SAT-Based Bounded Model Checking for Weighted Interpreted Systems and Weighted Linear Temporal Logic*

Bożena Woźna-Szcześniak, Agnieszka M. Zbrzezny, and Andrzej Zbrzezny

IMCS, Jan Długosz University
Al. Armii Krajowej 13/15, 42-200 Częstochowa, Poland
{b.wozna,a.zbrzezny,agnieszka.zbrzezny}@ajd.czest.pl

Abstract. We present a SAT-based bounded model checking (BMC) method for the weighted interpreted systems (i.e. interpreted systems augmented to include a weight function, one per each agent, that associates *weights* with actions, which are arbitrary natural numbers) and for properties expressible in the existential fragment of a weighted linear temporal logic with epistemic components (WELTLK). Since in BMC we translate both the system model and the checked specification to a propositional formula that is later analysed by a SAT-solver, we report on a propositional encoding of both the weighted interpreted systems and the WELTLK formulae. This encoding is designed specifically for managing weighted temporal operators and knowledge operators, which are commonly found in properties of multi-agent systems in models of which we assume that acting of agents may cost. We implemented the proposed BMC algorithm as a new module of VerICS, and we evaluated it by means of the following two examples: a weighted generic pipeline paradigm and a weighted bits transmission problem.

1 Introduction

Agents are autonomous and intelligent entities that can be engage in social activities such as coordination, negotiation, cooperation, etc. A multi-agent system (MAS) [16] is a distributed system composed of multiple interacting agents within an environment. There are variety of models of MASs, the most widely studied of which is the *interpreted system* (IS) [7], designed for reasoning about the agents' epistemic and temporal properties, and the *deontic interpreted system* (DIS) [12] that extends IS to make possible reasoning about correct functioning behaviour of MASs. An important assumption in this line of models is that there are no costs associated to agents' actions. The models become more expressive when this restriction is dropped. For example, the formalism of *weighted deontic interpreted systems* (WDISs) [17] extends DISs to make the reasoning possible about not only temporal, epistemic and deontic properties, but also about agents

* Partly supported by National Science Center under the grant No. 2011/01/B/ST6/05317.

G. Boella et al. (Eds.): PRIMA 2013, LNAI 8291, pp. 355–371, 2013.

quantitative properties. In the Kripke model of WDIS each transition is labelled not only by a joint action, but also by a positive integer value (e.g. 100 units) that represents the cost of acting agents. Such transitions could be simulated in the Kripke model of DIS by inserting 99 intermediate states. However, this increases the size of the Kripke model, and so it makes the model checking process more difficult.

Bounded model checking (BMC) [3,14] is one of the symbolic model checking techniques [4,16] that has gained popularity due to the immense success of SAT-solvers. In classical SAT-based BMC, we translate the existential model checking problem for a temporal epistemic logic to the satisfiability problem of a propositional formula. More precisely, we represent a counterexample of the bounded length by a propositional formula, and we check the satisfiability of the resulting propositional formula with a specialised SAT-solver. If this formula is satisfiable, then the SAT-solver returns a satisfying assignment that can be converted into a concrete counterexample showing the source of an error. Otherwise, the bound is increased until an error is found, or a pre-determined completeness threshold is reached (in practice, this is a rare case), or a pre-determined time/memory limits are reached.

Specification languages are most useful when they can be verified automatically. Therefore to model check the requirements of MASs various extensions of temporal logics [5] with epistemic (representing knowledge) [7], doxastic (representing beliefs) [10], and deontic (representing the distinction between ideal/correct behaviour and actual – possibly incorrect – behaviour of the agents) [12] components have been proposed. In this paper we aim at completing the picture of applying the SAT-based BMC techniques to MASs by looking at the existential fragment of the weighted LTLK (i.e. weighted LTL extended with epistemic components, called WLTLK), interpreted over the *weighted interpreted systems* (WISs), i.e. the WDISs formalism in which deontic properties cannot be expressed [17]. We restrict the presented BMC formalism to WISs, because adding the deontic modalities to the BMC method for the existential fragment of WLTLK that we present in the paper is straightforward.

The original contributions of the paper are as follows. First of all, we introduce the WLTLK language and its existential fragment, called WELTLK. In the second place, we propose a SAT-based BMC technique for WISs and for WELTLK. Finally, we report on the implementation of the proposed BMC method as a new module of VerICS [9], and evaluate it experimentally by means of a modified *generic pipeline paradigm* [13] and a modified *bit transmission problem* [1]. We would like to point out that to the best of our knowledge, this is the first work which provides a practical BMC algorithm for WELTLK interpreted over weighted interpreted systems. Moreover, the novelty with respect to [17] is the following: the language (WELTLK), a propositional encoding of WELTLK, a new propositional encoding of the weighted transition relation, an implementation, an experimental evaluation, and a new case study. Further, we do not compare our results with other model checkers for MASs, e.g. MCMAS [11] or MCK [8], simply because they do not support WELTLK and WIS.

The structure of the paper is as follows. In Section 2 we introduce WISs and WLTLK together with its existential fragment. In Section 3 we define a SAT-based BMC for WELTLK interpreted over WISs. In Section 4 we discuss our experimental results. In Section 5 we conclude the paper.

2 Preliminaries

Weighted Interpreted Systems. Let $Ag = \{1, \ldots, n\}$ denote the non-empty and finite set of agents, and $\mathcal{E}$ be a special agent that is used to model the environment in which the agents operate. The set of agents Ag together with the environment $\mathcal{E}$ constitute a multi-agent system (MAS). In the paper we use the weighted interpreted system (WIS), a formalism defined later on in the section, to model MAS. In the WIS formalism, each agent $\mathbf{c} \in Ag \cup \{\mathcal{E}\}$ is modelled using a non-empty set $L_\mathbf{c}$ of *local states*, a non-empty set $\iota_\mathbf{c} \subseteq L_\mathbf{c}$ of initial states, a non-empty set $Act_\mathbf{c}$ of *possible actions*, a *protocol function* $P_\mathbf{c} : L_\mathbf{c} \to 2^{Act_\mathbf{c}}$ that define rules according to which actions may be performed in each local state, a (partial) *evolution function* $t_\mathbf{c} : L_\mathbf{c} \times Act \to L_\mathbf{c}$ with $Act = Act_1 \times \cdots \times Act_n \times Act_\mathcal{E}$ (each element of Act is called a *joint action*), a *weight function* $d_\mathbf{c} : Act_\mathbf{c} \to \mathbb{N}$, and a *valuation function* $\mathcal{V}_\mathbf{c} : L_\mathbf{c} \to 2^{\mathcal{PV}}$ that assigns to each local state a set of propositional variables that are assumed to be true at that state. Further, we do not assume that the sets $Act_\mathbf{c}$ are disjoint for all $\mathbf{c} \in Ag \cup \{\mathcal{E}\}$.

Now for a given set of agents Ag, the environment $\mathcal{E}$ and a set of propositional variables $\mathcal{PV}$, we define the *weighted interpreted system* (WIS) as a tuple $(\{\iota_\mathbf{c}, L_\mathbf{c}, Act_\mathbf{c}, P_\mathbf{c}, t_\mathbf{c}, \mathcal{V}_\mathbf{c}, d_\mathbf{c}, \}_{\mathbf{c} \in Ag \cup \{\mathcal{E}\}})$. Next, for a given WIS we define: (1) a set of all *possible global states* $S = L_1 \times \ldots \times L_n \times L_\mathcal{E}$; by $l_\mathbf{c}(s)$ we denote the local component of agent $\mathbf{c} \in Ag \cup \{\mathcal{E}\}$ in a global state $s = (\ell_1, \ldots, \ell_n, \ell_\mathcal{E})$; and (2) a *global evolution function* $t : S \times Act \to S$ as follows: $t(s, a) = s'$ iff for all $\mathbf{c} \in Ag$, $t_\mathbf{c}(l_\mathbf{c}(s), a) = l_\mathbf{c}(s')$ and $t_\mathcal{E}(l_\mathcal{E}(s), a) = l_\mathcal{E}(s')$. In brief we write the above as $s \xrightarrow{a} s'$. Now, for a given WIS we define a *weighted model* (or *a model*) as a tuple $M = (\iota, S, T, \mathcal{V}, d)$, where

- $\iota = \iota_1 \times \ldots \times \iota_n \times \iota_\mathcal{E}$ is the set of all possible initial global state;
- S is the set of all possible global states as defined above;
- $T \subseteq S \times Act \times S$ is a transition relation defined by the global evolution function as follows: $(s, a, s') \in T$ iff $s \xrightarrow{a} s'$. We assume that the relation T is total, i.e. for any $s \in S$ there exists $s' \in S$ and a non empty joint action $a \in Act$ such that $s \xrightarrow{a} s'$;
- $\mathcal{V} : S \to 2^{\mathcal{PV}}$ is the valuation function defined as $\mathcal{V}(s) = \bigcup_{\mathbf{c} \in Ag \cup \{\mathcal{E}\}} \mathcal{V}_\mathbf{c}(l_\mathbf{c}(s))$;
- $d : Act \to \mathbb{N}$ is a "joint" weight function defined as follows: $d((a_1, \ldots, a_n, a_\mathcal{E})) = \sum_{\mathbf{c} \in Ag \cup \{\mathcal{E}\}} d_\mathbf{c}(a_\mathbf{c})$; note that this definition is reasonable, since we are interested in MASs, in which transitions carry some cost.

Given a WIS one can define the indistinguishability relation $\sim_\mathbf{c} \subseteq S \times S$ for agent $\mathbf{c}$ as follows: $s \sim_\mathbf{c} s'$ iff $l_\mathbf{c}(s') = l_\mathbf{c}(s)$. Further, a *path* in M is an infinite sequence $\pi = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} \ldots$ of transitions. For such a path, and for $j \leq m \in \mathbb{N}$, by $\pi(m)$ we denote the m-th state s_m, by π^m we denote the m-th suffix of the path π, which is defined in the standard way: $\pi^m = s_m \xrightarrow{a_{m+1}} s_{m+1} \xrightarrow{a_{m+2}}$

$s_{m+2}\ldots$. Next, by $\pi[j..m]$ we denote the finite sequence $s_j \xrightarrow{a_{j+1}} s_{j+1} \xrightarrow{a_{j+2}} \ldots s_m$ with $m-j$ transitions and $m-j+1$ states, and by $D\pi[j..m]$ we denote the (cumulative) weight of $\pi[j..m]$ that is defined as $d(a_{j+1})+\ldots+d(a_m)$ (hence 0 when $j=m$). By $\Pi(s)$ we denote the set of all the paths starting at $s \in S$, and by $\Pi = \bigcup_{s^0\in\iota} \Pi(s^0)$ we denote the set of all the paths starting at initial states.

The Logic WLTLK and Its Existential Fragment. WLTLK extends LTL with cost constraints on temporal modalities and with epistemic modalities. In the syntax of WLTLK we assume the following: $p \in \mathcal{PV}$ is an atomic proposition, $\mathbf{c} \in Ag$, $\Gamma \subseteq Ag$, I is an interval in $\mathbb{N} = \{0,1,2,\ldots\}$ of the form: $[a,b)$ and $[a,\infty)$, for $a,b \in \mathbb{N}$ and $a \neq b$, and $right(I) = b$ if $I = [a,b)$, otherwise $right(I) = \infty$. In the semantics we assume the following definitions of epistemic relations: $\sim_\Gamma^E \stackrel{def}{=} \bigcup_{\mathbf{c}\in\Gamma} \sim_\mathbf{c}$, $\sim_\Gamma^C \stackrel{def}{=} (\sim_\Gamma^E)^+$ (the transitive closure of $\sim_\Gamma^E$), $\sim_\Gamma^D \stackrel{def}{=} \bigcap_{\mathbf{c}\in\Gamma} \sim_\mathbf{c}$, where $\Gamma \subseteq Ag$.

The WLTLK formulae in the negation normal form are defined by the following grammar:

$$\varphi ::= \mathbf{true} \mid \mathbf{false} \mid p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathrm{X}_I\varphi \mid \varphi \mathrm{U}_I \varphi \mid \varphi \mathrm{R}_I \varphi$$
$$\mid \mathrm{K}_\mathbf{c}\varphi \mid \overline{\mathrm{K}}_\mathbf{c}\varphi \mid \mathrm{E}_\Gamma\varphi \mid \overline{\mathrm{E}}_\Gamma\varphi \mid \mathrm{D}_\Gamma\varphi \mid \overline{\mathrm{D}}_\Gamma\varphi \mid \mathrm{C}_\Gamma\varphi \mid \overline{\mathrm{C}}_\Gamma\varphi$$

The temporal modalities X_I, U_I and R_I are, respectively, named as the *weighted next step*, the *weighted until* and the *weighted release*. The derived basic temporal modalities for *weighted eventually* and *weighted globally* are defined as follows: $\mathrm{F}_I\varphi \stackrel{def}{=} \mathbf{true}\mathrm{U}_I\varphi$ and $\mathrm{G}_I\varphi \stackrel{def}{=} \mathbf{false}\mathrm{R}_I\varphi$. Hereafter, if the interval I is of the form $[0,\infty)$, then we omit it for the simplicity of the presentation.

The epistemic modality $K_\mathbf{c}\varphi$ represents "agent $\mathbf{c}$ knows φ" while the modality $\overline{\mathrm{K}}_\mathbf{c}\varphi \stackrel{def}{=} \neg K_\mathbf{c}\neg\varphi$ is the corresponding dual one representing "agent $\mathbf{c}$ considers φ possible". The epistemic modalities D_Γ, E_Γ, and C_Γ represent distributed knowledge in the group Γ, "everyone in Γ knows", and common knowledge among agents in Γ, respectively. The epistemic modalities $\overline{\mathrm{D}}_\Gamma$, $\overline{\mathrm{E}}_\Gamma$, and $\overline{\mathrm{C}}_\Gamma$ are the corresponding dual ones. The WELTLK is the existential fragment of WLTLK, defined by the following grammar:

$$\varphi ::= \mathbf{true} \mid \mathbf{false} \mid p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathrm{X}_I\varphi \mid \varphi \mathrm{U}_I \varphi \mid \varphi \mathrm{R}_I \varphi$$
$$\mid \overline{\mathrm{K}}_\mathbf{c}\varphi \mid \overline{\mathrm{E}}_\Gamma\varphi \mid \overline{\mathrm{D}}_\Gamma\varphi \mid \overline{\mathrm{C}}_\Gamma\varphi.$$

A WLTLK formula φ is true (valid) along the path π (in symbols $M,\pi \models \varphi$) iff $M,\pi^0 \models \varphi$, where the satisfaction relation $\models$ is defined inductively, with the classical rules for propositional operators and with the following rules for the temporal and epistemic modalities:

$M,\pi^m \models \mathrm{X}_I\alpha$ iff $D\pi[m..m+1] \in I$ and $M,\pi^{m+1} \models \alpha$,
$M,\pi^m \models \alpha \mathrm{U}_I \beta$ iff $(\exists i \geq m)(D\pi[m..i] \in I$ and $M,\pi^i \models \beta$ and $(\forall m \leq j < i) M,\pi^j \models \alpha)$,
$M,\pi^m \models \alpha \mathrm{R}_I \beta$ iff $(\forall i \geq m)(D\pi[m..i] \in I$ implies $M,\pi^i \models \beta)$ or $(\exists i \geq m)$ $(D\pi[m..i] \in I$ and $M,\pi^i \models \alpha$ and $(\forall m \leq j \leq i) M,\pi^j \models \beta)$,
$M,\pi^m \models \mathrm{K}_\mathbf{c}\alpha$ iff $(\forall \pi' \in \Pi)(\forall i \geq 0)(\pi'(i) \sim_\mathbf{c} \pi(m)$ implies $M,\pi'^i \models \alpha)$,
$M,\pi^m \models \overline{\mathrm{K}}_\mathbf{c}\alpha$ iff $(\exists \pi' \in \Pi)(\exists i \geq 0)(\pi'(i) \sim_\mathbf{c} \pi(m)$ and $M,\pi'^i \models \alpha)$,
$M,\pi^m \models Y_\Gamma\alpha$ iff $(\forall \pi' \in \Pi)(\forall i \geq 0)(\pi'(i) \sim_\Gamma^Y \pi(m)$ implies $M,\pi'^i \models \alpha)$,

$M, \pi^m \models \overline{Y}_\Gamma \alpha$ iff $(\exists \pi' \in \Pi)(\exists i \geq 0)(\pi'(i) \sim_\Gamma^Y \pi(m)$ and $M, \pi'^i \models \alpha)$, where $Y \in \{\mathrm{D}, \mathrm{E}, \mathrm{C}\}$.

A WLTLK formula φ *existentially holds* in the model M (denoted $M \models \varphi$) iff $M, \pi \models \varphi$ for some path $\pi \in \Pi$. The *existential model checking problem* asks whether $M \models \varphi$.

3 Bounded Model Checking for WIS and for WELTLK

As usual we begin with defining the notion of k-paths and loops, which are required by the bounded semantics - the basis of each SAT-based BMC.

Let M be a model, and $k \in \mathbb{N}$ a bound. A *k-path* π_l is a pair (π, l), where π is a finite sequence $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_k} s_k$ of transitions. A k-path π_l is a *loop* if $l < k$ and $\pi(k) = \pi(l)$. Note that if a k-path π_l is a loop, then it represents the infinite path of the form uv^ω, where $u = (s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_l} s_l)$ and $v = (s_{l+1} \xrightarrow{a_{l+2}} \ldots \xrightarrow{a_k} s_k)$. $\Pi_k(s)$ denotes the set of all the k-paths of M that start at s, and $\Pi_k = \bigcup_{s^0 \in \iota} \Pi_k(s^0)$.

Let $k \in \mathbb{N}$ be a bound, $0 \leq m \leq k$, $0 \leq l \leq k$, and φ a WELTLK formula. As in the definition of semantics we need to define the satisfiability relation on suffixes of k-paths, we denote by π_l^m the pair (π_l, m), i.e. the k-path π_l together with the designated starting point m. Further, $M, \pi_l^m \models_k \varphi$ denotes that the formula φ is k-true along the suffix $(\pi(m), \ldots, \pi(k))$ of π.

A WELTLK formula φ is k-true along the k-path π_l (in symbols $M, \pi_l \models_k \varphi$) iff $M, \pi_l^0 \models_k \varphi$, where where the bounded satisfaction relation $\models_k$ is defined inductively, with the classical rules for propositional operators and with the following rules for the temporal and epistemic modalities:

$M, \pi_l^m \models_k \mathrm{X}_I \alpha$ iff $(m < k$ and $D\pi[m..m+1] \in I$ and $M, \pi_l^{m+1} \models_k \alpha)$ or $(m = k$ and $l < k$ and $\pi(k) = \pi(l)$ and $D\pi[l..l+1] \in I$ and $M, \pi_l^{l+1} \models_k \alpha)$,

$M, \pi_l^m \models_k \alpha \mathrm{U}_I \beta$ iff $(\exists m \leq j \leq k)(D\pi[m..j] \in I$ and $M, \pi_l^j \models_k \beta$ and $(\forall m \leq i < j) M, \pi_l^i \models_k \alpha)$ or $(l < m$ and $\pi(k) = \pi(l)$ and $(\exists l < j < m)(D\pi[m..k] + D\pi[l..j] \in I$ and $M, \pi_l^j \models_k \beta$ and $(\forall l < i < j) M, \pi_l^i \models \alpha$ and $(\forall m \leq i \leq k) M, \pi_l^i \models_k \alpha))$,

$M, \pi_l^m \models_k \alpha \mathrm{R}_I \beta$ iff $(D\pi[m..k] \geq \mathit{right}(I)$ and $(\forall m \leq j \leq k)(D\pi[m..j] \in I$ implies $M, \pi_l^j \models_k \beta))$ or $(D\pi[m..k] < \mathit{right}(I)$ and $\pi(k) = \pi(l)$ and $(\forall m \leq j \leq k)(D\pi[m..j] \in I$ implies $M, \pi_l^j \models_k \beta)$ and $(\forall l \leq j \leq k)(D\pi[m..k] + D\pi[l..j] \in I$ implies $M, \pi_l^j \models_k \beta))$ or $(\exists m \leq j \leq k)(D\pi[m..j] \in I$ and $M, \pi_l^j \models_k \alpha$ and $(\forall m \leq i \leq j) M, \pi_l^i \models_k \beta)$ or $(l < m$ and $\pi(k) = \pi(l)$ and $(\exists l < j < m)(D\pi[m..k] + D\pi[l..j] \in I$ and $M, \pi_l^j \models_k \alpha$ and $(\forall l < i \leq j) M, \pi_l^i \models \beta$ and $(\forall m \leq i \leq k) M, \pi_l^i \models_k \beta))$,

$M, \pi_l^m \models_k \overline{\mathrm{K}}_{\mathbf{c}} \alpha$ iff $(\exists \pi'_{l'} \in \Pi_k)(\exists 0 \leq j \leq k)(M, \pi'^j_{l'} \models_k \alpha$ and $\pi(m) \sim_{\mathbf{c}} \pi'(j))$,

$M, \pi_l^m \models_k \overline{Y}_\Gamma \alpha$ iff $(\exists \pi'_{l'} \in \Pi_k)(\exists 0 \leq j \leq k)(M, \pi'^j_{l'} \models_k \alpha$ and $\pi(m) \sim_\Gamma^Y \pi'(j))$, where $Y \in \{\mathrm{D}, \mathrm{E}, \mathrm{C}\}$.

Let m be a formula evaluation position, k a bound, and $p, q \in \mathcal{PV}$. An illustration of the bounded semantics is shown in Fig. 1.

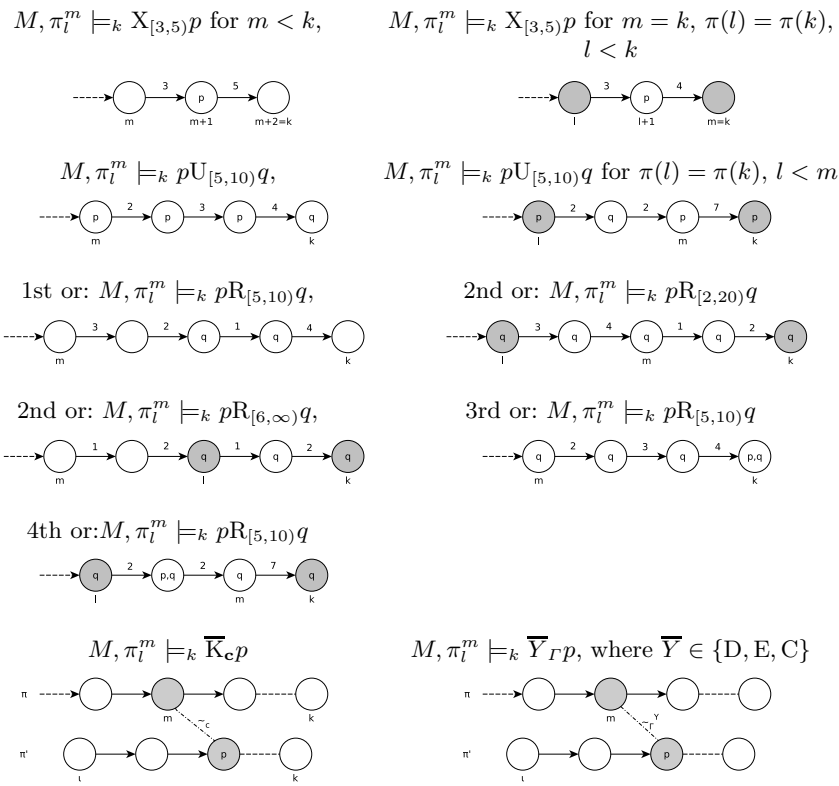

Fig. 1. Evaluation of temporal (the gray states are the same) and epistemic (the gray states are epistemically equivalent) formulae

Let M be a model, and φ a WELTLK formula. We use the following notations: $M \models_k \varphi$ iff $M, \pi_l \models_k \varphi$ for some $\pi_l \in \Pi_k$. The *bounded model checking problem* asks whether there exists $k \in \mathbb{N}$ such that $M \models_k \varphi$.

The following theorem states that for a given model and a WELTLK formula there exists a bound k such that the model checking problem ($M \models \varphi$) can be reduced to the bounded model checking problem ($M \models_k \varphi$). The theorem can be proven by induction on the length of the formula φ; we assume the size of M is the sum of the number of transitions and the number of states.

Theorem 1. *Let M be a model and φ a WELTLK formula. Then, the following equivalence holds: $M \models \varphi$ iff there exists $k \leq |M| \cdot |\varphi| \cdot 2^{|\varphi|}$ such that $M \models_k \varphi$.*

Note however that from the BMC point of view the bound k that makes the bounded and unbounded semantics equivalent is insignificant. This is because the BMC method for large k is unfeasible.

Translation to SAT. Let M be a model, φ a WELTLK formula, and $k \geq 0$ a bound. The presented propositional encoding of the bounded model checking problem for WELTLK is based on the BMC encoding of [18], and it relies on defining the propositional formula: $[M, \varphi]_k := [M^{\varphi,\iota}]_k \wedge [\varphi]_{M,k}$ which is satisfiable if and only if $M \models_k \varphi$ holds.

The definition of $[M^{\varphi,\iota}]_k$ assumes that the states and the joint actions of M, and the sequence of weights associated to the joint actions are encoded symbolically, which is possible, since both the set of states and the set of joint actions are finite. Formally, let $\mathbf{c} \in Ag \cup \{\mathcal{E}\}$. Then, each state $s \in S$ is represented by a vector $w = (\mathtt{w}_1, \ldots, \mathtt{w}_n, \mathtt{w}_\mathcal{E})$ (called a *symbolic state*) of *symbolic local states*, where each symbolic local state $\mathtt{w}_\mathbf{c}$ is a vector of propositional variables. Next, each joint action $\mathfrak{a} \in Act$ is represented by a vector $\mathfrak{a} = (\mathtt{a}_1, \ldots, \mathtt{a}_n, \mathtt{a}_\mathcal{E})$ (called a *symbolic action*) of *symbolic local actions*, where each symbolic local action $\mathtt{a}_\mathbf{c}$ is a vector of propositional variables. Next, each sequence of weights associated to a joint action is represented by a sequence $\delta = (d_1, \ldots, d_{n+1})$ of *symbolic weights*. The *symbolic weight* $d_\mathbf{c}$ is a vector $(\mathtt{d}_1, \ldots, \mathtt{d}_x)$ of propositional variables (called *weight variables*), whose length x depends on the weight functions $d_\mathbf{c}$. Further, in order to define $[M^{\varphi,\iota}]_k$ we need to specify the number of k-paths of the model M that are sufficient to validate φ. To calculate the number, we define the following auxiliary function $f_k : WELTLK \to \mathbb{N}$: $f_k(\mathbf{true}) = f_k(\mathbf{false}) = f_k(p) = f_k(\neg p) = 0$, where $p \in \mathcal{PV}$; $f_k(\alpha \wedge \beta) = f_k(\alpha) + f_k(\beta)$; $f_k(\alpha \vee \beta) = max\{f_k(\alpha), f_k(\beta)\}$; $f_k(\mathrm{X}_I\alpha) = f_k(\alpha)$; $f_k(\alpha \mathrm{U}_I \beta) = k \cdot f_k(\alpha) + f_k(\beta)$; $f_k(\alpha \mathrm{R}_I \beta) = (k+1) \cdot f_k(\beta) + f_k(\alpha)$; $f_k(\overline{\mathrm{C}}_\Gamma\alpha) = f_k(\alpha) + k$; $f_k(Y\alpha) = f_k(\alpha) + 1$ for $Y \in \{\overline{\mathrm{K}}_\mathbf{c}, \overline{\mathrm{D}}_\Gamma, \overline{\mathrm{E}}_\Gamma\}$. Now, since in the BMC method we deal with the existential validity, the number of k-paths sufficient to validate φ is given by the function $\widehat{f_k} : WELTLK \to \mathbb{N}$ that is defined as $\widehat{f_k}(\varphi) = f_k(\varphi) + 1$.

Given the above, the j-th symbolic k-path $\boldsymbol{\pi}_j$ is defined as the following sequence of transitions: $(w_{0,j} \xrightarrow{\mathfrak{a}_{1,j},\delta_{1,j}} w_{1,j} \xrightarrow{\mathfrak{a}_{2,j},\delta_{2,j}} \ldots \xrightarrow{\mathfrak{a}_{k,j},\delta_{k,j}} w_{k,j}, u)$, where $w_{i,j}$ are symbolic states, $\mathfrak{a}_{i,j}$ are symbolic actions, $\delta_{i,j}$ are sequences of *symbolic weights*, for $0 \leq i \leq k$ and $1 \leq j \leq \widehat{f_k}(\varphi)$, and u is the *symbolic number* that is a vector $u = (\mathtt{u}_1, \ldots, \mathtt{u}_y)$ of propositional variables with $y = max(1, \lceil log_2(k+1) \rceil)$.

Let w and w' be two different symbolic states, δ a sequence of symbolic weighs, $\mathfrak{a}$ a symbolic action, and u be a symbolic number. We assume definitions of the following auxiliary propositional formulae: $p(w)$ - encodes the set of states of M in which $p \in \mathcal{PV}$ holds, $I_s(w)$ - encodes the state s of the model M, $\mathcal{T}_\mathbf{c}(\mathtt{w}_\mathbf{c}, (\mathfrak{a}, \delta), \mathtt{w}'_\mathbf{c})$ - encodes the local evolution function of agent $\mathbf{c}$, $H(w, w')$ - encodes equality of two global states, $H_\mathbf{c}(w, w')$ - encodes the equivalence of two local states of agent $\mathbf{c}$, $\mathcal{N}_j^{\sim}(u)$ - encodes that the value j is in the arithmetic relation $\sim \in \{<, \leqslant, =, \geqslant, >\}$ with the value represented by the symbolic number u, $\mathcal{L}_k^l(\boldsymbol{\pi}_n) := \mathcal{N}_l^{=}(u_n) \wedge H(w_{k,n}, w_{l,n})$, $\mathcal{B}_k^I(\boldsymbol{\pi}_n)$ - encodes that the weight represented by the sequence $\delta_{1,n}, \ldots, \delta_{k,n}$ is less than $right(I)$, $\mathcal{D}_{a,b}^I(\boldsymbol{\pi}_n)$ for $a \leq b$ - if $a < b$, then it encodes that the weight represented by the sequence $\delta_{a+1,n}, \ldots, \delta_{b,n}$ belongs to the interval I; otherwise, i.e. if $a = b$, then $\mathcal{D}_{a,b}^I(\boldsymbol{\pi}_n)$ is true iff $0 \in I$, $\mathcal{D}_{a,b;c,d}^I(\boldsymbol{\pi}_n)$ for $a \leq b$ and $c \leq d$ - if $a < b$ and $c < d$, then

it encodes that the weight represented by the sequences $\delta_{a+1,n},\ldots,\delta_{b,n}$ and $\delta_{c+1,n},\ldots,\delta_{d,n}$ belongs to the interval I; if $a=b$ and $c<d$, then it encodes that the weight represented by the sequence $\delta_{c+1,n},\ldots,\delta_{d,n}$ belongs to the interval I; if $a<b$ and $c=d$, then it encodes that the weight represented by the sequence $\delta_{a+1,n},\ldots,\delta_{b,n}$ belongs to the interval I; if $a=b$ and $c=d$, then $\mathcal{D}^I_{a,b;c,d}(\boldsymbol{\pi}_n)$ is true iff $0\in I$. $\mathcal{A}(\mathfrak{a})$ - encodes that each symbolic local action $\mathfrak{a}_\mathbf{c}$ of $\mathfrak{a}$ has to be executed by each agent in which it appears.

The formula $[M^{\varphi,\iota}]_k$, which encodes the unfolding of the transition relation of the model M $\widehat{f_k}(\varphi)$-times to the depth k, is defined as follows:

$$[M^{\varphi,\iota}]_k := \bigvee_{s\in\iota} I_s(w_{0,0}) \wedge \bigvee_{j=1}^{\widehat{f_k}(\varphi)} H(w_{0,0},w_{0,j}) \wedge \bigwedge_{j=1}^{\widehat{f_k}(\varphi)} \bigvee_{l=0}^{k} \mathcal{N}^{=}_l(u_j) \wedge \bigwedge_{j=1}^{\widehat{f_k}(\varphi)} \bigwedge_{i=0}^{k-1} \mathcal{T}(w_{i,j},(\mathfrak{a}_{i,j},\delta_{i,j}),w_{i+1,j}) \tag{1}$$

where $w_{i,j}$, $\mathfrak{a}_{i,j}$, $\delta_{i,j}$, and u_j are, respectively, symbolic states, symbolic actions, sequences of symbolic weights, and symbolic numbers, for $0\le i\le k$ and $1\le j\le \widehat{f}_k(\varphi)$. Moreover, $\mathcal{T}(w,(\mathfrak{a},\delta),w')$ is defined as follows:

$$\mathcal{T}(w,(\mathfrak{a},\delta),w') := \bigwedge_{\mathbf{c}\in Ag\cup\{\mathcal{E}\}} \mathcal{T}_\mathbf{c}(\mathtt{w}_\mathbf{c},(\mathfrak{a},\delta),\mathtt{w}'_\mathbf{c}) \wedge \mathcal{A}(\mathfrak{a}) \tag{2}$$

Let $F_k(\varphi)=\{j\in\mathbb{N} \mid 1\le j\le \widehat{f}_k(\varphi)\}$, and $[\varphi]_k^{[m,n,A]}$ denote the translation of φ along the n-th symbolic path $\boldsymbol{\pi}^m_n$ with the starting point m by using the set $A\subseteq F_k(\varphi)$. Then, the next step is a translation of a WELTLK formula φ to a propositional formula $[\varphi]_{M,k} := [\varphi]_k^{[0,1,F_k(\varphi)]}$.

Let A be a set of k-paths such that $|A|=\widehat{f}_k(\varphi)$. In order to define $[\varphi]_{M,k}$, we have to know how to divide the set A into subsets needed for translating the subformulae of φ. To accomplish this goal we use some auxiliary functions (g_l, g_r, g_s, h^{U}_k, h^{R}_k) that were defined in [18].

Definition 1 (Translation of the WELTLK formulae). *Let M be a model, φ a WELTLK formula, and $k\ge 0$ a bound. We define inductively the translation of φ over a path number $n\in F_k(\varphi)$ starting at the symbolic state $w_{m,n}$ as shown below, where $A\subseteq F_k(\varphi)$, $n'=min(A)$; we assume the classical rules for propositional operators.*

$$[\mathrm{X}_I\alpha]_k^{[m,n,A]} := \begin{cases} \mathcal{D}^I_{m,m+1}(\boldsymbol{\pi}_n)\wedge[\alpha]_k^{[m+1,n,A]}, & \text{if } m<k \\ \bigvee_{l=0}^{k-1}(\mathcal{D}^I_{l,l+1}(\boldsymbol{\pi}_n)\wedge\mathcal{L}^l_k(\boldsymbol{\pi}_n)\wedge[\alpha]_k^{[l+1,n,A]}), & \text{if } m=k \end{cases}$$

$$\begin{aligned}[\alpha\mathrm{U}_I\beta]_k^{[m,n,A]} :=\ & \bigvee_{j=m}^{k}(\mathcal{D}^I_{m,j}(\boldsymbol{\pi}_n)\wedge[\beta]_k^{[j,n,h^{\mathrm{U}}_k(k)]}\wedge\textstyle\bigwedge_{i=m}^{j-1}[\alpha]_k^{[i,n,h^{\mathrm{U}}_k(i)]})\vee \\ & (\textstyle\bigvee_{l=0}^{m-1}(\mathcal{L}^l_k(\boldsymbol{\pi}_n))\wedge\bigvee_{j=0}^{m-1}(\mathcal{N}^{>}_j(u_n)\wedge[\beta]_k^{[j,n,h^{\mathrm{U}}_k(k)]}\wedge \\ & \textstyle\bigvee_{l=0}^{m-1}(\mathcal{N}^{=}_l(u_n)\wedge\mathcal{D}^I_{m,k;l,j}(\boldsymbol{\pi}_n))\wedge \\ & \textstyle\bigwedge_{i=0}^{j-1}(\mathcal{N}^{>}_i(u_n)\to[\alpha]_k^{[i,n,h^{\mathrm{U}}_k(i)]})\wedge\bigwedge_{i=m}^{k}[\alpha]_k^{[i,n,h^{\mathrm{U}}_k(i)]}), \\ [\alpha\mathrm{R}_I\beta)]_k^{[m,n,A]} :=\ & \textstyle\bigvee_{j=m}^{k}(\mathcal{D}^I_{m,j}(\boldsymbol{\pi}_n)\wedge[\alpha]_k^{[j,n,h^{\mathrm{R}}_k(k)]}\wedge\bigwedge_{i=m}^{j}[\beta]_k^{[i,n,h^{\mathrm{R}}_k(i)]})\vee \\ & (\textstyle\bigvee_{l=0}^{m-1}(\mathcal{L}^l_k(\boldsymbol{\pi}_n))\wedge\bigvee_{j=0}^{m-1}(\mathcal{N}^{>}_j(u_n)\wedge[\alpha]_k^{[j,n,h^{\mathrm{R}}_k(k)]}\wedge \end{aligned}$$

$$\begin{array}{l}
\bigvee_{l=0}^{m-1}(\mathcal{N}_l^{=}(u_n) \wedge \mathcal{D}^I_{m,k;l,j}(\boldsymbol{\pi}_n)) \wedge \\
\bigwedge_{i=0}^{j}(\mathcal{N}_i^{>}(u_n) \rightarrow [\beta]_k^{[i,n,h_k^{\mathrm{R}}(i)]}) \wedge \bigwedge_{i=m}^{k}[\beta]_k^{[i,n,h_k^{\mathrm{R}}(i)]}) \vee \\
(\neg\mathcal{B}^I_k(\boldsymbol{\pi}_n) \wedge \bigwedge_{j=m}^{k}(\mathcal{D}^I_{m,j}(\boldsymbol{\pi}_n) \rightarrow [\beta]_k^{[j,n,h_k^{\mathrm{R}}(k)]})) \vee \\
(\mathcal{B}^I_k(\boldsymbol{\pi}_n) \wedge \bigwedge_{j=m}^{k}(\mathcal{D}^I_{m,j}(\boldsymbol{\pi}_n) \rightarrow [\beta]_k^{[j,n,h_k^{\mathrm{R}}(k)]}) \wedge \\
\bigvee_{l=0}^{k-1}[\mathcal{L}^l_k(\boldsymbol{\pi}_n) \wedge \bigwedge_{j=l}^{k}(\mathcal{D}^I_{m,k;l,j}(\boldsymbol{\pi}_n) \rightarrow [\beta]_k^{[j,n,h_k^{\mathrm{R}}(k)]})]),
\end{array}$$

$$[\overline{\mathrm{K}}_{\mathbf{c}}\alpha]_k^{[m,n,A]} := \bigvee_{s\in\iota} I_s(w_{0,n'}) \wedge \bigvee_{j=0}^{k}([\alpha]_k^{[j,n',g_s(A)]} \wedge H_{\mathbf{c}}(w_{m,n}, w_{j,n'})),$$

$$[\overline{\mathrm{D}}_\Gamma\alpha]_k^{[m,n,A]} := \bigvee_{s\in\iota} I_s(w_{0,n'}) \wedge \bigvee_{j=0}^{k}([\alpha]_k^{[j,n',g_s(A)]} \wedge \bigwedge_{\mathbf{c}\in\Gamma} H_{\mathbf{c}}(w_{m,n}, w_{j,n'})),$$

$$[\overline{\mathrm{E}}_\Gamma\alpha]_k^{[m,n,A]} := \bigvee_{s\in\iota} I_s(w_{0,n'}) \wedge \bigvee_{j=0}^{k}([\alpha]_k^{[j,n',g_s(A)]} \wedge \bigvee_{\mathbf{c}\in\Gamma} H_{\mathbf{c}}(w_{m,n}, w_{j,n'})),$$

$$[\overline{\mathrm{C}}_\Gamma\alpha]_k^{[m,n,A]} := [\bigvee_{j=1}^{k}(\overline{\mathrm{E}}_\Gamma)^j\alpha]_k^{[m,n,A]}.$$

The theorem below states the correctness and the completeness of the presented translation. It can be proven by induction on the complexity of the given WELTLK formula.

Theorem 2. *Let M be a model, and φ a WELTLK formula. Then for every $k \in \mathbb{N}$, $M \models_k \varphi$ if, and only if, the propositional formula $[M, \varphi]_k$ is satisfiable.*

Our encoding of the WELTLK formulae is defined recursively over the structure of a WELTLK formula φ, over the current position m of the n-th symbolic k-path, and over the set A of symbolic k-paths, which is initially equal to $F_k(\varphi)$. Next, our encoding does not translate looping and non-looping witnesses separately, but it combines both of them. Further, it is parameterised by the bound k, the set of symbolic k-paths, and closely follows the bounded semantics. Therefore, for fixed n, m, k and A, each subformula ψ of φ requires the constraints of size $O(k \cdot f_k(\varphi))$ using the encoding of ψ at various positions. Moreover, since the encoding of a subformula ψ is only dependent on m, n, k, and A, and, multiple occurrences of the encoding of ψ over the same set of parameters can be shared, the overall size can be bounded by $O(|\varphi| \cdot k \cdot f_k(\varphi))$. Further the size of the formula $[M, \varphi]_k$ is bounded by $O(|T| \cdot k \cdot f_k(\varphi) + |\varphi| \cdot k \cdot f_k(\varphi))$.

The main difficulty in defining of the extension of the BMC method for ELTLK and for the interleaved interpreted systems (IIS) [15] to the BMC method for WELTLK and for WIS is in the encoding of the weighted conditions and in the encoding of the global evolution function. This is because, in contrary to the BMC method of [15], in the WELTLK case we need to deal with joint actions and paths the transitions of which carry a cost. Thus, we have to take care of the following issues: (1) the cumulative weight is less/greater than the given bound k and the considered path is not a loop. (2) the cumulative weight is less/greater than the given bound k and the considered path is a loop. (3) the cumulative weight is counted for the joint actions. The translation has to reflect these possibilities. Further in the IIS case there is no need to encode joint actions together with the corresponding weights. Only local (or synchronised) actions and their weights are encoded. In the WIS case the encoding of the global evolution depends on both the joint actions and the "joint" weight function.

The main difficulties in the extension of the BMC method for WECTLK and for the WIS [17] to the BMC method for WELTLK and for WIS are in the

encoding of the looping conditions. More precisely, in the WECTLK case the looping conditions are much simpler, since each epistemic and each temporal subformulae of a formula to be tested are evaluated over a new symbolic k-path that starts at an initial state, and therefore there is no possibility of getting different infinite paths from the same k-path. In the WELTLK case only epistemic subformulae can be evaluated over a new symbolic k-path that starts at the initial state. Thus, in the translation of X_I, U_I and R_I we cannot let different subformulae use different ways of bending a path into a loop, and thus, we have to disable the possibility of getting different infinite paths from the same k-path.

4 Experimental Results

In the section we experimentally evaluate the performance of our BMC encoding for WELTLK and for WIS, which is implemented as extensions of our tool VerICS [9]. We have conducted the experiments using one classical multi-agent scenario, i.e. the (weighted) modified *bit transmission problem*, and one benchmark that is not yet so common in the multi-agent community, i.e. the *(weighted) generic pipeline paradigm*. Further, for all the considered examples we describe specifications as universal formulae, for which we verify the corresponding counterexample formulae that are interpreted existentially and belong to WELTLK. Moreover, for every specification given, there exists a counterexample, i.e. the WELTLK formula specifying the counterexample holds in the model of the benchmark.

We computed our experimental results on a computer with Intel Xeon 2 GHz processor and 4 GB of RAM, running Linux 2.6. We set the CPU time limit to 1800 seconds, and the memory limit to 2GB. Moreover, we used PicoSAT [2] in version 957 to test the satisfiability of the propositional formulae generated by our SAT-based BMC encoding.

Weighted Generic Pipeline Paradigm. We adapted the benchmark scenario of *a generic pipeline paradigm* [13], and we called it the *weighted generic pipeline paradigm* (WGPP). The model of WGPP involves Producer producing data, Consumer receiving data, and a chain of n intermediate Nodes that transmit data produced by Producer to Consumer. Producer, Nodes, and Consumer have different producing, sending, processing, and consuming costs.

This system is scaled according to the number of its Nodes (agents), i.e. the problem parameter n is the number of Nodes. Fig. 2 shows the local states, the possible actions, and the protocol for each agent. Null actions are omitted in the figure. Further, we assume that the following local states $ProdReady$-0, $Node_i Ready$-0 and $ConsReady$-0 are initial, respectively, for Producer, Node i, and Consumer.

Given Figure 2, the local evolution functions of WGPP are straightforward to infer. Moreover, in the model we assume the following set of proposition variables: $\mathcal{PV}$=$\{ProdReady, ProdSend, ConsReady, ConsReceived\}$ with the following interpretation:

- $(M, s) \models ProdReady$ if $l_P(s) = ProdReady$-0

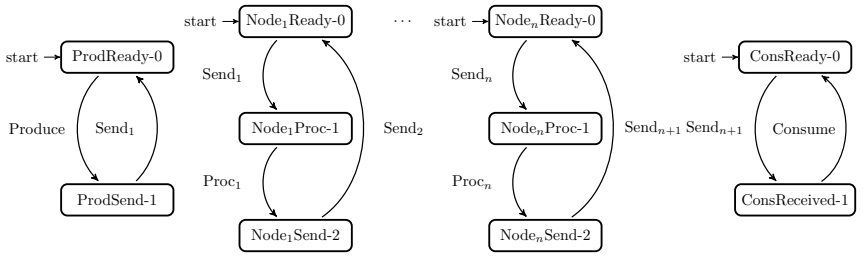

Fig. 2. The WGPP system

- $(M, s) \models \mathit{ProdSend}$ if $l_P(s) = \mathit{ProdSend}$-1
- $(M, s) \models \mathit{ConsReady}$ if $l_C(s) = \mathit{ConsReady}$-0
- $(M, s) \models \mathit{ConsReceived}$ if $l_C(s) = \mathit{ConsReceived}$-1

Let $1 \leqslant i \leqslant n$, and Min denote the minimum cost that is required to receive by Consumer the data produced by Producer. We have tested the WGPP with the following local weight functions: $d_P(\mathit{Produce}) = 4$, $d_P(\mathit{send}_1) = 2$, $d_C(\mathit{Consume}) = 4$, $d_C(\mathit{send}_{n+1}) = 2$, $d_{Ni}(\mathit{send}_i) = d_{Ni}(\mathit{send}_{i+1}) = 2$ and $d_{Ni}(\mathit{Proc}_i) = 2$, and their multiplications by 1,000 and 1,000,000 on the following specifications (universal formulae):

- $\varphi_1 = \mathrm{K}_P\mathrm{G}_{[Min, Min+1)}\mathit{ConsReceived}$, which expresses that Producer knows that always the cost of receiving by Consumer the commodity is Min.
- $\varphi_2 = \mathrm{K}_P\mathrm{G}(\mathit{ProdSend} \rightarrow \mathrm{F}_{[0, Min - d_P(\mathit{Produce}))}\mathit{ConsReceived})$, which states that Producer knows that always if she/he produces a commodity, then Consumer receives the commodity and the cost is less than $Min - d_P(\mathit{Produce})$.
- $\varphi_3 = \mathrm{K}_P\mathrm{G}(\mathit{ProdSend} \rightarrow \mathrm{K}_C\mathrm{K}_P\mathrm{F}_{[0, Min - d_P(\mathit{Produce}))}\mathit{ConsReceived})$, which states that Producer knows that always if she/he produces a commodity, then Consumer knows that Producer knows that Consumer has received the commodity and the cost is less than $Min - d_P(\mathit{Produce})$.
- $\varphi_4 = \mathrm{K}_C\mathrm{G}(\mathit{ProdReady} \rightarrow \mathrm{X}_{[d_P(\mathit{Produce}), d_P(\mathit{Produce})+1)}\mathit{ProdSend})$, which expresses that Consumer knows that the cost of producing of a commodity by Producer is $d_P(\mathit{Produce})$.

Table 1. WGPP with n nodes

Formula	**(n, k)** - **n** is the number of nodes, **k** is the bound	$\widehat{\mathbf{f_k}}(\varphi)$
φ_1	(1,3), (2,5), (3,6), (4,6), (5,7), (6,8), (7,8), (8,9), (9,9), (10,10), (15,12), (20,14), (25,16), (30,18), (35,19), (40, 21), (45,22), (50,23)	2
φ_2	(1,3), (2,5), (3,6), (4,6), (5,7), (6,8), (7,8), (8,9), (9,9), (10,10), (11,11),(12,11), (13,12), (14,12), (15,12), (20,14), (25,16), (30,18), (31,18), (32,18), (33,19), (34,19), (35,19)	2
φ_3	(1,3), (2,4), (3,5), (4,6), (5,6), (6,7), (7,8), (8,9), (9,9), (10,10)	4
φ_4	$(n, 4)$ for $n \geq 1$	2

The size of the reachable state space of the WGPP system is $4 \cdot 3^n$, for $n \geq 1$. The length of the counterexamples, and the number of the considered k-paths for the above formulae are shown in Table 1; note that for formulae φ_1-φ_3 we are not

able to provide a general formula that for a given number of nodes provides the bound k. The data presented in Table 1 are generated by our implementation. Moreover, in Table 2 we provide a witness that was generated for formula φ_1.

Table 2. The witness for WGPP and for formula φ_1 that consists of two 3-paths

Path nr 0 with l = 3			
Step	Action	Weights	Global state
0		$\langle 0, 0, 0\rangle$	$\langle$ProdReady, Node$_1$Ready, ConsReady$\rangle$
1	$\langle$Produce, ε_{N1}, $\varepsilon_C\rangle$	$\langle 4000, 0, 0\rangle$	$\langle$ProdSend, Node$_1$Ready, ConsReady$\rangle$
2	$\langle$Send$_1$, Send$_1$, $\varepsilon_C\rangle$	$\langle 2000, 2000, 0\rangle$	$\langle$ProdReady, Node$_1$Proc, ConsReady$\rangle$
3	$\langle\varepsilon_P$, Proc$_1$, $\varepsilon_C\rangle$	$\langle 0, 2000, 0\rangle$	$\langle$ProdReady, Node$_1$Send, ConsReady$\rangle$
Path nr 1 with l = 0			
0		$\langle 0, 0, 0\rangle$	$\langle$ProdReady, Node$_1$Ready, ConsReady$\rangle$
1	$\langle$Produce, ε_{N1}, $\varepsilon_C\rangle$	$\langle 4000, 0, 0\rangle$	$\langle$ProdSend, Node$_1$Ready, ConsReady$\rangle$
2	$\langle$Send$_1$, Send$_1$, $\varepsilon_C\rangle$	$\langle 2000, 2000, 0\rangle$	$\langle$ProdReady, Node$_1$Proc, ConsReady$\rangle$
3	$\langle$Produce, Proc$_1$, $\varepsilon_C\rangle$	$\langle 4000, 2000, 0\rangle$	$\langle$ProdSend, Node$_1$Send, ConsReady$\rangle$

The Weighted Bits Transmission Problem. We adapted the scenario of *a bit transmission problem* [1], and we called it the *weighted bits transmission problem* (WBTP). The WBTP involves two agents, a sender $\mathcal{S}$, and a receiver $\mathcal{R}$, communicating over a possibly faulty communication channel (the environment), and there are fixed costs $c_{\mathcal{S}}$ and $c_{\mathcal{R}}$ associated with, respectively, sending process of $\mathcal{S}$ and $\mathcal{R}$. $\mathcal{S}$ wants to communicate some information (e.g., the n-bit number) to $\mathcal{R}$. One protocol to achieve this is as follows. $\mathcal{S}$ immediately starts sending the n-bit number to $\mathcal{R}$, and continues to do so until it receives an acknowledgement from $\mathcal{R}$. $\mathcal{R}$ does nothing until it receives the n-bit number; from then on it sends acknowledgements of receipt to $\mathcal{S}$. $\mathcal{S}$ stops sending the n-bit number to $\mathcal{R}$ when it receives an acknowledgement. Note that $\mathcal{R}$ will continue sending acknowledgements even after $\mathcal{S}$ has received its acknowledgement. This system is scaled according to the number of bits the $\mathcal{S}$ wants to communicate to $\mathcal{R}$.

Each agent of the scenario can be modelled by considering its local states, local actions, local protocol, local evolution function, local weight function, and local valuation function. For $\mathcal{S}$, it is enough to consider 2^{n+1} possible local states representing the value of the n-bit number that $\mathcal{S}$ is attempting to transmit, and whether or not $\mathcal{S}$ has received an acknowledgement from $\mathcal{R}$. Thus, we have: $L_{\mathcal{S}} = \{0, \ldots, 2^n - 1, 0\text{-}ack, \ldots, 2^n - 1\text{-}ack\}$. Further, $\iota_{\mathcal{S}} = \{0, \ldots, 2^n - 1\}$. For $\mathcal{R}$, it is enough to consider $2^n + 1$ possible local states representing: the value of the received n-bit number, if any, and the circumstance in which no number has been received yet (represented by ϵ). Thus, we have $L_{\mathcal{R}} = \{0, \ldots, 2^n - 1, \epsilon\}$, and $\iota_{\mathcal{R}} = \{\epsilon\}$. For the environment $\mathcal{E}$, to simplify the presentation, we shall to consider just one local state: $L_{\mathcal{E}} = \{\cdot\} = \iota_{\mathcal{E}}$. Now we can define the set of possible global states S for the scenario as the product $L_{\mathcal{S}} \times L_{\mathcal{R}} \times L_{\mathcal{E}}$, and we consider the following set of initial states $\iota = \{(0, \epsilon, \cdot), \ldots, (2^n - 1, \epsilon, \cdot)\}$.

The set of actions available to the agents are as follows: $Act_{\mathcal{S}} = \{sendbits, \lambda\}$, $Act_{\mathcal{R}} = \{sendack, \lambda\}$, where λ stands for no action. The actions for $\mathcal{E}$ correspond to the transmission of messages between $\mathcal{S}$ and $\mathcal{R}$ on the unreliable communication

channel. The set of actions for $\mathcal{E}$ is $Act_{\mathcal{E}} = \{\leftrightarrow, \rightarrow, \leftarrow, -\}$, where $\leftrightarrow$ represents the action in which the channel transmits any message successfully in both directions, $\rightarrow$ that it transmits successfully from $\mathcal{S}$ to $\mathcal{R}$ but loses any message from $\mathcal{R}$ to $\mathcal{S}$, $\leftarrow$ that it transmits successfully from $\mathcal{R}$ to $\mathcal{S}$ but loses any message from $\mathcal{S}$ to $\mathcal{R}$, and $-$ that it loses any messages sent in either direction. The set $Act = Act_{\mathcal{S}} \times Act_{\mathcal{R}} \times Act_{\mathcal{E}}$ defines the set of joint actions for the scenario.

The local weight functions of agents are defined as follows: $d_{\mathcal{S}}(sendbits) = a$ with $a \in \mathbb{N}$, $d_{\mathcal{S}}(\lambda) = 0$, $d_{\mathcal{R}}(sendack) = b$ with $b \in \mathbb{N}$, $d_{\mathcal{R}}(\lambda) = 0$, and $d_{\mathcal{E}}(\leftrightarrow) = d_{\mathcal{E}}(\rightarrow) = d_{\mathcal{E}}(\leftarrow) = d_{\mathcal{E}}(-) = 0$. We assume zero-weight for the actions of $\mathcal{E}$, since we wish to only count the cost of sending and receiving messages.

The local protocols of the agents are the following: $P_{\mathcal{S}}(0) = \ldots = P_{\mathcal{S}}(2^n - 1) = \{sendbits\}$, $P_{\mathcal{S}}(0\text{-}ack) = \ldots = P_{\mathcal{S}}(2^n - 1\text{-}ack) = \{\lambda\}$, $P_{\mathcal{R}}(0) = \ldots = P_{\mathcal{R}}(2^n - 1) = \{sendack\}$, $P_{\mathcal{R}}(\epsilon) = \{\lambda\}$, $P_{\mathcal{E}}(\cdot) = Act_{\mathcal{E}} = \{\leftrightarrow, \rightarrow, \leftarrow, -\}$.

It should be straightforward to infer the model that is induced by the informal description of the scenario we considered above together with the local states, actions, protocols, and weight functions defined above.

In the model we assume the following set of proposition variables: $\mathcal{PV} = \{\mathbf{0}, \ldots, \mathbf{2^n - 1}, \mathbf{recack}\}$ with the following interpretation:
$(M, s) \models \mathbf{i}$ if $l_{\mathcal{S}}(s) = i$ or $l_{\mathcal{S}}(s) = i\text{-}ack$, for $i = 0, \ldots, 2^n - 1$
$(M, s) \models \mathbf{recack}$ if $l_{\mathcal{S}}(s) = 0\text{-}ack$ or $\ldots$ or $l_{\mathcal{S}}(s) = 2^n - 1\text{-}ack$.
We have tested the WBTP on the following specifications (universal formulae):

- $\varphi_1 = \mathrm{G}_{[a+b,a+b+1)}(\mathbf{recack} \rightarrow \mathrm{K}_{\mathcal{S}}(\mathrm{K}_{\mathcal{R}}(\bigvee_{i=0}^{2^n-2} \mathbf{i})))$ - the property says that if an ack is received by $\mathcal{S}$, then $\mathcal{S}$ knows that $\mathcal{R}$ knows at least one value of the n-bit numbers except the maximal value, and the cost is $a + b$.
- $\varphi_2 = \mathrm{G}_{[a+b,a+b+1)}(\mathrm{K}_{\mathcal{S}}(\bigvee_{i=0}^{2^n-1}(\mathrm{K}_{\mathcal{R}}(\mathbf{i})))$ – the property says that $\mathcal{S}$ knows that $\mathcal{R}$ knows the value of the n-bit number and the cost is $a + b$.

The size of the reachable state space of the WBTP system is $3 \cdot 2^n$ for $n \geq 1$. The number of the considered k-paths is the following: $\widehat{f}_k(\varphi_1) = 3$ and $\widehat{f}_k(\varphi_2) = 2^n + 2$. The length of the counterexamples for both formulae is equal to 2 for any $n > 0$.

4.1 Performance Evaluation

Table 3. The computation time and memory consumption

	WGPP with 1 node						**WBTP with 1 bit**					
	Time (sec.)			**Memory (MB)**			**Time (sec.)**			**Memory (MB)**		
Formula	x1	$\mathrm{x}10^3$	$\mathrm{x}10^6$	x1	$\mathrm{x}10^3$	$\mathrm{x}10^6$	x1	$\mathrm{x}10^3$	$\mathrm{x}10^6$	x1	$\mathrm{x}10^3$	$\mathrm{x}10^6$
φ_1	0.04	0.13	0.19	1.90	2.67	3.70	0.02	0.03	0.06	1.12	1.38	1.64
φ_2	0.04	0.11	0.16	1.90	2.82	3.78	0.03	0.05	0.10	1.12	1.66	1.92
φ_3	0.41	0.71	1.00	5.87	7.22	8.58	-	-	-	-	-	-
φ_4	0.09	0.22	0.34	1.91	3.18	4.25	-	-	-	-	-	-

The experimental results show that our SAT-based BMC method is slightly sensitive to scaling up the weights (see Fig. 3 and Fig. 4). To be more precise, we observed that when we scale up the weights for both benchmarks and for all

properties, the computation time and the memory usage grows linearly, regardless of the considered number of nodes or n-bit integer value. For example, we refer the reader to Table 3 for the detailed results we have for WGPP/WBTP with one node/bit and with the basic weights and their multiplication by 10^3 or 10^6. The sensitivity to growing weights follows from the encoding of the cumulative weight. Namely, the number of bits that is required to encode the cumulative weights depends on the number of agents, on the length of the counterexample (i.e. the bound k) and the maximal weight that appear in the whole system.

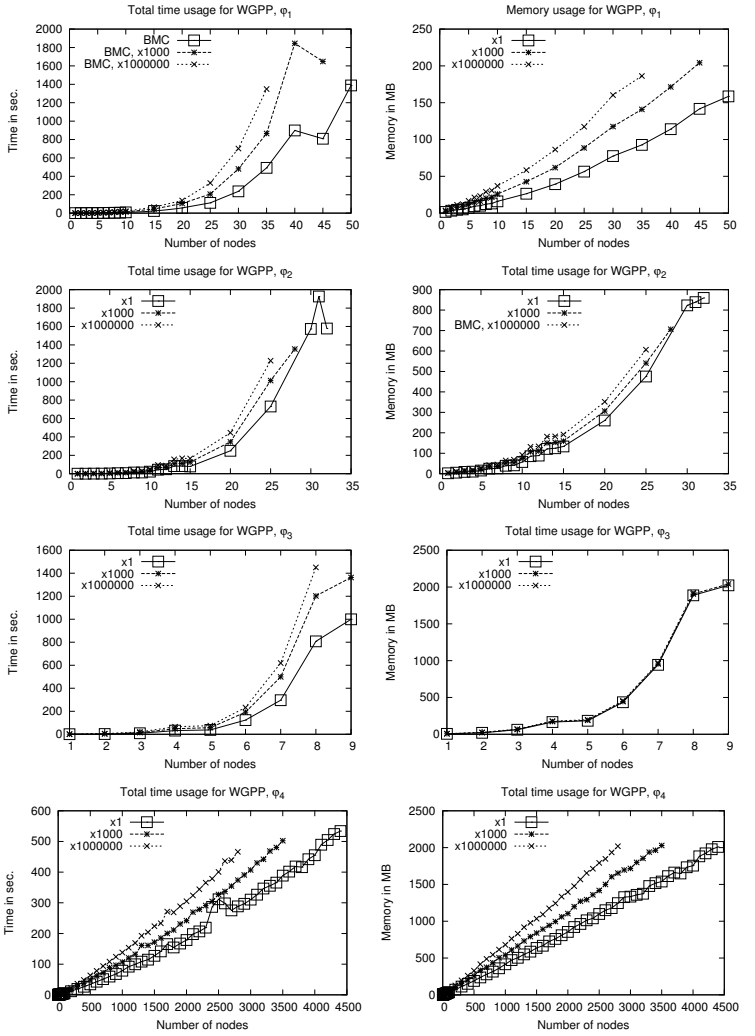

Fig. 3. WGPP with n nodes

As one can see from the line charts in Fig. 3 for WGPP with the basic weights, in the time limit set our method is able to verify the formulae φ_1 - φ_4,

respectively, for 50, 32, 9, and 4400 nodes. The high efficiency of our method in the case of the formula φ_4 results from the constant length of the counterexample. In all the other cases we can observe that our method is sensitive to scaling up the size of benchmarks. This is because the length of the counterexamples grows with the number of the components, and the efficiency of the SAT-based BMC strongly depends on the length of the counterexamples. Further, in the case of the formula φ_3 we get results for 9 nodes only. This follows from the fact that apart from the growing length of the counterexample we need to consider as many as four k-paths.

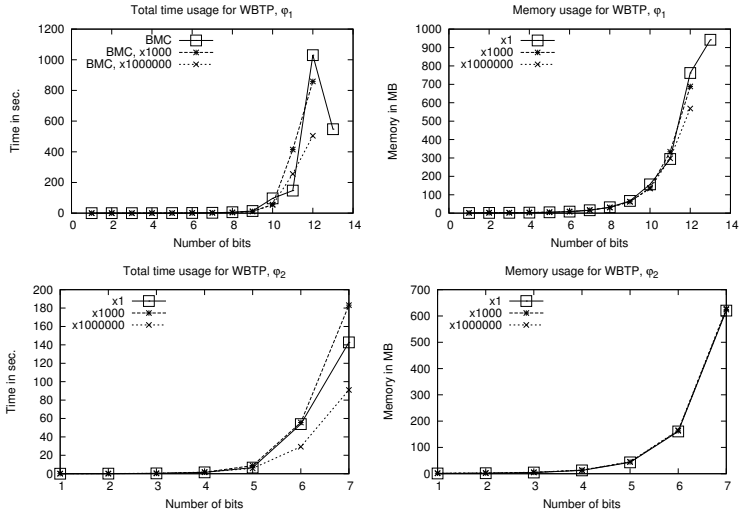

Fig. 4. WBTP with n bits integer value

As one can see from the line charts in Fig. 4 for WBTP with the basic weights (i.e. $a = 1$, $b = 2$), in the time limit set our method is able to verify the formulae φ_1 and φ_2, respectively, for 13 and 7 bits integer value. The inferiority of our method in the case of this benchmark results from the fact that our method has do deal with the model which has the exponential number of initial states. Observe that this is not the case in the WGPP benchmark.

5 Conclusions

We have defined and implemented the SAT-based BMC method for WELTLK and for WISs. The experimental results show that our method is slightly sensitive to scaling up the weights. Concerning the sensitivity of our SAT-based BMC encoding to growing size of the checked system, we can observe that it is rather standard, i.e. the method is not so efficient if many symbolic k-paths are encoded, and the length of the counterexample grows with the number of agents.

The BMC for WELTLK and for WISs may also be performed by means of Ordered Binary Diagrams (OBDD) and Satisfiability Modulo Theories (SMT). This will be explored in the future. Moreover, our future work includes a comparison of the OBDD- SMT- and SAT-based BMC method for WISs. Further, we would like to point out that the proposed BMC method can be used for solving some planing problems that can be formulated in terms of weighted automata. Namely, we can formalize the notion of the agent as a weighted automaton, and then apply our BMC technique for WIS that are generated by a given network of weighted automata. A planning problem defined in terms of weighted automata was considered, e.g. in [6].

References

1. Lomuscio, A., Sergot, M.: Violation, error recovery, and enforcement in the bit transmission problem. Imperial College Press (2002)
2. Biere, A.: Picosat essentials. Journal on Satisfiability, Boolean Modeling and Computation (JSAT) 4, 75–97 (2008)
3. Clarke, E., Biere, A., Raimi, R., Zhu, Y.: Bounded model checking using satisfiability solving. Formal Methods in System Design 19(1), 7–34 (2001)
4. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. The MIT Press, Cambridge (1999)
5. Emerson, E.A.: Temporal and modal logic. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science, vol. B, ch. 16, pp. 996–1071. Elsevier Science Publishers (1990)
6. Fabre, E., Jezequel, L.: Distributed optimal planning: an approach by weighted automata calculus. In: Proceedings of CDC 2009, pp. 211–216. IEEE (2009)
7. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning about Knowledge. MIT Press (1995)
8. Gammie, P., van der Meyden, R.: MCK: Model checking the logic of knowledge. In: Alur, R., Peled, D.A. (eds.) CAV 2004. LNCS, vol. 3114, pp. 479–483. Springer, Heidelberg (2004)
9. Kacprzak, M., Nabialek, W., Niewiadomski, A., Penczek, W., Pólrola, A., Szreter, M., Woźna, B., Zbrzezny, A.: VerICS 2007 - a model checker for knowledge and real-time. Fundamenta Informaticae 85(1-4), 313–328 (2008)
10. Levesque, H.: A logic of implicit and explicit belief. In: Proceedings of the 6th National Conference of the AAAI, pp. 198–202. Morgan Kaufman (1984)
11. Lomuscio, A., Qu, H., Raimondi, F.: MCMAS: A model checker for the verification of multi-agent systems. In: Bouajjani, A., Maler, O. (eds.) CAV 2009. LNCS, vol. 5643, pp. 682–688. Springer, Heidelberg (2009)
12. Lomuscio, A., Sergot, M.: Deontic interpreted systems. Studia Logica 75(1), 63–92 (2003)
13. Peled, D.: All from one, one for all: On model checking using representatives. In: Courcoubetis, C. (ed.) CAV 1993. LNCS, vol. 697, pp. 409–423. Springer, Heidelberg (1993)
14. Penczek, W., Lomuscio, A.: Verifying epistemic properties of multi-agent systems via bounded model checking. Fundamenta Informaticae 55(2), 167–185 (2003)
15. Penczek, W., Woźna-Szcześniak, B., Zbrzezny, A.: Towards SAT-based BMC for LTLK over Interleaved Interpreted Systems. Fundamenta Informaticae 119(3-4), 373–392 (2012)

16. Wooldridge, M.: An introduction to multi-agent systems. John Wiley (2002)
17. Woźna-Szcześniak, B.: SAT-based bounded model checking for weighted deontic interpreted systems. In: Correia, L., Reis, L.P., Cascalho, J. (eds.) EPIA 2013. LNCS, vol. 8154, pp. 444–455. Springer, Heidelberg (2013)
18. Zbrzezny, A.: A new translation from ECTL* to SAT. Fundamenta Informaticae 120(3-4), 377–397 (2012)