Solving large structured Markov Decision Problems for perishable inventory management and traffic control

Haijema, R.

Publication date
2008

Chapter 7

Single intersection with arrival information

7.1 The control problem with arrival information

Current technology allows not only to measure the number of cars waiting at each queue, but is also capable of detecting in the vicinity of intersections ([93], [65]). Information on near future arrivals comes from induction loops cut into the surface of the roads or from cameras positioned above or along the approach ([93], [65]). In the latter case, cars driving in the range of a camera are observed and by using image processing techniques their position on the approach and the driving speed can be estimated. The traveling speed of cars can also be derived from the time elapsed between a car passes two subsequent induction loops. The controller translates the information in an estimated time until a car reaches the tail of a queue (ToQ).

In the decision model that we introduce in the next section we assume that the estimated arrival times are rounded to slots of 2 seconds: cars that arrive at the ToQ within $\langle m-1, m]$ slots from now are said to arrive m slots from now. For each car on the approach one can thus estimate the number of slots it takes to reach the ToQ. This arrival information can be translated into a vector $\mathbf{a}^f = (a_1^f, \ldots, a_{M(f)}^f)$ for each flow f, with elements a_m^f the number of cars in flow f that arrive m slots from now at the end of Q_f. $M(f)$ denotes the (maximum) number of slots for which arrival information is available for flow f. For some flows only information on the queue length is available, and no information on near-future arrivals is available by a lack of cameras or upstream induction loops: $M(f) = 0$.

The information on the arrival times of near future arrivals can be used to reduce the long-run average waiting time by anticipating these arrivals. Inclusion of the arrival information in the MDP model, that we have introduced in Chapter 6, results in an explosion of the state space. The state of the system will then be $(x, \mathbf{q}, \mathbf{a}^1, \mathbf{a}^2, \ldots, \mathbf{a}^F)$ instead of $(x, \mathbf{q})$.

For example, consider the simple F4C2 case with only four flows that are served in two combinations, and suppose that for each flow 5 slots arrival information is available. When cars keep a safe driving distance of 1 slot, a^f_m can take only 2 values: 0 or 1 car. For each flow f, the vectors $\mathbf{a}^f$ can thus take $2^5 = 32$ possible values. The total number of arrival patterns over the next 5 slots is thus $32^4 \approx 1$ million. In Chapter 6, we have discussed that the number of traffic light states in the MDP model is $C \cdot (1 + 3) = 2 \cdot 4 = 8$ and the number of queue states is 10^4 when in the MDP model queues are truncated at $Q = 9$ cars. The total number of states becomes thus already $8 \cdot 10^{10}$ for a simple intersection with only 4 flows.

Clearly, the optimal MDP policy cannot be computed within a reasonable time. Fortunately, as we have learned in Chapter 6, a one-step policy improvement over FC is possible, since under FC the state space can be decomposed into F subspaces. In this chapter, a similar approach is followed and the RV policies are extended, such that the available arrival information is taken into account. This extension is of particular interest for the control of networks of intersections, as we will discuss in Chapter 8.

Outline

After having extended the RV policies in the next section, an advanced simulation model is presented, through which we test the policies in a quite realistic setting. The simulation model is discussed in Section 7.3. The new policies are put to the test for the F12C4 infrastructure, which is interesting for its combinations being asymmetric in the number of flows. The results are reported in Sections 7.4 and 7.5, for varying scenarios concerning the available number of slots of arrival information.

7.2 One-step policy improvement of FC

Through a one-step policy improvement algorithm one may improve FC and hopefully obtain a nearly optimal policy. Therefore the relative values of states $(t, \mathbf{q}, \mathbf{a}^1, \ldots, \mathbf{a}^F)$ are to be computed.

7.2.1 Relative values of states under FC

Decomposition

Under FC the signal state is a single number $t \in \{1, 2, \ldots, D\}$ indicating the slot number in the cycle. The signal state changes each slot from t to $t+1$ (or to 1 when $t = D$) independent of the state at the queues. Therefore the state space $(t, \mathbf{q}, \mathbf{a}^1, \mathbf{a}^2, \ldots, \mathbf{a}^F)$ can be decomposed into subspaces $(t, q_f, \mathbf{a}^f)$ for each flow f. The relative values of states $(t, \mathbf{q}, \mathbf{a}^1, \ldots, \mathbf{a}^F)$ can thus be computed flow-by flow for states $(t, q_f, \mathbf{a}^f)$.

State transitions

Before we show how the relative values of states $(t, q_f, \mathbf{a}^f)$ are computed for each flow f, the state transitions under FC are examined. We impose the following assumptions to structure the state transitions in the decision model, such that the arrival information is accurate:

- all cars drive at constant and identical speed,
- cars are not allowed to change lanes,
- no cars reach their destination somewhere along the approach.

Example − Consider the two successive state transitions depicted in Figure 7.1. The first picture shows some initial state $(t, q_f, \mathbf{a}^f) = (\text{"Y1"}, 2, 1, 0, 0, 1, 0)$. Suppose, signal state t is the first yellow slot ("Y1") of f, two cars, labeled A and B, are queued, and another two cars (C and D) are on their way to the intersection. One slot later, the new signal state relates to the second yellow slot ("Y2") to f, and car A has left the queue. The queue state stays $q_f = 2$, since car C has joined the queue. Car D has moved one slot, and is thus only 3 slots away from the end of the queue. After another slot, B has left the queue and the queue consists of car C only, car D is now two slots away from the tail of the queue.

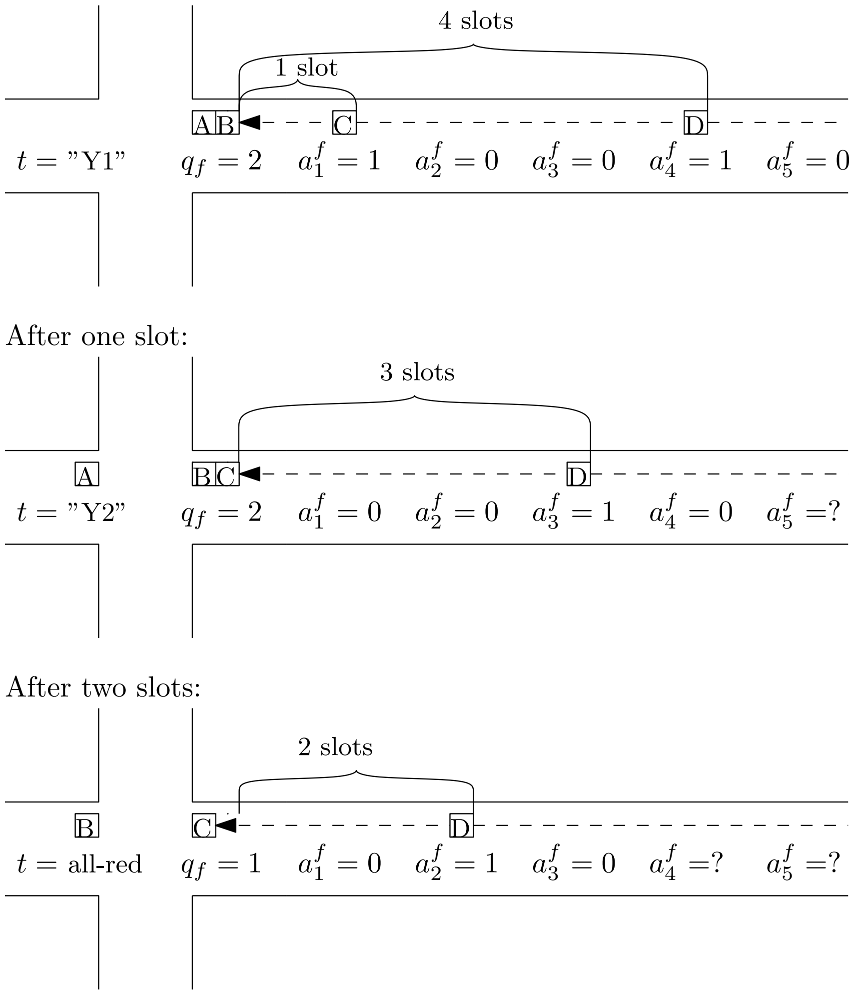

Figure 7.1: Successive transitions under FC with arrival information for a single flow.

Remark – In reality, the distance between car C and D is a bit more than 2 slots, since car C has moved forward in the queue. In the decision model this phenomenon is ignored by assuming vertical queueing: queued cars are assumed to take zero length. However, in the simulation model to be discussed in Section 7.3, we assume horizontal queueing: queued cars take a positive length.

Value iteration for relative values of states $(t, q_f, \mathbf{a}^f)$ under FC

Similar to Equation (6.22) in the previous Chapter, the relative values under FC can be computed by:

$$\lim_{N\to\infty} \sum_{f=1}^{F} \frac{1}{D} \sum_{d=0}^{D-1} \left(V_{N+d}\left(t, q_f, \mathbf{a}^f\right) - (n+d) \cdot g^{(f)} \right), \tag{7.1}$$

where $g^{(f)}$ and $V_{N+d}\left(t, q_f, \mathbf{a}^f\right)$ are approximated by a value iteration algorithm.

When five slots of arrival information is available to flow f, then $V_N^f\left(t, q_f, \mathbf{a}^f\right)$ is recursively computed from Equation (7.2), by successively increasing n, starting with $n = 0$ and $\mathbf{V}_0^f = \mathbf{0}$:

$$\begin{aligned} V_{n+1}^f\left(t, q_f, a_1^f, a_2^f, a_3^f, a_4^f, a_5^f\right) = q_f + \quad & (1-\lambda_f) \quad \cdot V_n^f\left(t, (q_f + a_1^f - \Delta_t^f)^+, a_2^f, a_3^f, a_4^f, a_5^f, 0\right) \\ + \quad & \lambda_f \quad \cdot V_n^f\left(t, (q_f + a_1^f - \Delta_t^f)^+, a_2^f, a_3^f, a_4^f, a_5^f, 1\right), \end{aligned} \tag{7.2}$$

where $\Delta_t^f = 1$ when t grants green or yellow to flow f, and 0 when t implies red to flow f.

The relative value in (7.1) can be used in a one-step policy improvement algorithm to construct an RV policy in which arrival information is used. The gain, or the long-run average costs per slot under FC, $g^{(f)}$, follows from the vector of constants $\lim_{N\to\infty}(\mathbf{V}_N^f - \mathbf{V}_{N-D}^f)$.

Complexity

Since $\mathbf{a}^f$ is a vector with dimension $M(f)$, the state description $(t, q_f, \mathbf{a}^f)$ is high-dimensional. The number of possible vectors $\mathbf{a}^f$ is $2^{M(f)}$, since at most 1 car arrives per slot. Even when Q and D are set as high as 100 cars respectively 100 slots and when $M(f) = 5$ the number of states, $D \cdot (1 + Q) \cdot 2^{M(f)}$, is still well below 1 million. Computation of the relative values for all states $(t, q_f, \mathbf{a}^f)$ under FC is then still possible. When $M(f)$ is larger, solving all relative values may consume too much time and memory. Therefore we discuss an alternative approach of computing the relative values.

Online computation of relative values with arrival information

Instead of computing the relative values for all states $(t, q_f, \mathbf{a}^f)$ under FC off-line, one may compute these relative values online. Therefore let $q_{f,m}$ denote the number of cars waiting at queue f after m slots. Starting with $q_{f,0} = q_f$, one may compute $q_{f,1}$, to $q_{f,M(f)}$, as the number of cars waiting at queue f can be predicted using the arrival information $\mathbf{a}^f$ for the coming $M(f)$ slots.

For computing the relative values for all states $(t, q_f, \mathbf{a}^f)$, one first computes for flow f the total waiting costs over the coming $M(f)$ slots, for which arrival information is available. Next, one adds to this the relative value for all states $v^f_{rel}(t', q_{f,M(f)})$ as terminal costs of ending in state $(t', q_{f,M(f)})$ after $M(f)$ slots.

The computation of the waiting costs over the first $M(f)$ slots is relatively easy, since the transitions happen deterministically during the first $M(f)$ slots, assuming the arrival information is accurate. One evaluates the state $(t', q_{f,M(f)})$ that is reached after $M(f)$ slots, by the relative value of state when no arrival information is available: i.e. $v^f_{rel}\left(t', q_{f,M(f)}\right)$ as defined in (6.22) in the previous chapter. In the next two subsections, the online evaluation of decisions under the RV policies is explained in more detail.

Remark – Online computation yields the same relative values but is favored above the off-line computation of $V^f_n(t, q_f, a^f_1, \ldots, a^f_{10})$, when $M(f)$ is large. When $M(f) = 10$, off-line-computation would imply the computation of about 10^7 elements $V^f_n(t, q_f, a^f_1, \ldots, a^f_{10})$ for each flow f. Whereas online computation requires only 10,000 relative values $v^f_{rel}\left(t', q_{f,M(f)}\right)$ to be evaluated for each flow f, as if no arrival information is available. The number of operations to be executed online to evaluate a single time jump is only $\mathcal{O}(M(f))$.

7.2.2 Extension of RV1 and RV1M with arrival information

Using a one-step policy improvement algorithm with initial policy FC, we have derived in the previous chapter the policies RV1 and RV1M. These policies are now extended such that arrival information is included in the selection of the best time jump τ.

The number of slots of arrival information is added to the name of the policy: e.g. RV1(5) assumes that accurate arrival information is available for all flows for the next 5 slots; under RV1(5,0,5,0) 5 slots arrival information is used for combinations 1 and 3, while no information is available to C_2 and C_4.

Extension of RV1

In order to find out what the best decision τ is, first the waiting costs over the next $M(f) = 5$ slots are determined, as illustrated in Figure 7.2. Next, the relative value of the state visited after 5 slots is added as terminal costs.

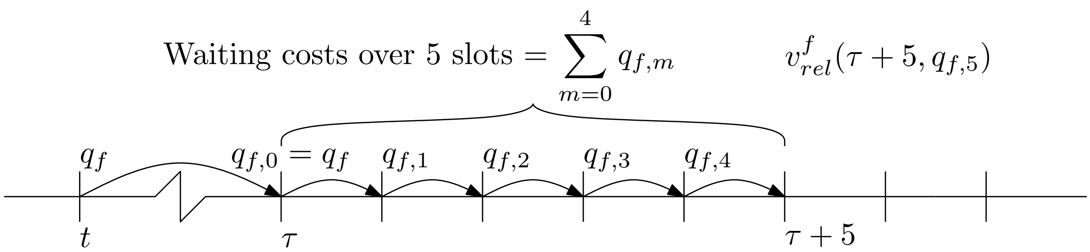

Figure 7.2: Evaluating RV1(5): RV1 with arrival information for the next 5 slots.

The state visited after $M(f) = 5$ slots is derived as follows, using the notation $q_{f,m}$ for the queue length after m slots. The initial queue length $q_{f,0}$ is q_f and changes after 1 slot into $q_{f,1} = \left(q_{f,0} + a^f_1 - \Delta^f_\tau\right)^+$, with Δ^f_τ is the maximal number of the departures from queue f within slot τ, which is in our case 1 when τ is a departure slot to flow f and 0 otherwise. Note that by this definition a car that arrives at an empty queue will depart during the very same slot when the respective lights are a green or yellow. Since the car is not queued it contributes zero waiting costs.

The queue length m slots after a time jump to slot τ, $q_{f,m}$, is recursively set by:

$$q_{f,m} = \begin{cases} \left(q_{f,m-1} + a^f_m - \Delta^f_{\tau+m-1}\right)^+ & \text{if } m \in \{1, 2, \ldots, M(f)\} \\ \\ q_f & \text{if } m = 0, \end{cases} \tag{7.3}$$

with $\Delta^f_y = \Delta^f_{y-D}$ when $y > D$ such that $y - D \in \{1, 2, \ldots, D\}$.

The waiting costs at queue f over the first $M(f)$ time slots is simply the sum $\sum_{m=0}^{M(f)-1} q_{f,m}$. After these $M(f)$ slots $q_{f,M(f)}$ cars are waiting at queue f, and the signal state is $\tau + M(f)$.

The relative cost value to flow f of starting in state $(\tau, q_f, \mathbf{a}^f)$ thus becomes:

$$\sum_{m=0}^{M(f)-1} q_{f,m} + v^f_{rel}\left(\tau + M(f), q_{f,M(f)}\right). \tag{7.4}$$

The relative value of state $(\tau, q_f, \mathbf{a}^f)$ can be computed online for each flow, once $v^f_{rel}(t, q_f)$ is computed and stored for each state (t, q_f) according to (6.22).

RV1(·) actions

The selected time jump τ in state $(\tau, q_f, \mathbf{a}^f)$ under RV1(·) follows from:

$$\arg \min_{\tau \in \mathcal{T}(t,\mathbf{q})} \sum_{f=1}^{F} \left(\sum_{m=0}^{M(f)-1} q_{f,m} + v^f_{rel}\left(\tau + M(f), q_{f,M(f)}\right) \right). \tag{7.5}$$

The cyclic version of RV1(·) with arrival information is simply obtained by changing the acyclic action space $\mathcal{T}(t, \mathbf{q})$ in Equation (7.5) by $\mathcal{T}^{\mathrm{C}}(t, \mathbf{q})$.

Remark – Note that the process of looking ahead a number of slots is similar to the evaluation of jumps under RV2. Big difference is that now the first $M(f)$ transition happen deterministic and are evaluated online. Furthermore, we do not need to subtract $m = M(f)$ times the gain, $g^{(f)}$, since $M(f)$ does not depend on the decision τ. (However, as we will see in the next subsection, one has to compensate for unequal horizon lengths when extending RV2.)

Extension of RV1M

The acyclic policy RV1M can also be extended to include arrival information. Under RV1M the evaluation of the waiting costs over the first $M(f)$ slots is done in a very similar way. The notations become more evolved, since under RV1M one optimizes over all possible cycles ϕ. Since slot numbers $1, 2, \ldots, D$ relate to the base cycle 1-2-...-C -1-2-etc., the slots are shifted by $\delta_f(\phi)$ slots under cyclic order ϕ , as discussed in Section 6.5.2.

Consequently we should replace τ by $\tau + \delta_f(\phi)$ in the definition of $q_{f,m}$ in (7.3):

$$q_{f,m} = \begin{cases} \left(q_{f,m-1} + a_m^f - \Delta^f_{\tau+\delta_f(\phi)+m-1}\right)^+ & \text{if } m \in \{1, 2, \ldots, M(f)\} \\ \\ q_f & \text{if } m = 0. \end{cases} \tag{7.6}$$

RV1M(·) actions

The selected time jump τ in state $(\tau, q_f, \mathbf{a}^f)$ under RV1M(·) follows from Equation (7.7):

$$\arg \min_{\substack{\tau \in \mathcal{T}(t, \mathbf{q}) \\ \phi \in \Phi(\tau)}} \sum_{f=1}^{F} \left(\sum_{m=0}^{M(f)-1} q_{f,m} + v^f_{rel}(\tau + \delta_f(\phi) + M(f), q_{f,M(f)}) \right). \tag{7.7}$$

7.2.3 Extension of RV2 and RV2M with arrival information

The inclusion of arrival information into RV2 is much more complicated, at least in its notations. Remember under RV2, two time jumps τ_1 and τ_2 are considered. Instead of presenting an algorithmic description we illustrate in Figure 7.3 how to evaluate a decision (τ_1, τ_2). We distinguish two cases, regarding the amount of arrival information $M(f)$ in relation to the time $(r - \tau_1)$ until the next green period, which would start at slot r when not interrupted by τ_2:

Case I. $M(f) > r - \tau_1$: the horizon of arrival information is longer than the time between the first and the second jump,

Case II. $M(f) \leq r - \tau_1$: no arrival information is available after the jump to τ_2.

In Case I, one simply determines the waiting costs over the next $M(f) = 5$ slots and add to it the relative value of the state in which one ends after the $M(f) = 5$ slots. To evaluate the first part one has to be aware of the second time jump that breaks FC.

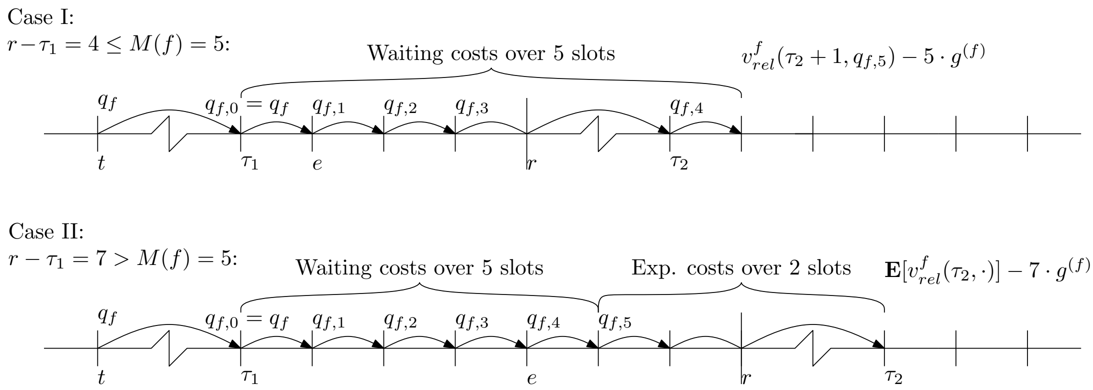

Figure 7.3: Evaluating RV2(5): RV2 with arrival information for the next 5 slots.

In Case II, the evaluation of (τ_1, τ_2) consists of three parts. For the first part arrival information is available and each transition happens deterministically, for the second part one computes the expected waiting costs over the slots until the second time jump τ_2, and finally the third part is the expected relative value of the states in which the system may end after $r - \tau_1$ slots. In fact, the latter two parts correspond to the evaluation of the decision $(\tau_1 = r - 2, \tau_2)$ under RV2, starting in state $(r - 2, q_{f,5})$. To compensate for unequal horizon lengths, one has to subtract $M(f)$ times the average cost per slot, $g^{(f)}$.

Both RV2 and RV2M can be extended this way, but we skip the formal notation of how the actions are selected.

7.3 The simulation model

In the next section the extended RV policies that include arrival information are tested by simulation. In this section, the simulation model of the previous chapter is extended, such that arrival information is available in the model. The extended simulation model is more realistic: cars are no longer stacked vertically, which was the case in the previous chapter. Instead cars are queued horizontal and we assume each queued car takes a fixed length, of say 7 meters. In addition, we allow cars to travel at different speeds.

In the next subsections we discuss principal components and characteristics of the simulation model.

7.3.1 Discrete time

As before the simulation happens in discrete time, since the lights are adjusted every 2 seconds (=1 slot). Information available, at the start of each slot, are the state of the traffic light, the length of the queues and the estimated times at which upstream driving cars arrive at the queue.

7.3.2 Car arrivals

In the simulation model randomly generated car arrivals enter a lane at some fixed distance from the respective stopping line, say at 500 meter. When cars travel at a speed of 50 km/h, it takes a car 18 slots to travel to the downstream stopping line. For each lane car arrivals are generated according to a Bernoulli experiment (0 or 1 arrival per slot per flow). Since we consider an intersection in isolation, we do not consider batch or platoon arrivals.

When an approach consist of three parallel (adjacent) lanes (e.g. for through-traffic and left and right turning traffic), maximal three cars may arrive at the approach per slot. Cars generated in the same slot at parallel lanes and that travel at identical speed, keep driving next to each other till they join their respective queues. When car speeds differ, a car may overtake another car that drives on another lane. However, a fast driving car may not change lane to overtake a slow moving car that drives on the same lane. In fact we do not allow cars to change lane, except when crossing an intersection. All generated cars stay in the system until they have passed the stopping line: no car has a destination along the approach to the intersection.

7.3.3 Car speed and arrival information

Upon entering the system a car gets assigned a *'desired' traveling speed*. The actual traveling speed is lower, when a car has to slow down because of slower moving traffic directly in front of it. A car keeps on driving at its desired speed until it joins a queue, or until it has to adjust its actual speed to slow moving cars directly in front of it. At any time cars keep a safe driving distance of 2 seconds (=1 slot). Speed adjustments are made instantaneously in zero time.

In cases where traveling speeds do vary, desired traveling speeds are generated randomly from a triangular distribution Tri$(a = 40, b = 50, c = 60)$, with mean and most likely speed set to 50 km/h, the minimal desired speed is 40km/h and maximum speed is 60km/h. According to this distribution: about 50% of the car drivers wish to drive at a constant speed between 47 and 53 km/h, 75% of the car drivers has a desired speed between 45 and 55 km/h. If not hampered by other cars a car that drives 60km/h takes only 30 seconds (15 slots) to reach the stopping line, which is located 500 meters downstream. A slowest moving car takes 45 seconds to travel the same distance.

The extended RV policies, introduced in this chapter, require estimated arrival times as input. To be more precise for a number of future slots ($M(f)$), one needs to know whether a car is expected to arrive or not at each of flows. Therefore the position of each car in the lanes is tracked by which the controller estimates the time it takes until a car joins a queue. Since cars are queued horizontally, queue spill back happens: the end of the queue moves upstream when cars join the queue and the lights are red. In estimating the arrival time, one thus considers the current length of the queue in meters, the actual position of the car and of all cars in front of it. When the driving speed of a car is not known to the controller, the time it takes before the car joins the queue is estimated based on the mean desired speed. Again, the length of a queue is computed as if queued cars take 7 meters each.

7.3.4 Departure process

Cars leave a queue one-by-one keeping an traveling distance of 1 slot. When passing the stopping line cars accelerate in zero time to their desired traveling speed and immediately leave the system. Blocking of the intersection by other cars does not happen. Furthermore, it is assumed that conflicting flows are not served simultaneously, hence, whenever a light shows green or yellow one car (if present) may leave the queue. (We have discussed already at the end of the previous chapter that the model can be modified such that conflicting flows are part of the same combination.) All other cars queued shift one position in the queue. If a queued car takes 7 meters, queued cars travel 7 meters a slot (3.5 m/s or 12.6 km/h), during green and yellow slots.

7.3.5 Queueing process

The simulation model is microscopic, since individual cars are distinguished and their position and speed in the lanes is tracked. The distance of a car to the queue changes over a slot, since the car itself moves towards the queue, but also the end of the queue moves up and downstream when cars are joining or leaving the queue. The arrival process of cars at the end of the queue is then not quite a Bernoulli process, also because cars travel in platoons caused by slow-moving traffic that delays other cars. In the Markov chain models by which feasible decision are evaluated, we stick to Bernoulli arrival processes.

An ambiguity – The rate at which cars arrive at the queue is not constant, when queued cars take non-zero length. In the MC model cars arrive at queue f at a fixed rate λ_f, while in the simulation cars enter the lane at an fixed upstream point at rated λ_f. Although the cars enter lane f at a fixed rate, they do not arrive at a fixed rate at the queue. The arrival rate is a bit higher when the position of the Tail of the Queue (ToQ) moves upstream, i.e. when the lights are red. Similarly the effective arrival rate is somewhat lower when the ToQ moves downstream, i.e. when the lights are green. The differences are expected to be small, but present as illustrated by the next example

Example – Suppose cars arrive at rate $\lambda = 0.2$ cars per slot at a lane and get assigned a constant speed of 50km/h, which equals $13\frac{8}{9}\text{ms}^{-1}$ or $27\frac{7}{9}$m. per slot. Then on average every 5 slots (10 seconds) a car enters the lane. Suppose two cars driving along the lane keeping a distance of 5 slots, which corresponds to $5 \cdot 27\frac{7}{9} = 139$ meters (from front bumper to front bumper). Just before the first car arrives at the queue the second car is still about 139 meters away from the ToQ. Just after the arrival of the first car at the ToQ, the queue length increases by seven meters when the light shows red. Hence the distance between the second car and the ToQ drops instantaneously, from 139 to 132 meters. Hence the second car reaches the end of the queue after about 9.5 seconds instead of after 10 seconds. When the light shows red the effective expected inter-arrival time is thus 9.5 seconds, which relates to an arrival rate of 0.21 instead of 0.2. When the lights show green the arrival rate is lower than 0.2 since the ToQ moves downstream.

The queue dynamics cause slight differences between the average waiting time reported by the simulation model and those computed by solving Markov chains. The Markov chain approach underestimates the average waiting time, since it assumes a constant arrival rate. The results are small and may be of no practical importance, given that we have simplified the process of joining a queue: cars may adjust their traveling speed based on the color of the light. When in the simulation model cars speeds are identical and queued cars take zero length, the results of the MC and the simulation model coincides.

7.3.6 Waiting time definition

Throughout the Chapters 6 to 8 we use the same definition of waiting time: the time spent in the queue. Nevertheless, we pay here some attention to this definition, since individual waiting times depend on whether cars are queued vertically, as in the previous chapter, or horizontally (as in this chapter).

For example, consider a car drives 55 meters upstream from the stopping line at speed 50km/h, and 4 cars are queued. When queued cars are queued vertically, i.e. queued cars take zero length, it will join the queue after 2 slots. However, when queued cars take 7 meters each and the lights show red for the coming slot, then the ToQ is located $4 \cdot 7 = 28$ meters upstream. Consequently the car is just 27 meters away from the ToQ, and it will join the queue already after 1 slot. The average waiting time for this car is consequently 1 slot higher than under vertical queueing. In queueing theory one usually assumes vertical queueing; then the waiting time is computed as if cars drive at normal speed to the stopping line.

We believe that the given definition of waiting time under horizontal queueing resembles the waiting time experienced by car drivers, although one may want to discriminate between being queued in front of a red light and being queued in front of a green light. Modifications at this point are possible but not considered.

7.4 Base cases and results

The extended RV policies that include arrival information are tested for the F12C4 infrastructure (see Figure 7.4) with 4 approaches each consisting of 3 lanes of 500 meters length. The following data further characterize the base case:

- Car arrivals happen 500m. upstream by Bernoulli experiments with probabilities λ_f,
- The arrival rates are identical to all flows f: $\lambda_f = \lambda$,
- Individual desired cars speeds (in km/h) are drawn from Tri(40,50,60),
- Five slots of arrival information is used for each flow, i.e. the arrival times are estimated for cars that are $5 \cdot 27\frac{7}{9} \approx 139$ meters away from the ToQ (a car that is between 111 and 139 meters from the ToQ is modeled to arrive after 5 slots when no cars travel in front of it and the lights stay red for 5 slots; when 4 cars travel in front of it, the car reaches the ToQ already after 4 slots).

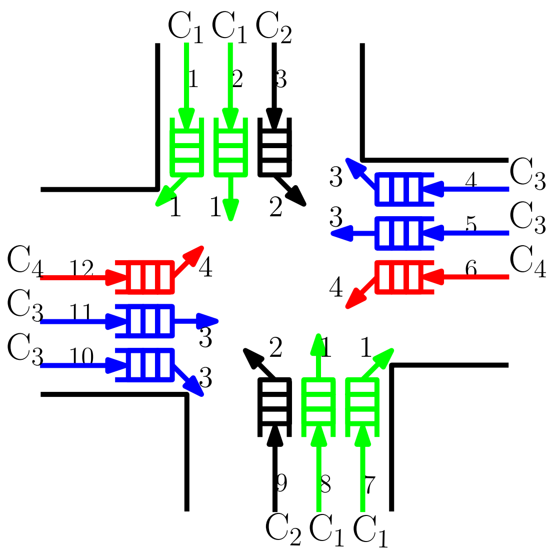

Figure 7.4: "F12C4" serve 12 flows in 4 asymmetric combinations

We investigate for varying loads $\rho = 4\lambda$ the impact of arrival information on the overall average waiting time. For $\rho = 0.6$ we check whether thin combinations of only 2 queues (C_2 and C_4) suffer a higher waiting time than thick combinations of 4 queues (C_1 and C_3).

A sensitivity study with respect to the number of slots of arrival information, and asymmetric arrival rates follows in Section 7.5.

7.4.1 Vary workloads

In Table 7.1 we report the overall average waiting time in seconds at loads $\rho = 0.4$, 0.6 and 0.8. The arrival rates are identical: 0.1, 0.15, and 0.2 respectively. As expected, the average waiting time for FC, RV1 and RV1M are somewhat higher than those reported in the previous chapter in Table 6.4, because now cars are queued horizontally and cars arrive in platoons due to speed differences. The difference is primarily the traveling time from the ToQ to the stopping line when traveling at normal speed. The difference is greater when the workload of the intersections is higher, since the average queue length is higher.

Taking into account 5 slots of arrival information yields, under cyclic RV1(5), a reduction of the overall long-run average waiting time by 6% when the workload is 0.4. The reduction is only 3% when the workload is 0.8.

Under acyclic control similar improvements are achieved by considering 5 slots of arrival information. Compared to cyclic RV1, the waiting time is reduced by 21% under RV1M(5) when the workload is 0.4. Again the difference is much smaller (3.5%) when the workload is 0.8. Just as we have seen in the previous chapter, the difference between cyclic and acyclic control is small when the workload is high.

To save space, the results for RV2(5) and RV2M(5) are not reported. Furthermore, the average waiting time under these policies are almost identical to those under RV1(5) and RV1M(5).

Table 7.1: Mean waiting time (in sec.) for partly-asymmetric F12C4 at varying loads (ρ).

Rule	$\rho = 0.4$		$\rho = 0.6$		$\rho = 0.8$	
Cyclic policies:						
RV1	13.9		20.2		44.2	
RV1(5)	13.1	−6.0%	19.1	−5.1%	42.8	−3.1
FC	15.5	+11.8%	24.9	+23.6%	53.4	+21.0%
FC cycle length (in sec.)	32		40		88	
FC departure times (in sec.)	(6, 6, 6, 6)		(8, 8, 8, 8)		(20, 20, 20, 20)	
Acyclic policies:						
RV1	12.8	−7.7%	19.7	−2.2%	44.2	0.0%
RV1M	12.2	−12.5%	19.1	−5.1%	44.0	−0.4%
RV1(5)	11.9	−14.6%	18.7	−7.4%	42.8	−3.0%
RV1M(5)	11.0	−21.1%	17.9	−11.4%	42.6	−3.5%

7.4.2 Average waiting time per queue at load 0.6

For the base case with workload 0.6 (arrival rates are 0.15), we are interested in whether the thick combinations (C_2 and C_4) benefit more from having arrival information than thin combinations (C_2 and C_4). Therefore consider Table 7.2, which reports the average waiting time for each combination under the different policies.

Accounting for 5 slots of arrival information, is in favor of both thick and thin combinations: the average waiting time at both combinations is reduced by about 1 second, which is about 5%. Under acyclic control the thicker combination benefit slightly more from having arrival information. For example, under RV1M(5) the waiting time at C_1 and C_3 has been reduced by 1.4 seconds compared to RV1M, which is about 8.5%, the waiting at the thin combinations is reduced by only 4.5% (1.1 seconds).

Remark - Although not reported in the table, of all cyclic policies, RV2(5) gives the lowest average waiting time at the thick combinations: 15.3 seconds compared to 17.5 seconds under RV1(5). Further we observe that acyclic control is in favor of the combinations C_1 and C_3 with 4 queues, while the waiting time at C_2 and C_4 are a bit higher but still lower than under FC. The lowest overall mean waiting time when including 5 slots arrival information comes from RV1(5) and RV1M(5).

Table 7.2: Mean waiting times (in sec.) for F12C4 ($\rho = 0.6$ and $\lambda = 0.15$).

Rule	*EW* overall		*EW* C_1, C_3	*EW* C_2, C_4
Cyclic policies:				
RV1	20.2		18.5	23.5
RV1(5)	19.1	−5.1%	17.5	22.5
FC dep. times 8, 8, 8, 8 sec.	24.9	+23.6%	24.8	24.9
Acyclic policies:				
RV1	19.7	−2.2%	17.5	24.2
RV1M	19.1	−5.1%	16.5	24.4
RV1(5)	18.7	−7.4%	16.3	23.4
RV1M(5)	17.9	−11.4%	15.1	23.3

7.5 Sensitivities

In this section we check for the F12C4 case with load 0.6,

- how the results change when more or less slots of arrival information are used,
- the impact of having no arrival information for some flows and
- the importance of arrival information to a combination that is three times as thin than the other (two or three) thick combinations.

7.5.1 Amount of arrival information

On the one hand, one may expect that decisions will improve when more arrival information is available, and used, in controlling the lights. On the other hand, a few slots of arrival may be sufficient, since the state of the lights can be revised at the start of any slot (except during the switching phase). Since during the two yellow slots of a switching phase cars may still pass the stopping line, using two slot of arrival information, as in RV1(2), may result in a significant improvement of RV1.

The RV policies evaluate decisions as if future decision follow from FC, instead of from an RV policy. Hence considering too many slots arrival information might result in decision that are good for the future arrivals, but that may be sub-optimal for the cars currently waiting. Since decisions can be revised every decision epoch, looking too many slots ahead may result in bad decisions.

For the F12C4 case with workload 0.6, we test the extended RV policies for varying number of slots ($M(f) = m$) of arrival information: m = 0–15. The resulting overall average waiting time is plotted in Figure 7.5. The fixed cycle on which the RV policies are based has cycle length 20 slots and each effective green period last 4 slots.

As expected, taking only 2 slots of arrival information gives already a significant reduction in the overall average waiting time. Considering a third slot of arrival information is beneficial for the RV1 policies (RV1(m) and RV1M(m)), but not under the RV2 policies (RV2(m) and RV2M(m)). Apparently, under an RV2 policy with exactly three slots of arrival information the decision (τ_1, τ_2) is too much dominated by τ_2. Using four (or more) slots of arrival information gives much better results.

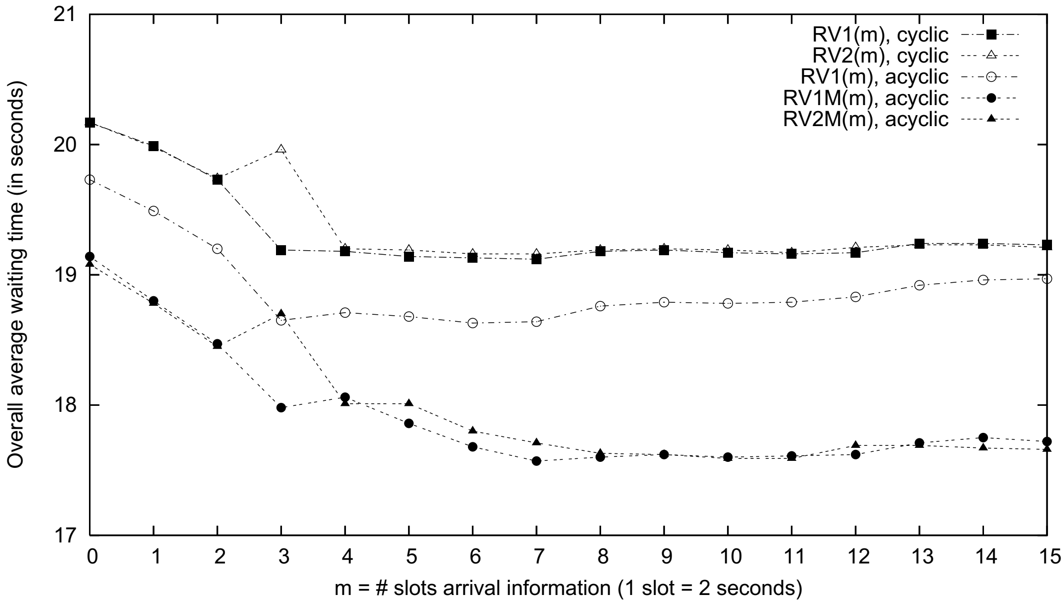

Figure 7.5: Impact of the amount of arrival information on long-run average waiting time.

With three slots of arrival information, the queue state after three slots is known exactly. Changing from green to yellow means that the exact queue length is known upon the jump to τ_2. Keeping the lights green for at one more slot, implies that the state is not exactly known when a next state τ_2 is to be selected. The choice of τ_1 determines, whether the queue state is known or upon the selection of τ_2. The uncertainty in the queue state may affect the choice of τ_1. This uncertainty is not present when four slots of arrival information is used. This explains the peak of the curves of RV2(m) and RV2M(m), at $m = 3$.

For most RV policies taking 4 or 5 slots of arrival arrival information gives already the best results. Considering more slots yields hardly any improvement, except for the RV1M and RV2M policies that benefit slightly from having a few more slots arrival information. Since the RV2(m) policies hardly improve the RV1(m) policies, we do not report on them in the remainder of this chapter.

7.5.2 Skewness in amount of arrival information

Thus far it was assumed that all flows have an equal number of slots of arrival information. In this and the next section cases are studied where each flow has either 5 slots of arrival information or has no arrival information at all.

Consider the party-asymmetric F12C4 case with arrival rates equal to $\lambda = 0.15$. For all flows, 5 slots of arrival information is available, except for C_2, which has no arrival information at all. The number of slots of arrival information is included in the name of a policy: e.g. under RV1(5,0,5,5) uses 5 slots of arrival information but not for C_2.

For being a thin combination with a low arrival rate, one may expect that the waiting time at C_2 is highest. Since arrival information is available for the two thick combination, we expect that the waiting time at the thick combinations is lowest. In Table 7.3 the waiting times at all queues are reported for both the cyclic and the acyclic RV policies. As a benchmark we report also FC and the RV policies that do not use any arrival information. For cyclic control RV1(5) is preferred above RV1. The best policy is RV1M(5), which improves RV1 by 10.1%. Taking 5 slots of arrival information into account and allow acyclic control seems to help reducing the long-run average waiting time.

Under all policies the average waiting time is indeed lowest at the thick combinations. The waiting time at the thin combination C_2, which has no arrival information, is under RV1(5) hardly higher than under RV1, which ignores any arrival information for all flows. All other flows benefit under RV1(5) from using arrival information. Seemingly, a combination suffers more from having a low number of flows, and consequently a low weight, than from lacking arrival information.

Table 7.3: Mean waiting times (in sec.) for partly-asymmetric F12C4 ($\rho = 0.6$ and $\lambda = 0.15$): impact of no arrival information for C_2.

Rule	$E(W)$ overall		EW C_1,C_3	EW C_2	EW C_4
Cyclic policies:					
RV1	20.2		18.5	23.5	23.5
RV1(5,0,5,5)	19.3	-4.1%	17.5	23.6	22.4
FC dep. times 8, 8, 8, 8 sec.	24.9	+23.6%	24.8	24.9	25.0
Acyclic policies:					
RV1	19.7	-2.2%	17.5	24.2	24.3
RV1M	19.1	-5.1%	16.5	24.4	24.4
RV1(5,0,5,5)	18.9	-6.4%	16.3	24.2	23.5
RV1M(5,0,5,5)	18.1	-10.3%	15.2	24.6	23.3

7.5.3 Skewness in arrival rates and arrival information

When arrival information is lacking for a very thin combination, i.e. a combination with a relatively low load, then cars arriving at that combination may experience high waiting times. This hypothesis did not hold for the case, studied in the previous subsection, where all arrival rates and all effective green times were identical. Now we check this hypothesis, for the following two fully-asymmetric F12C4 cases with workload 0.6:

Case I: the arrival rate at C_2 is only a third of that at the other queues: hence the thick combinations are six times as thick. Furthermore all flows except those in C_2 benefit from 5 slots arrival information.

Case II: the arrival rates at the North and South approach are a third of those at the West and East approach. Furthermore the West and East approach give 5 slots of arrival information, but the North and South approaches give no arrival information at all.

Case I – Combination 2 is third as busy and provides no arrival information

In this case, the arrival rates of flows in $\mathcal{C}_2$ are only 3/50, whereas for the other flows the arrival rates are 9/50. The total load is thus 0.6 (= $(9 + 3 + 9 + 9)/50$). Under RV1(5,0,5,5), RV1(5,0,5,5) and RV1M(5,0,5,5) all flows except those in C_2 have 5 slots arrival information.

According to the Optimal-fixed-cycle algorithm, the best configuration of FC has cycle length 54 seconds (26 slots), with effective green times or departure times 14, 6, 14, and 12 seconds for combinations 1 to 4 respectively. The effective green times of C_1 and C_3 is identical since they have the same load: i.e. both combinations consist of 4 flows with identical arrival rates. C_4 has the same arrival rate but consist of only 2 flows and is thus less thick: therefore its green time is a bit lower than that of C_1 and C_3. The very thin combination, C_2, has a very short effective green time of only 6 seconds. Its red period last $54 - 6 = 48$ seconds; that is 8 times as long as the green period.

On the one hand, there are three reasons for expecting C_2 to have a higher average waiting time than the other combinations:

- C_2 consist of only 2 flows, whereas C_1 and C_3 consists of 4 flows each,
- C_2 is six times as thin as both C_1 and C_3,
- C_2 has no arrival information, whereas all other flows may benefit from 5 slots of arrival information.

On the other hand, the green period of C_2 in FC is short, which means the queue lengths are to be kept short to avoid incurring high relative cost values for flows in $\mathcal{C}_2$. Hopefully, the average waiting times for C_2 under the RV policies do not exceed those of FC.

In Table 7.4, one reads the average waiting time under the different polcies with and with no arrival information. Under cyclic control, the waiting time of C_2 has increased by only 1 second as a result of using arrival information, while the waiting time at C_4 is reduced by 2 seconds. Under acyclic control, the average waiting time at C_2 is about twice as high as under cyclic control. Clearly, C_2 suffers, in favor of a reduction in the waiting time at all other flows. The difference in the overall average waiting time between using 5 slots of arrival information or none, is also under acyclic control only 1 or 2 seconds: the waiting at 10 queues has reduced by 1 to 2 seconds, whereas the waiting time at C_2 increases by less than 1 second.

Compared to FC, the overall average waiting can be reduced by 27%: from 24.8 seconds under FC to 18.1 seconds under RV1M(5,0,5,5). For cyclic control the reduction may be 19%: from 24.8 to 20.1 seconds under cyclic RV1(5,0,5,5).

Table 7.4: Mean waiting times (in sec.) for a fully-asymmetric F12C4 case ($\rho = 0.6$ and $\lambda = (\frac{9}{50}, \frac{3}{50}, \frac{9}{50}, \frac{9}{50})$): the impact of low arrival rates and no arrival information to C_2.

Rule	*EW* overall		*EW* C_1,C_3	*EW* C_2	*EW* C_4
Cyclic policies:					
RV1	21.1		20.5	24.7	22.0
RV1(5,0,5,5)	20.1	-4.4%	19.5	25.8	20.6
FC dep. times 14, 6, 14, 12 sec.	24.8	+17.6%	22.7	29.1	31.9
Acyclic policies:					
RV1	19.2	-8.9%	17.7	43.6	18.7
RV1M	19.0	-10.0%	16.0	52.3	19.9
RV1(5,0,5,5)	18.5	-12.3%	17.0	44.4	17.5
RV1M(5,0,5,5)	18.1	-14.2%	15.3	53.1	17.7

Case II – North and South approaches (C_1 and C_2) provide no arrival information and are only a third as busy

The workload set by the North and South approaches is a third of the workload at the East and West approach. To get a total workload of 0.6 the arrival rates are set to 3/40 at flows in $\mathcal{C}_1$ and $\mathcal{C}_2$, and 9/40 at all other flows. In Table 7.5 we report the waiting times for this fully-asymmetric F12C4 case. The best FC found by the Optimize-fixed-cycle algorithm grants green or yellow to C_1, C_2, C_3, and C_4, for respectively 6, 6, 16, and 14 seconds (or 3, 3, 8, and 7 slots). The cycle length is thus 50 seconds (25 slots). Under FC the waiting time is 23.2 seconds. The overall average waiting time drops by 25% from 23.2 under FC to 17.2 seconds under RV1M(0,0,5,5).

The conclusions are quite similar to those for Case I. Due to the asymmetry in the arrival rates, flows in C_2 experience a high average waiting time under acyclic control. Since C_1 consists of twice as many flows as C_2, the difference in the average waiting time between cyclic and acyclic control is much less to C_1 than to at C_2. The lowest waiting times are experienced at C_3 and C_4 since these combinations contribute most to the workload of the intersection and because they benefit from having arrival information.

The impact of taking into account 5 slots arrival information reduces the waiting time at C_3 and C_4, while it hardly increases the waiting times at C_1 and C_2. The reduction in the overall average waiting time is moderate: about 5%.

Table 7.5: Mean waiting times (in sec.) for a fully-asymmetric F12C4 cases ($\rho = 0.6$ and $\lambda = (\frac{3}{40}, \frac{3}{40}, \frac{9}{40}, \frac{9}{40})$): impact of low arrival rates and no arrival information to C_1 and C_2.

Rule	*EW* overall		*EW* C_1	*EW* C_2	*EW* C_3	*EW* C_4
Cyclic policies:						
RV1	20.4		22.9	24.9	18.7	20.6
RV1(0,0,5,5)	19.4	-4.9%	23.2	25.3	17.4	19.1
FC dep. times 6,6, 16, 14 sec.	23.2	+13.6%	30.0	30.0	18.8	25.2
Acyclic policies:						
RV1	18.8	-7.9%	22.5	33.9	15.6	17.8
RV1M	18.1	-11.3%	24.0	36.1	13.8	16.8
RV1(0,0,5,5)	18.0	-12.0%	23.0	34.0	14.4	16.4
RV1M(0,0,5,5)	17.2	-15.7%	24.8	36.9	12.5	15.0

7.6 Conclusions

The dynamic control of traffic lights using information on the queue lengths and on estimated arrival times of near-future car arrivals, can be formulated as an MDP. Even when the arrival information is omitted, the computational complexity is too high to solve realistic cases. With the inclusion of, say, 5 slots of arrival information for each flow, even small problems, e.g. an F4C2 case, cannot be solved. For practical problems one thus relies on heuristics and approximate solutions.

When cars drive at constant speed, are not allowed to change lanes, and are queued vertically, the state space under FC is decomposable into subspaces for each flow. This decomposition is exploited in the analysis of the underlying Markov chain. In principle the relative values for states including the arrival information can be computed flow-by-flow. Instead we exploit the fact that transitions happen (approximately) in a deterministic fashion. Therefore the (infinite) horizon is split into two parts: for the first part arrival information is available, for the second part relative values can be computed to evaluate the future under FC with no arrival information. Over the first part of the horizon, the transitions under FC from one state to the next happen deterministically; thus one may compute easily the waiting costs over this period. To evaluate the second part, the relative values under FC are used, as explained in Chapter 6.

By a one-step policy improvement algorithm, one selects the best time jump in FC given the waiting cost over the period with arrival information and the relative value of the state at which one ends under FC. All of the RV policies introduced in Chapter 6 can be extended to deal with arrival information. The extension of the RV2 policies is somewhat complicated. From simulation of the F12C4 intersection in isolation, we conclude:

- Only a few slots of arrival information suffices to achieve a reduction in the overall average waiting time by 3 to 6%.
- Compared to FC the new policy reduces the average waiting time by 20 to 30%.
- When the workload is low, the reduction is greater than when the workload is high.
- The cyclic RV1 policy with arrival information shows low waiting times; even in cases where some flows have no arrival information.

The conclusions may hold specifically for the selected F12C4 infrastructure. Results for the F4C2 intersection are presented in the next chapter, in which networks of intersections are discussed.