ELSEVIER
European Journal of Operational Research 131 (2001) 232–241
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

Reinforcement learning in neurofuzzy traffic signal control

Ella Bingham *

Laboratory of Transportation Engineering, Helsinki University of Technology, P.O. Box 2100, 02015 HUT, Finland

Abstract

A fuzzy traffic signal controller uses simple "if–then" rules which involve linguistic concepts such as *medium* or *long*, presented as membership functions. In neurofuzzy traffic signal control, a neural network adjusts the fuzzy controller by fine-tuning the form and location of the membership functions. The learning algorithm of the neural network is reinforcement learning, which gives credit for successful system behavior and punishes for poor behavior; those actions that led to success tend to be chosen more often in the future. The objective of the learning is to minimize the vehicular delay caused by the signal control policy. In simulation experiments, the learning algorithm is found successful at constant traffic volumes: the new membership functions produce smaller vehicular delay than the initial membership functions. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Fuzzy sets; Neural networks; Traffic signal control; Reinforcement learning

1. Introduction

This article discusses the use of reinforcement learning in neurofuzzy traffic signal control. Results of implementing a neural reinforcement learning algorithm in a fuzzy traffic control system are shown.

Most of the fuzzy traffic signal controllers used today are not adjustable, that is, the parameters of the controller remain the same in changing traffic situations. Using a neural network, different parameters can be found for different traffic volumes. This leads to a reduction in the vehicular delay caused by traffic signal control.

A combination of a neural network and a fuzzy system is called a *neurofuzzy* system. In neurofuzzy control, the parameters of the fuzzy controller are adjusted using a neural network. Neurofuzzy systems utilize both the linguistic, human-like reasoning of fuzzy systems and the powerful computing ability of neural networks. They can avoid some of the drawbacks of solely fuzzy or neural systems. The literature on neurofuzzy systems is wide. See, for example [4,6–10]. *Reinforcement learning*, on the other hand, is a learning algorithm for a neural network. It is based on evaluating the system performance and giving credit for successful actions. Reinforcement learning is used when the information obtained from the system is such that simpler supervised

* Present address: Neural Networks Research Centre, Helsinki University of Technology, P.O. Box 5400, FIN-02015 HUT, Finland. Tel.: +358-9-451-5282; fax: +358-9-451-3277.

E-mail address: ella.bingham@hut.fi (E. Bingham).

PII: S 0 3 7 7 - 2 2 1 7 (0 0) 0 0 1 2 3 - 5

learning algorithms common in neural networks cannot be used. The literature on reinforcement learning, especially in the context of fuzzy control, includes, e.g. [1,5,11,16].

The objective of our traffic signal controller is vehicular delay minimization. This is only one of several objectives of real-life traffic signal controllers. Others include, e.g. safety and environmental aspects, and optimizing one particular goal may lead further away from the optima of the remaining ones. Delay minimization was chosen as the goal in this work because it is fairly easy to measure, and because the interest was more in demonstrating the potential of neural networks in fuzzy traffic signal control than in studying the various aspects of traffic signal control.

A large series of experiments on an adjustable neurofuzzy traffic signal controller was conducted in the Laboratory of Transportation Engineering in Helsinki University of Technology. The traffic simulation system HUTSIM [15] includes a fuzzy signal controller FUSICO [12], and the simulation system interacts with a Matlab program especially designed for this purpose.

To our knowledge, this is the first application of neural networks in the fine-tuning of membership functions in fuzzy traffic signal control.

2. Fuzzy systems

The concepts and terminology of fuzzy logic were brought to public attention by Zadeh [17]. Fuzzy sets provide a mathematical interpretation for natural language terms. A fuzzy set is a generalization of a classical set in the sense that a fuzzy set may contain its elements partially, too, whereas an element of a classical set either belongs to the set or does not. In a fuzzy set S, each element x of the set is assigned with a *degree of membership* in S, which is measured by a *membership function* [17] $\mu_S : \mathbb{R} \to [0, 1]$. The membership function $\mu_S(x)$ is zero when x does not belong to S at all, one when x belongs to S totally and $0 < \mu_S(x) < 1$ when x belongs to S partially.

In this work, we use trapezoidal membership functions such as the one seen in Fig. 1. The adjustable parameters of the membership function are the locations of the four corners of the trapezoid: p_1, p_2, p_3 and p_4. By fine-tuning these parameters the shape and location of the membership function changes.

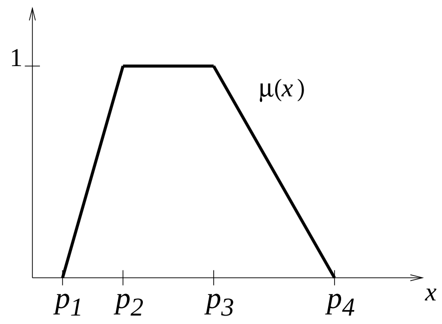

Fig. 1. Trapezoidal membership function. The parameters p_1, p_2, p_3 and p_4 determine the locations of the four corners.

One of the most practical and successful applications of fuzzy systems is fuzzy control. Fuzzy control uses a *rule base*, where the rules are propositions of the form "if X is S, then Y is T". Here X and Y are *linguistic variables*, for example *traffic volume* or *green signal extension*. In turn, S and T are *linguistic values* of the above variables, for example *medium* or *long*. Linguistic values are presented as fuzzy sets, each having its own membership function.

3. Fuzzy traffic signal control

Fuzzy traffic signal control is one type of vehicle-actuated, extension-based signal control, where the controller receives measurements of incoming traffic and chooses the length of the green signal accordingly. Pappis and Mamdani [13] were the first to apply fuzzy logic in traffic signal control.

A clear advantage of fuzzy control systems over traditional ones is their ability to use expert knowledge as such, in the form of fuzzy rules. Another advantage of fuzzy control is the small number of parameters needed: only the rule base and the parameters of the membership functions need to be selected, whereas in traditional traffic signal control, the number of parameters is very large. The parameters needed in fuzzy control are easy to comprehend, making the design process more suitable for human-like reasoning.

There is an important difference between fuzzy extension-based and traditional extension-based traffic signal controllers used today. A fuzzy controller uses the number of incoming vehicles in both the green and the red direction, whereas a traditional extension-based controller uses only the number of vehicles in the green direction when deciding the green time extension. A frequently occurring problem is that when vehicles approach the intersection separately but within a few seconds from each other, every vehicle is given a green light extension, and the total extension grows very large. Fuzzy control takes into account the length of the queue behind the red signal, too, and if the queue is too long compared to the amount of vehicles approaching from the green direction, no green extension is given anymore. In this way, a fuzzy traffic signal controller acts like a policeman who constantly weighs in his mind which of the directions deserves a green signal. In simulation tests, fuzzy control performs better than traditional extension-based signal control [12]. In the current study, fuzzy signal control is further developed by including a neural learning algorithm.

The traffic simulation environment used in this work is a two-phase controlled intersection of two-lane streets. The intersection configuration is the same as in Pappis and Mamdani's simulation [13]. In each approaching lane there are two traffic detectors, the first one before the stop line and the other at the stop line. These detectors send input measurements of traffic to the fuzzy controller: *APP*, number of approaching vehicles in the green direction and *QUE*, number of queuing vehicles in the red direction. Depending on the traffic situation, the green phase can be extended with one or several seconds, and the output of the fuzzy controller is *EXT*, green time extension (in seconds). The linguistic values of *APP* are *zero*, *a few*, *medium* and *many*; the linguistic values of *QUE* are *a few*, *medium* and *too long*; and the linguistic values of *EXT* are *zero*, *short*, *medium* and *long*.

The rule base consists of five rule sets. The choice of the rule set depends on how many green extensions have already been given. The objective of the rules is to split the green time and find the right moment of green termination so that the delay of vehicles is minimized. The rule base is

after minimum green (5 seconds)
- if *APP* is *zero*, then *EXT* is *zero*
- if *APP* is *a few* and if *QUE* is *less than medium*, then *EXT* is *short*
- if *APP* is *more than a few*, then *EXT* is *medium*
- if *APP* is *medium*, then *EXT* is *long*

after the first extension
- if *APP* is *zero*, then *EXT* is *zero*
- if *APP* is *a few* and if *QUE* is *less than medium*, then *EXT* is *short*
- if *APP* is *medium*, then *EXT* is *medium*
- if *APP* is *many*, then *EXT* is *long*

after the second extension
- if *APP* is *zero*, then *EXT* is *zero*
- if *APP* is *a few* and if *QUE* is *less than medium*, then *EXT* is *short*
- if *APP* is *medium* and if *QUE* is *less than medium*, then *EXT* is *medium*
- if *APP* is *many* and if *QUE* is *less than medium*, then *EXT* is *long*

after the third extension
- if *APP* is *zero*, then *EXT* is *zero*
- if *QUE* is *too long*, then *EXT* is *zero*
- if *APP* is *more than a few* and if *QUE* is *less than medium*, then *EXT* is *short*
- if *APP* is *medium* and if *QUE* is *less than medium*, then *EXT* is *medium*
- if *APP* is *many* and if *QUE* is *less than a few*, then *EXT* is *long*

after the fourth extension
- if *APP* is *zero*, then *EXT* is *zero*
- if *QUE* is *too long*, then *EXT* is *zero*
- if *APP* is *more than a few* and if *QUE* is *a few*, then *EXT* is *short*
- if *APP* is *medium* and if *QUE* is *less than a few*, then *EXT* is *medium*
- if *APP* is *many* and if *QUE* is *less than a few*, then *EXT* is *long*

Linguistic values of the form "*more than V*" and "*less than V*", where *V* is a linguistic value,

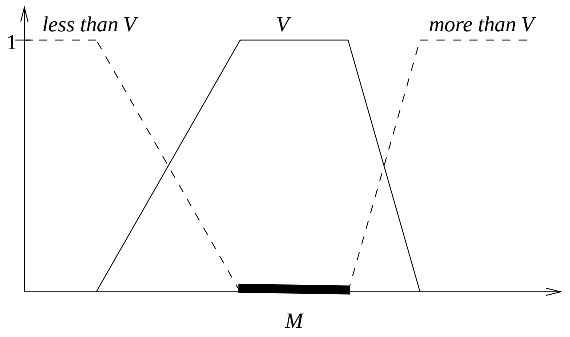

Fig. 2. Membership functions for linguistic values *V*, *less than V* and *more than V*. The thick line shows *M* (formula (1)).

are also used in the rule base. These are interpreted as functions of V, not as independent values, as seen in Fig. 2. The membership function of *more than V* is $\mu_{V+}(x) = 1 - \mu_V(x)$ when $x \geqslant \sup M$ and 0 otherwise, and the membership function of *less than V* is $\mu_{V-}(x) = 1 - \mu_V(x)$ when $x \leqslant \inf M$ and 0 otherwise. Here

$$M = \left\{ x \in X | \mu_V(x) = \max_x \{\mu_V(x)\} \right\} \tag{1}$$

which can also be seen in Fig. 2.

4. Reinforcement learning

Why is reinforcement learning needed? In the most simple case, the parameters of the fuzzy controller could be updated using the backpropagation algorithm [14] common in supervised learning in neural networks. In the backpropagation algorithm, the output of the network at each input is compared with a desired output, which is known in advance. In fuzzy traffic signal control, the output of the signal controller is the extension of the green signal, but the "desired" extension is not known. The objective of the signal controller is to minimize the delay of vehicles and not to reach a desired length of extension. Thus the standard backpropagation algorithm cannot be used; instead, a learning algorithm called *reinforcement learning* is used.

In reinforcement learning, the system evaluates whether the previous control action was good or not. If the action had good consequences, the tendency to produce that action is strengthened, that is, reinforced. Barto et al. [1] were among the first to discuss reinforcement learning in control problems. The algorithm used in this work follows mostly the GARIC algorithm presented by Berenji and Khedkar [2].

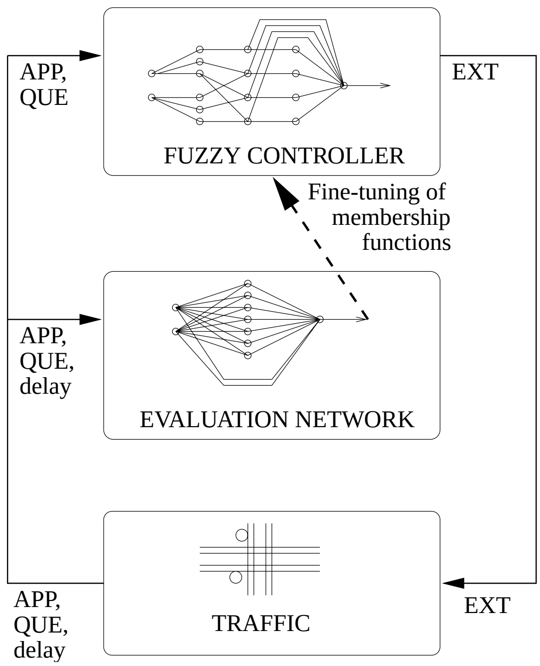

Fig. 3. The neurofuzzy traffic signal control system.

The structure of the neurofuzzy control system is seen in Fig. 3. The evaluation network gathers information about the decisions of the fuzzy controller and the delays of the vehicles. This reinforcement information is used in fine-tuning the membership functions of the fuzzy controller, which is also presented as a neural network. Thus there are actually two neural networks in the system: an evaluation network and a fuzzy controller network, both of which are presented in the following sections.

4.1. Evaluation network

The evaluation network evaluates the goodness of the actions of the fuzzy controller based on

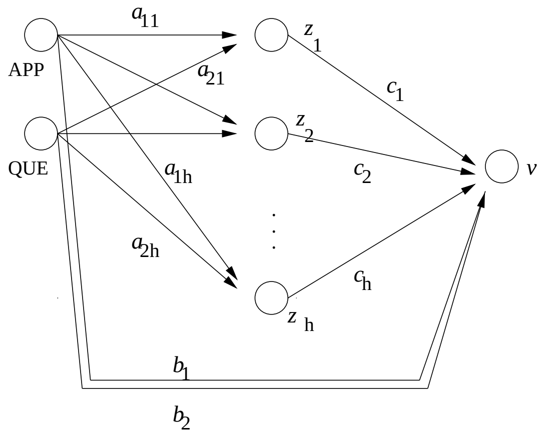

Fig. 4. The evaluation network [2].

information it has gathered by observing the process. The network fine-tunes the membership functions of the fuzzy controller by updating the parameters of the membership functions.

The structure of the evaluation network is presented in Fig. 4. It is a feedforward, multilayer perceptron-type network. The input variables of the network are *APP* and *QUE*, measurements of incoming traffic in green and red directions, respectively. The hidden layer activation function is a sigmoidal function $z_j(x_j) = 1/(1 + \exp(-x_j))$, where $x_j = a_{1j} * APP + a_{2j} * QUE$. The size h of the hidden layer may vary, and there are no precise rules for determining how many cells it should contain. Increasing the size gives a more powerful and flexible network but requires a longer learning time. In our work the size of $h = 10$ was found suitable.

The output layer of the evaluation network receives input values both from the input layer (*APP*, *QUE*) and the hidden layer ($z_j, j = 1, \dots, h$). The network output v is a measure of the goodness of the state of the network, a *prediction of future reinforcement* [2],

$$v = b_1 APP + b_2 QUE + \sum_{j=1}^{h} c_j z_j. \tag{2}$$

The *internal reinforcement* $\hat{r}$ rewards the system of successful behavior. For example, let the system move from a state with a low v (prediction of low reinforcement) to a state with a higher v (prediction of higher reinforcement). In other words, the state of the system improves. This positive change or internal reinforcement is used to reinforce the selection of the action which caused this move. The formula for internal reinforcement is [2]

$$\hat{r}(t) = -d(t) + \gamma v(t) - v(t-1), \tag{3}$$

where $d(t)$ is the delay of vehicles. In this formula, the value of v at time t is given less emphasis than the value of v at the previous time step $t-1$ by using a discount rate $0 \leqslant \gamma \leqslant 1$.

The gradient descent algorithm [14] is used in the learning phase of the evaluation network. If a *positive* internal reinforcement signal $\hat{r}$ is received, the network weights are rewarded by being changed in the direction which *increases* their contribution to the total sum. If a *negative* signal is received, the weights are punished by being changed in the direction which *decreases* their contribution [2]. For detailed learning formulae, see [3].

4.2. Fuzzy controller network

The fuzzy controller network is a feedforward network that encodes the decision-making in the fuzzy rule base. The parameters of the fuzzy controller network are the shape parameters of the membership functions, and the activation functions of the network are different fuzzy set operations. The rule base presented in Section 3 consists of five separate rule sets, the choice between which is done based on the situation. Each set forms a different neural network. As an example, consider the first rule set, whose neural network presentation is seen in Fig. 5. The layers of the network are presented in the following.

Similarly to the evaluation network presented in Section 4.1, the input variables of the fuzzy controller network are the measurements of incoming traffic, *APP* and *QUE*. The second layer computes the values of the membership functions of the input variables, for example, *zero* of *APP* or *less than medium* of *QUE*. Thus the second layer gives the degree to which *APP* is *zero* and *QUE* is *less than medium*, and so on.

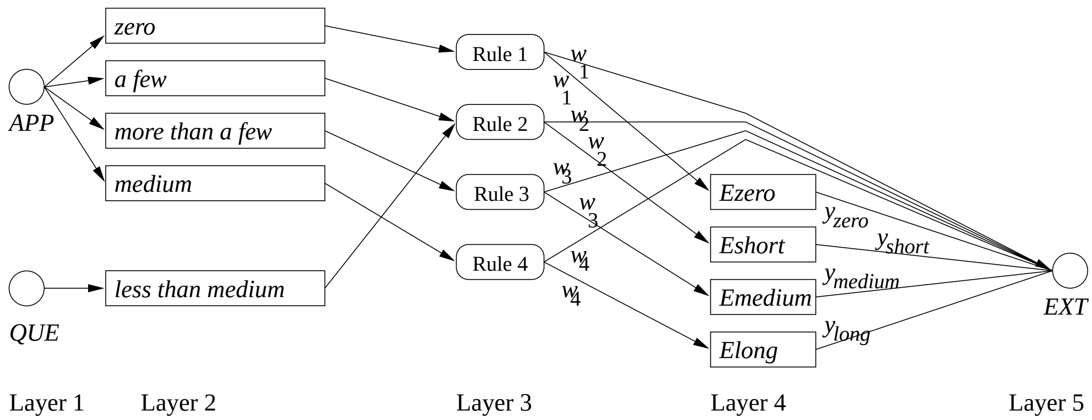

Fig. 5. Fuzzy traffic signal controller presented as a neural network. The first rule set of the rule base.

The third layer of the fuzzy controller network corresponds to the rule base. A cell i in the third layer combines all the conditions in the if-part of rule i and computes the rule firing strength w_i, the degree to which rule i is satisfied. In our network, the "and" combiner in the rules is interpreted as the minimum operator, so w_i is the minimum of the membership function values in rule i.

The fourth layer of the fuzzy controller network corresponds to the rule consequents. The fuzzy set (actually, the corresponding membership function) in the then-part of rule i is cut at the level indicated by the rule firing strength w_i, as seen in Fig. 6. The fuzzy set is defuzzified by computing the center of area y of the remaining set. The fifth layer calculates the total output of the rule base, a weighted average y^* of the outputs y of the rules, using rule firing strengths w as averaging weights. In this way those rules that fire to a high degree are given more emphasis.

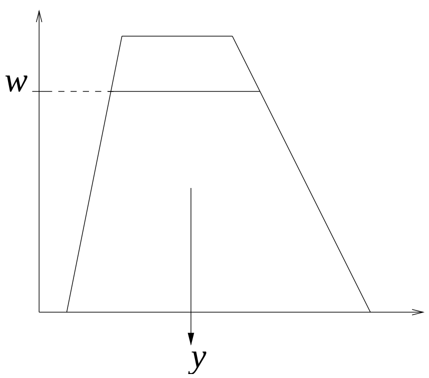

Fig. 6. The membership function in the then-part of the rule is cut at the level indicated by the rule firing strength w. The output y of the rule is the center of area of the remaining set.

Learning in the fuzzy controller network means updating the shape parameters of the membership functions. The only modifiable parameters in the fuzzy controller network in Fig. 5 are those in the second and the fourth layers – in the membership functions of if-parts and then-parts of the rules, respectively. We maximize v, the prediction of future reinforcement (Formula (2)), using the gradient descent algorithm [14]. Thus the parameters of the membership functions are modified in the direction which increases v: $p_{\text{new}} = p + \partial v/\partial p$. The detailed formulae are presented in [3].

5. Experimental results

We present here the results of using the reinforcement learning algorithm in a neurofuzzy traffic signal control system. The traffic volumes were 300, 500 and 1000 vehicles per hour. The location of the first traffic detector was 50 or 100 m from the stop line; these distances are often used in practice. Some other experimental results can be found in [3].

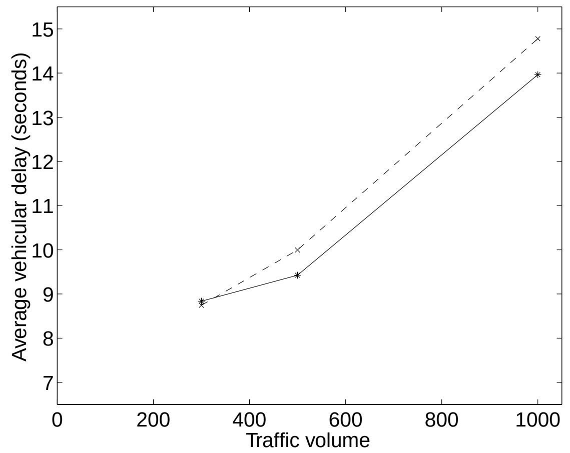

Fig. 7. Average vehicular delays (in seconds) before (dashed line) and after (solid line) the learning at traffic volumes of 300, 500 and 1000 vehicles per hour. The location of the first traffic detector is 50 m from the stop line.

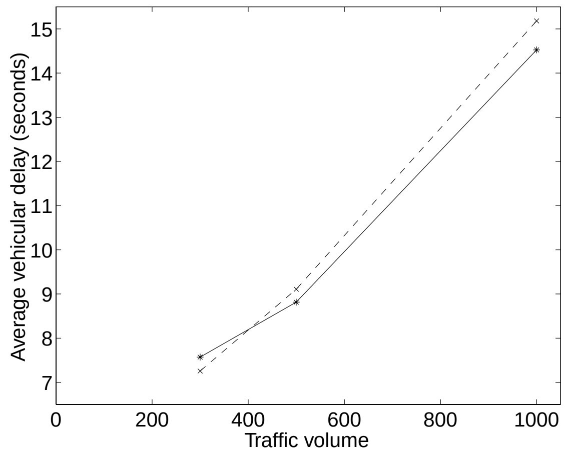

Fig. 8. Average vehicular delays (in seconds) before (dashed line) and after (solid line) the learning at traffic volumes of 300, 500 and 1000 vehicles per hour. The location of the first traffic detector is 100 m from the stop line.

Figs. 7 and 8 compare the delays before and after the learning. In the former figure, the location of the first traffic detector is 50 m from the stop line and in the latter, 100 m from the stop line. In both cases, the decrease in the delay is statistically significant at traffic volumes of 500 and 1000 vehicles per hour. The initial membership functions are obviously the most suitable ones at traffic volume of 300 vehicles per hour. Also, in the case of 300 vehicles per hour, the decision-making situations faced by the signal controller are often quite simple, and fuzzy control cannot show all its potential.

The statistical significance of the results in Figs. 7 and 8 is determined by t-tests on paired observations. (In the simulation experiments it is possible to use exactly the same vehicle sequences on both the initial and the new membership functions, whence the use of paired t-tests is appropriate.) The results are presented in Table 1. It is seen that the decrease in the delay is statistically significant at traffic volumes of 500 and 1000 vehicles per hour, at both traffic detector locations. The decrease in the delay is 3–6% per vehicle.

As an example on how the membership functions have changed, consider Figs. 9–11. They show how the membership functions of *APP*, *QUE* and *EXT* at a traffic volume of 500 vehicles per hour have changed during the learning, in the case where the first traffic detector is located 50 m

Table 1
Statistical significance of the decrease in the delay due to the learning[a]

Volume (veh/h)	Det. (m)	$\bar{d}_{\text{ini}}$	$\bar{d}_{\text{new}}$	$\bar{D}$	S_D	n	P-value	Conclusion
500	50	9.99	9.42	0.57	9.01	10 136	9.1×10^{-11}	Reject H_0
1000	50	14.78	13.96	0.81	12.32	20 016	0	Reject H_0
500	100	9.11	8.82	0.29	9.84	10 136	1.4×10^{-3}	Reject H_0
1000	100	15.18	14.52	0.66	14.35	20 016	5.1×10^{-11}	Reject H_0

[a] Hypotheses $H_0 : \mu_D = 0$ (the delay is not changed) and $H_1 : \mu_D > 0$ (the delay is decreased) are tested using a t-test on paired observations. Here $\bar{d}_{\text{ini}}$ and $\bar{d}_{\text{new}}$ are the average vehicular delays using the initial and the new membership functions, respectively, $D = d_{\text{ini}} - d_{\text{new}}$ is the difference of individual observations, $\bar{D}$ is the average of D, S_D is the sample standard deviation of D and n is the number of observations. It is seen that the decrease in the delay is statistically significant at traffic volumes of 500 and 1000 vehicles per hour, at both traffic detector locations.

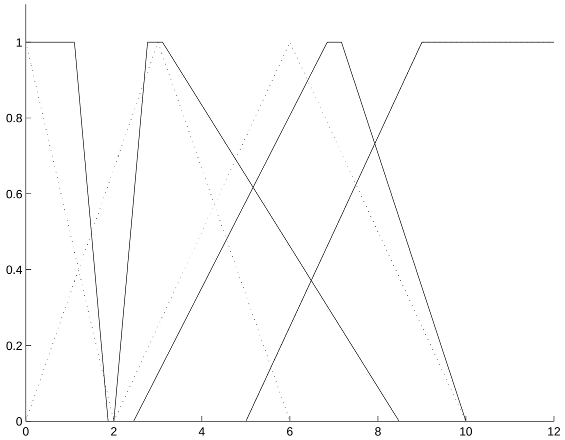

Fig. 9. Membership functions *zero*, *a few*, *medium* and *many* of *APP* before (dotted line) and after (solid line) the learning at a traffic volume of 500 vehicles per hour. The location of the first traffic detector is 50 m from the stop line. Horizontal axis: number of approaching vehicles. Vertical axis: value of membership function.

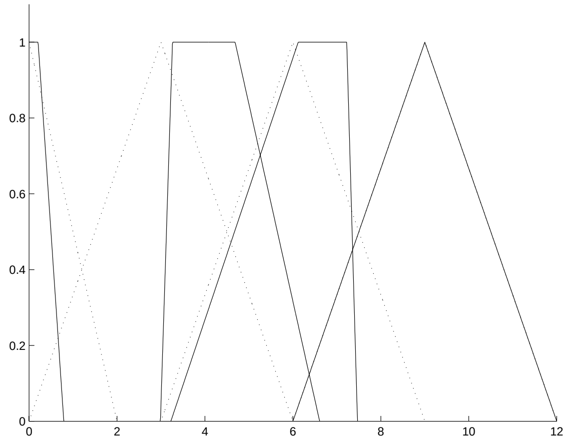

Fig. 11. Membership functions *zero*, *short*, *medium* and *long* of *EXT* before (dotted line) and after (solid line) the learning at a traffic volume of 500 vehicles per hour. The location of the first traffic detector is 50 m from the stop line. Horizontal axis: green signal extension in seconds. Vertical axis: value of membership function.

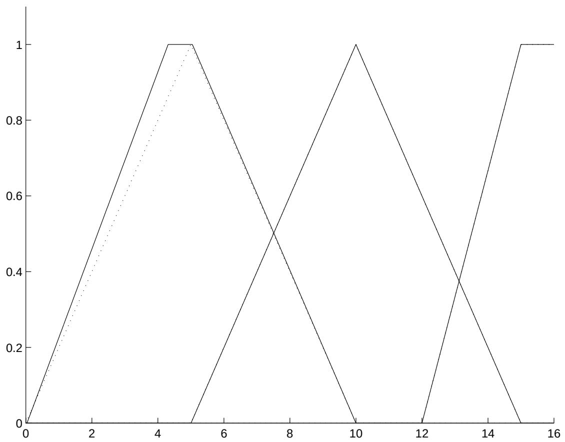

Fig. 10. Membership functions *a few*, *medium* and *too long* of *QUE* before (dotted line) and after (solid line) the learning at a traffic volume of 500 vehicles per hour. The location of the first traffic detector is 50 m from the stop line. Horizontal axis: number of queuing vehicles. Vertical axis: value of membership function.

from the stop line. It is observed that the functions *many* of *APP*, *medium* and *too long* of *QUE* and *long* of *EXT* were not updated. This is quite natural, because at a traffic volume of 500 vehicles per hour, there were seldom observations for which these membership functions were needed. As the first traffic detector was located only 50 m from the stop line, there were seldom very many vehicles between the detectors. The other membership functions were modified, function *zero* of *EXT* shrank and so did *short* and *medium* of *EXT*, but they also came closer to each other. The gap between *short* and *medium* indicates that extensions of two seconds are seldom given. In addition, as the first traffic detector is located 50 m before the stop line, the traffic signal controller cannot know if there are vehicles behind the 50 m point. With a speed of 40 km per hour a distance of 50 m takes 4.5 seconds, so it is wise to give an extension of at most 4–5 seconds so that the vehicles between the detectors can pass the stop line. A longer extension is unnecessary. Both *short* and *medium* of *EXT* are now concentrated around 5 seconds, so the fuzzy controller obeys this principle. Exactly the same phenomenon is seen when the traffic volume is 1000 vehicles per hour.

As another example of the modification of the membership functions, Fig. 12 shows how the membership functions of *APP* at a traffic volume of 1000 vehicles per hour have changed during the learning. The first traffic detector was located 50 m from the stop line. The membership functions *zero*, *a few* and *medium* of *APP* all grew wider and

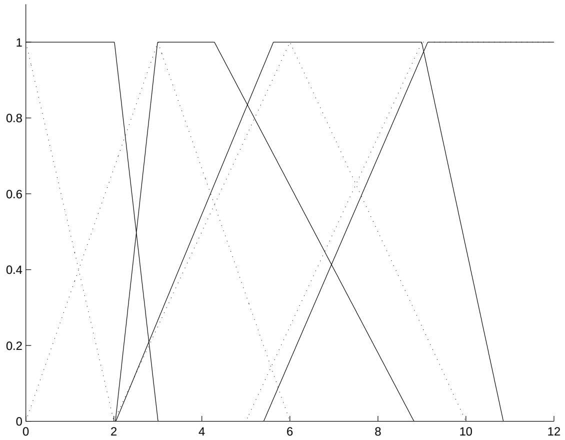

Fig. 12. Membership functions *zero*, *a few*, *medium* and *many* of *APP* before (dotted line) and after (solid line) the learning at a traffic volume of 1000 vehicles per hour. The location of the first traffic detector is 50 m from the stop line. Horizontal axis: number of approaching vehicles. Vertical axis: value of membership function.

moved rightward. This means that at large traffic volumes the "mean" values of *zero*, *a few* and *medium* are larger, which is quite easy to comprehend. The growth of *zero* means that input measurements of 0, 1 or 2 approaching vehicles are all interpreted as a fuzzy *zero* – this is intuitive, since

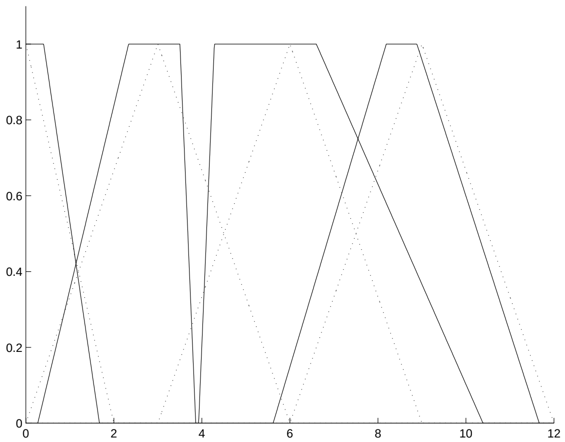

Fig. 13. Membership functions *zero*, *short*, *medium* and *long* of *EXT* before (dotted line) and after (solid line) the learning at a traffic volume of 500 vehicles per hour. The location of the first traffic detector is 100 m from the stop line. Horizontal axis: green signal extension in seconds. Vertical axis: value of membership function.

at a traffic volume of 1000 vehicles per hour the queue behind the red signal is typically so long that one or two vehicles in the green direction cannot be paid attention to.

The last example is the case of the membership functions of *EXT* at a traffic volume of 500 vehicles per hour but with the first traffic detector now located 100 m from the stop line. The membership functions are seen in Fig. 13. Similarly to Fig. 11, the membership functions *medium* and *long* of *EXT* overlap to a large degree, but now around 8–9 seconds instead of 4–5 seconds. This reflects the fact that the vehicles need now double the time to cover the 100 m (compared to 50 m) distance from the first traffic detector to the stop line.

6. Discussion

We have shown how a neural network can be used in fine-tuning the membership functions of a fuzzy traffic signal controller. The neural learning algorithm used was reinforcement learning [2] which gives credit for successful control actions and punishes for poor control actions.

Among several different neurofuzzy learning algorithms, this kind of reinforcement learning was chosen because of the nature of feedback information available. Some other reinforcement learning algorithms such as the one in [11] are capable of finding the rule base of the controller. In our case, the rule base was created using expert knowledge. Traffic signal control is an application, where expert knowledge can easily be included in the system, and there is no need to use more complicated algorithms. Also, some reinforcement algorithms [11] are capable of constructing a *multi-step* prediction of the future reinforcement v which is used when the success of a control action is not revealed until several time steps later. In the traffic control problem, the success of a control action (the delay) is revealed right after the action, so a multi-step prediction is not needed, and a single-step prediction v (Formula (2)) is enough.

Including a neural learning algorithm in fuzzy traffic signal control decreases the vehicular delay in simulation experiments. The simulations were run at several different traffic volumes and traffic detector locations. The new membership functions produce a 3–6% decrease in the vehicular delay, and the decrease in the delay is statistically significant. Different membership functions are found optimal at different traffic situations. The fuzzy traffic signal controller must thus identify the traffic volume and choose the proper membership functions accordingly.

The changes in the membership functions are quite intuitive. For example at large traffic volumes, the numerical values of the fuzzy concepts describing the number of approaching vehicles are larger than at small traffic volumes. Also, the membership functions take into account the distance from the first traffic detector to the stop line, such that vehicles between the detectors are given enough time to pass the stop line but longer green signal extensions are not given. The changes in the membership functions are often quite similar in traffic situations that share some common properties. All of these observations suggest that combining expert knowledge and neural learning could yield even better results: the expert may not initially come across the best membership functions but the neural learning may bring out some helpful details on how the membership functions should be modified.

The intersection configuration in our simulations was quite simple but suitable to demonstrate the potential of neural networks in the fine-tuning of the membership functions of a fuzzy traffic signal controller. In addition to the membership functions, the rule base of the fuzzy controller is, of course, very important. Fine-tuning the membership functions may not always be enough to solve the problems of the rule base, but it can indicate where the rule base is not optimal. After some modifications in the rule base, neural learning may again be used to fine-tune the system.

References

[1] A.G. Barto, R.S. Sutton, C.W. Anderson, Neuronlike adaptive elements that can solve difficult learning control problems, IEEE Transactions on Systems, Man and Cybernetics 13 (5) (1983) 834–846.

[2] H.R. Berenji, P. Khedkar, Learning and tuning fuzzy logic controllers through reinforcements, IEEE Transactions on Neural Networks 3 (5) (1992) 724–740.

[3] E. Bingham, Neurofuzzy traffic signal control, Master's thesis, Department of Technical Physics and Mathematics, Helsinki University of Technology, Finland, 1998. URL:http://www.hut.fi/~ella/dtyo.ps.zip/ (a PostScript file in .zip format).

[4] M. Brown, C.J. Harris, A perspective and critique of adaptive neurofuzzy systems used for modelling and control applications, International Journal of Neural Systems 6 (2) (1995).

[5] Y.-Y. Chen, K.-Z. Lin, S.-T. Hsu, A Self-Learning Fuzzy Controller, IEEE International Conference on Fuzzy Systems 1992, San Diego, CA, USA, March 1992.

[6] J.-S.R. Jang, ANFIS: Adaptive-network-based fuzzy inference system, IEEE Transactions on Systems, Man and Cybernetics 23 (3) (1993).

[7] J.-S.R. Jang, C.-T. Sun, Neuro-fuzzy modelling and control, Proceedings of the IEEE 83 (3) (1995).

[8] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall, Englewood Cliffs, 1995.

[9] B. Kosko, Fuzzy Engineering, Prentice-Hall, Englewood Cliffs, 1997.

[10] C.-T. Lin, C.S.G. Lee, Neural-network-based fuzzy logic control and decision system, IEEE Transactions on Computers 40 (12) (1991).

[11] C.-T. Lin, C.S.G. Lee, Reinforcement structure/parameter learning for neural-network-based fuzzy logic control systems, IEEE Transactions on Fuzzy Systems 2 (1) (1994) 46–63.

[12] J. Niittymäki, Isolated Traffic Signals – Vehicle Dynamics and Fuzzy Control, Publication 94, Transportation Engineering, Helsinki University of Technology, Finland, 1998.

[13] C. Pappis, E. Mamdani, A fuzzy logic controller for a traffic junction, IEEE Transactions on Systems, Man and Cybernetics 7 (10) (1977).

[14] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, in: D.E. Rumelhart, J.L. McClelland (Eds.), Parallel Distributed Processing. Explorations in the Microstructure of Cognition I, MIT Press, Cambridge, 1986.

[15] K.J. Sane, I. Kosonen, HUTSIM 4.2 Reference Manual, Publication 90, Transportation Engineering, Helsinki University of Technology, Finland, 1996.

[16] R.S. Sutton, A.G. Barto, R.J. Williams, Reinforcement learning is direct adaptive optimal control, IEEE Control Systems 12 (2) (1992).

[17] L.A. Zadeh, Fuzzy sets, Information and Control 8 (3) (1965) 338–353.