Access control on networks with unique origin–destination paths

David J. Lovell a,*, Carlos F. Daganzo b

a Department of Civil Engineering, University of Maryland, 1179 Engineering Classroom Building, College Park, MD 20742, USA
b Institute of Transportation Studies, University of California, Berkeley, CA 94720, USA

Received 29 May 1998; received in revised form 19 April 1999; accepted 23 April 1999

Abstract

This paper presents improved time-dependent control strategies for small freeway networks with bottlenecks and unique origin–destination paths. It is assumed that there are no spill-overs from any of the freeway exits so that freeway queues and delays can be completely avoided by regulating access to the system so as to maintain bottleneck flows strictly below capacity. It is also assumed that the time-dependent origin–destination table and the time-dependent bottleneck capacities are known, although not always a priori. The proposed control strategies attempt to minimize the total delay (including both system delay and access delay) while avoiding queues inside the system. The problem is formulated as a constrained calculus of variations exercise that can be cast in the conventional form of optimal control theory and can also be discretized as a mathematical program. Although the first-in-first-out (FIFO) requirement for the access queues introduces undesirable non-linearities, exact solutions for four important special cases can be obtained easily. More specifically, for networks with (1) a single origin or (2) a single bottleneck, a myopic strategy which requires the solution of a sequence of simple linear programs is optimal. For networks with (3) a single destination the non-linearities disappear and the problem becomes a large-scale linear program. This is also true for general networks if (4) the fractional distribution of flow across destinations for every origin is independent of time. A greedy heuristic algorithm is proposed for the general case. It has been programmed for a personal computer running Windows. The algorithm is non-anticipative in that it regulates access at the current time without using future information. As a result, it is computationally efficient and can be bolstered with dynamically-updated information. Globally optimal for cases (1) and (2), the heuristic has been developed with slow-varying O–D tables in mind. Significant improvements will likely require anticipatory information. An illustrative example is given. © 2000 Elsevier Science Ltd. All rights reserved.

*Corresponding author. Tel.: +301 405 7995; fax: +301 405 2585.
E-mail address: lovell@eng.umd.edu (D.J. Lovell).
1. Introduction

The problem of optimal control of metered networks has been studied extensively (see Lovell, 1997 for a review of this literature), but the lack of reliable time-dependent origin–destination (O–D) information may have discouraged efforts to develop “intelligent”, time-dependent strategies which would make full use of this information.

Hopefully, recent innovations in hardware and methodology, such as electronic toll collection and automatic vehicle identification (AVI), time-dependent O–D prediction, forecasting, and the processing of data from probe vehicles and detectors, should lead to improvements in the data. Therefore, it seems reasonable to re-visit the time-dependent control problem, seeking methods which take advantage of these developments.

As a small step in this direction, Section 2 of this paper describes the formulation of a continuous-time optimization problem for the control of simple freeway networks on which unique routes connect O–D pairs. The goal of the problem is to choose the rates at which vehicles are allowed to access the system from some of the entry points, so as to minimize the total system delay. It is assumed that a number of bottlenecks exist within the network, with known and possibly time-dependent capacities, and that no internal queues are allowed to develop. ¹

Although the problem is highly non-linear, it is shown in Section 3 that in some important special cases it can be decomposed and solved to optimality. For the remaining cases, Section 4 presents a non-anticipative heuristic technique which is fast, efficient, robust and parsimonious; i.e., which uses only a small amount of (likely to be) available information.

2. Formulation of continuous-time access control problem

It is assumed that reasonable estimates of time-dependent O–D patterns are available; i.e., that we know functions \( A_{ij}(t) \), which denote the cumulative number of vehicles wishing to depart origin \( i \) at time \( t \), whose ultimate destination is \( j \). As in a fluid approximation, we assume that these function are continuous and increasing. Thus, the inverses \( A_{ij}^{-1}(n) \) exist and give the desired departure time from origin \( i \) of the \( n \)th vehicle with destination \( j \). The aggregate cumulative demand curve for origin \( i \) is defined as:

\[
A_{i*}(t) = \sum_j A_{ij}(t) \quad \forall i, t. 
\]

Its inverse gives the desired departure time from origin \( i \) of the \( n \)th vehicle with any destination.

The function describing the cumulative number of vehicles of a given O–D pair \( \{i,j\} \) to have departed from the origin by time \( t \) is denoted \( D_{ij}(t) \). Like the demand curves, a departure curve is invertible, yielding the time of the \( n \)th departure for a given O–D pair. Departure curves can also be aggregated by origin:

¹ In the steady-state case, the output flow can be maximized without any internal queues; for example, with the method of Wattleworth (1967). Although reasonable, this is not necessarily true in the time-dependent case. The possible benefits of allowing internal queues to develop should be studied in the future.
The inverse of the aggregate curve gives the actual departure time of the \( n \)th vehicle with \textit{any} destination.

We will assume that the \( \{D_{ij}(t)\} \) can be affected by control but that the \( \{A_{ij}(t)\} \) are not. Thus, in our formulation, the \( \{A_{ij}\} \) are data and the \( \{D_{ij}\} \) are control or decision curves. These decision curves cannot be chosen arbitrarily; they must satisfy the feasibility conditions outlined in Sections 2.1–2.3.

2.1. Queueing conditions at access points: FIFO

Let the control period begin at time \( t = 0 \) and end at time \( t = t_E \). These times are fixed, but should be chosen so as to insure that all the congestion builds up and dissipates during the control period under any reasonable control strategy. This should not be difficult, since real traffic problems usually exhibit markedly uncongested periods, such as late at night. The entry queues are assumed to be empty prior to \( t = 0 \); thus, the \( D_{ij}(t) \) are assumed to satisfy:

\[
D_{ij}(t) = A_{ij}(t) \quad \forall i, j, \quad \forall t \leq 0.
\]  \hspace{1cm} (3)

Likewise, we will look for a control strategy that will have eliminated all queues by \( t = t_E \); i.e., where the control curves also satisfy:

\[
D_{ij}(t) = A_{ij}(t) \quad \forall i, j, \quad \forall t \geq t_E.
\]  \hspace{1cm} (4)

Furthermore, since vehicles must certainly depart after they arrive, \( D_{ij}(t) \) must also satisfy:

\[
D_{ij}(t) \leq A_{ij}(t) \quad \forall i, j, t.
\]  \hspace{1cm} (5)

Of course, flows must be non-negative:

\[
\dot{D}_{ij}(t) \geq 0 \quad \forall i, j, t,
\]  \hspace{1cm} (6)

where, as is customary in some fields of applied mathematics such as control theory, time-derivatives are denoted by overdots.

In addition, we assume that the access point discharge cannot exceed the access capacity of entry point \( i \), \( R_{i}^{\max} \):

\[
\dot{D}_{i}(t) \leq R_{i}^{\max} \quad \forall i, t.
\]  \hspace{1cm} (7)

Note that (3), (4) and (7) imply that \( \dot{A}_{i}(t) \leq R_{i}^{\max} \forall i, t \notin [0, t_E] \). Otherwise, queues would be present either before or after the control period.

It is assumed that one can exercise direct control over \( \{D_{i}(t)\} \) (i.e., over \( \{D_{i}(t)\} \)) by restricting the rate at which vehicles are released from the queue, but not over the \( \{\dot{D}_{i}(t)\} \) individually. These individual departure rates are related to the aggregate departure flows by the queue discipline. Under first-in-first-out (FIFO), any vehicles that depart together in the same short interval of time, regardless of their destination, must have arrived together some time earlier; i.e., with smooth O–D arrival curves, any vehicles departing at time \( t \) from \( i \) will share a common delay, \( d_{i}(t) \); see Fig. 1.

Fig. 1 shows how this property can be used to obtain the destination-specific curves (and rates) from a given aggregate departure curve. For example, to identify the ordinates of curves \( D_{i1} \) and \( D_{i2} \) at time \( t \), we drop vertical lines from (known) points “P” and “Q” on the aggregate curves to
identify points “R” and “S”. Their ordinates, \( n_1 \) and \( n_2 \), are the values of \( D_{i1}(t) \) and \( D_{i2}(t) \) being sought. The mathematical expression of this procedure is:

\[
D_{ij}(t) = A_{ij}(t - d_i(t)) \quad \forall i, j, t. \tag{8}
\]

Addition across \( j \) reveals that (8) also holds in the aggregate case (for \( j = \bullet \)). The delay functions thus defined must certainly be non-negative:

\[
d_i(t) \geq 0 \quad \forall i, t. \tag{9}
\]

Constraint (8) is a source of difficulties because it is non-linear. Note for example that even for a constant metering rate, the disaggregate departure curves by O–D pair could vary with time. The constraint also implies that the system remembers the past for a time equal to the longest delay, which is not known a priori.

2.2. Conditions for no queues in the network

It is assumed that bottleneck \( k \) has a time-dependent capacity \( c_k(t) \) and that queues do not form upstream of bottlenecks if the (desired) flow past each bottleneck does not exceed \( c_k(t) \). Only control policies that avoid queues are considered here. In the absence of queues, free-flow travel times (assumed to be known) prevail on the network. The travel time from entry \( i \) to bottleneck \( k \) is denoted \( \tau_{ik} \).

---

2 If it were possible to meter flows by destination (as occurs for airplanes held on the ground when a main hub airport is congested), the solution to the problem would be easier and perhaps also more effective (the system is more “controllable”).

3 Travel times for any {meter, bottleneck} pair can vary across individual vehicles and can also differ slightly from the expected values for other reasons, but experience suggests that these differences are only on the order of a few seconds per mile and can be neglected.
If a control strategy is in place which sends periodic updated control instructions to all meters simultaneously, the metering rates at the entries will change simultaneously, but the manifestations of these changes will appear at each bottleneck at different times. These trip time lags must be reflected in the bottleneck capacity constraints.

Fig. 2 shows two departure curves, \( D_{ij}(t) \) and \( D_{i'j'}(t) \), for two O–D pairs \((1j, 2j')\) with different origins and different destinations. If these are the only two O–D pairs using bottleneck \( k \), then the desired cumulative bottleneck count is the sum of the two time-shifted O–D departure curves: \( B_k(t) = D_{ij}(t - \tau_{1k}) + D_{i'j'}(t - \tau_{2k}) \), as shown. In general, the desired cumulative bottleneck count is

\[
B_k(t) = \sum_{i,j} c_{ijk} D_{ij}(t - \tau_{ik}) \quad \forall k, t,
\]

where \( c_{ijk} \) is an indicator constant which is 1 if O–D pair \((i,j)\) requires the use of bottleneck \( k \), and 0 otherwise. Thus, the queue avoidance restriction is:

\[
\dot{B}_k(t) \equiv \sum_{i,j} c_{ijk} \dot{D}_{ij}(t - \tau_{ik}) \leq c_k(t) \quad \forall k, t. \tag{10}
\]

2.3. Optional conditions

It may be necessary to require a minimum metering rate \( R_i^{\min}(t) \) at certain entry points, perhaps to help prevent external queues from growing too long, or to provide assurance to drivers that the signal is functioning correctly. \(^4\) In these cases, (6) should be strengthened with the implication \( d_i(t) > 0 \Rightarrow \dot{D}_i(t) \geq R_i^{\min}(t) \); i.e., with:

\[
d_i(t) \left[ \dot{D}_i(t) - R_i^{\min}(t) \right] \geq 0, \quad \forall i, t. \tag{11a}
\]

\(^4\) This is only one of the many possible ways in which external queues can be minimized. There is no obvious solution to this problem, as the question of where to store excess demand to a congested system can be very political. Our proposal simply illustrates that external queue considerations can be easily captured by our formulation.
Similarly, the cyclical nature of the signal may suggest a maximum rate \( R_{i}^{\text{max}}(t) \leq R_{i}^{\text{max}} \), in which case (7) should be strengthened with:

\[
\tilde{D}_{i}(t) \leq R_{i}^{\text{max}}(t) \quad \forall i, t. \tag{11b}
\]

For entry points which are not controlled by meters, the departure curve from the “meter” (this could be any conveniently chosen location along the entry link) should match the departure curve that would be obtained from the standard queueing model with a bottleneck of rate \( R_{i}^{\text{max}} \); i.e., be the highest possible curve for the given \( A_{i}(t) \) consistent with (5)–(9). This calculation can be done off-line, with the result being \( \tilde{A}_{i}(t) \). Then, we require:

\[
D_{ij}(t) = \tilde{A}_{ij}(t) \quad \forall i \in I_{\text{nm}}, \forall t, \quad j = \bullet, 1, 2, \ldots, \tag{12}
\]

where \( I_{\text{nm}} \) is the set of unmetered origins. Note that if \( I_{\text{nm}} \) is not empty and the flows from these origins are high, a feasible solution to our problem may not exist.

For the set of meters which are controlled by meters (denoted \( I_{m} \)), it may also be desirable to introduce a constraint to prevent the metering rates from varying too drastically over time:

\[
\left| \tilde{D}_{i}(t) \right| \leq U_{i} \quad \forall i \in I_{m}, \forall t. \tag{13}
\]

2.4. Objective function

A common objective in system control is to minimize the total time in the system for all users. This may not be reasonable in a system without alternative routes if it leads to very different treatment of various (otherwise similar) entry points, but constraints such as (11a) can be used to moderate these differences. Because travel times over the network are assumed to be constant, this objective is equivalent to minimizing the queueing delay at the access points:

\[
\min \sum_{i,j} \int_{t_{0}}^{t_{e}} [A_{ij}(t) - D_{ij}(t)] \, dt.
\]

Because the \( \{A_{ij}(t)\} \) are constant, this objective is equivalent to:

\[
\max \sum_{i,j} \int_{t_{0}}^{t_{e}} D_{ij}(t) \, dt. \tag{14}
\]

Our problem, thus, is to maximize (14) subject to constraints (2)–(13). For the remainder of this paper, this set of equations will be called the Continuous Optimization Problem (COP).

The COP can be cast in the conventional form of “optimal” control theory. Although many formulations are possible, the simplest way to see this given the equations we have already presented is to note that we could take the derivative of (8) as the state equation, the time-derivatives of the delays \( \{\tilde{d}_{i}\} \) as the controls and a state vector that would include \( \{D_{ij}\} \) (time-lagged so as to capture (10)). Unfortunately, the problem seems not to be of the type where the variational methods of control theory guarantee a global optimum. For the reader interested in other control-theoretic approaches to similar problems with different assumptions, see for example Papa-georgiou (1995), Stephanedes and Kwon (1993), Yang et al. (1994) and Zhang et al. (1996).
In view of the difficulty in solving the problem exactly, the rest of this paper is divided into two parts. First, there are a few special cases where the optimum is unique. These cases and efficient solution methods are discussed in the following section. For more general cases, a heuristic solution algorithm is developed in Section 4.

3. Exact solutions for special conditions

Fortunately, there are some special network structures and demand characteristics for which the problem is more tractable. For networks with (i) a single origin, (ii) a single bottleneck, (iii) a single destination and for the case where (iv) the O–D fractions are time-independent, the global optimum can be found efficiently. Admittedly, these may only represent some of the networks found in practice. These special instances, however, are sometimes the ripest opportunities to use traffic control in a beneficial and relatively unambiguous way.

3.1. Networks with a single origin

If a network has only one metered origin, then that meter does not have to compete with any other sources for service at the bottleneck(s). It should be intuitive, then, that a greedy strategy that always admits the highest flow possible without violating bottleneck capacity constraints would be optimal. As proof of this, notice that of the constraints in the COP, only (10) includes variables \( D_{ij}(t), d_i(t) \) and \( \bar{D}_{ij}(t) \) from different time periods for the same \( t \). The objective function is a sum of \( D_{ij}(t) \)'s across time slices, without cross-time interactions. When there is only one origin, however, the subindex \( i \) can be dropped from the formulation and \( t \) can be changed to \( t + \tau_k \) in (10) with the result:

\[
\sum_j \gamma_{jk} \dot{D}_{ij}(t') \leq c^*_k(t') \quad \forall k, t',
\]

where \( c^*_k \) is a time-shifted capacity function defined as: \( c^*_k(t') \equiv c_k(t' + \tau_k) \). Thus, the cross-period dependencies have been eliminated in all of the equations, establishing that the strategy which maximizes \( \sum_j D_{ij}(t) \equiv D_t(t) \) for all \( t \), with each \( t \) considered separately, is globally optimal.

At any time \( t \in [0, \tau_2] \), Eqs. (2)–(5), (8), (9) and (12) are all either irrelevant, automatically satisfied, or yield quantities derived from data known at the time. Eq. (10) is replaced by (15), which is linear in the destination-specific departure rates, as are (6), (7) and (11b). Because \( d_i(t) \) is known data at time \( t \), (11a) is irrelevant if \( d_i(t) = 0 \) and linear in the departure rates otherwise. Eq. (13) is also linear in \( \dot{D}_{ij}(t) \). Therefore, the individual subproblem solved for each \( t \) is a simple linear program, where the objective is to maximize either \( \dot{D}(t) \) or \( D(t) \).

In a discrete-time implementation where \( t_n \) denotes the current time and \( t_{n-1} \) the time at which the metering rate was last changed, (13) would be:

\[
-\ U \leq \frac{\dot{D}(t_n) - \dot{D}(t_{n-1})}{t_n - t_{n-1}} \leq \ U,
\]

and the COP equations would have to be complemented with a differentiation relation:

\[
D(t_n) = D(t_{n-1}) + (t_n - t_{n-1}) \dot{D}(t_{n-1}) \quad \forall n \geq 1.
\]
Note that none of the data in the LP subproblem comes from the future. Thus, in the case of a single origin, the optimal strategy is both simple and practical; it is greedy and not anticipative.

One particular advantage of strategies such as this that need not rely on future predictions is that data can be provided “just in time”, perhaps via some dynamic updating mechanism. This should serve to improve the reliability of the data and hence the control strategy as well.

3.2. Networks with a single bottleneck

In this case, a simple change of variables reveals that a greedy approach is also globally optimal. To see this, note that the bottleneck subscript \( k \) can now be dropped from the notation, so that the only COP constraint that links different time slices (10) becomes:

\[
\sum_{ij} \gamma_{ij} D_{ij}(t - \tau_i) \leq c(t) \quad \forall t. \tag{17}
\]

If we now introduce time-shifted arrival and departure curves \( V_{ij} \) and \( W_{ij} \) that are referred to the bottleneck passage time by means of:

\[
V_{ij}(t) = A_{ij}(t - \tau_i) \quad \text{and} \quad W_{ij}(t) = D_{ij}(t - \tau_i), \tag{18}
\]

then (17) becomes:

\[
\sum_{ij} \gamma_{ij} \dot{W}_{ij}(t) \leq c(t) \quad \forall t, \tag{19}
\]

and the dependence across time slices has been eliminated.

The remaining equations continue to apply as they are, provided \( V \) is substituted for \( A \), \( W \) for \( D \), and \( \delta_i(t) = d_i(t - \tau_i) \) for \( d_i(t) \), as can be seen by replacing \( t \) with \( (t - \tau_i) \) in all of the expressions and using (18) to examine the equivalence. The lower and upper limits of integration of (14) would be \( \tau_i \) and \( t_E + \tau_i \), respectively.

This shows that the optimal action is that which maximizes the total flow in each time slice emitted by the “virtual” origins. One can further show through a change of variable that this subproblem reduces to a continuous knapsack problem with the origins as the “items”, the amount of flow sent by each to the bottleneck as its “item weight”, the bottleneck capacity as the “knapsack capacity”, and the flow sent elsewhere in the network as the “value density” of each item. Thus, the optimal metering rates are given by a greedy procedure in which origins are ranked in reverse order of the percent of flow they send through the bottleneck and after minimum flows have been satisfied at all entries, the remaining capacity is assigned to entries strictly in rank order.\(^5\)

\(^5\) Payne (1973) suggested a similar greedy technique in which the time lags were not taken into account. This is appropriate for systems where the conditions (the arrival and departure curves) change slowly on a time scale which is long compared to the network travel times.
3.3. Networks with a single destination

In a many-to-one network, the problem does not decompose by time slice, but can be shown to be linear. If minimum metering rates (11a) are to be enforced, however, integer variables must be introduced.

To see that the problem is linear, assume for the moment that (11a) can be ignored, and note that the destination subscript \( j \) can be dropped from the formulation. With only one destination, Eqs. (2), (8) and (9) of the COP are irrelevant. Eqs. (6), (7), (10) and (11b) are linear in the departure rates and independent of the remaining variables. Eq. (13) can be replaced with (16a) (with the subscript \( \bullet \) replaced by \( i \)), which is also linear in the departure rates and independent of the remaining variables. Eqs. (3)–(5) and (12) are linear in the cumulative numbers of departures and independent of the remaining variables. Since the differentiation operator (which relates departure rates to cumulative numbers) is linear, the latter equations are linear in the departure rates also. The differentiation relation in discrete time is now:

\[
D_i(t_n) = D_i(t_{n-1}) + (t_n - t_{n-1}) \dot{D}_i(t_{n-1}) \quad \forall i, \forall n \geq 1,
\]

where \( t_n - t_{n-1} = \Delta t \) is a short differential of time (i.e., a time slice). To accommodate (10) in a discrete-time formulation, \( \Delta t \) should be sufficiently small that \( \tau_{ik} \) can be modeled as an integer multiple of \( \Delta t \), for all \( i, k \).

Clearly, then, if (11a) is not part of the formulation (and insofar as the objective function is a linear function of the cumulative departure curves), the discretized COP is now a large-scale linear program with the departure rates and cumulative departures, by time slice, as decision variables. Note that \( d_i(t) \) does not appear in the formulation.

Unfortunately, (11a) is somewhat problematic. A form of the minimum metering rate requirement that avoids the introduction of \( d_i(t) \) is:

\[
A_i(t_n) - D_i(t_n) > R_{ij}^{\text{min}}(t_n) \Delta t \Rightarrow D_i(t_n) \geq R_{ij}^{\text{min}}(t_n) \quad \forall i.
\]

This constraint forces a minimum metering rate at an entry point if enough vehicles are currently in queue at that entry to sustain that metering rate throughout the time slice. The constraint can be expressed in the usual way by means of two linear inequalities if we introduce a binary (0–1) variable \( y_i \) for each entry point, as follows:

\[
R_{ij}^{\text{min}}(t_n) - \dot{D}_i(t_n) \leq My_i,
\]

\[
A_i(t_n) - D_i(t_n) - R_{ij}^{\text{min}}(t_n) \Delta t \leq M(1 - y_i),
\]

where \( M \) is a very large constant. Thus, the system is now a large-scale mixed-integer linear program.

3.4. Demand patterns with constant O–D splits

This final special case assumes that the aggregate demand functions at each origin, \( \{A_o(t)\} \), vary across \( i \) and \( t \), but that the O–D fractions at each origin \( i \), denoted \( \{x_{ij}\} \), are independent of time; i.e., that

\[
A_{ij}(t) = A_i(t)x_{ij} \quad \forall i, j, t,
\]

(24a)
and

\[ \dot{A}_{ij}(t) = \dot{A}_i(t) \lambda_{ij} \quad \forall i, j, t. \tag{24b} \]

This type of scenario might be appropriate for the evening rush hour in an area where the distribution of traffic to different surrounding neighborhoods is independent of when people get off work, or possibly other traffic patterns when detailed O–D data are not available. This simplification has also been exploited in other papers (see for example Yuan and Kreer, 1968 and Wang and May, 1972). Note that \( \lambda_{ij} = \text{constant} \) does not imply that the exit percentages at offramps are constant or independent of the metering strategy, as was assumed in some earlier works.

Recall that one of the two non-linear equations in the COP is (8). In this case it can be replaced with:

\[ \dot{D}_{ij}(t) = \lambda_{ij} \dot{D}_i(t) \quad \forall i, j, t, \tag{25} \]

and (9) is not necessary. To see that this is true, take the derivative of (8) (including the redundant case \( j = \bullet \)) and using (24b), note that:

\[ \frac{\dot{D}_{ij}(t)}{\dot{D}_i(t)} = \frac{\dot{A}_{ij}(t - d_i(t))}{\dot{A}_i(t - d_i(t))} = \lambda_{ij}. \tag{26} \]

The remaining equations can be handled as in Section 3.3 and the system solved for all time slices simultaneously as in that case, e.g., as a large-scale (possibly mixed-integer) linear program.

### 4. Non-anticipatory heuristic algorithm

For situations other than the special cases illustrated earlier, the COP is difficult to solve. The non-linear Eqs. (8) and (11a) and the dependence across time slices exhibited by (10) are the chief sources of this difficulty. This section describes a heuristic algorithm which was developed to solve the COP quickly, avoiding these computational difficulties and using only information that is likely to be available. The algorithm is simple and non-anticipative, globally optimal for some of the special cases, and particularly suited for general cases when traffic conditions vary slowly.

The application situation we have in mind is one where traffic demand and bottleneck capacities change so slowly in time that changes to the state of the system (queues, vehicles in transit, etc.) occurring as a result of the control are not noticeable on a scale of measurement where the trip times \( \{\tau_{ik}\} \) can be discerned. The assumption is useful because it allows us to replace (10) by the same expression without the \( \{\tau_{ik}\} \); i.e., by

\[ \dot{B}_k(t) = \sum_{i,j} \gamma_{ijk} \dot{D}_{ij}(t) \leq c_k(t) \quad \forall k, t, \tag{27} \]

\(^6\) This assumption is most appropriate for small systems, e.g., in which the trip times are on the order of a few minutes, but we believe that the resulting strategy is likely to be very good in general.
which no longer exhibits any linkage in the decision variables across time slices. The physical effect of this modeling change is that traffic pulses no longer will arrive at the bottleneck(s) perfectly coordinated and as a result, transient queues may occasionally form and dissipate on the freeway. The sluggishness of the system, however, ensures that these episodes will be so fleeting and their queues so small that one is justified in neglecting the resulting freeway delay.

With (10) simplified in this way, the only remaining equations in the COP that introduce a linkage across time slices are (8) and (9). Thus, in any case where the demand varies slowly, and where (8), (9) can be replaced by unlinked expressions such as (25) of Section 3.4, the problem can be decomposed by time slice. In this case intuition suggests and inspection of the relevant equations confirms, that a greedy non-anticipatory algorithm that maximizes the sum of the demand rates in each time slice subject to the capacity constraints is optimal.

The sluggishness of the system, however, ensures that these episodes will be so fleeting and their queues so small that one is justified in neglecting the resulting freeway delay.

With (10) simplified in this way, the only remaining equations in the COP that introduce a linkage across time slices are (8) and (9). Thus, in any case where the demand varies slowly, and where (8), (9) can be replaced by unlinked expressions such as (25) of Section 3.4, the problem can be decomposed by time slice. In this case intuition suggests and inspection of the relevant equations confirms, that a greedy non-anticipatory algorithm that maximizes the sum of the demand rates in each time slice subject to the capacity constraints is optimal.

The reader can also verify based on both intuitive and mathematical grounds that the departure rates that can be achieved in each time slice do not depend on the arrival rates of the queued access points. Therefore, the chosen instantaneous controls will not depend on these arrival rates either. Of course, the controls will depend on the status (queued/unqueued) of all the access points and on the arrival rates at unqueued entrances. Insofar as all this information is local (involving events taking place in the neighborhood of the meter) it should be readily available.

The algorithm about to be described applies these ideas for the general case where (25) does not hold. Because the \( z_{ij} \)'s should change slowly, it is proposed to choose the control for each time slice as if the \( z_{ij} \)'s remained fixed at the current values. This is appealing for two reasons: (i) the resulting control rates at any time are the result of solving the optimization problem for a rolling (but long) horizon in which the \( z_{ij} \)'s do not change much, and (ii) the chosen rates leave the system with the shortest possible queues at the end of the horizon. This should prepare the system for future eventualities beyond the horizon in the most "robust" way possible.

While these arguments do not guarantee optimality, they suggest efficiency (even if the O–D demands vary quickly). Thus, the proposed algorithm can serve as a tough benchmark against which more complicated algorithms can be measured.

If the demand curves by O–D pair are approximated by piece-wise linear functions, the algorithm can be streamlined, because then the \( z_{ij} \)'s and the status of all the entrances remain constant for extended periods of time (intervals) during which the controls should not change. Thus, the complete control history can be determined by stepping through time in a finite number of steps of varying size. This is explained in the following two subsections, which show how to determine: (i) the best metering rates during the current time interval (Section 4.1) and (ii) the time when they should be changed again (Section 4.2).

An example is used to facilitate the presentation; see Fig. 3. The network consists of two origins, two destinations and three bottlenecks, whose capacities will be assumed constant, at 1500 vehicles per hour (vph) for bottlenecks 1 and 2, and 2000 vph for bottleneck 3. The O–D data for the example problem are shown as the solid black lines in Fig. 4.

---

7 It is possible to construct contrived examples where the heuristic performs sub-optimally. We expect that many of these situations would be recognizable in practice and therefore would exhibit characteristics which might be exploited in a more appropriate case-specific control strategy.
Fig. 3. Sample network for heuristic algorithm.

Fig. 4. O–D demands and initial controls.
4.1. Determining control during an interval

Let us assume for now that the piece-wise linear O–D demand curves are known and that the control history of the system \( f_{Dij} \) is also known up to the start of the \( n \)th control interval, at \( t = t_n \). This implies that we know \( di(t_n) \), because by virtue of (8):

\[
d_i(t_n) = t_n - A_{ij}(D_{ij}(t_n)).
\]

This information allows us to determine the O–D fractions that will depart each meter an instant after \( t_n \), at \( t_n^+ \), with (26); i.e., with:

\[
\alpha_{ij}(t_n^+) = \frac{A_{ij}(t_n - d_i(t_n)^+)}{A_{ij}(t_n - d_i(t_n)^+)}.
\]

Therefore the control at \( t = t_n^+ \) is determined by maximizing the sum of the departure rates at the current time:

\[
\max \sum_i \dot{D}_{ij}(t_n^+).
\] (28)

The decision variables \( \{\dot{D}_{ij}(t_n^+)\} \) must be non-negative, and may also be constrained by (6), (7) and (11b) with \( t = t_n^+ \). Additionally, the departure rate at an unqueued meter cannot exceed the arrival rate. Thus,

\[
D_{ij}(t_n) = A_{ij}(t_n) \Rightarrow \dot{D}_{ij}(t_n^+) \leq A_{ij}(t_n^+) \quad \forall i.
\] (29)

Furthermore, unlinked bottleneck capacity constraints similar to (27) must also be satisfied:

\[
\sum_{ij} \gamma_{ijk} \alpha_{ij}(t_n^+) \dot{D}_{ij}(t_n^+) \leq c_k(t_n^+) \quad \forall k,
\] (30)

and for the set \( I_{nm} \) of unmetered origins, the departure rates must be consistent with the standard queueing model, as described in Section 2.3:

\[
\dot{D}_{ij}(t_n^+) = \hat{A}_{ij}(t_n^+) \quad \forall i \in I_{nm}.
\] (31)

Finally, a multi-origin version of (16a) can be introduced for \( n \geq 2 \) if one desires to impose some smoothness on the controls.

The metering rates determined for \( t_n^+ \) by means of LP (28)–(31) will remain valid until one or more of the parameters in the LP changes. This can only happen as a result of one of the following four events: (i) a change in \( A_{ij} \) for an unqueued origin, (ii) a change in \( \alpha_{ij} \), (iii) a change in the status of a queued entrance (to unqueued), and (iv) a change in the capacity of a bottleneck, \( c_k(t) \).

Section 4.2 describes a simple procedure to keep track of this information and illustrates it with the example. The complete algorithm has been coded for a PC running Windows and can be applied to a network of arbitrary topology.

For the initial interval of the example problem (starting at time \( t_1 = 0 \) when the system first becomes congested), the program solves the first LP and determines that the appropriate metering rates are 1750 vph for both origins 1 and 2, which correspond to rates of 1500, 250 and 1750 vph for O–D pairs \( O_1 \rightarrow D_1 \), \( O_1 \rightarrow D_2 \) and \( O_2 \rightarrow D_2 \), respectively. The reader can verify that higher
rates are not possible. Fig. 4 shows the tentative construction of the resulting departure curves as gray dotted lines.

The following subsection describes how to determine $t_{n+1}$ from $t_n$ by keeping track of the four relevant events. The relevant procedures are described in the order in which they first occur in the example.

4.2. Determining when to change control

4.2.1. Change in O–D demand

A change in O–D demand is reflected by a bend in (at least) one of the disaggregate arrival curves. In the example problem, this occurs for O–D pair $O_1 \to D_1$ at time $t = 1.0$, $O_1 \to D_2$ at time $t = 1.8$ and $O_2 \to D_2$ at time $t = 2.5$. The appropriate response of the algorithm to this event depends on whether or not the entry point used by the relevant O–D stream is currently queued. If not, the demand change will be manifested at the meters immediately, thereby necessitating immediate re-optimization of the metering rates, as described in Section 4.1. If the entry point is queued, however, no change in control strategy is (yet) necessary, because no change in the composition of the traffic stream currently departing the meters will be immediately evident. However, the mix of destinations amongst vehicles joining the back of that particular queue will have changed and the algorithm needs to identify the position within the queue at which that change occurred. This is illustrated in Fig. 4 by the horizontal dashed lines which extend to the right from the bends in the O–D arrival curves at the times indicated above. As a consequence of the FIFO queue discipline, parallel lines would also be drawn starting at the same time from any arrival curves representing O–D pairs which share this same access point; e.g., as was done for the lower dashed line of the middle ($O_1 \to D_2$) diagram.

In the example problem, because all of the forthcoming demand changes were known ahead of time, the horizontal dashed lines were all plotted at the outset and this step was eliminated from the remainder of the algorithm. However, because the algorithm is not anticipative, each demand change need only be processed exactly as it occurs. Thus, up-to-the-minute demand data can be incorporated, in response to unforeseen changes in traffic patterns. In fact, it should also be clear that if one were to elicit destination information (e.g., at toll booths) prior to entry into the system, then $x_{ij}(t)$ would be known and the need for forecasting O–D curves would be completely eliminated.

4.2.2. Change in O–D mix at head of queue

The O–D mix at the head of the queue changes when a departure curve for that access point crosses one of the horizontal dashed lines mentioned above. As soon as this happens, one or more of the $x_{ij}(t)$’s change and the control must be re-optimized. It should be noted that these crossings always occur simultaneously for all O–D pairs sharing an access point because of the FIFO discipline.

Fig. 5 shows that in the example problem, an event of this type first occurs at time $t = 2.0$ for O–D pairs $O_1 \to D_1$ and $O_1 \to D_2$. Optimization reveals that the best metering rates after time $t = 2.0$ are 3000 and 500 vph, which correspond to O–D departure rates of 1500, 1500 and 500 vph. Fig. 5 also shows departure curves beyond $t = 2.0$, reflecting the new metering rates. The vertical
4.3. Queue dissipation

When a departure curve meets an arrival curve, the queue upstream of the meter serving that O–D pair has dissipated. Because constraint (29) then becomes relevant for that meter, the control should be re-optimized. Again, by virtue of FIFO, this event always occurs simultaneously for all O–D pairs sharing an access point. For the example problem, the first queue (at origin $1$) dissipates at time $t = 3.2$ and the control is then re-optimized.

Another change in O–D mix at the head of the queue at origin $2$ occurs at time $t = 4.1$, and the final queue is exhausted at time $t = 5.7$; see Fig. 6. No more queues grow from then on. Com-
puter-generated, Table 1 summarizes all of the events (and their consequences) which the algorithm processed for the example problem.

4.3.1. Change in bottleneck capacity

If the capacity of a bottleneck changes during the control period, perhaps due to some effect of non-recurrent congestion, an immediate control update is necessary with the new values of $c_k(t)$. This type of event is very simple and was not included in the example.

5. Summary and conclusions

This paper has explored control strategies for metered networks with unique O–D paths, with particular emphasis on those situations where some time-dependent O–D information is known in
The application in mind is small freeway networks with signalized meters, such as circumferential ring freeways or congested weaving sections where upstream access could be controlled. Other transportation networks might also be good candidates, such as signalized roundabouts. The mathematics of our approach are general and might be applied to any time-dependent constrained resource-allocation problem. We present a formulation in continuous time for the general case and highlight the solution difficulties associated with this formulation. Exact methods are described for some important special cases and a non-anticipative heuristic algorithm is described for more general situations.

The COP developed herein incorporates several features not adequately treated in previous similar efforts, including the modeling of trip time lags from origins to bottlenecks and the enforcement of FIFO queueing upstream of the meters. The requirement that no internal queues be generated is severe, however, and future studies should investigate the potential benefits of allowing internal queues, as well as the modeling framework required to do so accurately.

The COP can be applied to sub-networks within a larger topological structure; it is in these circumstances that the exact and efficient algorithms proposed for the four special cases are expected to be most useful. While the computational difficulties associated with the general COP provide part of the impetus for the heuristic algorithm described herein, it is likely nonetheless that the COP can be re-cast in some discrete, non-linear form and future studies should address this possibility. Dynamic programming (enumeration) and bi-level programming in a discrete time context have been considered, but are practical only for small networks, as explained in Lovell (1997).

Table 1
Numerical details of algorithm operation on example data

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Departure rates</th>
<th>Queues</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>O₁ → D₁</td>
<td>O₂ → D₂</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1500</td>
<td>1750</td>
</tr>
<tr>
<td></td>
<td></td>
<td>250</td>
<td>1750</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>50/50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.8</td>
<td>1750</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>33.3/66.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>33.3/66.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.27</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.5</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.2</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.1</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.7</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.7</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3/66</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>3/66</td>
</tr>
</tbody>
</table>

a The number of distinct mixes of destinations present in the queue upstream of the meter.
b This refers to the split between vehicles destined for 1 and vehicles destined for 2.
c No destination split is shown because only one O-D pair uses this entry point.
The heuristic algorithm is founded on intuitive grounds as well as computational expediency. It follows a general, simple rule: that leaving residual queues as short as possible, from one time interval to the next, is as good a strategy as could be hoped for without trying to predict the future. It does so using very little data, none of which is anticipated. As further testimony, if the time lags to bottlenecks are so short that they can be neglected, then the heuristic solves special cases such as those described in Sections 3.1 and 3.2 exactly.

The heuristic may cause small internal queues to be generated, but the mechanism for such is equally likely to cause the bottlenecks to be temporarily under-utilized. Thus, the process should be self-regulating and any such queues should be transient.

The heuristic is fast (comparable with the exact methods for the special cases). Its critical computational task is the solution of a linear program. A problem with \( I \) origins would require the solution of LP’s with a number of decision variables on the order of \( I \), roughly one for each origin. With non-negativity, minimum, maximum and smoothness constraints for each decision variable, plus capacity constraints for \( K \) bottlenecks, the total number of constraints would be on the order of \( 4I + K \). Freely available LP algorithms can easily handle a problem of these dimensions for any realistically sized network.

Acknowledgements

This work was supported under a research grant from California PATH.

References


