ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2000
Seria: AUTOMATYKA z. 131 Nr kol. 1476

Adam GAŁUSZKA, Andrzej ŚWIERNIAK
Politechnika Śląska

PLANOWANIE W WARUNKACH DOSTĘPU DO NIEPEWNEJ I NIEPEŁNEJ INFORMACJI

Streszczenie. W artykule rozważane są problemy planowania zadań i osiągania celów w sensie *sztucznej inteligencji*. Szczególnie analizowane jest modelowanie problemu planowania jako systemu zawierającego niepełną i niepewną informacje. W części artykułu przedstawione są dwa klasyczne podejścia do planowania w warunkach niepewności spotykane w literaturze. Wyjaśnienia jednego z nich dokonano na przykładzie. Inne podejście do modelowania systemu, w którym niepewna informacja reprezentowana jest poprzez „stopień pewności", jest przedstawione w pozostałej części artykułu. Przykład wyjaśnia istotę działania takiego systemu w środowisku tzw. „świata klocków". We wszystkich przypadkach problem planowania zamodelowany został przy użyciu reprezentacji STRIPS.

PLANNING IN THE PRESENCE OF UNCERTAIN AND INCOMPLETE INFORMATION

Summary. In this paper planning and goal persuit problems in artificial intelligence are considered. Especially modelling of planning problem which contain incomplete and uncertain information is analysed. In the first part of the paper two classical approaches to planning under uncertainty are presented. One of these aproaches is explained by the example. Yet another approach to modelling system in which uncertainty is represented by „certain degree" is presented in the second part of the paper. The essence of this approach is explained by the example in block world environment. In all cases planning problems have been modelled using STRIPS representation.

1. Wstęp

Współczesne obszary zainteresowań *sztucznej inteligencji* można by podzielić na kilka części [13]:

- przetwarzanie języka naturalnego;
- rozpoznawanie obrazów;
- wizja komputerowa;
- mechatronika (inteligentna robotyka);
- systemy ekspertowe.

Próby implementacji robota inteligentnego pociągają za sobą konieczność analizy i rozwiązywania takich zadań, jak percepcja i rozumienie otoczenia, osiąganie celu, planowanie

zadań, wykonywanie zadań i monitorowanie ruchów robota. Osiąganie celu oraz planowanie zadań należą do głównych problemów mechatroniki.

W sposób ogólny można powiedzieć, że planowanie zadań w *sztucznej inteligencji* jest wnioskowaniem mającym na celu znalezienie sekwencji operacji, która pozwala na osiągnięcie sytuacji docelowej wychodząc z danej sytuacji początkowej [2]. Poniższy przykład w prosty sposób ilustruje istotę takiego planowania [14].

Przykład 1

Rozważmy następujące operatory:

1: *Bob-has-1000DM* ⇒ ¬*Bob-has-1000DM* and *Bob-in-Kyoto*
2: *Bob-has-1000DM* ⇒ ¬*Bob-has-1000DM* and *Bob-in-Paris*
3: *food-in-Kyoto* and *Bob-in-Kyoto* and *Bob-has-5DM* ⇒ ¬*Bob-has-5DM* and ¬*Bob-hungry*
4: *food-in-Paris* and *Bob-in-Paris* and *Bob-has-5DM* ⇒ ¬*Bob-has-5DM* and ¬*Bob-hungry*

W sytuacji początkowej fakty *food-in-Kyoto, Bob-has-1000DM, Bob-has-5DM, Bob-hungry* są prawdziwe, reszta jest fałszywa. Symbol "¬" oznacza zaprzeczenie faktu. Celem jest najedzenie się Boba, a więc fakt ¬*Bob-hungry*.
Sekwencja operacji rozwiązujących problem to oczywiście 1, następnie 3.

Przez robota inteligentnego rozumiemy raczej robota autonomicznego, zdolnego do analizy otoczenia, planowania i wykonywania ruchów w otoczeniu niezdeterminowanym [21]. Jądrem implementacji takiego robota jest system, w którym zapisana wiedza oraz określona metoda poszukiwania rozwiązań problemów pozwala na generowanie planu ruchów robota.

W ogólnym przypadku powyższy problem jest bardzo złożony. Wynika to ze złożoności i sposobu monitorowania otoczenia robota, liczby stopni swobody robota wynikających z jego konstrukcji mechanicznej, liczby poziomów sterowania robotem, złożoności inteligentnego oprogramowania. Wszystko to spowodowało, że inteligentna robotyka stała się dziedziną interdyscyplinarną [21, 13]. Praktyczne implementacje inteligentnych systemów wskazują, że zarówno procesy planowania zadań, jak i planowania trajektorii powinny uwzględniać niepewność dostępnej informacji [21].

Klasyczny problem planowania zadań robota polega na przekształceniu bieżącego stanu świata do pożądanego, docelowego stanu świata. Jedną z pierwszych reprezentacji problemów planowania (rok 1960) był tzw. system STRIPS [12], który do dnia dzisiejszego wydaje się najbardziej popularny [10]. Istnieje wiele różnych metod służących do rozwiązywania problemów planowania [11, 8, 10, 16, 1, 5, 6, 14, 18]. Problem złożoności obliczeniowej tych algorytmów jest także często poddawany dyskusji [11, 16, 1, 6]. Zwykle algorytmy rozwiązujące problemy planowania są NP złożone [3, 7].

Problem planowania STRIPS (*propositional STRIPS planning problem*) składa się z czwórki $<P, O, I, G>$ (2), gdzie:

- P jest skończonym zbiorem podstawowych formuł, zwanych warunkami (*conditions*),

- O jest skończonym zbiorem operatorów, gdzie każdy operator $o \in O$ przyjmuje formę $o^+, o^- \Rightarrow o_+, o_-$, gdzie
- $o^+ \subseteq P$ są pozytywnymi warunkami stosowalności operatora (*positive preconditions*),
- $o^- \subseteq P$ są negatywnymi warunkami stosowalności operatora (*negative preconditions*),
- $o_+ \subseteq P$ są pozytywnymi skutkami zastosowania operatora (*positive postconditions*),
- $o_- \subseteq P$ są negatywnymi skutkami zastosowania operatora (*negative postconditions*),
- $I \subseteq P$ jest stanem początkowym zadania (*initial state*),
- $G = \langle G_+, G_- \rangle$ jest spełnialną koniunkcją pozytywnych i negatywnych warunków, w której $G_+ \subseteq P$ są pozytywnymi warunkami a $G_- \subseteq P$ są negatywnymi warunkami.

Każdy stan problemu planowania może być opisany jako podzbiór $S \subseteq P$, w taki sposób, że warunek $p \in P$ jest spełniony (prawdziwy) w opisie stanu, jeśli $p \in S$ i niespełniony (fałszywy) w przeciwnym wypadku. Zbiór I precyzuje, które warunki są prawdziwe, a które fałszywe w stanie początkowym problemu, tzn. warunek $p \in P$ jest prawdziwy w stanie początkowym, jeśli $p \in I$ lub fałszywy w przeciwnym przypadku. Zbiór G precyzuje cel, tzn. $S \subseteq P$ jest stanem docelowym problemu, jeśli S spełnia G, tzn. jeśli $G_+ \subseteq S$ i $G_- \cap S = \emptyset$.

Efekt zastosowania skończonej sekwencji operatorów do stanu S może być opisany poprzez funkcję *Result*, która jest zdefiniowana następująco (symbol "\" oznacza różnice zbiorów):

$$Result(S, \langle\rangle) = S,$$

$$Result(S, \langle o \rangle) = (S \cup o_+) \setminus o_-, \text{ jeśli } o^+ \subseteq S \wedge o^- \cap S = \emptyset;$$

$$S, \text{ w przeciwnym przypadku,}$$

$$Result(S, \langle o_1, o_2, \ldots, o_n \rangle) = Result(Result(S, \langle o_1 \rangle), \langle o_2, \ldots, o_n \rangle).$$

Z definicji funkcji *Result* wynika, że każdy operator może być zastosowany do każdego stanu, natomiast tylko ten operator daje efekt, dla którego spełnione są warunki stosowalności (*positive and negative preconditions*). Dany operator może występować wiele razy w sekwencji operatorów tworzących plan.

Skończona sekwencja operatorów $\langle o_1, o_2, \ldots, o_n \rangle$ jest rozwiązaniem problemu planowania, jeśli stan wynikający z zastosowania funkcji $Result(I, \langle o_1, o_2, \ldots, o_n \rangle)$ jest stanem docelowym problemu.

Systemy planowania w *sztucznej inteligencji* z wykorzystaniem reprezentacji STRIPS znajdują obecnie zastosowania praktyczne w różnych implementacjach systemów inteligentnych. Jako przykładowe można by podać: autonomicznego robota inteligentnego poruszającego się i rozwiązującego problemy osiągania celu w otoczeniu niezdeterminowanym [15] oraz system planowania i osiągania celów sterujący sondą kosmiczną *Deep Space One*, która została wysłana w przestrzeń kosmiczną w 1998 [19].

2. Planowanie w warunkach niepewności

Reprezentacja rzeczywistego otoczenia w systemie planowania jest niekompletna, niepewna i aproksymowana [15]. Klasyczne algorytmy planujące poszukują planu rozwiązującego problem przy nierealistycznym założeniu o pełności i pewności informacji w początkowym stanie problemu. Stąd potrzeba poszukiwania i rozwijania systemów planowania w warunkach niepewności. W literaturze spotykamy dwa podstawowe podejścia do planowania zadań w warunkach niepewności [17]. Pierwsze z nich, zwane planowaniem warunkowym (ang. *contingency planning*), polega na generowaniu planu, w którym niektóre działania są wykonywane warunkowo, w zależności od dodatkowej informacji pochodzącej z sensorów. W tym przypadku niepewność dostępnej informacji jest redukowana przy wykorzystaniu informacji uzyskiwanej z sensorów. Jeśli pozyskana w ten sposób informacja jest relatywnie tania, natomiast wykonywanie działań jest drogie lub też skutki tych działań mogą być niebezpieczne, to planowanie warunkowe znajduje zastosowanie. Jeśli uzyskanie informacji z sensorów jest drogie, trudne lub wręcz niemożliwe, to planowanie warunkowe jest niewystarczające. Możemy wtedy wykorzystać drugie podejście do planowania w warunkach niepewności – planowanie dostosowujące się (ang. *conformant planning*).

Planowanie dostosowujące się polega na generowaniu planu bez wykorzystania dodatkowych informacji pochodzących z sensorów, który prowadzi do pożądanej sytuacji docelowej niezależnie od początkowego, niepewnego stanu. Jako przykład można by podać pacjenta, który cierpi na jedną z kilku chorób, ale nie można sprecyzować, na którą. Planowanie dostosowujące się polega w tym przypadku na znalezieniu terapii lekowej, która prowadzi do wyleczenia pacjenta niezależnie od choroby, na którą cierpi.

W planowaniu warunkowym operatory (działania) wykonywane są warunkowo, w zależności od informacji uzyskanej z sensorów. Model takiego operatora odbiega od definicji operatora w systemie STRIPS. Dla przykładu rozpatrzmy działanie polegające na przyjmowaniu lekarstwa (operator *Medicate*).

Medicate: preconditions: effects: (when I ¬I)
(when ¬H D)

gdzie „I" oznacza, że pacjent jest zarażony (*Infected*), „H" oznacza, że pacjent nie jest odwodniony (*Hydrated*), „D" oznacza śmierć pacjenta (*Dead*). Symbol "¬" oznacza zaprzeczenie faktu (odpowiada operacji logicznej *NOT*) Operator *Medicate* jest operatorem warunkowym, w którym lista warunków stosowalności (*preconditions*) jest pusta, natomiast efekty zastosowania operatora zależą od spełnienia pewnych warunków.

Najprostszym sposobem przekształcenia tak zdefiniowanego operatora do systemu STRIPS jest skonstruowanie kilku różnych operatorów zgodnych z systemem STRIPS,

których sumaryczne działanie odpowiada oryginalnemu operatorowi [20]. Nawiązując do przykładu *Medicate* konstruujemy 4 rozłączne operatory STRIPS:

Med_1:	pre: (I and H)	eff: ¬I
Med_2:	pre: (I and ¬H)	eff: (¬I and D)
Med_3:	pre: (¬I and H)	eff: brak
Med_4:	pre: (¬I and ¬H)	eff: D

Tak zmodyfikowane operatory warunkowe mogą być zostosowane do poszukiwania rozwiązań części problemów planowania w warunkach niepewności z wykorzystaniem znanych algorytmów planujących (np.: GRAPHPLAN[1]).

Przypadek, w którym informacje pochodzące z sensorów są nieosiągalne, jest bardziej złożony. Do wyjaśnienia tego problemu niezbędne jest wprowadzenie relacji wzajemnego wykluczania (ang. *mutual exlusion relation*) zarówno operatorów, jak i warunków.

Definicja. Dwa operatory są wzajemnie wykluczające się, jeśli:

1) zastosowanie jednego operatora powoduje likwidację warunków stosowalności (mówimy wtedy o interferencji oparatorów) lub efektów działania drugiego (mówimy wtedy o niezgodnych efektach operatorów),
2) warunki stosowalności dwóch operatorów są wzajemnie wykluczające się.

Definicja. Dwa warunki są wzajemnie wykluczające się, jeśli wszystkie działania prowadzące do osiągnięcia tych warunków są wzajemnie wykluczające.

Przykład 2 wyjaśnia problem planowania dostosowującego się.

Przykład 2

operatory: Med_1 - Med_4 zdefiniowane jak wyżej oraz operator:

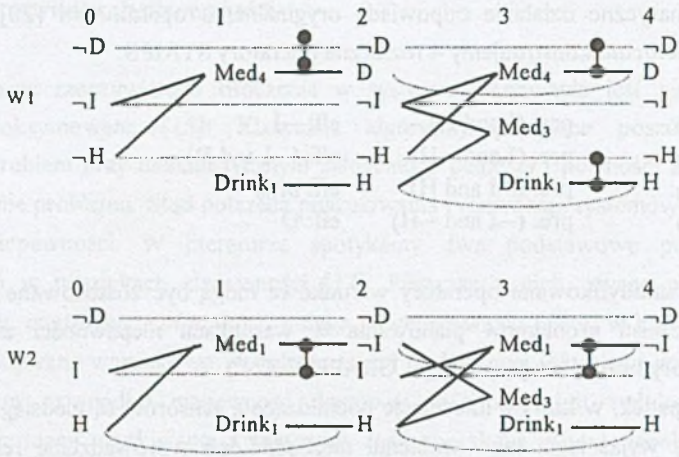
$Drink_1$: pre: eff: H

stan początkowy: {¬I and ¬H and ¬D} lub {I and H and ¬D}

stan końcowy (cel): ¬I and ¬D

Na rysunku 1 zilustrowane jest poszukiwanie rozwiązania problemu. W kolejnych etapach stosowane są możliwe operatory oraz dopisywane skutki ich działania. Linie przerywane symbolizują brak zastosowania operatora, pionowymi kreseczkami zaznaczono niektóre relacje wzajemnego wykluczania się. Symbole 'W1' oraz 'W2' oznaczają dwa możliwe światy wynikające z dwóch możliwych sytuacji początkowych.

[1] www.cs.washington.edu/research/projects/www/sgp.html



Rys.1. Ekspansja grafu dla zadania z przykładu 2
Fig.1. Graph expansion for example 2

Stan początkowy w przykładzie 2 jest alternatywą dwóch możliwych sytuacji. Celem problemu planowania jest, aby pacjent wyzdrowiał i żył (¬I and ¬D). Na rysunku 1 przedstawiony jest graf poszukiwania rozwiązania równolegle dla dwóch możliwych stanów początkowych. Poszukiwanie rozwiązania problemu (planu) składa się z dwóch faz: ekspansji grafu oraz ekstrakcji rozwiązania. Fazy te są wykonywane naprzemiennie. Dopóki obydwa warunki definiujące cel nie są wzajemnie wykluczające się w każdej możliwej sytuacji w poziomie 2, dopóty potencjalnie może istnieć rozwiązanie problemu i algorytm poszukuje go w fazie ekstrakcji rozwiązania.

Ponieważ dodatkowe informacje pozwalające zdeterminować, która z dwóch sytuacji występuje w rzeczywistości, są niedostępne, należy analizować skutki zastosowania operatora w każdej możliwej sytuacji. Podanie lekarstwa w sytuacji 2 (operator Med_1 w w2) prowadzi wprost do wyleczenia pacjenta, ale nie wiadomo, która z dwóch sytuacji jest prawdziwa (w1 czy w2). Dlatego należy zbadać skutki zastosowania operatora również w w1. Okazuje się, że wtedy podanie lekarstwa powoduje śmierć pacjenta. Tak więc plan polegający na podaniu lekarstwa nie rozwiązuje problemu z niepewną sytuacją początkową.

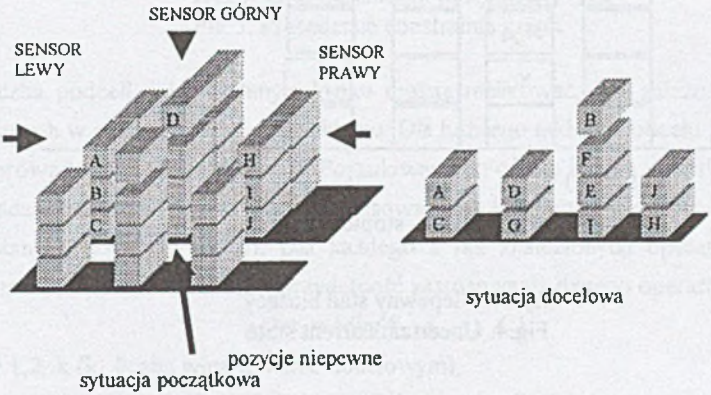
Planem rozwiązującym problem z przykładu 2 jest, oczywiście, zastosowanie 2 operatorów: najpierw *Drink*, potem *Medicate*. Stan docelowy jest wtedy osiągany niezależnie od możliwego stanu początkowego.

3. Niepewność w środowisku „świata klocków"

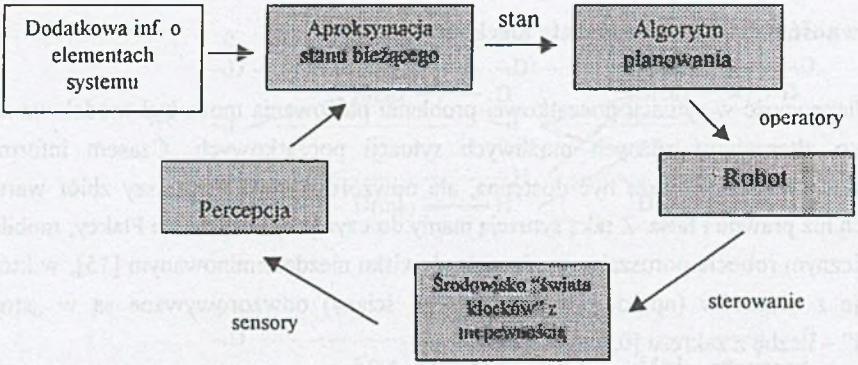
Niepewność w sytuacji początkowej problemu planowania może być modelowana nie tylko jako alternatywa różnych możliwych sytuacji początkowych. Czasem informacja pochodząca z sensorów może być dostępna, ale odwzorowywana w gęstszy zbiór wartości logicznych niż prawda i fałsz. Z taką sytuacją mamy do czynienia w systemie Flakey, mobilnym autonomicznym robocie poruszającym się w środowisku niezdeterminowanym [15], w którym informacje z sensorów (np. odległość robota od ściany) odwzorowywane są w „stopień pewności" – liczbę z zakresu [0,1].

Podobną sytuację można zamodelować w środowisku „świata klocków". W tym przypadku źródłem niepewności jest położenie niektórych elementów w systemie STRIPS. Źródło niepewności można by schematycznie przedstawić na rys.2. Sytuacja początkowa problemu to kilka szeregów wież klocków. Wtedy elementy wewnętrzne sytuacji początkowej zadania są nierozróżnialne dla sensorów zaznaczonych na rysunku. Sytuacja docelowa problemu jest znana, tzn. położenie wszystkich elementów w scenie docelowej jest określone. Jeśli dodatkowa informacja o położeniu tych elementów mogłaby być reprezentowana przez „stopień pewności", wtedy system taki ma cechy systemu z niepewnością.

Schemat blokowy systemu STRIPS z uwzględnieniem niepewnej informacji jest przedstawiony na rys.3. Schemat ten w swojej idei został oparty na architekturze systemu Flakey.

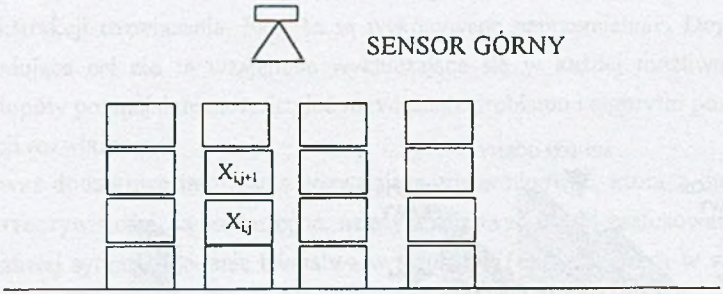


Rys.2. Sytuacje początkowe i docelowe w systemie z niepewnością
Fig.2. Initial and goal states in uncertain system



Rys.3. Architektura systemu STRIPS z uwzględnieniem niepewności
Fig.3. Uncertain STRIPS system architecture

Zadaniem dla bloku „Percepcja" jest rozumienie informacji dostarczonej przez sensory (np. kamery), tzn. generownie predykatów opisujących położenie znanych elementów (np. $IS(X_{i,j}) = 1$). Blok „Dodatkowa informacja o elementach systemu" dostarcza informacji o pozostałych elementach, których położenie nie jest określone w sposób pewny. Informacja ta jest odwzorowywana w liczbę z przedziału [0,1], która określa „stopień pewności" położenia elementu na danej pozycji (np. $IS(X_{i,j}) = A$, $A \in [0,1]$). Sytuacja ta jest przedstawiona na rysunku 4.



$IS(X_{i,top}) = 1$

$IS(X_{i,j}) = A$, $A \in [0,1]$ - stopień pewności

Rys.4. Niepewny stan bieżący
Fig.4. Uncertain current state

Blok „Aproksymacja stanu bieżącego" generuje predykaty dla systemu STRIPS na podstawie informacji dostarczonej przez poprzednie bloki systemu. Predykaty generowane są z wykorzystaniem reguł charakterystycznych dla systemów rozmytych [9]:

$$\text{IF } x_1 \text{ is } A_1 \text{ AND } x_2 \text{ is } A_2 \text{ AND} \ldots x_n \text{ is } A_n \text{ THEN } y \text{ is } B$$

gdzie: A_i, $i=1,2...n$ oraz B - stopień pewności wyrażony liczbą z przedziału [0,1],

x_i, i=1,2...n oraz y - dziedziny (predykaty).

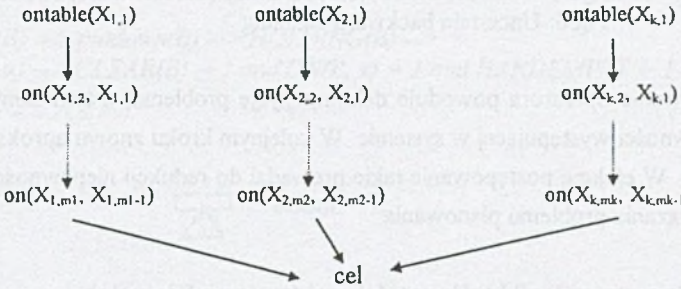
Korzystając dodatkowo z definicji koniunkcji w logice wielowartościowej:

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$$

można opisać reguły generujące opis niepewnego stanu bieżącego w sposób następujący:

1. IF $IS(X_{i,j}) = A$ AND $IS(X_{i,j+1}) = B$ THEN $ON(X_{i,j}, X_{i,j+1}) = \min(A,B)$ (1)
2. IF $IS(X_{i,0}) = A$ THEN $ONTABLE(X_{i,0}) = A$ (2)
3. IF $IS(X_{i,j}) = 1$ AND $IS(X_{i,j+1}) = 0$ THEN $CLEAR(X_{i,j}) = 1$ (3)

Zadaniem dla bloku „Algorytm planowania" jest generowanie operatorów STRIPS modelujących działania robota przekształcającego stan bieżący problemu. Zadanie to jest bardziej złożone. Dla danego stanu docelowego zadania należy wygenerować graf ograniczeń kolejnościowych (rys.5), który ogranicza wybór kolejnych podceli do realizacji. Wygenerowanie grafu takiej postaci pozwala uniknąć również sytuacji typu anomalia Susmanna [8].



Rys.5. Graf ograniczeń kolejnościowych
Fig.5. Precedence constraints graph

Liczba podceli, jaką w danym kroku można realizować, jest zależna od liczby wież zdefiniowanych w stanie docelowym problemu. Dla każdego takiego podcelu poszukiwany jest operator prowadzący do jego realizacji. Poszukiwanie to oparte jest na wnioskowaniu wstecz i przeprowadzane dopóty, dopóki warunki stosowalności jednego z operatorów są spełnione w definicji stanu bieżącego (rys. 6). Dla każdego z tak znalezionych operatorów obliczamy wartość funkcji wyboru określającej przydatność zastosowania danego operatora [15]:

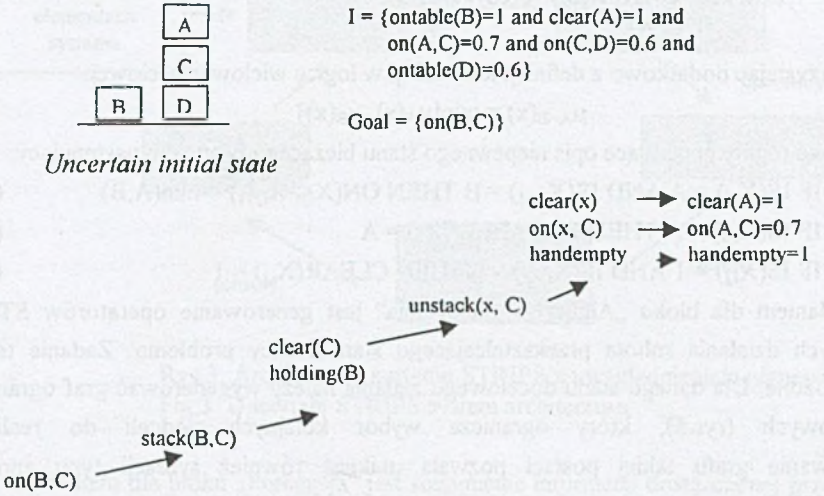
$$Des_i = Min(A_1, A_2, ..., A_n)$$

gdzie: i = 1,2..k (k- liczba wież w stanie docelowym),

A_j, j = 1,2..n - stopnie pewności warunków stosowalności danego operatora.

Na podstawie wartości funkcji przydatności operatora można wybrać ten operator do realizacji, dla którego wartość tej funkcji jest najwyższa:

$$Des = Max(Des_1, Des_2, ..., Des_k).$$



Rys.6. Wnioskowanie wstecz z uwzględnieniem niepewności
Fig.6. Uncertain backward chaining

Zastosowanie operatora powoduje dekompozycję problemu, a tym samym częściową redukcję niepewności występującej w systemie. W kolejnym kroku znowu aproksymowany jest stan bieżący itd. W efekcie postępowanie takie prowadzi do redukcji niepewności w kolejnych krokach i rozwiązania problemu planowania.

4. Zastosowanie systemu logiki trójwartościowej (systemu Kleene'a)

W systemie logiki trójwartościowej wartość logiczna stwierdzenia jest odwzorowywana w zbiór trójwartościowy {0, ½, 1}. Stwierdzenia T(a) = 0, T(a) = 1 oraz T(a) = ½ oznaczają odpowiednio, że „a" jest fałszywe, prawdziwe oraz że nic nie można powiedzieć o prawdziwości „a" [4].

Przypadek w środowisku „świata klocków", w którym nie dysponujemy dodatkową informacją o elementach ukrytych przed sensorami, może być reprezentowany poprzez system logiki Kleene'a. Wtedy 'stopień pewności" o położeniu elementów ukrytych odwzorowujemy wartością ½. Dla wszystkich bloków ukrytych generujemy predykaty:

$$IS(X_{ij}) = ½.$$

Pociąga to za sobą, że korzystając z reguł (1), (2), (3) otrzymujemy dla wszystkich bloków ukrytych:

$$ON(X_{i,j}, X_{i,j+1}) = ½$$

i

$$ONTABLE(X_{i,0}) = ½.$$

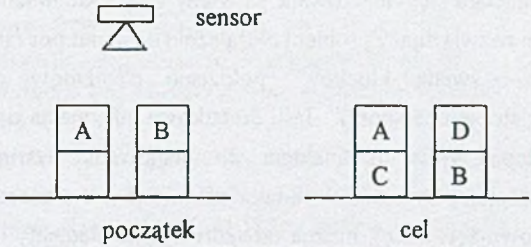
Działanie systemu korzystającego z logiki trójwartościowej jest zilustrowane na przykładzie 3. Informacje o stanie początkowym uzyskane z sensora nie pozwalają na pewne zdeterminowanie położenia elementów C i D. Wobec tego niektóre predykaty przyjmują wartości logiczne ½. Z grafu ograniczeń kolejnościowych wynika, że najpierw muszą być rozwiązane podcele ONTABLE(C) lub ONTABLE(B). Dla pierwszego podcelu wnioskowanie wstecz przybiera postać:

1. *ONTABLE(C) --> putdown(C) --> HOLDING(C) -->*
--> pickup(C) --> HANDEMPTY and ONTABLE(C) and CLEAR(C) -->
unstack(x, C) --> CLEAR(A) = 1 and ON(A, C) = ½ and HANDEMPTY = 1

podczas gdy dla kolejnego:

2. *ONTABLE(B) --> putdown(B) --> HOLDING(B) -->*
-> unstack(B, x) --> CLEAR(B) = 1 and ON(B, x) = 1 and HANDEMPTY = 1

Przykład 3



CLEAR(A) = 1	ONTABLE(C) and
CLEAR(B) = 1	ONTABLE(B) and
ON(A,C) = 1/2	ON(A,C) and ON(D,B)
ON(B,D) = 1/2	
ONTABLE(C) = 1/2	
ONTABLE(D) = 1/2	
HANDEMPTY = 1	

Poszukując wartości funkcji przydatności operatora obliczamy dla operatora *unstack(A, C)*:

$$Des_1 = Min(A_1, A_2, A_3) = Min(1, ½, 1) = ½$$

oraz dla operatora *unstack(B, D)*:

$$Des_2 = Min(A_1, A_2, A_3) = Min(1, 1, 1) = 1.$$

Wybór operatora do realizacji następuje po obliczeniu wartości funkcji *Des*:

$$Des = Max(Des_1, Des_2) = Max\ (½, 1) = 1.$$

Wartości „1" odpowiada operator *unstack(B, D)* . Po zastosowaniu operatora blok „Percepcja" może już dostarczyć informacji, jaki rzeczywiście element znajduje się pod klockiem „B". Informacja ta redukuje niepewność w omawianym przykładzie.

5. Wnioski

Niepewność w systemach planowania *sztucznej inteligencji* może być modelowana na kilka sposobów. Jeśli dodatkowe informacje są dostarczane przez sensory, do poszukiwania planu rozwiązującego problem planowania można stosować algorytmy planowania warunkowego. Operacje są wtedy wykonywane warunkowo, w zależności od dostarczonej informacji. Jeśli informacja dodatkowa jest niedostępna, zastosowanie znajdują algorytmy planowania dostosowującego się. Generowane są wtedy wszystkie możliwe stany początkowe i poszukiwany jest plan rozwiązujący problem niezależnie od stanu początkowego.

W środowisku „świata klocków" położenie elementów ukrytych może być modelowane poprzez „stopień pewności". Jeśli dodatkowa informacja o położeniu nieznanych elementów jest dostępna (jest to analogia do planowania warunkowego), jest ona odwzorowywana w „stopień pewności". Jeśli taka informacja jest niedostępna (jest to anologia do planowania dostosowującego się), można odwzorowywać elementy ukryte, wykorzystując przykładowo system logiki trójwartościowej.

Niniejsza praca była finansowana z grantu KBN.

LITERATURA

1. Barret A. i Weld D.S.: Partial-order planning: evaluating possible efficiency gains. Artificial Intelligence, 1994, 67:71-112.
2. Bylander T.: The computational complexity of propositional STRIPS planning. Artificial Intelligence, 1994, 69:165-204.
3. Cormen T.H., Leiserson Ch.E., Rivest R.L.: Introduction to Algorithms. The Massachusetts Institute of Technology, 1994.
4. Dougherty, E.R. i Giardina Ch.R.: Mathematical Methods for Artificial Intelligence and Autonomous Systems. Prentice-Hall International, Inc. USA, 1988.

5. Gałuszka A.: Block world - state space searching. Proc. IMACS World Congress'97, Berlin 1997, vol.6:571-576.
6. Gałuszka A.: „Distinguish- and indistinguishability of STRIPS planning problem elements". In Proceedings of the 11th European Simulation Symposium, Erlangen 1999, 95-99.
7. Garey M.R. i Johnson D.S.: Computers and Intractability - A Guide to the Theory of NP-Completeness. Freeman, San Francisco 1979, CA.
8. Joslin D. i Roach J.: A Theoretical Analysis of Conjuctive-Goal Problems. Artificial Intelligence 1989/90, 41:97-106.
9. Klir G.J., Folger T.A.: Fuzzy Sets, Uncertainty, and Information. Prentice-Hall International, Inc. 1988, USA.
10. McDermott D. i Hendler J.: Planning: what it is, what it could be, an introduction to the special issue on planning and scheduling. Artificial Intelligence 1995, 76:1-16.
11. Nebel B. i Koehler J.: Plan reuse versus plan generation: a theoretical and empirical results. Artificial Intelligence 1995, 76:427-454.
12. Nilson N. J.: Principles of Artificial Intelligence. Toga Publishing Company, Palo Alto, California 1980.
13. Popovic D. i Bhatkar V.P.: Methods And Tools For Applied Artificial Intelligence. Marcel Dekker, Inc., New York 1994, NY.
14. Rintanen, J.: Constructing Conditional Plans by a Theorem-Prover. Journal of Artificial Intelligence Research, 1999,10: 323-352.
15. Ruspini, E.H., A. Saffiotti, K. Konolige.: Progress in Research on Autonomous Vehicle Motion Planning. [in] Yen, J.; R. Langari; L.A. Zadeh. Industrial Applications of Fuzzy Logic and Intelligent Systems. IEEE Press. New York 1995.
16. Sierocki I.: A Serial Decomposition of Planning Problems. Proc. Fourth International Symposium on Methods and Models in Automation and Robotics. Międzyzdroje 1997, Poland, 1179-1184.
17. Smith D.E. i Weld D.S.: Conformant Graphplan. Proc. 15th National Conf. on AI, 1998..
18. Świerniak A. i Gałuszka A.: Betweenness and Indistinguishability in Modeling and Control of Uncertain Systems, Proc. of 18th IASTED Conference Modeling, Identification and Control, Innsbruck 1999, 56-58.
19. Weld D.S.: Recent Advantages in AI Planning. Technical Report UW-CSE-98-10-01; to appear in AI Magazine, 1999.
20. Weld D.S., Anderson C.R. i Smith D.E.: Extending Graphplan to Handle Uncertainty & Sensing Actions. Proc. 15th National Conf. on AI, 1998, 897-904.
21. Yen J., Langari R., Zadeh L.A.: Industrial Applications of Fuzzy Logic and Intelligent Systems. IEEE Press. New York 1995.

Recenzent: Prof. dr hab.inż. Zbigniew Banaszak

Abstract

STRIPS formalism is a convenient representation for artificial intelligence planning problems. There are many different algorithms of state space searching which use STRIPS representation for planning. Some of them search for solution through a space of world-states. In general to

generate an optimal plan for planning instances is NP- hard. Block world environment, which represents problems of robot action planning, is very often modelled as a STRIPS system.

Classical planners make the unrealistic assumption that information about world is comlete. More realistic case is making the assumption that uncertainty is present in the world. There exist different approaches to planning under uncertainty. In most cases uncertainty is limited to initial state but methods of handling actions with uncertain effects are also known.

In this paper planning and goal persuit problems in artificial intelligence are considered. Especially modelling of planning problem which contain incomplete and uncertain information is analysed. In the first part of the paper two classical approaches to planning under uncertainty are presented. One of these aproaches is explained by the example. Yet another approach to modelling system in which uncertainty is represented by „certain degree" is presented in the second part of the paper. The essence of this approach is explained by the example in block world environment. In all cases planning problems have been modelled using STRIPS representation.