Adam GAŁUSZKA, Eryka PROBIERZ
Politechnika Śląska

PLANOWANIE TYPU STRIPS Z NIEPEWNOŚCIĄ JAKO ZADANIE PROGRAMOWANIA LINIOWEGO MIESZANEGO

Streszczenie. Klasyczne planowanie w sztucznej inteligencji jest kosztownym obliczeniowo problemem znalezienia sekwencji działań, które przekształcają dany stan początkowy problemu do pożądanej sytuacji docelowej. Sekwencja działań nazywana jest planem. Brak informacji o stanie początkowym prowadzi do poszukiwania planu warunkowego lub odpornego na tę niepewność, co jest zadaniem trudniejszym od klasycznego. Plan równoległy to plan, w którym niektóre czynności mogą być wykonywane równolegle, co zwykle prowadzi do zmniejszenia czasu wykonania planu, ale zwiększenia trudności jego znalezienia. W pracy sformułowano powyższe problemy jako zadanie programowania liniowego mieszanego (MILP).

STRIPS PLANNING WITH UNCERTAINTY AS A MIXED LINEAR PROGRAMMING PROBLEM

Summary. Classical planning in artificial intelligence is the computationally expensive problem of finding a sequence of actions that transform a given initial state of a problem to a desired target situation. The sequence of actions is called a plan. Lack of information about the initial state leads to the search for a conditional plan or a plan immune to this uncertainty, which is a more difficult task than the classical one. A parallel plan is a plan in which some activities can be performed in parallel, which usually leads to a decrease in the execution time of the plan, but an increase in the difficulty of finding the plan. The paper formulates the above problems as a mixed linear programming task (MILP).

1. Wprowadzenie

Planowanie jako problem sztucznej inteligencji jest formułowane jako proces poszukiwania prowadzący do sekwencji działań agenta (zwanych planem), które przekształcają początkowe środowisko agenta (zwane stanem początkowym problemu planistycznego) do pożądanej sytuacji docelowej (np. [6, 8, 10, 11, 12]).

Problem komplikuje się, jeśli informacja o modelowanym świecie nie jest wystarczająca do określenia wszystkich faktów niezbędnych do opisania stanu początkowego świata. Mówimy wtedy, że stan początkowy problemu jest niepewny, ale może być reprezentowany przez zbiór możliwych stanów początkowych. Plan rozwiązania takiego problemu może mieć postać działań, które są wykonywane warunkowo, na podstawie nowych informacji pojawiających się w trakcie poszukiwania planu. Takie podejście nazywane jest planowaniem warunkowym [11, 14].

W niektórych przypadkach informacje z sensorów mogą być niedostępne np. sensory są uszkodzone lub zepsute, odbieranie informacji sensorycznych jest zbyt kosztowne lub niebezpieczne. Wówczas zasadne jest poszukiwanie planu, który jest rozwiązaniem problemu planowania niezależnie od możliwych stanów początkowych. Takie podejście nazywane jest planowaniem odpornym (ang. *conformant planning*) [11, 14]. Zarówno planowanie warunkowe, jak i odporne są trudniejsze do rozwiązania niż planowanie klasyczne [2]. W pracy dokonano wielomianowej transformacji problemu planowania do problemu MILP w oparciu o [5].

2. Zadanie planowania

Za Bylanderem [3] przyjmujemy, że problem planowania składa się z czterech zbiorów {C, O, I, G}:

– C jest skończonym zbiorem warunków,
– O jest skończonym zbiorem działań, gdzie każde działanie oO przyjmuje postać c^+, $c^- \to c_+, c_-$, gdzie: c^+ to tzw. pozytywne przesłąnkami; c^- są tzw. negatywnymi przesłankami, c_+ są tzw. pozytywnymi wnioskami, c_- są tzw. negatywnymi wnioskami,
– I jest stanem początkowym,
– G = {G+, G-} jest sytuacją docelową, gdzie G+ C są warunkami pozytywnymi (tzn. są prawdziwe), a G- C są warunkami negatywnymi (tzn. są fałszywe).

Aby zawrzeć w opisie aktualnego stanu problemu informację, że niektóre warunki są nieznane (załóżmy, że k warunków może być prawdziwych lub fałszywych), można wprowadzić tzw. k-stany zaproponowane przez Barala et al. [1]. W uproszczeniu k-stan to para (s, Σ), gdzie s jest aktualnym stanem problemu, a Σ jest zbiorem, który składa się ze wszystkich możliwych stanów początkowych I. Dla nieznanego stanu początkowego zbiór składa się ze wszystkich stanów s, dla których:

– warunek c jest prawdziwy w stanie początkowym,
– warunek c jest fałszywy,
– jeśli nie wiadomo, czy warunek c jest prawdziwy czy fałszywy w stanie początkowym, to zbiór zawiera oba stany, dla których warunek ten jest prawdziwy i fałszywy.

Stanem początkowym I może być potencjalnie dowolny stan ze stanów zawartych w zbiorze. Liczba możliwych stanów początkowych jest oznaczana przez w i jest ograniczona przez k takie, że: w ≤ 2k. Taki problem planowania z niepełną informacją o stanie początkowym nazywany jest konforemnym problemem planowania i przyjmuje postać czwórki (C, O, Σ, G).

Współczesne systemy inteligentne są często wyposażone w różnego rodzaju czujniki, które służą do określania różnych właściwości otoczenia robota. Informacje te mogą być odwzorowane na stopień prawdy warunków, które definiują aktualny stan problemu. Zazwyczaj odbywa się to poprzez wprowadzenie specjalnych akcji zwanych akcjami sensorycznymi [7, 14]. Ponieważ nie ma formalnego rozszerzenia planowania STRIPS o akcje sensoryczne [1], poniżej zaproponowano definicję tych akcji dla k nieznanych warunków, jako specjalnego podzbioru akcji STRIPS.

Definicja. Dla k nieznanych warunków zbiór działań sensorycznych O_s jest skończonym zbiorem działań, gdzie dla każdego działania sensorycznego $o_s \in O_s$ należy wprowadzić dwa działania sensoryczne STRIPS $\{ o_s^t, o_s^f \} \in O_s$, które przyjmują postać:

$o_s^t : c^+, c^- \rightarrow c_i$, jeśli warunek ci jest prawdziwy po wykonaniu czynności os,
$o_s^f : c^+, c^- \rightarrow \neg c_i$, jeśli warunek ci jest fałszywy po wykonaniu działania os,
i = 1,2...k.

Wynika z tego, że maksymalna liczba działań sensorycznych | Os | = 2k. Wynik zastosowania akcji do stanu bieżącego zależy od tego, czy akcja jest zwykła, czy sensoryczna. Taki problem planowania z niepełną informacją o stanie początkowym i akcjach sensorycznych nazywany jest problemem planowania warunkowego i przyjmuje postać piątki (C, O, O_s, Σ, G).

3. Transformacja do MILP

Podążając za [4], transformacja z planowania do programowania liniowego opiera się na mapowaniu warunków i operatorów w każdym kroku planu na zmienne. Wartości prawdy warunków są mapowane na "0" i "1" dla planowania bez niekompletności oraz na dowolne wartości pomiędzy "0" i "1" dla planowania z niekompletną informacją.

Przyjmijmy, że jeśli c jest warunkiem i jeśli proces planowania jest podzielony na l kroków, a i jest indeksem kroku ($i = 0, 1, ..., l$), to dla tego warunku potrzeba $(l + 1)$ zmiennych: c(0), c(1), ..., c(l) niecałkowitoliczbowych. Jeśli z kolei o jest działaniem, to dla każdego działania potrzebujemy l zmiennych całkowitoliczbowych (podjęcie działania lub jego niepodjęcie): o(0), o(1), ..., o(l-1). Jeśli $o_s \in O_s$ jest akcją sensoryczną, to dla każdej akcji sensorycznej potrzebujemy 2l zmiennych całkowitoliczbowych (użycie sensora lub jego nieużycie): : $o_s^t(0), o_s^t(1), ..., o_s^t(l\text{-}1)$, and $o_s^f(0), o_s^f(1), ..., o_s^f(l\text{-}1)$. W ten sposób argumenty warunków c i operatorów o są rozszerzone o indeks kroku planowania.

Funkcja celu osiąga swoją maksymalną wartość, jeśli wartość zmiennej $c(l)$ odpowiadającej warunkowi wynosi: $c(l) = 1$ jeśli $c \in G_+$ i $c(l) = 0$ jeśli $c \in G_-$, czyli stan docelowy jest prawdziwy w ostatnim kroku planowania l.

Na podstawie powyższego można zbudować LPP mając wektor zmiennych decyzyjnych x,:

$$x = \{ c(0), o(0), o_s^t(0), o_s^f(0), c(1), o(1), o_s^t(1), o_s^f(1),... \\ .... c(l\text{-}1), o(l\text{-}1), o_s^t(l\text{-}1), o_s^f(l\text{-}1), c(l) \}.$$

Przyjmijmy teraz, że zbiór $G_+ = \{c^{pos}{}_1, c^{pos}{}_2, \dots c^{pos}{}_n\}$ oraz zbiór $G_- = \{c^{neg}{}_1, c^{neg}{}_2, \dots c^{neg}{}_m\}$, czyli cel składa się z n warunków pozytywnych i m negatywnych, więc istnieje $(n + m)$ zmiennych stanowiących funkcję celu LP:

$$c^{pos}(l) = [c^{pos}{}_1(l), c^{pos}{}_2(l), \dots c^{pos}{}_n(l)];$$
$$c^{neg}(l) = [c^{neg}{}_1(l), c^{neg}{}_2(l), \dots c^{neg}{}_m(l)].$$

Funkcja celu do maksymalizacji to:

$$\text{Max} \leftarrow f\,(c^{pos}(l), c^{neg}(l)) = [\ \textstyle\sum_{i=1}^{n}\ (c^{pos}{}_i\,(l)) + \sum_{j=1}^{m}\ (1 - c^{neg}{}_j\,(l))].$$

Ponieważ każdy składnik funkcji celu powinien być równy 1 (jeden), jeśli cel zostanie osiągnięty, optymalna wartość funkcji celu (f_{opt}) do maksymalizacji jest znana przed rozwiązaniem problemu LP i wynosi $f_{opt} = (n + m)$. Problem optymalizacyjny sformułowany jest jako:

Znajdź minimalną liczbę kroków „l", dla których $f_{opt} = (n + m)$.

Ograniczenia nierównościowe (w postaci "większe lub równe") wyrażają własność, że działania mogą być stosowane, jeśli ich warunki wstępne są prawdziwe. Lewa strona nierówności składa się ze zmiennej odpowiadającej warunkowi wstępnemu. Jeśli warunek wstępny nie jest spełniony, to wartość zmiennej dla tego warunku wynosi "0". Prawa strona nierówności jest sumą zmiennych odpowiadających wszystkim działaniom mającym warunek wstępny. Tak więc, jeśli lewa strona jest '0' żadna akcja mająca warunek wstępny nie może być zastosowana. W zależności od reprezentacji stanu początkowego i liczby akcji wykonywanych równolegle nierówności różnią się w szczegółach.

Ograniczenia równości opisują zmiany wartości zmiennych dla warunków w wyniku zastosowania akcji. W zależności od reprezentacji stanu początkowego i liczby równolegle wykonywanych akcji równości te również różnią się w szczegółach.

Do modelowania niepewności co do prawdziwości lub fałszywości nieznanego warunku c proponuje się wykorzystanie trójwartościowego systemu logicznego Kleena. W systemie tym wartości logiczne warunku są odwzorowane na zbiór {0, ½, 1}. Zdania T(a) = 0, T(a) = 1 oraz T(a) = ½ oznaczają, że warunek "a" jest fałszywy, prawdziwy lub że nic nie można powiedzieć o prawdziwości "a" (Dougherty i Giardina 1988). Poniżej wprowadzamy zbiór ograniczeń dla zadania planowania warunkowego.

4. Ograniczenia dla zadania planowania warunkowego

Rozważmy przypadek zadania planowania warunkowego z dwoma akcjami klasycznymi i jedną akcją sensoryczną sprawdzającą niepewny warunek c_3:

o_1: $c_1 \rightarrow c_2$,
o_2: $\neg c_1, \neg c_2 \rightarrow \neg c_3$,
$o_{s1}{}^t$: $\{\} \rightarrow c_3$,
$o_{s1}{}^f$: $\{\} \rightarrow \neg c_3$,

wtedy zbiór ograniczeń nierównościowych przyjmuje postać:

$c_1(i) \geq o_1(i)$,
$1 - c_1(i) \geq o_2(i)$,
$1 - c_2(i) \geq o_2(i)$,

natomiast zbiór ograniczeń równościowych przyjmuje postać:

$c_2(i+1) = c_2(i) + o_1(i)$,
$c_3(i+1) = c_3(i) - o_2(i) + ½\, o_{s1}{}^t(i) - ½\, o_{s1}{}^f(i)$.

Należy zauważyć, że wartość c3 w następnym kroku planowania, $c_3(i+1)$, może być modyfikowana na dwa sposoby:

1) jeśli c3 jest prawdziwe w bieżącym stanie problemu planistycznego (tj. $c_3(i) = 1$), to w następnym kroku po zastosowaniu akcji o2 może pozostać fałszywe;
2) jeśli c3 jest nieznane w bieżącym stanie problemu planowania (np. $c_3(i) = ½$), to może pozostać prawdziwe lub fałszywe w następnym kroku w zależności od wyniku działania sensorycznego.

5. Podsumowanie i wnioski

W pracy sformułowano zadanie planowania z niepewnym stanem początkowym problemu jako zadania programowania liniowego mieszanego. Wyjściowym modelem problemu był system STRIPS rozszerzony o zbiór możliwych stanów początkowych problemu oraz zbiór akcji sensorycznych, umożliwiających redukcję niepewności części stanu początkowego zadania. Model ten został przekształcony do zadania programowania liniowego mieszanego poprzez wprowadzenie zbioru zmiennych całkowitoliczbowych odzwierciedlających podejmowane decyzje oraz zbioru zmiennych ciągłych odzwierciedlających stopień spełnienia warunków opisujących stan problemu.

Podziękowania. Praca na podstawie artykułu (Gałuszka i Probierz 2021)[5]. Praca AG została wsparta w części przez Politechnikę Śląską (SUT) w ramach grantu 02/060/BK_22/0030: dotacja na utrzymanie i rozwój potencjału badawczego w 2022 roku. Praca EP została wsparta w części przez Unię Europejską w ramach Europejskiego Funduszu Społecznego jako stypendium w ramach Grantu POWR.03.02.00-00-I029, a w części przez Politechnikę Śląską (SUT) w ramach grantu: dotacja na utrzymanie i rozwój potencjału badawczego w 2022 roku dla młodych naukowców.

LITERATURA

1. Baral Ch., Kreinovich V., Trejo R. (2000). Computational complexity of planning and approximate planning in the presence of incompleteness. Artificial Intelligence, 122, pp. 241-267.
2. Bonet B. (2010). Conformant plans and beyond: Principles and complexity. Artificial Intelligence 174 (2010) 245–269.
3. Bylander, T. (1994). The Computational Complexity of Propositional STRIPS Planning. Artificial Intelligence, 69,165-204.
4. Bylander T. (1997). A Linear Programming Heuristic for Optimal Planning. In AAAI97/IAAI-97 Proceedings, 1997, 694–699.
5. Gałuszka A., Probierz E. (2021). On transformation of conditional, conformant andparallel planning to linear programmin. Archives of Control Sciences 31, 2, pp. 375–399.
6. Ghooshchi N.G., Namazi M., Newton M.A., Sattar A. (2015). Transition Constraints for Parallel Planning. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 3268-3274.
7. Oglietti M. (2005). Understanding planning with incomplete information and sensing. Artificial Intelligence 164, pp. 171–208.
8. Palacios H., Geffner H. (2009). Compiling Uncertainty Away in Conformant Planning Problems with Bounded Width. Journal of Artificial Intelligence Research 35, pp. 623-675.
9. Rintanen J. (1999). Constructing Conditional Plans by a Theorem-Prover. Journal of Artificial Intelligence Research, 10, 323-352.
10. Rosa T., Jimenez S., Fuentetaja R., Barrajo D. (2011). Scaling up Heuristic Planning with Relational Decision Trees. Journal of Artificial Intelligence Research , 40, pp. 767-813.
11. Russell S.J., Norvig P. (2009). Artificial Intelligence, A Modern Approach. 3rd Edition. Prentice Hall Series in Artificial Intelligence, Berkeley.
12. Skrzypczyk K. (2010). Time optimal tracking a moving target by a mobile vehicle - game theoretical approach, Przegląd Elektrotechniczny (Electrical Review), R. 86 NR 3/2010 s. 211-215
13. Son T.C., Phan H.T., Gelfond M., Morales A.R. (2005). Conformant planning for domains with constraints: a new approach. In Proceedings of the 20th national conference on Artificial intelligence - Volume 3 (AAAI'05), Anthony Cohn (Ed.), Vol. 3. AAAI Press pp. 1211-1216.
14. Weld D.S., Anderson C.R., Smith D.E. (1998). Extending Graphplan to Handle Uncertainty & Sensing Actions. Proc. 15th National Conf. on AI, pp. 897-904.