ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2006
Seria: AUTOMATYKA z. 143 Nr kol. 1726

Wojciech THOMAS
Politechnika Wrocławska

PORÓWNANIE ALGORYTMÓW ROZWIĄZANIA PROBLEMU SZEREGOWANIA ZADAŃ Z RUCHOMYMI REALIZATORAMI DLA KRYTERIUM MFT

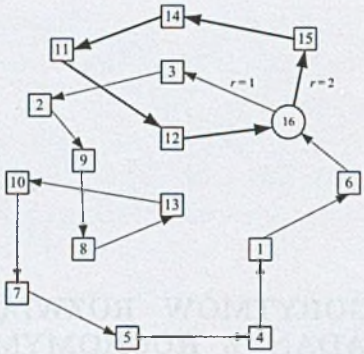
Streszczenie. W pracy przedstawiono porównanie dotychczas opracowanych algorytmów dotyczących problemu szeregowania zadań z ruchomymi realizatorami i kryterium w postaci średniego czasu przepływu. Algorytmy porównano pod względem osiąganej wartości kryterium oraz pod względem czasu działania. W pracy porównano algorytm symulowanego wyżarzania, algorytm ewolucyjny, dwa algorytmy hybrydowe oraz algorytm deterministyczny.

COMPARISON OF SOLUTION ALGORITHMS FOR TASK SCHEDULING PROBLEM WITH MOVING EXECUTORS AND MEAN FLOW TIME CRITERION

Summary. In the paper a comparison of heuristic algorithms for solving a task scheduling problem with moving executors and mean flow criterion is presented. The following algorithms are considered in this paper: evolutionary algorithm, simulated annealing algorithm, two hybrid algorithms and a deterministic algorithm. All algorithms are compared with respect to a value of performance index and execution time.

1. Wstęp

Prezentowana praca dotyczy problemu szeregowania zadań z ruchomymi realizatorami. Z zagadnieniami tego typu można spotkać się w elastycznych systemach produkcyjnych, w systemach usługowych, gdy zadania albo czynności są realizowane na stanowiskach rozmieszczonych na płaszczyźnie lub w przestrzeni, przy czym nie można zaniedbać czasów przejazdu podmiotów wykonujących te czynności (realizatorów, robotów, pojazdów autonomicznych). Założono, że na każdym ze stanowisk może być wykonywana jedna czynność, dlatego stanowiska można utożsamić z czynnościami. Specyfika problemu polega na tym, że jako czas wykonania zadania h przez realizator r przyjęto sumę czasów dojazdu z poprzed-



Rys. 1. Przykładowa instancja problemu dla $R = 2$ i $H = 15$

niego stanowiska g do stanowiska h oraz czas wykonania czynności na stanowisku h:

$$\tau_{r,h} = \hat{\tau}_{r,g,h} + \bar{\tau}_{r,h}.$$

Przyjęte założenie o czasie trwania zadania powoduje, że czas jego wykonania nie będzie znany, dopóki nie zostaną określone trasy przejazdu realizatorów.

Na rys. 1 przedstawiono przykładową instancję problemu dla $H = 15$ zadań i $R = 2$ realizatorów. Numerem 16 oznaczono bazę, czyli specjalne stanowisko, na którym rozpoczynają pracę wszystkie realizatory i do którego powracają po wykonaniu przypisanych im zadań.

W badanym problemie jako kryterium jakości przyjęto średni czas przepływu (ang. *mean flow time*) zdefiniowany jako suma momentów zakończeń wykonywania zadań:

$$\bar{F} = \sum_{1}^{H} C_h.$$

2. Algorytm ewolucyjny EA

Do zakodowania rozwiązania badanego problemu zaproponowano wykorzystanie dwuwierszowej macierzy g, w której elementy pierwszego wiersza $g_{1,h}$ wskazywały na którym realizatorze jest wykonywane zadanie o numerze $g_{2,h}$. Kolejność wykonania zadań przez realizatory odpowiada kolejności kolumn w macierzy g. Przykładowa macierz g, opisująca trasy przejazdu dwóch realizatorów z rys.1, wygląda następująco:

$$g = \begin{bmatrix} 2 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 1 \\ 15 & 3 & 14 & 2 & 9 & 8 & 13 & 10 & 7 & 5 & 4 & 1 & 11 & 12 & 6 \end{bmatrix}.$$

W algorytmie EA zastosowano specjalnie zmodyfikowany operator krzyżowania z cięciem jednopunktowym, który nie powodował powstawania rozwiązań niedo-

puszczalnych oraz dostosowane do niego operatory mutacji, zmieniające kolejność kolumn w macierzy g.

Dokładny opis wykorzystanych operatorów ewolucyjnych, wykorzystanych do rozwiązania przedstawionego problemu, przedstawiono w pracy [4]. Szczegółowe porównanie algorytmów ewolucyjnych zarówno dla rozważanego problemu, jak również dla problemów tego typu uwzględniających inne kryteria jakości zaprezentowano w pracy [1].

3. Algorytm symulowanego wyżarzania SA

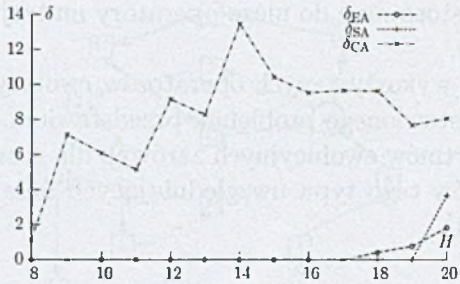
Drugim algorytmem heurystycznym, wykorzystanym w pracach, był algorytm symulowanego wyżarzania. Badany problem jest problemem optymalizacji kombinatorycznej, w związku z czym określenie otocznia bieżącego rozwiązania nie jest tak oczywiste, jak w przypadku problemów ciągłych. W literaturze przyjmuje się w takich przypadkach, że otoczeniem jest zbiór rozwiązań niewiele różniących się od bieżącego rozwiązania. W badanym problemie zaproponowano, aby za otoczenie bieżącego rozwiązania traktować wszystkie takie rozwiązania, w których jedno, losowo wybrane, zadanie jest uszeregowane na wszystkich możliwych pozycjach. W zaproponowanym algorytmie wykorzystano logarytmiczny schemat schładzania.

4. Algorytm deterministyczny CA

W takcie prowadzonych prac pojawiła się idea prostego algorytmu, służącego do szybkiego generowania rozwiązań wstępnych dla algorytmu SA. Działanie algorytmu CA polega na szeregowaniu kolejnych zadań na realizatorach wg kolejności ich numerów. Kolejne zadania są szeregowane na wszystkich możliwych pozycjach (względem już uszeregowanych zadań) i ostatecznie zadanie szeregowane jest na tej pozycji, która powoduje najmniejszy przyrost wartości kryterium.

5. Algorytmy hybrydowe EASA i EA(SA)

Zaprezentowane powyżej algorytmy heurystyczne EA i SA miały swoje zalety i wady. Algorytm EA dość szybko znajdował rozwiązania leżące w pobliżu optimum globalnego, jednak nawet znaczny wzrost liczby iteracji nie powodował znaczącej poprawy wartości kryterium. W przeciwieństwie do algorytmu EA, algorytm SA był w stanie znacząco poprawiać znalezione rozwiązanie, wymagał jednak znacznego powiększenia liczby iteracji. W ramach prowadzonych badań postanowiono przebadać dwa algorytmy hybrydowe, stanowiące połączenie obu podejść heurystycznych. W pierwszym z nich, nazwanym EASA, algorytm EA miał służyć do szybkiego znalezienia rozwiązania, które następnie stawało się rozwiązaniem początkowym algorytmu SA. W drugim przypadku, w algorytmie EA(SA), algo-



Rys. 2. Porównanie znalezionych wartości kryterium dla $H \leqslant 20$

rytm SA występował jako specyficzny operator ewolucyjny w algorytmie EA. Ze względu na to, iż liczba iteracji algorytmu EA jest duża, wywoływanie algorytmu SA w każdej z nich znacząco wydłużyłoby czas działania algorytmu EA(SA). Nie należało również wywoływać algorytmu SA dla wszystkich rozwiązań w każdej iteracji, ponieważ zbiór rozwiązań stałby się wtedy jednorodny. Z tych powodów zdecydowano, że w algorytmie EA(SA) algorytm SA będzie wywoływany co określoną liczbę iteracji, wyłącznie dla najlepszego rozwiązania znalezionego w danej iteracji.

Zasadę działania oraz szczegółowe wyniki przedstawiono w pracy [5]. Więcej informacji o zastosowaniu rozwiązań hybrydowych w algorytmach ewolucyjnych można znaleźć m.in. w [2].

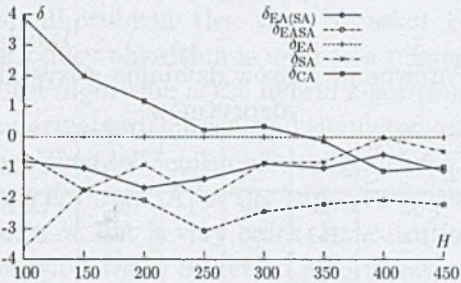
6. Badania symulacyjne

Badania symulacyjne składały się z dwóch etapów. W pierwszym etapie badań dobrano parametry algorytmów tak, aby uzyskać najlepsze wyniki. Następnie dla wybranych parametrów porównywano ze sobą poszczególne algorytmy. Wartości kryterium znalezione przez algorytmy heurystyczne dla problemów o małych rozmiarach ($H \leqslant 20$) porównano przy użyciu wskaźnika δ_x:

$$\delta_x = \frac{F_x - F_{BB}}{F_{BB}} \cdot 100\%,$$

gdzie x oznacza badany algorytm, zaś F_{BB} oznacza optymalną wartość kryterium dla badanego problemu. Wszystkie wyniki przedstawione na wykresach są uśrednionymi wynikami z dziesięciu uruchomień każdego z algorytmów. Jak wynika z rys. 2, dla problemów o $H \leqslant 17$ algorytmy EA i SA znajdowały rozwiązania dokładne. Dla większych H znajdowane rozwiązania stopniowo się pogarszały, przy czym były one gorsze od dokładnych tylko o $2-3\%$. Dla małych problemów algorytm CA uzyskiwał zdecydowanie gorsze wyniki.

Sytuacja zmieniła się dla problemów o dużych rozmiarach $100 \leqslant H \leqslant 450$. Na rys. 3 przedstawiono porównanie wartości kryteriów znajdowanych przez poszczególne algorytmy. Ze względu na brak znajomości dokładnych wartości kry-



Rys. 3. Porównanie znalezionych wartości kryteriów dla dużych wartości H

terium dla problemów o tak dużych rozmiarach, wartości kryteriów znajdowane przez badane algorytmy porównywano z wartościami znalezionymi przez algorytm EA, przy pomocy wskaźnika:

$$\delta_{\mathrm{x}} = \frac{F_{\mathrm{x}} - F_{\mathrm{EA}}}{F_{\mathrm{EA}}} \cdot 100\%.$$

Jak widać, dla zadań o dużych rozmiarach najlepsze wyniki osiąga algorytm hybrydowy EASA. Niewiele gorszy od niego jest algorytm SA. Warte podkreślenia są dobre (lepsze niż algorytmu EA) wyniki uzyskane przez algorytm CA. Algorytm CA jest bardzo prosty, ale dla największych badanych rozmiarów problemu uzyskuje wyniki lepsze od algorytmów heurystycznych, ustępując algorytmom hybrydowym.

Wyniki te są efektem zastosowanych w algorytmach EA i SA warunków stopu - w obu przypadkach założono, że algorytmy działają przez określoną z góry, stałą dla wszystkich badań, liczbę iteracji. Z uzyskanych wyników można wyciągnąć wniosek, że wraz ze wzrostem rozmiaru problemu należy również zwiększać liczbę iteracji w algorytmach EA i SA. Algorytm EA wykonywał $I = 100000$ iteracji, zaś algorytm SA - $K = 200000$. W przypadku algorytmu EASA algorytm EA wykonywał tylko $I = 1000$ iteracji, zaś algorytm SA - $K = 50000$, ale i tak znalezione przez algorytm EA początkowe rozwiązanie dla algorytmu SA pozwoliło wyznaczyć rozwiązania o lepszych wartościach kryterium. Warto zwrócić uwagę na czasy działania algorytmów przedstawione w tab. 1. Jak widać algorytm CA znajduje rozwiązanie w ciągu kilku sekund, zaś dla największych problemów ($H = 450$) czasy działania algorytmów sięgają tysięcy sekund.

7. Podsumowanie

W pracy przedstawiono porównanie opracowanych dotychczas algorytmów heurystycznych i hybrydowych, znajdujących rozwiązania dla problemu szeregowania zadań z uwzględnieniem ruchu realizatorów dla kryterium średniego czasu przepływu. Najlepsze efekty dało zastosowanie algorytmu hybrydowego EASA, który nawet dla największych rozmiarów badanych problemów znajdował rozwią-

Tabela 1
Porównanie czasów działania wszystkich algorytmów

H \ $T[s]$	T_{SA}	T_{EA}	T_{CA}	T_{EASA}	$T_{EA(SA)}$
50	49	148	1	51	1342
100	162	250	1	165	2534
150	374	439	1	378	5847
200	633	659	1	638	9895
250	1048	845	1	1056	16383
300	1431	958	2	1440	22368
350	1915	1128	2	1925	29931
400	2429	1252	2	2441	37965
450	3340	1391	2	3353	52200

zania lepsze niż pozostałe algorytmy. Warto również zwrócić uwagę na algorytm deterministyczny CA, który, pomimo swojej prostoty, dla problemów o dużych rozmiarach osiągał rezultaty lepsze od algorytmów heurystycznych EA i SA.

LITERATURA

1. Kowalski M., Thomas W.: Eksperymentalna ocena ewolucyjnych algorytmów szeregowania zadań na ruchomych realizatorach. W: M. Zaborowski (red.): Automatyzacja procesów dyskretnych, WNT, 2004.
2. Preux P., Talbi E.G.: Towards hybrid evolutionary algorithms. Intl. Tans. in Op. Res., t. 6, 1999, p. 557–570.
3. Thomas W.: Algorytm dokładny rozwiązania problemu szeregowania zadań z ruchowymi realizatorami dla kryterium MFT. W: M. Zaborowski (red.): Automatyzacja procesów dyskretnych, WNT, 2004.
4. Thomas W.: Application of the genetic algorithm to solve the task scheduling problem with moving executors. Proc. of AI-meth 2003, Gliwice, 2003.
5. Thomas W.: Hybrydowe algorytmy szeregowania zadań na ruchomych realizatorach dla kryterium średniego czasu przepływu. Krajowa Konferencja Automatyki, Warszawa, 2005.

Recenzent: Dr hab. inż. Konrad Wala, prof. AGH

Abstract

This paper concerns a task scheduling problem with moving executors for the mean flow time criterion. In the paper heuristic and hybrid algorithms are compared. Best criterion values found by the algorithms are compared with optimal

criterion value for the small problems (less than 20 tasks). For the larger problems ($H \geqslant 100$ tasks), evolutionary algorithm is used as a reference algorithm. The research shows that the best algorithm is the hybrid algorithm EASA that is a serial composition of evolutionary algorithm EA and simulated annealing algorithm SA. Very good results are achieved by deterministic algorithm CA, which outperforms both heuristic algorithms EA and SA for the large problems ($H = 450$ tasks). It is worth to note that algorithm CA is very quick. Its execution time is shorter than 2 seconds, whereas execution times of hybrid algorithms are about thousands of seconds for the largest examined problems.