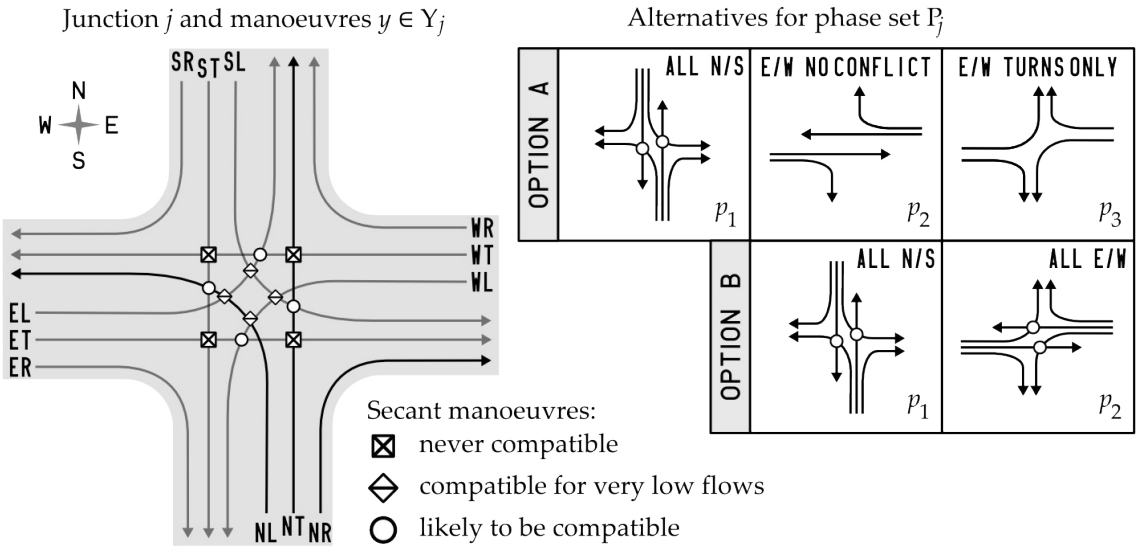

FIGURE 1.2 – Manoeuvres at an intersection, conflict areas and possible phasing options: option A avoids direct conflicts between Eastbound (E-) and Westbound (W-) manoeuvres, as would be desirable if high volumes were expected along that direction; option B favours a lower number of phase changes (less time lost) assuming flows to be such that left turning vehicles have space to wait at the middle of the intersection, until the oncoming through flow decreases enough to let them cross.

Given the layout of a junction j, different manoeuvres may or may not be safe to perform simultaneously, as exemplified in Figure 1.2. This information, which may well depend on the flow conditions, is easily represented by a square Boolean matrix where rows and columns correspond to each manoeuvre and elements comply with the following rule:

$$\delta_{yz} = \begin{cases} 1 & \text{if } y \text{ and } z \text{ are compatible} \\ 0 & \text{otherwise} \end{cases} \quad \forall y, z \in \mathrm{Y}_j \quad . \tag{1.1}$$

Each possible subset of manoeuvres $p \subseteq \mathrm{Y}_j$ potentially identifies a *signal phase*. A viable set of phases P_j for the junction however must belong to the space of *feasible* signal phases, i.e. all possible sets of manoeuvres contained in the power set $\wp(\mathrm{Y}_j)$ whose elements are mutually compatible according to (1.1) . The union of all phases must also include every available manoeuvre at least once.

Formally, P_j must therefore comply with the following properties:

$$\mathrm{P}_j = \left\{ p \in \wp(\mathrm{Y}_j) : \prod_{y \in p} \prod_{z \in p} \delta_{yz} = 1 \right\} \quad , \quad \bigcup_{p \in \mathrm{P}} p = \mathrm{Y}_j \quad . \tag{1.2}$$

Clearly, the power set $\wp(\mathrm{Y}_j)$ contains sets of manoeuvres that, although compatible and technically feasible, make little practical sense. The selection of an optimal set of phases P_j satisfying relation (1.2) with respect to a specific objective (e.g. minimum total delay for given demand flows) is a combinatorial bi-level problem, usually solved through a *what-if* approach in which the selection of a good set of phases remains largely a traffic engineer's task.

Conceptually, the determination of signal phases is thus driven by the interactions between manoeuvres. From a practical point of view, however, administration of the right of way by means of traffic signals cannot transcend the junction layout. For example, it is only possible to separate manoeuvres into different phases if each has a dedicated lane that allows vehicles to queue for it without hindering traffic that is headed elsewhere. In fact, as everyday experience testifies, traffic signals do not allow or prohibit manoeuvres directly, but rather regulate vehicle egress from lanes (or lane groups) dedicated to specific sets of manoeuvres.

Each lane or group of adjacent lanes a sharing the same manoeuvre set $\mathrm{Y}_a \in \mathrm{Y}_j$ can be conceptually assimilated into a *lane group*: a single independent arc $a \in \mathrm{A}_j^-$ of the node backward star. Let A_p be the set of lane groups which are given the green light during signal phase p, and Y_a the manoeuvres that can be performed from lane group a. The set of manoeuvres allowed during phase p is therefore

$$p = \bigcup_{a \in \mathrm{A}_p} \mathrm{Y}_a \quad . \tag{1.3}$$

The set of manoeuvres Y_a specific to each lane group a is relevant for the determination of the arc effective outflow capacity, which may be affected by partial conflicts with other manoeuvres allowed during the same phase. Highway Capacity Manuals such as [Special Report 209, 1985] present practical methods for quantifying such effects.

Signal Programs

A signal program contains the state switching times for all signals at a given junction. For signal planning and optimisation, it is practical to view the program as a succession of signal phases with specific durations, as portrayed in Figure 1.1: during each phase a set of arcs are open, allowing users to carry out the corresponding manoeuvres, while the others arcs remain closed and accumulate queues.

A program for junction j consists therefore of a cyclic set of instructions spanning a period called *cycle time* t_j^C: given a phase set P_j, these specify the start and end of each signal phase with respect to the beginning of the signal cycle.

Transitions between subsequent phases are usually enacted via pre-timed signal state change sequences that handle the closure of a set of lane groups before opening the next.

Daily Schedule

It is common practice to tailor several signal programs to the traffic conditions normally observed at different times of the day, in order to meet each scenario with the best possible allocation of resources. The daily schedule defines the sequence of programs that each junction will run over the course of the day.

Cycle Time

The cycle time t_j^C is the *period* of the signal program, i.e. the time lapse between two occurrences of the same signal phase at a given junction. It affects the average delay and the level of saturation at which the intersection may operate. In general, longer cycle times imply larger average delays, but increase the total throughput, which may be necessary to deal with high demand flows by attenuating the effects of the time lost in signal phase changes.

Effective Green Shares

The nominal duration of each phase t_p is seldom exploited by demand flows at the full capacity of the corresponding arcs: even assuming that vehicles are not held back by downstream congestion, it is necessary to account for some transient phenomena affecting the performance of a junction.

As the signals turn green at the beginning of each phase, some time is lost before the queuing vehicles start moving, and some more passes before the flow through the stop line reaches the arc capacity. On the other hand, if a lane group remains open during two subsequent phases such effects will be smaller, in proportion. After taking into account all delays and extensions, the portion of cycle time during which a given lane group a may allow traffic onto the junction at full capacity is referred to as its *effective green share*. The absolute and relative durations of effective green experienced by lane group a during phase p are denoted respectively as:

$$g_{a,p} \in [0, t_p] \qquad \text{and} \qquad \gamma_{a,p} = \frac{g_{a,p}}{t_j^C} \quad . \tag{1.4}$$

It is not uncommon to have a lane group open during more than one phase: typically, an approach experiencing high traffic volumes is given the right of way over two or more consecutive phases without incurring further lost time in the phase change.

The effective green of each arc a is then calculated from the total effective green time it gathers over all relevant phases:

$$g_a = \sum_{p \in \mathrm{P}_j} g_{a,p} \quad \text{with} \begin{cases} 0 < g_{a,p} \leqslant t_p & \text{if } a \in \mathrm{A}_p \\ g_{a,p} = 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \gamma_a = \frac{g_a}{t_j^C} \quad . \tag{1.5}$$

Signal Offset

When multiple signals are involved, it is important to consider that vehicles that cross a signalised junction become packed into *platoons*, which will eventually reach yet another signal-controlled stop line: adjusting the relative timing of adjacent junctions so that platoons meet a green light greatly affects the average delay incurred by the user.

Synchronisation issues are addressed by defining a global time reference, with all junctions sharing the same cycle time or integer fractions thereof. Each junction may then have all of its phase switching times anticipated or delayed in order to operate in concert with the neighbouring ones. The amount of time t_j^O, by which the beginning of a cycle at one junction j lags or leads the global reference instant, is referred to as a positive or negative *offset*, respectively.

1.3 Signal Setting

Long before microprocessors and sensors made adaptive real-time traffic control an everyday reality, the notion of signal plan *optimisation* identified with the plan design phase.
The well known techniques used throughout the last century to design good signal plans based on historical demand flows will henceforth be referred to as *offline* signal setting.

It is worth noting that such methods are not only still used for planning, but lie at the core of several adaptive signal setting approaches: once a *signal setting policy* is chosen to determine the best signalisation parameters for given traffic conditions, it makes little difference from the methodological point of view whether the input variables are determined from historical data or fed in real-time by sensors.

Naturally, the notion of offline planning does not imply that the dynamic interaction between signal setting and driver behaviour can be disregarded: for example, the assumption often made that route choices are fixed and unaffected by signal settings has warranted the formulation of planning strategies which have proven quite patently inadequate in the real world, as first discussed in [Dickson, 1981].

While optimisation of a single junction for given flows may be a relatively simple problem with an analytical solution, devising a plan for an entire network is an entirely different task.

This section introduces the fundamentals of network signalisation design, describing the methods commonly used to determine the foremost features of a signal program, including cycle time, offsets and green share allocation.

1.3.1 Performance of Isolated Signalised Junctions

Before considering a whole signalised network, it is useful to define the concept of *performance* of a generic junction. To understand the quantities and processes involved, an*isolated* junction may be considered, to allow disregarding the effects of other junctions (such as vehicle platooning) and concentrate on the modelling of congestion phenomena. A signalised junction behaves exactly like any junction where flows merge, diverge and cross, with the exception that the availability of certain manoeuvres is time dependent and the corresponding flows, administered by means of traffic lights, may be periodically forced to zero.

The performance of a signalised junction may be defined in several ways, but in general terms it represents a gauge of the interaction between supply and demand with respect to a choice of metrics. As such, it depends on the junction physical layout, on the distribution of vehicle arrivals in time and on the signal that regulates their departure times.

Several flow models were introduced in scientific literature to reproduce arrival and departure phenomena. For all signal planning purposes, traffic flow is usually assimilated to a fluid stream according to the *macroscopic* paradigm, which differs substantially from the microscopic approach where the trajectory of each single vehicle is explicitly considered.

More specifically, vehicle *departures* from a stop line are modelled as a uniform flow. If the *arrival* flows are sufficiently lower than capacity, their inherent random component can be neglected and they are also considered deterministic. Conversely, if stochasticity of arrival flows is significant, as it occurs when they approach the relevant arc capacity, or are very low, a random component is added to the simple deterministic model as in [Webster, 1958]. This section will present the basic relationships between signal timing variables and junction performance with reference to the simple deterministic model.

Queues and Queue Clearance

Consider a single arc (lane group) $a \in \mathrm{A}_j^-$ entering a signalised junction $j \in \mathrm{N}$, with a constant demand flow of vehicles q_a arriving over the entire cycle. The flow can only be discharged onto the junction during the effective green time, at the constant saturation flow rate $\hat{q}_a$ given by the arc capacity and possibly degraded due to conflicts with other arc flows. The *flow ratio* between demand and saturation is denoted as:

$$\phi_a = \frac{q_a}{\hat{q}_a} \quad . \tag{1.6}$$

During the rest of the cycle, the departure rate is zero and vehicles have to stop, forming a *queue*, which has to be discharged during the next green phase if it is not to grow indefinitely.

The saturation flow $\hat{q}_a$ must therefore be sufficient to serve the queue accumulated over the red phase, which has duration $t_j^C - g_a$, in addition to the flow of vehicles that keep arriving during the green phase g_a.

This relationship is illustrated in figure 1.3 and may be formalised by considering the following expression for the *queue clearance time* in terms of the signal timing and flows just described:

$$t_a^Q = \frac{q_a \left(t_j^C - g_a\right)}{\hat{q}_a - q_a} = \frac{\phi_a(1-\gamma_a)}{1-\phi_a}\, t_j^C \quad , \quad \forall a \in \mathrm{A}_j^- \quad . \tag{1.7}$$

Vehicle Stops

In this context, it makes sense to assume that vehicles will stop if they reach the stop line during the red phase or if they have to join the back of a queue that has yet to be fully discharged, although this is a slightly conservative approximation as the back of the queue might not be standing still during the green phase.

The number of vehicles that end up stopping (or significantly slowing down) during every signal cycle can therefore be expressed as

$$n_a = q_a \left(t_j^C - g_a + t_a^Q\right) = \hat{q}_a\, t_a^Q \tag{1.8}$$

where the right-hand side equality is justified simply by the definition of clearance time t^Q given by equation (1.7) under the assumption that standing vehicles will discharge onto the junction at the maximum possible flow rate during the effective green phase.

This in turn leads to the theoretical definition of the *stop ratio*, an essential metric indicating what fraction of the total flow of vehicles will have to stop at the junction:

$$\omega_a^S = \frac{\hat{q}_a t_a^Q}{q_a t_j^C} = \frac{1-\gamma_a}{\phi_a} \quad , \tag{1.9}$$

which is proportional to the red share of the cycle time and increases as the arrival rate approaches the discharge capacity. Quite obviously for values of $\phi_a \geqslant 1$, but also if $\gamma_a < \phi_a$ queues cannot be fully discharged at every cycle, and all vehicles end up stopping: in this case, the queue can grow indefinitely.

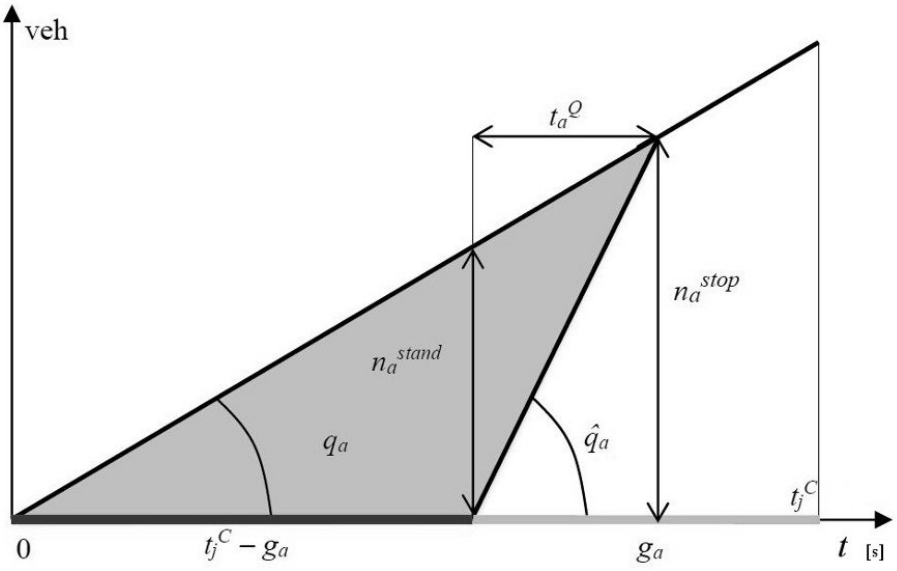

FIGURE 1.3 – Geometric determination of stopped vehicles and queue clearance for one approach given the relevant demand flow, saturation flow, cycle and green time. The grey triangle between the arrival cumulative, the departure cumulative and the horizontal axis covers the number of vehicles queuing at any given moment. Notice that the number of standing vehicles $n_a^standing$ at the beginning of the effective green does not account for all vehicles that need to stop n_a^stop according to the approximation given by equation (1.8).

Average Delay

Assuming constant arrival and departure rates, the total delay experienced at each cycle by all users from a given approach a corresponds to the integral over time of the queue size (the area of the greyed out triangle in Figure 1.3), whence the average delay ω_a^D per vehicle is found to be

$$\omega_a^D = \frac{\left(t_j^C - g_a\right)\left(\hat{q}_a\, t_a^Q\right)}{2\,\left(q_a\, t_j^C\right)} = \frac{\left(t_j^C - g_a\right)^2}{2\,\left(1-\phi_a\right)\,t_j^C} \quad , \tag{1.10}$$

using (1.7) for the queue clearance time t_a^Q.

Clearly, the above equation (1.10) assumes no standing queues at the end of a cycle. More complex delay functions can be obtained by considering stochastic fluctuations of arrival flows as stated by Webster [1958]. Flows exceeding the arc capacity require the introduction of either simulation models or empirical adaptations of analytical models, such as the coordinate transformation method introduced by Kimber and Hollis [1979] and later adopted by the popular [Special Report 209, 1985] and subsequent revisions known as the Highway Capacity Manual.

Critical Flow Ratio and Saturation

The saturation flow characterising each lane group depends on various factors, such as

- total road width,
- visibility,

- conflicts with other manoeuvres served during the same phase,
- presence of dedicated turn bays to alleviate such conflicts.

Conflicts are particularly relevant to left turns, or turns encroaching a pedestrian crossing: scrupulous phase planning can minimise the number and entity of such conflicts.

The flow ratio ϕ_a quantifies the expected demand on a given lane group a in relation to its *nominal* saturation capacity. The saturation level χ_a is determined by the ratio of demand flow to its *outflow capacity*, which is further limited by the signal, inasmuch as each arc can only be open for a limited share of the available green time:

$$\chi_a = \frac{q_a}{\gamma_a \hat{q}_a} = \frac{\phi_a}{\gamma_a} \quad . \tag{1.11}$$

For values of $\gamma_a < \phi_a$ the saturation level is above 100 % and the flow cannot be served, leading to queues that grow indefinitely until demand drops.

When multiple lane groups are to be open simultaneously during phase p, the *critical flow ratio* ϕ_p is given by the approach which is relying most heavily on the phase in question. The concept is formalised in equation (1.12) by scaling the flow ratio of each approach in proportion to the share of its green time represented by the current phase.

In other words, in searching for the maximum flow ratio, only the share of flow that each lane group must serve during the specific phase is considered:

$$\phi_p = \max \left\{ \phi_a \frac{\gamma_{a,p}}{\gamma_a} \mid a \in \mathrm{A}_p \right\} \quad , \tag{1.12}$$

whence conversely the *critical lane group* of phase p is also identified as

$$\mathrm{A}_p^* = \left\{ a \in \mathrm{A}_p \mid \phi_p = \phi_a \frac{\gamma_{a,p}}{\gamma_a} \right\} \quad . \tag{1.13}$$

The *critical saturation* of signal phase p is obtained by applying (1.11) to its critical lane group:

$$\chi_p = \frac{q_p}{\gamma_{\mathrm{A}_p^*}} \quad , \tag{1.14}$$

noting that in the particular case where each lane group is only open during a single phase, critical saturation occurs on the one registering the highest flow ratio.

Since different lane groups may experience different effective green shares, should be calculated using the effective green experienced by the same lane group during that phase, which is practically considered the *phase effective green*:

$$g_p = g_{\mathrm{A}_p^*,p} \quad . \tag{1.15}$$

Finally, the total *junction flow ratio*, which gives a measure of how busy the intersection really is, can be calculated as the sum of the critical flow ratios over all phases of the signal cycle:

$$\phi_j = \sum_{p \in \mathrm{P}_j} \phi_p \quad . \tag{1.16}$$

Lost Time

Driver reactions are not instantaneous, and vehicles take a finite amount of time to accelerate and clear the junction. This implies that a non-negligible share of the signal cycle goes wasted, since demand is not served efficiently during the phase transitions:

- at every phase start, a few seconds pass before vehicles can flow at full capacity, causing a *start-up time loss*;
- at every phase end, sufficient time must be allowed for vehicles to clear the junction before others may safely carry out a conflicting manoeuvre, which represents a *clearance loss*.

The start-up loss may be reduced by helping drivers to react more promptly, e.g. using a pre-green amber light or red count-down timers, which also seem to alleviate the stress of being stuck in a queue. The clearance loss may only be mitigated by an accurate choice of signal phase sequence for given traffic conditions or, wherever possible, by appropriate modification of the junction layout, e.g. implementation of protected turn bays.

The total lost time t_j^L then depends on phase design and sequence, which in turn should be tailored to the geometry of junction j in relation to the expected traffic conditions. Each phase contributes its own time losses $t_{j,p}^L$ to the total lost time, which may be quantified by the following relation between the effective phase green and the phase duration:

$$t_p^L = t_p - g_p \quad . \tag{1.17}$$

The total time loss and the total effective green thus account for the whole signal cycle period:

$$t_j^C = t_j^L + \sum_{p \in \mathrm{P}_j} g_p \quad . \tag{1.18}$$

1.3.2 Formulation of the Signal Setting Problem

Conflicting sets of manoeuvres compete for the right of way at road intersections, and the main purpose of signalization is to distribute the junction capacity amongst them.

It follows naturally that the allocation of green time to signal phases is the single most important step in signal setting: the cycle must be allotted according to the relative distribution of demand, lest the junction capacity go wasted and unnecessary queues form on critical approaches.

As far as fixed timing is concerned, optimal allocation of green time is a straightforward process, yet it can be undertaken according to a number of different principles: early studies aimed to develop analytical equations, while modern simulation based methods rely on heuristics to shape the signal setting around a cost function that formalises the chosen signal setting policy. The next sections provide a general formulation of the problem and a few examples of objective implementation through different setting policies.

Lagrangian Formulation

The Signal Setting of junction j can be formulated as an optimisation problem, i.e. to find effective green durations for each phase and cycle time that minimise an objective function while complying with a set of constraints.

A popular choice of cost function may be the average delay at the intersection, given by the weighted average vehicle delay ω_a^D on all lane groups.
Delay on each lane depends according to equation (1.10) on effective green shares, cycle length, and the relevant flows q_a as illustrated in section 1.3.1.

For average delay optimisation of a junction j, consider a well-designed phase sequence P_j ensuring minimal conflicts and time losses. The signal program is then fully characterised by a vector of effective phase green shares $\mathbf{g}_{\mathrm{P}_j} \in \mathbb{R}^{|\mathrm{P}_j|}$ together with the cycle time t_j^C.

The problem takes the following form:

$$\begin{aligned} \underset{\mathbf{g}_{\mathrm{P}_j},\, t_j^C}{\mathbf{min}} \qquad & \omega_j^D = \sum_{a \in \mathrm{A}_j^-} \omega_a^D q_a \\ \text{subject to} \quad & t_j^C - t_j^L = \sum_{p \in \mathrm{P}_j} g_p \\ & g_p \geqslant \phi_p \cdot t_j^C \quad \forall p \in \mathrm{P}_j \end{aligned} \tag{1.19}$$

where the first constraint simply enforces the conservation of cycle time, and the other $|\mathrm{P}_j|$ inequalities ensure that phase durations are sufficient to meet demand where possible. It is therefore evident that the number of constraints is equal to the number of variables, but since green time constraints are inequalities the problem has as many degrees of freedom as the signal program phases. This is solved by introducing a vector of positive auxiliary variables so that for each phase p of the program

$$g_p - \phi_p \cdot t_j^C - \eta_p^2 = 0 \quad . \tag{1.20}$$

The optimisation can then be solved with the Lagrange method, i.e. finding the stationary points of the linear combination of objective function and equality constraints. To this end, Lagrangian multipliers are introduced as auxiliary variables λ_p for the phase constraints and μ for the cycle total, forming the Lagrangian function

$$\mathscr{L} = \omega_j^D \left(\mathbf{g}_{\mathrm{P}_j}, t_j^C \right) + \sum_{p \in \mathrm{P}_j} \lambda_p \left(g_p - \phi_p \cdot t_j^C - \eta_p^2 \right) + \mu \left(\sum_{p \in \mathrm{P}_j} g_p + t_j^L - t_j^C \right) \quad . \tag{1.21}$$

Since by definition all partial derivatives of the Lagrangian function must be zero at stationary points, the following are obtained for each control and auxiliary variable:

$$\begin{aligned} \frac{\partial \mathscr{L}}{\partial t_j^C} &= \frac{\partial \omega_j^D}{t_j^C} - \sum_{p \in \mathrm{P}_j} \lambda_p \cdot \phi_p - \mu = 0 && && \text{(a)} \\ \frac{\partial \mathscr{L}}{\partial g_p} &= \frac{\partial \omega_j^D}{g_p} + \lambda_p + \mu = 0 && \forall p \in \mathrm{P}_j && \text{(b)} \\ \frac{\partial \mathscr{L}}{\partial \lambda_p} &= g_p - \phi_p \cdot t_j^C - \eta_p^2 = 0 && \forall p \in \mathrm{P}_j && \text{(c)} \\ \frac{\partial \mathscr{L}}{\partial \mu} &= \sum_{p \in \mathrm{P}_j} g_p + t_j^L - t_j^C = 0 && && \text{(d)} \\ \frac{\partial \mathscr{L}}{\partial \eta_p} &= \eta_p \cdot \lambda_p = 0 && \forall p \in \mathrm{P}_j && \text{(e)} \end{aligned} \tag{1.22}$$

This non-linear system can be readily examined without loss of generality by referring to a simple case of a junction j with only two phases, p and q. The complementarity conditions $\eta_p \cdot \lambda_p = 0$ and $\eta_q \cdot \lambda_q = 0$ then have four possible solutions, and an optimal solution must be identified by comparing the objective function for all candidate stationary points.

Case A - minimum cycle: $\lambda_p, \lambda_q \neq 0\,;\ \eta_p, \eta_q = 0$
Lagrangian multipliers are inconsequential because capacity constraints (1.22) (c) are active during both phases, and with (1.22) (d) are sufficient to determine effective greens and cycle time:

$$\begin{cases} g_p = \phi_p \cdot t_j^C \\ g_q = \phi_q \cdot t_j^C \\ t_j^C = t_j^L + (g_p + g_q) \end{cases}$$

has a unique solution, corresponding to the minimum green times that exactly match demand on the critical lane groups of each phase (i.e. all critical lanes operate at full saturation $\chi_a = 1$) and the subsequent cycle length that allows to serve demand despite the time wasted during phase changes. Each phase should get at the very least a green share of the total cycle time equal to its critical flow rate, hence

$$t_j^C = t_j^{Cmin} = \frac{t_j^L}{1 - \phi_j} \quad . \tag{1.23}$$

Case B - main phase, secondary phase: $\lambda_p = 0,\ \eta_p \neq 0\,;\ \lambda_q \neq 0,\ \eta_q = 0$
Capacity constraint is active only during phase p, while the other Lagrangian multiplier is zero. By solving the equation for the active constraint, the minimum green g_p is found; the other can similarly be determined as a function of the cycle time, reducing the objective to a single variable function. In the case of deterministic constant arrivals, minimisation of delays yields a unique stationary point in terms of cycle length, which can be found in closed form:

$$t_j^C = t_j^{Cdet} = \sqrt{\frac{\rho_q \left(t_j^L\right)^2}{\rho_p \left(1 - \phi_p\right)^2 + 2\rho_q \phi_p^2}} \quad , \tag{1.24}$$

where the ρ coefficients are determined for each phase as

$$\rho_p = \frac{1}{2} \cdot \frac{\sum_{a \in \mathrm{A}_p} \frac{q_a}{1 - q_a}}{\sum_{a \in \mathrm{A}_j^-} q_a} \quad . \tag{1.25}$$

This solution assigns minimum green to phase p and is reasonable if the major demand flow is handled during phase q, while the opposite scenario covers the specular case.

Case C - low saturation: $\eta_p, \eta_q \neq 0\,;\ \lambda_p, \lambda_q = 0$
In this case no capacity constraints are active. The problem consists in searching for the solution of the following system of three non-linear equations in the three unknowns g_p, g_q and t_j^C:

$$\begin{cases} \frac{\partial \omega_j^D}{\partial g_p} + \frac{\partial \omega_j^D}{\partial t_j^C} = 0 \\ \frac{\partial \omega_j^D}{\partial g_q} + \frac{\partial \omega_j^D}{\partial t_j^C} = 0 \\ g_p + g_q - t_j^C + t_j^L = 0 \end{cases} \quad . \tag{1.26}$$

In this case the analytical solution is not as straightforward, but a solution can be easily found numerically since the delay function is bounded and convex under usual realistic assumptions

on vehicle arrivals. The explicit formulation of this method rests upon the assumption that the capacity condition be respected, i.e. the capacity of the junction be sufficient to serve demand. This may be relaxed in practice for heuristic optimisation, but little changes about the fundamental fact that no green share allocation will ever enable a junction to operate *above capacity* without delay.

Webster Optimal Solution

The first and foremost formulation of optimal signal settings to lift the assumption of uniform vehicle arrivals is due to Webster (1958). The approach is based on a queueing system with Poissonian arrivals and a constant service rate equal to the capacity $\gamma\hat{q}$ of the signalised lane group. The average delay given for the steady state case by equation (1.10) was extended to obtain a more complete delay function for random arrivals, with an additional empirical term needed to improve the fit with *experimental* observations.

To simplify the optimisation problem, a reasonable green share allocation policy (widely known as *Equisaturation Policy*) was chosen. This revolves around the idea that an equitable distribution of green share is obtained when all critical manoeuvres operate at the same saturation level: the higher the demand for a manoeuvre *with respect to the capacity* of the relevant infrastructure, the higher the green share allocated to the corresponding signal phase.

Furthermore, Webster worked under the assumptions that no over-saturation occur and average demand *flows* are stable, i.e. and path choices made by road users are in no way a consequence of the signal setting.
Under the equisaturation policy, all phase saturation levels at a given junction are equal by definition. The *available green time* can simply be allocated proportionally to the critical flow ratio of each phase:

$$\gamma_p = \frac{\phi_p}{\phi_j} \frac{t_j^C - t_j^L}{t_j^C} \quad \forall p \in \mathrm{P}_j \tag{1.27}$$

which yields meaningful results provided that the junction total flow ratio does not exceed its maximum value of 1 and the cycle time is sufficiently long to amortise the lost time.

The approach can be extended to design for specific (not necessarily even) saturation values for each phase by rearranging equation (1.11) and solving for the green share: this may have practical sense in order to design a higher tolerance to high arrival rates into a given phase e.g. if it is strategically more important to keep queues at a minimum on a certain set of lanes than it is elsewhere.

With this green share setting policy in place, the problem of minimising the average delay is reduced to a single variable function of the cycle length.
The resulting solution for the cycle time that minimises average delay under probabilistic arrivals is rather complex and was approximated it through an empirical formula, widely known as the Webster optimum cycle time:

$$t_j^{C,Webster} = \frac{\frac{3}{2}\, t_j^L + 5}{1 - \phi_j} \quad . \tag{1.28}$$

Notice from equations (1.23) and (1.28) how the cycle time invariably grows with the total flow ratio of the junction. It is also possible to extend (1.23) to get a target saturation

level χ_j for the junction:

$$t_j^C(\chi_j) = \frac{t_j^L}{1 - \frac{\phi_j}{\chi_j}} \quad , \tag{1.29}$$

or even a vector $\vec{\chi}_{\mathrm{P}_j}$ of critical saturation level values each phase, as in

$$t_j^C(\vec{\chi}_{\mathrm{P}_j}) = \frac{t_j^L}{1 - \sum_{p \in \mathrm{P}_j} \frac{\phi_j}{\chi_j}} \quad . \tag{1.30}$$

It should be evident that saturation values greater than 1 correspond to *oversaturated* conditions, under which the demand flows are not met with sufficient capacity and queue buildup is inevitable: such traffic conditions require radically different timing approaches. The rule of thumb mentioned in the Highway Capacity Manual [Special Report 209, 1985] and generally followed in practice is that signals should be timed so that lanes operate at saturation levels below 0.85, allowing sufficient margin to deal efficiently with most possible traffic fluctuations, and discharge any queues within a few signal cycles.

1.4 Signal Coordination

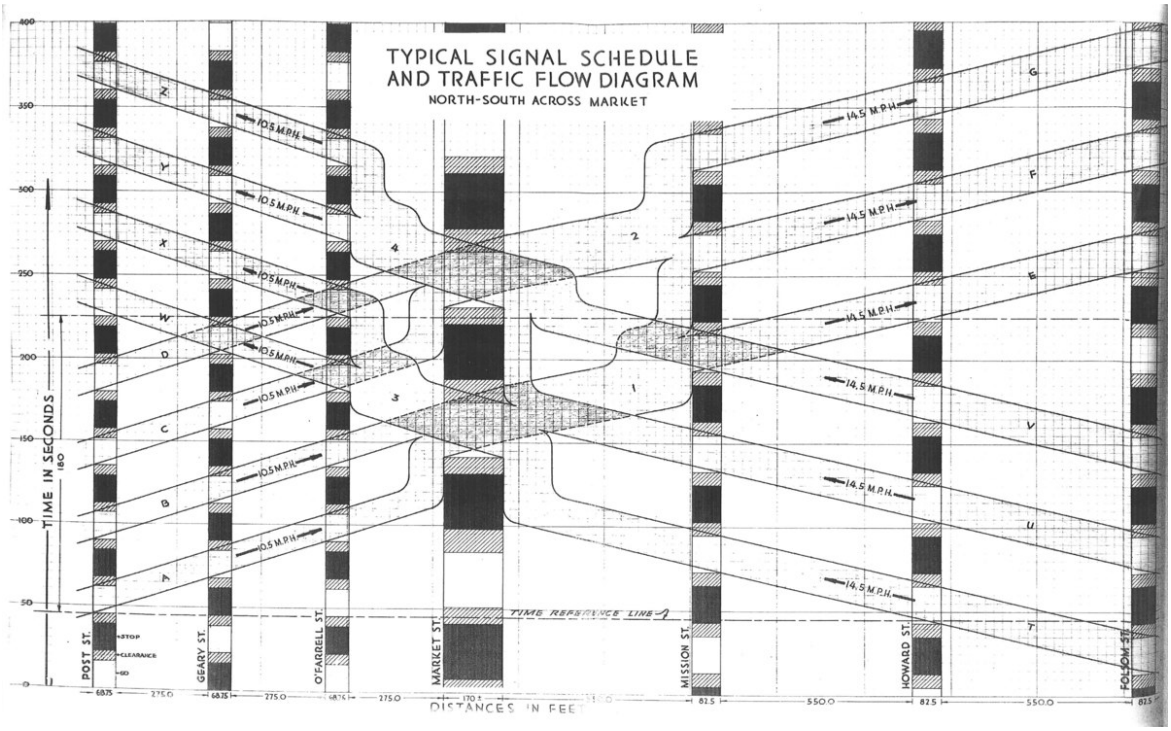

FIGURE 1.4 – Early signal synchronisation along a San Francisco arterial road, circa 1929. Bands A through T represent vehicle platoons [1].

Traffic light coordination between adjacent junctions is an essential aspect of an optimal signalisation plan, with disposition of *green waves* as its most notable and popular feature. Traffic in fact mostly travels along a limited number of main corridors, commonly referred to as *arteries* carrying *arterial traffic*.

[1] By City of San Francisco - Public domain (via Eric Fischer), CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=34715929

It has long been accepted as a reasonable compromise to minimise user discomfort along those, rather than taking on the much more intricate problem of reducing the total network delay. Although, undeniably, being able to drive through a streak of green signals already goes a long way towards improving the quality of a trip from the user point of view, signal coordination chiefly serves the purpose of ensuring an efficient use of the available infrastructure.

It is in fact of the utmost importance to avoid unnecessary signal-induced delays and stops which could rapidly bring traffic to a grinding halt, even under rather mild conditions which the network could otherwise cope with.

The search for a coordination solution that maximises usability of urban arteries under specific traffic conditions is still mostly carried out offline — as it was for the first attempts at smart arterial signalization, such as the pen-and-paper method portrayed in Figure 1.4. To this end, a wide variety of methods have been the object of intensive research since the early 1980s, ranging from simple analytical approaches to heuristics.

Analytical methods have brought about a number of popular applications which are still in use despite the fact that they mainly apply to low congestion scenarios; more complex methods, which account for demand flows and their propagation along the arterial, can deal with heavy congestion related phenomena, but invariably require a more detailed network model and rely on computationally demanding simulations rather than a closed-form problem formulation. An overview of the most prominent approaches to the signal coordination problem is given in the following sections.

1.4.1 The Traffic Corridor

The fulcrum of signal coordination is the *traffic corridor* (i.e. an arterial road, as defined in the previous section) selected for its strategic relevance. Since the flow on the corridor is supposedly much higher than on its side roads, it is deemed acceptable to concentrate optimisation efforts on the arterial traffic conditions, as improvements will benefit the largest number of road users.

A traffic corridor C may be defined as an *ordered* set of n *connected* arcs:

$$\mathrm{A} \supset \mathrm{C} = \{a_1, a_2, \ldots, a_n\} \qquad \text{with} \quad \begin{cases} a_{i-1} \in \mathrm{A}^-_{a_i} & \forall i > 1 \\ a_{i+1} \in \mathrm{A}^+_{a_i} & \forall i < n \end{cases} . \tag{1.31}$$

Although all nodes along the corridor are, strictly speaking, junctions, it makes sense in this context to define the ordered subset $\mathrm{J_C}$ of the m *signalised* junctions that actually regulate the flow on the corridor. This may be formalised as

$$\bigcup_{a \in C} \{\mathrm{N}^-_a, \mathrm{N}^+_a\} \supset \mathrm{J_C} = \{j_1, j_2, \ldots, j_m\} \quad \text{such that} \quad \forall j \in \mathrm{J_C} \quad \exists y \in \mathrm{P}_j | \{\mathrm{A}^-_y, \mathrm{A}^+_y\} \subset \mathrm{C} \tag{1.32}$$

where it is simply stated that a corridor node is considered a relevant *signalised junction* if features one *signalised* manoeuvre $y \in \mathrm{P}_j$ whose origin and destination lanes $\{\mathrm{A}^-_y, \mathrm{A}^+_y\}$ both lie on the corridor (with the exception of the first node of the corridor, which may be included in $\mathrm{J_C}$ as long as it regulates at least one turn onto the corridor, and the last one if the corridor outflow may be affected by its signal).

Coordination of junctions $\mathrm{J_C}$ is handled by offsetting their local timing instructions (as described at the end of section 1.2), i.e. anticipating or delaying all phase changes rigidly

without altering the necessary green shares determined on the basis of average demand flows. The global offset values (with respect to an arbitrary global time reference) of the junctions of corridor C may be represented by a vector $\mathbf{t}_{\mathrm{C}}^{\mathrm{O}}$.

Furthermore, it is assumed that all junctions of the corridor share the same cycle time, so that in the context of signal coordination the symbol t_{C}^{C} refers to all junctions, and may be even used without the subscript C.

1.4.2 Bandwidth Maximisation

In relation to arterial traffic, the concept of *progression bandwidth* emerges as a measure of the quality of a green wave setup along a *corridor* and can be defined as the duration of the time window through which a vehicle may enter the artery and travel its entire length without encountering red lights nor standing queues.

By reducing delays and number of stops along the most critical paths, bandwidth maximisation is a relatively straightforward but effective way to help the system meet user expectations about traffic fluidity, mitigating the stress associated with driving in a congested urban environment. Moreover, this type of signal coordination has proven highly beneficial in reducing the chance of rear end collisions and red signal violations [Li and Tarko, 2010] as well as pollution levels associated with the hiccupping stop-and-go driving often experienced under poorly coordinated signalisation.

Bandwidth maximisation has been formulated as a Linear Optimisation problem since [Little, Kelson, and Gartner, 1981] which led to development of the MAXBAND/MULTIBAND series of software solutions. These considered the offsets between junctions as the only decision variables, but provided a computationally viable method for one-way and two-way bandwidth maximisation relying solely on the target travel times between junctions and predetermined signal cycle length and green times.

However, relevant discrepancies — dubbed *bandwidth degradation* — were observed between the expected outcome and the real-world performance of the signal plans generated by these early methods: it is now universally accepted that, as [Tsay and Lin, 1988] amongst many others pointed out, the underlying models were oversimplified and no account was taken of side flows and platoon dispersion.

Proposed extensions of the original method aimed to factor in queue and side flow clearance times, to produce a more realistic bandwidth model for phase offset determination. The analytical relationship between maximal bandwidth and minimum delay problems was finally formalised in [Papola and Fusco, 2000], where travel times and delays are expressed as a function of the maximal bandwidth and other variables accounting for the entity of side flows, interstage sequences etc.

At present, offline arterial progression optimisation techniques invariably rely on some formulation of the *bandwidth maximisation problem* (as in the cases illustrated in the next section), which is to say that their common objective is to maximise a *theoretical* traffic throughput, often without much consideration for network performance. This is also true for *online* optimisation tools that evaluate signal plan updates with a similar goal, ignoring the fact that traffic propagation is a rather complex phenomenon which has the utmost relevance upon bandwith degradation: as explained in detail in Chapter 4, one of the foremost aims of this work is to renounce the geometric formalisation of bandwidth as a measure of progression

in favour of a simulation based approach, to better reproduce the relation between coordination and queue dynamics, and possibly look past the long standing preconception that to maximise progression identifies with optimal operation conditions for any road under any circumstance.

The next sections illustrate two numerical approaches to the complex problem of two-way bandwidth maximisation: the first is an elegant implementation of the classic paradigm of progression optimisation, fully featured in closed form and solved as a linear program with the addition of variable speed limits; the second is a much simpler yet effective geometrical method developed in the context of this work.

Mixed Integer Linear Programming Approach

One of the most complete and effective implementations of the MILP approach to two-way arterial coordination is presented in [De Nunzio, Gomes, de Wit, Horowitz, and Moulin, 2015]. The bandwidth maximisation was extended to include *Variable Speed Limits* (VSL) as control variables, allowing for a wider range of high bandwidth solutions. The method is outlined here as an example of the degree of complexity that can be managed by mathematical optimisation.

The concept of VSL has been applied to motorway traffic for quite some time, to enhance traffic fluidity in response to congestion, accidents or adverse weather, but its application to urban traffic presents new challenges, not least the need for effective means of introducing it and getting it across to the drivers: in pilot projects this is quite effectively achieved by variable led panels, mimicking an ordinary speed limit sign, showing the target synchronisation speed. Were such measures to gain popularity, the already promising degree of driver compliance can only be expected to improve.

The simple MILP approach to two-way bandwidth maximisation can be summarised by considering a corridor C (see section 1.4.1) running through an ordered set of intersections $\mathrm{J_C}$ along its *main* driving direction, while the opposite, possibly lower priority direction traverses the same nodes in reverse order.

If positive travel speeds v_j and $\bar{v}_j$ are defined between j and $j+1$, in the main and return direction respectively, and a generic spatial coordinate x is considered, increasing along the corridor with the index j, travel times for perfect green waves should then be:

$$\begin{cases} t_j = \frac{x_{j+1}-x_j}{v_j} > 0 \\ \bar{t}_j = \frac{x_j-x_{j+1}}{\bar{v}_j} < 0 \end{cases} \qquad \forall j \in [1\,, |\mathrm{J_C}| - 1] \quad . \tag{1.33}$$

Assuming a common cycle time t^C_{C}, consider at each node and for both directions:

effective green duration of the arterial *through* movement phases g_j and $\bar{g_j}$;

absolute offset as the time between the midpoint of a green phase and the closest multiple of the cycle time $t^O_j, t^{\bar{O}}_j$.

A nonstandard modulo operation $\|\bullet\|_t$ can be defined for brevity, to refer *any* time to the corridor cycle, such that

$$\|t^*\|_t \in \left]-\frac{t^C}{2}, \frac{t^C}{2}\right] \tag{1.34}$$

returns the distance from t^* to the nearest multiple of t^C.

The modulo is used to define the *internal offset* given by

$$t_j^\delta = \left\| \bar{t_j^O} - t_j^O \right\|_t \tag{1.35}$$

and the *relative offset* t_j^Δ. The latter represents the time coordinate of the mid-green instant of the relevant phase with respect to a *moving* frame of reference travelling along the *main* driving direction, starting in x_1 at the zero instant and moving with the specified speeds v_j between nodes.

Hence, the relative offset at each node after the first can be computed easily from the offsets at upstream nodes:

$$\begin{aligned} t_j^O - t_j^\Delta = t_{j-1}^O - t_{j-1}^\Delta + t_{j-1} \quad &\Rightarrow \quad t_j^\Delta = \left\| t_{j-1}^\Delta + t_j^O - t_{j-1}^O - t_{j-1} \right\|_t \\ &\Rightarrow \quad \begin{cases} t_j^\Delta = \left\| t_1^\Delta - t_1^O + t_j^O - \sum_{i=1}^{j-1} t_i \right\|_t \\ t_j^O = \left\| t_1^O - t_1^\Delta + t_j^\Delta + \sum_{i=1}^{j-1} t_i \right\|_t \end{cases} . \end{aligned} \tag{1.36}$$

In order to express the bandwith in both directions in terms of the relative offsets, it is also beneficial to map all t_j^δ to the time reference of the first junction using

$$t_j^{\delta 0} = \left\| t_j^\delta + \sum_{i=1}^{j-1} (t_i - \bar{t}_i) \right\|_t , \tag{1.37}$$

and considering that the t_j^δ are described by the signal program at each intersection, which leads to the vector equation linking the offsets in the two directions

$$\bar{\mathbf{t}}^{\mathbf{\Delta}} = \mathbf{t}^{\mathbf{\Delta}} - \mathbf{t}^\delta \qquad \text{with} \begin{cases} \mathbf{t}^{\mathbf{\Delta}} = \left(t_1^\Delta, t_2^\Delta, \ldots, t_{|\mathrm{J_C}|}^\Delta \right) \\ \mathbf{t}^\delta = \left(t_1^{\delta 0}, t_1^{\delta 0} - t_2^{\delta 0}, \ldots, t_1^{\delta 0} - t_{|\mathrm{J_C}|}^{\delta 0} \right) \end{cases} . \tag{1.38}$$

Finally, it is possible to express bandwidth as a function of travel times and offsets. According to the definition given at the start of this section and considering Figure 1.5, it is the intersection of all green windows as seen in the moving frame of reference:

$$\bigcap_{j \in \mathrm{J_C}} \left[t_j^\Delta - \frac{g_j}{2} \, , \, t_j^\Delta + \frac{g_j}{2} \right] . \tag{1.39}$$

The bandwidth value in the *main* direction is then calculated from the decision variables as

$$\omega_{\mathrm{C}}^B = \omega^B \left(\mathbf{t}^{\mathbf{\Delta}} \right) = \min \left\{ \left(t_i^\Delta - t_j^\Delta + g_{ij} \right) \quad \forall i, j \in \mathrm{J_C} \right\} \qquad \text{with } g_{ij} = \frac{g_i + g_j}{2} \tag{1.40}$$

which is the *smallest possible* overlap between *any two* green phases in the moving FoR; the equivalent in the other direction is found using the relevant green times $\bar{g}$ and the relation given by (1.38) .

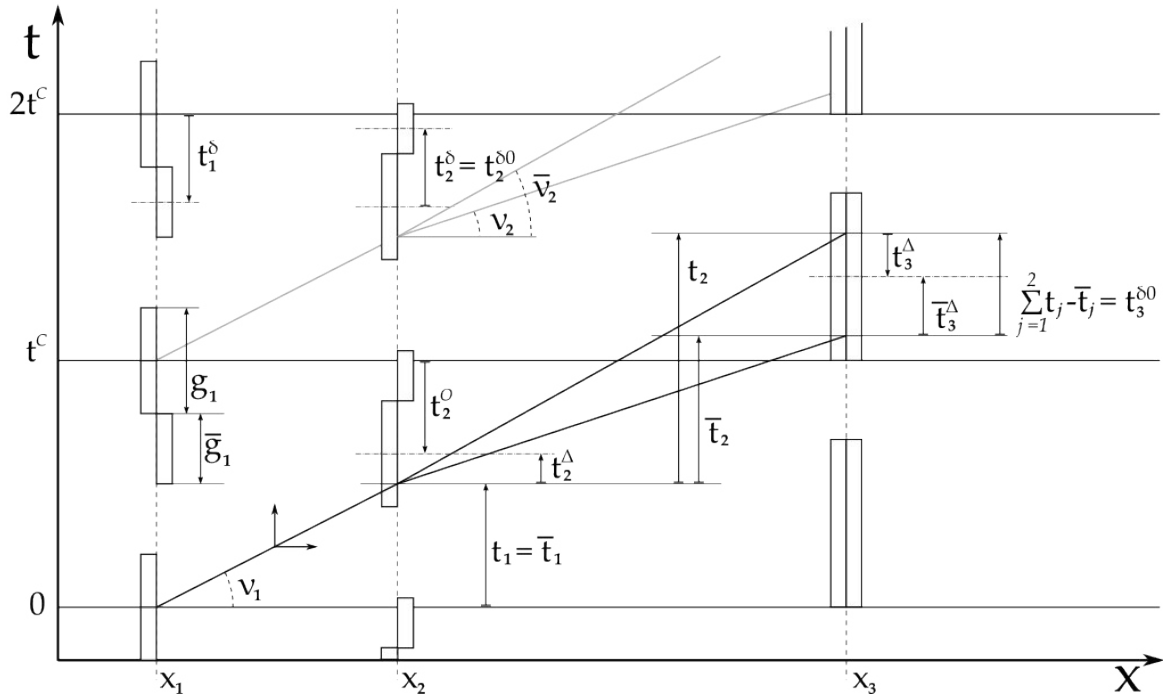

FIGURE 1.5 – Bandwidth Problem Formulation: the signal coordination parameters are portrayed on a distance-time (D-T) graph. Temporal references are given by integer multiples of the cycle time and by the synchronisation frame of reference, moving along the diagonal trajectories at speeds v_j. The green phase in the main direction is drawn on the left of each junction's temporal line, that of the inverse direction is to its right. Notice the offsets measured between the phase midpoints and the time of arrival of the moving FoR.

The sum of the bandwidths in the two directions can then be the objective of the linear optimiser — bounded by appropriate constraints such as maximum speed values — in conjunction with any function of the decision variables used to favour a certain type of solution: for example, the optimisation presented in [De Nunzio et al., 2015] is driven by an extended utility function aiming to favour low travel times and minimise the speed indication variance across segments so as to ease drivers into complying with apparently arbitrary limits.

Real world statistics are beginning to back up the simulation results that originally validated these studies, proving the following interesting points about modern bandwidth maximisation techniques:

- the best combinations of optimal offsets and VSL drastically reduce the number of stops and energy consumption;
- lower and smoother speed limits reduce energy consumption at *no disadvantage* to the total arterial travel time;
- VSL brings about larger bandwidth and faster solution of the LP.

It must be noted however that despite the practically negligible computation times associated with the LP methods just mentioned, these remain conceptually unfit for *real time* signal optimisation since they take in no account the flow and speed of the actual traffic, nor they apply outside the safe boundaries of *capacity* conditions (whereby green time is always assumed sufficient to deal with demand).

1.4.3 The Slack Band Approach

The idea of *slack* bandwidth is an answer to the very strict definition of bandwidth given at the beginning of section 1.4.2, according to which only the band running through all junctions counts for something, implying that:

- if one passing phase is particularly short, coordination between longer green phases may be disregarded: because of (1.40) the bandwidth is throttled to be at most as wide as the shortest phase;
- in bi-directional optimisation, maximisation of the return band may prioritise a very narrow band that *just* makes it through all junctions (possibly degrading the main band significantly) over a very wide band divided in two or more chunks.

To avoid such inconveniences, which are intrinsic in the definition of what will henceforth be referred to as the *canonical* bandwidth ω_{C}^{B}, the slack bandwidth paradigm attempts to describe the overall *"progressivity"* of the corridor along its whole length, by considering the sum of the individual green bands leading up to and following *any* of the corridor junctions.

Rather than a length of time (the width of ω_{C}^{B} on a T-D graph) the slack bandwidth has dimensions $[\mathrm{L} \cdot \mathrm{T}]$ (an area on a T-D graph) i.e. it is the product of the time during which a vehicle may enter each section of the corridor and the distance it will travel unhindered as a result. If the times are normalised with respect to the cycle time, the slack band becomes a *probability* $\times$ *distance* product (just as the canonical bandwidth would represent the overall chance of travelling the whole corridor without stopping).

The formalisation of this idea is much simpler than it may sound at first. Consider a junction j somewhere along corridor C: the *forwards slack progression band* ω_j^{b+} is the integral of: the distance l_j that may be travelled without stopping, with respect to the time t at which a vehicle leaves from j; the former obviously a function of the latter which may be expressed as $l_j(t)$.

Using a compact definition of the *through* phase at j as the interval during which the corresponding manoeuvres are open

$$G_j = \left[t_j^g \, , \;\; t_j^g + g_j \right] \quad , \tag{1.41}$$

where t_j^g is the beginning time of the through phase at j and g_j its duration, as before, the forward band can be expressed as

$$\omega_j^{b+} = \int_{G_j} l(t) \, \mathrm{d}t \quad . \tag{1.42}$$

The integral in (1.42) clearly formalises the definition of slack bandwidth illustrated in Figure 1.6.a but it's not practical to compute, and does not provide an explicit form for $l(t)$.

Consider therefore the interval G_{ji} during which a vehicle that *left* j during G_j may drive through a *subsequent* junction $i \geqslant j$. As the vehicles progress along the corridor, only the ones that reach each junction i during the corresponding through phase G_i can proceed without stopping.

Now, the interval over which vehicles that left i during G_i reach $i+1$ can be expressed as

$$G_i^+ = [\inf G_i + t_i \, , \, \sup G_i + t_i] \quad . \tag{1.43}$$

with a simple forward translation to account for the travel time t_i of the relevant corridor section, whence the passing band can be shown to gradually narrow down by intersection with each subsequent green phase

$$G_{j\,i} = G_i \cap G^{+}_{j\,i-1} \quad . \tag{1.44}$$

Finally, using $|G|$ to indicate the length of a passing interval, the forward slack band can be calculated recursively from any junction j to the end of the corridor:

$$\omega^{b+}_{j} = \sum_{i\mathrm{C}\geqslant j} |G_{ji}| \cdot \ell_i \quad , \tag{1.45}$$

considering that at the reference junction the passing interval *is* the green phase $G_{j\,i=j} = G_j$.

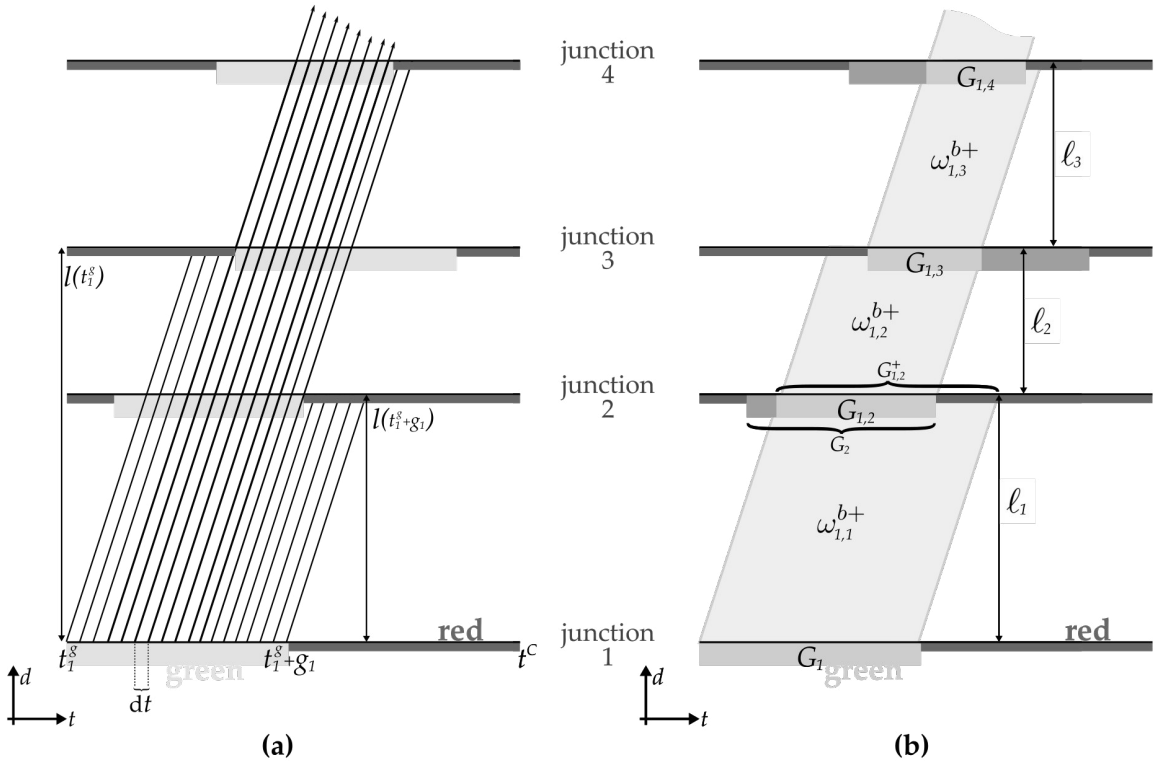

FIGURE 1.6 – The *forward slack band* definition on T-D diagrams expressed in **(a)** as the integral (1.42) and in **(b)** as the discrete sum (1.45) .

Plainly following the specular process back to the beginning of the corridor, it is possible to calculate the *backwards slack progression band* ω^{b-}_{j}, to quantify the chances of a vehicle reaching junction j unhindered by red lights. It is also plain that the subsequent applications of (1.44) may well yield an empty intersection before the end or the beginning of the corridor are reached: this is not an issue, as the value of this *something-is-better-than-nothing* approach lies exactly in the ability to consider *any* length that can be travelled without stopping. Some computation time can be saved by checking for this condition and stopping the recursive process (1.45) as soon as all vehicles that left j have stopped at i (or, going the other way, as soon as none of the vehicles leaving i will reach j without stopping).

The *total* slack band for a given corridor is the normalised sum of the forwards and backwards

bands calculated at each junction:

$$\omega_{\mathrm{C}}^{b} = \frac{1}{|\mathrm{J_C}|} \sum_{j \in \mathrm{C}} \omega_j^{b-} + \omega_j^{b+} \quad . \tag{1.46}$$

With the formalisation complete, it is worth noting the following aspects about the new metric, also illustrated in Figure 1.7:

- normalisation implies that the method favours letting vehicles onto *longer* arcs, which maximise the product in (1.45), rather than short ones, which are more vulnerable to spillback;
- if a perfect, continuous green wave can be obtained along the whole corridor, the result is identical to the canonical bandwidth value multiplied by the length of the corridor $\omega_{\mathrm{C}}^{b} = \omega_{\mathrm{C}}^{B} \cdot \ell_{\mathrm{C}}$ (this requires that all green phases also have the same length);
- in all other cases, the slack band value is *strictly greater* than $\omega_{\mathrm{C}}^{B} \cdot \ell_{\mathrm{C}}$ as it factors in all fringes and partial bandwidths that (1.39) necessarily excludes, which is the main point of this metric.

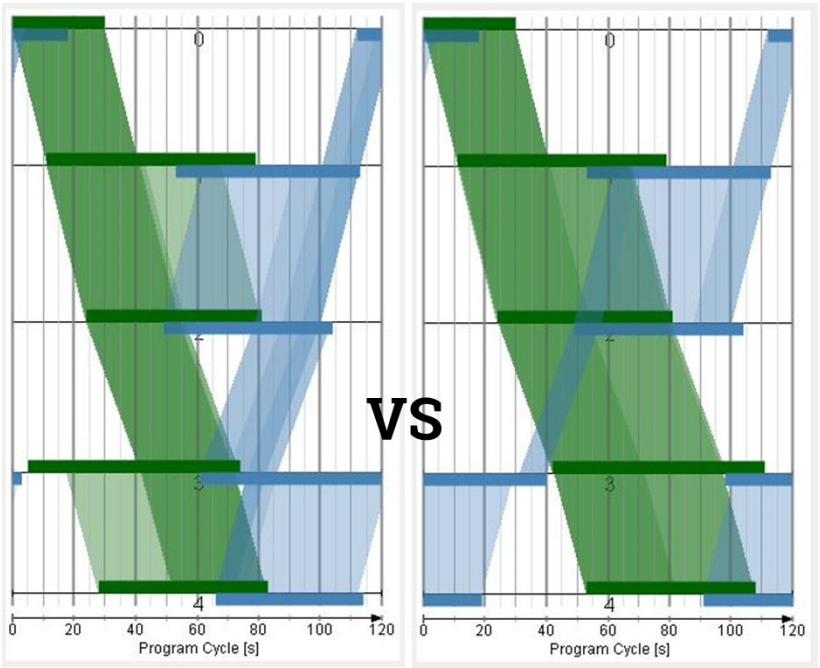

FIGURE 1.7 – A comparison of the results obtained by optimising the canonical progression bandwidth (left) and the slack band metric (right) on T-D diagrams.
Bands crossing from the top-left to the bottom-right are travelling in the main direction, the others are in the secondary direction, going through the junctions in reverse order. Darker bands on the slack band diagram are in common between a higher number of junctions, the darkest corresponding to the canonical bandwidth and all others to the fringes and partial green waves that the new metric allows to weigh in.

With given offsets and signal programs, the computation of this metric is almost instantaneous for any conceivable real-world traffic arterial, allowing to find an optimal solution in seconds using a stochastic search method as described in section 3.4.2. Its effectiveness confirmed by the results presented in 6.3, this method was used to find ideal initial conditions for the real-time traffic coordination module presented in this work.

1.5 Advanced Offline Signal Planning

The simple signal setting problems presented so far are quasi-convex, but more realistic traffic models that include and quantify global performance indicators such as total delay introduce an inherent non convexity, better addressed with the aid of heuristic methods.

With the increase in computing power availability, metaheuristics have seen a substantial rise in popularity as means to overcome the inherent limitations of analytical formulations: heuristic approaches to this class of problems involve the generation of a large — yet manageable, compared to the dimensions of the search space — number of candidate timing solutions, the effects of which are then simulated to evaluate their fitness. At each iteration, a variety of methods ranging from Genetic Algorithms to Simulated Annealing and Particle Swarm Optimisation can then be used to modify and combine the most successful solutions into a new set of candidates.

Such methods are particularly suited for solving obscure problems as they require no attempt to establish an explicit correlation between the control variables and the desired outcome. Rather, they rely on the assumption that if any relevant phenomena can be modelled with sufficient accuracy and a performance index can describe the degree of achievement of the optimization objectives, then the system can be led to evolve towards an optimal solution.

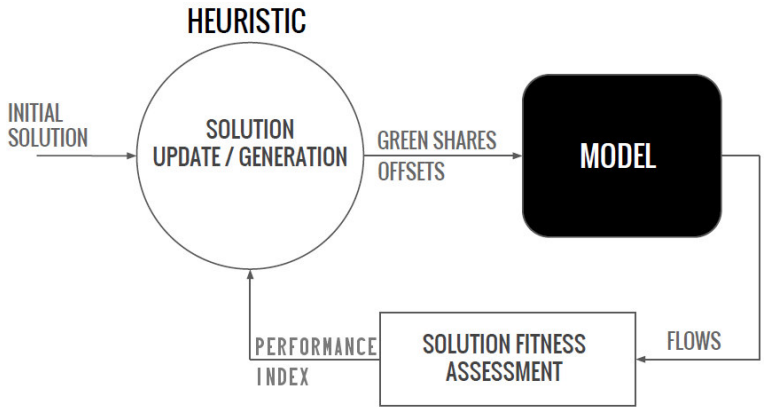

FIGURE 1.8 – Conceptual information flow in a heuristic approach to signal optimisation

It is therefore obvious that the model used to assess the fitness of candidate solutions should represent a sensible trade-off between speed and completeness: the real-world performance will inevitably be disappointing if the optimisation does not account for relevant traffic phenomena that were simplified out of the solution assessment, while on the other hand the need to evaluate huge numbers of candidate solutions calls for a lean and fast method to predict the outcome of a given timing plan. Furthermore, heuristics that depend heavily on the choice of initial conditions often use maximum bandwidth solutions as starting point in the search for minimum total delay, to shave off convergence time and increase the quality and applicability of solutions. The present study takes full advantage of both features.

The heuristic optimisation approach has been taken most notably by the Transport Research Laboratory, the UK based institution that since [Robertson, 1969] has been developing the TRAffic Network StudY Tool, which was born as a software tool to minimise stops along arterial roads while accounting for reasonably realistic vehicle behaviour, and was gradually extended to model ever more complex phenomena.

Today, TRANSYT can handle pedestrian flows, optimise green shares as well as junction offsets and include actuated signals, all the while monitoring a custom set of network-wide performance indicators that can implement whatever policy the traffic administration desires. The optimisation relies on the availability of a complete transportation network model, possibly including detailed junction geometry, roughly corresponding to the requirements for the present application as described in Chapter 4. A range of search algorithms can be used to explore complex timing solutions, which are then evaluated using either micro– or macrosimulation models. Earliest version of TRANSYT implemented a simple hill climbing algorithm that explored the non convex performance function by executing a predetermined set of short and long steps to vary each control variable in both directions alternatively. At each step, the changed value of the control variable is kept if it improved the performance index.

Park, Messer, and Urbanik [1999] introduced a traffic signal optimization program for oversaturated intersections consisting of two modules: a genetic algorithm optimizer and mesoscopic simulator. Colombaroni, Fusco, Gemma, Demiralp, Baykara, and Mastorakis [2009] devised a solution procedure that first applies a genetic algorithm and then a hill climbing algorithm for local adjustments; solution fitness being evaluated by means of a traffic model that computes platoon progression along the links, simulating their combination and possible queuing at nodes through analytical delay formulations. The model was also extended to design optimal signal settings for a synchronised artery with predetermined rules for dynamic bus priority.

Metaheuristics often see applications in traffic signal engineering that reach beyond ordinary signal planning, and have more than once played an important role in research by aiding the formalisation of less intuitive correlations between signal settings and traffic behaviour. In [Gentile and Tiddi, 2009] a Genetic Algorithm was used to venture out into the yet uncharted territory of arterial synchronisation *under heavy congestion and queue spillback*. To predict the outcome of candidate signal plans, the heuristic method relied on the General Link Transmission Model (see Chapter 4 and Gentile et al. [2010]), which implements the Kinematic Wave Theory to allow accurate simulation of traffic dynamics and model physical blockage of links, while requiring sufficiently short computation times to deal with the very large number of solutions to be evaluated. In this case, the optimisation revealed a crucial difference between subcritical and supercritical flow conditions: while in the former case the optimal green wave is led as usual by the flow velocity, the same approach proves completely ineffective under supercritical conditions, which oppositely demand that the backwards propagating jam wave speed should set the pace of upstream signals, to ensure that the residual capacity of saturated links is fully exploited.

It must be noted that the level of detail taken into account when using metaheuristics comes at a *heavy* cost in terms of computation time, which has *so far* limited the functionality of this type of software to that of advanced - yet offline - planning tools; commonly accessible computing power being insufficient for true real time operation, advanced optimization suites are staying on top of the game by attempting to streamline the interactions between the development environment and the street-level equipment, e.g. providing offline optimisation based on real time readings and quick and simple deployment of new plans.

The present work aims to break new ground by exploiting the considerably shorter computation times brought to macroscopic traffic modelling by parallel computing, as presented in [Attanasi, Silvestri, Meschini, and Gentile, 2015], and coupling them with a consolidated real-time traffic management environment to make a first step towards simulation-based heuristic optimisation based on real-time data.

Chapter 2

Smart Signals

Over the years, many attempts have been made to render the signalisation system of urban networks capable of reacting autonomously to the traffic conditions, to address the mutable nature of transportation demand.

In this context, the term *optimisation* is used in its broader sense of *choice of the best option*, whether this is picked out of a set of previously planned solutions, tailored on-the-fly onto the current traffic conditions, or simply the result of a sequence of best possible actions evaluated individually: the most relevant traits of each class of very different approaches will be illustrated in the following sections.

The one thing that all responsive traffic control systems have is the need to perceive the traffic state on the network by means of detectors. The type and amount of information required for different optimisation approaches may vary, but in the end it always boils down to one, or a combination, of the following quantities:

flow : the number of vehicles crossing a road section in a given amount of time [veh/s]

occupancy : the share of time during which a road section is occupied by any vehicle [%]

velocity : the average speed of the vehicles through a road section [m/s]

If an adaptive system is to operate the signals effectively, the above quantities must be known for all relevant arcs of the network (or sub-network) that the system is in charge of. Unless otherwise specified, it is assumed throughout this chapter that all required inputs be available and reliable for each of the described methods.

The means for obtaining and processing traffic data are beyond the scope of this work, and the integration with real-time traffic management software allows this separation of tasks; it also guarantees that reasonably reliable traffic data can be obtained for any arc of the network regardless of the physical presence of a detector on a particular road section.

2.1 Adaptive Signalisation

Signalisation of road intersections is necessary because the safety of road users and their fair sharing of the infrastructure cannot be entrusted to their own good sense. The practices used in signal planning aim to produce signal plans that are as efficient as possible, i.e. that minimise the waste of time and infrastructure capacity, under the traffic conditions that can be reasonably expected at each particular junction.

This however, as is evident from the everyday experiences of any driver, implies that often some green time that was allocated for some *potential* flow on a given approach is wasted on an empty street, while on the busiest road vehicles queue fruitlessly at the red light. It may also happen that a road that is not usually busy will temporarily become a main traffic artery because of some special event or accident, rendering the level of priority assigned to it for signal planning completely inadequate.

Adaptive traffic signals represent an attempt to avoid the inevitable inefficiencies of fixed, pre-timed signalisation by responding in real-time to the actual traffic conditions. Their main goals are therefore to:

- adapt to short term fluctuations in the vehicle arrival pattern, in order to allocate green efficiently on a cycle-to-cycle basis;
- adapt to unexpected flows deviating from statistical forecast, in order to prioritise approaches according to the real saturation levels;
- adapt to special events and accidents, and possibly act preemptively to avoid deterioration of the network performance if the occurrence is expected or can be recognised in advance.

It should be clear that these objectives, listed here in order of time frame length and complexity, may or may not be met by each class of adaptive signals, but conceptually represent the direction of desired improvements over fixed time signalisation.

The next few sections present different adaptive signalisation solutions, distinguishing between *actuated* signals that simply respond through a set of rules, and *plan generation* systems, each best suited to address one issue or the other. The specific problems addressed by the approach presented in this work are not dissimilar, as will be detailed in Chapter 4.

2.2 Traffic Actuated Signals

2.2.1 Traffic Actuated Control

The class of traffic control methods referred to as *actuated* generally don't rely much (if at all) on an underlying network model, and seldom deal with the very concept of signal program as anything beyond a predetermined sequence of phases.

In essence, an actuated controller put in charge of a junction inherits the task that once was a traffic officer's: by applying a set of rules it attempts to give as much right-of-way as possible to congested approaches — for as long as it's needed — while keeping an eye on other flows in check to avoid having any queue standing for too long. Just like their human counterparts, actuated signals are extremely effective at maximising the throughput of their own junction, thanks to the direct gauge of traffic on every approach and very fast reaction times, but may prove disastrous at the wider network level since a poorly designed control strategy may introduce self-induced oscillations in the traffic flows, rendering the whole system unstable — particularly when flows approach critical values.

Signal actuation depends on real time data acquired at the junction by short range sensors that monitor individual approaches: to this end, cameras have recently started replacing street level inductive loops, as a single device is often capable of monitoring several approaches. The first traffic actuated intersection was tested in the USA in 1930. The controller

relied on microphones to detect vehicles waiting on the lesser approaches, and drivers had to honk to signal their presence. Since then, the available technologies have improved, but the simple fact remains that with cheap electronics and very simple logic (analog friendly, if necessary) an actuated controller has long been capable of looking after a junction better than any pretimed plan ever will, no matter how well the timings are optimised to fit *expected* flows.

Different levels of automation are generally classified into two categories:

semi-actuated : the controller monitors the *low flow* secondary approaches to allocate them green time only as required, and otherwise serves the main approaches;

actuated : the controller monitors all approaches and continuously updates the duration of each phase to distribute green time optimally.

Principles of Operation

Every few seconds, an actuated controller must answer the question: *"should the transition to the next phase start right now?"* or some variant thereof (e.g. to include the possibility to skip a phase). Actuated junction control is now commonplace around the world: any given implementation may rely on different types of sensor and data (e.g. cameras or pressure plates, simple counts or occupancy), but could likely be reduced to the basic principles (based on common induction loop readings) presented in this section.

Consider a signal phase serving a single approach a to a junction.
The incoming lane is equipped with an induction loop a short way back from the stop line (just enough to remain upstream of the back of the queue for most of the time), through which the signal controller measures the time interval t_a^h between subsequent vehicle detections, commonly referred to as *headway*.

After the phase has started, the signal controller determines its duration on-the-fly, based on sensor readings in relation to a few fundamental parameters such as minimum and maximum green durations g_p^{min} and g_p^{max} and maximum headway $\hat{t}_a^h$. The latter can be considered to represent a minimum flow rate required to extend the phase duration, and does not necessarily concern a single lane group. In the scope of this example, since only one phase and one lane group are considered, the lane subscript a will be dropped.

The actuation parameters can be fixed, or determined in real time by taking into account the traffic flow on the relevant manoeuvres, the standing queues before the phase start n_a^Q, the number of vehicles queuing for lane groups served by other phases $n_{b\neq a}^Q$, or all of the above.

Between the minimum and maximum green duration values, the junction signal controller continuously checks whether the time elapsed since the last vehicle passage has exceeded the maximum headway value for the current phase. If so, the transition to the next phase begins:

$$\text{Initiate phase transition } p \to p+1 \text{ if } \begin{cases} t \geqslant g_p^{min} \\ t \geqslant g_p^{min} \vee t^h \geqslant \hat{t}_p^h \end{cases} \tag{2.1}$$

where the minimum green value can be obtained similarly to (1.7) in order to at least ensure discharge of the standing queue, possibly using an estimate of the incoming flow, and the maximum may depend on the minimum green of other manoeuvres and residual cycle

time. After the initial minimum green, the headway threshold can be a dynamic function of flows, queues and time since phase start t, e.g:

$$\hat{t}_p^h(\tau) = \hat{t}^{h0}\left(1 - \tau^{\beta\left(n_{q\neq p}^Q\right)}\right) \qquad \text{with} \quad \tau = \frac{t - g_p^{min}}{g_p^{max} - g_p^{min}} \tag{2.2}$$

whereby the maximum headway decays from its initial value $\hat{t}_a^{h0}$, over the time span between the minimum and maximum green, at a rate determined by any positive nondecreasing function β of the queues accumulated on other approaches $n_{q\neq p}^Q$.

Note that the independent variable $\tau \in [0, 1]$, which means that in case of very high or very low queues on other approaches the headway threshold drops to zero either right after the minimum green, or not until the end of the maximum green, respectively.

Furthermore, even as queues on other approaches grow between g_p^{min} and g_p^{max}, the maximum threshold remains monotonic nonincreasing as long as β is a sensible function of the (nondecreasing) queues, as seen in Figure 2.2. The basic principles expressed so far in (2.1) and (2.2) can be shaped into any of the most common categories of actuated control illustrated hereafter.

Volume actuation: in the simplest case, the green signal duration is bound between fixed minimum and maximum design values. It can be extended beyond the minimum value only as long as vehicles keep reaching the junction at sufficiently short intervals, as seen in Figure 2.1. Each vehicle arrival starts or resets a timer, and the next phase is initiated as soon as the gap between subsequent vehicles surpasses the headway threshold $\hat{t}_a^h$, which is also constant.

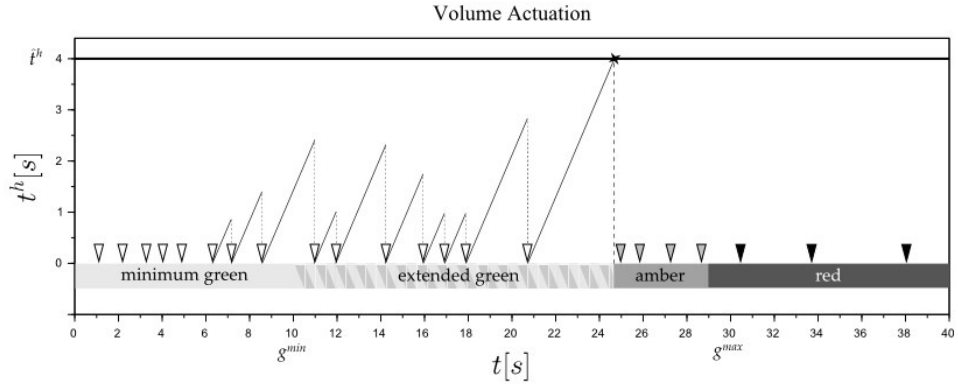

FIGURE 2.1 – Volume actuation: on the horizontal axis, time since the start of the current phase; on the vertical axis, time elapsed after each vehicle detection (triangular markers). Each vehicle reaching the sensor before the headway threshold resets the timer. Shaded and black markers respectively represent vehicles reaching the sensor after the maximum headway time has been exceeded, and vehicles that must stop at the red light.

Volume-density actuation: follows the same principles of volume actuation but the minimum green time is determined by the amount of vehicles initially queuing at the stop line. The maximum headway allowed to extend the current phase becomes more and more restrictive as the maximum green duration is approached, as portrayed by any of the lightly-shaded curves in Figure 2.2, each corresponding to a different fixed value of β.

Density actuation: the headway threshold decay rate is governed by the number of vehicles detected on the other approaches through the exponent β, so that at high saturation levels a drop in arrival rate, which denotes the end of a queue or the rear of a dense vehicle platoon, may trigger the transition to the next phase.

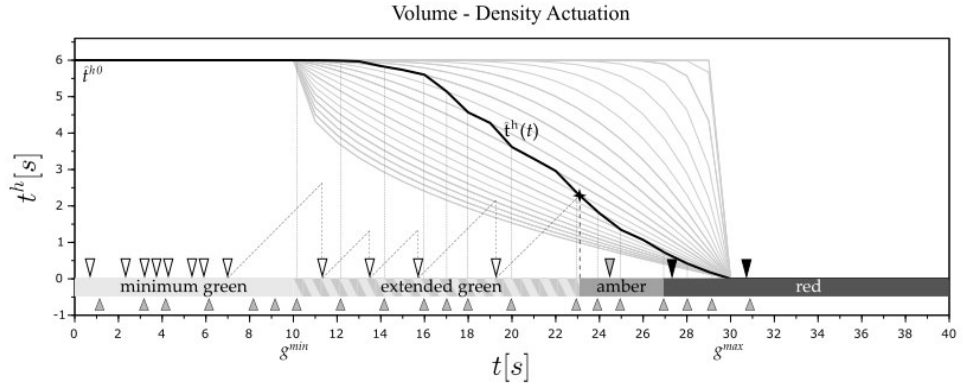

FIGURE 2.2 – Density actuation: symbols and quantities as in Figure 2.1; on the vertical axis, the headway threshold is also shown declining to zero over the green extension period following the shaded lines in the background, which correspond to polynomial curves as in (2.2) with fixed values of β. The maximum headway curve latches onto increasingly rapid decay curves with each arrival detected on other approaches, marked by the small triangles.

Although actuated controllers are mostly regarded as autonomous entities, it should be evident that phase duration limits and threshold function parameters associated with each approach can be finely tuned by a centralised system to deal with specific traffic scenarios.

Isolated actuated controllers are relatively undemanding from the infrastructural point of view, but the considerable drawback is that without junction coordination the flexibility in phase duration may come at a heavy cost in terms of arterial progression disruption.

2.2.2 Automatic Plan Selection

Plan selection systems, such as the *Urban Traffic Control System* developed by the Federal Highway Administration, aim to ensure that the most suitable amongst a set of predetermined signal plans is enacted, on the basis of real time information about the traffic conditions.

Automatic plan selection is a straightforward enhancement for both isolated traffic lights and centralised traffic control systems, which could otherwise rely only on daily plan scheduling to use different plans tailored to specific traffic conditions. These plans can be developed offline using any of the techniques mentioned in Chapter 1, with no concern for execution time or computational cost; stochastic search methods such as the one proposed in this work could well be used to devise plans for different times of day as well as response plans for specific events, with any performance objective of the planner's choosing.

Plan selection is typically performed by comparing real time detector readings with the conditions for which each plan was designed. Readings may be validated using historical data and otherwise filtered to protect the stability of the system against measurement errors

and faults. The pre-processed input is then fed into an objective function that computes the degree of suitability for each plan.

Consider for example a bank of signalisation plans $s \in S$, each representing a *solution* designed around a given traffic scenario — the generalisation applies at the network level just as well as for a single intersection, where the concepts of *plan* and *program* are equivalent. Each scenario is represented by a snapshot of the traffic conditions: assume this to come in the form of flow and occupation values measured on a subset $\mathrm{A}^{\oplus} \subseteq \mathrm{A}$ of detector-equipped arcs of the network.

The core objective function of a plan selection method quantifies the degree of *coincidence* between the flow and occupancy values $\bar{q}_{a,s}$ and $\bar{o}_{a,s}$ associated with each of the pre-timed solutions with those measured on the corresponding network arcs in real time. A possible form for such a function is e.g.

$$\omega_s = \sum_{a \in \mathrm{A}^{\oplus}} \alpha_a \cdot \left[\beta_a^q \left(q_a - \bar{q}_{a,s} \right)^2 + \beta_a^o \left(o_a - \bar{o}_{a,s} \right)^2 \right] \quad , \tag{2.3}$$

where the current flow and occupation values q and o refer to each individual arc a, as do the location weights α_a (some locations may be strategically more important than others) and the measurement weights β_a which reflect the relevance (or accuracy) of each reading at the given location.

Equation (2.3) can easily be extended to account for additional reading types. The most suitable plan is the one that minimises the performance index ω_s , representing the divergence of the current traffic conditions from its signature traffic snapshot $(\bar{\mathbf{q}}_\mathbf{s}, \bar{\mathbf{o}}_\mathbf{s})$. The system may further require the best candidate solution to beat the currently running plan by more than a predefined threshold before confirming a plan change: a cautionary measure called *Anti-hunting* taken to avoid continuous switching between similar plans, particularly in applications where a large number of plans are used to closely follow the evolution of demand throughout the day.

Switching between different plans may momentarily disrupt corridor progression, therefore in some cases a hybrid transition cycle is synthesised from the outgoing and incoming plans. The above principles equally apply to single junctions, areas or entire networks, and require a relatively low number of strategically placed detectors, making automated plan selection a viable and cost-effective option for many applications.

2.3 Real Time Signal Plan Generation

Real time optimisers that perform plan generation are a class of proactive signal control systems that, based on current traffic conditions, seek to develop an optimal plan to apply in the immediate future, either from first principles or by continuous update of an existing pre-timed plan. While each plan plays out, the system gathers information to make the next.

This mode of operation is often referred to as rolling horizon, and in order for the system to respond effectively (i.e. to capture and react to rapid changes in traffic conditions) the rolling horizon time step should be reasonably short, which imposes austere constraints on the optimisation methods. Some real-time optimisers with a very short rolling horizon step update the signalisation plan at every cycle, so that their behaviour may appear indistinguishable from that of an actuated controller.

It is important however to understand the clear conceptual difference between the two: actuated controllers perform second-by-second decisions about the best action to perform instantly, while the systems considered in this section plan ahead, producing fully featured signal plans made of cycle times, offsets and green shares deemed optimal for dealing with the traffic conditions observed.

2.3.1 Incremental Analytical Optimisation

The most prominent member of this category is the *Split Cycle and Offset Optimisation Technique* developed for research purposes in Glasgow, and first applied there in 1975 under the acronym SCOOT by which it is now popular all over the world, counting over a hundred active installations.

Continuous optimisation revolves around a centralised control unit which generates plans based on a real-time traffic snapshot gathered from detectors. The signalisation plans are continuously updated, with a frequency in the order of one to three cycle times, and may concern the entire network or *regions* thereof which are expected to feature homogeneous traffic conditions.

One of the main advantages is that optimisation requires very little information about the network. All the system needs, for each approach to a controlled junction, is the following:

- **distance** from each detector (at least one is needed) to the stop line
- **saturation flow** of the detector lane at the junction
- total vehicle **storage capacity**
- initial lost time and **clearance time** for the corresponding signal phase

The SCOOT optimisation method described in [Robertson, 1986] is based on *Cyclic Flow Profiles:* for each approach to a controlled junction these represent the continuously updated flow profile covering the span of a signal cycle with a resolution of 4 s, obtained from the readings gathered by sensors. The centralised control unit integrates CFPs to estimate the number of vehicles arriving at the stop line while the signal is red, which combined with saturation flows yields the queue sizes and clearance times as pictured in Figure 2.3.

The system is therefore all the more effective if detectors are placed far from the stop line — possibly just downstream of the previous junction — to give as early a warning as possible of changes in the expected flow pattern. This also allows the system to detect significant spillback situations, triggering different operation modes aimed at gridlock avoidance.

It may be advisable to trade accuracy for early detection by overlooking minor side streets which may alter the flow rate and progression between two major intersections. However, best results are obtained with more sensors spaced out along each inbound arc. With the flow conditions described by this simple traffic model, the optimiser proceeds to calculate cycle times, offsets and green shares based on explicit mathematical formulations (see Section 1.3).

The fitness of the solution found is quantified by a global cost function built on a linear combination of delays and number of stops. The method runs as follows:

1. *Cycle Time Computation:* each region of the network shares a single cycle time; its ideal value determined by an empirical formula similar to Webster's 1.28 based on the saturation conditions at the critical (i.e. most saturated) intersection;

2. *Green Share Optimisation:* once the cycle time is determined, green shares are updated at every intersection: as soon as it possesses relevant flow information, the optimiser decides whether to anticipate/delay each phase change by up to 4 s, depending on which alternative scores best according to an objective function (aiming to reduce the saturation level of the most saturated approach to the junction);

3. *Offset Optimisation:* at every cycle, the central unit may shift the pre-timed offsets by up to 4s in either direction, if this leads to an improvement in an explicit objective function which accounts for the degree of synchronisation with the *adjacent* junctions, possibly accounting for updated travel times of the relevant arcs.

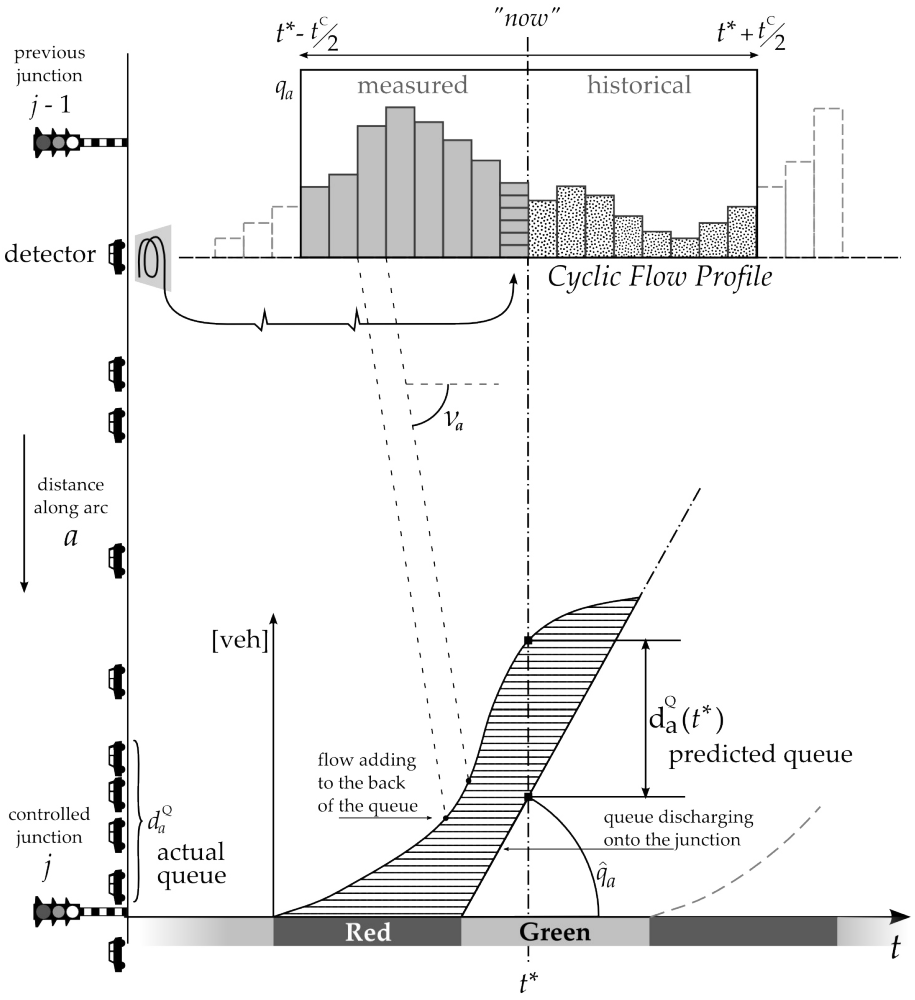

FIGURE 2.3 – SCOOT Cyclic Flow Profiles and queue prediction: detector readings are used to update the flow profile, which is integrated to predict the queue forming at the downstream junction during the red phase. The information may prompt the system to anticipate or delay a phase change in order to accommodate the measured demand.

This type of optimiser has the advantage of low modelling requirements and very fast computation times, combined with the ability to operate quite close to complete saturation —allegedly up to 90% critical junction saturation.
Even with modest prediction capabilities and no full network model, it has proven capable of dealing reasonably well with moderate flow pattern alterations and unusual route choices

such as might be caused by accidents or road works.

It does however rely heavily on the accuracy of detectors, which if insufficient may cause the performance of the system to decline rapidly: the modified timings are in fact set to degrade back to the pre-timed plan if sensor faults are detected.
The small adjustment step sizes are also chosen to increase the robustness of the system to detection faults: unfortunately, this goes to the detriment of its responsiveness, which has been pointed out as the main weakness of SCOOT.

2.3.2 Linear Quadratic Optimal Control

The *Traffic Urban Control* system commonly referred to as TUC was developed in the scope of TABASCO (Telematics Applications in BAvaria SCotland and Others), a late '90s European project aimed at demonstrating the applicability of advanced transport telematics as innovative solutions for traffic management. Initially conceived for green split optimisation, it was extended to deal with cycle and offsets as well, and later enabled to perform on-the-fly Public Transport prioritisation.

It therefore constitutes a direct alternative to the SCOOT system mentioned in the previous section, and was designed to build upon the latter's ease of applicability while addressing its main issues: most notably its slow response to rapid traffic variations (due to the incremental correction approach), and scarce effectiveness under high saturation conditions. The former was made unnecessary by the verified robustness and stability of the system, while a stronger interdependence of measurements and signal settings across the entire network helped counteract the tendency shown by more localised control policies to accelerate the onset of saturation by blindly favouring high flows.

The system inputs are the average numbers of vehicles on network links (which may be estimated from occupancy readings if video detection is not possible) and public transport information, at least accurate enough to detect the *presence* of public vehicles on a given link. Cycle and offset optimisation are carried out independently and in much the same way as it was described in the previous section.

What characterises the TUC strategy however is its approach to *green split optimisation*, based on a Store-and-Forward traffic model [Aboudolas, Papageorgiou, and Kosmatopoulos, 2009] and simple control theory. These are combined to formulate the control problem as a Linear Quadratic optimisation, as illustrated in detail in [Diakaki, Papageorgiou, and Aboudolas, 2002] and summarised here.

The instantaneous network state is represented solely by the number of vehicles on each link. The discrete-time evolution rule of the network dynamic system encapsulates its dependency on the decision variables and on previous instantaneous states, and in matrix form may be written simply as

$$\mathbf{n_{a,t+1}} = \mathbf{A}\mathbf{n_{a,t}} + \mathbf{B}\boldsymbol{\Delta}\gamma_{\mathbf{p,t}} \quad , \tag{2.4}$$

where $\mathbf{An_{a,t}}$ is the vector of states containing the number of vehicles on each link and $\boldsymbol{\Delta}\gamma_{\mathbf{p,t}}$ is the vector of variations in green share applied to each signal phase, with respect to a baseline signal plan assumed to lead to steady state queues under non-saturating conditions.

$\mathbf{A}$ and $\mathbf{B}$ are the state and input matrices: they respectively encapsulate the network topology and the expected impact of signalisation (based on signal staging, turning rates, saturation flows) on the movements of traffic volumes across time intervals.

It should be noted that the *expected* demand is taken into no account: this is reasonable as TUC aims to react to the *manifest* impact of disturbances on the controlled network rather than to their forecast consequences. At the core of this approach lies a simple gain matrix — introduced in Equation (2.6) — rather than accurate modelling of physical phenomena and constraints: its calculation and calibration however are most computationally demanding processes.

In order to minimise the risk of oversaturation and queue spillback on all network links, the chosen strategy is to attempt to balance the link relative occupancies (with respect to each link's jam storage capacity), as expressed by the following quadratic criterion:

$$\omega^{TUC} = \frac{1}{2} \sum_{t=0}^{\infty} \|\mathbf{Q}\,\mathbf{n_{a,t}}\|^2 + \|\mathbf{R}\,\mathbf{\Delta}\gamma_{\mathbf{y,t}}\|^2 \quad , \tag{2.5}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are non-negative definite diagonal matrices of weights, so that the cost function is compatible with the standard form of a Linear Quadratic Cost.

Matrix $\mathbf{Q}$ contains the inverse storage capacities of links, so that the first term of the sum drives the relative occupancy balancing, while the second term favours smooth changes in the control variables, influencing the magnitude of control reactions through appropriate scaling factors contained in matrix $\mathbf{R}$. The infinite time horizon of the sum reflects the necessity to obtain a time-invariant feedback control law in accordance with LQ optimisation theory.

The LQ feedback control law is then obtained by minimisation of the performance criterion (2.5) subject to (2.4) : calculation of the control matrix $\mathbf{L}$ is straightforward, but can only be performed offline by solving the infinite-horizon *Discrete-time Algebraic Riccati Equation* from the network topology and objective function weights described by the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{Q}$, and $\mathbf{R}$. These must be computed and calibrated individually for the specific network topology, capacities, signal staging etc, by simulation or other optimisation methods: this is a lengthy and demanding task to be performed as part of the system setup.

However, after finding the stabilising solution $\mathbf{L}$ to the dynamical system expressed by the DARE, things get much simpler, with the control law taking the standard form

$$\gamma_{p,t} = \bar{\gamma}_p - \mathbf{L}\,\mathbf{n_{a,t}} \tag{2.6}$$

where $\bar{\gamma}_p$ is the vector of baseline green shares. Optimal modifications to the green times are linearly dependent on the current network state vector of link occupancy measurements through the matrix $\mathbf{L}$, which provides both the discharge and gating functionalities: intuitively, as the occupancy of a link increases, so does the green share that favours its outflow, while upstream arcs experience a reduction in green time to avoid its oversaturation.
These effects can be accentuated or mitigated by weighing elements of the state vector according to specific rules, e.g. to prioritise desaturation of certain links as they approach critical saturation levels.

A simple form of *public transport prioritisation* can be integrated into the application of the control equation (2.6) , by further weighting link occupancy values in function of the number of public transport vehicles detected on them.

Since control constraints such as green time upper and lower bounds cannot be directly accounted for by the LQ methodology, the green shares output by the regulator are further processed on the fly by a simple optimisation algorithm that, in linear time, finds the set of feasible green times that least deviate from the optimum.

Although several software packages are available for solving the DARE for this standard LQ control problem using a variety of well documented methods, the calculation of an effective control matrix remains a time consuming task, particularly for large networks, and must be performed anew every time the controlled network is modified or extended.
This lack of flexibility represents the main drawback of the approach just presented, although it has been proven that reasonable variations of traffic parameters such as turning rates and link saturation flows have little effect on the control matrix.
On the other hand, the real-time operation of the TUC control strategy only consists of the solution of the simple matrix equation (2.6) followed by the application of green time constraints, which are both extremely fast and undemanding operations, making the quadratic regulator a particularly suitable approach for real time applications. Furthermore, the feedback controller is perfectly capable of responding appropriately to very specific traffic anomalies such as accidents or roadworks, as confirmed by both simulation and empirical data gathered from real world installations.

Since the first TUC installation in Glasgow, further applications of optimum control theory to the signal setting problem, including open-loop Quadratic Programming and Nonlinear Optimum Control based on the same store-and-forward traffic paradigm, have been developed and investigated. These aim to improve upon the performance of the simple feedback controller by accounting for more detailed network dynamics, factoring in time varying demand, or allowing for a larger and more effective set of decision variables: encouraging results presented in [Diakaki, Dinopoulou, Aboudolas, Papageorgiou, Ben-Shabat, Seider, and Leibov, 2003] suggest that despite an increased real-time computation complexity, these may be considered strong competitors and potential successors to the Linear Quadratic TUC approach.

2.3.3 Traffic Gating

Feedback Traffic Gating as described by Keyvan-Ekbatani, Kouvelas, Papamichail, and Papageorgiou [2012] is a form of actuated signal control aiming to prevent oversaturation of critical portions of the network by holding back the incoming traffic flows — using deliberately exaggerated red phases — rather than attempting to deal with the flows already trapped in a congested area.

In these respects, it constitutes a simple yet innovative method to induce more efficient utilisation of the existing infrastructure, and an answer to the patent performance degradation that currently feasible real-time optimisation solutions face under saturated conditions; it is therefore also closely related to the object of this study.

Based on the general principle that even from the users' point of view there is no advantage to getting close to one's destination sooner, only to be stuck in traffic for longer, the system delays incoming vehicles in order to keep the controlled network near to but *below* its saturation occupancy level, which can be monitored effectively even with a small number of detectors [Keyvan-Ekbatani, Papageorgiou, and Papamichail, 2013]: this was proven the most effective strategy to maximise *network throughput*, which constitutes a good measure of how efficiently the network is being used.

A feedback controller is used to ensure that the only vehicles pre-emptively delayed are those which not only would, on average, be delayed anyway further down their path, but would *critically* increase congestion — causing themselves as well as others greater delays were they to access the critical region.

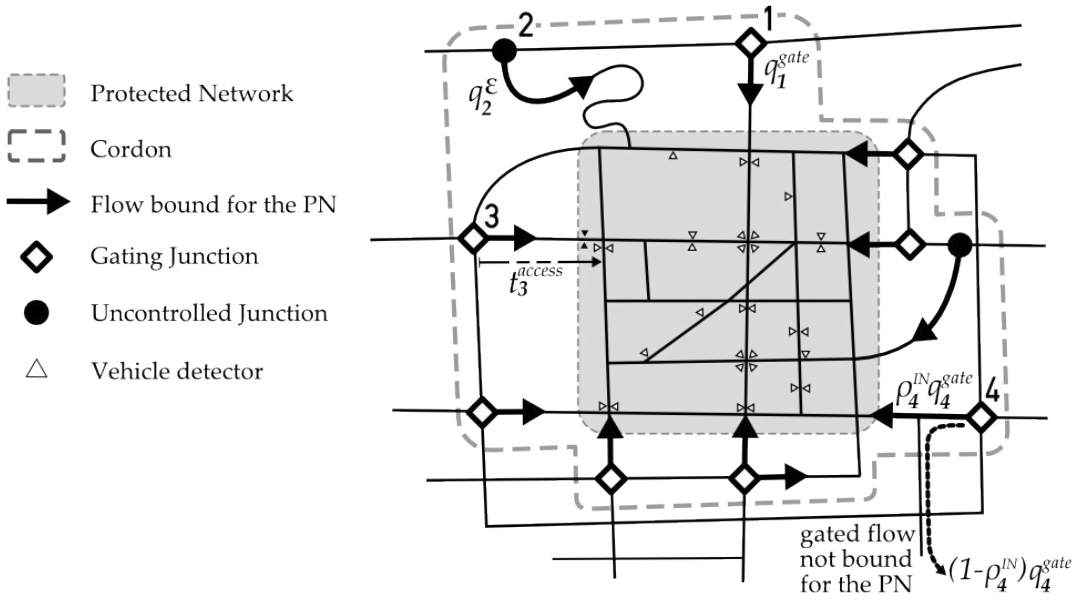

FIGURE 2.4 – Traffic Gating: a cordon of gating junctions holds back traffic attempting to access the *Protected Network*. If it is not possible to implement gating at one or more cordon junctions (see $j = 2$), these may allow some disturbance flows to sneak past the feedback controller. Conversely, part of the gated flow from a cordon junction may not in fact be bound for the PN (see $j = 4$). The system must account for the delay between control application and effect (due to the physical distance between the gating junctions and the PN, see $j = 3$) and incomplete or uneven detector placement in the protected network.

2.3.4 Network Fundamental Diagram Formulation

Feedback Traffic Gating revolves around the concept of Network Fundamental Diagram introduced in [Keyvan-Ekbatani et al., 2012], profiling throughput as a function of occupancy, as seen in Figure 2.5 where the axes of the sample NFD correspond to *Total Time Spent* (in vehicle-hours per hour) and *Total Travelled Distance* (in vehicle kilometres per hour) cumulatively by all users hourly.

Such relationship may be obtained empirically from observation of the area of interest, and allows to identify with certainty the optimal operation point for the feedback controller to suit the behaviour of a specific network.

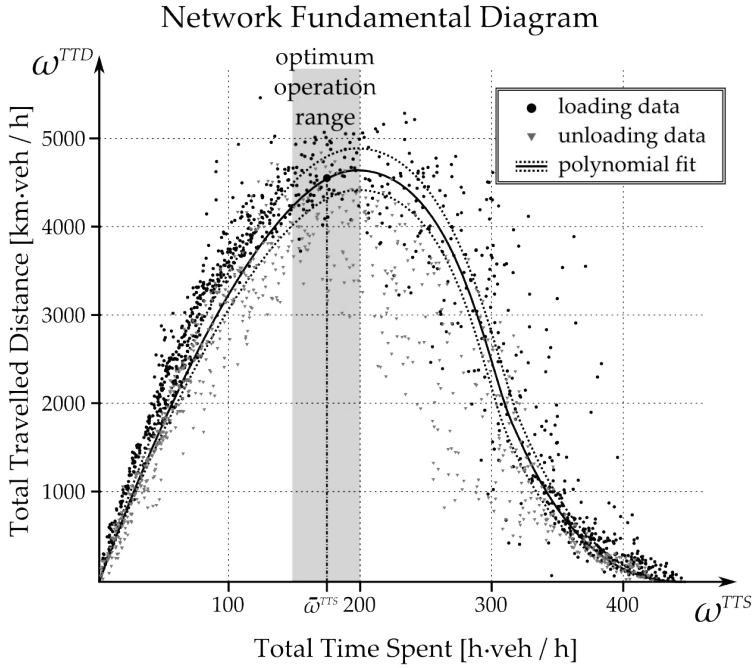

FIGURE 2.5 – An experimental Network Fundamental Diagram: a polynomial fit of the *TTD* curve is obtained from flow and occupancy measurements, identifying an optimum operation point on the *TTS* axis. As long as $\omega_{\mathrm{A}^{\oplus}}^{TTS}$ is kept within the optimum operation range, the number of vehicles in the Protected Network is expected to maximise the infrastructure efficiency, resulting in shorter travel times for all users. The dynamics of the system during network loading and unloading can be expected to differ as flows tend to be slower during the relaxation of a more congested state.

An operational NFD is derived from real or simulated occupancy measurements a o taken on a set of detector equipped arcs $\mathrm{A}^{\oplus} \subseteq \mathrm{A}$ at discrete time intervals t corresponding to signal cycles. Occupancy is converted into an estimate $n_{a,t}$ of the number of vehicles on each arc during the t^{th} signal cycle, given by

$$n_{a,t} = \frac{\ell_a \cdot o_{a,t}}{100\,\ell_{veh}} \tag{2.7}$$

where ℓ_a is the length of link a, $o_{a,t}$ its occupancy (given as time percentage) during the interval, and ℓ_{veh} the average vehicle length. Hence, the relevant quantities are obtained by summing over the measurement arcs:

$$\omega_t^{TTS} = \sum_{a \in \mathrm{A}^{\oplus}} \frac{n_{a,t} \cdot t_{\mathrm{J}}^{C}}{t_{\mathrm{J}}^{C}} = \sum_{a \in \mathrm{A}^{\oplus}} n_{a,t} \quad ; \tag{2.8}$$

$$\omega_t^{TTD} = \sum_{a \in \mathrm{A}^{\oplus}} \frac{q_{a,t} \cdot \ell_a \cdot t_{\mathrm{J}}^{C}}{t_{\mathrm{J}}^{C}} = \sum_{a \in \mathrm{A}^{\oplus}} q_{a,t} \cdot \ell_a \quad . \tag{2.9}$$

The values thus obtained are sufficiently precise for the purpose of traffic gating, especially if detectors are located around the arc midpoints. Although a high number of detector links (ideally $\mathrm{A}^{\oplus} = \mathrm{A}$) yields a more accurate NFL, [Keyvan-Ekbatani et al., 2013] proves that fully functional results can be obtained also from a *reduced* NFL in more likely scenarios where only a costeffective subset of links has detection capabilities, such as would be sufficient for ordinary traffic monitoring, plan-selection schemes, or actuated signal control applications, as portrayed e.g. in Figure 2.4.

Feedback Controller Design

The gating control problem is to regulate the *TTS* in the Protected Network via appropriate manipulation of gated inflows, so as to maintain the *TTD* around its optimal maximum value. The task can be accomplished as summarised in Figure 2.6 based solely on real-time measurements via a simple and robust feedback regulator, taking advantage of the basic system dynamics described by the NFD.

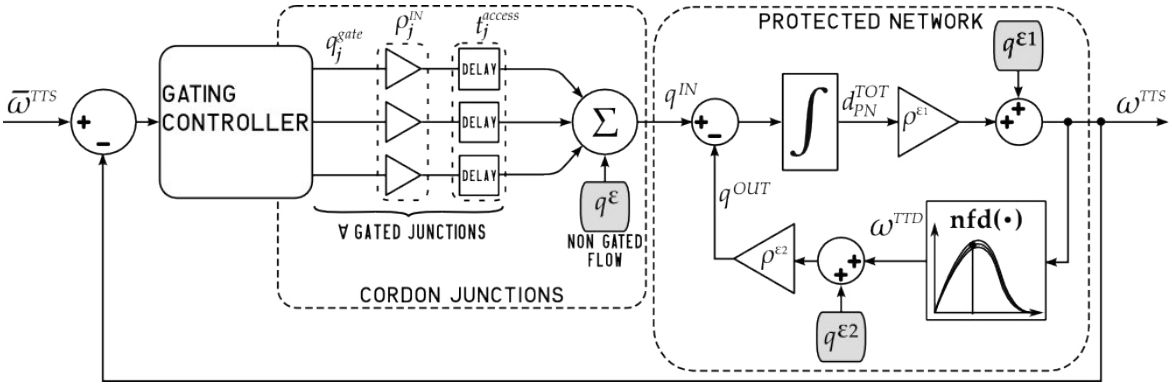

FIGURE 2.6 – Gating Feedback Controller and Protected Network Dynamic Plant.

The controlled input to the PN system is the gated flow q^g, and the main disturbance the uncontrolled inflow q^{ϵ}. Referring to Figure 2.6, the inflow q^{IN} in continuous time is

$$q_t^{IN} = \rho^{IN} \cdot q^g \left(t - t^{access}\right) \tag{2.10}$$

where ρ^{IN} is the portion of gated flow entering the PN, and t^{access} the time it takes for vehicles to reach the PN from gating junctions not directly located on its boundary.

Consider the total number n_{PN} d of vehicles in the PN: its rate of change is determined form vehicle conservation, which reads

$$\dot{n_{PN}} = q^{IN} + q^{\epsilon} - q^{OUT} \quad . \tag{2.11}$$

However, $\omega_{PN}^{TTS} = n_{PN}$ only if all PN links are monitored, which is not generally the case. Realistically, the TTS is smaller than the true number of vehicles by some factor $\rho^{\epsilon 1} \leqslant 1$. Allowing for an additional measurement error $q^{\epsilon 1}$ the TTS value to be used in the NFD is

$$\omega_{PN}^{TTS} = \rho^{\epsilon 1} \cdot n_{PN} + q^{\epsilon 1} \quad . \tag{2.12}$$

and finally, if $nfd(\omega_{PN}^{TTS})$ is a nonlinear best fit of the NFD data (see Figure 2.5) and $q^{\epsilon 2}$ the error due to the data scatter, the resulting TTD is

$$\omega_{PN}^{TTD} = nfd\left(\omega_{PN}^{TTS}\right) + q^{\epsilon 2} \tag{2.13}$$

which as seen in Figure 2.6 is proportional to the network outflow q^{OUT} aside for a scaling factor $\rho^{\epsilon 2}$ analogous to $\rho^{\epsilon 1}$, yielding a time delayed nonlinear first-order model between the initial q^{g} and the resulting TTS which can be linearised around the optimum steady state.

The following proportional-integral controller is then well suited to handle the gated flows:

$$q_t^{IN} = q_{t-1}^{IN} - K^P \left(\omega_{t-1}^{TTS} - \omega_{t-2}^{TTS}\right) + K^I \left(\bar{\omega}^{TTS} - \omega_{t-1}^{TTS}\right) \tag{2.14}$$

where K^I and K^P are the integral and proportional gains to be fine tuned.
The flow values thus determined have to be shared amongst all gated junctions, after accounting for monitored or estimated disturbance flows, and subjected to minimum/maximum green time constraints.

The resulting system is largely robust to measurement errors, low signal timing resolution, and fluctuations in demand. It may be activated at specific times or as the traffic conditions approach a critical state, and requires virtually no additional infrastructure with respect to an ordinary plan-selection centralised signal setting system. Provided that appropriate gating locations can be found, where gate-delayed flows do not risk compromising the mobility of vehicles not bound for the PN, the principles just illustrated will undoubtedly form the core of future sustainable approaches to relieve urban congestion by delaying or avoiding the extreme traffic conditions that frustrate most currently available signal optimisation techniques.

Chapter 3

Modelling, Simulation and Optimisation Tools

This chapter presents the relevant elements of the real time traffic management framework in which the work was conducted, illustrating the most interesting features in light of their role in the optimisation. An introduction to the basic principles of the Genetic Algorithm completes the inventory of the tools used to bring together the optimisation presented in Chapter 4.

3.1 The Optima Framework

It is important to understand that much of the value of the present work lies in the fact that it has been carried out in a production environment and that the resulting product is embedded in a real-time traffic management system.

The main advantages of this integration can be summarised by considering the *requirements* for an effective and informed optimisation of a complex real world phenomenon:

knowing the resources - integration with state-of-the-art modelling software allows the optimisation to work from a representation of the real world which is as accurate as possible, encompassing everything from dynamic user demand to industry-grade models of the relevant road networks, down to detailed junction geometries;

knowing what's happening - the static model is overlaid with real-time updates about the current state of supply and demand, coming from a variety of sources and harmonised into a coherent snapshot of traffic conditions;

knowing the processes - using accurate macroscopic simulation as a means of evaluating potential solutions allows to optimise without oversimplifying the dynamics of the system;

knowing what's best - or at least working in an environment where complex performance objectives can be easily defined, calculated, evaluated and improved as necessary, allowing not only to optimise towards specific goals but to identify their side effects and explore less obvious approaches;

knowing whether it's working - by relying on specific visualisation tools to streamline the analysis of results.

The most relevant components of the Optima Real-Time Framework involved in the aforementioned integration are shown in Figure 3.1, to clarify the functionalities they cover and their utility in relation to each other and to the new development.

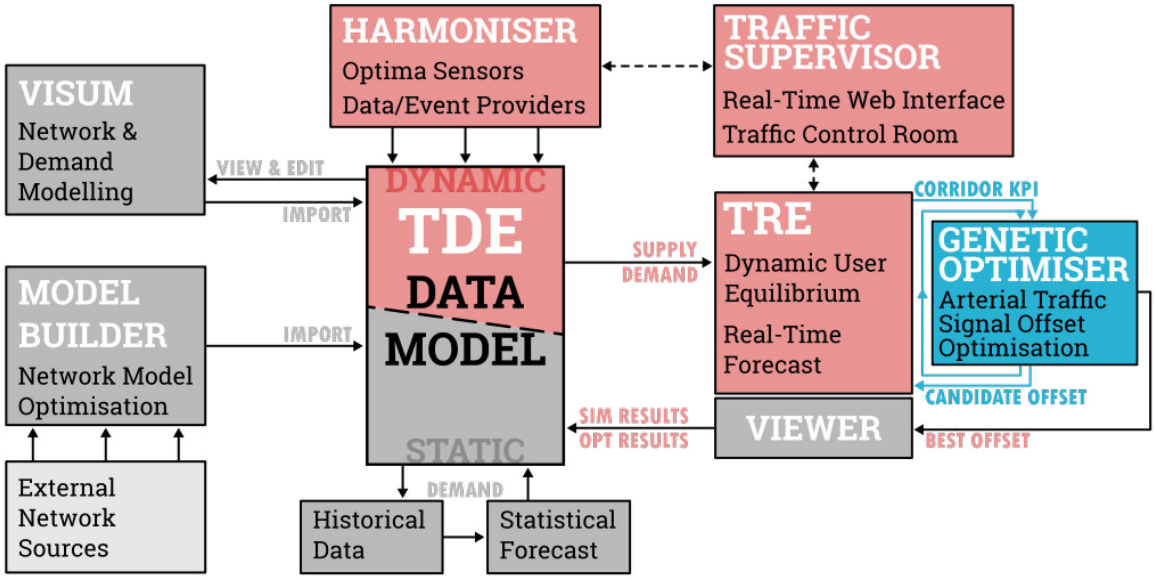

FIGURE 3.1 – The Optima framework for real time traffic management. Arrows roughly represent the direction of interactions and data exchange between the different components. The optimiser module is shown on the right, directly attached to the simulation engine TRE which it uses for solution evaluation.

Besided the simulation engine (presented in detail in Section 3.2) the Optima environment relies on all of these components, articulated around a data model called *Traffic Data Exchange*, to bring together static and dynamic data into a complete transportation system. In this framework, each functionality is covered by dedicated software, and all resources are accessible to all components concerned.

The optmiser itself is relieved from the necessity to incorporate its own traffic model, or to handle and validate sensor data as many currently available signal optimisation systems must do. In fact, it doesn't need to know the model at all, but can take full advantage of it and of the network-wide harmonised traffic data available within the system via the simulation engine that provides the necessary evaluation capabilities. At the other end, it is still the framework that provides the interface with street-level equipment (the actual signal controllers) and ensures that the optimised plans are readily and safely implemented.

3.2 TRE simulation engine

The proposed optimisation method relies on the macrosimulation engine known as TRE, based on the eponymous Dynamic User Equilibrium assignment algorithm.

TRE lies at the core of the Optima traffic management software suite, and is used in traffic control centres around the globe.
This section aims to illustrate its fundamental principles of operation, in order to clarify how they might affect the optimisation and better understand the role of the simulation engine within the architecture.

3.2.1 Continuous Dynamic Traffic Assignment

The general idea of *Traffic Assignment* is rather intuitive: it is the modelling of the interaction between *supply*, i.e. the roads, infrastructures and public transport options; and the *demand* for mobility, i.e. people that need to travel using a choice of the available resources.

Since supply is limited, its availability and performance are a consequence of the choices made by users, which in turn are affected by the perceived discomfort of travelling across the network in the state it actually is: to predict with any plausibility the way in which traffic will spread across the network, it is necessary to resolve this reciprocal influence between supply and demand.

Of the many approaches proposed to this end throughout the history of transport research, the most successful are based on the *Selfish User Equilibrium* principle first stated by Wardrop [1952]. This follows from the simple and sound behavioural assumption that every user will choose the route and mode of transport which are best for them, and implies that the most reasonably foreseeable traffic scenario is that in which no user would benefit from making a different choice: hence the notion of user *equilibrium*.

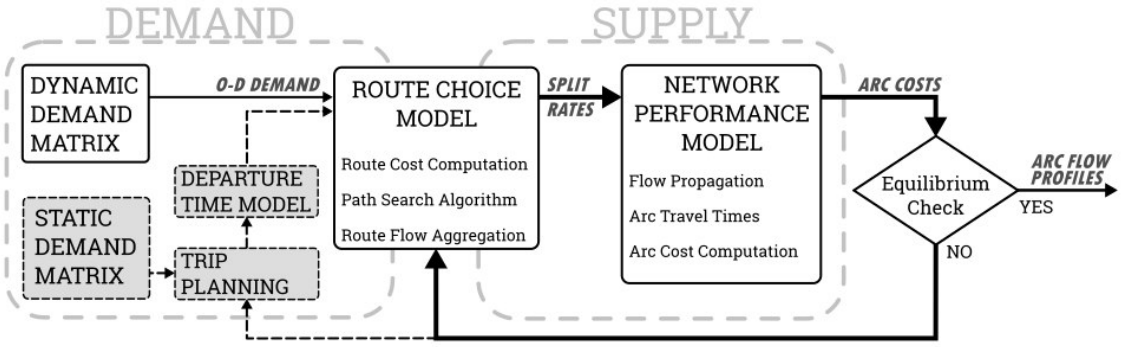

FIGURE 3.2 – Flowchart schematic of the a Dynamic Traffic Assignment based on the Selfish User Equilibrium condition. The algorithm searches for arc flow profiles and route choices that satisfy the equilibrium condition by cyclically evaluating the reciprocal influences between supply and demand, until convergence.

Even in the simplest possible *static* case, with steady demand and travel times only dependent on user choices, the equilibrium point must be found by an iterative process as illustrated in Figure 3.2. Thus, at each iteration

1 . demand is routed through the network according to the arc costs, route flows are calculated;

2 . flows are assigned to the relevant arcs, costs are updated to account for congestion and checked against convergence criteria;

3a. until convergence is obtained, costs are fed back into a new demand routing (step 1);

3b. when arc costs converge, it means users are confirming the route choices made in the previous iteration: the *user equilibrium* is satisfied, and the last route flows and costs calculated are the best estimate of the outcome from the given demand and supply.

In reality, demand is hardly constant throughout the day and congestion occurs as a consequence of the *history* of the system; therefore any real-world application must account for the fact that travel times and user choices evolve *dynamically* over time.

A *Dynamic Traffic Assignment* (DTA) model allows to determine the dynamic interaction of supply and demand to predict the evolution of traffic conditions over any length of time, conceptually without further complication beyond the addition of the temporal dimension.

The user equilibrium condition can still be found via the mechanism illustrated in Figure 3.2, working from the knowledge of demand and supply; with the difference that demand and user choices will be time dependent, and flows will propagate in space *and time* so that arc costs become dynamic too. Traffic is considered to be a continuum, both as far as vehicle movements and trip-maker decisions are concerned; the equilibrium is then found between time profiles of arc flows and costs.

Trip planning and route choice models are extremely relevant to the accuracy of a DTA and to the computational effort involved, but in the general DTA framework they are quite independent of each other and of the supply model.

The representation of traffic as it propagates and interacts with the network and signals, and all related phenomena within the scope of the present study, fall upon the Dynamic Network Loading model. The most suitable macroscopic model for the task at hand is the General Link Transmission Model, which will be analysed in further detail over the next few sections.

3.2.2 General Link Transmission Model

The *General Link Transmission Model*, henceforth referred to as GLTM, is a model for continuous dynamic network loading: it can be used to determine time-dependent link flows $q_{a,\tau}$ and travel times $t^t_{a,\tau}$ given the time-dependent route flows.

It is built upon the representation of traffic as a partially compressible one-dimensional fluid flowing through the network according to the principles of *Kinematic Wave Theory* (KWT), as developed independently by Lighthill and Whitham [1955] and Richards [1956].

Its origins can be traced back to the *Cell Transmission Model* first presented in relation to highway traffic by Daganzo [1994] and shortly after applied to network traffic in [Daganzo, 1995]. CTM was the first dynamic traffic representation based on hydrodynamic theory, and borrowed heavily from that field, as is most evident from the cell-based space discretisation of the road network adopted directly from computational fluid dynamics.

The need for cell discretisation was eliminated in the *Link Transmission Model* presented by Yperman, Logghe, and Immers [2005]. This innovative approach allowed dynamic network loading of large scale networks using a computationally efficient algorithm that only required calculations at intersection nodes, while solving for traffic propagation along whole links using kinematic wave theory: this allowed to do away with a significant complexity factor while still accurately modelling local flow restrictions and junction delays.

The original LTM, presented in full detail in Yperman [2007], rather simplified the wave propagation problem relying on the *simplified* kinematic wave theory proposed by [Newell, 1993], whereby only two possible wave propagation speeds are contemplated: a forwards one for free-flowing traffic, and one for the congested flow states to propagate backwards. This is a considerable approximation, as the relation between vehicle density, speed and the resulting flow is rather more complex in reality: the instrument provided by kinematic wave theory to express such relations in general is the Density-Flow Fundamental Diagram, illustrated in section 3.2.3.

While in truth the work of Yperman already improved on the simplified KWT approach to include any piecewise linear fundamental diagram, the GLTM presented in Gentile et al.

[2010] was developed to extend the LTM formulation to any concave fundamental diagram, considerably improving the accuracy in representing delays due to congestion. The GLTM also uses time-varying capacity adjustments at nodes to accurately model conflicts at intersections and the so called *spillback* of traffic states from downstream links to the relevant upstream ones.

The features of the supply model described in the next sections, together with the computational efficiency and the possibility to perform a DTA in high temporal resolution, make the General Link Transmission Model an optimal candidate for the present application.

3.2.3 Link Model

In the GLTM, traffic propagates along links according to kinematic wave theory. The ability to model complex phenomena, such as the influence of congestion on driving speeds and the formation of queues, is crucial for optimisation since it enables to capture more realistic traffic dynamics.

As stated in section 1.1, links are assimilated to weighted arcs of a directed graph, and as such are one-dimensional, one-directional and homogeneous along their length, stretching between locations x^0 (the tail node) and $x^1 = x^0 + \ell$ (the head node): the actual link shape is inconsequential. As far as the arc model is concerned, there is no need to disambiguate arcs since they exist and are processed independently: arc subscripts can be dropped for ease of reading but are to be implied on all relevant quantities henceforth.

The traffic state at a specific location $x \in \left[x^0, x^1\right]$ along a link is characterised by three macroscopic variables:

flow q_x : vehicles through the link section per unit time;

density k_x : average number of vehicles per unit length;

speed v_x : average distance covered per unit time.

As is evident from their dimensions, only two of these quantities can be independent, and if two are known the third may be readily calculated using the relationship

$$v = \frac{q}{k} \quad . \tag{3.1}$$

The idea that vehicle density and speed can be completely independent, however mathematically sound, does not seem practically plausible. Kinematic wave theory provides a device for solving this contradiction as illustrated in the following section.

Fundamental Diagram

Kinematic wave theory assumes a functional relation between traffic density and flow, known as the *Fundamental Diagram* of traffic flow. It approximates the changes in the average behaviour of drivers as the road gets more crowded, and may take several forms, but invariably follows from the properties of the road, e.g. width, slope or parking. As such it is itself, conceptually, a property of the link, although it could also be made to depend on environmental factors and driver behaviour, or be specific to a particular class of vehicles.

A generic fundamental diagram expresses the relationship between flow and density under *stationary* traffic conditions, i.e. it is derived as an equilibrium condition between flow speed and available space taking the general form

$$q = f(k) \tag{3.2}$$

which may be represented on a Density-Flow graph like the one shown in Figure 3.3.

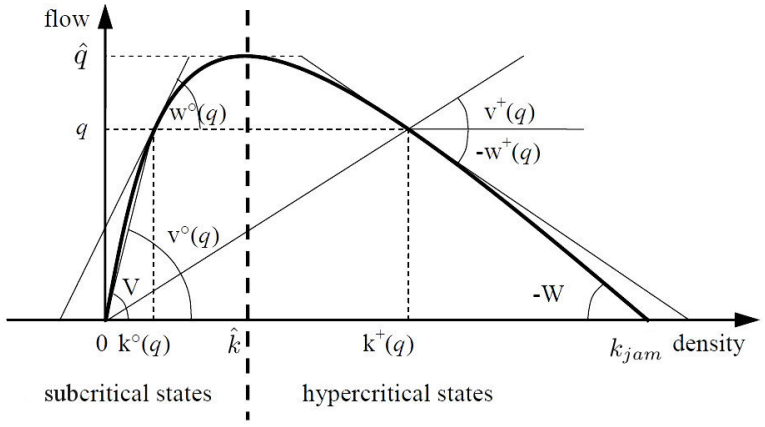

FIGURE 3.3 – Fundamental Diagram of a link, representing the functional relation between vehicular density and speed, resulting in different flow values for different congestion levels. The curve is shaped mainly by the critical density value $\hat{k}$, the jam density k_{jam} and the free flow speed v^0.

The shape of a fundamental diagram reflects different assumptions about traffic flow dynamics, but there are a few key features that are shared by all formulations:

- when density approaches *zero* the speed approaches the maximum value attainable on the link, i.e. the *free flow* speed v^0, but the flow tends to zero;
- maximum flow occurs at the *critical density* $\hat{k}$ also referred to as the link *capacity*;
- beyond capacity, further increase in vehicular density induces a speed penalty that causes the flow to decrease;
- when vehicles reach the *jam density* k_{jam} they are packed as closely as possible, and come to a standstill;
- for any flow state on the $k - q$ curve, the speed is given by the slope of the line connecting it to the origin;
- the rising branch diagram (i.e. to the left of $\hat{k}$) represents free flowing states, the descending branch represents congested states.

In a simple triangular diagram like the one shown in Figure 3.3 (left) the speed is assumed constant at its maximum value for all subcritical states, while above capacity it decreases linearly with density. More subtle modelling may yield a diagram shape more similar to Figure 3.3 (right), where the speed is shown to decrease even in subcritical conditions as the road gets more crowded due to the natural variance in driving speed which, as more vehicles become involved, yields a higher chance of having a slow vehicle delaying all the others *(subcritical spacing)*.

In both cases it is assumed that as density increases, the available space becomes insufficient to maintain safe distances between vehicles, causing drivers to slow down *(hypercritical spacing)*. A detailed analysis of the presented fundamental diagram alternatives, along others that have been proposed in literature, is given in [Tiddi, 2012].

If a model is to rely on the fundamental diagram to hold for non-stationary traffic as well, it must allow vehicles to change speed instantly with infinite acceleration, as is the case with GLTM and in general with first-order implementations of KWT.
Higher order traffic phenomena such as the emergence of stop-and-go waves along the link, or fundamental diagram hysteresis (due to traffic states evolving asimmetrically when leading up to congestion or recovering from it) are knowingly neglected.

Traffic State Propagation

A brief overview of the fundamentals of the simplified KWT is given here inasmuch as it is relevant to the context: for a more detailed discussion of these well-known principles the reader is encouraged to refer to the original works of Yperman [2007] and Gentile et al. [2010].

Consider the *cumulative flow* $N(x,t)$, i.e. the number of vehicles that have passed location x along a link before time t. Assuming that vehicle conservation is respected along the link, i.e. that no vehicle is created or destroyed between the tail and the head node, the trajectory of the n^{th} vehicle to enter the arc can be traced on a time-space diagram as the locus of points for which $N(x,t) = n$ as shown in Figure 3.4.

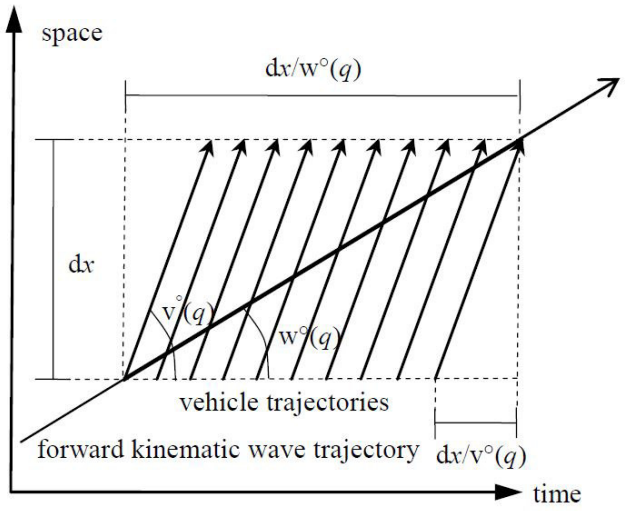

FIGURE 3.4 – Vehicle trajectories and a kinematic wave front trajectory on a time-distance diagram: the inclination is determined respectively by the vehicle speed and the forward propagation speed. Notice these two are not necessarily the same since the kinematic wave represents the propagation in space of a flow *state*, and not a specific vehicle: the exception would be the "first" vehicle on an empty arc (referring to the origin of the subcritical branch in Figure 3.3) which would carry its own flow state.

The cumulative function $N(x,t)$ is clearly discontinuous in both time and space, but it is possible to consider a smooth approximation that is differentiable in either direction without altering the essence of the phenomenon. Flow and density values at a given location and

time can then be expressed as the partial derivatives

$$q(x,t) = \frac{\partial N(x,t)}{\partial t} \quad , \tag{3.3}$$

$$k(x,t) = \frac{-\partial N(x,t)}{\partial x} \quad , \tag{3.4}$$

the latter requiring a sign change simply because density is defined positive but the cumulative decreases along the positive spatial direction.

Given a generic link a of length $\ell > 0$, let $f(t) = q(x^0, t)$ be its inflow and $e(t) = q(x^1 t)$ its outflow at time t, assuming the link's own spatial frame of reference so that the initial section x^0 is simply 0 and the final section x^1 is found at ℓ.

By definition, the cumulative inflow and outflow are given by:

$$n^F_{a,t} = N(0,t) = \int_0^t f(\tau)\mathrm{d}\tau \quad ; \tag{3.5}$$

$$n^E_{a,t} = N(\ell,t) = \int_0^t e(\tau)\mathrm{d}\tau \quad . \tag{3.6}$$

A forward kinematic wave generated at time t at the initial section of the link reaches the generic section x at instant $u(x,t) \geqslant t$ given by

$$u(x,t) = t + x/w^0(f(t)) \tag{3.7}$$

where w^0 is the forward kinematic wave speed depending on the inflow, as seen in Figures 3.4 and 3.3. In general, $u(x,t)$ is not invertible, since more than one kinematic wave generated on the initial point may reach the final point at the same time (for decreasing inflows). If $f(t)$ is the prevailing flow state at time $u(x,t)$ at the final section, the corresponding cumulative flow is given by n^F_t plus the number of vehicles that have passed the forward kinematic wave generated at t in the initial point.

The Newell-Luke Minimum Principle (NLMP) states that, among all forward kinematic waves that reach the final point at time t, the one yielding the *minimum* cumulative flow denoted n^H_t dominates.

Conversely, the instant $z(x,t) \geqslant t$ when the backward kinematic wave generated by the hypercritical *outflow* $e(t)$ reaches the link entry section is given by:

$$u(x,t) = t - \ell/w^+(e(t)) \quad , \tag{3.8}$$

which features the hypercritical wave propagation speed $w^+ < 0$ and is also not invertible, since more than one kinematic wave generated on the final point may reach the initial point at the same time for decreasing outflows. Once again if $e(t)$ is the prevailing flow state at time $z(0,t)$ in the initial point, the corresponding cumulative flow is given by n^E_t plus the number of vehicles that have passed the backward forward kinematic wave, and the NLMP states that among all backward kinematic waves that reach the initial point at time t the one yielding the *minimum* cumulative flow, denoted n^G_t, dominates the others.

The network is thus modelled as a set of links, each consisting of a homogeneous channel with bottlenecks at its entrance exit sections, that connect the nodes where mergings and diversions take place. Cumulative flows $n^H_{a,t}$ and $n^G_{a,t}$ are used to determine the sending and receiving flows, which are input to the node model at N^+_a and N^-_a respectively.

3.2.4 Node Model

The node model handles the merging and diversion of link flows, and encapsulates precedence rules, conflicts between manoeuvres and signalisation in the form of dynamic capacity bottlenecks applied at the arcs' exit and entry sections. It is of fundamental importance to the present application as it is responsible for the implementation of signals in the simulated environment.

At each node n, the model takes as input the sending flow of all its backward star links $a \in \mathrm{A}_n^-$ and the receiving flow of all forward star links $b \in \mathrm{A}_n^+$ to provide as output the inflow to forward links and the outflow of backward links, according to the rules presented in this section.

In a diversion node $n \in \mathrm{N}$ where routing takes place, the node model consists in propagating flows consistently with the given path choices and satisfying the FIFO rule (no overtaking allowed). Path choices determined by the demand model are represented here by the splitting rate p_{ab}, expressing the probability that the next link of a path coming from link $a \in \mathrm{A}_n^-$ is $b \in \mathrm{A}_n^+$. The *demand* flow d_{ab} that will try to perform turn $a \to b$ is given by

$$d_{ab} = s_a \cdot p_{ab} \quad , \tag{3.9}$$

where the *sending flow* s_a represents the rate at which vehicles reach the exit of link a, capped where appropriate by its exit bottleneck: the problem is to determine the most severe restriction (if there is any) upon demand flow d_{ab} among those produced by the receiving flow r_b of each possible destination link $b \in \mathrm{A}_a^+$ and by the turn capacities $\hat{q}_{ab}$. The resulting *sending flow share*, i.e. the share of demand that completes the desired manoeuvre is applied to all vehicles exiting from a single lane group to ensure the FIFO rule, and is given by

$$\rho_a = \min \left\{ 1,\ \frac{\hat{q}_{ab}}{d_{ab}},\ \frac{r_{ab}}{d_{ab}} \mid b \in \mathrm{A}_a^+,\ d_{ab} > 0 \right\} \quad . \tag{3.10}$$

When considering a generic node with both mergings and diversions, the resulting inflows and outflows are simply given by

$$\begin{aligned} e_a &= \sum_{b \in \mathrm{A}_a^+} d_{ab} \\ f_b &= \sum_{a \in \mathrm{A}_b^-} d_{ab} \end{aligned} \quad , \tag{3.11}$$

where all symbols have their usual meanings as introduced over this section and the previous.

In this simple case drivers are assumed not to occupy the intersection if they cannot cross it due to the presence of a queue on their destination link, waiting until the necessary space becomes available. In fact, node model implemented in TRE is also capable of addressing the deterioration of performances due to misuse of the intersection capacity and modelling vehicles sneaking out of queues (in an exception to the FIFO rule and (3.10)) if *their* destination is not blocked, as introduced in all due detail in [Tiddi, 2012].

3.3 GLTM as Flow Simulation

The network loading and flow propagation model is central to the optimisation process proposed in this work. So far, its position in the Dynamic Traffic Assignment has been

clarified, but it may be useful to recapitulate and formalise what its input and output are as a stand-alone *flow simulator* component; before proceeding to Section 3.4 where its role in relation to the Genetic Algorithm will be clarified. These are summarised in Figure 3.5 and presented in more detail in the following sections.

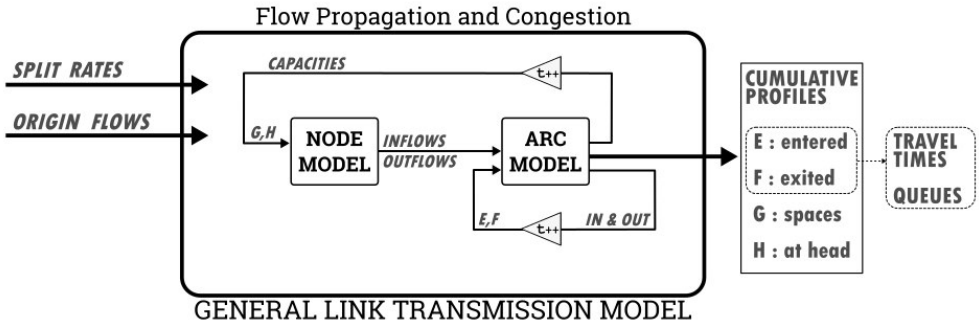

FIGURE 3.5 – Dynamic Network Loading Input, Output, and internal workflow of the algorithm. Origin flows determined by the departure model are propagated according to the split rates resulting from the user route choice model, producing cumulative profiles of the vehicles entering and leaving the arcs. These can be further processed to obtain arc travel times, which are then fed back to the demand model.

3.3.1 Simulation Input

The General Link Transmission Model operates on the basis of

splitting rates resulting from the aggregation of the dynamic route flows, used by the node model to distribute the outflow of an arc to its forward star;

origin flows representing the vehicles *injected* at specific network locations at every simulation interval, according to the demand data.

It is obviously important that the input data cover the entire span of the simulation, if realistic results are to be obtained. However, the time resolution of the input data is irrelevant and the algorithm can operate with constant values as well as weighted averages where the time intervals do not correspond; in fact, it is robust even with respect to incomplete input, since it may split flows based on the relative capacity of downstream arcs, and if demand flows are unknown they will simply be assumed to be zero.

3.3.2 Real Time Data Integration

The GLTM implemented in TRE can draw real-time corrections from the OPTIMA framework, which are based on harmonised data coming from a variety of public and private sources (loops, cameras, floating car data etc.) which are integrated into the simulation as:

capacity corrections when an accident, road closure or other modification to the supply is broadcast by the authorities or inferred automatically, and the fundamental diagram of the relevant arcs is updated;

speed corrections when the speed is measured or inferred, and either the fundamental diagram is updated to match the traffic state or a flow correction is applied;

flow corrections when a real flow value (often the outflow from an arc) is available and the simulated value overwritten.

These corrections are all applied in the inner loops of Figure 3.5, during the simulation intervals for which they are relevant. Their effect is then propagated in time and space, increasing the fidelity of the simulation to the real world traffic conditions.

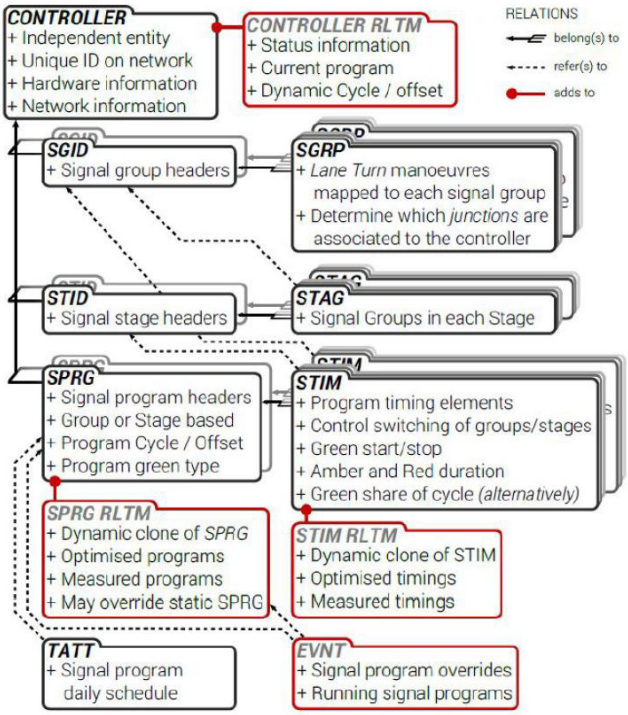

FIGURE 3.6 – Signals are modelled as time varying capacity reductions influencing the flow propagation. The central database contains all static signal plans and dynamic timing information, as well as all the necessary mapping of signal groups onto junction lane groups described in Chapter 1. The image shows the tables and relations that form the signal data model, their slightly cryptic four-letter-names a legacy of the coding style of the early developers of Optima.

3.3.3 Simulation Output

Strictly speaking, the results of the Dynamic Network Loading are, for every arc, the cumulative inflow and outflow profiles (namely F and E), the cumulate number of spaces available and vehicles that reached the head of each arc (respectively G and H) defined in Sections 3.2.3 and 3.2.4.

These values refer to *instants* of the simulation span, and as illustrated in Figure 3.5 can be readily used to obtain interval averages of the following quantities:

q_a **: flow** onto the arc during each interval in vehicles per unit time;

t_a^t **: travel time** that users entering during each interval will spend on the arc;

ω_a^Q **: queue** length given as average share of the arc length;

ω_a^n **: total vehicles** on the arc during the interval.

3.3.4 Optimisation Corridor

The present work aims to optimise signal timings in relation to the performance of what is usually called a *Traffic Corridor* or *Arterial Road*, referring to a stretch of road designed or happening to carry particularly high volumes of traffic.
While the concept is not strictly related to urban traffic, it is in the urban environment that traffic corridors most often suffer significant performance degradation due to congestion, aggravated by the numerous intersections with other busy roads where consistent traffic flows compete for the right of way and must be regulated by traffic lights.

The proposed optimisation method revolves around an *Optimisation Corridor* object that essentially implements the formalisation illustrated in Section 1.4.1. It consists of an *ordered set of links* connected head-to-tail, and may run through any number of signalised intersections: the problem size is then determined exactly, since the task at hand is simply to optimise signal coordination and each junction has a predetermined program that can only be offset in time.

This definition of corridor blends seamlessly into the Optima network model as well as in TRE result computation, and allows relevant key performance indicators to be calculated from continuous network loading results, i.e. the arc profiles just introduced in section 3.3.3: the process is fully detailed in Chapter 5 where performance indicators are discussed.

The corridor object represents the interface between the optimisation and simulation processes, and allows their separation (as may be more clear from Figure 4.3), leaving the possibility to exploit the work done in this context e.g. with different optimisation methods. Whatever the optimisation procedure, the corridor defines an additional input and output for the DNL. For a corridor with n arcs and m signalised junctions (see section 1.4.1):

- the **input** is a vector of m offset values, which affect the turn capacities used by the Node Model at the relevant junctions, altering the simulated flow propagation;
- the **output** is a vector of performance indices calculated for the n arcs of the corridor, which can be aggregated into global corridor performance indices.

Although for the rest of this dissertation only one corridor will be considered, the application might easily be scaled to multiple corridors or extended to sub-networks: such efforts are beyond the scope of this experiment but their potential is discussed in 4.3.2 alongside other scalability considerations.

Finally, it should be noted that links that are not strictly part of the corridor are *not* factored into the KPI computation. Some may be considered relevant, e.g. the inroads to the corridor; however, it is far from straightforward to *automatically* determine which links should be included based solely on the corridor definition, and in general there is no guarantee that the network model should be constructed in such a way as to render it possible at all. Although *in principle* some consideration for the consequences of choices made on the corridor on the neighbouring roads may help make better decisions, this would require preprocessing of the network and the associated complications are deemed unnecessary given the current task.

Return Corridor Definition

The return corridor cannot simply be defined as the sequence of links traversing the same nodes in reverse order: there is no guarantee that for any pair of subsequent nodes representing the tail and head of a given link there should exist another link joining them in the opposite direction (the network is a directed graph).

Furthermore, the two directions of a traffic corridor may well be modelled as completely disjoint sets of arcs, sharing no nodes between them.

The present approach can handle two-way optimisation without loss of generality in this respect: it is sufficient to define the return corridor in the exact same way as the primary direction, and to indicate it as *return* direction along with a weight coefficient, which can be used to scale KPI values to reflect its importance with respect to the main direction.

If the junctions traversed by the return corridor are handled by the same set of controllers as the main, the problem size remains the same and the extra computation time required to calculate the relevant KPI is negligible.
If more controllers are involved, they can be ignored (which makes little sense unless they actually cannot be controlled and adjusted remotely) or included in the optimisation, which will increase the solution space size and the time required to explore it.

3.4 Genetic Algorithm

A Genetic Algorithm is an evolutionary computation technique that explores a solution space by mimicking a process of evolution by natural selection. It is particularly suitable for heuristic optimisation approaches as it does not rely on *a priori* knowledge of the problem.

The present application for corridor offset optimisation represents a typical use case: it is easy to define the inputs (the offset values at each junction along a corridor) and straightforward to calculate the resulting performance indices (see Chapter 5), but the traffic propagation (i.e. *the function*) with its numerous parameters and complex interactions, cannot be inverted or described in closed form to try and approach the optimisation analytically.

Given a generic process or single valued function of n arguments, the Genetic Algorithm identifies candidate solutions as *individuals*, each characterised by a *chromosome* $\mathbf{s}$ which is nothing but a vector of n viable input values to the process. The candidate solution chromosomes are then fed through the process or function to determine the fitness f of the corresponding individual:

$$\omega_i = \phi(\mathbf{s}_i) \qquad \text{with} \quad \mathbf{s}_i = \left\{ s_i^1, s_i^2, \dots s_i^n \right\} \tag{3.12}$$

where the subscript $i \in I$ identifies an individual member of the population, i.e. a specific vector among the pool of candidate solutions whose components are referred to as *genes*.

The population may initially be randomised or otherwise generated. Through the fundamental operations of *selection, crossover* and *mutation* (detailed in the next few sections) the best individuals are allowed to live on and pass their genes to their successors, which gradually replace the lower ranking individuals.

Through iterations of this mechanism, illustrated in Figure 3.7, the population undergoes an evolutionary process whereby all individuals become (on average) better suited for the process considered: there is no guarantee that a global optimum will be found, but good-enough solutions for practical applications can be obtained in relatively few generations.

There is no generalised consensus regarding the optimal population size in relation to the problem size; if anything, researchers agree that no implementation of the Genetic Algorithm can be expected to work equally well with different problem types, and that some trial and error is always required in practice: [Eberhart and Shi, 1998] provide an insightful analysis of the matter.

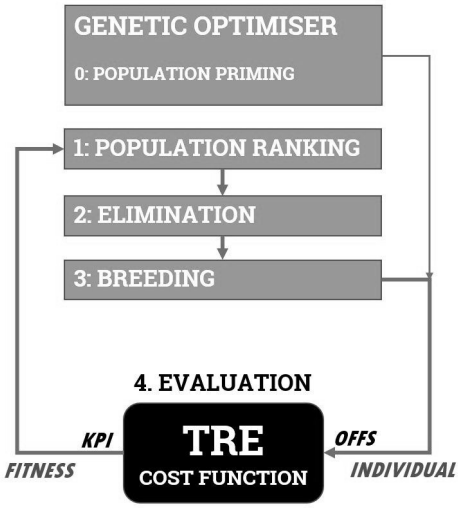

FIGURE 3.7 – The Genetic Algorithm: with each cycle, a new *generation* carrying the most successful traits of the previous one supplants the least successful individuals.

This work is no exception, and a study of the algorithm performance with different configurations is presented in the Results chapter. The next sections are dedicated to the formalisation of the genetic algorithm operators and will illustrate in more detail some of the design choices made for the current GA implementation.

3.4.1 Evolutionary Operators

Selection

Selection is the process whereby the survival and breeding chances of an individual are determined based on its fitness. Selection for survival is necessary because some individuals must be eliminated from the pool to make room for the new generation, and is generally applied before the other for obvious reasons, although this is not strictly necessary. Selection for breeding further enforces the inheritance of the *best* genes to the new generations. They are applied in this order in the GA implemented for this work, and will be presented accordingly.

Considering the population I_g at generation g, the selection process κ determines the subset I_g^* that survives to maturity based on the individual fitness values $\mathbf{f}$

$$I_g^* = \kappa\left(I_g, \mathbf{f}\right) \quad , \tag{3.13}$$

then the the breeding chance of each surviving individual $\mathbf{p}$ may be calculated by an independent process β

$$\mathbf{p}^\beta = \beta\left(I_g^*, \mathbf{f}^*\right) \tag{3.14}$$

where all terms have the same cardinality equal to the number of individuals in I_g^*.

In this instance, the selection function ϕ takes the form of a dynamic step function allowing an arbitrary percentile of the k fittest individuals to make it to adulthood; the breeding chances are then determined by a linear function of the individual ranking, which can be adjusted via the ratio p_1/p_k, expressing how much more likely the top individual is to breed with respect to the least fit surviving one.

Crossover

The combination of two individuals to generate a new one is inspired by the naturally occurring event of genes *crossing over* between chromosomes during meiosis in sexually reproducing organisms. Given two chromosomes with n genes, and assuming only one random crossing-over locus x, the crossover operator ξ used to produce a new one may be formalised as follows:

$$\xi(\mathbf{s}_1, \mathbf{s}_2) = \left\{s_1^1, \ldots, s_1^{x-1}\right\} \cup \left\{s_2^x, \ldots, s_1^n\right\} \qquad \text{with} \quad x \in [1, n+1] \quad , \tag{3.15}$$

meaning that the resulting chromosome inherits the genes from one parent up to a random position, and the rest from the second parent. The same principle may be intuitively extended to multiple random crossover loci.

Mutation

Mutation is the process whereby a random gene on a solution chromosome changes value. This introduces variability in the population that is not directly related with fitness: on one side, this is beneficial as it prevents to some degree that a sub-optimal solution should take over the entire population; conversely, too high a mutation rate may end up compromising otherwise successful individuals, with the risk of eliminating positive traits from the gene pool.

To get the best effects while circumventing the downsides, this application uses variable mutation rates, starting out at an empirically determined safe value which is increased proportionally with a measure of the *similarity* amongst the top individuals. In this way, mutations increase with the risk of stagnation and help counteract it, but the replication of successful traits across many similar individuals limit the possibility of weeding them out.

3.4.2 Initial Population Seeding with Slack Bandwidth

The speed of convergence of the Genetic Algorithm is strongly influenced by the initial population. Theoretically, an infinitely large random population would contain the global optimum right from the first iteration, but with any manageable number of candidate solutions the chance of having a randomly generated solution performing well becomes very slim.

Depending on the problem size, there is a chance that a single random solution may be rather near an optimum, but in a randomly generated gene pool it will struggle to find a worthy partner, and most crossover operations will result in low fitness individuals with the exception of those which happen to inherit most genes from the successful one (which increases the average population fitness but almost exclusively through loss of diversity). This leads to slow performance improvements over the first iterations, and rather unpredictable results in the long run.

By priming the algorithm with a population of selected individuals that can be reasonably expected to perform well, obtained by some fast and cheap approximation of the problem, it is possible to greatly improve the initial performance; the downside being the risk of *driving* evolution too hard into a local optimum.

Based on the results presented in the Results chapter, section 6.4.1, it was determined that the best results for the problem at hand could be obtained by priming the population with solutions derived from the Slack Bandwidth approach presented in section 1.4.3.

To maximise the chance of obtaining good solutions right from the first generation while reducing the risk of driving the algorithm into a local optimum, the maximum bandwidth solution was cloned into 50% of the initial population while applying small random mutations and a constant shift to all values, so that the *relative* offsets (which can reasonably be expected to be nearly correct from the geometric method) could be phased over the entire signal cycle. This introduces a certain degree of diversity in the population, complemented by the remaining 50% of the initial population generated randomly.

Chapter 4

A Real-Time Forecast-Based Optimiser

The foremost aim of this work is to exploit the versatility and speed of advanced macroscopic traffic simulation to bring forth a heuristic approach capable of improving signal plans *in real time* using live reliable data and dynamic traffic forecast.
It represents an attempt to bring together the best of most signal setting approaches introduced in the previous chapters, in that by integration within the real-time traffic management environment outlined in section 3.1 it aims to be:

adaptive – since real-time operation should guarantee a degree of adaptivity so far only expected of actuated signals, besting other plan-generating systems particularly in terms of response times;

accurate – thanks to the detailed network and traffic propagation models provided by the traffic management environment, coupled with solid real-time data, which enable it to operate on more reliable assumptions about traffic and its movements;

impartial – by relying on an objective-driven heuristic search method to avoid the simplifications involved in a strictly analytical approach, behaving like a feedback controller and accounting for the short-term consequences of its decisions, rather than making assumptions about the best way to operate signals optimally;

versatile – because once the principles of operation are proven sound and the system integration is functional, the same can be used to approach more complex optimisation problems, operate on longer time scales or be used as powerful offline planning tools;

scalable – by relying on task distribution and parallel computing.

The proposed *active signal control* approach uses a Genetic Algorithm coupled with a superior macro-simulation traffic forecast engine to generate and select candidate signal timings, gradually guiding their evolution towards a global optimum that yields the best network performance on the affected area.

Previous chapters introduced the different components that allow the real-time optimisation, from the Dynamic Traffic Assignment engine to the Genetic Algorithm itself, detailing their inner workings and the importance of their contribution. The most relevant known signal optimisation approaches, a study of which drove the design of this unprecedented alternative, were also presented.

This chapter illustrates the approach in detail, describing how the different components come together and communicate to provide signal timings for optimal arterial traffic control.

4.1 Heuristic Offset Optimisation

The task at hand is to develop an optimiser that can choose the timing *offsets* (see section 1.2) between a group of adjacent signalised junctions that regulate traffic progression along a *corridor* as defined in section 1.4.1.

The problem of arterial *coordination* has been tackled in a variety of ways, of which some are presented in section 1.4: mostly, as the relevant scientific literature testifies, analytical approaches to the problem have been sought which revolve around the concept of *bandwidth* as the driving metric. These stem from the extremely reasonable assumption that to increase the chance of encountering a *green wave* along a traffic artery should ultimately mean to maximise its throughput; they are also a testimony to the extreme difficulty of encapsulating the complex dynamics of traffic itself into a closed-form analytical formulation that would be nearly as elegant as those that can be built upon the relatively simple paradigm of bandwidth, discussed in detail in section 1.4.2.

Although moving *from* the idea of bandwidth maximisation, as illustrated in this chapter, this work aims to do *away* not only with the search for green waves but with the very need to explicitly model the correlation between signal offsets and arterial traffic fluidity.
The idea is to rely on simulation to verify *a posteriori* whether certain timing choices bring about an improvement in performance of the corridor, and to what extent: this allows to concentrate on the *results*, which can be assessed according to any chosen metric and may well arise from less obvious decisions than those which would only seek to maximise the throughput.

The one presented in this work is therefore in essence a *heuristic* approach to arterial coordination, which should be able to operate in a wider range of traffic conditions thanks to the Dynamic Traffic Assignment algorithm described in Chapter 3 to improve articulated and sustainable performance objectives. While the objectives driving the optimisation deserve an in-depth discussion, presented in Chapter 5, the next few sections are concerned with the practical aspects of the implementation of the proposed method.

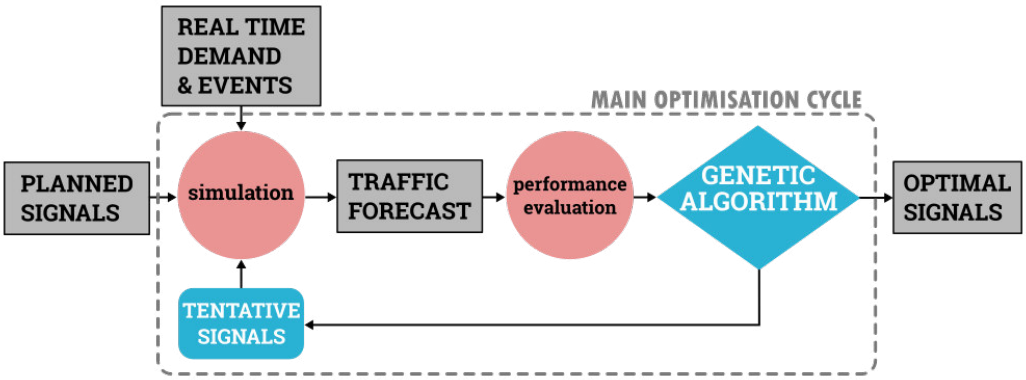

FIGURE 4.1 – The optimisation relies on accurate propagation of real-time traffic data to assess the benefits in the near future of potential modifications to the scheduled signal plans. The performance of the corridor is evaluated on the basis of the resulting forecast, which serves as a ranking metric for the evolutionary algorithm to generate better offset combinations until the optimal solution is found.

4.1.1 Rolling Look-Ahead Window Optimisation

Two of the strongest features of the heuristic optimisation presented are the capability to account for transient traffic phenomena (such as the gradual build-up or dissipation of queues over a number of signal cycles) and of future events which may be known in advance (such as road closures and deviations). In order to fully exploit these advantages, the optimisation is performed by evaluating traffic conditions as they develop over a *rolling look-ahead window*, i.e. a time span in the order of a few signal cycles that is completely in the future with respect to the real time during which the optimisation occurs.

The boundary conditions at the beginning of the time window are known, and the optimiser evaluates the corridor performance arising from the predicted demand and tentative signal timings until a satisfactory choice is found, as illustrated in Figure 4.2 and discussed in more detail throughout this chapter.

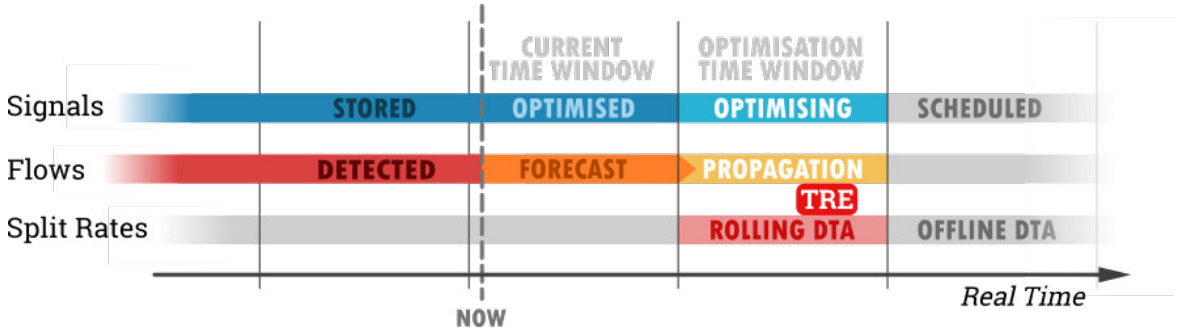

FIGURE 4.2 – Rolling Optimisation is performed in real-time on a time window in the near future, while the previous (now current) time window plays out and results of the corresponding optimisation are implemented on the street-level equipment.

On one hand, the look-ahead window allows to account for the short term effects of signal timing choices, protecting the optimisation from *greedy* solutions that may promote a fast progression only to cause graver congestion down the line; but also to consider events that may radically change the outcome of a given set of timings.
These beneficial effects would call for the look-ahead window to be as long as possible, for the optimisation to be best *informed*.

On the other hand, however, since signal timings are constant over the optimisation window, increasing its length reduces the responsiveness and adaptivity of the system: the point is to find the best timings for very specific traffic conditions as they arise, and it would be counter-productive to allow them to change significantly over the evaluation period, thus confusing results.

Another limiting factor is imposed by the computational requirements of the simulation, which grow more or less linearly with the number of intervals that need to be computed, as should be evident from the outline of the algorithm given in section 3.2: for constant time resolution, a longer look-ahead window takes longer to simulate, and it needs to be simulated in the order of a few hundred times for the genetic algorithm to yield significant improvements in performance.

Time resolution cannot be sacrificed, as it is crucial to the correct reproduction of the rapid within-cycle queue dynamics, which are one of the main components of corridor performance indicators and would get averaged out by longer simulation intervals. For the intended real-time operation, the one on execution times is a particularly stringent limitation.

Sizing of the look-ahead window should strike a good compromise between execution times, responsiveness and control feedback, and values in the order of 10 minutes, or 5-10 signal cycles should be optimal for most applications, if the computing resources available allow it. Rough sizing calculations are presented in the relevant section (6.2) of the Results chapter, alongside performance results for the validation tests performed.

4.2 TRE as Performance Function

The Dynamic Traffic Assignment algorithm known as TRE (described in more detail in section 3.2) is used in this integration to provide the optimiser with the solution evaluation capabilities it requires for its own stochastic search algorithm.

It should be clear by now that the task of optimising offsets *per se* does not require knowledge of the entire network, of demand profiles or of the events that may modify one or the other: the optimisation loop strictly entails solution generation and evaluation.

TRE can then serve as a *single point of contact* between the traffic management system and the offset optimiser, to which it delivers all the precious information available in the only form that is really useful, i.e. that of accurate predictions about the outcome of a choice of signal offsets. This is illustrated in Figure 4.3.

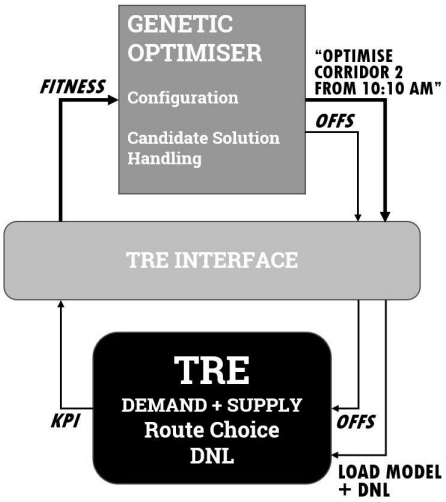

FIGURE 4.3 – Interaction between the optimiser and TRE

To be more precise, it should be specified that in order to perform the rolling real-time optimisation illustrated in Figure 4.2 TRE must carry out a few operations beyond the simple traffic propagation that directly yields the fitness values for the offset solutions. It may be useful to exemplify the optimiser operation in relation to a single corridor and examine the different phases of the rolling optimisation cycle, which can be summarised as follows.

Step 0 - start: when the optimiser is first launched, the only parameter it requires beyond its own algorithm configuration is the index of the corridor to be optimised. TRE starts with its own configuration, loads the model and returns to the optimiser what little corridor data it requires: number of junctions and cycle length at the very least, plus the slack-band starting offsets (refer to section 1.4.3) for GA population priming if desired (which can be calculated in seconds as a property of the corridor).

Step 1 - DTA: TRE performs a full dynamic user equilibrium assignment from the current time t_0, covering a span of *two* optimisation windows $[t_0, t_1] \cup [t_1, t_2]$ into the future, and saves:

- the flows onto and out of the cordon (access and egress) links of the corridor,
- the turn rates (averaged over the entire period) that determine the splitting of flows at each intersection all along the corridor
- the vehicles present on each arc at the beginning of the next window, i.e. t_1

obviously starting itself with a *loaded* network if arc occupancy data is available for the initial instant of the DTA span.

Step 2 - DNL: using the split rates just calculated, TRE propagates flows very rapidly over the *optimisation window* $[t_1, t_2]$ for each set of offsets proposed by the optimiser, returning the desired performance indices: this is not an equilibrium assignment, and is performed *only* on the corridor and cordon links.

Step 3 - finalisation: when the optimisation objectives are satisfied or the current time window is almost elapsed, optimised offsets are sent to the data model to be included as *future events* by subsequent simulations, and handled as appropriate by Optima for their implementation on the field.

Step 4 - rolling forward: with the new offsets finalised and the state of the network known, TRE can go back to Step 1 and perform a new DTA, so that the *next* window $[t_2, t_3]$ can be optimised while the one just finalised $[t_1, t_2]$ plays out in the real world.

In this way, TRE can rapidly evaluate hundreds of solutions using the most up-to-date supply and demand information. Details on the most crucial steps 2 and 3 are given in the next sections.

4.2.1 Network Wide DTA

The Dynamic Traffic Assignment provides the basis in terms of origin flows and split rates for the evaluation of candidate solutions, and must be performed on the whole network to fully exploit the advantages of real time traffic forecast. It is the single most time-consuming task required for the optimisation of each time window, as it involves the iterative algorithm described in section 3.2 that searches for the User Equilibrium condition.

The latter is considered satisfied when the user route choices (depending on the arc costs, including their travel times) are in equilibrium with the arc costs (determined by the collective route choices of the users).

The simulation engine can decouple the route choice model intervals from the flow propagation model to greatly reduce execution times, so while the latter must be numerous and short due to resolution requirements and inherent limitations of the *General Link Transmission Model* (see section 3.2.2), the route choices can be averaged over the whole time window. This is a more than reasonable assumption, and in fact reflects rather well the fact that over a relatively short time window (in the order of 10 minutes) the ratio of vehicles getting on and off the corridor at each junction with respect to the total flows can be expected not to change significantly.

The results of the DTA Equilibrium step of the optimisation are therefore:

- **initial conditions** in the form of vehicles already on corridor and cordon arcs at the initial instant of the optimisation window;
- **flow profiles** that enter the corridor's cordon arcs, with a fine ($\sim$ 1 s) time resolution over the entire window;
- **split rates** for all diversion nodes in the corridor, constant over the simulation window.

which are the input for the Dynamic Network Loading.

4.2.2 Solution Evaluation with DNL

The Dynamic Network Loading algorithm handles the propagation in space and time of traffic flows, across the network and the optimisation window respectively. As detailed in section 3.3, it is also responsible of implementing the effects of time-dependent phenomena such as the capacity reductions administered by *traffic signals*, producing a detailed forecast of the evolution of traffic which includes congestion, queue formation and spillback.

In order to evaluate the effects of a choice of signal offsets, DNL can be performed for each candidate solution proposed by the genetic algorithm, covering all relevant arcs (a subset of the network that only includes the optimisation corridor and its cordon arcs) for the entire span of the optimisation window with a fine time resolution, in order to capture transient traffic phenomena.

The result are cumulative profiles of the vehicles entering and leaving each arc, which are easily processed to obtain queue lengths and travel times and hence Key Performance Indicators for the corridor operating under the given signal timings. These are returned as the *fitness value* used by the genetic algorithm to rank tentative offset solutions.

4.3 Performance and Scalability

Computational efficiency is of the utmost importance in a real-time environment. As already mentioned, the aim of this optimiser is to operate in *rolling horizon*, meaning that each optimisation must be carried out in a limited time, while the results of the previous are implemented. This means that, beyond all considerations about optimisation window sizing, the likely limiting factor will be the lower bound imposed by the time it takes for the optimiser to reach performance improvements that justify the effort: this will be already in the order of a few signal cycles; if resources are in excess and there is no reason to allow the genetic algorithm a longer time to optimise, the window length may then be increased slightly to enhance the look-ahead capability.

Fortunately, as detailed in the previous sections, the single most time consuming task is the initial network-wide equilibrium assignment (DTA phase) which must be performed only once; furthermore, the following circumstances alleviate its computational cost:

- the route choice part of the algorithm is extremely time consuming and scales badly with the problem size, but it is performed over very large time intervals, and therefore once or twice at most per optimisation;
- both the route choice and the dynamic network loading can be *significantly* sped up by parallelisation of the route choice and network loading algorithms: as attested in our [Attanasi et al., 2015] and further discussed in section 4.3.2.

Solution evaluation then relies on several short DNL performed on a comparatively *minuscule* network, and a single TRE instance is already capable of effectively optimising a 10' window in rolling horizon on an ordinary computer.

The next few sections will go into more detail about the communication between the different components concurring to the corridor offset optimisation, showing how physical separation of tasks between different machines is not only possible, but can lead to significant performance improvements that should open up several venues for broadening the scope of this initial proof-of-concept application.

4.3.1 Calling Method and Data Exchange

The Optima traffic management environment (section 3.1) is, in general, a *distributed* system: the various software components illustrated for example in Figure 3.1 can be and often *are* in practice located on different machines, communicating with each other over internet protocols and exchanging data via the Traffic Data Exchange central database.

This is an advantage, as it allows to distribute tasks across different inexpensive machines, but the cost of communications can quickly become prohibitive if the payloads are too large or if they need to be sent too often: even over the fastest networks available, data travels several orders of magnitude slower than it can be shared between processes on the same machine, and the connection overhead on TCP connections means that any time-critical repeated action should *not* entail creating a new connection.

Fortunately, with reference to the rolling optimisation described in section 4.2 of this chapter and a peek-ahead to Figure 4.4 it is plain that the amount of data exchanged between the different components concurring to the optimisation is more than manageable, and are summarised in Table 4.1 for each optimisation phase (message headers not considered).

The above rough estimates suggest that the optimiser can easily be de-localised with respect to the machine running TRE, which in turn potentially allows to increase the number of simulator instances running in parallel to process the optimisation tasks faster, as discussed in the next section.

4.3.2 Task Parallelisation

Although the DTA phase of the optimisation process cannot be broken down or distributed across different TRE instances to speed it up, the TRE algorithm can fully exploit the computing power of a single machine thanks to multi-threading: in our article [Attanasi et al., 2015] it is shown how *near-linear* performance gains can be obtained by increasing the number of threads (i.e. DTA execution times are almost inversely proportional to the number of cores). Greater returns are obtained for larger, more complex networks, which benefit the most from running on more and more threads; results from the original article for up to 16 threads on large real-world networks are presented in Appendix A.

The performance benefits are obtained by distributing the fundamental tasks that make up the DTA algorithm across all available processor cores:

- during the **route choice phase**, parallel threads handle the *serial* A* single-source Dynamic Shortest Path searches that must be performed for each network zone for O-D demand routing (which is *much* faster than a *parallel* A* search);

TABLE 4.1 – DATA TRAFFIC DURING OPTIMISATION

	Data Exchange	Payload Size
Startup:	TRE loads the network from the TDE database: this may be lengthy but it is only done *once* on startup and doesn't weigh on optimisation	N/A
Step 0:	the optimiser sends a single integer corridor index plus the time window boundaries to start TRE, receiving the cycle length, the number of junctions $n = \|C\|$ plus n geometric offsets for population priming	$< 1kB$
Step 1:	TRE reads real time data for the DTA then writes relevant results, once per window	$\sim 1 - 3$ Mb
Step 2:	at each Genetic Algorithm generation, if the latter and TRE are on different machines, they exchange a batch of candidate solutions for as many performance indices	*population size* $\times$ $(n+1)\times 2$ bytes $\simeq$ 10 kB
Step 3:	the final offsets are sent to TDE	$n \times 2$ bytes
Step 4:	the optimiser sends the new time window start time and if necessary receives a new set of slack-band offsets	$8 + n \times 2$ bytes

- during the **network loading phase**, threads perform node model calculations in parallel then proceed to propagate flow states on the same node's backward and forward star links.

Under these premises, and bearing in mind the necessary data exchanges summarised in Table 4.1, a few task parallelisation options appear feasible for future up-scaling of the application, as seen in Figure 4.4. It should be noted that from the *optimiser* point of view the only relevant parameter is the problem *size* i.e. the number of variables and the search space they span, while the *type* of problem and the size of the area of interest affect the load on the simulator.

Depending on the problem type, and in all cases exploiting the extant possibility of running different TRE instances on different machines, the options to reduce the time required for optimisation are summarised as follows.

Distributed Serial DNL: this is the simplest case, involving the least data transfers and additional machines; best suited to all cases where the DNL is simple enough (< 1000 links) that it would not benefit from parallelisation. The possibility to evaluate *batches* of candidate solutions in parallel using *serial* DNL on different single-thread TRE instances may be used to improve the performance of an analogous optimisation to the one presented here, or to tackle a slightly more complicated application such as could be a corridor optimiser that can also determine phase green shares.

Parallel TRE Cluster: if the DNL complexity is such that it *would* significantly benefit from multi-threading – e.g. for performance evaluation over large network portions – a cluster of independent machines running multi-thread TRE would add to the benefits of the simpler solution (evaluation batching) the possibility to exploit all processor cores, to evaluate *each* solution in a fraction of the time.

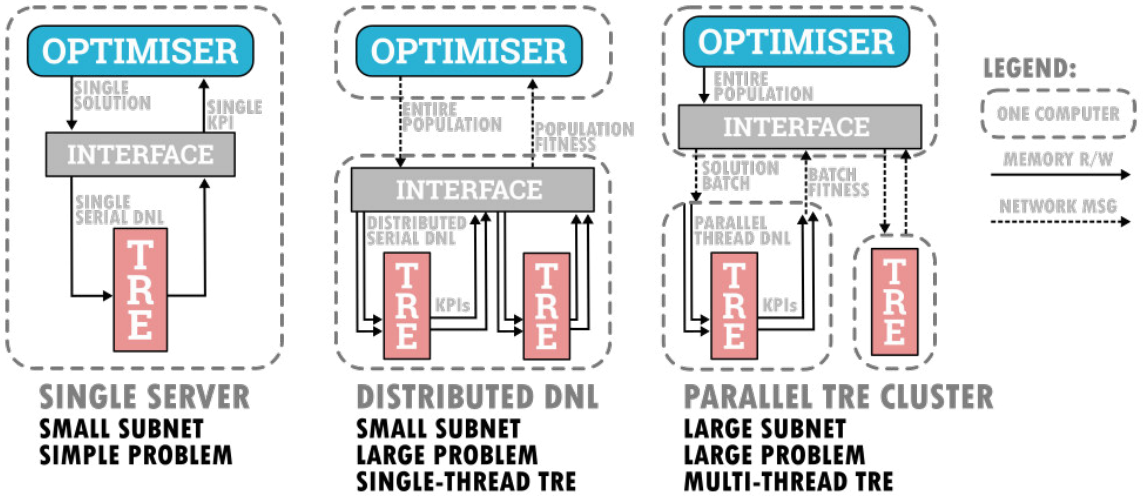

FIGURE 4.4 – Parallelisation Options: several single-thread TRE instances running on the same machine can speed up solution evaluation by distributing the load without competing for processor cores; by increasing the number of machines TRE can *also* deal with more demanding DNL in a short time by fully exploiting multi-threading.

Notice that for reasonably sized problems and sub-networks, ordinary inexpensive four-core processors would already be more than enough to reap all the available benefits of parallel evaluations or multi-thread DNL. In any case, the DTA should be performed on a single machine (the best performing, if there is one) to obtain corridor flows and initial conditions for all consumer TRE instances tasked with solution evaluation.

These parallel computing solutions could be easily implemented with a relatively small development effort, but could further increase the performance of this solution far beyond that of the already viable case presented in this work.

Chapter 5

Smart Objectives

The very idea of optimisation cannot transcend the definition of its objectives: in fact, any search for an *optimum* in any context first requires a clear answer to the question:

what *is* good?

This chapter presents a few fundamental questions that must *lead up* to the definition of the optimisation objectives, to point out what might have gone wrong in the past and illustrate how simulation-based optimisation might suit future developments in traffic optimisation.

Finally, the network performance indicators used in this study are presented and discussed in relation to the model outputs described in Chapter 3.

5.1 The Optimisation Dilemma

When it comes to making choices about the development of the spaces we live in, and compromising between opposing interests, any responsible policy maker will be afflicted by the nagging doubt: *will this actually be good for us in the long run?* Our recent history is full of sad examples of daft resource allocation driven only by someone's good will to improve our lot. Dishonesty and deceit notwithstanding, the task of determining far-sighted policies truly in the best interest of the majority of people (let alone of the only habitable planet we know of) is a colossal one, and riddled with contradictions to boot.

Although the difficulty of *identifying good* permeates almost every aspect of the human experience, it can be examined in this context with a most fitting example. The development of transport systems, hand in hand with the evolution of the very idea we hold of *human mobility* (i.e. how far and fast we think we should be able to travel), has seen a vertiginous acceleration over the last century, propelled by technological progress and economic growth. It serves as a stark example of what, with the best yet sorely misguided intentions, we can end up doing to ourselves: ancient cities are eroded and clogged by grinding traffic, more modern ones *sprawl* for miles in a self-fuelling flight for space (a phenomenon distinct from simple urban growth whose causes and consequences are object of several socio-economic studies, as condensed and analysed e.g. in [Brueckner, 2000]).

The century of the car has taught us to make better, wider roads, to grant *everyone* the right to faster and more efficient private transport, for the best of all... and left us with inhospitable cities where humans have to contend their living space with (mostly stationary)

cars, as public service is dismantled, distances grow and walking ceases to be an option. Most ironically, the very cars to whom we handed over our cities, have nowhere to go, and tax our waking time with dreadful walking-pace commuting and prowling for parking: it comes as no surpreise that the stress to daily commuting and urban congestion have been proven in many studies to have a direct correlation to public health issues such as obesity and hypertension [Ewing, Schmid, Killingsworth, Zlot, and Raudenbush, 2008], [Lopez, 2004].

It is only natural then, when talking about signal optimisation, to be aware of not one but several *elephants in the room*:

- signal optimisation is sound in principle, as it represents a way to maximise *efficiency* of the already existing infrastructure, minimising the *waste* of time: unfortunately, considering the fact that demand for private transport always closely matches the supply [Linda, 2003], and that it is in direct, unfair competition for space with its more sustainable alternatives [Winston, 2000], any improvements to the supply tend to be quickly saturated, turning as a matter of fact into an overall *loss*;
- not only to ease private traffic is of dubious benefit to our cities, health and safety; there isn't even a clear picture to understand whether the approaches followed thus far (e.g. bandwidth maximisation) are really beneficial to traffic fluidity in the short term, or if the worshipped *green waves* lead to *worse* traffic conditions in the more critical areas of urban networks.

It is true that it is extremely hard to model these effects explicitly, which is why this work aims to contribute to the affirmation of heuristic approaches that may help investigate the consequences of different choices, and maybe reverse-engineer better traffic control policies. It is also why, rather than tackling the rather more daunting challenge of *re-defining good*, this proof-of-concept revolves around rather familiar performance indicators such as average progression speed and queue length.

It is of *fundamental* importance, however, to understand that at least conceptually this approach does not make *assumptions* about what the best design policies should be, but focusses on the *results*.

Putting aside the cold hard fact that the only way to alleviate traffic is to *reduce it* (as many cities around the world are finally doing, with *well quantifiable* economic and psychological gains [Flusche, 2012], the plan is to try and see if at least we can be sure to actually *reduce* the short term discomfort and externalities by increasing the efficiency of the existing roads with low-cost, non invasive infrastructure upgrades.

If the technology should prove effective, the object of its future developments might then well be to use simulation to enable *long term* traffic control policies and define the objectives of optimisation in terms of modal shift to cycling and public transport, and in general tie them to the regaining of public spaces.

5.2 Optimisation Objective Functions

This section presents the objective functions used during this study to drive the Genetic Algorithm and obtain the results presented in Chapter 6.

They reflect very intuitive and down-to-earth objectives not dissimilar to the aims of classical optimisation techniques based on the analytical (and rather simplified) representations of congestion phenomena illustrated in Chapter 1:

- *minutes per kilometre travelled*, as a user-centred measure of discomfort;
- *stop ratio*, the minimisation of which is the presumed outcome of bandwidth maximisation;
- *relative queue length* as a measure of congestion, spillback and risk of gridlock.

These however are obtained directly from aggregation of the network performance model results (defined in section 3.3.3) thanks to existing and purpose-developed KPI calculation features of the simulation engine, as illustrated in the sections to follow.

5.2.1 Fundamental Quantities

The following quantities are calculated on the corridor over the entire simulation, and represent the reference quantities for calculation of key performance indicators.

The subscript T is often dropped for readability, but is *implied* for all quantities aggregated at the simulation span level and presented in the following section.

Section and Corridor Total

The *section total* is defined as the integral of the inflow to a given section of the corridor $a \in \mathrm{C}$, obtained piecewise in this case, as the total number of vehicles entered during each interval of the simulation window:

$$\omega_a^n = \sum_{\tau \in \mathrm{T}} q_{a,\tau} \Delta t^\tau \quad . \tag{5.1}$$

The *corridor total* gives an aggregate measure of how frequented the corridor is on the whole: it does not carry information on which sections are busier, but accounts for all vehicles that accessed *any* section during the simulation.

It is obtained as the cumulative total over all corridor sections, according to

$$\omega_{\mathrm{C}}^n = \sum_{a \in \mathrm{C}} \omega_a^n \quad . \tag{5.2}$$

Notice that ω_a^n implies no distinction based on whether the flows are coming from the previous section of the corridor or from a cordon arc, therefore vehicles travelling on more than one section are counted several times. This reflects the fact that the corridor is being used *more* if vehicles travel a greater portion of it than if they only were to use one section.

The total inflow index ω_{C}^n covers an important role as a *checksum*, since it ensures that any improvements in other cumulative indices are not really due to the corridor accepting fewer vehicles because of a deterioration in the traffic conditions.

Geometrical Hypercritical Queue Length

The General Link Transmission Model provides a convenient and plausibly accurate linear approximation of the queue length, which is obtained at every simulation interval from the comparison of cumulative profiles on each arc (see section 3.2.3). In particular, the difference

between the vehicles that reached N_a^+ (i.e. the head of link a) and those that were able to leave it before time t can be noted as the *vertical queue*

$$n^Q_{\mathrm{N}_a^+,t} = n^H_{a,t} - n^E_{a,t} \quad , \tag{5.3}$$

which is simply a vehicle count and would theoretically have *zero* length by definition.

To obtain an estimate of the real queue length, (5.3) is considered in conjunction with the link *receiving capacity*, determined in turn by the difference between the spaces that reached the link tail and those already consumed by vehicles entered before the current instant, i.e. $n^G_{a,t} - n^F_{a,t}$.

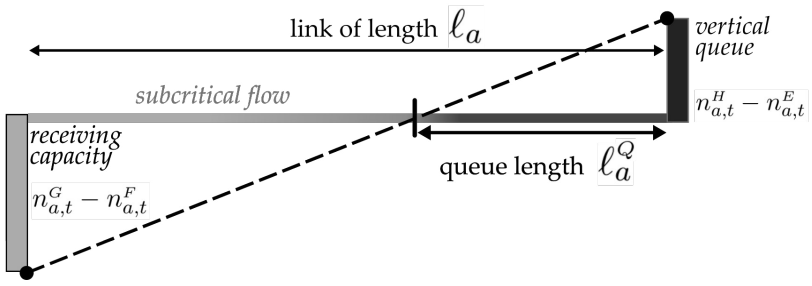

FIGURE 5.1 – Geometrical approximation of the hypercritical queue length.

As shown in Figure 5.1, they yield an estimate of the queue length as the *position* along the link at time t of the shock wave separating the oncoming subcritical flow from the prevailing hypercritical flow state propagating backwards from the exit section, found at the section where the hypocritical and hypercritical cumulative flows are equal. It is then assumed that vehicular density is constant in each of the two regions, and that the *back of the queue* (which is not necessarily at a standstill) corresponds to the shock wave position:

$$\ell^Q_{a,t} = \frac{\ell_a}{\dfrac{n^H_{a,t} - n^E_{a,t}}{n^G_{a,t} - n^F_{a,t}} + 1} \quad . \tag{5.4}$$

This approximation provides substantially more realism in the representation of queue propagation (blocking back) and dissipation compared to other existing macroscopic DTA models, as detailed in the original work by Yperman [2007], and serves the purposes of this application much better than the vertical queue paradigm used in many of its commercial alternatives.

User Time Spent and User Time Travelled

The most direct way to calculate how much time is spent by users on the corridor during the simulation window is to integrate the total number of vehicles present on any section over all time intervals. The total and section *User Time Spent* can be expressed as

$$\omega^t_{\mathrm{C}} = \sum_{a \in \mathrm{C}} \omega^t_a \qquad \text{where} \qquad \omega^t_a = \sum_{\tau \in \mathrm{T}} n_{a,\tau} \Delta t^\tau \tag{5.5}$$

therefore accounting for any vehicles already on the corridor at the start of the simulation, but not for the time that will be spent to get out of it beyond the end of the look-ahead window. However, since it is impossible to know how much time the vehicles have *already*

spent on the corridor when the simulation begins, nor how far they have got down the arc they're found on, the time spent ω^t is not suitable for estimating the corridor performance with respect to travelled distances.

Disregarding the initial vehicles n_C and only considering flows that enter a corridor section during the simulation, it is possible to extrapolate from the results exactly how much time those vehicles will spend *travelling* the length of each arc, even beyond the end of the simulation. The average *User Time Travelled* is still a measure of time, but obtained from flows and travel times as

$$\omega_a^{tt} = \sum_{\tau \in \mathrm{T}} t_{a,\tau}^t \; q_{a,\tau} \; \Delta t^\tau \quad . \tag{5.6}$$

5.2.2 Performance Indicators

The performance indicators used to test the functionality of the present optimiser implementation are described in the following sections. An analysis of the optimiser behaviour under the single-purpose drive of each of them is presented in section 6.1 of the Results. This provided a good understanding of their applicability and allowed to isolate their effects, while it was not deemed necessary to explore composite performance functions at this stage; although it is understood that any aggregation (e.g. a linear combination with weights) of the presented performance indices, or indeed of the model output in general may be used to fine-tune the optimiser response and address any specific scenario.

Minutes per Kilometre Travelled

From the user point of view, it makes sense to evaluate the performance of the corridor by considering the time required to travel the desired distance.

Referring to the User Time Travelled ω_a^{tt} expressed by (5.6) the *Minutes per Kilometre* cost function

$$\omega_{\mathrm{C}}^T = \sum_{a \in \mathrm{C}} \frac{\omega_a^{tt}}{\ell_a \quad \omega_{\mathrm{C}}^n} \cdot \frac{1000}{60} \tag{5.7}$$

uses the travel times experienced by all users, normalised with respect to the relevant section lengths and averaged over all vehicles involved with any part of the corridor during the simulation. This gives an overall measure of the fluidity of traffic on the corridor, and has the dimensions of a time per unit length. The choice of units (and name) for this performance indicator is therefore dictated solely by human-readability: it makes sense to count minutes spent in traffic to cover one kilometre, and it is easy to refer to the fact that for an average speed of 60 km/h the value of ω^T would be 1.

Congestion

To measure the state of congestion along the corridor on the whole, the queue length ℓ^Q expressed by (5.4) is normalised with respect to the arc length, i.e.

$$\omega_{a,t}^Q = \frac{\ell_{a,t}^Q}{\ell_a} \quad . \tag{5.8}$$

This yields a useful measure of the severity of a queue in relation to the risk of *gridlock* it entails: the network may perform reasonably well even under critical flow conditions, but a

queue *spilling back* to block egress from a junction will lead to the collapse of that junction and start blocking significantly the entire node backwards star.

To drive the optimisation by this metric means focussing on queue control: in particular, if the aim is to minimise the entity of all queues with respect to the capacity of the links they affect, the following metric may be defined

$$\omega_{\mathrm{C}}^{Q} = \frac{1}{|\mathrm{T}|} \sum_{t \in \mathrm{T}} \sum_{a \in \mathrm{C}} \frac{\omega_{a,t}^{Q}}{|\mathrm{C}|} \tag{5.9}$$

to try and equalise the congestion levels on all arcs, favouring a gating behaviour whereby queues are allowed to build up more on longer arcs.

This however proved rather unstable, as it was still possible for a short arc to be the *only* one experiencing severe spillback and for the effect to be averaged out by all other relatively free flowing corridor sections. The inner normalised sum in (5.9) was therefore swapped for a maximum function to focus on the *worst* congestion occurring along the corridor at each simulation interval

$$\omega_{\mathrm{C}}^{Q} = \frac{1}{|\mathrm{T}|} \sum_{t \in \mathrm{T}} \max\left(\left\{\omega_{a,t}^{Q},\ a \in \mathrm{C}\right\}\right) \tag{5.10}$$

thus implicitly enforcing queue equalisation without incurring loss of detail. This proved much more effective and was selected as the best definition of ω_{C}^{Q} for optimisation.

Stop Ratio

The objective that most closely resembles the *ideal* outcome of bandwidth maximisation is stop ratio reduction. While in section 1.3.1 this was modelled under the assumption of *uniform arrivals* it is evident that for arterial progression optimisation it is necessary to consider the within-cycle dynamics of vehicle arrivals to account for the platooning effect of signals.

This is often done by propagating the cyclic flow profiles rigidly all the way to the next junction, and integrating them into the queue if they reach it when the signal is red, as illustrated in Figure 2.3. In this case the flow propagation performed by TRE is not only more realistic, but also accounts for the queue growth and spillback, which may cause a vehicle to stop well before the stop line.

Using the practical queue length approximation described in section 5.2.1 it is easy to obtain an estimate of the number of vehicles in the *hypercritical queue* as a proportional share of the vehicles on the link which are not *also* in the vertical queue. Dropping the time subscript for readability, the total number of vehicles in the *hypercritical* queue at each interval is easily expressed as

$$n_a^{Q} = \left(n_a - n_{\mathrm{N}_a^+}^{Q}\right) \cdot \ell_a^{Q} + n_{\mathrm{N}_a^+}^{Q} \tag{5.11}$$

which of course includes both the vehicles in the vertical queue and those within ℓ^{Q} from the link exit section.

If the definition of *stop* is extended, to include any event in which a vehicle joins the back of the queue, then the stops occurring during an interval $\tau = [t_0, t_1]$ are given by any increase

in the number of queued vehicles corrected to account for those that in the meanwhile leave the vertical queue, thusly:

$$\omega_\tau^S = \begin{cases} n_{a,t_1}^Q - n_{a,t_0}^Q + e_{a,\tau} \cdot \Delta t^\tau & \text{for } n_{a,t_1}^Q > 0 \text{ and } n_{a,t_1}^Q > n_{a,t_0}^Q \\ 0 & \text{for } n_{a,t_1}^Q = 0 \text{ or } n_{a,t_1}^Q < n_{a,t_0}^Q \end{cases} \quad . \tag{5.12}$$

where the different cases are used to exclude a *negative* stop count during the interval (not meaningful) and to avoid counting stops if there is definitely no queue and vehicles are flowing freely.

This can be integrated over the entire simulation window and for all arc sections to determine the *total number of stops*. Naturally, the number of vehicles that *may* stop depend on the number of vehicles that do access the corridor in the first place: it will suffice to normalise with respect to the *corridor total* expressed by (5.2) to obtain the *stop ratio* performance indicator:

$$\omega_{\mathrm{C}}^S = \frac{1}{\omega_{\mathrm{C}}^n} \sum_{\tau \in \mathrm{T}} \sum_{a \in \mathrm{C}} \omega_{a,\tau}^S \quad . \tag{5.13}$$

5.2.3 Dynamic Weighting

In order to magnify the short-term effects of signal timings on the traffic conditions, and isolate them from the *initial* situation in which they are enacted, it is possible to further shape the cost function by using time-dependent weights. This allows, for example, to give *more relevance* to the traffic conditions towards the end of the interval, favouring solutions that bring about a *negative* trend in the performance indicators (such as a progressive dissipation of the queues on a short arc) over solutions that lead to steady-state conditions, whence they may be indistinguishable if the results were simply time-averaged.

A generic scalar cost function ω of the decision variable x

$$\omega(x) = \int_{t \in \mathrm{T}} \omega(x, t) \tag{5.14}$$

may be shaped using a generic function of time $\Theta(t)$ as in

$$\omega^*(x) = \int_{t \in \mathrm{T}} \Theta(t) \cdot \omega(x, t) \tag{5.15}$$

which may take any form, e.g. it could be a step function to cut off a portion of the initial values, or a linear function of $t/|\mathrm{T}|$.

An analysis and comparison of the results obtained using each of the metrics just introduced is presented in the Results chapter.

Chapter 6

Results

This chapter collects results of the most relevant tests performed on the application.

First, an assessment of the overall effectiveness of the optimiser is presented. A brief analysis of computational performance is also given, alongside an estimation of the time requirements in correlation to the problem size. Finally, the choice of parameters for the Genetic Algorithm is discussed.

6.1 Corridor Performance Optimisation

Several tests were performed on randomly generated 8 junction corridors, with sections of variable length between the controlled junctions and varying demand flows accessing the corridor and leaving it at each intersection.

The tests presented in this section are concerned with the optimisation phase: models represent the corridor sub-networks, as would be produced for the rapid execution of the DNL algorithm. Traffic flow profiles entering the cordon arcs and hence the corridor, as well as turn rates, are assumed pre-calculated in the DTA phase as detailed in section 4.2.

This section considers several optimisation runs performed on the same representative 8-junction corridor, since the process is not deterministic and it is important to assess the consistency of results as well as their quality. The evolution of solution fitness as driven by each of the corridor performance functions is shown relative to the initial value, which invariably corresponds to one of the slack bandwidth *geometrical* solution variants used to prime the GA population as described in section 6.4.1.

The results presented demonstrate how the proposed method can significantly improve the corridor performance in relation to all of the proposed metrics, although each of them is more or less susceptible to optimisation under different traffic conditions.

6.1.1 Stable Subcritical Demand

To assess performance under *sub-critical* traffic conditions, several optimisation runs are performed while ensuring that no corridor link is subjected to demand flows that exceed its capacity after the reduction imposed by the effective green time, which is taken to be constant for the present application, i.e. $\chi_a = \frac{\phi_a}{\gamma_a} < 1 \; \forall a \in \mathrm{C}$.

Performance improvements over the simple geometrical solution are shown in Figure 6.1.

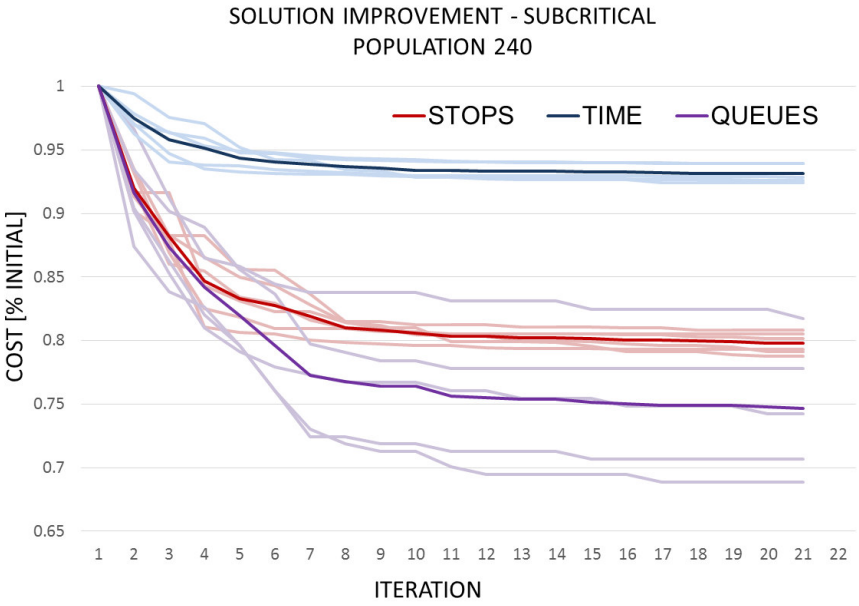

FIGURE 6.1 – Evolution of cost function values for subcritical flows and high side flows between 10% and 30% : solid lines in the foreground represent the average trend of the corresponding lighter sets in the background.

It is also interesting to note that if the traffic conditions are close to the theoretical premises of the simple bandwidth maximisation approach (i.e. if congestion does not significantly affect travel times and side flows are almost negligible with respect to the main corridor flow) the solutions found by the optimiser, even with a *completely random* initial population, never stray far from the geometrical solution, as shown in Figure 6.2, and are *identical* in case of no relevant side flows.

This is an important empirical confirmation of the system's stability and coherence with its theoretical background. As side flows increase, the optimal solutions diverge further and further from the simple spatio-temporal alignment of the green phases. This is only to be expected: vehicles that enter from cordon links reach the next junction ahead of those travelling along the corridor; as their number grows they start swaying the cost function significantly, so that the genetic algorithm will favour synchronisation patterns that minimise their discomfort.

The onset of the green phase at each junction is hence anticipated to meet the platoon coming from the side road, so that it may incur as little stops as possible and leave little or no queue behind for the main flows to run into. The effect is visible in Figure 6.2 where the green wave trajectories (which can never travel faster than free-flow speed) are shown to slow down to meet the onset of the next green phase.

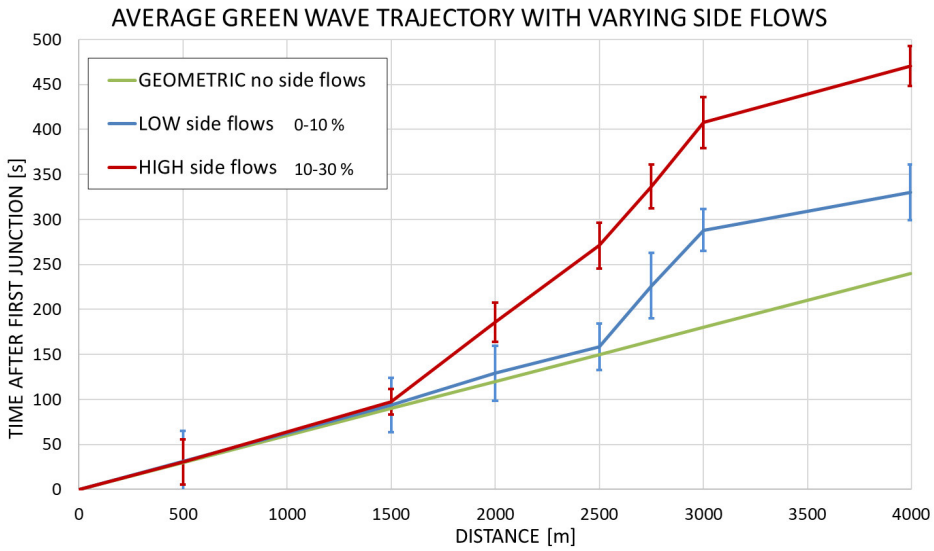

FIGURE 6.2 – Comparison of the average trajectory of a *green wave front* under sub-critical conditions and side flows of varying entity in relation to the total flow through each junction. The trajectory is the path a hypothetical vehicle would have to follow to drive through the start of every green phase along the corridor, obtained from the average results of ten runs for each scenario starting from random solutions (vertical bars show the standard deviation). With increasing side flows, offsets are adjusted to meet the side-flow platoons, and the main green wave must move more slowly, as visible from the steeper segments on the T-D diagram.

6.1.2 Stable Supercritical Demand

The foremost advantage of simulation-based heuristic optimisation is the possibility to search for better signal timings even if the traffic conditions are so distant from the ideal bandwidth maximisation scenario that making sensible *a priori* assumption about platoon arrival and queue discharge times becomes practically impossible.

The proposed optimiser was therefore extensively tested under *super-critical* traffic conditions, with relevant side flows entering the corridor during the main red phase and several arcs operating above their saturation capacity ($\chi \simeq 1.3 \pm 0.1$). Under these conditions, queues are *bound* to form on all such arcs (considerably affecting travel times) and a relevant fraction of the vehicles is not travelling through all junctions in sequence: simple geometrical solutions become completely inadequate.

The first scenario considered reproduces the optimisation of a time window during which high demand volumes rapidly enter an otherwise tranquil network, and aims to assess the capacity of the optimiser to keep corridor performance indicators in check. Figure 6.3 confirms that compared to the geometrical solutions that would serve the uncongested scenario, optimised plans can considerably reduce the number of stops and the growth of queues on short arcs.

It appears that queue management is most susceptible to early convergence into locally optimal solutions, as shown by the evident fork in the relevant values across the sample of optimisation runs, which for the test scenario presented in Figure 6.3 represents a 66% difference in gains over the starting solution between the worst and best cases; in every instance, however, gains are relevant and above 15% of the initial cost.

The average travel time may *seem* less sensitive to optimisation, due to the relatively small improvement in the relevant cost function. However, if the metric is inverted for readability

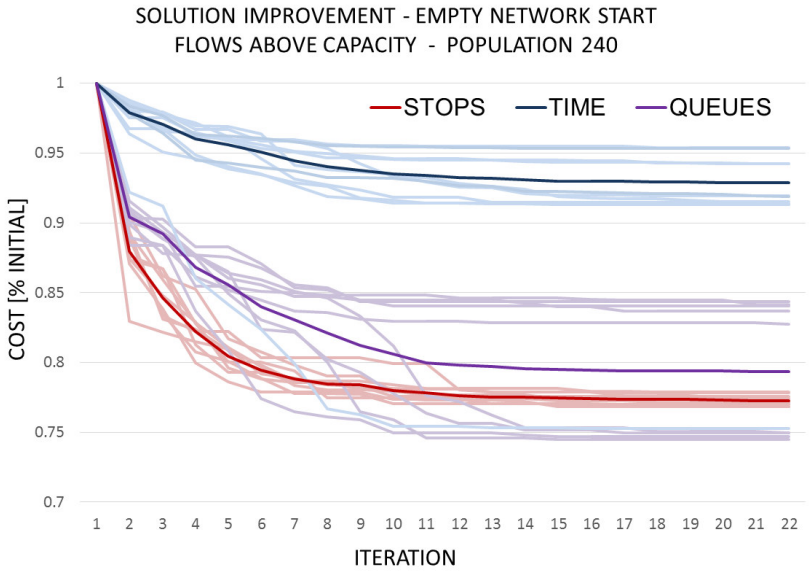

FIGURE 6.3 – Evolution of cost function values for the optimisation window containing the onset of high demand flows. Solid lines in the foreground represent the average trend of the corresponding lighter sets in the background.

as shown on the left-hand side of Figure 6.4, it is plain that the effect is due to the average speed being high to begin with: in fact, the optimiser is able to improve the performance with respect to the maximum bandwidth solution by a respectable 3 km/h, almost as well as in the case of sub-critical demand.

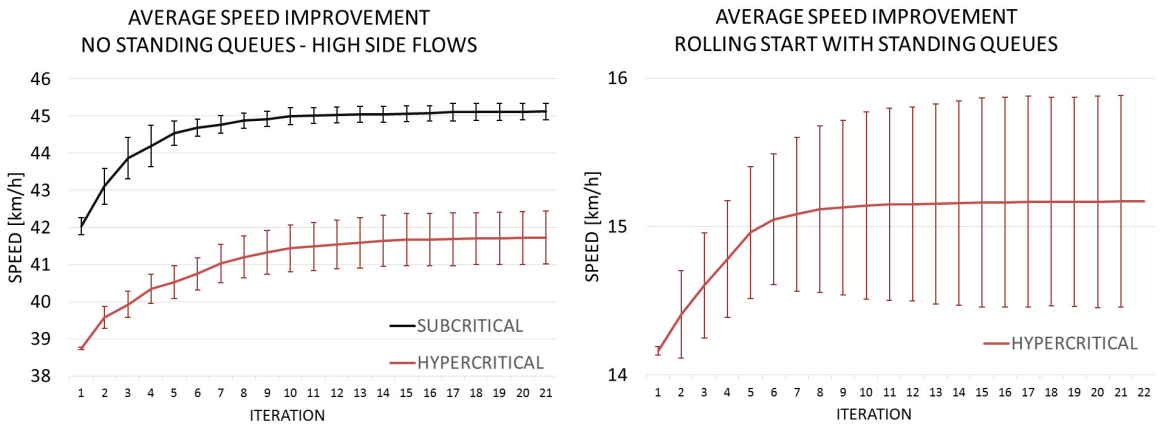

FIGURE 6.4 – Improvements in progression speed, averages and standard deviations corresponding to the scenarios presented in Figures 6.1, 6.3 (left) and 6.5 (right). The average user speed can be maintained effectively and consistently as long as the corridor does not collapse under sustained flows above capacity, after which gains become almost irrelevant and the optimiser behaviour is less reliable.

The second test scenario envisions an already heavily congested corridor with standing queues on most arcs, as the optimiser would inevitably face either during normal operation at peak hours or upon being switched back on after a down time.

It is plain to see from Figure 6.5 that while the standing queues mean that little can be done to avoid further stops, the *queue length* can be managed rather effectively and consistently, with

improvements between 6% and 8% shown across all tests. The same gains are obtained with respect to travel times, which being the inverse of the average speed across the subcritical and hypercritical portions of each arc are directly correlated with the queue length, as discussed in section 5.2.1.

Predictably, these large relative improvements are not as striking in terms of speed, which would be low in any case due to congestion as shown by the right-hand side of Figure 6.4. Nevertheless, they represent a positive reduction in the time lost by users in peak-hour traffic, e.g. for the test corresponding to Figure 6.5 users travelling the whole corridor would shave about two minutes off a seventeen minute trip in the best case scenario.

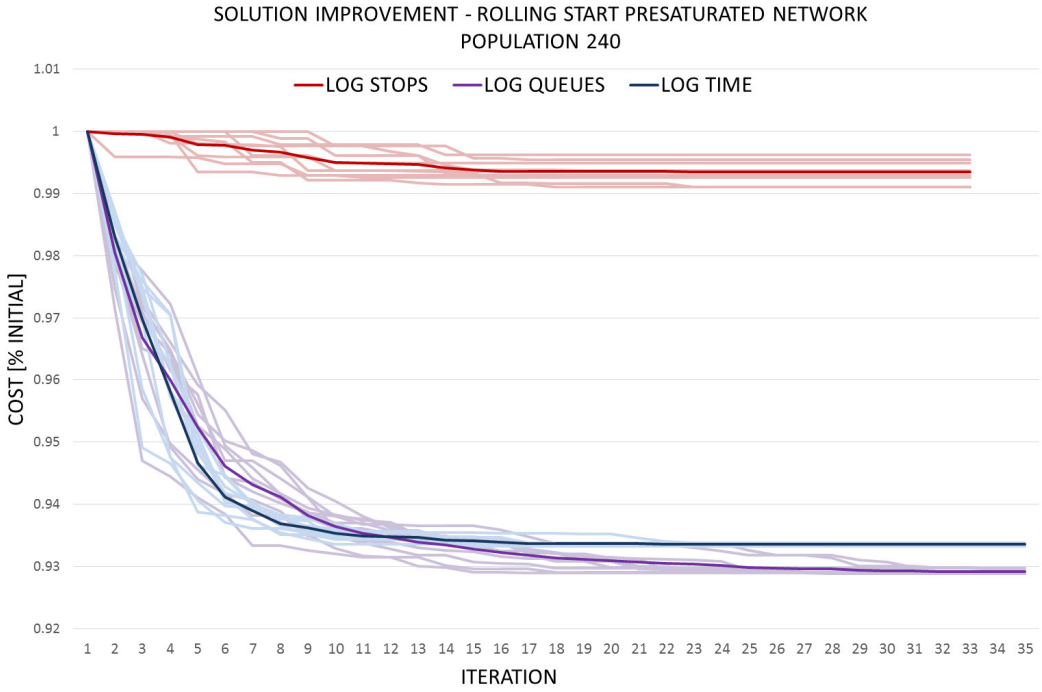

FIGURE 6.5 – Evolution of cost function values for an already congested corridor experiencing sustained flows above capacity. Solid lines in the foreground represent the average trend of the corresponding lighter sets in the background.

6.2 Computational Performance

All tests presented in this section were performed on a 3.4 GHz Intel®Core™i7-2600 CPU running up to 8 parallel threads on four processor cores. The computation times already proved adequate for *large* real world applications and entirely sufficient for smaller networks, as shown in Table 6.1.

Sizing of the optimisation look-ahead window must chiefly account for the lower bound imposed by the minimum time required by the optimiser to reach satisfying results, as discussed in Chapter 4.

With the optimal population size determined as in section 6.4.2 and the performance goal set to the best gains obtained across other tests (during which no time limitation was imposed), it was ascertained that corridor optimisation is indeed possible within the desired rolling-horizon time window of 10 minutes.

TABLE 6.1 – OPTIMISATION PERFORMANCE AND COMPUTATION TIMES

Network	Düsseldorf (small)	Turin / Piedmont Region (large)
Number of Links	1796	95794
Number of Zones	155	2009
Corridor Junctions	5	8
Single DTA iteration	~ 1 s	~ 40 s
Single DNL evaluation	~ 150 ms	~ 250 ms
Equilibrium DTA iterations	10	5
Genetic Algorithm Generations	8 with population 240 or 20 with population 120	8 with population 240 or 17 with population 120
Total Time	5 to 6 minutes	**~ 10 minutes**
Average Subcritical Stop Reduction	**- 20 ± 3 %**	**- 19 ± 2 %**
Average Congested Queue Reduction	**- 7 ± 1 %**	**- 6 ± 1 %**

6.3 Bi-directional Slack Bandwidth

Although to better understand the optimiser's behaviour the tests presented in this chapter mainly concern *one-way* offset optimisation, the methodology is easily extended to *two-way* corridor optimisation. This comes at no greater computational cost if the two directions run through the same junctions, which is the most common real-world use case.

In any case, the optimisation may account for a corridor *and* its return corridor, if one is defined on the network model (see section 1.4.1), by selecting solutions based on a linear combination of the cost functions calculated separately for the two directions. The sum can be weighted to favour optimisation in the main direction and protect its progression e.g. *in sight* of known traffic dynamics, although indicators such as the *stop ratio* and *time per kilometre* are intrinsically normalised on the number of vehicles as is evident from their definitions given in Chapter 5.

Figure 6.6 shows how the simple *slack band* method introduced in section 1.4.3 of this work can provide better starting solutions for two-way optimisation compared to the equally simple canonical two-way all-or-nothing bandwidth maximisation approach. Results show that favouring long and wide *partial* green waves over obtaining the narrowest *continuous* band positively affects a much larger number of vehicles: consequently, the fraction of stopped vehicles drops.

Furthermore, the outcomes are much more consistent across different scenarios, since as it is more clearly understood by referring to Figure 1.7, accounting for the band *fringes* arising from different green durations reduces the erratic wandering of the priming method amongst apparently equivalent solutions, which reveal their true efficacy when put to the test in simulation.

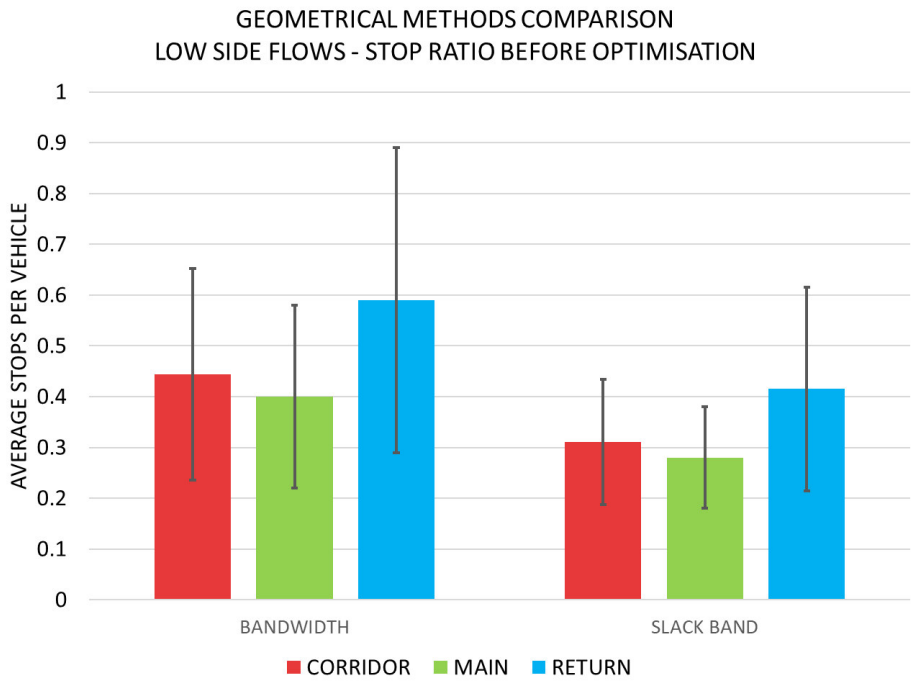

FIGURE 6.6 – Comparative fitness of the *initial solutions* for two-way optimisation provided by bandwidth and slack band maximisation respectively. Cost function values are obtained for each direction from the results of DNL, as would happen during optimisation, and linearly combined into the corridor performance indicator using a return direction weight of 30%. Values shown in the graph correspond to average values and standard deviations for a sample consisting of 100 randomly generated five-junction corridors, with varying phase durations and internal offsets.

6.4 Algorithm Parameters

This section presents results of the preliminary studies aimed at determining an optimal choice of parameters for the Genetic Algorithm.

6.4.1 Population Priming

Stochastic search methods are extremely sensitive to the choice of initial conditions. To speed up convergence and obtain better results, the present application initialises the genetic algorithm population using a geometrical solution (illustrated in section 1.4.3) that aims to align the green phases on the corridor so as to maximise the chance of driving through the longest possible distance without encountering a red light.

As discussed in section 3.4.2, it is necessary to ensure that the more *informed* starting point does not imply a much narrower-sighted search of the solution space, ultimately leading to early convergence and sub-optimal results. The problem is clearly *not* independent of the population size, and priming becomes at the same time more useful *and* more risky for smaller and smaller populations (which might be preferable from the computational point of view).

The choice of *partial* population priming with slack band solutions made for the present application is the consequence of the results shown in Figure 6.7. For a reasonably sized population (see section 6.4.2) the best results are obtained by priming *half* of the initial individuals with the geometrical solution, while the rest are generated randomly.

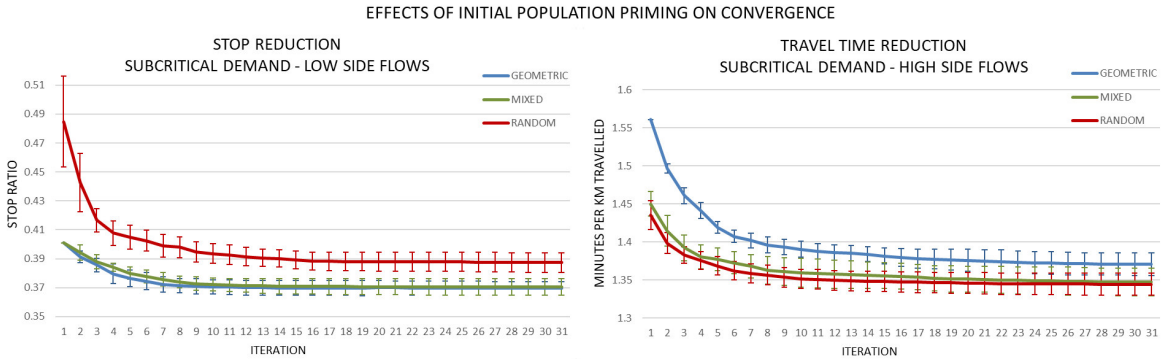

FIGURE 6.7 – Average cost function evolution arising from different population priming methods (sample size 20 optimisation runs for each scenario). Different problems favour geometrical solutions (if congestion and side flows are negligible, see left-hand side) or an initially random, unbiased population (as traffic conditions become less predictable, see right-hand side). A mix of the two performs practically as well as either in its best case, but across the whole range of test scenarios.

As expected, geometrical priming is particularly beneficial if the traffic conditions are close to the uncongested free flowing state on large sections of the corridor, while as delays and side flows increase an entirely random population performs better thanks to the more thorough and unbiased search of the solution space, but in a practical application the call could not be made a-priori without some complicated selection logic.

However, provided with a mixed initial population, the Genetic Algorithm can effectively select those *traits* of the geometrical solutions that apply to some corridor section, while including the diversity brought about by random solutions to yield near-optimal results across the whole range of tests performed with varying congestion and side flow levels.

6.4.2 Population Size

In general, a larger population is able to preserve more diversity and avoid early convergence, ultimately reaching better results, but as the number of individuals increases so does the time required to complete one full generation and hence improve solutions. The correlation between optimal population size and problem size (i.e. number of controlled junctions) was analysed by running randomly generated corridors with 3 to 10 junctions through the optimiser, and varying the population size between 8 and 256 individuals.

The results, classified by corridor length, were examined to determine the point at which the performance advantages of increasing the population are balanced out by the computational costs involved; an operation that obviously presumes some limitation on the maximum computation time which for the present study was set at or possibly *below* 8 minutes: the duration of the simulation time window minus the time required for a few DNL iterations (see section 6.2). Despite the variability of results due to the diversity of traffic conditions and topology encountered by the optimiser, a clear pattern emerges for corridors of any length confirming the expected diminishing returns, as exemplified in Figure 6.8 for the same class of problems presented in the previous sections. While minor scaling along the temporal axis due to the size of the sub-network influencing the DNL duration, the dominant factor in determining the largest useful population seems to be the cycle length over which the offsets are picked rather than the number of junctions (which is one order of magnitude smaller).

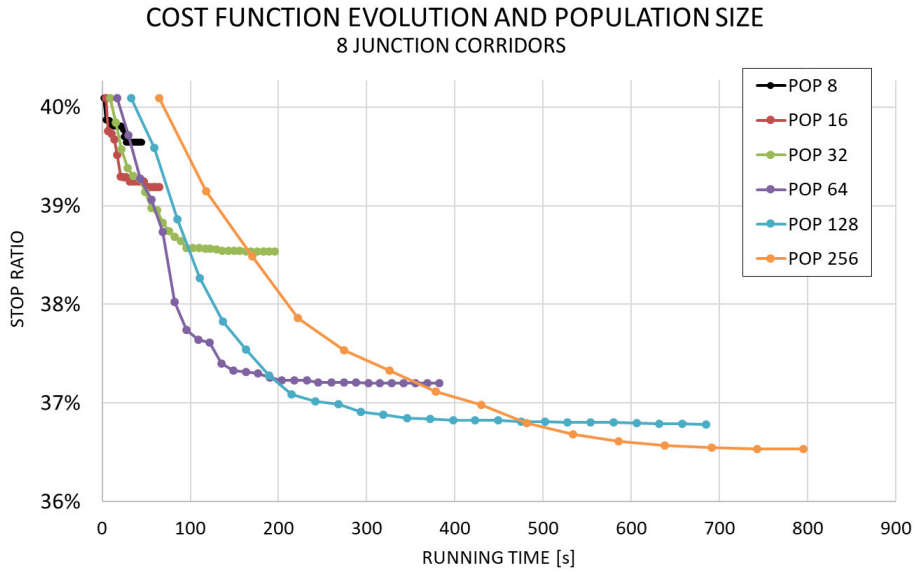

FIGURE 6.8 – Cost function evolution over time for different population sizes. Markers represent Genetic Algorithm generations.

The ideal population size for the chosen operational constraints is evidently around 128 individuals. The balance between solution fitness and time to convergence is ideal, allowing on average to reach performance gains which the larger population can *barely* match before the 8 minute mark, with about two minutes to spare on the imposed time limit.

Ultimately, as in most cases relating to stochastic search methods, a general rule for population sizing is hard to obtain and would not necessarily apply effectively to all instances of the same problem. Results shown in Figure 6.8 represent the application of a rule of thumb for practical application sizing.

Chapter 7

Conclusions

In the light of results presented in the previous chapter, it is possible to draw some conclusions regarding the endeavour undertaken in the context of this doctorate, and outline some of the major points of necessary improvement for the future of the proposed application.

Using an advanced Dynamic Traffic Assignment algorithm as heuristic for a Genetic Algorithm, the approach has proven to be effective in adjusting the offsets for a series of signal controlled junctions along a traffic corridor, to mitigate the undesirable consequences of traffic congestion and poor signal coordination.

Based on the supply and demand data provided by the Optima Real-Time Traffic Management environment, it was possible to significantly and reliably reduce the value of the selected cost functions, which describe the average progression speed, the number of stopped vehicles, and the queue lengths with the associated risk of gridlock.

Each performance function better serves different traffic conditions, but the optimisation process was entirely stable for a large number of randomly generated corridors. Priming of the Genetic Algorithm solution pool using the slack band method showed very positive results in terms of repeatability of the performance for different test scenarios. Nevertheless, fine tuning of the algorithm parameters should be performed for specific applications, as no blanket rule could be determined based on the tests alone.

The corridor performance improvements, as determined by macroscopic traffic flow simulation, obviously vary with the saturation levels of the network and deteriorate rapidly as congestion increases, but remain relevant even in the worst case scenarios.

The simulation-based optimiser could regularly achieve a reduction of no less than 5% in the average queue length under heavy congestion, ranging all the way up to a 20% reduction in the number of stops when more favourable conditions applied, when compared to simpler geometrical considerations known in literature; the advantages over fixed signal plans are bound to be even greater.

Finally, Real-Time Simulation-Based Optimisation appears to be computationally viable, as the execution times of the proposed algorithm would allow the optimiser to operate in rolling-horizon with a look-ahead window of 10 minutes on large real-world networks.

The simple application presented in this thesis therefore serves mainly as a proof of concept that signal optimisation may be fruitfully integrated into a real-time traffic management environment, allowing to make better objective-driven decisions based on more realistic models and more reliable data, and providing a stable and cost effective alternative to many of the currently commercialised solutions.

7.1 Future Work

The aim of this work was to explore the possibility to integrate active signal control into the real-time traffic management environment and take advantage of its simulation and modelling capabilities to drive the optimisation. A minimal application, i.e. an arterial coordination module, was proposed to gauge the potential benefits of such integration.

This first step was successful, and results cannot but encourage further work along the following main directions:

Consolidation of the performance assessment: the tests presented, however thorough, were performed using the same tools that made the optimisation possible. While this does not in any way diminish the validity of their results, it would be absolutely mandatory, before proceeding to further developments, to confirm the effectiveness of the solutions provided by the optimiser using an aptly calibrated external simulation tool. This process is already under way as this PhD draws to its conclusion.

Comparison with existing alternatives: once equal grounds for comparison are established, this application should be pitched against its direct competition to assess if it can provide significantly better signal control and by what margin. This may take a while, since reliable performance data of commercial systems is hard to obtain, but it would be possible to start by using the existing in-house signal optimiser known as PTV-BALANCE as a benchmark before committing to the new approach. It is worth noting, however, that the potential to enact signal control based on harmonised data, interposing a level of abstraction between the data gathering and the decision making, is known as a fact to be a unique feature of this integration at the time of writing.

Extension and Scaling: while undoubtedly useful, the proposed application is rather limited in its scope, and only aimed to serve as a proof of concept. With an appropriate test environment in place and adequate computing power, the application should be extended to more complex problems, such as area optimisation, and scaled using task distribution to handle several optimisation tasks at once on large real-world networks.

Bibliography

Konstantinos Aboudolas, Markos Papageorgiou, and E Kosmatopoulos. Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. *Transportation Research Part C: Emerging Technologies*, 17(2):163–174, 2009.

Alessandro Attanasi, Edmondo Silvestri, Pietro Meschini, and Guido Gentile. Real world applications using parallel computing techniques in dynamic traffic assignment and shortest path search. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pages 316–321. IEEE, 2015.

Jan K Brueckner. Urban sprawl: diagnosis and remedies. *International regional science review*, 23(2):160–171, 2000.

Chiara Colombaroni, Gaetano Fusco, Andrea Gemma, M Demiralp, N Baykara, and N Mastorakis. Optimization of traffic signals on urban arteries through a platoon-based simulation model. In *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*, number 11. World Scientific and Engineering Academy and Society, 2009.

Carlos F Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287, 1994.

Carlos F Daganzo. The cell transmission model, part II : network traffic. *Transportation Research Part B: Methodological*, 29(2):79–93, 1995.

Giovanni De Nunzio, Gabriel Gomes, Carlos Canudas de Wit, Roberto Horowitz, and Philippe Moulin. Arterial bandwidth maximization via signal offsets and variable speed limits control. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pages 5142–5148. IEEE, 2015.

Christina Diakaki, Markos Papageorgiou, and Kostas Aboudolas. A multivariable regulator approach to traffic-responsive network-wide signal control. *Control Engineering Practice*, 10(2):183–195, 2002.

Christina Diakaki, Vaya Dinopoulou, Kostas Aboudolas, Markos Papageorgiou, Elia Ben-Shabat, Eran Seider, and Amit Leibov. Extensions and new applications of the traffic-responsive urban control strategy: Coordinated signal control for urban networks. *Transportation Research Record: Journal of the Transportation Research Board*, (1856):202–211, 2003.

Thomas J Dickson. A note on traffic assignment and signal timings in a signal-controlled road network. *Transportation Research Part B: Methodological*, 15(4):267–271, 1981.

Russell C Eberhart and Yuhui Shi. Comparison between genetic algorithms and particle swarm optimization. In *International conference on evolutionary programming*, pages 611–616. Springer, 1998.

Reid Ewing, Tom Schmid, Richard Killingsworth, Amy Zlot, and Stephen Raudenbush. Relationship between urban sprawl and physical activity, obesity, and morbidity. In *Urban Ecology*, pages 567–582. Springer, 2008.

Darren Flusche. Bicycling means business: The economic benefits of bicycle infrastructure. 2012.

Guido Gentile and Daniele Tiddi. Synchronization of traffic signals through a heuristic-modified genetic algorithm with gltm. In *Proceedings of XIII Meeting of the Euro Working Group on Transportation, Padova University Press, Padua, Italy*, 2009.

Guido Gentile et al. The general link transmission model for dynamic network loading and a comparison with the due algorithm. *New developments in transport planning: advances in Dynamic Traffic Assignment*, 178:153, 2010.

Mehdi Keyvan-Ekbatani, Anastasios Kouvelas, Ioannis Papamichail, and Markos Papageorgiou. Congestion control in urban networks via feedback gating. *Procedia-Social and Behavioral Sciences*, 48:1599–1610, 2012.

Mehdi Keyvan-Ekbatani, Markos Papageorgiou, and Ioannis Papamichail. Urban congestion gating control based on reduced operational network fundamental diagrams. *Transportation Research Part C: Emerging Technologies*, 33:74–87, 2013.

RM Kimber and Erica M Hollis. Traffic queues and delays at road junctions. Technical Report LR909 Monograph, Transport and Road Research Laboratory, 1979.

Peter Koonce, L Rodegerdts, K Lee, S Quayle, S Beaird, C Braud, J Bonneson, P Tarnoff, and T Urbanik. Traffic signal timing manual, no. Technical report, FHWA-HOP-08-024, 2008.

Wei Li and Andrew P Tarko. Safety consideration in signal coordination and road design on urban streets. In *4th International Symposium on Highway Geometric DesignPolytechnic University of ValenciaTransportation Research Board*, 2010.

Michael James Lighthill and Gerald Beresford Whitham. On kinematic waves II. a theory of traffic flow on long crowded roads. *Proc. R. Soc. Lond. A*, 229(1178):317–345, 1955.

STEG Linda. Can public transport compete with the private car? *Iatss Research*, 27(2): 27–35, 2003.

John DC Little, Mark D Kelson, and Nathan H Gartner. Maxband: A versatile program for setting signals on arteries and triangular networks. 1981.

Russ Lopez. Urban sprawl and risk for being overweight or obese. *American journal of public health*, 94(9):1574–1579, 2004.

Highway Capacity Manual. Special report 209. *Transportation Research Board, Washington, DC*, 1:985, 1985.

Gordon F Newell. A simplified theory of kinematic waves in highway traffic, part I: General theory. *Transportation Research Part B: Methodological*, 27(4):281–287, 1993.

Natale Papola and Gaetano Fusco. A new analytical model for traffic signal synchronization. In *Traffic and Transportation Studies (2000)*, pages 499–506. 2000.

Byungkyu Park, Carroll Messer, and Thomas Urbanik. Traffic signal optimization program for oversaturated conditions: genetic algorithm approach. *Transportation Research Record: Journal of the Transportation Research Board*, (1683):133–142, 1999.

Paul I Richards. Shock waves on the highway. *Operations research*, 4(1):42–51, 1956.

Dennis I Robertson. Transyt: a traffic network study tool. 1969.

Dennis I Robertson. Research on the transyt and scoot methods of signal coordination. *ITE journal*, 56(1):36–40, 1986.

Daniele Tiddi. Models for dynamic network loading and algorithms for traffic signal synchronization. 2012.

Huel-Sheng Tsay and Liang-Tay Lin. *NEW ALGORITHM FOR SOLVING THE MAXIMUM PROGRESSION BANDWIDTH (WITH DISCUSSION AND CLOSURE)*. Number 1194. 1988.

J G Wardrop. Road paper. some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 1(3):325–362, 1952.

F V Webster. Traffic signal settings. Technical report, Transport and Road Research Laboratory, 1958.

Clifford Winston. Government failure in urban transportation. *Fiscal Studies*, 21(4):403–425, 2000.

Isaak Yperman. *The link transmission model for dynamic network loading*. PhD thesis, Katholieke Universiteit Leuven, 2007.

Isaak Yperman, Steven Logghe, and Ben Immers. The link transmission model: an efficient implementation of the kinematic wave theory in traffic networks. In *Proceedings of the 10th EWGT Meeting*, pages 122–127. Poznan Poland, 2005.

Appendix A

Parallel Dynamic Traffic Assignment

This section summarises the results obtained during the preparation of the article on *Real World Applications using Parallel Computing for Dynamic Traffic Assignment and Shortest Path Search*, presented at ITSC 2015 in Las Palmas de Gran Canaria [Attanasi et al., 2015].

They are profoundly relevant to the application proposed in this thesis, as performance issues are what has kept simulation-based real-time optimisation (and in fact even offline heuristic approaches coupled with traffic simulation) essentially *unfeasible* until the present day. The possibility to scale up the presented approach thanks to the in-house developments in parallel computing is a fundamental asset for the future development of the presented approach.

The metrics of the benchmark networks usd to study performance gains as a function of network sheer size and complexity are shown in Table A.1. These are real-world commercial use networks, ranging from a small city to a huge whole-region model.

Table A.1 – Test Networks

Network Metrics			
Region	N. of nodes	N. of links	N. of zones
5T-Piedimont	34606	95794	2009
Moscow	18749	46334	644
Catania	2052	6006	89
Düsseldorf South	656	1696	155

The tests considered the execution times of the two main (and most time-consuming) phases of the Dynamic Traffic Assignment algorithm used in the present application, namely the demand routing using *Dynamic Path Search* and the *Dynamic Network Loading*, i.e. the traffic propagation using the General Link Transmission Model. It should be noted that the Dynamic Path Search complexity increases with the number of arcs but the overall cost of the path search phase is mainly affected by the number of O-D *zones*, while the Dynamic Network Loading computation time scales linearly with the number of nodes and links.

Route Choice Model

Results for the Route Choice parallelisation are shown in Figure A.1, whence it is evident that performance gains are linear for 2 and 4 parallel threads, and that for larger networks the trend continues *almost* linearly up to 16 threads, leading to computation time reductions between 80% and 90%.

It is worth noting that these are *much* greater that what could be gained by parallelisation of the A* path search algorithm, even though the network used to obtain the results shown in Figure A.2 was even larger than those in Table A.1, totalling 1.1 Million nodes and 2.6 Million arcs over the *entire* Austrian territory.

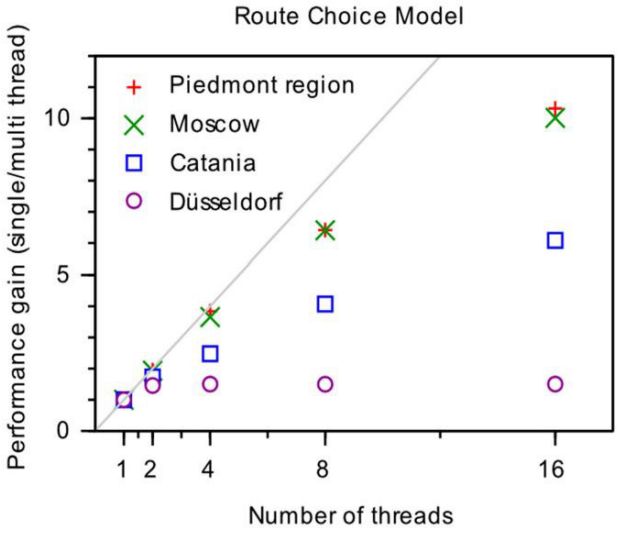

FIGURE A.1 – Performance gains as reductions in Route Choice Model computation time against the number of parallel threads running serial A* searches. Different networks have very different sizes indicated in A.1.

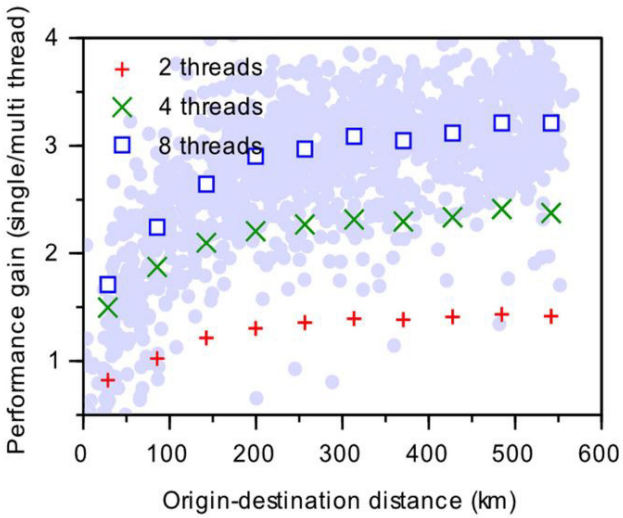

FIGURE A.2 – Performance gains resulting from parallelisation of the A* path search algorithm, shown as the ratio of single/multi-thread execution times for the same batch of several thousand shortest path requests. Light data points in the background show the actual scatter of the execution times for the 8-thread A* search.

Network Performance Model

The results for the Dynamic Network Loading shown in Figure A.3 are equally encouraging; they show how parallelisation overhead and inter-thread conflicts for this phase are slightly heavier, resulting soon in sub-linear gains for each additional thread.

This *saturation* effect is more and more relevant as the networks grow smaller, which would justify the choice of parallel single-thread TRE instances to share solution evaluation requests during optimisation over parallel-thread DNL.

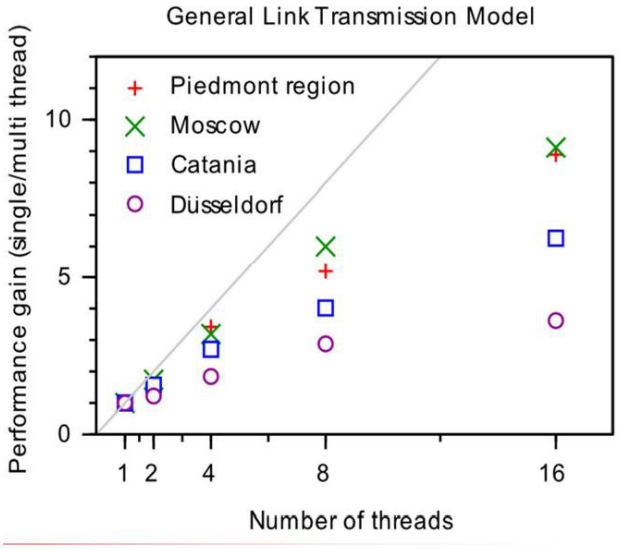

FIGURE A.3 – Performance gains as reductions in Dynamic Network Loading execution time against the number of parallel threads extracting network nodes at each time step of the algorithm to compute flow cumulatives and perform kinematic wave propagation. Different networks have very different sizes indicated in A.1.