

Czesław Smutnicki, Józef Grabowski,
Eugeniusz Nowicki, Stanisław Zdrzałka
Politechnika Wrocławska

BIBLIOTEKA PROCEDUR OPTIMALIZACJI DLA ZAGADNIEŃ KOLEJNOŚCIOWYCH
CZ. II - STRUKTURA BIBLIOTEKI. TECHNIKI OBLICZENIOWE. EFEKTYWNOŚĆ

Streszczenie. W artykule przedstawiono aktualny stan biblioteki procedur optymalizacji dla zagadnień kolejnościowych, będącej wynikiem realizacji podproblemu resortowego R.I.21.01. Przedstawiono także wyniki badań eksperymentalnych algorytmów oraz dyskusję zastosowanych technik obliczeniowych i efektywności algorytmów.

1. Wstęp

Biblioteka^o procedur optymalizacji dla zagadnień kolejnościowych jest wynikiem realizacji podproblemu resortowego R.I. 21.01 pt.: "Wybrane problemy i programy optymalizacji dyskretnej dla szeregowania zadań z zastosowaniem przy sterowaniu dyskretnymi procesami produkcyjnymi", realizowanego w latach 1982-85. Celem prowadzonych prac jest utworzenie biblioteki zawierającej dość obszerną grupę specjalizowanych algorytmów optymalizacyjnych /zarówno dokładnych jak i przybliżonych/ o różnych własnościach numerycznych. Algorytmy są konstruowane głównie w oparciu o prace własne z uwzględnieniem najnowszych metod rozwiązywania zagadnień kolejnościowych [4], ..., [8].

Realizowana biblioteka jest pierwszą w kraju próbą stworzenia kompleksowego pakietu procedur optymalizacyjnych dla zagadnień kolejnościowych.

2. Założenia oraz zasady tworzenia biblioteki

Biblioteka procedur optymalizacji zagadnień kolejnościowych /APOD/, będąca przedmiotem opracowania, została pomyślana jako system oprogramowania użytkowego dla maszyn cyfrowych serii ODRA 1300, przeznaczony do rozwiązywania szerokiej klasy zagadnień kolejnościowych.

Przy opracowaniu biblioteki przyjęto następujące założenia ogólne:

- /i/ biblioteka powinna zawierać algorytmy optymalizacyjne dla zagadnień najczęściej występujących w praktyce,
- /ii/ algorytmy umieszczone w bibliotece powinny się charakteryzować "dobrymi" własnościami obliczeniowymi,
- /iii/ biblioteka powinna zapewniać użytkownikowi wygodę w korzystaniu oraz swobodę wyboru sposobu użytkowania.

Konsekwencją założenia /i/ było wybranie do opracowania, z szerokiej klasy zagadnień kolejnościowych, grupy zagadnień o charakterze uniwersal-

nym /i jednocześnie podstawowym/. Zastosowania wybranych zagadnień kolejnościowych są związane z szeregowaniem zadań w dyskretnych procesach produkcyjnych z uwzględnieniem ewentualnego występowania wąskich gardeł. Wybór problemów kolejnościowych przedstawiony w Cz. I pracy [3] pozwala na wykorzystanie skonstruowanych algorytmów bezpośrednio /lub po niewielkich modyfikacjach/, co jednocześnie wynika z postulatu /iii/.

W konsekwencji założenia /ii/ zdecydowano się wybrać następujące rozwiązanie:

- biblioteka powinna zawierać grupę algorytmów specjalizowanych pod kątem rozwiązywania pojedynczych zagadnień lub grup zagadnień,
- algorytmy należy konstruować z uwzględnieniem najnowszych wyników badań w literaturze światowej, mając na względzie uzyskanie maksymalnej ich efektywności,
- dla zagadnień o dużej złożoności obliczeniowej /FP-zupełne/ należy skonstruować różne algorytmy rozwiązywania, zarówno dokładne jak i przybliżone,
- dla wszystkich algorytmów należy przedstawić ocenę złożoności obliczeniowej, zarówno teoretyczną jak i doświadczalną.

Konsekwencją założenia /iii/ jest przyjęcie szeregu następujących założeń szczegółowych dotyczących struktury biblioteki:

- /a/ w skład biblioteki wchodzi podprogramy oraz programy napisane w języku FORTRAN,
- /b/ wszystkie algorytmy optymalizacyjne oraz procedury pomocnicze są zrealizowane w postaci podprogramów,
- /c/ wszystkie podprogramy mają postać segmentów SUBROUTINE lub FUNCTION o nazwach utworzonych wg określonego klucza,
- /d/ sposób przekazywania parametrów między segmentami /np. kolejność występowania, typy tablic, deklaracje dynamiczne/ powinien zostać ujednolicony w celu zwiększenia uniwersalności,
- /e/ segmenty rozwiązujące to samo zagadnienie różnymi metodami powinny posiadać jednakowy sposób wywoływania,
- /f/ nowo tworzone podprogramy biblioteki powinny maksymalnie korzystać z opracowanych wcześniej procedur,
- /g/ algorytmy przewidziane do wykorzystania w różnych podprogramach biblioteki powinny być zrealizowane w postaci oddzielnych segmentów,
- /h/ dla umożliwienia użytkownikowi skorzystania z wybranych podprogramów bez konieczności tworzenia własnego programu, wybrane podprogramy realizujące główne procedury optymalizacji są wkomponowane w programy zawarte w bibliotece.

Przyjęta organizacja oprogramowania biblioteki pozwala na jej użytkowanie na jeden z dwóch sposobów:

- użytkownik tworzy własny program z wykorzystaniem podprogramów zawartych w bibliotece oraz segmentów własnych,
- użytkownik korzysta bezpośrednio z programów biblioteki.

3. Aktualny stan biblioteki APOD

W wyniku realizacji dwóch pierwszych etapów podproblemu R.I. 21.01 zostało opracowanych ok. 50 różnych procedur optymalizacyjnych i pomocniczych dla szeregu podstawowych zagadnień kolejnościowych występujących w dyskretnych procesach produkcyjnych [1]. W tabelicy 3.1 przedstawiono aktualny stan biblioteki, zaś na rys. 3.1 jej strukturę /z uwzględnieniem podziału na podprogramy i programy oraz zależności pomiędzy nimi/.

Aktualny stan biblioteki /liczba procedur/

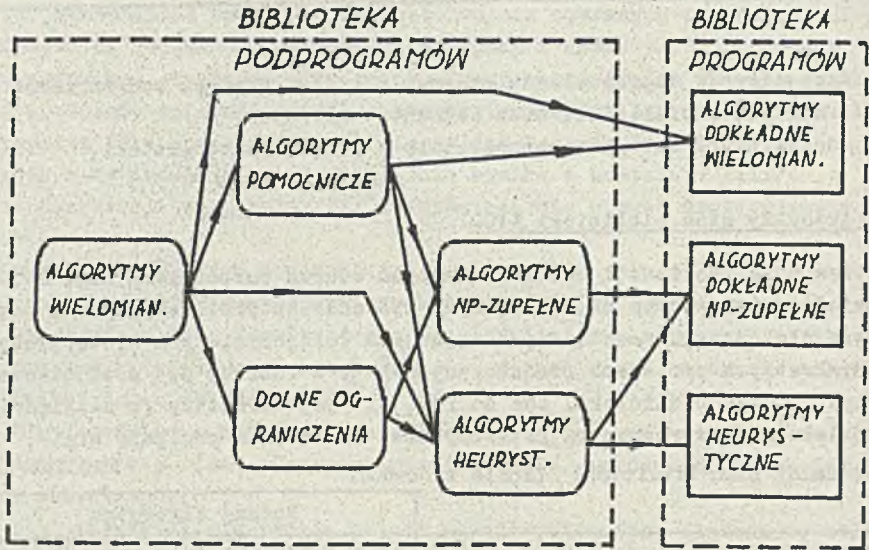
Tablica 3.1

Zagadnienie	Rodzaj algorytmu			
	dokład.	przybliż.	dol.ogr.	pomocn.
$n 1 r_1 \geq 0 C_{\max}$	1	-	-	-
$n 1 r_1 \geq 0, < C_{\max}$	1	-	-	1
$n 1 q_1 \geq 0 C_{\max}$	1	-	-	-
$n 1 q_1 \geq 0, < C_{\max}$	1	-	-	1
$n 1 L_{\max}$	1	-	-	-
$n 1 < L_{\max}$	1	-	-	1
$n 1 f_{\max}$	1	-	-	4
$n 2 P C_{\max}$	2	-	-	1
$n 3 P, M_2 - \text{bez ogr. } C_{\max}$	1	-	-	1
$n 1 r_1 \geq 0, q_1 \geq 0 C_{\max}$ $\pi/$	1	3	1	-
$n m P, < C_{\max}$ $\pi/$	1	14	2	2
$n m P, r_1 \geq 0, q_1 \geq 0, < C_{\max}$ $\pi/$	1	8	3	2
pomocnicze inne	-	-	-	5

$\pi/$ Zagadnienie NP-zupełne

Analiza Tabl. 3.1. wskazuje, że aktualnie biblioteka zawiera:

- grupę algorytmów optymalizacyjnych o niskim poziomie ogólności /zagadnienie 1-, 2-maszynowe/; są one głównie wykorzystywane jako algorytmy pomocnicze przy wyznaczaniu dolnych ograniczeń oraz rozwiązań przybliżonych,
- dość liczną grupę algorytmów heurystycznych zagadnień o średnim poziomie ogólności,
- grupę algorytmów optymalizacyjnych dokładnych /opartych na metodzie podziału i ograniczeń/ dla zagadnień o średnim poziomie ogólności,
- dość liczną grupę algorytmów pomocniczych.



Rys. 3.1. Struktura biblioteki

W latach 1984-85 przewiduje się uzupełnienie biblioteki o procedury optymalizacyjne zagadnień o dużym poziomie ogólności: $n|m|G|C_{\max}$, $n|m|G, r_i \geq 0|L_{\max}$ oraz $n|m|G, M_k$ - bez ogr. $|C_{\max}$, $n|m|I, r_i \geq 0, RT|L_{\max}$, $n|m|I, RT|C_{\max}$, $n|m|I, M_k$ - bez ogr., $RT|C_{\max}$ wraz z grupą niezbędnych procedur pomocniczych.

Niezależnie od zaplanowanych kierunków rozwoju przewiduje się w dalszej przyszłości rozwój biblioteki w następujących kierunkach:

/a/ rozwój ilościowy poprzez umieszczanie w bibliotece specjalizowanych procedur optymalizacyjnych dla nowych zagadnień kolejnościowych; przedmiotem opracowania byłyby w tym przypadku:

- zagadnienia szeregowania z dodatkowymi założeniami ograniczającymi np. postaci: M_k - bez ogr., nie składować, bez czekania, itd.,
- zagadnienia z ogólniejszymi postaciami funkcji celu,
- zagadnienia z ograniczeniami zasobowymi,
- zagadnienia o b. dużym rozmiarze z uwzględnieniem agregacji zadań,

/b/ rozwój jakościowy, polegający na umieszczeniu w bibliotece nowej generacji algorytmów o zwiększonej efektywności obliczeniowej; rozwój jakościowy polegałby zarówno na wprowadzaniu nowych teoretycznych własności zagadnień poprawiających cechy obliczeniowe algorytmu jak i na zmianach organizacyjno-programowych istniejących procedur, zmierzających do zwiększenia szybkości działania,

/c/ rozwój strukturalny polegający na konstrukcji, w kolejnym etapie, specjalizowanego makrogeneratora procedur optymalizacyjnych, umożliwiającego generowanie - dla podanego typu zagadnienia kolejnościowego - programu najefektywniejszego obliczeniowo.

4. Techniki obliczeniowe. Efektywność

W przypadku zagadnień szeregowania /optymalizacji dyskretnej/ niebagatelną sprawą jest złożoność obliczeniowa algorytmu. Miara tej złożoności limituje praktyczny rozmiar zagadnienia, jakie jesteśmy w stanie rozwiązać w "rozsądnym" czasie. Dla zagadnień stosunkowo prostych istnieją algorytmy o znanej wielomianowej złożoności obliczeniowej i dobór właściwej techniki realizacji programowej tych algorytmów zapewnia otrzymanie stosunkowo krótkich czasów obliczeń nawet dla zagadnień o dużym rozmiarze. Inaczej jest w przypadku algorytmów dokładnych dla zagadnień o średnim i dużym poziomie ogólności. Ze względu na ich NP-zupełność oraz silny wzrost czasu obliczeń wraz z rozmiarem zagadnienia, dobór odpowiedniej techniki obliczeniowej /tzw. implementacja/ decyduje o praktycznej przydatności lub nieprzydatności algorytmu. Elementami mającymi wpływ na czas obliczeń są: liczba i rodzaj własności eliminacyjnych zastosowanych w algorytmie, technika przeglądania drzewa rozwiązań, złożoność pojedynczej iteracji, moc dolnego ograniczenia, wartość początkowego górnego ograniczenia, itp. Ocena praktycznej przydatności algorytmu uwzględnia m.in. takie elementy, jak: średni czas obliczeń do chwili otrzymania rozwiązania optymalnego, wartość rozwiązania przybliżonego otrzymana w dowolnej chwili przerwania pracy algorytmu /tzw. szybkość zbiegania do rozw. optymalnego/. Zwykle realizowana implementacja uwzględnia kompromis między nakładami obliczeniowymi a szybkością przeglądania drzewa.

Alternatywą dla algorytmów dokładnych opartych na technikach przeglądania drzewa są wielomianowe algorytmy heurystyczne. W tym przypadku oceniana jest, oprócz czasu obliczeń, dokładność otrzymywanego rozwiązania suboptymalnego.

Uwzględniając wszystkie opisane powyżej elementy oraz obszernie wyniki badań teoretycznych i doświadczalnych, w bibliotece umieszczono oprócz algorytmów heurystycznych charakteryzujących się małym błędem /patrz tabl. 5.1/ algorytmy dokładne charakteryzujące się następującymi cechami:

- /i/ algorytmy te mogą funkcjonować jako algorytmy dokładne, bądź jako algorytmy /nie wielomianowe/ o zagwarantowanej dokładności,
- /ii/ algorytmy posiadają w każdej iteracji rozwiązanie dopuszczalne, poprawiając je w kolejnych krokach,
- /iii/ algorytmy otrzymują stosunkowo szybko dobre rozwiązanie,

/iv/ algorytmy wykorzystują eliminacyjne własności ścieżki krytycznej i bloków zadań, silnie redukujące wielkość przeglądanych drzewa rozwiązań,

/v/ algorytmy posiadają możliwość zastosowania różnych metod wyznaczania dolnego ograniczenia, w zależności od oczekiwanej skali trudności zagadnienia i czasu przeznaczanego na obliczenia.

Z technicznego punktu widzenia procedury realizujące w/w algorytmy są programami ze stosową organizacją obszaru danych i dość złożoną jego adresacją. Sam zaś algorytm jest zwykle zbiorem kilku różnych współzależnych algorytmów.

Szereg przeprowadzonych badań numerycznych dowiodło, że zrealizowane algorytmy są efektywniejsze od znanych do tej pory [5], [9] oraz mogą być stosowane w wybranych systemach sterowania dyskretnymi procesami produkcyjnymi.

5. Wyniki badań testowych

Dla każdego z opracowanych algorytmów przeprowadzono badanie testowe /dla losowej serii danych/ dotyczące średniego czasu obliczeń dla zagadnień o określonym rozmiarze. Dla zagadnień NP-zupełnych badano dodatkowo:

- liczbę iteracji algorytmu dokładnego niezbędną do uzyskania rozwiązania optymalnego,
- odległość rozwiązania suboptymalnego od optymalnego - dla algorytmów heurystycznych.

Wybrane wyniki przedstawiono w tabl. 5.1-5.2. Sposób generacji zestawu danych testujących dla algorytmów jest identyczny z opisanym w pracy [6]. Dodatkowo dla zagadnienia $n|m|P|C_{max}$ testowanie przeprowadzono na przykładach otrzymanych od A.H.G. Rinnooy Kana z Instytutu Ekonometrii w Rotterdamie. Szczegółowe wyniki można znaleźć w pracach [1], [4], [5].

Uzyskane wyniki potwierdzają "dobre" własności obliczeniowe algorytmów opartych na wykorzystaniu "podejścia blokowego". Rzeczywiście, z analizy Tabl. 5.1 - 5.2 wynika, że zaprogramowane algorytmy heurystyczne powodują około 3 - krotne zmniejszenie czasu obliczeń przy średnim błędzie około 5%. Z kolei algorytmy dokładne w zdecydowanej większości przykładów /patrz mediana liczby węzłów/ znajdują rozwiązanie przeglądając tylko jeden wierzchołek drzewa. Ponadto, średni czas obliczeń algorytmów dokładnych przykładowo dla najogólniejszego zagadnienia $n|m|P$, $r_1 \geq 0 | L_{max}$ przy n zmieniającym się od 6 do 50 waha się w granicach 0.1 - 2.4 [s].

Tablica 5.1

Wyniki testowania algorytmów heurystycznych

Zagadnienie	Algo- rytm	Rozmiar n/m	Średni błąd wzgl. [%]	Max błąd wzgl. [%]	Średni czas obl. [s]
$n 1 r_i \geq 0, q_i \geq 0 $ $ C_{\max}^{\text{KX}}/$	[7]	20/1	0.00	0.08	0.02
		40/1	0.00	0.00	0.05
		80/1	0.00	0.09	0.20
$n m P C_{\max}^{\text{KX}}/$	[2]	6/3	1.6	18.6	0.10
		6/5	1.8	5.6	0.12
		6/8	2.1	7.6	0.20
		10/3	0.9	5.2	0.16
		10/5	2.9	13.0	0.34
		15/3	0.9	3.3	0.20
		20/3	1.3	7.8	0.30
50/3	0.6	2.2	1.00		
$n m P, r_i \geq 0 L_{\max}^{\text{KX}}/$	własny	6/2	4.8	33.3	0.10
		6/3	2.8	25.3	0.18
		6/5	7.2	25.8	0.38
		6/8	8.8	30.6	0.70
		10/2	4.8	32.9	0.20
		10/3	4.5	22.1	0.40
		10/5	7.3	41.3	0.78
		15/3	4.8	39.3	0.20
		20/2	0.4	37.1	0.38
		20/3	4.5	37.4	0.66
		20/5	7.7	32.8	1.70
		30/2	4.7	60.7	0.60
		40/2	6.3	74.0	0.99
50/3	7.0	63.4	2.40		

KX/ liczba przykładów testujących 80 dla każdego n/m

KXK/ liczba przykładów testujących 12 dla każdego n/m

Tablica 5.2

Wyniki testowania algorytmów dokładnych

Zaradnienie	Algorytm	Rozmiar	Liczba węzłów		Czas obl. s	
		n/m	A	M	A	M
$n 1 r_1 \geq 0, q_1 \geq 0 C_{\max}^{*/}$	[1]	20/1	1.24	1	0.07	0.06
		40/1	1.35	1	0.25	0.25
		80/1	1.18	1	0.59	0.50
		150/1	1.15	1	1.87	1.75
		200/1	1.07	1	3.13	2.75
$n m P C_{\max}^{**/}$	[1], [5]	6/3	15.1	1	2.8	0.2
		6/5	51.2	10	16.2	0.3
		6/8	64.8	15	26.5	0.4
		10/3	28.1	4	6.1	0.2
		10/5	58.5	6	21.8	0.3
		15/3	31.3	3	10.2	0.3
		20/3	38.9	2	18.2	0.4
		50/3	82.7	10	122.7	1.7
$n m P, r_1 \geq 0 L_{\max}^{*/}$	[1], [5]	6/2	2.7	1	0.3	0.1
		6/3	5.0	1	0.9	0.2
		6/5	7.9	1	1.7	0.3
		6/8	8.9	1	3.0	0.4
		10/2	4.6	1	0.9	0.1
		10/3	6.1	1	1.2	0.2
		10/5	8.9	1	2.1	0.3
		15/3	7.4	1	1.7	0.3
		20/2	5.4	1	1.6	0.3
		20/3	6.6	1	1.9	0.4
		20/5	10.3	2	3.6	0.9
		30/2	6.1	1	2.7	0.7
		40/2	6.7	1	4.1	0.9
		50/3	7.5	1	7.9	1.8

*/ dolne ograniczenie 1-maszynowe; liczba przykładów testujących 80 dla każdego n/m; A - średnia, M - mediana,

**/ dolne ograniczenie 2-maszynowe; liczba przykładów testujących 24 dla każdego n/m; A - średnia, M - mediana.

6. Zakończenie

Procedury zawarte w realizowanej bibliotece APOD mogą być wykorzystane w praktyce bezpośrednio, lub też mogą stanowić materiał wyjściowy do opracowywania nowych specjalizowanych algorytmów szeregowania zadań. Oczywiście każda kolejna specjalizacja może poprawiać własności obliczeniowe algorytmu. Opracowane algorytmy mogą być również wykorzystywane przy realizacji systemów planowania i zarządzania opartych na taśmowych lub dyskowych zbiorach danych dotyczących procesu produkcyjnego.

LITERATURA

- [1] Wybrane problemy i programy optymalizacji dyskretnej dla szeregowania zadań z zastosowaniem przy sterowaniu dyskretnymi procesami produkcyjnymi /pr. zbiorowa/, Raporty Serii Sprawozdania nr 20/82, 21/82, 22/82, Wydawnictwo Politechniki Wrocławskiej, Wrocław 1982, nr 7/83, 8/83, Wrocław 1983
- [2] Campbell H.H., Dudek R.A., Smith M.L.: A Heuristic Algorithm for the n Job, m Machine Sequencing problem, Management Science, vol 16, s. 630-637 B, 1970
- [3] Grabowski J., Nowicki E., Smutnicki C., Zdrzałka S.: Biblioteka procedur optymalizacji dla zagadnień kolejnościowych, Cz. I. Klasyfikacja, własności i algorytmy. ZN Pol.Sł. /referat na IV Krajową Konferencję Automatyzacji DPP/, s. Automatyka, nr 74, 1984
- [4] Grabowski J.: On two-machine scheduling with release and due dates to minimize maximum lateness, Opsearch 17, 1980 s. 133-154
- [5] Grabowski J., Skubalska E., Smutnicki C.: On Flow Shop with Release and Due Dates to Minimize Maximum Lateness, Journal of Opnal Res. Soc., vol. 34, nr 7, s. 615-620, 1983
- [6] Lenstra J.K.: Sequencing by Enumerative Methods, Mathematical Centre Track 69, Matematish Centrum, Amsterdam 1977
- [7] Potts C.W.: An Adaptive Branching Rule for the Permutation Flow-Shop Problem, European Journal of Operational Research 5, 1980, s. 19-25
- [8] Rinnooy Kan A.H.G.: Machine Scheduling Problems. Classification, Complexity and Computations, Nijhoff, The Hague, 1976
- [9] Boura R.W.: Job-shop scheduling: A comparison of three enumeration schemes in a branch-and-bound approach. Technical Report, Erasmus University Rotterdam, Department of Econometrics/Operations Research, 1982.

Recenzent: Prof.dr hab.inż. Jan Węglarz

Wpłynęło do Redakcji do 30.03.1984r.

БИБЛИОТЕКА ОПТИМИЗАЦИОННЫХ ПРОЦЕДУР ДЛЯ ПРОБЛЕМ ЧЕРЕДОВАНИЯ.
Часть II. СТРУКТУРА БИБЛИОТЕКИ . ВЫЧИСЛИТЕЛЬНЫЕ ВОЗМОЖНОСТИ. ЭФФЕКТИВ-
НОСТЬ

Р е з ю м е

В статье представлено актуальное состояние библиотеки процедур оптимизации для проблем чередования, являющейся результатом реализации ведомственной подпроблемы Р.И. 21. 01. Представлены также результаты экспериментальных исследований алгоритмов и дискусию вычислительных возможностей и эффективность.

THE PACKAGE OF OPTIMIZATION PROCEDURES FOR SEQUENCING JOBS. PART II.
PACKAGE STRUCTURE, COMPUTATIONAL TECHNICS, EFFECTIVENESS

С и ж м а р у

In the paper the actual state of the package of optimization procedures for sequencing jobs is presented. Some computational results are shown. The discussion of applied computational techniques and effectiveness is presented. The package is the results of research made in the frame of Departmental Subproblem R.I. 21.01.