Using Player Models to Improve Robustness of HTN Plans in Multi-Agent Domains

Eduard Semsch, Michal Jakob, Jan Doubek and Michal Pěchouček
{eduard.semsch,michal.jakob,jan.doubek,michal.pechoucek}@agents.felk.cvut.cz
Agent Technology Center, Department of Cybernetics, Faculty of Eletrical Engineering
Czech Technical University in Prague

Abstract

When multiple agents act concurrently towards achieving their goals in a shared environment, interactions between their actions may arise, affecting the outcome of their plans. We propose an approach to planning in such environments, termed *interference robustness optimization planning* that builds upon the HTN planning paradigm and extends it with explicit consideration and optimization of plan robustness. Plan robustness is calculated from the domain model and probabilistic models of other agents in the domain. A method is presented for automatic conversion of standard HTN planning tasks into planning tasks whose output maximizes plan success probability. The method is evaluated on a test domain based on a realistic multi-agent disaster relief scenario. The empirical results indicate that the effectiveness of the method depends strongly on the predictability of other agents' behaviour and the ratio of interaction action pairs. For any values of these control parameters, the proposed method significantly outperforms standard HTN planning.

Introduction

When autonomous agents operate in a shared environment, situations arise in which the actions and strategies of individual agents interact. Examples of such scenarios include auctions, traffic control, military operations, disaster recovery etc. Situations, in which the outcome of an agent's actions depends on the actions chosen by others, are often termed *games*[1] and have been long studied from different, though recently converging perspectives. AI research has traditionally focused on devising concrete decision-making algorithms that agents could employ to pursue their objectives in game-like situations; research on game theory has focused on the analysis and descriptions of the properties of games as such.

At the same time, planning has been successfully applied to enable autonomous agents to construct sequences of actions leading towards achieving their objectives. For real-world applications, hierarchical task network (HTN) planning has proved particularly effective thanks to its ability to reduce the computational complexity of planning by allowing the incorporation of expert knowledge in the planning problem (Erol 1996; Nau *et al.* 2005). Although very successful in scenarios where the planning agent has (nearly) total control over the environment, HTN planning has provided little means to model and plan in multi-agent settings.

The Challenge of Planning in Games

The fundamental challenge presented by planning in game-like domains is the non-stationarity of the environment brought about by the concurrent activity of other agents present. In many such situations, the other agents are not willing to coordinate or even just disclose their future actions to the planning agent. Under these circumstances, it is impossible to accurately project the future evolution of the world state, an essential requirements of most planning methods, as this is no longer under the control of the sole planning agent. This problem can be alleviated if the planning agent can *anticipate* the impact that the other agents can have on its plans.

Depending on the detail and accuracy of such anticipation, planning algorithms with different optimality and complexity properties can be constructed. On one end of the spectrum, there are game-tree search algorithms which perform an exhaustive search through all future world state evolution trees generated by explicitly considering actions carried by all agents in the environment. A major disadvantage of such an approach is the exponential complexity of the search (resulting from the branching of the world state evolution tree), and also the inability to utilize sophisticated search heuristics known from planning. On the other end of the spectrum are the classical single-agent planning approaches, which do not consider other agents at all.

In this paper, we try to strike the middle ground by extending HTN with the ability to anticipate other players' actions. In doing so, we hope to gain (1) more robust plans than if other players are not considered, and (2) lower computational complexity compared to full game tree search. Specifically, we propose *interfer-*

[1]Agents participating in a game-like situation are then often termed *players*.

ence robustness optimization planning (IROP) which maximizes *plan success probability*, i.e. the probability that the plan will succeed even in the presence of possible interference of other players. In our approach, plan robustness is automatically derived from (1) the model of the planning domain and (2) models of other players' behavior (termed *player models*) which can either be designed by a domain expert or created automatically.

Organization of the Paper

We begin our exposition in Section 2 where we define individual components of the approach. In Section 3, we show how the interference robustness optimization can be implemented using the well-known JSHOP2 HTN planner. Section 4 provides the results of method's empirical evaluation. Section 5 lists alternative approaches and discusses the main differences. Section 6 summarizes the contribution of the article.

Approach

Our approach aims to extend planning with the ability to anticipate the impact of other players on the plan executed by the planning agent. In contrast to game tree search algorithms, which model such impact in detail, our approach models the impact on an aggregate level by considering an overall probability that a plan will not fail due to interference by another player.

We divide the discussion of the approach into several steps. First, we introduce a simple description framework for the problem of planning in game-like scenarios. This will allow us to define the concept of *action interference robustness* and how such robustness can be derived from the domain model and player models. Afterwards, we show how interference robustness of a whole plan can be derived from the interference robustness of its constituent actions. Finally, we outline how the robustness information can be used by the planning agent in deciding its course of action. The overall structure of the plan interference robustness aggregation process is depicted in Figure 1.

Description Framework

Domain model is a pair $(\mathcal{W}, \mathcal{A})$ where $\mathcal{W}$ is the set of all possible world states and $\mathcal{A}= \{a_1, \ldots, a_n\}$ is the set of all ground actions executable by agents in the domain. On a given domain, we can define a game as a triple $(\mathcal{W}, \mathcal{A}, \mathcal{P})$ where $\mathcal{P} = \{P_0, P_1, \ldots, P_n\}$ is the set of players in the game; P_0 denotes the planning player – the rest of the players are termed *non-planning players* to emphasize that their decision-making process is not the focus of the investigation.

We say an action $a_i \in \mathcal{A}$ *interferes* with action $a_j \in \mathcal{A}$ if the execution of a_i causes a_j to fail when performed simultaneously. Formally, we define the *action interference indicator* as a predicate

$$\begin{cases} I(a_i, a_j) = false & \text{if } a_i \text{ does not affect } a_j \\ I(a_i, a_j) = true & \text{if } a_i \text{ affects } a_j \end{cases} \quad (1)$$

Further, we define the *player model* of player $P \in \mathcal{P}$ as a probability distribution

$$\pi_P(a|w) \quad (2)$$

specifying the probability that player P performs the action a in the world state $w \in \mathcal{W}$.

Action Interference Robustness

We can now define *mutual action interference robustness*

$$r(a, a'|w) \quad (3)$$

as the probability that action $a \in \mathcal{A}$ will not be interfered with by action $a' \in \mathcal{A}$ when performed simultaneously in the world state $w \in \mathcal{W}$.

Further, we define the *action interference robustness*

$$r(a|w) \quad (4)$$

as the probability that action $a \in \mathcal{A}$ will not be interfered with by any action when performed in the world state $w \in \mathcal{W}$.

We can derive the action interference robustness from the domain and player models as follows

$$\begin{aligned} r(a|w) &= \prod_{a' \in \mathcal{A}} r(a, a'|w) = \\ &= \prod_{a' \in \mathcal{A}, I(a,a')} r(a, a'|w) \prod_{a' \in \mathcal{A}, \neg I(a,a')} r(a, a'|w) \end{aligned} \quad (5)$$

Here the second product represents the probability that none of the non-interfering actions will interfere, which is equal to one. Continuing with the first product yields

$$\begin{aligned} r(a|w) &= \prod_{a' \in \mathcal{A}, I(a,a')} r(a, a'|w) = \\ &= \prod_{a' \in \mathcal{A}, I(a,a')} \prod_{P \in \mathcal{P} \setminus P_0} (1 - \pi_P(a'|w)) \end{aligned} \quad (6)$$

In the above, we assume that an action fails if at least one action which affects it is performed by at least one player.

Plan Robustness Aggregation

In the interference robustness optimizing planning (IROP), we are interested in evaluating the robustness of whole plans, denotes as $R(a_1, a_2, ..., a_n|w)$ where $a_1, \ldots, a_n$ are the plan's constituent actions and $w \in \mathcal{W}$ is that state of the world from which the plan is executed.

The way the probability of successful plan execution is aggregated from the robustness of its constituent actions depends strongly on the structure of the plan. In the case of a totally ordered plan, where each action has to be successfully performed before the next action can start, we can calculate the overall plan success probability as

$$\begin{aligned} R(a_1, ..., a_n|w_0) &= \\ &= r(a_1|w_0) r(a_2|w_1) \ldots r(a_n|w_{n-1}) \end{aligned} \quad (7)$$

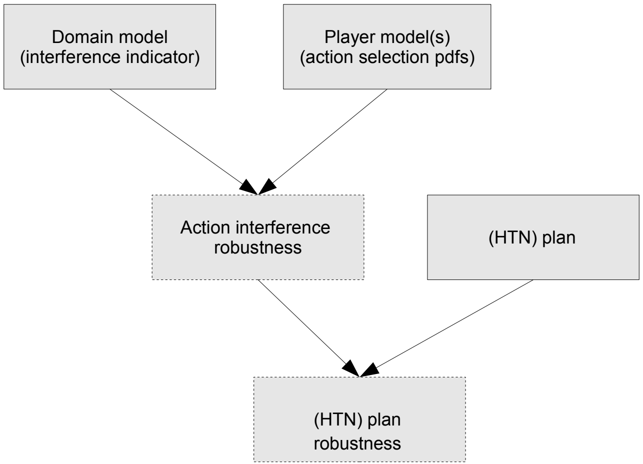

Figure 1: Interference robustness aggregation process. Solid line boxes are the input information; dashed line boxes are the derived information

where w_0 is the initial state of the world and w_i are the states of the world at the time action a_{i+1} is performed. Note that this aggregation scheme fits also partially ordered plans and plans that are intended to be executed in parallel, as long as the independence of robustness $r(a_i|w_j)$ and $r(a_k|w_l)$ is maintained. This holds because no matter how the plan is executed, all the actions have to be performed successfully. However, it holds only for the case where there are no redundancies in the plan. If the planner purposefully inserts redundant actions to increase the robustness of the plan, the aggregation scheme becomes more complex.

Interference Robustness Handling

We have shown how the robustness of a plan can be derived from the domain model and other players' models. The ability to calculate plan robustness gives the planning agent extra information it can utilize in selecting its course of action. Such information can be used in several ways:

- **robustness thresholding** – the planning agent searches for a plan whose robustness exceeds a specific threshold
- **robustness maximization** – the planning agent searches for a plan with maximum robustness
- **robustness optimization** – the planning agent does not optimizes the robustness alone but considers it in combination with plan utility (less robust plans can be preferable if their potential pay-off is high)

Although there is a very close connection between robustness thresholding and robustness maximization, the former may be easier to implement with existing planning algorithms, in particular if the robustness threshold is not stated too tightly. Robustness optimization requires that a plan utility function is defined. In the rest of the paper, we focus on robustness thresholding.

Implementation

We have implemented the IROP approach using the HTN planner JSHOP2[2].

JSHOP2

JSHOP2 is a widely-used HTN planner which has been successfully applied to numerous planning domains. The core of JSHOP2's search algorithm is the *ordered task decomposition* search-control strategy that reduces the uncertainty about the world state during the search. The reduction enables the use of highly expressive language features including external function calls (Nau *et al.* 2005). The external function calls are used to retrieve data from sources outside the planner and to perform operations on the world state using algorithms implemented in arbitrary languages. The inputs of the planner are the task-network describing the domain with the added background knowledge, and the problem definition describing the initial state and the tasks to be solved.

Extending JSHOP2 with Robustness Handling

We have developed a JSHOP2-based implementation of IROP, specifically interference robustness thresholding. The approach does not require any modification of the planner. It reuses the task network of the planning problem and extends it with interference robustness handling logic. The modification of the task network is straightforward and can be done automatically.

Specifically, the following steps are performed:

- Operator effects are extended with robustness aggregation commands. Robustness aggregation relies on the values of action interference robustness which are obtained through a call to an external model; the model evaluates the robustness $r(a|w)$ for the current state according to equation (6).

```
(
    <delete-list>
    (success-prob ?sp)
)
(
    <add-list>
    (success-prob (call * ?sp ?risk))
)
```

- Preconditions are added to all operators in the task-network to ensure that when the operator is applied, the overall robustness of the plan does not drop under the specified limit threshold.

```
(:operator (![name] <?parameters>)
    (
        <preconditions>
        (assign ?risk (call [RiskProcedure] <?parameters>))
        (success-prob ?sp)
        (success-prob-threshold ?threshold)
        (call < (call * ?sp ?risk) ?threshold)
    )
```

[2] http://sourceforge.net/projects/shop

Experimental Evaluation

IROP has been empirically evaluated on an experimental domain based on a realistic multi-agent disaster relief scenario. The experiments have been implemented in a simulation testbed based on the A-globe multi-agent platform[3].

Experimental Domain

The experimental domain involves multiple players with different objectives operating on a small island affected by a tsunami. The government player P_{GOV} uses its units to rebuild the infrastructure and restore order in the northern tsunami-affected regions of the island. The separatist player P_{SEP} tries to disrupt the government's recovery operation and to increase its control over the affected regions. Finally, the humanitarian organisation player P_{NGO} is neutral, trying to organize transport of humanitarian supplies to the affected region regardless of the political situation. A key component of the operations of all players on the island are logistic supply chains through which resources required for performing actions are delivered to the required locations. The supply chains can be disrupted by actions of other players, giving rise to a number of ways in which the plans of individual players can interfere. See Figure 2 for a graphical overview of the domain.

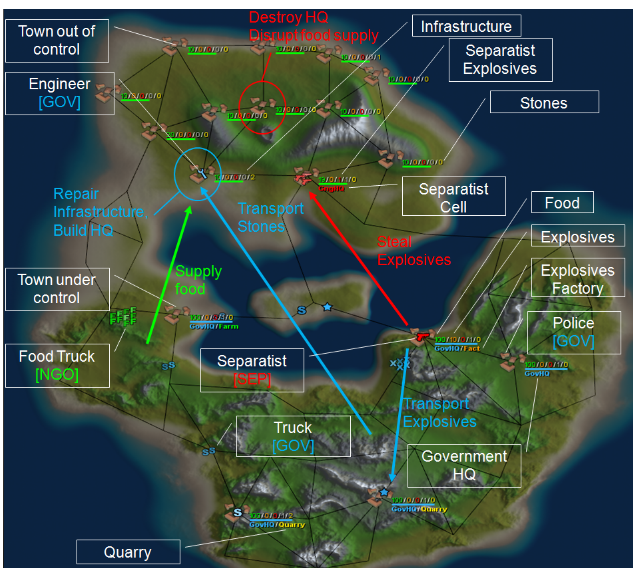

Figure 2: Overview of the Tsunami recovery experimental domain. Units, actions and plans are depicted for each player (blue = government P_{GOV}, red = separatist P_{SEP}, green = humanitarian organisation P_{NGO})

Domain states The state of the game is given by propositional and numerical variables. The set of all states $\mathcal{W}$ constitutes all the valid assignments to these variables, in particular locations of units, load of units, characteristics of cities and the amount of commodities they hold. More specifically, each player has a set of resources. These are commodities (given in integer amounts), and units that can either carry commodities or serve as security forces. There are a number of locations connected by roads; the units can move between the locations. In some locations there are cities in which commodities are stored and where some commodities can be transformed into other commodities. The cities have characteristics which can change over time which influence what actions can be performed in cities and whether the commodities are transformed into other commodities. An overview of game state variables is given in Table 1.

Object	Associated state variables
City	farm presence, quarry presence, explosives factory presence, amount of food, amount of explosives, amount of stones, infrastructure level, government HQ presence, separatist camp presence
Unit	position, food load, explosives load, stones load (depends on unit type)

Table 1: Types of state variables in the Tsunami recovery game. State variables associated with cities and units are distinguished.

Actions Actions are defined using a set of action schemata, which can have symbolic and numerical parameters. All valid assignments of parameters in the action schemata constitute the set of all ground actions $\mathcal{A}$. The list of action schemata is given in Table 2.

Interference Indicator Most of the interference in the domain takes place between the transport and steal actions. Specifically, action $transportX(C_1, C_2)$ interferes with action $stealX(C_3, C_4, S)$ if there is an overlap between the transport route $C_1 \leftrightarrow C_2$ and the route between $C_3 \leftrightarrow C_4$ along which the separatists try to steal the transported commodity. Furthermore, action $transportXInConvoy(C_1, C_2, S_1)$ interferes with action $stealX(C_3, C_4, S_2)$ if the previously mentioned condition on segment overlap holds and $S_2 > S_1$ (i.e. the number of robbing units is greater than the size of the accompanying security force). No other actions can directly interfere in the Tsunami recovery domain.

Players As already mentioned, there are three players in the domain, each having a number of units at their disposal. The players use a decision-making mechanism to construct actions and action sequences for their units in order bring the world to a (more) desirable state. The government player uses HTN-based decision making, both in the standard version and the IROP version. The other two players employ rule-based decision making.

[3] http://agents.felk.cvut.cz/aglobe/

Schemata
$loadFood(C, N)$
$loadStones(C, N)$
$loadExplosives(C, N)$
$unloadFood(C, N)$
$unloadStones(C, N)$
$unloadExplosives(C, N)$
$buildGovernmentHQ(C)$
$destroyGovernmentHQ(C)$
$buildSeparatistCamp(C)$
$destroySeparatistCamp(C)$
$createSuicideBomber(C)$
$repairInfrastructure(C, N)$
$stealFood(C_1, C_2, S)$
$stealStones(C_1, C_2, S)$
$stealExplosives(C_1, C_2, S)$
$transportStones(C_1, C_2, N, S)$
$transportFood(C_1, C_2, N, S)$
$transportExplosives(C_1, C_2, N, S)$
$transportExplosivesInConvoy(C_1, C_2, N, S)$
$transportStonesInConvoy(C_1, C_2, N, S)$

Table 2: Action schemata in the Tsunami recovery domain. C/C_i means city, N amount of commodity, and S number of security forces. Some actions can only be performed by units belonging to a particular player (e.g. $createSuicideBomber(C)$ is only available to the Separatist player)

Experiment Configuration

To test the developed approach, we have created a representative scenario in the experimental domain that provides enough possible plans to the planning player to choose from and enough actions to the other players to interfere with the plan. In all experiments, we focus on the performance of the government player P_{GOV} which uses the IROP method; the other two players are controlled by a set of action selection rules.

For each experiment configuration, we define the following two properties:

- **player non-determinism** captures how well a behaviour of a player P can be anticipated. It is defined as

$$\overline{H_P} = -\frac{1}{|\mathcal{W}|}\sum_{w\in\mathcal{W}}\sum_{a\in\mathcal{A}}\pi_P(a|w)\log_2\pi_P(a|w) \quad (8)$$

  i.e. as an average entropy of actions selection distributions in the respective player model. The higher the entropy, the more difficult it is to predict which actions the player will carry out in a given state[4]. We define the overall average player non-determinism $\overline{H}$ as a sum of average non-determinism values for all players other than the planning player (Separatist and Humanitarian organisation player in our domain).

- **interference density** Interference density measure the ratio of potential intereferencing action pairs, i.e.,

$$d_I = \frac{\|\{(a_1, a_2) \in \mathcal{A}^2 | I(a_1, a_2)\}\|}{\|\mathcal{A}^2\|} \quad (9)$$

  where a_1 and a_2 are ground actions.

[4]This does not mean that the player chooses its actions randomly. There might be a deeper order in its behaviour which is not representable by the action selection player model.

Parameter	Values
Player non-determinism	0.00, 0.29, 0.47, 0.61, 0.72, 0.81, 0.88, 0.93, 0.99
Interaction density	0.26, 0.30, 0.35, 0.50

Table 3: Values for experiment control parameters. A total 32 different configurations, one for each combination of the two control parameters, have been experimentally evaluated. Ten runs has been performed for each configuration.

Evaluation Criteria

At the beginning of each experimental run, the planning player is given the initial state of the world and the models of the other players in the form of $\pi_i(a|w)$[5] and produces a plan that is executed in the simulator. The plan is not changed during the experimental run. If any of the actions in the plan fails, the goals are not reached and the plan is considered unsuccessful. Note that the outcome of an experiment run can differ even for the same configuration because of inherent noise in the decision making of rule-based players in the domain. For each experiment configuration, we therefore perform multiple (ten) experimental runs. We then calculate the *plan success rate* as the ratio between the number successful plan executions and the total number of runs for the given configuration; the plan success rate is the empirical measure of plan robustness defined in the 'Approach' section.

Results

We have evaluated the plan robustness for a total of 32 configurations, eight and four different settings for overall player non-determinism $\overline{H}$ and interference density d_I, respectively. See the values of the control parameters in Table 3. Note that due to the way decision-making algorithms of the Separatist and Humanitarian organisation players are designed, it is not possible to directly control the entropy of their action selection distributions. Therefore, the distribution of the values of the non-determinism parameter across configurations is not even. The same applies to the interference density

[5]In all of the experiments, we use player models which accurately reflect the probability with which the rule-based government and humanitarian organisation players choose their actions.

which could be varied only indirectly and within a limited range allowed by the constraints of the experimental domain.

We have performed three groups of experiments. The first group of experiments compares the performance of IROP with standard HTN planning. The second group of experiments evaluates the impact of player non-determinism on IROP performance; the third group of experiments evaluates the effect of interference density.

Comparison with Standard HTN Planning First, we compare the robustness of plans produced by the Government player using the standard HTN planning and the enhanced IROP planning. When averaged over all scenario configurations, the plan success rate values are 0.34 for HTN and 0.71 for IROP, clearly demonstrating the benefit of explicitly considering robustness in the planning.

Effect of Player Non-Determinism We further studied the dependency of player non-determinism on the plan robustness of both the HTN and IROP-based Government player. The results given in Figure 3 show clearly that the performance of both algorithms decreases with the rising non-determinism. For the IROP-based player, this can be explained as due to the fact that increased non-determinism in other players' strategies makes it more difficult for the planning player to avoid actions that can be disrupted. Decrease in the case of the HTN-based player is for a different reason. It is due to the fact that the player's task network has been designed to cope well with the deterministic versions of other players. The more the players deviate from their deterministic versions, the worse the task network's performance.

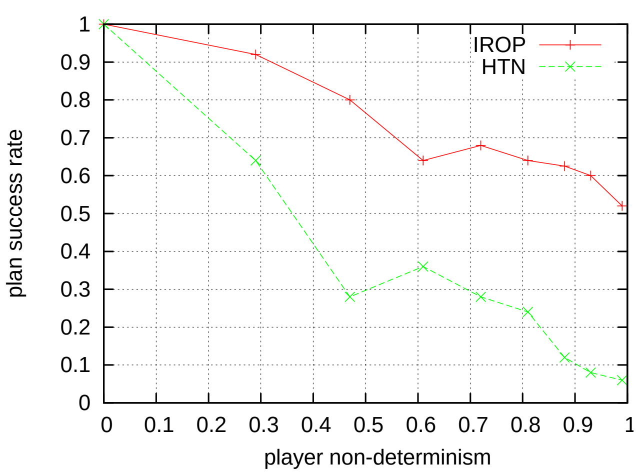

Figure 3: Dependency of plan success rate on the overall non-determinism of players.

Effect of Interference Density Finally, we evaluate the effect of action interference density. The results given in Figure 4 partially confirm our hypothesis that increasing density of interactions between the players' actions renders the approach less effective, as it makes it more difficult for the Government player to avoid disruptive actions of the other players, in particular the Separatist. The dependency as displayed by the experiments, however, is not monotonous. A possible explanation is divergence between the potential of interference as modelled by the interference indicator (from which the density is derived) and the frequency of interferences that actually arise in the simulation. This is an important topic for further exploration.

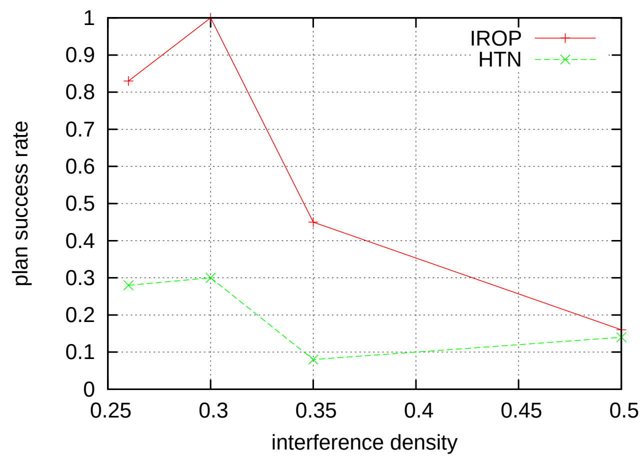

Figure 4: Dependency of plan success rate on action interference density.

Discussion

The experiments indicate that both the player non-determinism and action interference density have a strong impact on the performance of the HTN-based interference robustness thresholding planning. This is actually not surprising and can be explained by a closer examination of the method. The IROP method works best if the probability of action's successful performance can be accurately determined, because it allows the planner to either rely on the action or to avoid it. This is the case if action interference robustness is either close to zero (the action is very likely to fail) or to one (the action is very likely to succeed). Inspection of equation (6) reveals that this is more likely the case when the number of interfering action pairs is low or when the players select their actions deterministically (in a given world state). High interference densities act as an "amplifier" for player non-determinism, making the determination of successful action execution less certain. We can conclude that the mechanism proposed is suitable for domains where interactions between actions are sparse or where players behave deterministically.

Related Work

We discuss relevant prior work in two steps. First we overview general approaches to dealing with uncertainty in planning problems, both arising from the property of the environment itself and/or from the activity of other agents; second we focus specifically on utilizing HTN planning in uncertain domains.

Planning Under Uncertainty

Decision making in game-like scenarios has been traditionally solved using game tree search algorithms such as minimax (Korf 1990) and its many variants. The main advantage of this approach is that all relevant game evolutions can be inspected and thus the strategy provided is optimal. This advantage, however, comes at the price of large computational complexity; the number of possibilities that have to be searched increases exponentially with the length of the game, the possible moves and possible states. For this reason, the applicability of game tree search to larger-scale domains, such as the Tsunami recovery domain presented in this paper, is greatly limited.

Probabilistic planning has been studied as another approach explicitly considering uncertainty in the planning problem (Majercik & Littman 1998; Kuter & Nau 2005; Bryce 2006). The uncertainty can be in imprecise knowledge about the initial state and/or in uncertain outcomes of actions. In contrast to our approach, probabilistic planning considers single-agent domains in which the uncertainty is included in the domain model rather than being modelled as an outcome of simultaneous activity of other agents in the domain. Based on desired planning goals, probabilistic planning forms action-selection policies through the use of reinforcement learning algorithms such as value iteration and policy iteration. Attributing all non-determinism to the environment makes the probabilistic planning conceptually simpler; at the same time, however, it makes the planning computationally more expensive than in our approach, which, by explicitly modeling players and the interference they generate, allows searching through the space of all possible plans in a more selective manner.

HTN Planning in Uncertain Environments

HTN planning has also been adapted to solve a variety of non-standard problems. One prominent example is the Bridge Baron computer bridge player that utilized an adapted HTN planner to search effectively the vast game-tree of the game (Smith, Nau, & Throop 1998).

In recent years there has been an interest in using HTN planners for computer games. Preliminary work on this subject is described in (Kelly 2007; Muñoz-Avila & Fisher 2004).

Conclusion

We have presented an approach for increasing the robustness of plans in multi-agent domains, in which interactions between agents' actions and plans may arise due to simultaneous activity of the agents in a shared environment. We have shown how, in such environments, plan robustness can be calculated from the model of the domain and action selection models of other agents. We then proposed a method termed *interference robustness optimization planning* which explicitly considers plan robustness in order to produce plans less prone to failure due to other player's interference. We have described a particular implementation of the method based on the well-known HTN planner JSHOP2. The implementation works by augmenting standard task networks with additional robustness calculation and handling logic. The augmentation can be made in an automatic way and does not require modification of the planner.

We have evaluated the approach on a test domain based on a realistic multi-agent disaster recovery scenario. Experimental results have shown that if other agents in the environment behave deterministically or if the number of interfering actions is small, the method significantly improves the robustness of plans compared to standard HTN planning. With increasing nondeterminism of the other agents or increasing number of interfering action pairs, the advantage of the method diminishes.

References

Bryce, D. 2006. POND: The Partially-Observable and Non-Deterministic Planner.

Erol, K. 1996. *Hierarchical task network planning: formalization, analysis, and implementation.* Ph.D. Dissertation, College Park, MD, USA.

Kelly, J.-P. 2007. Planning with Hierarchical Task Networks in Video Games.

Korf, R. E. 1990). Depth-limited search for real-time problem solving. *Real-Time Systems.*

Kuter, U., and Nau, D. S. 2005. Using domain-configurable search control for probabilistic planning. In *AAAI*, 1169–1174.

Majercik, S. M., and Littman, M. L. 1998. Maxplan: A new approach to probabilistic planning. In *Proceedings of the Fourth International Conference on Artificial Intelligence Planning Systems*, 86–93. AAAI Press.

Muñoz-Avila, H., and Fisher, T. 2004. Strategic planning for unreal tournament bots. In *In Proceedings of AAAI-04 Workshop on Challenges on Game AI.* AAAI Press.

Nau, D.; Au, T.-C.; Ilghami, O.; Kuter, U.; Munoz-Avila, H.; Murdock, J. W.; Wu, D.; and Yaman, F. 2005. Applications of SHOP and SHOP2. *IEEE Intelligent Systems* 20(2):34–41.

Smith, S.; Nau, D.; and Throop, T. 1998. Computer Bridge: A Big Win for AI Planning. *AI Magazine* 19(2).