Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 687578, 9 pages
http://dx.doi.org/10.1155/2013/687578

Research Article

Phase Transitions of Contingent Planning Problem

Junping Zhou, Weihua Su, Zhiqiang Ma, and Minghao Yin

College of Computer, Northeast Normal University, Changchun 130117, China

Correspondence should be addressed to Zhiqiang Ma; mazq@nenu.edu.cn and Minghao Yin; ymh@nenu.edu.cn

Received 19 June 2013; Accepted 18 July 2013

Academic Editor: William Guo

This paper explores the phase transitions of the contingent planning problems. We present CONTINGENT PLAN-EXISTENCE algorithm and CONTINGENT PLAN-NONEXISTENCE algorithm for quickly proving that the contingent planning instances have solutions and have no solutions, respectively. By analyzing the two algorithms, the phase transition area of the contingent planning problems is obtained. If the number of the actions is not greater than θ_{ub}, the CONTINGENT PLAN-NONEXISTENCE algorithm can prove that nearly all the contingent planning instances have no solution. If the number of the actions is not lower than θ_{lb}, the CONTINGENT PLAN-EXISTENCE algorithm can prove that nearly all the contingent planning instances have solutions. The results of the experiments show that there exist phase transitions from a region where almost all the contingent planning instances have no solution to a region where almost all the contingent planning instances have solutions.

1. Introduction

There are many artificial intelligence (AI) problems that have been shown to be inherently intractable. For example, the propositional satisfiability problem, or k-SAT problem for short, is a prototypical NP-complete problem [1], which means that k-SAT problem cannot be solved in polynomial time if $P \neq \text{NP}$. It is significant to investigate the phase transitions of the intractable problems. That is because by working on it the researchers cannot only analyze the structure of intractable problems, but also understand the average-case performance of the solvers that solve these AI problems.

In recent years, much attention has been given to the phase transitions of the random k-SAT problems [2–8]. The researchers found that the phase transitions of the random k-SAT problems did exist, where instances changed from being almost all satisfiable to being almost all unsatisfiable, and the hard instances only occurred in the phase transition area. The same phenomena were also observed for some other NP-complete problems, such as the traveling salesman problems [9], the constraint satisfaction problems [10], and the Max-SAT problems [11]. Some other researchers made tremendous progress towards the phase transitions of the artificial planning problems, for instance, STRIPS planning problems [12], conformant planning problems [13], and plan modification of the conformant planning problems [14]. In consequence, one interesting question was put forward whether there were phase transitions in other artificial planning problems.

The aim of this paper is to explore the phase transitions of the contingent planning problems. In order to investigate the phenomena, CONTINGENT PLAN-EXISTENCE algorithm and CONTINGENT PLAN-NONEXISTENCE algorithm are presented. The phase transition point of the contingent planning problems is obtained by providing probabilistic analyses of the CONTINGENT PLAN-EXISTENCE algorithm and the CONTINGENT PLAN-NONEXISTENCE algorithm. The results of the experiments show that the empirical results match the phase transitions from the theoretic analyses.

2. Background

2.1. Definitions and Notations. This section describes some notations about contingent planning problem relevant to our work. Further details, such as the exclusive actions and the consistent actions, can be found in [15–19].

Definition 1 (contingent planning problem). A contingent planning problem is a 5-tuple $P = \langle S, A, O, I, G\rangle$ with the following components:

(i) S is a finite set of states and the subset of S is called belief state;

(ii) A is a finite set of actions. An action is a pair $\langle \text{pre}, \text{eff} \rangle$, respectively, called preconditions and effects. Effects are a triple $\langle \text{cond}, \text{del}, \text{add} \rangle$, where cond is the effect conditions, del is the deleting effects and add is the adding effects;

(iii) O is a finite set of observations;

(iv) $I \subseteq S$ is a set of states over S, which is called the initial belief state;

(v) $G \neq \oslash$ is the goal, which is actually a partial state composed of the goal conditions.

More specifically, contingent planning problem is the task of generating a conditional plan given uncertainty about the initial state and action effects, but with the ability to observe some aspects of the current state [20]. In this paper we only focus on the contingent planning problem with uncertainty about the initial state and determinacy about the actions effects, but with the ability to sense the current state.

Definition 2 (action applicability). Let Bs be a belief state. The action a is applicable in Bs if $\forall s \in \text{Bs}$: $\text{pre}(a) \subseteq s$.

If the preconditions of an action are satisfied by all states of the belief state, then the action can be applicable. On the contrary, if the preconditions of an action are not satisfied, the action cannot be carried out, which means that the belief state is changeless. The resulting belief state by executing an action is determined by the adding effects and the deleting effects.

Definition 3 (observation function). Let O be a finite set of observations and Bs an arbitrary belief state. The observation function over Bs and O is a function $\varphi : \text{Bs} \rightarrow 2^{O} \setminus \emptyset$ such that $\forall s \in \text{Bs} : \varphi(s) \subseteq O$.

In the following, we will use $\varphi^{-}(o)$ to denote the set of states that are compatible with the observation o; $\varphi^{-}(o^{-})$ to denote the set of states that are compatible with other observations rather than o.

Definition 4 (conditional plan). The set of conditional plans Π is the minimal set such that

(i) $\emptyset \in \Pi$;

(ii) for an action a, if $a \in A$ and $\pi \in \Pi$, then $a \circ \pi \in \Pi$;

(iii) if $o \in O$ and $\pi_1, \pi_2 \in \Pi$, then if o then π_1 else $\pi_2 \in \Pi$.

Definition 5 (contingent plan). Given a contingent planning problem $P = \langle S, A, O, I, G \rangle$, a contingent plan or solution is a conditional plan π to the problem P if and only if

(i) π is applicable in the initial belief state I;

(ii) every run of π from the initial belief state I ends in G.

A random contingent planning model, called variable model, is used to carry out an investigation on the phase transitions, which we will address in the following.

Definition 6 (variable model). A variable model M_v is a 5-tuple $\langle S, A, O, I, G \rangle$, where

(i) S is a finite set of states and every state is made up of some propositions;

(ii) A is a finite set of actions. For $\forall a_1, a_2 \in A$, $\text{Num}(\text{pre}(a_1)) = \text{Num}(\text{pre}(a_2))$ and $\text{Num}(\text{eff}(a_1)) = \text{Num}(\text{eff}(a_2))$, where $\text{Num}(\text{pre}(a_1))$ and $\text{Num}(\text{pre}(a_2))$ are the numbers of preconditions of a_1 and a_2 respectively, $\text{Num}(\text{eff}(a_1))$ and $\text{Num}(\text{eff}(a_2))$ are the numbers of postconditions of a_1 and a_2, respectively;

(iii) O is a finite set of observations;

(iv) $I \subseteq S$ is a set of initial states over S, which is also called the initial belief state;

(v) $G \neq \oslash$ is the goal, which is actually a partial state composed of the goal conditions.

2.2. The Fundamental Analysis of the Variable Model. In this section, we present the distributions of instances under the variable model. Before addressing the analysis of the variable model, two assumptions of the random contingent planning instances generated in this paper are provided. The first one is that each precondition of an action is selected independently of other precondition and postcondition. Similar rule is hold for the postconditions. The second one is that each action has a fixed number of preconditions. And the number of postconditions also agrees with the rule.

Because of the assumptions that each precondition (or postcondition) of an action is selected independently of other preconditions and postconditions, and the precondition or the postcondition is either a proposition or its negation, the random contingent planning instances under the variable model are distributed as follows.

Given n propositions, m actions, r preconditions, c postconditions, and g goal conditions, for a random proposition p and an action a, p is a precondition of the action with the probability $r/(2n)$; alternatively $\neg p$ is a precondition with the probability $r/(2n)$. And $c/(2n)$ is the probability for postconditions.

Because the initial state of the contingent planning problem is partially known, there exists a set of initial states, where the number of the initial states is k in this paper. Every initial state is made up of conditions, each of which is selected at random from the $2n$ conditions since there are n propositions. For the goal conditions, the g goal conditions are also selected at random from the $2n$ conditions.

Furthermore, in the rest of paper, three inequalities are also used to analyze the phase transitions. The detailed proofs can be found in [21]

$$e^{-x/(1-x)} \leq 1 - x, \quad x \in [0, 1) \tag{1}$$

$$1 - x \leq e^{-x} \tag{2}$$

$$\frac{xy}{1 + xy} \leq 1 - (1 - x)^{y}, \quad x \in [0, 1). \tag{3}$$

```
Algorithm CONTINGENT PLAN-NONEXITENCE (S, A, O, I, G)
  for each s ∈ I
    if pre(A) ⊄ s;
    then return failure;
  endfor
  for each g ∈ G
    if g ⊄ eff(A)
    then return failure;
  endfor
  for each o ∈ O
    Bs_T = I ∩ φ⁻(o);
    Bs_F = I ∩ φ⁻(o⁻);
    for each s ∈ Bs_T
      if pre(A) ⊂ s;
        then return don't know;
    endfor
    for each s' ∈ Bs_F
      if pre(A) ⊂ s';
        then return don't know;
    endfor
  endfor
  return failure;
end
```

Algorithm 1: Contingent plan-nonexistence algorithm.

Actually, the following analyses for the phase transitions of the contingent planning problems are based on the fundamental analysis of the variable model. Because it is very simple to solve the sort of contingent planning problems that the goal conditions are satisfied when there is not any actions, in this paper we only deal with the sort of contingent planning problems that the goal conditions are not satisfied by the initial belief state.

3. Contingent Plan Nonexistence Algorithm

In this section, we present the CONTINGENT PLAN-NONEXISTENCE algorithm for quickly proving that a contingent planning instance has no solutions. By analyzing the algorithm, the phase transition area is obtained. The framework of the CONTINGENT PLAN-NONEXISTENCE algorithm is presented in Algorithm 1.

The CONTINGENT PLAN-NONEXISTENCE algorithm firstly considers whether the preconditions of all actions are not in any state s in the initial belief state. If a state in the initial belief state does not satisfy the preconditions of all actions, then the instance has no solutions. Then the algorithm checks out whether a goal condition is not a postcondition of any action. If the postconditions of the actions do not cover all the goal conditions, then the instance has no solutions. The algorithm finally applies an observation to split the initial belief state into two belief states and checks out whether the preconditions of all actions are in any state in the two belief states, respectively. If there is a state in one of the belief states that satisfies the preconditions of all actions, then we cannot decide whether the instance has solutions or not. In addition, if there is not a state in all the split belief states that satisfies the preconditions of all actions, the instance has no solutions. For a contingent planning instance with k initial states, m actions, b observations, and g goal conditions, the time complexity of the algorithm is $O((k+g+2kb)m)$. Therefore, the CONTINGENT PLAN-NONEXISTENCE algorithm can decide that the instance has no solutions efficiently.

Theorem 7. *The contingent planning instance has no solution if CONTINGENT PLAN-NONEXISTENCE algorithm returns failure; the contingent planning instance cannot be determined whether it has solutions if CONTINGENT PLAN-NONEXISTENCE algorithm returns do not know.*

Proof. Let us first list the cases that the CONTINGENT PLAN-NONEXISTENCE algorithm returns *failure*. (1) The initial belief state involves a state that does not contain the preconditions of all the actions. As we know, if the preconditions of all the actions are not in the state, any action cannot be executed. Because in this paper we only deal with the sort of contingent planning problems that no goal conditions are satisfied by the initial belief state, a sequence of empty actions cannot be the solution of the instances. Therefore, the contingent planning instances do not have solutions. (2) There is a goal condition that is not an effect of any action, which means that the goal condition cannot be obtained by executing any action. Therefore, the contingent planning instances do not have solutions. (3) There is not a state in all the split belief states that satisfies the preconditions of all actions, which means that any action cannot be executed. Therefore, the contingent planning instances have no solutions. Then let us discuss the case that the CONTINGENT PLAN-NONEXISTENCE algorithm returns *do not know*. There is a state in one of the belief states that satisfies the preconditions of all actions, which means that at least an action can be executed from the belief state. However, we cannot guarantee that all the goal conditions are satisfied by executing any conditional plan from each state in the initial belief state. Therefore, the contingent planning instance cannot be determined whether it has solutions.

By Theorem 7, we know that the CONTINGENT PLAN-NONEXISTENCE algorithm is correct, but it cannot guarantee the completeness. In other words, if a contingent planning instance has solutions, the CONTINGENT PLAN-NONEXISTENCE algorithm could not determine that the instance has solutions. However, we can prove in Theorem 8 that nearly all contingent planning instances have no solutions when σ is small enough (σ is a constant and $0 < \sigma < 1$). □

Theorem 8. *For random contingent planning instances under the variable model, if the number of actions satisfies*

$$m \le \theta_{ub}, \quad \theta_{ub} = \min\left(\frac{(1-(1-r/2n)^n)(-\ln\ln 1/\sigma)}{(1-r/2n)^n}, \left(\frac{(2n-c)}{c}\right)\left(\ln g - \ln\ln\frac{1}{\sigma}\right)\right), \tag{4}$$

where n is the number of propositions; m is the number of actions; r and c, respectively, are the expected numbers of preconditions and postconditions within an action; g is the number of goal conditions; k is the number of initial states; σ $(0 < \sigma < 1)$ is a constant, then the CONTINGENT PLAN-NONEXISTENCE algorithm proves that no solution exists for at least $1-\sigma$ of the instances.

Proof. If we are to prove that the algorithm determines that no solution exists for at least $1-\sigma$ of the instances, we only need to prove that the probability that every initial state satisfies the preconditions of any action, the probability that a goal condition is not a postcondition of any action, and the probability that every initial state in the belief state Bs_T (or Bs_F) satisfies the preconditions of any action are not more than σ. Suppose (4) is true, we have

$$m \le \min\left(\frac{(1-(1-r/2n)^n)(-\ln\ln 1/\sigma)}{(1-r/2n)^n}, \left(\frac{(2n-c)}{c}\right)\left(\ln g - \ln\ln\frac{1}{\sigma}\right)\right). \tag{5}$$

Because the initial state of the contingent planning problem is partially known, there exists a set of initial states, where the number of the initial states is k in this paper. Every initial state is made up of conditions, each of which is selected at random from the $2n$ conditions since there are n propositions. At first, the probability that every initial state satisfies the preconditions of any action is given as follows:

$(1-r/2n)^n$: probability that a state satisfies the preconditions of an action;

$1-(1-r/2n)^n$: probability that a state does not satisfy the preconditions of an action;

$(1-(1-r/2n)^n)^m$: probability that a state does not satisfy the preconditions of any action;

$1-(1-(1-r/2n)^n)^m$: probability that a state satisfies the preconditions of any action;

$(1-(1-(1-r/2n)^n)^m)^k$: probability that every initial state satisfies the preconditions of any action.

Then, the probability that every initial state in the belief state Bs_T satisfies the preconditions of any action is given as follows, where the number of the states in Bs_T (or Bs_F) is x $(1 \le x \le k)$,

$(1-r/2n)^n$: probability that a state satisfies the preconditions of an action;

$1-(1-r/2n)^n$: probability that a state does not satisfy the preconditions of an action;

$(1-(1-r/2n)^n)^m$: probability that a state does not satisfy the preconditions of any action;

$1-(1-(1-r/2n)^n)^m$: probability that a state satisfies the preconditions of any action;

$(1-(1-(1-r/2n)^n)^m)^x$: probability that every initial state in the belief state Bs_T (or Bs_F) satisfies the preconditions of any action.

For the goal conditions, the g goal conditions are also selected at random from the $2n$ conditions. Then, the probability that a goal condition is not a postcondition of any action is as follows:

$c/2n$: probability that the goal condition is a postconditions of the action;

$1-c/2n$: probability that the goal condition is not a postconditions of the action;

$(1-c/2n)^m$: probability that the goal condition is not a postconditions of any action;

$1-(1-c/2n)^m$: probability that the goal condition is a postconditions of some action;

$(1-(1-c/2n)^m)^g$: probability that every goal condition is a postconditions of some action.

In the following, we will prove that the probability that every initial state satisfies the preconditions of any action and the probability that every initial state in the belief state Bs_T (or Bs_F) satisfies the preconditions of any action both are not more than σ. Because the probability that every initial state satisfies the preconditions of any action is less than the probability that every initial state in the belief state Bs_T (or Bs_F) satisfies the preconditions of any action, we only prove that the probability that every initial state in the belief state Bs_T (or Bs_F) satisfies the preconditions of any action is not more than σ.

By (5), we get

$$m \le \frac{(1-(1-r/2n)^n)(-\ln\ln 1/\sigma)}{(1-r/2n)^n}. \tag{6}$$

This is equivalent to

$$\ln\left(\frac{1}{\ln(1/\sigma)}\right) \geq \frac{m(1-r/2n)^n}{1-(1-r/2n)^n}, \quad (7)$$

$$\ln\left(\ln\left(\frac{1}{\sigma}\right)\right) \leq -\frac{m(1-r/2n)^n}{1-(1-r/2n)^n}. \quad (8)$$

Let $x = (1-r/2n)^n$. By (1), we have

$$e^{-(1-r/2n)^n/(1-(1-r/2n)^n)} \leq 1-\left(1-\frac{r}{2n}\right)^n. \quad (9)$$

From (8) and (9), we have

$$\ln\left(\ln\left(\frac{1}{\sigma}\right)\right) \leq m\ln\left(1-\left(1-\frac{r}{2n}\right)^n\right). \quad (10)$$

This is equivalent to

$$-\left(1-\left(1-\frac{r}{2n}\right)^n\right)^m \leq (\ln\sigma). \quad (11)$$

Let $x = (1-(1-r/2n)^n)^m$. By (2), we have

$$\ln\left(1-\left(1-\left(1-\frac{r}{2n}\right)^n\right)^m\right) \leq -\left(1-\left(1-\frac{r}{2n}\right)^n\right)^m. \quad (12)$$

From (11) and (12), we have

$$\ln\left(1-\left(1-\left(1-\frac{r}{2n}\right)^n\right)^m\right) \leq (\ln\sigma). \quad (13)$$

Simplify (13), and we get

$$1-\left(1-\left(1-\frac{r}{2n}\right)^n\right)^m \leq \sigma. \quad (14)$$

In addition, for

$$\left(1-\left(1-\left(1-\frac{r}{2n}\right)^n\right)^m\right)^x \leq 1-\left(1-\left(1-\frac{r}{2n}\right)^n\right)^m; \quad (15)$$

therefore, we obtain

$$\left(1-\left(1-\left(1-\frac{r}{2n}\right)^n\right)^m\right)^x \leq \sigma. \quad (16)$$

Thus, if the inequality of the theorem is satisfied, then the probability that every initial state in the belief state Bs_T (or Bs_F) does not satisfy the preconditions of any action and the probability that every initial state satisfies the preconditions of any action are at least $1-\sigma$.

Finally, we prove that the probability that a goal condition is not a postcondition of any action is not more than σ.

From (5), we get

$$m \leq \left(\frac{(2n-c)}{c}\right)\left(\ln g - \ln\ln\frac{1}{\sigma}\right). \quad (17)$$

This is equivalent to

$$\frac{-mc}{(2n-c)} \geq \ln\left(\frac{\ln(1/\sigma)}{g}\right). \quad (18)$$

By (1), we get

$$\frac{-c}{(2n-c)} \leq \ln\left(1-\frac{c}{2n}\right). \quad (19)$$

From (18) and (19), we have

$$\ln\left(\frac{\ln(1/\sigma)}{g}\right) \leq m\ln\left(1-\frac{c}{2n}\right). \quad (20)$$

This is equivalent to

$$-\left(1-\frac{c}{2n}\right)^m \leq \frac{(\ln\sigma)}{g}. \quad (21)$$

By (2), we get

$$\ln\left(1-\left(1-\frac{c}{2n}\right)^m\right) \leq -\left(1-\frac{c}{2n}\right)^m. \quad (22)$$

From (21) and (22), we have

$$\ln\left(1-\left(1-\frac{c}{2n}\right)^m\right) \leq \frac{(\ln\sigma)}{g}. \quad (23)$$

Simplify (23), and we get

$$\left(1-\left(1-\frac{c}{2n}\right)^m\right)^g \leq \sigma. \quad (24)$$

Thus, if the inequality of the theorem is satisfied, then the probability that some goal condition is not a postcondition of any action is at least $1-\sigma$.

Therefore, if the inequality of the theorem is satisfied, then the algorithm will determine that no solution exists for at least $1-\sigma$ of the instances. □

Theorem 8 presents an upper bound of the phase transition. If we can obtain a lower bound, the phase transition area of the contingent planning problems will be acquired. Therefore, in the next section, we will present the CONTINGENT PLAN-EXISTENCE algorithm and discuss the lower bound of the contingent planning problems by analyzing the algorithm.

4. Contingent Plan Existence Algorithm

In this section, we present the CONTINGENT PLAN-EXISTENCE algorithm for powerfully determining that a contingent planning instance has solutions. By analyzing the algorithm, the lower bound of the phase transition is obtained. The framework of the CONTINGENT PLAN-EXISTENCE algorithm is presented in Algorithm 2.

In the algorithm, N denotes the set of unsatisfied goal conditions, Y denotes the set of satisfied goal conditions, Bs denotes the considered belief state; add(a) denotes the adding effects of the action a, del(a) denotes the deleting effects of the action a, and applicable (Bs, a) denotes the action a applicable in the belief state Bs. The algorithm firstly initializes N, Y, and Bs, that is, $N = G$, $Y =$ NULL, and Bs $= I$. Then it checks whether all states in the considered belief states satisfy

```
Algorithm CONTINGENT PLAN-EXISTENCE (N, Y, Bs, A, O)
    flag = 0;
    for each s ∈ Bs
        if N ⊄ s
        then flag = 1;
    endfor
    if flag == 0
    then return true
    for each a ∈ A
        if applicable(Bs, a) and N ∩ add(a) ≠ ∅ and Y ∩ del(a) = ∅
        then Bs' = Bs + add(a) − del(a);
             N = N − (N ∩ add(a));
             Y = Y + (N ∩ add(a));
             CONTINGENT PLAN-EXISTENCE (N, Y, Bs', A, O);
    endfor
    for each o ∈ O
        Bs_T = Bs ∩ φ⁻(o);
        Bs_F = Bs ∩ φ⁻(o⁻);
        if Bs_T ≠ ∅ and Bs_F ≠ ∅
        then CONTINGENT PLAN-EXISTENCE (N, Y, Bs_T, A, O) ∧
             CONTINGENT PLAN-EXISTENCE (N, Y, Bs_F, A, O)
    endfor
    return don't know
end
```

Algorithm 2: Contingent plan-existence algorithm.

the goal. If each state in the considered belief state satisfies the goal, the value of *flag* will not change, which means that the instance has solutions. Otherwise, the algorithm considers whether an action can be applied to the considered belief state to ensure that the postconditions of the actions satisfy more goal conditions. If there exists such actions, the process is repeated. The algorithm finally checks whether there is an observation that can split the considered belief state into two nonnull belief states. If the observation does exist, the similar process is repeated in the two branches, and the instance has solutions only when the two branches return *true*. Except the above three cases, the algorithm cannot determine whether the contingent planning instance has solutions.

Theorem 9. *The contingent planning instance has solutions if the CONTINGENT PLAN-EXISTENCE algorithm returns true; the contingent planning instance cannot be determined whether it has solutions if the CONTINGENT PLAN-EXISTENCE algorithm returns do not know.*

Proof. Let us first discuss the case that the CONTINGENT PLAN-EXISTENCE algorithm returns *true*. When all the goal conditions are satisfied by a conditional plan from each state in the initial belief state, we can determine that the instance has solutions. Then we list the cases that the CONTINGENT PLAN-EXISTENCE algorithm returns *do not know*. (1) The considered belief state has no applicable actions. (2) The number of unsatisfied goal conditions does not decrease by executing all applicable actions. (3) One of branches that the observation splits returns *do not know*. The above three cases imply that we cannot determine whether the instance has solutions or not.

The CONTINGENT PLAN-EXISTENCE algorithm is correct, but we also cannot guarantee the completeness. In the following, we prove that almost all the contingent planning instances have solutions by analyzing the algorithm when σ is small enough (σ is a constant and $0 < \sigma < 1$). □

Lemma 10. *Consider random planning instances under the variable model expecting that d of the g goal conditions are not satisfied, if*

$$m \geq e^{(1-(1-r/2n)^n)^k/(1-(1-(1-r/2n)^n)^k)} e^{c(g-d)/n} \times \left(\left(\frac{2n}{cd}\right)+1\right)\left(\ln\frac{1}{\sigma}\right), \tag{25}$$

where n is the number of propositions; m is the number of actions; r and c, respectively, are the expected numbers of preconditions and postconditions within an action; g is the number of goal conditions; k is the number of initial states; x is the number of states of an arbitrary belief state; σ $(0 < \sigma < 1)$ is a constant, then the CONTINGENT PLAN-EXISTENCE algorithm applying some action will increase the number of satisfied goal conditions for at least $1-\sigma$ of the instances.

Proof. At first, the probability that some action will increase the number of satisfied goal conditions is given as follows:

$(1 - r/2n)^n$: probability that a state satisfies the preconditions of an action;

$1-(1-r/2n)^n$: probability that a state does not satisfy the preconditions of an action;

$(1-(1-r/2n)^n)^x$: the least probability that every state in an arbitrary belief state does not satisfy the preconditions of an action;

$1-(1-(1-r/2n)^n)^x$: probability that every state in the arbitrary belief state satisfies the preconditions of an action;

$1-(1-(1-r/2n)^n)^k$: the most probability that every state in the arbitrary belief state satisfies the preconditions of an action. Because in this paper we research on the contingent planning problem whose actions have deterministic effects, the belief state during the execution has most k states;

$(1-c/2n)^{g-d}$: probability that the postconditions of an action are consistent with the $g-d$ goal conditions already achieved;

$1-(1-c/2n)^d$: probability that the postconditions of an action achieve at least one of d remaining goal conditions.

Thus, the probability p that a particular action can be applied will not exclude any satisfied goals and will achieve at least one more goal condition which is

$$p=\left(1-\left(1-\left(1-\frac{r}{2n}\right)^n\right)^k\right)\left(1-\frac{c}{2n}\right)^{g-d}\left(1-\left(1-\frac{c}{2n}\right)^d\right). \tag{26}$$

$1-p$ is the probability that the action is missing one or more of these properties, and $(1-p)^m$ is the probability that m actions are unsatisfactory.

Because $(1-p)^m \le e^{-pm} \le e^{-\ln(1/\sigma)} = \sigma$, then there will be some satisfactory action with probability at least $1-\sigma$. The inequality is satisfied if $m \ge (1/p)(\ln 1/\sigma)$.

By (1), we get

$$\begin{gathered}1-\left(1-\left(1-\frac{r}{2n}\right)^n\right)^k \ge e^{-(1-(1-r/2n)^n)^k/(1-(1-(1-r/2n)^n)^k)}\\ \left(1-\frac{c}{2n}\right)^{g-d} \ge e^{-c(g-d)/(2n-r)} \ge e^{-c(g-d)/n}.\end{gathered} \tag{27}$$

By (3), we get

$$1-\left(1-\frac{c}{2n}\right)^d \ge \frac{cd}{2n+cd}. \tag{28}$$

Then, finally,

$$\begin{aligned}\left(\frac{1}{p}\right)\left(\ln\frac{1}{\sigma}\right) &= e^{(1-(1-r/2n)^n)^k/(1-(1-(1-r/2n)^n)^k)} e^{c(g-d)/n}\\ &\quad\times\left(\left(\frac{2n}{cd}\right)+1\right)\left(\ln\frac{1}{\sigma}\right).\end{aligned} \tag{29}$$

Thus, if the inequality is satisfied by the number of the actions, the CONTINGENT PLAN-EXISTENCE algorithm applying some action will increase the number of satisfied goal conditions for at least $1-\sigma$ of the instances. □

Theorem 11. *Consider random planning instances under the variable model if;*

$$\begin{aligned} m \ge \theta_{lb}, \quad \theta_{lb} &= e^{1/(1-(1-(1-r/2n)^n)^k)} e^{cg/n}\\ &\quad\times\left(\left(\frac{2n}{c}\right)+1\right)\left(\ln\frac{g}{\sigma}\right),\end{aligned} \tag{30}$$

where n is the number of propositions; m is the number of actions; r and c, respectively, are the expected numbers of preconditions and postconditions within an action; g is the number of goal conditions; k is the number of initial states; σ $(0<\sigma<1)$ is a constant, the CONTINGENT PLAN-EXISTENCE algorithm will find solutions for at least $1-\sigma$ of the instances.

Proof. For a contingent planning instance with g goal conditions, the number of satisfied goal conditions will be increased at most g times. If each increase occurs with probability at least $(1-\sigma)/g$, g increase (the most possible) will occur with probability at least $1-\sigma$.

Thus, Lemma 10 can be applied using σ/g instead of σ. Maximizing over the g goal conditions leads to

$$\begin{aligned}&\max_{d=1}^{g} e^{(1-(1-r/2n)^n)^k/(1-(1-(1-r/2n)^n)^k)} e^{c(g-d)/n}\\ &\quad\times\left(\left(\frac{2n}{cd}\right)+1\right)\left(\ln\frac{g}{\sigma}\right)\\ &\le e^{1/(1-(1-(1-r/2n)^n)^k)} e^{cg/n}\left(\left(\frac{2n}{c}\right)+1\right)\left(\ln\frac{g}{\sigma}\right).\end{aligned} \tag{31}$$

Therefore, if the inequality is satisfied by the number of the actions, the algorithm will find solutions for at least $1-\sigma$ of the instances. □

Theorem 11 shows that when the number of actions is not lower than θ_{lb}, the CONTINGENT PLAN-EXISTENCE algorithm can prove that nearly all the contingent planning instances have solutions.

5. Experimental Results

In this section, we take experiments on the contingent planning instances under the variable model in order to investigate the relationship between the densities (ratio of the number of actions to the number of propositions) and the effectiveness of the contingent planning instances. In the experiment, we generate a large collection of instances with three preconditions and two postconditions within an action, 2^i (i is a constant) initial states, one observation, and two goal conditions. In addition, the propositions are 6, 8, 10, 12, 14, 16, and 18, respectively. All experiments are run on a cluster of 2.4 GHz Intel Xeon machines with 2 GB memory running Linux CentOS 5.4. Figure 1 illustrates the relationship between the densities and the effectiveness of the contingent planning instances. From the results we can see that the effectiveness grows when the density increases and the numbers of propositions and goal conditions are constant. Furthermore, the experiments show that there is a transition from insoluble

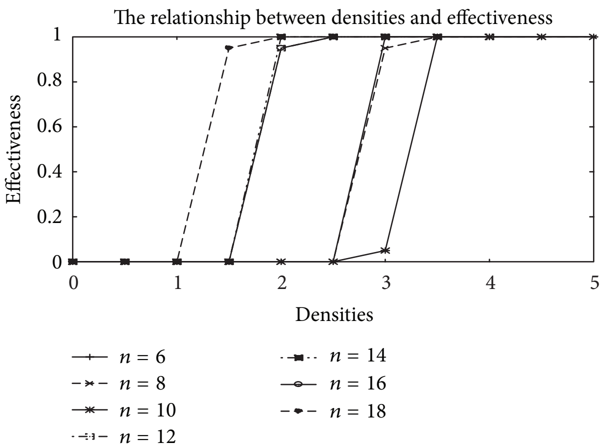

Figure 1: The Relationship between densities and effectiveness of contingent planning problems.

to soluble instances as the density grows and the transition points drop with the propositions growing.

6. Conclusions

In this paper, we present analyses for the phase transitions of the contingent planning problems. By analyzing the CONTINGENT PLAN-NONEXISTENCE algorithm and the CONTINGENT PLAN-EXISTENCE algorithm, quantitative results are obtained. If the number of actions is not greater than θ_{ub}, the CONTINGENT PLAN-NONEXISTENCE algorithm can prove that almost all the contingent planning instances have no solution; if the number of actions is not lower than θ_{lb}, the CONTINGENT PLAN-EXISTENCE algorithm can prove that almost all the contingent planning instances have solutions. The results of the experiments also show that there exist the phase transition phenomena.

Acknowledgments

This work was fully supported by the National Natural Science Foundation of China (Grants nos. 11226275 and 61070084), the Fundamental Research Funds for the Central Universities (Grant no. 11QNJJ006), the Ministry of Education (Grant no. 20120043120017), the Postdoctoral Science Foundation of China (2012M520658), and the opening fund of top key discipline of computer software and theory in Zzhejiang Provincial Colleges at Zhejiang Normal University (Grant no. ZSDZZZZXK37).

References

[1] S. A. Cook, "The complexity of theorem proving procedures," in *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, M. A. Harrison, R. B. Banerji, and J. D. Ullman, Eds., pp. 151–158, New York, NY, USA, 1971.

[2] J. M. Crawford and L. D. Auton, "Experimental results on the crossover point in satisfiability problems," in *Proceedings of the 11th National Conference on Artificial Intelligence*, W. R. Swartout, Ed., pp. 21–27, Washington, DC, USA, July 1993.

[3] B. Selman, D. G. Mitchell, and H. J. Levesque, "Generating hard satisfiability problems," *Artificial Intelligence*, vol. 81, no. 1-2, pp. 17–29, 1996.

[4] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky, "Determining computational complexity from characteristic 'phase transitions,'" *Nature*, vol. 400, no. 6740, pp. 133–137, 1999.

[5] D. Mitchell, B. Selman, and H. Levesque, "Hard and easy distributions of SAT problems," in *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI '92)*, W. R. Swartout, Ed., pp. 459–465, San Jose, Calif, USA, July 1992.

[6] E. Friedgut, "Sharp thresholds of graph properties, and the k-sat problem," *Journal of the American Mathematical Society*, vol. 12, no. 4, pp. 1017–1054, 1999.

[7] A. Frieze and S. Suen, "Analysis of two simple heuristics on a random instance of k-SAT," *Journal of Algorithms*, vol. 20, no. 2, pp. 312–355, 1996.

[8] L. M. Kirousis, E. Kranakis, D. Krizanc, and Y. C. Stamatiou, "Approximating the unsatisfiability threshold of random formulas," *Random Structures and Algorithms*, vol. 12, no. 3, pp. 253–269, 1998.

[9] I. P. Gent and T. Walsh, "The TSP phase transition," *Artificial Intelligence*, vol. 88, no. 1-2, pp. 349–358, 1996.

[10] K. Xu and W. Li, "Exact phase transitions in random constraint satisfaction problems," *Journal of Artificial Intelligence Research*, vol. 12, pp. 93–103, 2000.

[11] D. Coppersmith, D. Gamarnik, M. Hajiaghayi, and G. B. Sorkin, "Random MAX SAT, random MAX CUT, and their phase transitions," *Random Structures and Algorithms*, vol. 24, no. 4, pp. 502–545, 2004.

[12] J. Rintanen, "Phase transitions in classical planning: an experimental study," in *Proceedings of the 14th International Conference on Automated Planning and Scheduling*, S. Zilberstein, J. Koehler, and S. Koenig, Eds., pp. 101–110, British Columbia, Canada, 2004.

[13] J. Zhou, P. Huang, M. Yin, and C. Zhou, "Phase transitions of expspace-complete problems," *International Journal of Foundations of Computer Science*, vol. 21, no. 6, pp. 1073–1088, 2010.

[14] J. Zhou, M. Yin, X. Li, and J. Wang, "Phase transitions of expspace-complete problems: a further step," *International Journal of Foundations of Computer Science*, vol. 23, no. 1, pp. 173–184, 2012.

[15] H. Palacios and H. Geffner, "From conformant into classical planning: efficient translations that may be complete too," in *Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS '07)*, M. S. Boddy, M. Fox, and S. Thiébaux, Eds., pp. 264–271, Providence, RI, USA, September 2007.

[16] D. Bryce, S. Kambhampati, and D. E. Smith, "Planning graph heuristics for belief space search," *Journal of Artificial Intelligence Research*, vol. 26, pp. 35–99, 2006.

[17] M. Helmert, "The fast downward planning system," *Journal of Artificial Intelligence Research*, vol. 26, pp. 191–246, 2006.

[18] U. Kuter, D. Nau, M. Pistore, and P. Traverso, "Task decomposition on abstract states, for planning under nondeterminism," *Artificial Intelligence*, vol. 173, no. 5-6, pp. 669–695, 2009.

[19] J.-P. Zhou, M.-H. Yin, W.-X. Gu, and J.-G. Sun, "Research on decreasing observation variables for strong planning under partial observation," *Journal of Software*, vol. 20, no. 2, pp. 290–304, 2009.

[20] R. P. Goldman and M. S. Boddy, "Expressive planning and explicit knowledge," in *Proceedings of the 3rd International Conference on Artificial Intelligence Planning Systems*, B. Drabble, Ed., pp. 110–117, Edinburgh, Scotland, 1996.

[21] J. Hoffmann and R. I. Brafman, "Contingent planning via heuristic forward search with implicit belief states," in *Proceedings of the 15th International Conference on Automated Planning and Scheduling*, S. Biundo, K. L. Myers, and K. Rajan, Eds., pp. 71–80, Monterey, Calif, USA, 2005.