Use of a Relational Reinforcement Learning Algorithm to Generate Dynamic Activity-Travel Patterns

Marlies Vanhulsel
Davy Janssens
Geert Wets[1]

Hasselt University - Campus Diepenbeek
Transportation Research Institute
Wetenschapspark 5, bus 6
BE - 3590 Diepenbeek
Belgium
Tel: +32(0)11 26 {9133; 9128; 9158}
Fax: +32(0)11 26 91 99
E-mail: {marlies.vanhulsel;davy.janssens; geert.wets}@uhasselt.be

[1] Corresponding author

Abstract

In the course of the past decade activity-based models have entered the area of transportation modelling. Such models simulate the generation of individual activity-travel patterns while deciding simultaneously on the different dimensions of activity-travel behaviour, such as the type of activity, the activity location, the transport mode used to reach this location, the starting time and duration of the activity, etc. However, as real-world activity-travel patterns prove not to be static due to short-term adaptation and long-term learning, the scheduling algorithm needs to be adapted in order to be able to account for these dynamics. Short-term adaptation refers to within-day rescheduling as the result of the occurrence of unexpected events in the course of the execution of individual planned activity programmes, for instance congestion, or unexpected changes in the duration of an activity. Long-term learning denotes the change in activity-travel behaviour caused by the occurrence of key events, such as residential relocation and obtaining one's driving license.

In order to capture these dynamics, the current research will implement a technique originating from the area of artificial intelligence, in particular on a reinforcement learning technique extended with inductive learning. This method will be based on Q-learning in which the estimation of the traditional Q-function has been substituted by inductive learning. The approach aims at generalizing the (state, action, Q-value)-triplet by the induction of a regression tree. The major advantage of this technique consists of the fact that the Q-function no longer needs to be represented by means of a reward table which grows exponentially as the number of (state, action)-pairs rises due to both an increase in the number of exploratory and/or explanatory variables and an increase in the granularity of those variables.

This technique will be examined for its applicability to the current research area. To start with, the key component of the reinforcement learning algorithm, the reward function, will be elaborated. This function will be founded on the starting time and weekday, the activity type, the activity duration, the waiting time and the activity history in order to reflect the underlying needs of the agents. Subsequently the parameters of the algorithm will be tuned. Afterwards the improved reinforcement learning technique will be implemented to simulate observed activity sequences of sixteen full-time working individuals being part of a four-headed household.

The results will be explored, showing that the agent is able to determine autonomously activity sequences and to take into account temporal constraints and limits with respect to rather fixed activities, including work and night's sleep. To end with, the generated activity schedules will be validated. The simulated activity patterns will be compared to actual, revealed patterns by means of the distance method SAM (Sequence Alignment Method). This method calculates the distance between the generated and the actual activity schedule, reflecting as such the (dis)similarity of these patterns.

1 Settings and primary research objectives

Models are the result of the human urge to organize facts and behaviour. To confirm this one only has to consider several recent advances in modelling techniques, for instance in the area of artificial intelligence and data mining. In the transportation field, modelling mainly consists of estimating the number of trips for a certain origin and destination matrix, a transportation network, and a transportation mode. Traditionally, transportation modelling was founded on trip-based modelling. However more recently, activity-based models have entered the area of transportation modelling. Such models aim at predicting simultaneously the different dimensions of activity-travel behaviour on an individual level: which activity will be perform, at which location (origin-destination), when will this activity start and for how long, which transport mode will be used to get to the desired location, who will accompany the individual during the activity, etc. Such activity-travel patterns constitute the basis of the assignment of individual routes to the transportation network. (Ettema and Timmermans, 1997; Arentze, *et al*., 2004; Arentze and Timmermans, 2005)

However, dynamic activity-based modelling is gaining importance, as activity-travel patterns are subject to both occasional (short-term) and structural (long-term) fluctuations. Short-term adaptation accounts for within-day rescheduling due to the occurrence of unexpected events during the execution of the individual planned activity program, such as the (unexpected) change in the duration of preceding activities, congestion or changes in the availability of transport modes. Long-term learning accounts for the behavioural change within activity-travel patterns caused by the experiences of previous actions, so-called key events, for instance obtaining one's driving licence, residential relocation or a change in the household composition. (Joh, *et al*., 2004; Arentze, *et al*., 2005; van der Waerden and Timmermans, 2003)

Before being able to incorporate these processes of rescheduling and learning, activity-travel patterns need to be encapsulated in a model. The present research aims at contributing to this exercise, relying on modelling techniques that originate from the research area of artificial intelligence. The approach used in this paper has proven to be an appropriate learning algorithm for micro-simulation in dynamic multi-dimensional activity-based transportation models. (Charypar *et al*, 2004; Janssens, 2005; Vanhulsel *et al*, 2007)

The next section will discuss the data founding all analyses in the current research. The third part of this paper will briefly examine the simple reinforcement learning technique and the Q-learning technique in particular. Subsequently the reinforcement learning algorithm extended with an inductive learning approach for the purposes of predicting activity-travel sequences will be described. The fifth part will investigate the results of the current extended reinforcement learning algorithm. To end with some topics for further research and some conclusions will be formulated.

2 Data

The data stem from a project entitled "An Activity-based Approach for Surveying and Modelling Travel Behaviour" executed by the Transportation Research Institute, aiming at collecting activity-travel diaries from 2,500 households across Flanders (Belgium) both by means of a paper-and-pencil survey and a GPS-based survey. The data used in this research embrace a selection of 254 individual activity-travel diaries, which have been gathered through the paper-and-pencil survey and which are linked to individual and household data. This research focuses on one specific cluster of individuals, in particular full-time working individuals being part of a four-member household. After carefully cleaning and completing of the activity diaries, sixteen individuals remained to be taken into consideration for further analysis.

The goal of the current research includes estimating weekly activity patterns. To serve this purpose the activity-travel diaries were reshaped so as to characterise the activity episodes by their starting time, activity type, activity duration - and thus the finishing time - and the day of the week. Six activity types can be distinguished: shopping and services, leisure, working, sleeping, in-home and a residual category, other. The time variables, starting time and duration, were transformed to represent time units of fifteen minutes instead of employing continuous time variables. The activity history, which represents the amount of time between two consecutive episodes of the same activity category, is also

taken into account. This activity history can also be estimated based on the frequency that a particular activity occurs in an individual's activity pattern (per week/month/year).

It should be noted that - within the research area of activity-based travel behaviour - activity location and travel mode used to reach this location are of high importance. Although these aspects are omitted in the current research, it has been proven in earlier research that these aspects can be integrated smoothly into the simulation approach used in the current research. (Vanhulsel *et al*, 2007)

Some activities, such as night's sleep and work, impose constraints on the activity schedule, creating a so-called activity skeleton in which the remaining activities need to be fitted. These activities turn out to be rather fixed in time and space (Frusti *et al*, 2002; Schwanen and Dijst, 2003; Arentze and Timmermans, 2004) and will be treated somewhat differently throughout this research. To this end, execution boundaries – defined as the time windows between which the majority of individuals perform this activity - are deduced from the available data. The following graphs show the dispersion of the starting times and durations of the work and night's sleep activities and the ensuing boundaries.

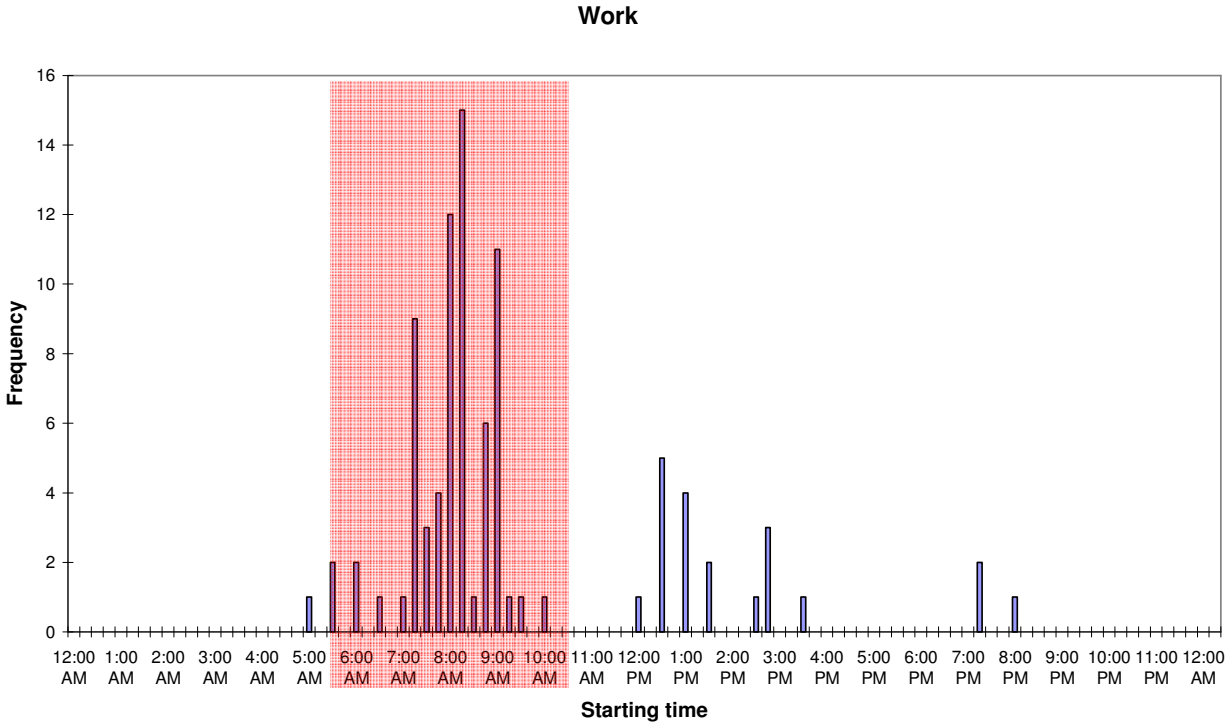

Figure 1. Work activity: dispersion of starting time

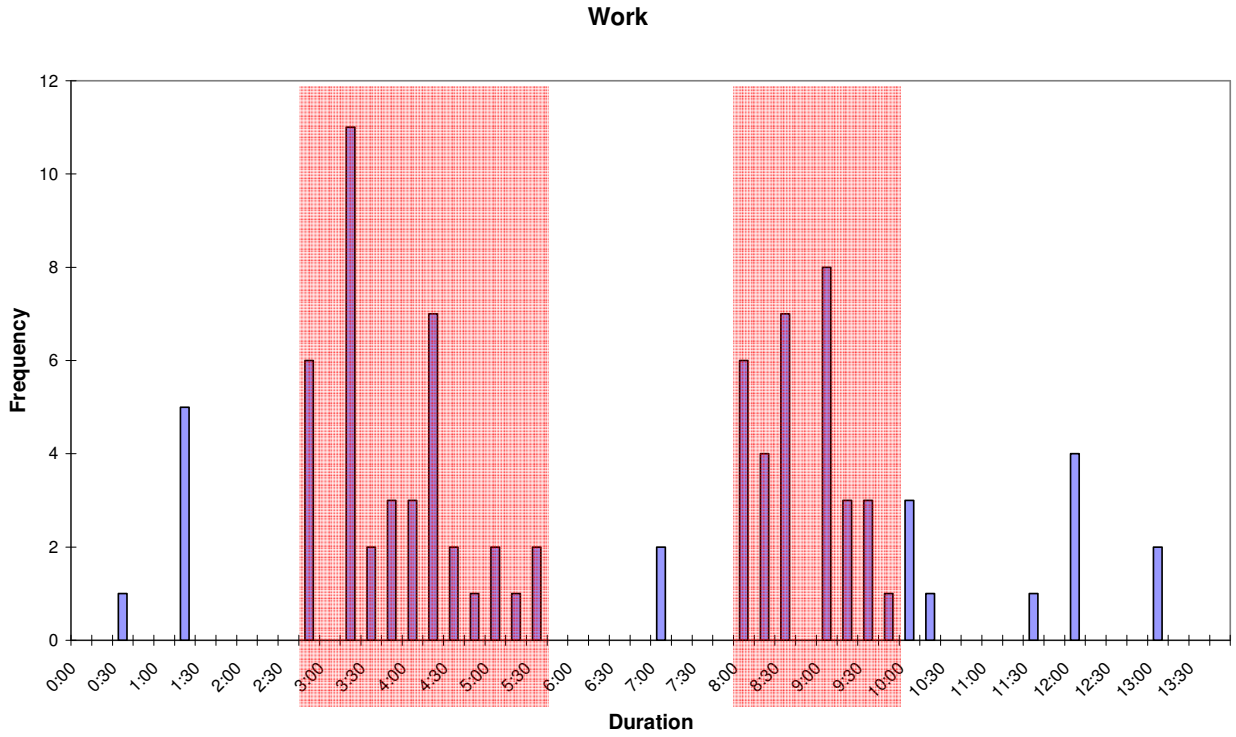

Figure 2. Work activity: dispersion of duration

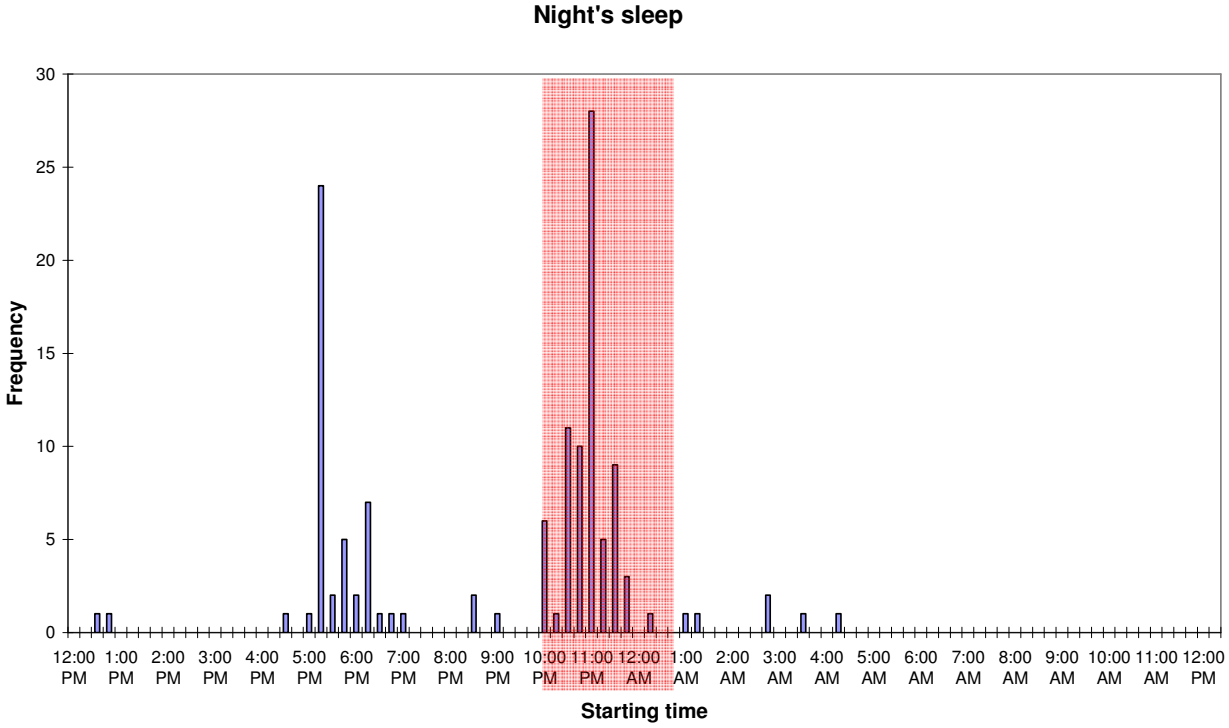

Figure 3. Night's sleep activity: dispersion of starting time

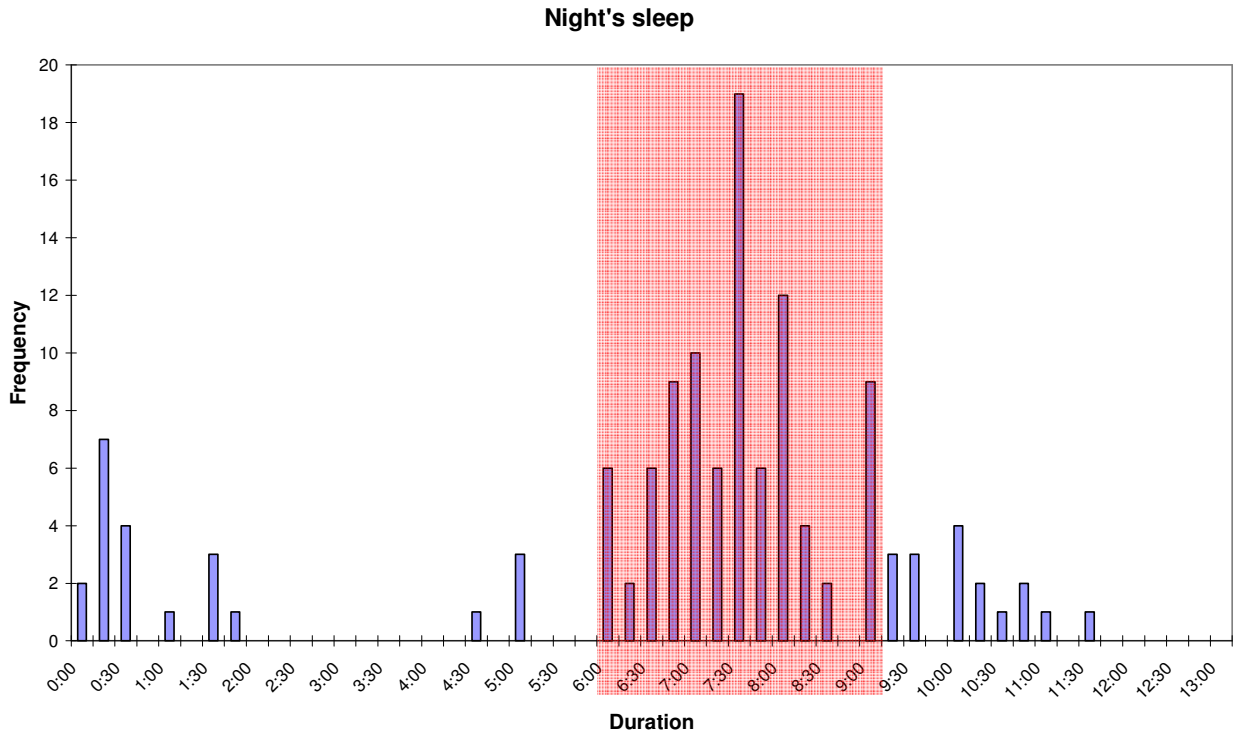

Figure 4. Night's sleep activity: dispersion of duration

3 Basics of reinforcement learning

The modelling approach proposed in the present study is based on reinforcement learning with function approximation, combining reinforcement learning with inductive learning. In order to comprehend the reinforcement learning problem following concepts need to be explained first:

- an *agent*: an agent denotes the decision making unit under consideration, being an individual or household within the context of this research.
- a set of possible *states S*: a system is composed of a finite set of states *S*, which are compounded of a number of dimensions, such as the time frame and the activity history.
- a set of possible *actions A(s)*: the action set *A(s)* for a certain state *s* refers to all possible decisions given that states. Such action can be determined for instance by the choice of the activity type and its duration.
- an unknown *transition* function $\delta : S \times A \rightarrow S$: for a given state *s*, the transition function determines the next state *s'* following action *a*.
- an unknown *reward* function $R(s,a) : S \times A \rightarrow R$: while making decisions an agent receives feedback, which can be either immediate or delayed, and direct or indirect. This feedback is called the *reward R(s,a)* and can be compared to the concept of utility in conventional choice models. The reward depends on the goal of the reinforcement learning problem.
- a *discounting factor* γ: a reinforcement learning agent does not only take into account immediate rewards, he also incorporates the effect of delayed rewards. To serve this purpose, a discounting factor $\gamma < 1$ has been introduced, which reflects the weight assigned to immediate versus future rewards.

(Kaelbling *et al*, 1996; Sutton and Barto, 1998)

The goal of the agent consists of learning an optimal *policy* $\pi^* : S \rightarrow A$ that maximizes the discounted sum of the rewards V(s).

$$V^{\pi}(s_t) = \sum_{t=0}^{\infty} \gamma^i r_{t+i}$$

In the present research, Q-learning will be used to select this optimal policy. This approach to reinforcement learning is based on the Q-value of an action *a* given a state *s*, which denotes the expected utility of taking action *a* in state *s*. This value is defined as follows:

$$Q^{\pi}(s,a) = r(s,a) + \gamma Q^{\pi}(s',a')$$

(Watkins, 1992; Kaelbling *et al*, 1996; Sutton and Barto, 1998)

The actual Q-learning process can be written as follows:

Initialize Q-values.
Repeat N times (N = number of learning episodes)
Select a random state *s* which has at least one possible action to select from.
Select an action *a*, one of the possible actions in state *s*.
Receive an immediate reward *R(s,a)*.
Observe the next state *s'*.
Update the Q-value of the state-action pair (*s*,*a*) according to following update rule:

$$Q_{t+1}(s,a) = (1-\alpha).Q_t(s,a) + \alpha\left[R(s,a) + \gamma.\max_{a'} Q_t(s',a')\right]$$

Where

$Q_t(s,a)$ is the Q-value of the current time step,

$Q_{t+1}(s,a)$ is the updated Q-value,

α is called the step-size parameter or learning rate of the algorithm and expresses the weight assigned to the "newly" calculated Q-value in comparison with the old, saved estimate of the Q-value. This learning rate will decrease as the number of learning episodes increases, reflecting the fact that the Q-value actually equals a weighted average of all experiences.

Set *s = s'*

(Kaelbling *et al*, 1996)

Faced with a decision in each state, the agent has to decide whether to explore the possible actions or to exploit the previously gathered knowledge when taking an action. Exploiting signifies choosing the action that is known to yield the highest reward. By doing so, the agent aims at reaching the state that is next to the currently best solution. This is a so-called greedy approach. Exploring denotes the random selection of a possible action. The goal of exploring is arriving at a state that might not be visited otherwise, and which may produce a higher reward than that of the most optimal action so far. (Mitchell, 1997; Sutton and Barto, 1998; Janssens, 2005)

A method to incorporate the trade-off between exploration and exploitation includes introducing a parameter, the so-called exploration rate $p_{explore}$, which reflects the probability of selecting a random action instead of the optimal one. Exploration generally occurs more in the beginning of the learning process – at that moment the agent still needs to "discover" his environment – and thus the exploration rate often decreases with increasing learning episodes. (Janssens, 2005)

4 Disadvantages of simple RIL

Above described Q-learning process requires storing Q-values of all possible state-action pairs calculated during the iterative learning process in a look-up table. Furthermore the traditional Q-learning approach requires all state-action pairs to be visited at least once and preferably a considerable number of times during the training process in order to determine the optimal policy. Because of this, the simple algorithm is only applicable to small state-space problems, due to the fact that the look-up table will explode exponentially whenever either the number of states or the number of possible actions increases. Large state-space problems will thus require a huge amount of memory to store the large Q-tables as well as a huge amount of time to estimate the Q-values accurately in the course of the learning process. (Sutton and Barto, 1998)

For instance in this area of research, one wants to learn an activity sequence for 1 week (7 days), being composed of 6 activities. Assume that in this case the state is composed of the day of the week,

the starting time of the activity (expressed in time slots of 15 minutes) and the activity history for all 6 activities which reflects time elapsed since the last episode of each activity (also expressed in time slots of 15 minutes). The action consists of the choice of the activity and the duration of this activity (also expressed in time slots of 15 minutes) with a maximum duration of 12 hours. The look-up table of this problem will contain 780,337,152[2] entries which need to be visited 'often enough'!

Furthermore when the goal of the learning problem changes, or when environmental changes occur, the simple reinforcement learning method requires retraining the Q-function from scratch. For example, when the individual moves, or when he changes jobs, the traditional reinforcement algorithm will not recover previously acquired knowledge, though the majority of his settings (for instance working hours, opening hours of shops and public services and leisure preferences) will remain unchanged.

The key issue to meet these problems consists of applying so-called function approximation. This signifies generalisation from the experience of only a limited subset of the state-action space to all state-action pairs, even the ones that have never been visited. To this end, existing generalization techniques from the area of supervised learning, such as neural networks, pattern recognition and statistical curve fitting, can be used. (Sutton and Barto, 1998)

5 Extended reinforcement learning algorithm

5.1 The algorithm

To face the disadvantages of the traditional reinforcement learning algorithm, inductive learning can be incorporated into the reinforcement learning approach. Inductive learning will allow generalization based on the state, the actions and the Q-values of subsequent learning phases. To this purpose the current approach utilizes a regression tree technique. From that perspective, the traditional learning algorithm can be rewritten into the following scheme:

Initialize the Q-tree.
Repeat N times (N = number of learning episodes)
    Select a random state s_0.
    Set $s = s_0$.
    Repeat until end of episode
        Select action a
            Choose exploration parameter randomly.
            If exploration parameter < exploration rate
                Choose best action.
                    Walk through Q-tree given the values of state s to list all possible actions and their corresponding Q-value.
                    Select from this list the action a with the highest Q-value.
            Else
                Choose random action
                    Choose action a randomly.
        Receive immediate reward R(s,a).
        Calculate Q-value Q(s,a).

$$Q_t(s,a) = R(s,a) + \gamma . \max_{a'} \hat{Q}_t(s',a')$$

        Save triplet [s, a, Q(s,a)].
        Observe next state s'.
        Set s = s'.
    Fit Q-tree.

[2] 7 days * 24*4 starting times * 6*(7*24*4) activity history possibilities * 6 activities * 12*4 duration possibilities

5.2 The reward function

Another major challenge of this research includes the fact that the reward the agent receives from it's "environment" cannot be observed directly from the activity-travel diary data. Therefore, the current research has tried to capture the reward based on the available data, in particular the state, which is composed of the starting time, the day of the week and the history data for each activity, and the action, which consists of the activity and the activity duration. The immediate reward is composed of the following components:

- Reward based on **activity duration**: this research assumes that, if the activity duration falls within the range of feasible durations, the reward of duration increases as the activity duration approaches the average activity duration. When the activity duration is less than the minimum activity duration or more than the maximum duration, a negative reward or penalty of duration will be assigned. The average duration, as well as the minimum and maximum feasible duration, are calculated from the available data. Furthermore, the average duration takes into account the time-of-the-day the activity is executed and the activity history.

$$R_{duration} = \begin{cases} -25 * e^{\left(\frac{duration - min\,duration}{stdev\,duration}\right)^2} & \text{if duration} < \text{minduration} \\ -25 * e^{\left(\frac{max\,duration - duration}{stdev\,duration}\right)^2} & \text{if duration} > \text{maxduration} \\ 100 * e^{\left(\frac{avg\,duration - duration}{stdev\,duration}\right)^2} & \text{else} \end{cases}$$

- Reward based on **day of the week**: this reward incorporates the fact that individuals prefer to perform certain activities on certain days within the week. For instance, for a particular individual the shopping activity usually occurs on Saturday, but occasionally – and for a number of reasons - he will go for grocery shopping on Monday. This preference can be reflected by calculating the portion of activity episodes on a certain day of the week compared to the number of episodes within the entire week. This preference can be computed from the available data as well.

$$R_{weekday} = \frac{number\ of\ activity\ episodes\ on\ weekday}{total\ number\ of\ activity\ episodes}$$

- Reward based on **time-of-the-day**: according to the same line of thought, the preference to perform an activity on a specific time-of-the-day has been set. To this end, eight time-of-the-day categories are determined based on the starting time of the activity. Each time-of-the-day category compromises a time slot of three hours. For instance, for the time window between 12 AM up to 3 AM, time-of-the-day equals 0; between 3 AM up to 6 AM time-of-the-day equals 1, and so on. The time-of-the-day preference is calculated for each activity by averaging the portion of activity episodes within a certain time-of-the-day category with regard to the number of episodes of the activity under consideration and with regard to the number of episodes of all activities executed during this specific time-of-the-day category. This preference can also be estimated from the available data.

$$R_{time\ of\ day} = \frac{\frac{number\ of\ activity\ episodes\ on\ time\ of\ day}{total\ number\ of\ activity\ episodes} + \frac{number\ of\ activity\ episodes\ on\ time\ of\ day}{total\ number\ of\ all\ activity\ episodes\ on\ time\ of\ day}}{2}$$

- Reward based on **activity history**: the reward of history rises when the activity history comes closer to the average activity history, for an activity history within the range of feasible activity histories. When the activity history is less than the minimum activity history or more than the maximum activity history, the reward of history becomes negative. The average history, the minimum and maximum history can be determined from the available data.

$$R_{history} = \begin{cases} -100 * e^{\left(\frac{history - \min history}{stdev\ history}\right)^2} & \text{if history} < \text{minhistory} \\ -100 * e^{\left(\frac{\max history - history}{stdev\ history}\right)^2} & \text{if history} > \text{maxhistory} \\ +100 * e^{\left(\frac{avg\ history - history}{stdev\ history}\right)^2} & \text{else} \end{cases}$$

- Additional reward of **fixed activities**: as already mentioned above, some activities, such as work and night's sleep, can be considered to be rather fixed. In order to enforce these activities within the observed boundaries, an additional reward is granted to these activities if they occur within the postulated boundaries. The agent receive an additional reward of 200 for starting to work between 5 AM and 10 AM for 2:45 up to 4:15 hours or for 8 up to 10 hours on working days (Monday to Friday). In case he executes the sleeping activity, starting between 10 PM and 1:15 AM for 6 to 9 hours, the agent also receives an additional reward of 200.

$$R_{fixed} = +200 \qquad \text{if activity between fixed boundaries}$$

- Reward based on **waiting time**: some activities can only be performed within certain feasible time windows, due to opening hours of shops and public services. To incorporate such constraints, an action variable waiting time has been introduced. This waiting time indicates the amount of time that the agent needs to wait between the ending time of the previous activity and the earliest starting time of the currently selected activity. Instead of assigning a large penalty to waiting time, the size of negative reward of waiting is rather low for small waiting times but rises quickly with increasing waiting time, as it might be more useful to wait a few minutes, e.g. for a shop to open, instead of performing another activity, with a lower overall reward, first. (Charypar and Nagel, 2005)

$$R_{waiting\ time} = -2 * waiting\ time$$

  The constraints enforced in this manner include the opening hours of shops for the activity "shopping and services", accessible from 8 AM on, and working hours for the activity "working", accessible from 5 AM on.

  The closing time for these activities is set to 6 PM for the shopping and services activity and 8 PM for the work activity. These ending times will be enforced as the duration is decreased when the finishing time of the activity exceeds the latest possible finishing time, so as to ensure that the ending time of the activity lies within the feasible time window. This adaptation causes the duration reward either to increase when the duration approaches the average duration, or to decrease or even become negative when the duration shifts away from the average duration to or below the minimum duration.

It should be mentioned that this immediate reward entirely depends on the underlying data. Such reward function thus enables capturing differences in activity patterns according to socio-economic variables as these are also recorded in the observed dataset.

5.3 Parameters of the algorithm

In the following section the parameters of the current reinforcement learning algorithm will be discussed. The algorithm will learn the "optimal" activity sequence in the course of only 500 learning episodes as opposed to required by the traditional reinforcement learning algorithm.

Furthermore, the step size parameter α depends on the number of cases in the leaf of the Q-tree corresponding to the (state, action)-pair and can be calculated as follows:

$$\alpha = \frac{1}{1 + \text{number of cases}}$$

The discounting factor γ has been 0.5 set to in this research.

$$\gamma = 0.5$$

The exploration rate used in the current research equals:

$$\text{Exploration rate} = 1 - e^{-\frac{500 - \text{learning episode number}}{250}}$$

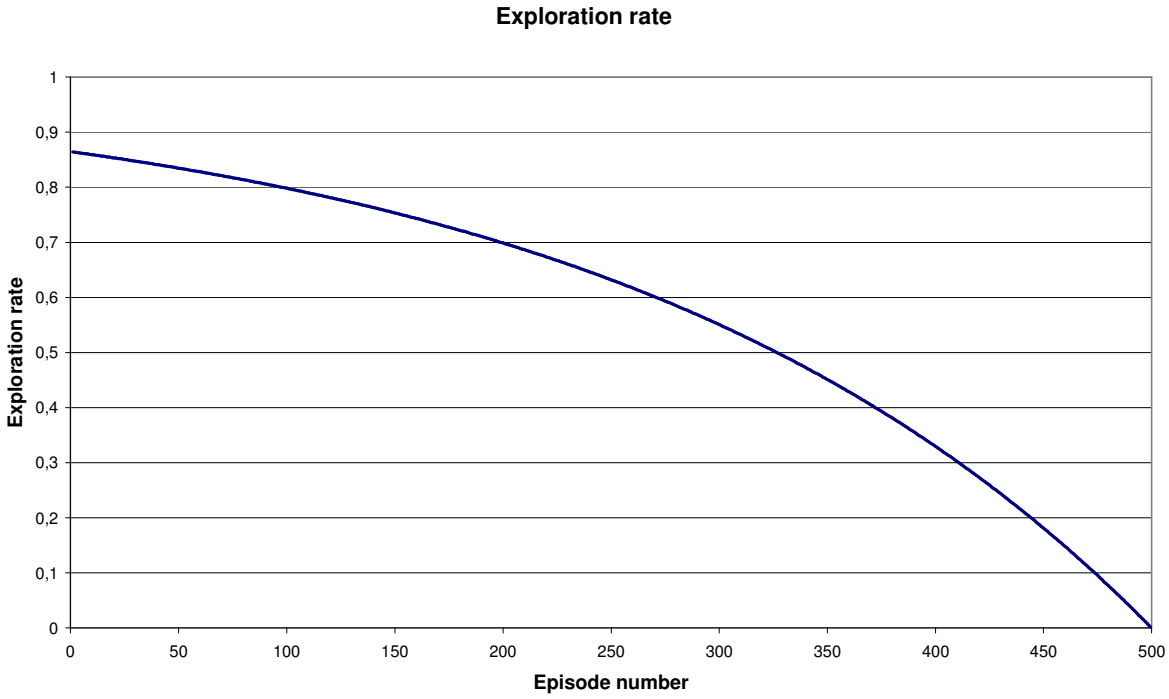

Figure 5. Evolution of exploration rate with regard to the episode number

The regression tree constitutes the innovative part of the current reinforcement learning algorithm. This regression tree, or Q-tree, generalizes the (state, action, Q-value)-triplets and is estimated based on the examples gathered during the learning phase of the reinforcement learning algorithm. However, as agents face a continuously changing environment, causing (state, action, Q-value)-triplets to become obsolete or even invalid, and as the number of examples increases with the number of learning episodes, only a portion of the experienced triplets are used to fit the regression tree. In addition, these examples are weighted based on their relative ages, assigning as such more weight to cases currently experienced than those experienced in the past. The number of examples selected depends on the learning episode:

$$\text{Percentage of examples selected} = \frac{500 - \frac{\text{learning episode number}}{5}}{500}$$

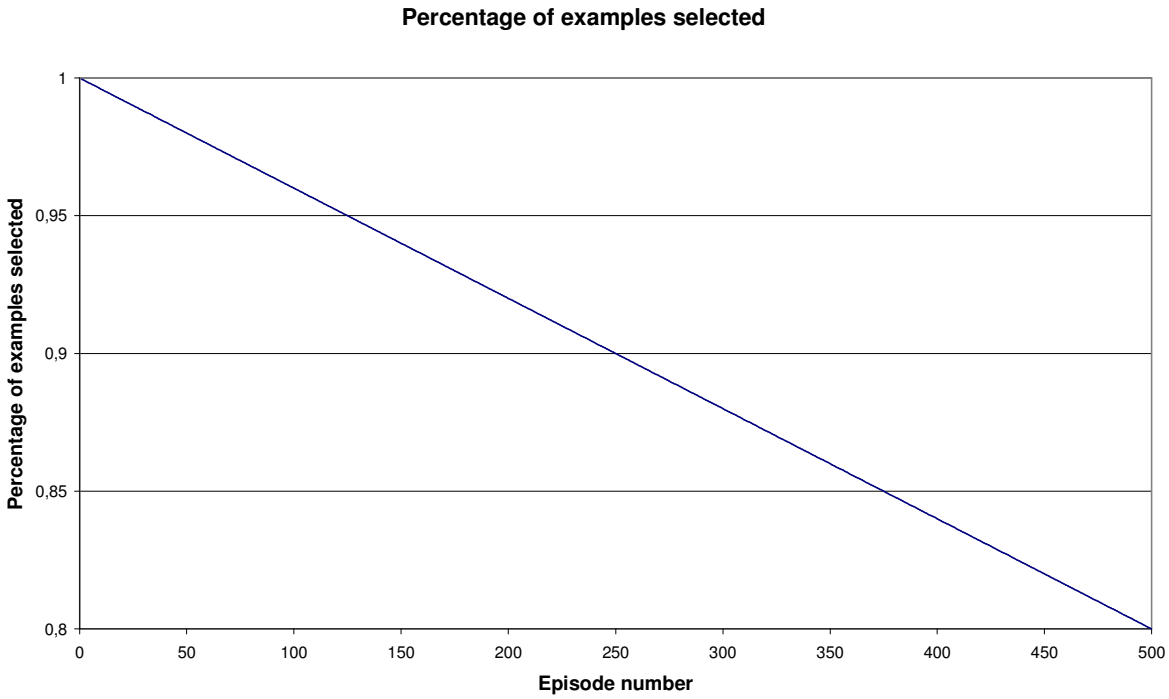

Figure 6. Evolution of the percentage of examples selected with regard to the episode number

The regression tree algorithm applied in this paper is based on the Classification And Regression Tree (CART) induction algorithm elaborated in Breiman *et al* (1984).

5.4 Results

Having implemented above extended reinforcement learning algorithm, the method was run sixteen times to match the sixteen observed activity sequences. The time to learn the "optimal" activity sequence for one individual fluctuated around one hour. The table on the next page contains one of simulated activity patterns.

As one will notice, the agent does take into account the postulated temporal constraints, as well as the boundaries of the skeleton activities, work and night's sleep. The simulated activity pattern still reveals some rather unusual behaviour, but – as one might expect – the performance will increase when either the number of learning episodes increases, or when the reward function is tuned more accurately to the observed data.

Weekday	Starting time	Finishing time	Duration	Activity
Monday	7:15 AM	5:15 PM	10:00	Working
	5:15 PM	9:15 PM	4:00	In-home
	9:15 PM	11:15 PM	2:00	In-home
	11:15 PM	5:00 AM	5:45	Sleeping
Tuesday	5:00 AM	3:00 PM	10:00	Working
	3:00 PM	6:00 PM	3:00	In-home
	6:00 PM	9:30 PM	3:30	Leisure
	9:30 PM	5:00 AM	7:30	In-home
Wednesday	5:00 AM	1:30 PM	8:30	Working
	1:30 PM	3:30 PM	2:00	Leisure
	3:30 PM	7:30 PM	4:00	In-home
	7:30 PM	9:15 PM	1:45	In-home
	9:15 PM	12:15 AM	3:00	Sleeping
Thursday	12:15 AM	8:45 AM	8:30	Sleeping
	8:45 AM	5:15 PM	8:30	Working
	5:15 PM	5:45 PM	0:30	In-home
	5:45 PM	6:30 PM	0:45	Leisure
	6:30 PM	3:00 AM	8:30	Sleeping
Friday	3:00 AM	3:15 AM	0:15	In-home
	3:15 AM	6:45 AM	3:30	Sleeping
	6:45 AM	7:15 AM	0:30	Working
	7:15 AM	12:30 PM	5:15	In-home
	12:30 PM	4:45 PM	4:15	In-home
	4:45 PM	6:15 PM	1:30	Leisure
	6:15 PM	2:00 AM	7:45	Sleeping
Saturday	2:00 AM	7:45 AM	5:45	In-home
	7:45 AM	9:15 AM	1:30	Sleeping
	9:15 AM	9:30 AM	0:15	Working
	9:30 AM	12:30 PM	3:00	Leisure
	12:30 PM	5:30 PM	5:00	Leisure
	5:30 PM	10:45 PM	5:15	Sleeping
	10:45 PM	7:15 AM	8:30	Sleeping
Sunday	7:15 AM	7:45 AM	0:30	Working
	7:45 AM	1:45 PM	6:00	Other
	1:45 PM	3:15 PM	1:30	Leisure
	3:15 PM	11:15 PM	8:00	Sleeping
	11:15 PM	7:15 AM	8:00	Sleeping

Table 1. Example of a simulated activity pattern

The simulated activity patterns can be compared to the observed ones by calculating a distance measure by means of the DANA-tool (Dissimilarity ANalysis of Activity-travel patterns) developed by C.H. Joh, T.A. Arentze and H.J.P. Timmermans (2001). The distance between two activity patterns is based on a Sequence Alignment Method (SAM) and indicates how much effort is needed to transform one sequence into the other one. The higher the SAM-score, the more maneuvers (inserting, deleting or reordering of activities) have to be performed in order to equalize the patterns and thus the less similar the sequences are. (Joh *et al*, 2001)

The mutual distances between the observed activity sequences vary from 142 to 467, whereas the distances between the observed activity sequences and the corresponding simulated ones range from 317 to 587 (average of 450). The chart on the next page visualises the similarity of the observed and simulated weekly activity pattern of one of the individuals.

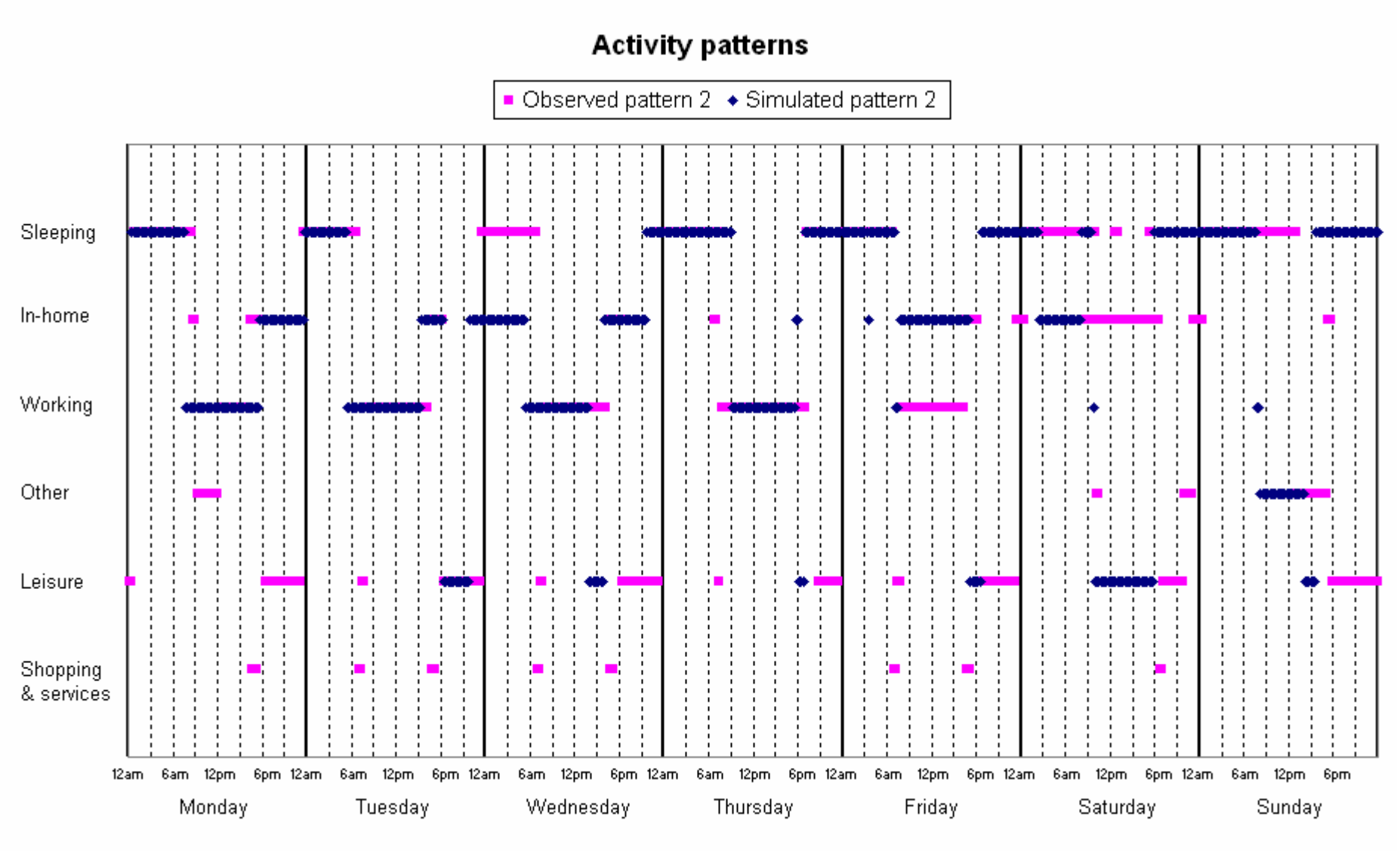

Figure 6. Comparison of an observed activity pattern with it's corresponding simulated activity pattern

6 Further research

As already mentioned earlier, the current research will have to be improved in order to incorporate location choice and travel mode decisions. Location choice can be introduced into the extended reinforcement learning algorithm by allocating a reward founded on land use characteristics, location preference and distance from the base locations, which are either the home or the work location. Travel mode choice will be included through the assignment of a negative reward depending on travel time and travel distance, and a positive reward based on travel mode preference and ease.

Furthermore, based on the available socio-demographic data individuals will be classified into clusters of individuals revealing similar activity-travel sequences. These clusters will form the basis of the simulation of the improved extended reinforcement learning algorithm as the algorithm will be applied to all of these clusters. In addition interactions between agents will have to be taken into account within the learning algorithm as well. After all, such interactions cause the development of certain spatial and temporal constraints. Just consider coupling constraints, resulting from the interaction between household members, or congestion, resulting from the interaction of a large number of agents entering the transport network simultaneously.

For the moment, in order to generalize the (state, action, Q-value)-triplets within the reinforcement learning algorithm a regression tree is utilized. As the Q-tree is re-estimated at the end of each learning episode, all encountered (state, action, Q-value)-triplets need to be stored, the disadvantages of which are triple. Firstly the amount of memory used to store these examples increases. Secondly the amount of time needed to estimate the Q-tree and to retrieve information from this Q-tree also rises as the number of learning episodes goes up. Finally, previously experienced (state, action, Q-values)-triplets might become outdated or even invalid, and should thus be excluded from the Q-tree estimation. The latter issue has partly been tackled by the current algorithm as only a part of the cases are used to estimate the Q-tree. Moreover the (state, action, Q-value)-triplets are weighted as such that the most recently experienced examples are more likely to be selected for estimation of the Q-tree. However other approaches, such as incremental inductive learning techniques, enable handling all issues. The use of such techniques within this area of research will be examined.

7 Conclusions

It has been proven in this paper that - while several other scheduling algorithms exist -, the presented approach using algorithms originating from the research area of artificial intelligence, offers a great opportunity of becoming a reliable learning algorithm for micro-simulation in dynamic activity-based transportation models. The currently used technique has combined reinforcement learning with inductive learning, allowing as such the incorporation of a more extensive set of variables as well as an increased granularity of both the explanatory and the predicted variables.

After a profound examination of the extended reinforcement learning algorithm, this approach has been applied to fit the observed activity sequences of sixteen full-time working individuals being part of a four-headed household. The reward function, one of the major building blocks of a reinforcement learning algorithm, has been designed carefully based on the starting time and weekday, the activity type, the activity duration, the waiting time and the activity history in order to reflect the underlying needs of the agents.

The results of this study have proved to be very promising. These results have shown that, after the initial learning phase, the agent is able to determine a quite optimal activity sequences autonomously. In addition, in the course of the decision process, the agent has also been found to respect both the postulated temporal constraints and boundaries of fixed activities, such as work and night's sleep. Furthermore, this research has revealed that different activity sequences can be modelled based on socio-demographics.

The results have been validated by means of the distance measure, generated by SAM (Sequence Alignment Method) determined in Joh, *et al.* (2001) and indicating the (dis)similarity of activity patterns. This analysis has revealed that the average distance between the observed activity patterns and the simulated ones falls within the range of mutual distances of the observed activity sequences.

In spite of these favourable results, some issues still remain to be examined. These include among other things: incorporating the location choice and travel mode decision, applying the (adjusted) algorithm to all clusters of individuals and improving the Q-tree induction algorithm.

8 References

Arentze, T., Pelizaro, C. and Timmermans, H. (2005) "Implementation of a Model of Dynamic Activity-Travel Rescheduling Decisions: an Agent-Based Micro-Simulation Framework", *9th International Conference on Computers in Urban Planning and Urban Management*, July 2005, London, UK.

Arentze, T.A. and Timmermans, H.J.P. (2004) A Learning-Based Transportation Oriented Simulation System. *Transportation Research Part B: Methodological*, Vol. 38, No. 7, pp.613-633.

Arentze, T. and Timmermans, H. (2005) "Modeling Learning and Adaptation in Transportation Contexts", *Transportmetrica*, Vol. 1, No. 1 - Special issue: Some Recent Advances in Transportation Studies, pp.13-22.

Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classifcation and Regression Trees*. Wadsworth.

Charypar, D., Graf, P. and Nagel, K. (2004) "Q-Learning for Flexible Learning of Daily Activity Plans", *Proceedings of the 4th Swiss Transport Research Conference* (STRC), Monte Verità, Ascona, Czechoslovakia.

Charypar, D. and Nagel, K. (2005) "Generating Complete All-Day Activity Plans with Genetic Algorithms", Springer, *Transportation*, Volume 32, Issue 4, July 2005, pp. 269-397.

Ettema, D. and Timmermans, H. (1997) *Activity-Based Approaches to Travel Analysis*. Elsevier Science Ltd, Oxford, UK, 1st edition.

Frusti, T., Bhat, C.R. and Axhausen, K.W. (2002) "An Exploratory Analysis of Fixed Commitments in Individual Activity-Travel Patterns", *Transportation Research Record: Journal of the Transportation Research Board*, Washington DC, pp. 101-108.

Janssens, D. (2005) *Calibrating Unsupervised Machine Learning Algorithms for the Prediciton of Activity-Travel Patterns*. Doctoral dissertation, Hasselt University, Faculty of Applied Economics, Belgium.

Joh, C.-H. Arentze, T.A. and Timmermans, H.J.P. (2001) "Pattern Recognition in Complex Activity-Travel Patterns: A Comparison of Euclidean Distance, Signal Processing Theoretical, and Multidimensional Sequence Alignment Methods", *Presented at the 80th Annual Meeting of the Transportation Research Board*, January 7-11, Washington D.C., USA.

Joh, C.-H., Arentze, T.A. and Timmermans, H.J.P. (2004) "Activity-Travel Rescheduling Decisions: Empirical Estimation of the Aurora Model", *Transportation Research Record*, Vol. 1898, pp. 10-18.

Kaelbing, L.P., Littman, M.L. and Moore, A.W. (1996) "Reinforcement Learning: A Survey", *Journal of Artificial Intelligence Research*, Volume 4, p. 237-285.

Mitchell, T.M. (1997) *Machine Learning*. The McGrawhill Companies, Inc., 1997, USA, Chapter 13.

Schwanen, T. and Dijst, M. (2003) "Time Windows in Workers' Activity Patterns: Empirical Evidence from the Netherlands", *Transportation*, Volume 30, Number 3, August 2003, pp. 261-283.

Sutton, R.S. and Barto, A.G. (1998) *Reinforcement Learning: An Introduction*. The MIT Press, 1998, HTML-version: http://www.cs.ualberta.ca/~sutton/book/ebook/the-book.html, Cambridge, Massachusetts, USA/London, England.

van der Waerden, P. and Timmermans, H. (2003) “Key Events and Critical Incidents Influencing Transport Mode Choice Switching Behavior: An Exploratory Study”, *Proceedings of the 82nd Annual Research Board Meeting*, January 12-16, Washington DC.

Vanhulsel, M., Janssens, D. and Wets, G. (2007) “Calibrating a New Reinforcement Learning Mechanism for Modeling Dynamic Activity-Travel Behavior and Key Events”,*Presented at the 86th Annual Research Board Meeting*, January 21-25, Washington DC.

Watkins, C.J.C.H. and Dayan, P. (1992) "Technical Note: Q-Learning", *Machine Learning*, Volume 8, Number 3-4, May 1992, 279-292.