

An Intelligent Traffic Signal Control Based on EVALPSN

Kazumi Nakamatsu¹ and Jair Minoro Abe²

¹ University of Hyogo, Himeji, JAPAN
nakamatu@shse.u-hyogo.ac.jp

² Paulista University, Sao Paulo, BRAZIL
jairabe@uol.com.br

Abstract. We introduce two kinds of traffic signal control methods based on a paraconsistent annotated logic program called EVALPSN. One is for single traffic signal control and another one is for coordinated traffic signal control. We explain both of the methods with a simple example of intersections and introduce the simulation results of the two kinds of EVALPSN traffic signal control methods by comparing them with two conventional ones.

Keywords : EVALPSN(Extended Vector Annotated Logic Program with Strong Negation), traffic signal control, paraconsistent annotated logic program, defeasible deontic reasoning.

1 Introduction

We have proposed a paraconsistent annotated logic program called EVALPSN (Extended Vector Annotated Logic Program) that can deal with defeasible deontic reasoning and paraconsistency [9, 10]. Some applications of EVALPSN to various kinds of control and safety verification, robot action control, discrete event control, and safety verification for railway interlocking and air traffic control have been introduced [9, 12, 15]. Moreover, we have shown that some specific EVALPSNs can be easily implemented on microchips [13].

Traffic jams caused by inappropriate traffic signal control are serious environmental issues that we have to overcome. Conventionally, traffic signal control is divided into three kinds; single, coordinated and broad area ones according to the number of traffic signals to be controlled. In single traffic signal control, each traffic light is controlled independently. On the other hand, in coordinated ones, a series of traffic lights on the same route are controlled.

Recently various intelligent traffic signal control methods in which fuzzy logic, neural networks, GA(Genetic Algorithm), etc. are used for optimizing control have been proposed [1, 5, 6, 16]. Furthermore a traffic signal control method based on GA has been applied in a real scenario [4, 19, 20]. However, the method takes a long time to compute optimal solutions and is not so appropriate for real-time control. We have already proposed an intelligent real-time single traffic signal control system based on EVALPSN and have shown that it could reduce the traffic density 10% to 15% in simulation [14]. In this paper, we extend the idea

of EVALPSN single traffic signal control to a coordinated one and propose an EVALPSN based coordinated traffic signal control system.

This paper is organized in the following manner: firstly, we review EVALPSN briefly; next we introduce the idea of EVALPSN single traffic signal control with a simple example, how to formalize it in defeasible deontic formulas, and how to translate them into EVALPSN; subsequently, we formalize the coordinated traffic signal control for the example in defeasible deontic formulas, and translate them in EVALPSN similarly; last, we show the simulation results of the EVALPSN based coordinated traffic signal control system by comparing them with two conventional ones.

We assume that the reader is familiar with the basic knowledge of logic programming[7].

2 EVALPSN

First we will briefly review EVALPSN, the details of EVALPSN are found in [9–11]. Generally, a truth value called an *annotation* is explicitly attached to each literal in annotated logic programs [2]. For example, let p be a literal, μ an annotation, then $p:\mu$ is called an *annotated literal*. The set of annotations constitutes a complete lattice. An annotation in EVALPSN has a form of $[(i, j), \mu]$ called an *extended vector annotation*. The first component (i, j) is called a *vector annotation* and the set of vector annotations constitutes the complete lattice,

$$\mathcal{T}_v(n) = \{ (x, y) | 0 \leq x \leq n, 0 \leq y \leq n, x, y \text{ and } n \text{ are integers} \}$$

in Fig.1. The ordering ($\preceq_v$) of the lattice $\mathcal{T}_v(n)$ is defined as : let $(x_1, y_1), (x_2, y_2) \in \mathcal{T}_v(n)$,

$$(x_1, y_1) \preceq_v (x_2, y_2) \text{ iff } x_1 \leq x_2 \text{ and } y_1 \leq y_2.$$

For each extended vector annotated literal $p:[(i, j), \mu]$, the integer i denotes the amount of positive information to support the literal p and the integer j denotes that of negative one. The second component μ is an index of fact and deontic notions such as obligation, and the set of the second components constitutes the complete lattice,

$$\mathcal{T}_d = \{\perp, \alpha, \beta, \gamma, *1, *2, *3, \top\}.$$

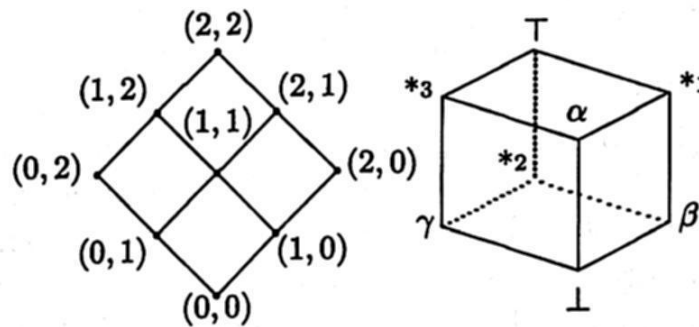


Fig. 1. Lattice $\mathcal{T}_v(2)$ and Lattice $\mathcal{T}_d$

The ordering ($\preceq_d$) of the lattice $\mathcal{T}_d$ is described by the Hasse's diagram in Fig.1. The intuitive meaning of each member of the lattice $\mathcal{T}_d$ is $\perp$ (unknown), α (fact), β (obligation), γ (non-obligation), $*_1$ (fact and obligation), $*_2$ (obligation and non-obligation), $*_3$ (fact and non-obligation), and $\top$ (inconsistency). Then the complete lattice $\mathcal{T}_e$ of extended vector annotations is defined as the product $\mathcal{T}_v(n) \times \mathcal{T}_d$. The ordering ($\preceq_e$) of the lattice $\mathcal{T}_e$ is defined as follows :
let $[(i_1, j_1), \mu_1]$ and $[(i_2, j_2), \mu_2] \in \mathcal{T}_e$,

$$[(i_1, j_1), \mu_1] \preceq_e [(i_2, j_2), \mu_2] \text{ iff } (i_1, j_1) \preceq_v (i_2, j_2) \text{ and } \mu_1 \preceq_d \mu_2.$$

There are two kinds of *epistemic negation* ($\neg_1$ and $\neg_2$) in EVALPSN, which are defined as mappings over $\mathcal{T}_v(n)$ and $\mathcal{T}_d$, respectively.

Definition 1(epistemic negations $\neg_1$ and $\neg_2$ in EVALPSN)

$$\begin{aligned} \neg_1([(i, j), \mu]) &= [(j, i), \mu], \quad \forall \mu \in \mathcal{T}_d \\ \neg_2([(i, j), \perp]) &= [(i, j), \perp], \quad \neg_2([(i, j), \alpha]) = [(i, j), \alpha], \\ \neg_2([(i, j), \beta]) &= [(i, j), \gamma], \quad \neg_2([(i, j), \gamma]) = [(i, j), \beta], \\ \neg_2([(i, j), *_1]) &= [(i, j), *_3], \quad \neg_2([(i, j), *_2]) = [(i, j), *_2], \\ \neg_2([(i, j), *_3]) &= [(i, j), *_1], \quad \neg_2([(i, j), \top]) = [(i, j), \top]. \end{aligned}$$

If we regard the epistemic negations as syntactical operations, the epistemic negations followed by literals can be eliminated by the syntactical operations. For example, $\neg_1 p : [(2, 0), \alpha] = p : [(0, 2), \alpha]$ and $\neg_2 q : [(1, 0), \beta] = p : [(1, 0), \gamma]$.

There is another negation called *strong negation* ($\sim$) in EVALPSN, and it is treated as classical negation.

Definition 2(strong negation $\sim$ in EVALSPN)

Let F be any formula and $\neg$ be $\neg_1$ or $\neg_2$.

$$\sim F =_{def} F \rightarrow ((F \rightarrow F) \wedge \neg(F \rightarrow F)).$$

Definition 3 (well extended vector annotated literal)

Let p be a literal. $p : [(i, 0), \mu]$ and $p : [(0, j), \mu]$ are called *weva*(*well extended vector annotated*)-literals, where $i, j \in \{1, 2, \dots, n\}$, and $\mu \in \{\alpha, \beta, \gamma\}$.

Defintion 4 (EVALPSN)

If $L_0, \dots, L_n$ are weva-literals,

$$L_1 \wedge \dots \wedge L_i \wedge \sim L_{i+1} \wedge \dots \wedge \sim L_n \rightarrow L_0$$

is called an *EVALPSN clause*. An *EVALPSN* is a finite set of EVALPSN clauses.

Fact and deontic notions, "obligation", "forbiddance" and "permission" are represented by extended vector annotations, $[(m, 0), \alpha]$, $[(m, 0), \beta]$, $[(0, m), \beta]$, and $[(0, m), \gamma]$, respectively, where m is a positive integer. For example,

$p : [(2, 0), \alpha]$ is intuitively interpreted as "it is true of strength 2 that p is a fact";

$p : [(1, 0), \beta]$ is as "it is true of strength 1 that p is obligatory";

$p : [(0, 2), \beta]$ is as "it is false of strength 2 that p is obligatory", that is to say, "it is true of strength 3 that p is forbidden";

$p : [(0, 1), \gamma]$ is as "it is false of strength 1 that p is not obligatory", that is to say, "it is true of strength 1 that p is permitted".

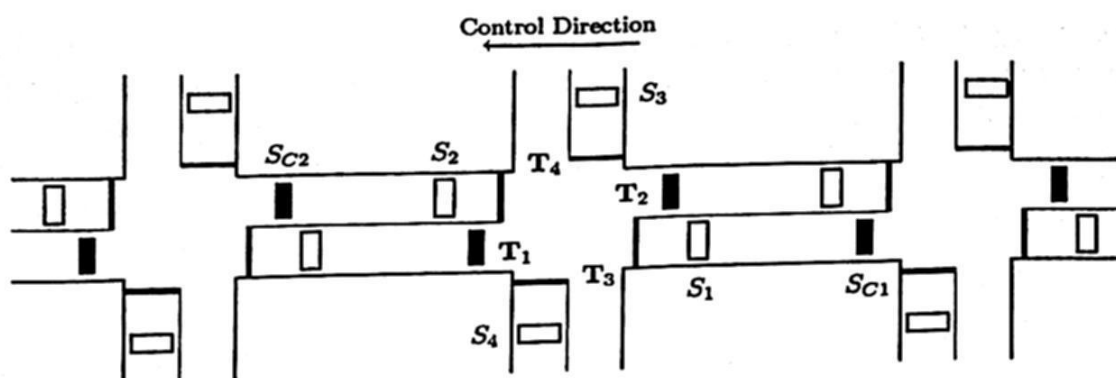


Fig. 2. Intersections in Japan

3 Traffic Signal Control in EVALPSN

First of all we suppose a basic traffic signal control policy in which traffic lights treating a larger amount of traffic should have the priority to be green until reducing traffic density. Based on this control policy the traffic signal is controlled in real-time by defeasible deontic reasoning in EVALPSN.

Suppose that you are waiting for the traffic light turning from red to green at an intersection. Then, you must have a demand for turning the traffic light from red to green in your mind. On the other hand, if you are driving through the intersection with green you must have a demand for keeping the traffic light green. The first demand can be regarded as permission for turning the light from red to green and the second one can be regarded as forbiddance from turning the traffic light from green to yellow or red. Then, there is a conflict between the permission and forbiddance. The basic idea of EVALPSN traffic signal control is to resolve such conflicts by EVALPSN defeasible deontic reasoning. We will formalize the traffic signal control in defeasible deontic formulas firstly and translate them into EVALPSN later.

3.1 Single Traffic Signal Control

We take a series of three typical intersections in Japan as described in Fig.2. In single traffic signal control, we focus on only the middle intersection among the three in Fig.2 as the object of the single traffic signal control. We assume that:

- the middle intersection has four traffic lights $T_{1,2,3,4}$, each one has three kinds of displays, green, yellow and red, and they have the following signal chart,

$$\begin{aligned} T_{1,2} &\rightarrow \text{red} \rightarrow \text{red} \rightarrow \text{all red} \rightarrow \text{green} \rightarrow \text{yellow} \rightarrow \text{all red} \rightarrow \\ T_{3,4} &\rightarrow \text{green} \rightarrow \text{yellow} \rightarrow \text{all red} \rightarrow \text{red} \rightarrow \text{red} \rightarrow \text{all red} \rightarrow \end{aligned}$$

- the intersection has four sensors $S_{1,2,3,4}$ to detect traffic density or flowing traffic, which are described as white boxes in Fig.2;
- yellow and all red terms are fixed and only green term is variably controlled according to the traffic detected by the sensors;

Table 1. Defeasible Reasoning

Superiority Relation	<i>a</i>	<i>b</i>	conclusion
R1 < R2	true	true	$\neg p$
R1 < R2	true	false	<i>p</i>
R1 < R2	false	true	$\neg p$
No relation	true	true	nothing
No relation	true	false	<i>p</i>
No relation	false	true	$\neg p$

- the minimum and maximum lengths of green term are given in advance, that is to say, green term must be controlled between the minimum and maximum terms.

For example, if the sensor S_1 detects flowing traffic, the green term of the traffic light T_1 should be extended within the range between the minimum and maximum terms. In the EVALPSN traffic signal control, the flowing traffic or traffic density of each road connected to the intersection are regarded as the forbiddance or permission for turning the object traffic light from green to yellow, and the conflict between the permission and forbiddance should be resolved by EVALPSN defeasible deontic reasoning. As the first step, we will formalize the traffic signal control in defeasible deontic formulas and translate them into EVALPSN.

Defeasible Deontic Rules for Single Traffic Signal Control Firstly, we introduce defeasible reasoning briefly. The details are found in [17, 18]. Basically, two kinds of rules, a strict rule $A \rightarrow p$ and a defeasible rule $B \Rightarrow q$ are used in defeasible logic [17], where A, B and p, q are called the prerequisites and consequents of the rules $A \rightarrow p$ and $B \Rightarrow q$, respectively. Defeasible reasoning is carried out based on the superiority relations between defeasible rules. Strict rules are used for deriving facts and superior to defeasible rules. Suppose that there are conflicting defeasible rules :

$$\mathbf{R1} \quad a \Rightarrow p \quad \text{and} \quad \mathbf{R2} \quad b \Rightarrow \neg p.$$

Then, we have the defeasible reasoning results in Tab.1. For example, if we take the first line. it shows that: if there is the superiority relation **R1 < R2**, and both a and b are true, then the rule **R2** defeats the rule **R1** and only the consequent $\neg p$ of the defeasible rule **R2** is derived.

Now we will start formalizing the single traffic signal control for the middle intersection in defeasible deontic formulas. We suppose that the traffic lights $T_{1,2}$ and $T_{3,4}$ are red and green, respectively. Then, the following conditions should be considered as strict or defeasible rules.

1. If one of the sensors $S_{1,2}$ detects traffic density more than the criterion at the time t , the permission for turning the traffic lights $T_{3,4}$ from green to yellow

should be derived. This condition is represented by the defeasible rule,

$$S_{1,2}^{rg}(t) \wedge T_{1,2}(r, t) \wedge T_{3,4}(g, t) \Rightarrow \neg \bigcirc \neg T_{3,4}(y, t), \quad (1)$$

where, $S_{1,2}^{rg}(t)$ denotes that one of the sensors $S_{1,2}$ detects the traffic density over the criterion at the time t ; $T_{1,2}(r, t)$ and $T_{3,4}(g, t)$ denote that the traffic lights $T_{1,2}$ and $T_{3,4}$ are red and green at the time t , respectively; and the symbol $\bigcirc$ is a modal operator to denote obligation, thus, the symbols $\neg \bigcirc \neg$ and $\bigcirc \neg$ denote permission and forbiddance, respectively.

2. On the other hand, if one of the sensors $S_{3,4}$ detects the flowing traffic over the criterion at the time t , the forbiddance from turning the traffic lights $T_{3,4}$ from green to yellow should be derived. Then, we have the defeasible rule conflicting with the defeasible rule (1),

$$S_{3,4}^{rg}(t) \wedge T_{1,2}(r, t) \wedge T_{3,4}(g, t) \Rightarrow \bigcirc \neg T_{3,4}(y, t). \quad (2)$$

We have to define the superiority relation ($<$) between the defeasible rules (1) and (2). As we assume that maintaining the current signal state is superior to turning it to the next signal state, the forbiddance from turning the traffic lights $T_{3,4}$ from green to yellow is prior to the permission for it, that is to say, the defeasible rule (2) is considered to be superior to the defeasible rule (1), i.e., (1) $<$ (2).

We need more traffic signal control rules. If we take only the defeasible rules (1) and (2) into account, we might have an extreme situation such that the green term of $T_{1,2}$ is 2 hours but that of $T_{3,4}$ is 1 minute. In order to avoid such an extreme control, we have to control green term in an appropriate range, which are represented by the following definite rules.

Minimum Green Term Rule Each green term of the traffic lights $T_{1,2,3,4}$ should be guaranteed its minimum length. Let $MIN_i(g, t)$, ($i \in \{1, 2, 3, 4\}$) denote that the green time of the traffic light T_i has not passed its minimum term at the time t . Then we have a rule : if the green time is shorter than its minimum term at the time t , it is forbidden from the traffic light turning from green to yellow. This rule can be represented by the strict rules,

$$MIN_{1,2}(g, t) \wedge T_{1,2}(g, t) \rightarrow \bigcirc \neg T_{1,2}(y, t), \quad (3)$$

$$MIN_{3,4}(g, t) \wedge T_{3,4}(g, t) \rightarrow \bigcirc \neg T_{3,4}(y, t). \quad (4)$$

Maximum Green Term Rule Each green term of the traffic lights $T_{1,2,3,4}$ also should have its maximum term. Let $MAX_i(g, t)$, ($i \in \{1, 2, 3, 4\}$) denotes that the green time of the traffic light T_i has already passed its maximum term at the time t . Then we have a rule : if the green time is longer than its maximum term at the time t , it is definitely permitted for the traffic light turning from green to yellow. This rule can be represented by the strict rules,

$$MAX_{1,2}(g, t) \wedge T_{1,2}(g, t) \rightarrow \neg \bigcirc \neg T_{1,2}(y, t), \quad (5)$$

$$MAX_{3,4}(g, t) \wedge T_{3,4}(g, t) \rightarrow \neg \bigcirc \neg T_{3,4}(y, t). \quad (6)$$

There are conflicting defeasible rules (1) and (2) whose consequents are permission and forbiddance. As a defeasible reasoning result, if the permission is derived at the time t , the traffic light has to be turned from green to yellow at the next time $t + 1$. On the other hand, if the forbiddance is defeasibly derived at the time t , the signal has to keep the current state at the next time $t + 1$. These conditions can be represented by the following strict rules :

$$T_{1,2}(g, t) \wedge \neg \bigcirc \neg T_{1,2}(y, t) \rightarrow \bigcirc T_{1,2}(y, t + 1), \quad (7)$$

$$T_{3,4}(g, t) \wedge \neg \bigcirc \neg T_{3,4}(y, t) \rightarrow \bigcirc T_{3,4}(y, t + 1), \quad (8)$$

$$T_{1,2}(g, t) \wedge \bigcirc \neg T_{1,2}(y, t) \rightarrow \bigcirc T_{1,2}(g, t + 1), \quad (9)$$

$$T_{3,4}(g, t) \wedge \bigcirc \neg T_{3,4}(y, t) \rightarrow \bigcirc T_{3,4}(g, t + 1). \quad (10)$$

Moreover, we need more strict rules for synchronizing all the traffic lights. For example, if it is obligatory that the traffic light T_1 is red at the time t , the traffic light T_2 also must be red at the same time, and if it is obligatory that the traffic light T_3 is green at the time t , the traffic light T_1 must be red at the same time. Such synchronization is represented by the following strict rules,

$$\bigcirc T_1(r, t) \rightarrow \bigcirc T_2(r, t) \quad \text{and} \quad \bigcirc T_3(g, t) \rightarrow \bigcirc T_1(r, t) \quad (11)$$

EVALPSN for Single Traffic Signal Control Now, we will translate the strict and defeasible rules into EVALPSN. The strict and defeasible rules can be translated by formalizing their semantics in EVALPSN. The details of the translation are described in [8]. The defeasible rules (1) and (2) are translated into the EVALPSN clauses,

$$\begin{aligned} & S^{rg}_{1,2}(t) : [(2, 0), \alpha] \wedge \\ & T_{1,2}(r, t) : [(2, 0), \alpha] \wedge T_{3,4}(g, t) : [(2, 0), \alpha] \wedge \\ & \sim MIN_{3,4}(g, t) : [(2, 0), \alpha] \wedge \\ & \sim S^{rg}_3(t) : [(2, 0), \alpha] \wedge \sim S^{rg}_4(t) : [(2, 0), \alpha] \\ & \rightarrow T_{3,4}(y, t) : [(0, 1), \gamma], \end{aligned} \quad (12)$$

$$\begin{aligned} & S^{rg}_{3,4}(t) : [(2, 0), \alpha] \wedge \\ & T_{1,2}(r, t) : [(2, 0), \alpha] \wedge T_{3,4}(g, t) : [(2, 0), \alpha] \wedge \\ & \sim MAX_{3,4}(g, t) : [(2, 0), \alpha] \rightarrow T_{3,4}(y, t) : [(0, 1), \beta]. \end{aligned} \quad (13)$$

The EVALPSN clause (12) represents that :

if the traffic sensors S_1 or S_2 have detected the traffic density over the criterion, the minimum green term has already passed, and the traffic sensors neither S_3 nor S_4 has detected the flowing traffic over the criteria at the time t under the condition that the traffic lights $T_{3,4}$ are green, then the traffic lights $T_{3,4}$ are permitted for turning to yellow.

On the other hand, the EVALPSN clause (13) represents that :

if the traffic sensors S_3 or S_4 have detected the flowing traffic over the criterion,
and
the maximum green term has not passed yet under the same condition that
the traffic lights $T_{3,4}$ are green,
then the traffic lights $T_{3,4}$ are forbidden for turning to yellow.

Similarly the strict rules (4), (6), (8) and (10), are also translated into the EVALP clauses,

$$MIN_{3,4}(g, t) : [(2, 0), \alpha] \wedge T_{3,4}(g, t) : [(2, 0), \alpha] \rightarrow T_{3,4}(y, t) : [(0, 2), \beta], \quad (14)$$

$$MAX_{3,4}(g, t) : [(2, 0), \alpha] \wedge T_{3,4}(g, t) : [(2, 0), \alpha] \rightarrow T_{3,4}(y, t) : [(0, 2), \gamma], \quad (15)$$

$$T_{3,4}(g, t) : [(2, 0), \alpha] \wedge T_{3,4}(y, t) : [(0, 1), \gamma] \rightarrow T_{3,4}(y, t+1) : [(2, 0), \beta], \quad (16)$$

$$T_{3,4}(g, t) : [(2, 0), \alpha] \wedge T_{3,4}(y, t) : [(0, 1), \beta] \rightarrow T_{3,4}(g, t+1) : [(2, 0), \beta]. \quad (17)$$

Example 1

Suppose that the traffic lights $T_{1,2}$ are red and the traffic lights $T_{3,4}$ are green, furthermore, it has already passed the minimum green term then.

Case-1. If one of the sensors $S_{1,2}$ detects the traffic density over the criterion at the time t and the rest of the sensors do not react then, the EVALPSN clause (12) is fired and the permission $T_{3,4}(y, t) : [(0, 1), \gamma]$ is derived, furthermore, the EVALPSN clause (16) is also fired and the obligation $T_{3,4}(y, t+1) : [(2, 0), \beta]$ is also derived.

Case-2. If both the sensors S_1 and S_3 detect the traffic density and flowing traffic over the criteria at the time t , respectively, and the rest of the sensors do not detect then, the EVALPSN clause (13) is fired and the forbiddance $T_{3,4}(y, t) : [(0, 1), \beta]$ is derived, furthermore, the EVALPSN clause (17) is also fired and the obligation $T_{3,4}(g, t+1) : [(2, 0), \beta]$ is also derived.

3.2 Coordinated Traffic Signal Control

We exhibit the basic idea of the EVALPSN coordinated traffic signal control by taking the same intersections in Fig.2 as an example. Here we focus on only the traffic from the right to the left and its converse direction as the object of the EVALPSN coordinated traffic signal control. In conventional coordinated traffic signal control, not only each term of green, yellow and red but also a timelag called an *off set* between two neighbor traffic lights are controlled, and they should be pre-installed. On the other hand, in EVALPSN coordinated traffic signal control, only the green term of each coordinated traffic light should be controlled by EVALPSN defeasible deontic reasoning based on traffic amount information, which is detected by traffic sensors attached to not only the object intersection but also its both of neighbor intersections. Therefore, when we consider the EVALPSN coordinated traffic signal control in terms of the middle intersection, the extra traffic sensors $Sc_{1,2}$ described by black boxes in Fig.2 attached to the right and left intersections should be taken into account to detect the traffic flowing into the middle intersection. Moreover, we note that when the

traffic information detected by the remote sensors $Sc_{1,2}$ is processed, appropriate timelags between the detection and processing of the traffic information should be taken into account according to the distances between the object intersection and the sensors $Sc_{1,2}$.

Defeasible Deontic Rules for Coordinated Traffic Signal Control We assume the same condition as the single traffic signal control, that is to say the traffic lights $T_{1,2}$ and $T_{3,4}$ are red and green, respectively. Then, we have the following conditions in defeasible deontic formulas.

3. If one of the sensors Sc_1 or Sc_2 detects flowing traffic over the criteria at the time t , the permission for turning the traffic lights $T_{3,4}$ from green to yellow should be derived. Then, we have the defeasible rule,

$$Sc_{1,2}^{rg}(t) \wedge T_{1,2}(r, t) \wedge T_{3,4}(g, t) \Rightarrow \neg \bigcirc \neg T_{3,4}(y, t), \quad (18)$$

where $Sc_{1,2}^{rg}(t)$ denotes that one of the sensors $Sc_{1,2}$ detects the flowing traffic over the criterion.

We also need to consider the superiority relation between the conflicting defeasible rules (2) and (18). As the coordinately controlled route traffic from the right to the left or its converse are regarded to be superior to other traffic, the permission for turning the traffic lights $T_{3,4}$ from green to yellow has a prior to the forbiddance from it, that is to say, the defeasible rule (18) is considered to be superior to the defeasible rule (2), i.e., $(2) < (18)$.

EVALPSN for Coordinated Traffic Signal Control Taking the superiority relations among the defeasible rules (1), (2) and (18), i.e., $(1) < (2) < (18)$ into account, those defeasible rules are retranslated into the EVALPSN clauses,

$$\begin{aligned} & S^{rg}_{1,2}(t) : [(2, 0), \alpha] \wedge \\ & T_{1,2}(r, t) : [(2, 0), \alpha] \wedge T_{3,4}(g, t) : [(2, 0), \alpha] \wedge \\ & \sim MIN_{3,4}(g, t) : [(2, 0), \alpha] \wedge \\ & \sim S^{rg}_3(t) : [(2, 0), \alpha] \wedge \sim S^{rg}_4(t) : [(2, 0), \alpha] \\ & \rightarrow T_{3,4}(y, t) : [(0, 1), \gamma], \end{aligned} \quad (19)$$

$$\begin{aligned} & S^{rg}_{3,4}(t) : [(2, 0), \alpha] \wedge \\ & T_{1,2}(r, t) : [(2, 0), \alpha] \wedge T_{3,4}(g, t) : [(2, 0), \alpha] \wedge \\ & \sim MAX_{3,4}(g, t) : [(2, 0), \alpha] \wedge \\ & \sim Sc^{rg}_1(t) : [(2, 0), \alpha] \wedge \sim Sc^{rg}_2(t) : [(2, 0), \alpha] \\ & \rightarrow T_{3,4}(y, t) : [(0, 1), \beta], \end{aligned} \quad (20)$$

$$\begin{aligned} & Sc^{rg}_{1,2}(t) : [(2, 0), \alpha] \wedge \\ & T_{1,2}(r, t) : [(2, 0), \alpha] \wedge T_{3,4}(g, t) : [(2, 0), \alpha] \wedge \\ & \sim MIN_{3,4}(g, t) : [(2, 0), \alpha] \rightarrow T_{3,4}(y, t) : [(0, 1), \gamma], \end{aligned} \quad (21)$$

As well as the EVALPSN single traffic signal control the coordinated one also has Minimum and Maximum Green Term Rules, therefore, we also have the EVALPSN clauses (14), (15), (16), (17) as the translation of those rules.

Example 2

Suppose the same conditions as Example 1.

Case-3. If one of the sensors $Sc_{1,2}$ and the sensor S_4 detect the flowing traffic over the criteria at the time t , respectively, the EVALPSN clause (21) is fired and the permission $T_{3,4}(y, t) : [(0, 1), \gamma]$ is derived, furthermore, the EVALPSN clause (16) is also fired and the obligation $T_{3,4}(y, t + 1) : [(2, 0), \beta]$ is also derived.

Case-4. If all the sensors $S_{1,2,3,4}$ and $Sc_{1,2}$ detect the flowing traffic or traffic density over the criteria at the time t , respectively, the EVALPSN clause (21) is fired and the permission $T_{3,4}(y, t) : [(0, 1), \gamma]$ is derived, furthermore, the EVALPSN clause (16) is also fired and the obligation $T_{3,4}(y, t + 1) : [(2, 0), \beta]$ is also derived.

Case-5. If the sensors $S_{1,2}$ and $Sc_{1,2}$ detect the traffic density and flowing traffic over the criteria at the time t , respectively, both the EVALPSN clauses (19) and (21) are fired and the same permission $T_{3,4}(y, t) : [(0, 1), \gamma]$ is derived, furthermore, the EVALPSN clause (16) is also fired and the obligation $T_{3,4}(y, t + 1) : [(2, 0), \beta]$ is also derived.

4 Simulation

We now present the simulation results of the four kinds of traffic signal control, conventional (fixed time) single, coordinated, EVALPSN single, and EVALPSN coordinated ones for the intersections in Fig.2 by the cellular automaton simulation method. We have assumed the following conditions for simulation,

- the unit time called "step" is defined in the simulation system, which is the time that a car travels one cell to the next cell;
- 5000 steps/hour;
- each distance between the three intersections is the same 20 cells;
- cars are flowing into all the intersections from each road in 10% probability except for the traffic from the right road, which is 15%, where 15% probability means 15 cars/100 steps appear at the road;
- for fixed-time traffic signal control, the green term is 60 steps, the yellow term is 3 steps, the off-set in the coordinated traffic signal control is 24 steps;
- for EVALPSN traffic signal control, the green term is between 14(min) and 60(max) steps, the yellow term is fixed 5 steps.

We took the simulation data of 50 cars that were chosen at random among all the cars flowing into the intersections from the right and traveling to the left. All four kinds of simulation have been carried out during 5000 steps each and repeated ten times. The simulation data in Tab.2 show the average values of the ten times simulation results, where

“total car” denotes the total number of cars that flowed into all the three intersections from any road ;

“stop step” denotes the total number of steps that the 50 cars stopped ;

“travel step” denotes the total number of steps for the 50 cars having traveled from the right to the left ;

“travel car” denotes the total number of cars that traveled from the right up to the left.

Table 2. Simulation Results

	total car	stop step	travel step	travel car
Fixed-time Single	3804	66	177	495
Fixed-time Coordinated	3823	65	173	500
EVALPSN Single	4149	43	152	577
EVALPSN Coordinated	4169	34	122	607

This simulation results clearly say that : EVALPSN traffic signal control shows more efficient results than a conventional fixed time one in both single and coordinated traffic signal control methods, especially EVALPSN coordinated control is most efficient among them.

5 Conclusion and Future Work

We have introduced two kinds of traffic signal control methods for single and coordinated traffic signal controls as an application of EVALPSN with their computer simulation results by the cellular automaton method. Then we obtained the following conclusion : EVALPSN traffic control is a sensor based real-time one and more efficient than conventional ones according to the simulation results ; however, if it is implemented practically, lots of sensor installation are required and too much cost; EVALPSN control can be implemented in both existent software and hardware such as PLC(Programmable Logic Controller), although it has not been addressed in this paper; since each traffic light is controlled independently as one autonomous agent in the EVALPSN coordinated traffic signal control, it is more fault tolerant than conventional coordinated one.

We are planning to apply the EVALPSN traffic signal control methods to network control, for example, network routing control, electric power supply network control etc. in our future work.

References

1. Abdulhai,B.: Reinforcement Learning for the True Adaptive Traffic Signal Control. J.Transportation Engineering **129** (2003) 278–285
2. Blair,H.A., Subrahmanian,V.S.: Paraconsistent Logic Programming. Theoretical Computer Science **68** (1989) 135–154

3. Da Costa, N.C.A., Subrahmanian, V.S., Vago, C.: The Paraconsistent Logics PT. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* **37** (1989) 139–148
4. Foy, M.D., Benekohal, R.F., Goldberg, D.E.: Signal Timing Determination Using Genetic Algorithms. *Transportation Research Record* **1365** (1993) 108–105
5. Girianna, M., Benekohal, R.F.: Application of Genetic Algorithms to Generate Optimum Signal Coordination for Congested Networks. *Proc. 7th Int'l Conf. Applications of Advanced Technology in Transportation*, ASCE (2002) 762–769
6. Li, R., Li, J., Lu, H.: Multi-Layer Traffic Signal Control Model Based on Fuzzy Control and Genetic Algorithm. In *Proc. 9th Int'l Conf. Applications of Advanced Technology in Transportation*, ASCE (2006) 461–466
7. Lloyd, J.: *Foundations of Logic Programming* (2nd Edition). Springer-Verlag (1987)
8. Nakamatsu, K.: On the Relation Between Vector Annotated Logic Programs and Defeasible Theories. *Logic and Logical Philosophy* **8** (2001) 181–205
9. Nakamatsu, K.: Intelligent Information Systems Based on Paraconsistent Logic Programs. In *Innovations in Intelligent Systems and Applications. Studies in Fuzziness and Soft Computing Series 140*, Springer-Verlag (2004) 257–283
10. Nakamatsu, K., Abe, J.M., Suzuki, A.: Annotated Semantics for Defeasible Deontic Reasoning. In *Proc. the Second International Conference on Rough Sets and Current Trends in Computing*, LNAI **2005**, Springer-Verlag (2001) 470–478
11. Nakamatsu, K., Abe, J.M., Suzuki, A.: A Defeasible Deontic Reasoning System Based on Annotated logic Programming. In *Computing Anticipatory Systems CASYS2000*, AIP Conference Proceedings **573**, AIP Press (2001) 467–478
12. Nakamatsu, K., Chung, S-L., Komaba, H., Suzuki, A.: A Discrete Event Control Based on EVALPSN Stable Model Computation. In *Rough Set, Fuzzy Set, Data Mining and Granular Computing*, LNAI **3641**, Springer-Verlag (2005) 671–681
13. Nakamatsu, K., Mita, Y., Shibata, T.: An Intelligent Action Control System Based on Extended Vector Annotated Logic Program and its Hardware Implementation. *Intelligent Automation and Soft Computing* **13** (2007) 289–304
14. Nakamatsu, K., Seno, T., Abe, J.M., Suzuki, A.: Intelligent Real-time Traffic Signal Control Based on a Paraconsistent Logic Program EVALPSN. *Rough Set, Fuzzy Set, Data Mining and Granular Computing*, LNAI **2639** Springer-Verlag (2003) 719–723
15. Nakamatsu, K., Suito, H., Abe, J.M., Suzuki, A.: Paraconsistent Logic Program Based Safety Verification for Air Traffic Control. In *Proc. 2002 IEEE Int'l Conf. Systems, Man and Cybernetics*, IEEE (2002) CD-ROM
16. Nishikawa, I., Iritani, T., Sakakibara, K.: Improvement of the Traffic Signal Control by Complex-Valued Hopfield Networks. In *Proc. Int'l Joint Conf. Neural Networks 2006*, IEEE (2006) 1186–1191
17. Nute, D.: Basic Defeasible Logic. In *Intensional Logics for Programming*, Oxford Science Publications (1992) 126–154
18. Nute, D.: Apparent Obligation. In *Defeasible Deontic Logic*, Kluwer Academic Publishers (1997) 287–316
19. Park, B., Messer, C.J., Urbanik II, T.: Traffic Signal Optimization Program for Oversaturated Conditions: A Genetic Algorithm Approach. *Transportation Research Record* **1683** (1999) 133–142
20. Park, B., Messer, C.J., Urbanik II, T.: Enhanced Genetic Algorithms for Signal Timing Optimization of Oversaturated Intersections. *Transportation Research Record* **1727** (2000) 32–41