From Bayesian Decision Makers to Bayesian Agents

Vaclav Šmídl, Jan Přikryl
Institute of Information Theory and Automation, Czech Academy of Sciences, Prague, Czech Republic

AbstractBayesian approach to decision making is successfully applied in many practical situations. Typically, however, a decision maker is considered to be the only active part of the system. Extension of the classical decision-making scenarios has been presented as multiple-participant decision-making. The resulting DM units has all features of an agent as it is understood in multi-agent systems. A distinctive feature of such a Bayesian agent is that all information is represented by probability density functions. Moreover, communication between two agents is also facilitated by probabilities. All subsequent operations can be formalized in terms of probability calculus. Majority of the necessary algorithms was already derived for a single Bayesian decision maker, however, more work is required to resolve issues related to communication and cooperation of Bayesian agents. The proposed approach is illustrated on the problem of feedback control of urban traffic networks.

Keywords. Bayesian decision making, multi-agent systems, fully probabilistic design

1. Introduction

Decision making (DM) [17,5] is an active and purposeful selection of *actions* among several alternative options. For humans, it is a natural part of everyday life. Any decision-maker—biological or artificial—never has complete knowledge of the mechanism relating the actions and their dynamically delayed consequences, hence the decisions are always made under uncertainty. Therefore, design of artificial decision-makers (controllers) is typically based on the theory of statistical DM [10,5]. The task is to design a *DM strategy*, i.e. such a sequence of rules (mapping the available knowledge onto actions) that is optimal for reaching the aims of the decision maker. Bayesian statistics provides a consistent framework for this task, and many techniques for practical design of Bayesian decision-makers are available [15].

However, practical algorithms for design of the optimal DM strategy are typically based on the following assumptions:

1. The optimized strategy is the only system that intentionally influences the optimized responses.
2. Only one DM aim is given, and it is known a priori.

These assumptions are too restrictive for certain type of problems. For example, application of the traditional approach to large systems (such as traffic control in urban ar-

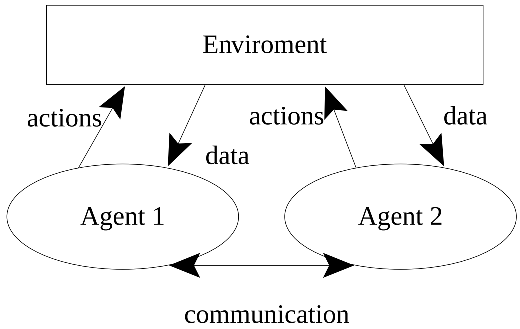

Figure 1. Multi-agent scenario.

eas) is conceptually possible, but practically intractable due to the implied computational burden. The problem is practically solved by decomposition of the whole DM problem, leading to—necessarily approximate—*distributed DM* [13,7] and multi-agent systems [25,26]. However, there is no generally accepted methodology for design of DM strategy for these agents. In this text, we address the problem by extension of the classical Bayesian DM theory for multiple-participant DM [1].

1.1. Bayesian Agents

An agent, as a general building block of multi-agent systems, can be seen as a decision maker that can influence only a part of its *environment*. Each agent interacts with its environment via (i) observations, (ii) decisions, (iii) and communication to other agents. This is illustrated in Figure 1. If the agents are not aware of each others presence, or they do not care about the others, they act *autonomously*, in the same way as classical Bayesian decision-makers. However, mutual effect of uncoordinated autonomous decision-makers is generically adverse and yields poor overall performance. This can be remedied by their mutual *communication*, which is not considered by a classical Bayesian decision-maker.

The *Bayesian agent* is an extension of a single Bayesian decision-maker—which typically has its individual aims, and pre-determined abilities to observe and act autonomously—by the ability to *communicate* its knowledge *and aims* with other agents. The last requirement is a new challenge for the Bayesian paradigm, since the communicated information must be merged with the existing structures. It can be shown that the technique of *fully probabilistic design (FPD)* of DM strategy [14] reduces the task of agent cooperation into *reporting and merging of probability density functions* [1].

In this paper, we review the basic theory of Bayesian DM in Section 2, and define the Bayesian decision-maker in Section 3. The Bayesian Agent is outlined in Section 4 and its practical use is illustrated on the problem of urban traffic control.

2. Bayesian Decision Making

Bayesian DM is based on the following principle [5]: *Incomplete knowledge and randomness have the same operational consequences for decision-making*. Therefore, all unknown quantities are treated as random variables and formulation of the problem and its solution are firmly based within the framework of probability calculus.

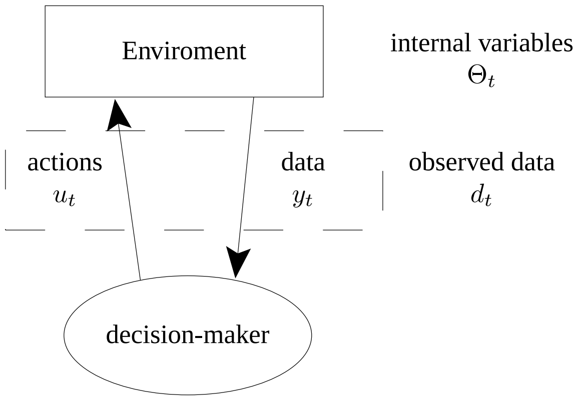

Figure 2. Basic DM scenario

This task of decision making can be decomposed into the following sub-tasks.

Model Parametrization: Each agent must have its own model of its neighbourhood, i.e. part of the environment. Uncertainty in the model is described by parametrized probability density functions.

Learning: Reduces uncertainty in the model of the neighbourhood, using the observed data. In practical terms, parameters of the model are inferred.

Strategy Design: Choose the rule for generating decisions using the parametrized model and given aims.

These tasks will be now described in detail.

2.1. Model Parametrization

The basic scenario of decision-making is illustrated in Figure 2. Here, d_t denotes all observable quantities on the environment, i.e. data, y_t, and actions, u_t, $d_t = [y_t', u_t']'$. Θ_t is an unknown parameter of the model of the environment. In Bayesian framework, the *closed loop*—i.e. the environment *and* the decision-maker—is described by the following probability density function:

$$f\left(d\left(t\right),\Theta\left(t\right)\right) = \prod_{\tau=1}^{t} f\left(y_\tau | u_\tau, d\left(\tau-1\right), \Theta_\tau\right) f\left(\Theta_\tau | u_\tau, d\left(\tau-1\right), \Theta_{\tau-1}\right) f\left(u_\tau | d\left(\tau-1\right)\right). \quad (1)$$

Here, $f\left(\cdot\right)$ denotes probability density function (pdf) of its argument. $d\left(t\right)$ denotes the observation history $d\left(t\right) = [d_1, \ldots, d_t]$. The model represents the whole trajectory of the system, including inputs u_τ which can be influenced by the decision maker. The chosen order of conditioning distinguishes the following important pdfs:

observation model $f\left(y_t | u_t, d(t-1), \Theta_t\right)$, which models dependency of the observed data on past data $d\left(t-1\right) = [d_1, \ldots, d_{t-1}]$, model parameters Θ_t and actions u_t.

internal model $f\left(\Theta_t|u_t, d(t-1), \Theta_{t-1}\right)$, which models evolution of parameters of the model via data history $d\left(t-1\right)$, previous model parameters Θ_{t-1} and the chosen actions u_t.

DM strategy $f\left(u_t|d\left(t-1\right)\right)$, is a probabilistic description of the decision rule.

2.2. Learning via Bayesian filtering

The task of learning is to infer posterior distribution of unknown parameters from the observed data, $f\left(\Theta_t|d\left(t\right)\right)$. This pdf can be computed recursively as follows:

$$f\left(\Theta_t|u_t, d\left(t-1\right)\right) = \int f\left(\Theta_t|u_t, d\left(t-1\right), \Theta_{t-1}\right) f\left(\Theta_{t-1}|d\left(t-1\right)\right)\, d\Theta_{t-1}, \tag{2}$$

$$f\left(\Theta_t|d\left(t\right)\right) \propto \frac{f\left(y_t|u_t, d\left(t-1\right), \Theta_t\right) f\left(\Theta_t|u_t, d\left(t-1\right)\right)}{f\left(y_t|u_t, d\left(t-1\right)\right)}, \tag{3}$$

$$f\left(y_t|u_t, d\left(t-1\right)\right) = \int f\left(y_t|u_t, d\left(t-1\right), \Theta_t\right) f\left(\Theta_t|u_t, d\left(t-1\right)\right) d\Theta_t. \tag{4}$$

In general, evaluation of the above pdfs is a complicated task, which is often intractable and many approximate techniques must be used [9]. In this text, we are concerned with conceptual issues and we assume that all operation (2)–(4) are tractable.

2.3. Design of DM strategy

In this Section, we review *fully probabilistic design (FPD)* of the DM strategy [14]. This approach is an alternative to the standard stochastic control design, which is formulated as minimization of an expected loss function with respect to decision making strategies [2,6]. The FPD starts with specification of the decision-making aim in the form of *ideal pdf* of the closed loop. This ideal pdf—which is the key object of this approach—is constructed in the same form as (1) distinguished by superscript ${}^{\lfloor I}$:

$$f\left(d\left(t\right), \Theta\left(t\right)\right) \rightarrow {}^{\lfloor I}f\left(d\left(t\right), \Theta\left(t\right)\right). \tag{5}$$

Similarly to (1), the ideal distribution is decomposed into ideal observation model, internal model, and ideal DM strategy. Recall, from Section 2.1, that model (1) contains the DM strategy, which can be freely chosen. Therefore, the optimal DM strategy can be found by functional optimization of the following loss function

$$\mathcal{L}\left(f\left(u_t|d\left(t-1\right)\right), \mathring{t}\right) = \mathsf{KL}\left(f\left(d\left(\mathring{t}\right), \Theta\left(\mathring{t}\right)\right) || {}^{\lfloor I}f\left(d\left(\mathring{t}\right), \Theta\left(\mathring{t}\right)\right)\right),$$

where $\mathsf{KL}\left(\cdot,\cdot\right)$ denotes the Kullback-Leibler divergence between the current (learnt) and the ideal pdf [20], and $\mathring{t} > t$ is the decision-making horizon.

The approach has the following special features.

- The KL divergence to an ideal pdf forms a special type of loss function that can be simply tailored both to deterministic and stochastic features of the considered DM problem.

- Minimum of the KL divergence—i.e. the optimal DM strategy—is found in *closed form*:

$$f\left(u_t|d\left(t-1\right)\right) = {}^{\lfloor I}f(u_t|d(t-1))\frac{\exp[-\omega(u_t, d(t-1))]}{\gamma(d(t-1))}, \tag{6}$$

  where $\omega\left(\cdot\right)$ and $\gamma\left(\cdot\right)$ are integral functions of all involved pdfs (these are not presented for brevity, see [15] for details). The decisions are then generated using a simplified version of stochastic dynamic programming [4].
- Multiple-objective decision-making can be easily achieved using multi-modal ideal distributions [**?**,12].

3. Bayesian Decision Maker

In practise, the task of adaptive decision making is typically solved in two stages [15]: (i) off-line, and (ii) on-line. The off-line stage is dedicated to design of the structure and fixed parameters (such as initial conditions) of the decision-maker. When the structure and fixed parameters are determined, the decision-maker operates autonomously in on-line mode, where it is able to adapt (by adjusting model parameters) to changes in the environment and improve its DM-strategy. Operation needed in both stages are described in this Section.

3.1. Off-line stage

In this stage, it is necessary to determine structure of the model (1) and prior distribution of model parameters. These tasks are solved using archives of the observed data as follows.

Model selection: if there is no physically justified model of the environment, this technique test many possible parameterization of the model, and selects one, which is best suited for the observed data. Typically, only a class of models that yields computationally tractable algorithms is examined. A key requirement of tractability is, that the learning operation (2) can be reduced into algebraic operations on finite-dimensional *sufficient statistics*.

Elicitation of prior distributions: The expert knowledge which is not available in the form of pdfs must be converted (often approximately) into probabilistic terms. Moreover, if the available knowledge is not compatible with the chosen model, a suitable approximation (projection into the chosen class) must be found. If there is more sources of prior information are available, these must be merged into a single pdf. This operation will be described in detail at the end of this Section.

Design of DM strategy: When the model and ideal distributions are chosen, the optimal DM strategy is given in closed form by the FPD (Section 2). In special cases, the equation (6) can be parametrized by a finite-dimensional parameters, and the implied dynamic programing is reduced into algebraic operations on these parameters.

These tasks are computationally demanding and thus they are traditionally solved off-line, i.e. only once for all available data. This is acceptable, since all expert information is available a priori, and model of the environment is assumed to be constant.

3.2. On-line stage

A typical adaptive decision maker operates by recursive repetition of the following steps:

1. **read**: the observed data are read from the environment. All the necessary pre-processing and transformation of data is done in this step.
2. **learn**: the observed data are used to increase the knowledge about the environment, namely the sufficient statistics of the model parameters are updated.
3. **adapt**: the decision-maker use the improved knowledge of the system to improve its DM strategy. Specifically, parameters of the DM strategy are re-evaluated using the new sufficient statistics.
4. **decide**: the adapted DM strategy is used to choose an appropriate action. Recall, that the DM-strategy is a pdf. Hence, a realization from this pdf must be selected. Typically, the optimal decision is chosen as expected value of (6).
5. **write**: the chosen action is written into the environment. Similar to the first step, transformation of the results is done in this step.

Note that due to computational constraints, all operations in this stage are defined on finite dimensional parameters or statistics.

3.3. Merging of pdfs

For the task of prior elicitation, we need to define a new probabilistic operation for merging of information from many sources. The merging operation is defined as a mapping of two pdfs into one:

$$f_1\left(\Theta_t|d\left(t\right)\right), f_2\left(\Theta_t|d\left(t\right)\right) \stackrel{\text{merge}}{\longrightarrow} \tilde{f}\left(\Theta_t|d\left(t\right)\right), \tag{7}$$

where f_1 and f_2 are the *source pdfs*, and the $\tilde{f}$ is the *merged pdf*. The aim of the merging operation is to preserve within one pdf, $\tilde{f}$, as much information from the sources, f_1 and f_2, as possible.

Note that the source pdfs in (7) are defined on the same variable as the merged pdf, hence (7) will be known as *direct merging*. Alternatively, the sources can be defined on the variable in condition of the merged pdf,

$$f_1\left(d_t|d\left(t-1\right)\right), f_2\left(d_t|d\left(t\right)\right) \stackrel{\text{merge}}{\longrightarrow} \tilde{f}\left(\Theta_t|d\left(t\right)\right), \tag{8}$$

in which case, the mapping is known as *indirect merging*. The problem is discussed in detail in [**?**], we shortly outline the principle of its solution in the sequel.

In direct merging, the merged pdf $\tilde{f}\left(d\right)$ is selected so that a weighted sum of Kullback-Leibler divergences between the source pdfs, f_1 and f_2, and the merged pdf, $\tilde{f}$, is minimized:

$$\tilde{f} = \arg\min_f \left(\alpha \mathsf{KL}\left(f_2||f\right) + \left(1-\alpha\right)\mathsf{KL}\left(f_1||f\right)\right). \tag{9}$$

The weight $\alpha \in \langle 0, 1\rangle$ governs the level of importance of each source. The optimum of (9) for merging of distributions of the *same* variable, is found in the form of a probabilistic mixture of the source pdfs:

$$\tilde{f}(d) = \alpha f_2(d) + (1-\alpha) f_1(d). \tag{10}$$

Note that the result is significantly dependent on the weight α, which is an important tuning knob in the operation.

4. Bayesian Agents

The Bayesian agent is an extended Bayesian decision maker described in previous Section. The additional features are the ability and need of agents to communicate and cooperate. In the Bayesian framework, all knowledge is stored in pdfs. The challenge is to formalize communication and cooperation within the framework of probability calculus. In this Section, we propose a simple probabilistic model of negotiation. For clarity of explanation, we consider only two agents, $A_{[1]}$ and $A_{[2]}$, where agent number is always in subscript in square brackets.

Each agent has the following quantities

Observed data d_t: Naturally, each agent can observe different subset of variables, i.e. $d_{t,[1]}$ and $d_{t,[2]}$, for A_1 and A_2, respectively. The agents can exchange knowledge only in terms of variables that are common for both of them, i.e. $d_{t,[1\cap 2]}$. Any communication is meaningful only with respect to this subset.

Internal quantities Θ_t: We do not impose any structure of the environment model for the agents, hence, internal quantities $\Theta_{t,[1]}$ and $\Theta_{t,[2]}$ are in general disjoint sets.

Environment Model: $f_{[1]} = f\left(d_{[1]}(t), \Theta_{[1]}(t)\right)$ and $f_{[2]} = f\left(d_{[2]}(t), \Theta_{[2]}(t)\right)$ for each agent.

Ideal distributions: ${}^{\lfloor I}f_{[1]} = {}^{\lfloor I}f\left(d_{[1]}(t), \Theta_{[1]}(t)\right)$ and ${}^{\lfloor I}f_{[2]} = {}^{\lfloor I}f\left(d_{[2]}(t), \Theta_{[2]}(t)\right)$ for each agent.

Negotiation weights: For the purpose of negotiation, we define a scalar quantity $\alpha_{2,[1]} \in \langle 0, 1\rangle$ denoting the level of belief of agent A_1 in information received from A_2. Analogically, $\alpha_{1,[2]}$ is defined in A_2.

4.1. *Communication*

The agents can communicate two kinds of information: (i) about the environment, and (ii) about their individual aims. In both cases, the information is stored in the form of pdfs, namely marginal distribution from the environment model for (i), and marginal distribution on ideal pdfs for (ii).

The model of the environment (1) is fully determined by the observation model (Section 2.1) and parameter distribution (3), which is estimated from the observed data $d(t)$. The easiest way how to exchange the information about the environment is to exchange the observed data. The observed data can be seen as a special case of pdf, namely empirical pdf $f\left(d_{[2]}(t)\right)$. Then, the task is formally identical with the task of indirect merging of pdf (8) as described in Section 3.3. The observed data from A_2 are merged with the existing model of A_1 using

$$f_{[1]}, f\left(d_{[2]}(t)\right) \stackrel{\text{merge}}{\longrightarrow} \tilde{f}_{[1]}.$$

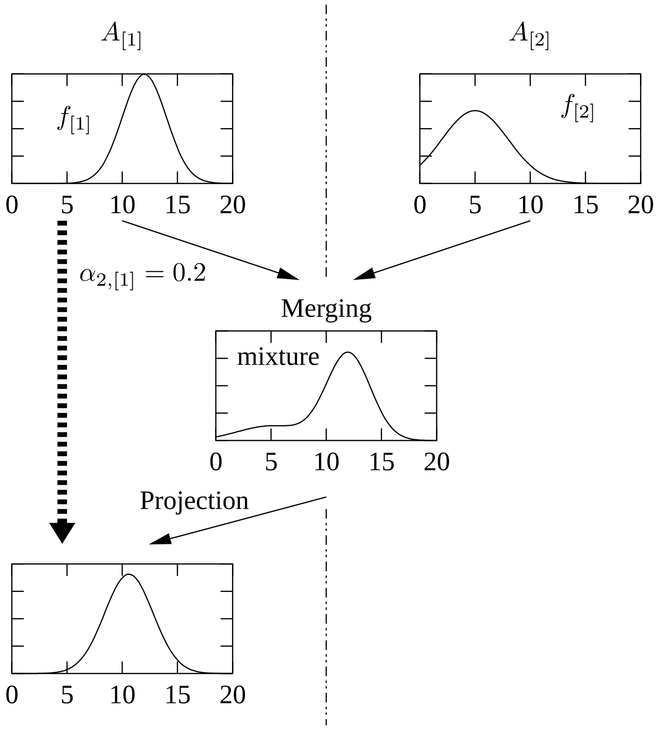

Figure 3. Illustration of merging of two Gaussian distribution. The merged distribution is a mixture of Gaussians for which the operations of learning and design of DM strategy do not have a finite-dimensional parametrization. Thus, the merged distribution is projected into the class of single Gaussians.

and the negotiation weight $\alpha_{2,[1]}$. This weight can be chosen constant or it can be negotiated with the neighbour. When the negotiation is finished, the merged pdf $\tilde{f}_{[1]}$ is then used as the new model of the environment.

The ideal distributions can be communicated and merged in the same way, using direct merging (7). Note that merging of the ideal distributions influences the aim of the agent. The FPD procedure must be performed after each merge in order to recompute the DM strategy. Once again, the result is strongly influenced by the negotiation weights α. These weights can be determined by negotiation strategies.

If the merging operation yields pdfs that are not compatible with the observation model (i.e. can not be reduced into algebraic form), the merged distribution must be projected into the compatible class, as illustrated in Figure 3.

4.2. Negotiation strategies

We distinguish three basic strategies [16]:

- **selfish** — a strategy where each agent freely chooses its own weights. Agent A_1 accepts all information from its neighbour, but it refuse any attempts to change the weight $\alpha_{2,[1]}$ that may be suggested by A_2.

- **hierarchical** — a strategy where the agent have a fixed values of $\alpha_{2,[1]}$, however if the neighbour A_2 is superordinate to A_1, it can assign the value of $\alpha_{2,[1]}$ by communication.
- **cooperative** — a strategy, where both participants have common aim (given by the user using ideal pdfs) to reach an agreement on the negotiation values, i.e. $\alpha_{2,[1]} = \alpha_{1,[2]}$.

4.3. On-line algorithm of Bayesian agents

On-line operation of each Bayesian agent is an extension of the on-line steps of Bayesian decision-maker (Section 3).

1. **read**: the observed data are read from the system (environment). Possible communication (via pdf) from the neighbour is also received in this step. We assume that only one neighbour can communicate in one time step.
2. **learn**: the observed data are used to increase the knowledge about the system (environment).

2a. **merge**: if the communication from the neighbour contains information about the environment, the merge operation is called in order to merge it with the current knowledge. In case of communication of ideal distributions, the FPD procedure is run. Note that this step may be computationally expensive.

3. **adapt**: the decision-maker use the improved knowledge of the environment to improve its DM strategy.
4. **decide**: the adapted DM strategy is used to choose an appropriate action. In multi-agent scenario, the tasks of communication and negotiation are also part of the decision making process. Therefore, in this step, decisions on communication (request communication, negotiate, refuse communication) and negotiation (propose new value of $\alpha_{1,[2]}$) must be also made.
5. **write**: the chosen action is written into the system (environment). If the decision to communicate was made, a message to the neighbour is also written in this step.

Note that acquisition of the observed data is synchronized with communication. In each time step, only one message from the neighbour is received, processed and answered. This allows seamless merging of knowledge from direct observations and from communication. If the periods of data sampling and communication differ, the smaller one is chosen as the period of one step of an agent.

5. Application in Traffic Control

Urban surface transport networks are characterised by high density of streets and a large amount of junctions. In many cities these structures cannot easily accommodate the vast volume of traffic, which results in regular traffic congestions. Efficient traffic control mechanisms are sought that would provide for higher throughput of the urban transport network without changing its topology.

Due to space constraints we cannot present the reader with full introduction to the principles of urban traffic control (UTC). We will just briefly outline terms that will be needed below. More thorough explanation of the UTC methodology exceeds the scope of this paper. Interested readers should refer to any of the existing monographs on this topic, e.g. [21,24].

In most cases, UTC is targeted on *signalled intersections*, where traffic is controlled by *traffic signals*. The sequence of traffic signal settings for an intersection is called a *signal plan*. A signal plan cycle typically consist of several *stages*, where one of the conflicting traffic flows has green and the others have to wait. The lengths of stages, the overall signal plan duration and other parameters are bounded by values reflecting either physical shape of the intersection or other (usually normatively given) rules. An *intersection controller* is an industrial micro-controller that attempts to select the order of stages and to modify stage lengths in such a way that a maximum possible throughput of the intersection is achieved. The ordering of stages may be influenced by public transport vehicles in order to minimise their waiting at an intersection.

Intersection controllers are very often autonomous devices that do not react on traffic conditions at neighbouring intersections. However, in areas with high traffic intensity, intersection controllers may be mutually interconnected in a kind of hierarchical controller that attempts to optimize the throughput of the whole traffic network. Several interesting UTC approaches exist that attempt to solve the traffic control problem using feedback from different traffic detectors [18]. Many agent-based approaches have been implemented as well. For example: (i) agents for setting of the optimal signal plan cycle length [11], and (ii) for distributed coordination of signal plans. The latter are based on game theory [3,8] or Bayesian learning [22]. These applications often use approximate heuristics and long-time statistics to derive the optimal control strategy. Our proposal is to build the strategy adaptively in a collaborative agent-based environment.

In the following text, agents are intersection controllers of some street network. The agents shall agree on the overall traffic signal setting that would minimise time spent by vehicles inside the controlled region and thus maximise the throughput of the network.

5.1. Model

Papageorgiou [23] shows that the total time spent by a vehicle in a controlled micro-region is strongly correlated to queue lengths at signalised intersections of this micro-region. Hence, minimization of waiting queues results in faster vehicle transition and in higher throughput of the network.

We start with a deterministic model describing the behaviour of the traffic at an intersection as a particle flow system [19]:

$$\Theta_t = A\Theta_{t-1} + Bu_{t-1} + F$$
$$y_t = C\Theta_t + G$$

where

$$\Theta_t = \begin{bmatrix} \xi_t \\ O_t \end{bmatrix}$$

is a state vector holding information about waiting queue lengths ξ_t and detected input lane occupancies O_t, u_t is an input variable which represents green settings for a signal plan cycle at this intersection. Matrix A defines transition from an old state to the new one. It is composed from information about waiting queue development, and mutual in-

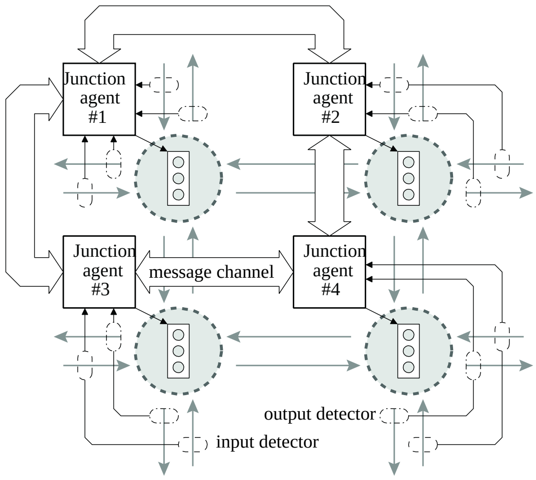

Figure 4. Simple urban traffic network with four controlled junctions and four agents.

fluence of queues at one lane on other lanes. Matrix B models throughput of the junction, and vector F is composed from the observed incoming traffic intensity. Output vector

$$y_t = \begin{bmatrix} \eta_t \\ O'_t \end{bmatrix}$$

contains information about outgoing traffic intensity η_t and output arm occupancies O'_t. C is a matrix of coefficients transforming waiting queue data into outgoing traffic intensities and vector G models the influence of current incoming traffic and past queues on outgoing traffic.

This model can be transcribed into the following probabilistic internal and observation models:

$$f(\Theta_t|\Theta_{t-1}, u_{t-1}) = \mathcal{N}\left(A\Theta_{t-1} + Bu_{t-1} + F, Q\right) \tag{11}$$

$$f(y_t|\Theta_t) = \mathcal{N}\left(C\Theta_t + G, R\right) \tag{12}$$

where $\mathcal{N}(\mu, \sigma)$ is a Gaussian probability distribution and Q and R are allowed variances. The internal model (11) describes the probability distribution of queue lengths at an intersection at time t given the green settings u_{t-1} and incoming traffic data Θ_{t-1} at time $t-1$. The observation model (12) yields probability distribution of outgoing traffic intensity of the modelled junction at time t, given the queue pdf from the internal model.

5.2. *Ideal distributions*

The global aim of the proposed UTC approach is to minimise waiting queues at every junction. As said in Section 4, agents attempt to reach this aim by exchanging their ideal pdfs, defined on their common data. In our case, agents share information about traffic intensity at particular intersection arms. Hence, the exchanged ideal pdfs specify wishes about intensity of outgoing and incoming traffic.

We propose to model the ideal pdfs as follows:

$$ {}^{\lfloor I}f\left(\xi_{t}\right) = t\mathcal{N}\left(0, V_{\xi}, \langle 0, \xi_{\max}\rangle\right), \tag{13} $$

$$ {}^{\lfloor I}f(I_{t}|\xi_{t}) = t\mathcal{N}\left(I\left(\xi_{t}\right), V_{I}, \langle 0, I_{\max}\rangle\right), \tag{14} $$

$$ {}^{\lfloor I}f(\eta_{t}|\xi_{t}) = t\mathcal{N}\left(\eta_{\max}, V_{\eta}, \langle 0, \eta_{\max}\rangle\right). \tag{15} $$

Here, $t\mathcal{N}\left(0, U, \langle 0, \xi_{\max}\rangle\right)$ denotes a Gaussian distribution with mean value 0 and variance V_{ξ}, truncated on the interval $\langle 0, \xi_{\max}\rangle$. $\xi_{\max}$ denotes maximal allowed queue length. The ideal (13) favours minimal queue lengths value, since estimates $f\left(\xi_{t}\right)$ with lower mean value are closer to the ideal that those with larger mean value. The ideal pdf (14) models the agents wishes for input intensities coming from its neighbours. The requested mean value $I\left(\xi_{t}\right)$ is changing with the current traffic conditions. The variance V_{I} expresses the "strength" of the request; higher V_{I} allows higher deviation from ideal $I\left(\cdot\right)$ and leaves the agent more room for adapting to other requests. $I_{\max}$is the maximum possible intensity at the arm (or lane) in concern. In order to communicate these wishes to the neighbours, they must be independent of the internal quantity ξ_{t}. This can be achieved by marginalization, i.e. ${}^{\lfloor I}f(I_{t}|\xi_{t}) \rightarrow {}^{\lfloor I}f(I_{t})$.

Note that output intensities of one intersection are input intensities of its neighbours, i.e. $\eta_{t,[1]} \rightarrow I_{t,[2]}$. Hence, the communicated ideals on input intensities $f\left(I_{t,[2]}\right)$ will be merged with ideals on output intensities $f\left(\eta_{t,[1]}\right)$.

5.3. Control cycle

The proposed control cycle of a single agent follows the decomposition from Section 4.3:

1. **read:** The agent reads observed data from the environment and checks for incoming communication from some neighbour.
2. **learn:** Observed data of the agent (measured traffic intensities) are used to increase its knowledge about current traffic conditions, namely pdfs of waiting queue lengths and unobserved intensities of traffic flow.
3. **merge:** If a message from some neighbour arrived, its pdf is merged with the agent's pdfs — either with the current knowledge or with ideal pdfs. In the latter case, FPD procedure that evaluates Eq. (6) is called after the merge in order to reflect the change in ideal aims in the optimal DM strategy.
4. **adapt:** The agent uses the updated knowledge to adapts its DM strategy. Hence, the strategy can be changes in reaction to the changed traffic conditions or in reaction to the message from the neighbour.
5. **decide:** Based on the adapted strategy, the agent decides about its signal plan parameters for the next period. The signal values are typically taken as expected values of the adapted strategy pdf. Decisions whether and what to communicate with agent's neighbour is also made in this moment.
6. **write:** The chosen signal plan is written to the intersection controller. Optionally, communication message is sent to the chosen neighbour.

6. Conclusion

The Bayesian methodology provides a consistent theory of decision making under uncertainty. We have presented an extension of this methodology in the area of multi-agent systems. Since the Bayesian approach formalizes all available knowledge in the form of probability density functions, we had to formalize the key features of agents—i.e. communication and negotiation—using probability calculus. We have shown that the formalization can be achieved using techniques of fully probabilistic design and merging of pdfs.

The work presented in this paper is a conceptual outline of the approach. In spite of the fact that the key techniques are available, many practical issues must be solved before it is ready for real application. The presented application in urban traffic control will be used as testing environment for further research and development of Bayesian agents.

References

[1] J. Andrýsek, M. Kárný, and J.. Kracík, editors. *Multiple Participant Decision Making*, Adelaide, May 2004. Advanced Knowledge International.

[2] K.J Astrom. *Introduction to Stochastic Control*. Academic Press, New York, 1970.

[3] Ana L. C. Bazzan. A distributed approach for coordination of traffic signal agents. *Autonomous Agents and Multi-Agent Systems*, 10(1):131–164, @jan 2005.

[4] R. Bellman. *Introduction to the Mathematical Theory of Control Processes*. Academic Press, New York, 1967.

[5] J.O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, New York, 1985.

[6] D.P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Nashua, US, 2001. 2nd edition.

[7] R. Caballero, T. Gomez, M. Luque, F. Miguel, and F. Ruiz. Hierarchical generation of pareto optimal solutions in large-scale multiobjective systems. *Computers and operations research*, 29(11):1537–1558, 2002.

[8] Eduardo Camponogara and Werner Kraus Jr. Distributed learning agents in urban traffic control. In Fernando Moura Pires and Salvador Abreu, editors, *Progress in Artificial Intelligence: Proceedings of the 11th Portuguese Conference on Artificial Intelligence (EPIA 2003)*, volume 2902 of *Lecture Notes in Computer Science*, pages 324–335, Beja, Portugal, December 2003. Springer-Verlag.

[9] Z. Chen. Bayesian filtering: From Kalman filters to particle filters, and beyond. Technical report, Adaptive Syst. Lab., McMaster University, Hamilton, ON, Canada, 2003.

[10] M.H. DeGroot. *Optimal Statistical Decisions*. McGraw-Hill, New York, 1970.

[11] L. A. García and F. Toledo. An agent for providing the optimum cycle length value in urban traffic areas constrained by soft temporal deadlines. In L. Monostori, J. Váncza, and M. Ali, editors, *Engineering of Intelligent Systems: 14th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2001*, volume 2070 of *Lecture Notes in Computer Science*, pages 592–601, Budapest, Hungary, June 2001. Springer-Verlag.

[12] T.V. Guy, J. Böhm, and M. Kárný. Multiobjective probabilistic mixture control. In IFAC, editor, *IFAC World Congress, Preprints*. IFAC, Prague, 2005. accepted.

[13] Y.Y. Haimes and D. Li. Hierarchical multiobjective analysis for large scale systems: Review and current status. *Automatica*, 24(1):53–69, 1988.

[14] M. Kárný. Towards fully probabilistic control design. *Automatica*, 32(12):1719–1722, 1996.

[15] M. Kárný, J. Böhm, T.V. Guy, L. Jirsa, I. Nagy, P. Nedoma, and L. Tesař. *Optimized Bayesian Dynamic Advising: Theory and Algorithms*. Springer, London, 2005. ISBN 1-85233-928-4, pp. 552.

[16] M. Kárný and T.V. Guy. On dynamic decision-making scenarios with multiple participants. In J. Andrýsek, M. Kárný, and J. Kracík, editors, *Multiple Participant Decision Making*, pages 17–28, Adelaide, May 2004. Advanced Knowledge International.

[17] R.L. Keeny and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Trade-offs*. J. Wiley, New York, 1978.

[18] J. Kratochvílová and I. Nagy. Bibliographic Search for Optimization Methods of Signal Traffic Control. Technical Report 2081, ÚTIA AV ČR, Praha, 2003.

[19] J. Kratochvílová and I. Nagy. Traffic model of a microregion. In IFAC, editor, *IFAC World Congress, Preprints*. IFAC, Prague, 2005. submitted.

[20] S. Kullback and R. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–87, 1951.

[21] Michael Meyer and Eric J. Miller. *Urban Transportation Planning*. McGraw-Hill, 2 edition, December 2000.

[22] Haitao Ou, Weidong Zhang, Wenjing Zhang, and Xiaoming Xu. Urban traffic multi-agent system based on rmm and bayesian learning. In *Proceedings of the American Control Conference*, Chicago, Illinois, June 2000.

[23] M. Papageorgiou. *Applications of Automatic Control Concepts to Traffic Flow Modeling and Control*, volume 50 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, Berlin, 1983.

[24] Roger P. Roess, Elena S. Prassas, and William R. McShane. *Traffic Engineering*. Prentice Hall, 3 edition, October 2003.

[25] T. Sandholm. Distributed rational decision making. In G. Weiss, editor, *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence*, pages 201–258. 1999.

[26] Gerhard Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, 2000.