ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2006
Seria: AUTOMATYKA z. 145 Nr kol. 1728

Adam GAŁUSZKA
Politechnika Śląska

O MOŻLIWOŚCI TRANSFORMACJI PLANOWANIA Z REPREZENTACJĄ STRIPS

Streszczenie. Planowanie z reprezentacją STRIPS jest problemem złożonym obliczeniowo. Jednym ze sposobów poprawy efektywności poszukiwania rozwiązania może być przekształcenie problemu do innej postaci i poszukiwanie rozwiązania problemu przekształconego. W pracy sprowadzono problem planowania posiadający reprezentację STRIPS do dwóch postaci: programu liniowego oraz układu równań i nierówności. Następnie porównano efektywność uzyskania rozwiązania problemów przekształconych oraz możliwość uzyskania rozwiązania, na bazie rozwiązań problemów przekształconych, problemu pierwotnego.

ON POSSIBILITY OF TRANSFORMATION OF STRIPS PLANNING

Summary. STRIPS planning is a difficult computational problem. One way to increase efficiency of searching for a solution is to transform a problem to another problem and then search for a solution of the transformed problem. In this work a simple STRIPS problem has been transformed to two problems: linear programming and linear equalities and inequalities. Then the efficiency and quality of this approach has been analyzed and compared.

1. Wprowadzenie

Jądrem implementacji robota (agenta) inteligentnego jest system, w którym zapisana wiedza oraz określona metoda, poszukiwania rozwiązań problemów pozwala na generowanie planu ruchów robota (działań agenta). Przykładem takiego systemu jest STRIPS (Yen i inni 1995, Popovic i Bhatkar 1994). Planowanie z reprezentacją STRIPS jest problemem złożonym obliczeniowo (Bylander 1994). Obecnie różne sposoby poprawy efektywności planowania z wykorzystaniem reprezentacji STRIPS są często poruszane w literaturze (np.: Slaney i Thiebaux 2001, Howe i Dahlman 2002, Weld 1999, Weld i inni 1998, Sierocki 1997, Vossen et. Al. 2000, Nebel i Koehler 1995). W niniejszej pracy rozważane są problemy planowania zadań i osiągania celów w środowisku tzw. świata klocków. Tego typu problemy mogą być reprezentowane za pomocą systemu STRIPS (Nilson 1980).

Jednym ze sposobów poprawy efektywności poszukiwania rozwiązania może być przekształcenie problemu do innej postaci i poszukiwanie rozwiązania problemu przekształconego.

W pracy sprowadzono problem planowania w środowisku tzw. świata klocków, posiadający reprezentację STRIPS, do dwóch postaci: programu liniowego według heurystyki Bylandera (1997) oraz, na tej podstawie, układu równań i nierówności. Następnie porównano efektywność uzyskania rozwiązania problemów przekształconych oraz możliwość uzyskania rozwiązania, na bazie rozwiązań problemów przekształconych, problemu pierwotnego. Do porównania efektywności wykorzystano algorytmy MATLAB-a: programowania liniowego i całkowitoliczbowego binarnego dla programu liniowego, oraz programowania kwadratowego dla rozwiązania układu równań i nierówności.

Organizacja pracy jest następująca: w sekcji 2 zdefiniowano system STRIPS, w sekcji 3 pokazano transformację do programu liniowego oraz do układu równań i nierówności, w sekcji 4 podano wyniki symulacji. Całość zakończono podsumowaniem i wnioskami.

2. Problem planowania STRIPS

Problem planowania STRIPS (*STRIPS planning problem*) składa się z czterech zbiorów $\{P, O, I, G\}$ (Nilson 1980, Bylander 1984), 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,
  - $o^- \subseteq P$ są negatywnymi warunkami stosowalności operatora,
  - $o_+ \subseteq P$ są pozytywnymi skutkami zastosowania operatora,
  - $o_- \subseteq P$ są negatywnymi skutkami zastosowania operatora,
- $I \subseteq P$ jest stanem początkowym zadania (*initial state*),
- $G = \{G_+, G_-\}$ 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 $G_- \cap S = \varnothing$.

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

$Result(S, \{\}) = S$,

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

S, w przeciwnym przypadku,

$Result(S, \{o_1, o_2, \ldots, o_n\}) = Result(Result(S, \{o_1\}), \{o_2 \ldots, o_n\})$.

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. Dany operator może występować wiele razy w sekwencji operatorów tworzących plan. Skończona sekwencja operatorów $\{o_1, o_2, \ldots, o_n\}$ jest rozwiązaniem problemu planowania, jeśli stan wynikający z zastosowania funkcji $Result(I, \{o_1, o_2, \ldots, o_n\})$ jest stanem docelowym problemu.

3. Transformacja do programu liniowego oraz układu równań i nierówności

Idea transformacji do zadania programowania liniowego polega na przypisaniu każdemu warunkowi i operatorowi systemu STRIPS w każdym etapie planowania zmiennej (Bylander 1997). Zakładamy, że jeśli p jest warunkiem i jeśli planowanie jest podzielone na l etapów, to potrzebujemy $l+1$ zmiennych dla tego warunku: $p(0), p(1), \ldots, p(l)$. Jeśli z kolei op będzie operatorem, to potrzebujemy l zmiennych dla każdego operatora: $op(0), op(1), \ldots, op(l-1)$. Funkcja celu osiąga wartość maksymalną, jeśli wartości warunków definiujących cel w ostatnim etapie planowania są równe 1, przy założeniu że wartości zmiennych zawierają się w przedziale <0,1> odpowiadającym wartościom prawdy dla warunków i operatorów. Dla przykładowych transformacji świata klocków patrz (Gałuszka i Świerniak 2004).

Otrzymany w wyniku powyższej transformacji problem liniowy ma rozmiar (dla „n " klocków i „l " kroków planowania, gdzie zawsze $l \leq n$ dla problemu dekompozycji):

- liczba zmiennych: mniej niż $2l(n^2-1)$;
- liczba ograniczeń nierównościowych: $l\,(n^2 + n + 1)$;
- liczba ograniczeń równościowych: $l\,(n^2 + n)$.

Przy założeniu że problem liniowy ma postać:

$$\max_x f^T x \qquad Ax \leq b$$
$$A_{eq}x = b_{eq}$$
$$0 \leq x \leq 1$$

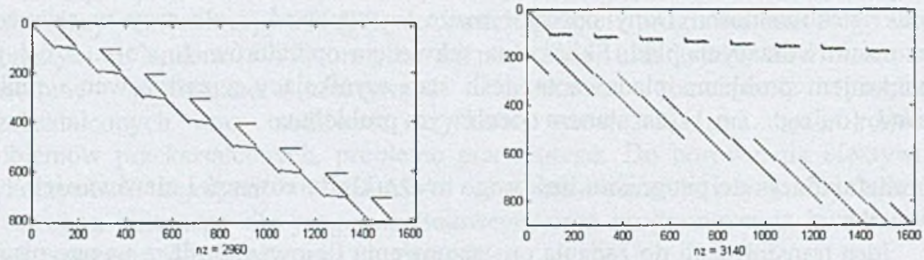
wtedy wektor f definiuje pożądany stan docelowy problemu planowania, natomiast stan początkowy zawarty jest w wektorze b_{eq}. Macierze A oraz A_{eq} są rzadkie, a ich postać dla 10 klocków oraz 8 kroków planowania jest przedstawiona na rysunku 1.

Jeśli wartości zmiennych odpowiadających za stan docelowy problemu planowania potraktowane zastaną jako ograniczenia równościowe, wtedy problem planowania można przekształcić do układu równań i nierówności:

$$Cx = d$$
$$Ax \leq b$$
$$0 \leq x \leq 1$$

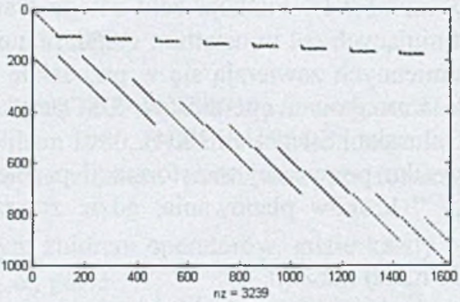
gdzie macierz C jest macierzą A_{eq} poszerzoną o zmienne dla ograniczeń równościowych definiujących stan docelowy, natomiast wektor d jest wektorem b_{eq} poszerzonym o wartości zmiennych dla stanu docelowego. Macierze A oraz wektor b

pozostają bez zmian w stosunku do przykładu z rysunku 1. Na rysunku 2 pokazano przykład macierzy C dla problemu jak z rysunku 1.



A A eq

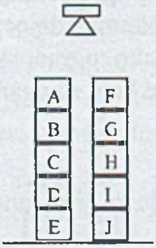
Rys. 1. Macierze dla programu liniowego



Rys. 2. Macierz C dla układu równań i nierówności

4. Wyniki symulacji

Dla analizy efektywności transformacji świata klocków do programu liniowego oraz układu równań i nierówności zaimplementowano problemy przekształcone, odpowiadające problemowi dekompozycji 10 klocków o konfiguracji jak na rysunku 3 (do dekompozycji wystarczy 8 kroków).



Rys. 3. Sytuacja początkowa

Problemy przekształcone posiadają w tym przypadku 1620 zmiennych. Zadanie programowania liniowego zastało rozwiązane za pomocą algorytmu dla dużych skal (metoda I) oraz binarnego programowania całkowitoliczbowego (metoda IV), natomiast układ równań i nierówności został rozwiązany za pomocą algorytmu programowania kwadratowego, gdzie funkcja celu przyjęła postać: $min(Cx\text{-}d)^2$ (metoda III) oraz postać: $min(x^*\text{-}1)^2$ (metoda IV), przy czym x^* jest 1 dowolną zmienną prawdziwą w problemie planowania (np. zmienną, która mówi, że w sytuacji docelowej, po 8 kroku, na klocku J nie ma być innego klocka: x^*: clear(J,8)=1). Wyniki symulacji przedstawiono w tabeli 1, w której dla ilustracji przedstawiono wartości zmiennych dla warunków w ostatnim kroku planowania (warunek np. clear(A)=1 mówi, że na klocku A nie ma innego klocka).

Tabela 1

Wyniki symulacji dla 4 różnych metod:
I – programowanie liniowe
II – programowanie kwadratowe
III – programowanie kwadratowe – minimalizacja 1 zmiennej
IV – programowanie całkowitoliczbowe binarne

8	clear	A	1	0,9389	1	1
8	clear	B	1	0,9389	1	1
8	clear	C	1	0,9389	1	1
8	clear	D	1	0,9389	1	1
8	clear	E	1	0,9389	1	1
8	clear	F	1	0,9389	1	1
8	clear	G	1	0,9389	1	1
8	clear	H	1	0,9389	1	1
8	clear	I	1	0,9389	1	1
8	clear	J	1	0,9389	1	1
		method:	**I**	**II**	**III**	**IV**
		time[s]:	2	499	974	594

5. Podsumowanie i wnioski

Rozwiązanie problemu przekształconego do zadania programowania liniowego jest łatwiej znajdowane przez algorytm klasyczny (metoda I), jednak rozwiązanie to może nie być łatwo interpretowalne (tzn. zmienne mogą przyjmować dowolne wartości z przedziału <0,1>, co nie ma bezpośredniego przełożenia na spełnienie bądź niespełnienie warunku). Programowanie całkowitoliczbowe (metoda IV) da rozwiązanie dokładne (tzn. plan dla problemu pierwotnego), jednak jest nieefektywne dla problemów dużych rozmiarów, co wynika z wykładniczego wzrostu czasu obliczeń wraz ze wzrostem rozmiaru problemu. Z kolei, reprezentacja planowania poprzez układ równań i nierówności jest ideą atrakcyjną, gdyż w wyniku transformacji nie otrzymujemy problemu optymalizacyjnego. Jednak dostępne metody znajdowania rozwiązań takich układów bazują na metodach optymalizacji (tutaj wykorzystano programowanie kwadratowe), które dla dużych rozmiarów nie okazują się efektywne (metody II – rozwiązanie w ogóle nie zostało znalezione i III).

Podziękowania. Praca była finansowana z funduszu BK w roku 2006.

LITERATURA

1. Bylander T.: The computational complexity of propositional STRIPS planning. Artificial Intelligence, 69/1994, p. 165-204.
2. Bylander T.: A Linear Programming Heuristic for Optimal Planning. Int. Conf. American Association for Artificial Intelligence, 1997, www.aaai.org.
3. Gałuszka A., Świerniak A.: 2004. Translation STRIPS Planning in Multi-Robot Environment to Linear Programming. LNCS, Springer-Verlag, Volume 3070/2004, p.768-773.
4. Howe A.E., Dahlman E.: A Critical Assesment of Benchmark Comparison in Planning. Journal of Artificial Intelligence Research, 17/2002, p. 1-33.
5. Nebel B., Koehler J.: Plan reuse versus plan generation: a theoretical and empirical results. Artificial Intelligence, 76/1995, p. 427-454.
6. Nilson N. J.: Principles of Artificial Intelligence. Toga Publishing Company, Palo Alto, California 1980.
7. Popovic D., Bhatkar V.P.: Methods And Tools For Applied Artificial Intelligence. Marcel Dekker, Inc., New York 1994.
8. Rintanen J.: Constructing Conditional Plans by a Theorem-Prover. Journal of Artificial Intelligence Research, 10/1999, p. 323-352.
9. Sierocki I.: A Serial Decomposition of Planning Problems. Proc. Fourth International Symposium on Methods and Models in Automation and Robotics. Międzyzdroje, Poland 1997, p. 1179-1184.
10. Slaney J., Thiebaux S.: Block World revisited. Artificial Intelligence, 125/2001, p. 119-153.
11. Vossen T., Ball M., Lotem A., Nau D.: Applying Integer Programming to AI Planning. Knowledge Engineering Review, 2000.
12. Weld D.S.: Recent Advantages in AI Planning. AI Magazine, 1999.
13. 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. Tadeusz Puchałka

Abstract

STRIPS planning is a difficult computational problem. One way to increase efficiency of searching for a solution is to transform a problem to another problem and then search for a solution of the transformed problem. In this work a simple STRIPS problem has been transformed to two problems: linear programming and linear equalities and inequalities. The transformation from planning to Linear Programming is based on mapping of conditions and operators in each plan step to variables. Truth-values of conditions are mapped to 0 and 1 values. For linear programming the objective function reaches the maximum if the goal situation is true in last step of planning. For inequalities and equalities the goal situation is mapped to equality constraints. The efficiency and quality of this approach has been analyzed and compared.