

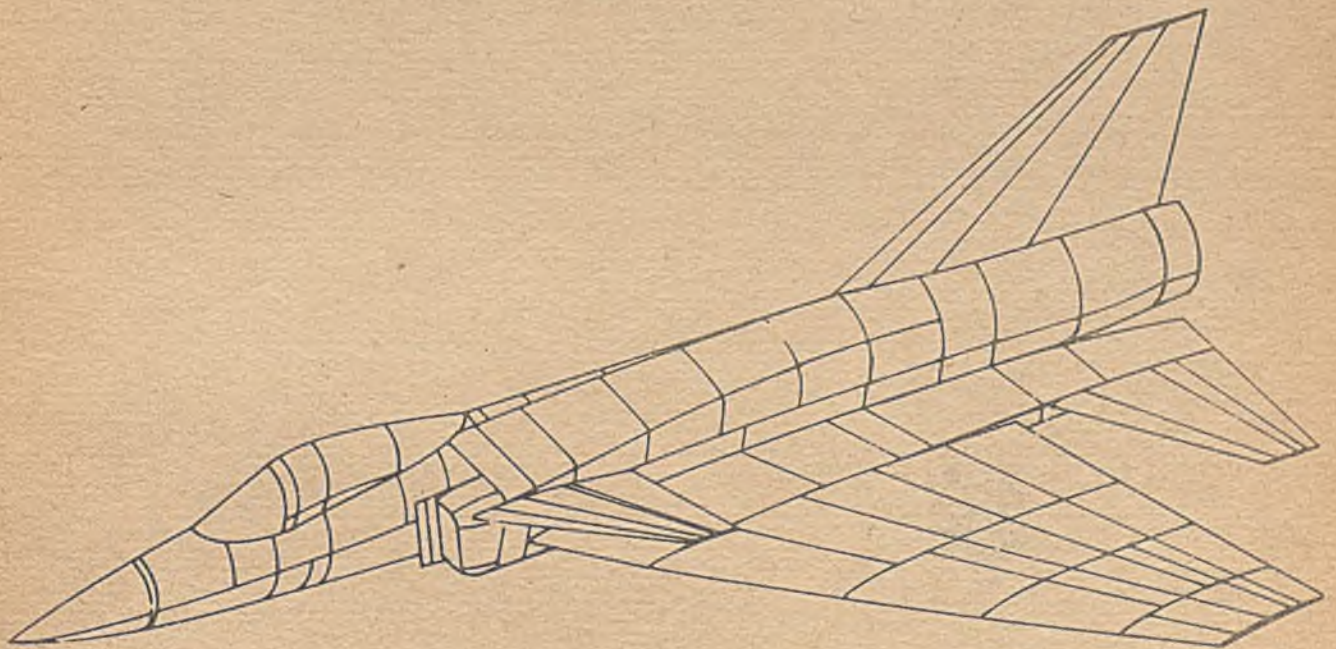
# biuletyn informacyjny

P.3057/79

1  
'79



OBIKTOWE  
SYSTEMY  
KOMPUTEROWE



Zjednoczenie Przemysłu Automatyki i Aparatury Pomiarowej „MERA”  
Instytut Maszyn Matematycznych „MERA IMM” Branżowy Ośrodek INTE

**Rysunek na okładce:** Projekt trasowania powłoki samolotu /wg Computer Aided Design 1977 nr 4/

**U w a g a .** Rysunek na okładce numeru 5-6/1977 wykonała Barbara KOSTRZEWSKA

**Druk okładki** IMM zam. 112/79 nakł. 820 egz.

P. 30.57/79

Komputeryzacja projektowania inżynierskiego

Spis treści

Содержание

Contents

Rohatyński R. - Niektóre zagadnienia projektowania wspomagającego komputerowo ..... s. 3	Рохатыньски Р.-Некоторые вопросы с применением компьютерных вспомогательных средств .....с. 3	Rohatyński R. - Some problems of computer-aided design..p. 3
Grodzicki S., Olech J. - Wybrane aspekty zobrazowania wyników obliczeń komputerowych .. s. 12	Гродзицки С., Олех Я.- Избранные аспекты представления результатов компьютерных вычислений ....с. 12	Grodzicki S., Olech J. - Selected aspects of the computer computing results .....p. 12
Mocala J. - Przegląd metod wspomaganego komputerowo projektowania linii elektroenergetycznych ..... s. 36	Моцала Е.- Обзор методов проектирования воздушных линий электропередачи с применением компьютерных вспомогательных средств .....с. 36	Mocala J. - A review of computer aided design of electropower lines .....p. 36
z cyklu: Programowanie obrabiarek	из цикла: Программирование обрабатывающих станков	from the cycle: Machine tool design
Berthold A., Galicki J., Topolski S.- Automatyczna synteza elementów programu dla obrabiarek sterowanych numerycznie ..... s. 47	Бертольд А., Галицки Е., Топольски С.- Автоматический синтез элементов программ для станков с числовым программным управлением .....с. 47	Berthold A., Galicki J., Topolski J.- Automatic synthesis of program elements for numerical control machines .p. 47
Bobińska I. - Automatyczna budowa technologicznych dróg obróbki detali ..... s. 53	Бобиньска И.- Автоматическая конструкция технологических линий обработки деталей .....с. 53	Bobińska I.- Automatic construction of technological process for metalworking ....p. 53

319  
P1

D W U M I E S I Ę C Z N I K

Wydaje:

CENTRUM NAUKOWO-PRODUKCYJNE TECHNIK KOMPUTEROWYCH I POMIARÓW  
INSTYTUT MASZYN MATEMATYCZNYCH  
Branżowy Ośrodek Informacji Naukowej Technicznej i Ekonomicznej

KOMITET REDAKCYJNY

dr inż. Stanisława BONKOWICZ-SITTAUER, mgr Hanna DROZDOWSKA  
/sekretarz redakcji/, dr inż. Marek HOLYŃSKI  
doc.dr inż. Henryk ORŁOWSKI /redaktor naczelny/,  
mgr inż. Jerzy MYSIOR, mgr inż. Józef SZMYD, mgr Robert ZAJĄC

Opracowanie graficzne: Barbara KOSTRZEWSKA

Adres redakcji: ul. Krzywickiego 34, 02-078 Warszawa  
tel. 21-84-41 w. 431 lub 28-37-29

doc. dr hab. inż. Ryszard ROMATYŃSKI  
Politechnika Wroclawska

## Niektóre zagadnienia projektowania wspomaganego komputerowo

### 1. Sformułowanie zagadnienia

Współczesna wiedza o procesach projektowo-konstrukcyjnych nie jest jeszcze zadowalająca, mimo dużych postępów w ostatnich dwóch dziesięcioleciach. Nie maleje rozbieżność między teorią i praktyką projektowania. Praktyczne osiągnięcia usprawniające proces projektowania związane są z konkretnymi dziedzinami techniki i nie są przenoszone do innych zastosowań. Zbyt duża rola ciągle jeszcze przypada intuicji i doświadczeniu projektantów, zwłaszcza przy poszukiwaniu i ocenie rozwiązań zadania projektowego. Zastosowanie komputerów do projektowania okazało się trudniejsze niż oczekiwano i jak dotąd nie wpłynęło istotnie na poprawę wyników projektowania. Przyczyny tych niepowodzeń są niezwykle złożone i na pełne ich wyjaśnienie trzeba jeszcze poczekać.

Jedną z istotnych przyczyn tkwi w niedostatecznym poznaniu wymagań procesu projektowania i jego powiązań z procesem wytwarzania. Drugim powodem wydaje się być nieuwzględnianie faktu, że efektywne wprowadzenie komputeryzacji wymaga dobrego rozpoznania struktury projektowania, ponieważ jest to warunkiem racjonalnego podziału zadań między człowieka i komputer.

Zamierzeniem niniejszej pracy jest omówienie wymienionych zagadnień ze szczególnym uwzględnieniem tych spraw, które mogą mieć istotny wpływ na sprawność projektowania. Najpierw będzie więc ogólnie omówiona relacja między projektowaniem i procesem wytwarzania a następnie, bardziej szczegółowo, różne zagadnienia związane z zastosowaniem środków elektronicznej techniki obliczeniowej. Dwa z poruszonych zagadnień zasługują na szczególną uwagę. Pierwsze dotyczy operacyjnych cech, które powinien posiadać system komputerowy, aby jego użyteczność dla rozwiązywania zadań projektowych była jak największa. Cechy te dają się przewidzieć na podstawie jakościowej charakterystyki zadań projektowych, a przykład takiej analizy podano w punkcie "Zadania projektowo realizowane przez komputer". Drugie ważne zagadnienie - to ocena poziomu automatyzacji projektowania na podstawie cybernetycznego podziału czynności projektotwórczych między człowiekiem i komputerem. Jest to omówione w punkcie "Działania projektowe wykonywane na komputerze", a wynikające z tego wnioski na temat strategii rozwoju komputeryzacji projektowania sformułowano w punkcie "Strategia procesu automatyzacji projektowania".

Rozważania i analizy przedstawione w niniejszej pracy nie wyczerpują problemów komputerowego wspomaganie projektowania, a żadne z poruszonych zagadnień nie zostało podane w formie zamkniętej. Praca jest właściwie tylko omówieniem kilku wybranych, istotnych dla rozwoju projektowania wspomaganego komputerem problemów i spełni swoje zadanie jeśli Czytelnik zaakceptuje proponowany sposób myślenia lub nie zgodzi się z nim i przedstawi doskonalsze idee.

### 2. Projektowanie etapów procesu wytwarzania

Systemowe ujęcie projektowania określa się jako część złożonego procesu uzyskiwania nowego wytworu [1], [2].

W tym procesie podstawowym zadaniem projektowania jest wytworzenie informacji niezbędnej do wykonania wytworu spełniającego postawione wymagania. Wymagania te są zwykle bardzo liczne i często przeciwstawne. Uwzględniają one potrzeby procesu wytwarzania, dystrybucji, eksploatacji, ergonomii, ochrony środowiska itp.

---

autor niniejszego artykułu jest docentem w Instytucie Konstrukcji i Eksploatacji Maszyn Politechniki Wrocławskiej i kierownikiem Laboratorium Projektowania Wspomaganego Komputerem w Mechanice oraz kierownikiem Zespołu Badawczego Modelowania Urządzeń Hydraulicznych; specjalizuje się w komputerowo wspomaganym metodach projektowania urządzeń hydraulicznych; jest autorem licznych prac z tego zakresu.

Projektowanie spełnia w procesie wytwarzania funkcje kreująco-sterujące. Znamienne dla projektowania jest to, że przy stosunkowo małych kosztach własnych decyduje ono o jakości wytworu, kosztach jego produkcji i eksploatacji. Koszt prac projektowo-konstrukcyjnych w przemyśle maszynowym wynosi zwykle tylko 10-20% całkowitych kosztów wytwarzania. Bardziej znaczący jest czas projektowania, gdyż co najmniej 50% czasu realizacji zamówienia przypada na prace projektowo-konstrukcyjne i przygotowawcze [3].

Wielu specjalistów wyraża przekonanie, że dalsze usprawnianie metod organizacji i technologii produkcji będzie mało skuteczne, jeśli nie nastąpi zdecydowany postęp w dziedzinie projektowania i konstruowania. Można wymienić wiele czynników składających się na pojęcie postępu w projektowaniu, ale nadrzędny cel projektowania jest zbieżny z celem całego systemu wytwórczego: wytworzenia obiektu o wymaganej jakości w zadanym czasie. Inne wskaźniki jakości projektowania, np. racjonalizacja i skrócenie czasu projektowania, minimalizacja zużycia zasobów materialnych itp. mogą być tylko wtedy brane pod uwagę, jeśli nie przeszkadza to realizacji celu nadrzędnego.

Jednym ze środków usprawnienia procesu projektowo-konstrukcyjnego, potencjalnie najbardziej skutecznym jest wspomaganie komputerowe. Komputery mogą być obecnie z powodzeniem stosowane:

- w obliczeniach inżynierskich,
- do przechowywania informacji,
- w procesie poszukiwania i oceny rozwiązań,
- do wykonywania dokumentacji technicznej,
- do opracowania ofert,
- do bezpośredniego sterowania procesami technologicznymi.

### 3. Projektowanie a środki elektronicznej techniki obliczeniowej

#### 3.1. Proces projektowania

Powien model procesu projektowania

Wprowadzenie komputeryzacji do projektowania wymaga stworzenia odpowiedniego do tych celów modelu procesu projektowania. Musi on być przedstawiony w postaci procedury jawnie opisującej przebieg projektowania, zawierającej kompletny zbiór algorytmów, reguł podejmowania decyzji, podstawowych danych, zależności matematyczno-fizycznych itd. Sformułowanie takich procedur w dziedzinach techniki mających wieloletnią tradycję, np. w budowie maszyn, jest w zasadzie zawsze możliwe. Na przykład, dla układów napędu i sterowania hydraulicznego udało się opracować system wzajemnie powiązanych algorytmów obejmujących proces projektowania od sformułowania zadania do projektu wstępnego [4], [5]. Proces projektowania jest tam podzielony na kilka etapów:

- formułowanie problemu,
- analiza problemu i zbieranie informacji,
- poszukiwanie rozwiązań,
- ocena i wybór rozwiązań,
- opracowanie wybranego rozwiązania na poziomie projektu wstępnego,
- opracowanie rozwiązania na poziomie projektu szczegółowego,
- wykonanie i badanie prototypu.

Momenty przejścia od etapu do etapu mogą nie być wyraźne, ponieważ realny proces projektowania składa się z iteracyjnie wykonywanych czynności heurystycznych i algorytmicznych. Czynności heurystyczne nie dają się ściśle sformalizować i często mają twórczy charakter, jak np. poszukiwanie sposobów realizacji zadanej funkcji albo modelowanie struktury i kształtu projektowanego obiektu. Czynności algorytmiczne są zdeterminowane i na ogół dają się łatwo zaprogramować na maszynę cyfrową.

Możliwość zastosowania komputera w procesie projektowania pojawia się wyraźnie już przy poszukiwaniu rozwiązań, np. jeśli przestrzeń rozwiązań przedstawić w postaci wielowymiarowej skrzynki morfologicznej, to można wykorzystać komputer do systematycznego lub losowego przeszukiwania tej przestrzeni. Znane są także eksperymentalne programy wspomagające heurystyczne zdolności ludzkiego umysłu [6].

Przy ocenie i wyborze rozwiązań, a także w pierwszej fazie projektowania wstępnego, wykonuje się szkice wariantów rozwiązań i wstępne obliczenia podstawowych rozmiarów, przepływów energii i masy itp. Obliczenia te mają charakter iteracyjny i opierają się najczęściej na empirycznych i półempirycznych wzorach. Zastosowanie komputera do tych obliczeń umożliwia dokładne wykonanie tych na ogół wieloparametrowych i pracochłonnych analiz, jakkolwiek wymagałoby to zapewne przygotowania odpowiednich programów konwersacyjnych lub problemowo zorientowanego języka programowania.

W czasie projektowania wstępnego wykonuje się badania modeli matematycznych charakteryzujących wybrane własności projektowanego obiektu. Stosowane jest również badanie systemów złożonych z obiektów fizycznych, współpracujących z modelami symulacyjnymi. Jest to uzasadnione, gdyż eksperymentowanie na teoretycznych, zadowalająco adekwatnych modelach jest znacznie tańsze od wykonywania i badania fizycznych prototypów. Wszeczhronne badanie modeli matematycznych i optymalizacja obiektu projektowania mogą być zrealizowane tylko na komputerach cyfrowych lub analogowo-cyfrowych.

W etapie projektowania szczegółowego opracowuje się dokładne charakterystyki elementów projektowanego obiektu. Wymaga to wykonania wielu specjalistycznych obliczeń, które mogą być wydawnie skrócone przez zastosowanie środków elektronicznej techniki obliczeniowej. Bardzo pracochłonną czynnością jest również opracowanie dokumentacji technicznej. Na podstawie badań dużej liczby projektów stwierdzono, że najwięcej czasu, bo ok. 30% zużywa się na wykonanie rysunków, a na obliczenia przeznaczają się tylko ok. 5% całkowitego czasu wykonania projektu [3]. Błędem byłoby jednak wnioskować na tej podstawie o drugorzędnej roli obliczeń. W tradycyjnym projektowaniu maszyn konstruktorzy wykonują tylko obliczenia przybliżone i szacunkowe, zabezpieczając się zwiększonymi współczynnikami bezpieczeństwa. W projektowaniu wspomaganym komputerowo można wykorzystać zaawansowane metody analizy i optymalizacji, co umożliwia dokładniejsze przewidywanie zachowania się projektowanego obiektu i zwiększa znaczenie obliczeń w projektowaniu.

#### Rodzaje zadań projektowych

Zadania projektowe mogą być klasyfikowane według stopnia oryginalności projektowanego urządzenia lub według wymagań stawianych systemowi komputerowemu.

Ze względu na oryginalność projektowanego urządzenia można wyróżnić:

- projektowanie urządzeń oryginalnych,
- projektowanie wariantów (modyfikacji) wcześniej opracowanych urządzeń,
- optymalizację urządzeń.

Urządzenia oryginalne charakteryzują się nowym sposobem realizacji zadanej funkcji roboczej i nowym wykonaniem konstrukcyjnym. Projektowanie urządzeń oryginalnych daje dużą swobodę wyboru różnorodnych rozwiązań, ale ryzyko niepowodzenia projektu jest również duże. Przy projektowaniu takich urządzeń mogą być wykorzystane programy stymulujące wyobraźnię projektanta, programy przeszukiwania przestrzeni rozwiązań oraz programy obliczeniowe ogólnego zastosowania, np. programy obliczania sił, ciśnień, natężeń przepływu itp.

Przy projektowaniu wariantów istnieje już podstawowe rozwiązanie określające zasadę działania urządzenia. Warianty podstawowej koncepcji mają na celu dostosowanie urządzenia do zmienionych warunków pracy i wymagań. Jest to najczęściej występujący w praktyce projektowo-konstrukcyjny typ zadania projektowego. W budowie maszyn, ze względu na zaawansowaną typizację i standardyzację elementów, takie zadania dają się w dużym zakresie zalgorytmizować i wspomagać komputerowo.

W projektowaniu optymalizacyjnym dąży się do optymalizacji istniejącego urządzenia ze względu na przyjęte kryterium, np. poszukuje się rozwiązania charakteryzującego się minimalnym ciężarem przy zadanej wydajności i ograniczonych kosztach produkcji. Obliczenia optymalizacyjne są pracochłonne i kosztowne, ale są na ogół tańsze od badań doświadczalnych, które byłyby konieczne dla osiągnięcia tych samych wyników.

Optymalizacja projektowanego obiektu jest najczęściej zadaniem bardzo złożonym. Kryterium optymalności jest rzadko dobrze określone; przeważnie wyraża się je za pomocą kombinacji zależnych

działania i kosztu. Ze względu na złożoność metod optymalizacji oraz trudności i koszty ich realizacji, obliczenia optymalizacyjne były początkowo wykonywane tylko tam, gdzie były one niezbędne lub zapewniałyby zysk, jak np. w przemyśle lotniczym i raketowym, w przemyśle stoczniowym i w produkcji masowej. Obecnie dość powszechnie stosuje się już uproszczone obliczenia optymalizacyjne w budowie maszyn. Wykonanie rachunku optymalizacyjnego wymaga uniwersalnych programów komputerowych. W tej dziedzinie wiele jest jeszcze do zrobienia, ponieważ realne problemy są zbyt złożone dla klasycznych metod optymalizacji.

Inny sposób klasyfikacji zadań projektowych przedstawiono w [7]. Zadania scharakteryzowano za pomocą czasu obliczania, liczby danych wejściowych i wyjściowych, pożądanej postaci danych oraz zakresu ingerencji człowieka w tok obliczeń. Następnie, za pomocą trójstopniowej skali, oszacowano przydatność różnych sposobów wykorzystania komputera do zadań o tych cechach. Wyniki zestawiono w tabl. 1. Ułatwia ona wybór sposobu współpracy z systemem komputerowym, jeśli charakterystyka zadania jest określona. W pracy [7] tabl. 1 została wykorzystana do racjonalnego wyboru wyposażenia laboratorium projektowania wspomaganego komputerowo w mechanice. W tym celu najpierw określono charakterystyczne cechy typowych zadań obliczeniowych związanych z procesem dydaktycznym na wydziałach mechanicznych oraz oszacowano liczbę tych zadań rocznie. Wyniki zestawiono w tab. 2.

Tab. 1

### UŻYTECZNOŚĆ SPRZĘTU KOMPUTEROWEGO W ZALEŻNOŚCI OD TYPU ZADANIA

Oznaczenia:  $\oplus$  - bardzo użyteczny  $\ominus$  - mało użyteczny  $\bigcirc$  - użyteczność ograniczona

Sposób wykorzystania systemu komputerowego		Praca wsadowa		TERMINAL w TIME-SHARINGU		PIÓRO ŚWIETLNE
		Zwykła	Z urządzeniami graficznego wyjścia	ALFANUMERYCZNY	GRAFICZNY	
1	2	3	4	5	6	7
1 CZAS OBLICZANIA	Długi /minuty/	$\oplus$	$\oplus$	$\bigcirc$	$\bigcirc$	$\ominus$
	Krótki /sekundy/	$\bigcirc$	$\bigcirc$	$\oplus$	$\oplus$	$\oplus$
2 LICZBA DANYCH WEJŚCIA/WYJŚCIA	Duża	$\oplus$	$\oplus$	$\ominus$	$\bigcirc$	$\bigcirc$
	Mala	$\bigcirc$	$\bigcirc$	$\oplus$	$\oplus$	$\oplus$
3 POŻĄDANA POSTAĆ DANYCH WEJŚCIA/WYJŚCIA	Cyfrowa	$\oplus$	$\bigcirc$	$\oplus$	$\bigcirc$	$\ominus$
	Graficzna	$\ominus$	$\oplus$	$\ominus$	$\oplus$	$\oplus$
4 INGERENCJA W TOK OBLICZEŃ	Pożądana	$\ominus$	$\bigcirc$	$\oplus$	$\oplus$	$\oplus$
	Niepotrzebna	$\oplus$	$\oplus$	$\bigcirc$	$\bigcirc$	$\ominus$

Z tab. 2 wynika, że na kierunku dydaktycznym "Mechanika" przeważają zadania o krótkim czasie obliczania (poz. 2), ale o zróżnicowanej strukturze, wymagające ingerencji w tok obliczeń (poz. 7), charakteryzujące się bardzo różnorodną liczbą danych wejścia i wyjścia, które wygodnie jest przedstawić w postaci graficznej (analogowej) (poz. 6).



Lp.	Cecha charakterystyczna zadania dla systemu komputerowego	Liczba przedmiotów, w których występują zadania o danej cenie	Przybliżona liczba obliczanych zadań rocznie	
			w semestrach nieparzystych	w semestrach parzystych
1	2	3	4	5
1.	Długi czas obliczania	10	150	70
2.	Krótki czas obliczania	19	550	400
3.	Duża liczba danych Wejścia/Wyjścia	13	350	200
4.	Mala liczba danych Wejścia/Wyjścia	13	420	220
5.	Pożądana cyfrowa postać Wejścia/Wyjścia	12	250	130
6.	Pożądana graficzna postać danych Wejścia/Wyjścia	17	500	270
7.	Pożądanym tryb konwersacyjny obliczania	17	650	400
8.	Tryb konwersacyjny niepotrzebny	4	100	40
9.	Pożądana symulacja komputerowa	17	350	200

Istnieje duża liczba zadań, w których zastosowanie symulacji komputerowej byłoby korzystne (poz. 9). Zwraca uwagę mała liczba zadań przetwarzanych wsadowo (poz. 1), w których tryb konwersacyjny nie jest potrzebny (poz. 8). Także zadań o cyfrowej postaci danych jest o około połowę mniej od zadań o pożądanej graficznej postaci danych wejścia i wyjścia.

Zestawione w tab. 2 charakterystyczne cechy zadań przeznaczonych do rozwiązania za pomocą systemu komputerowego posłużyły, na podstawie tab. 1, do określenia potrzeb sprzętowych i programowych dla laboratorium projektowania wspomagane komputerowo.

### 3.2. Udział środków elektronicznej techniki obliczeniowej w procesie projektowania

#### Zadania projektowe realizowane przez komputer

Udział i znaczenie środków ETO w procesie projektowania można rozpatrywać obiektowo, tzn. zależnie od projektowanego urządzenia lub problemowo, tj. zależnie od rodzaju zadania projektowego lub od rodzaju działań realizowanych za pomocą komputera.

Obiektowe ujmowanie procesu projektowania jest zgodne z tradycyjnym podziałem wiedzy inżynierskiej na specjalizacje według rodzajów urządzeń. Ponieważ procesy projektowania różnych urządzeń mogą się znacznie od siebie różnić, to takie podejście jest naturalne i uzasadnione. Inne mogą być wymagania stawiane systemowi komputerowemu przy projektowaniu dźwignio, a inne przy projektowaniu samolotów. Różnice te wynikają jednak z systemowych właściwości projektowanego obiektu, takich np. jak stopień złożoności, ustrukturalizowania, modułowość, liczba stopni swobody, sposób i zakres oddziaływania na otoczenie itp. Dlatego bardziej obiecujące dla usprawnienia projektowania jest wyodrębnienie tych jego cech, które są wspólne dla projektowania różnych klas obiektów technicznych. Prowadzi to do rozpatrywania problemów projektowania bardziej ze względu na systemowe, interdyscyplinarne cechy zadań projektowych niż ze względu na rodzaj obiektu fizycznego. Dzięki temu, mimo nieuchronnie pogłębiającej się specjalizacji przedmiotowej, pozostaje możliwe wzajemne zrozumienie między specjalistami z zakresu różnych dyscyplin technicznych.

#### Działania projektowe wykonywane na komputerze

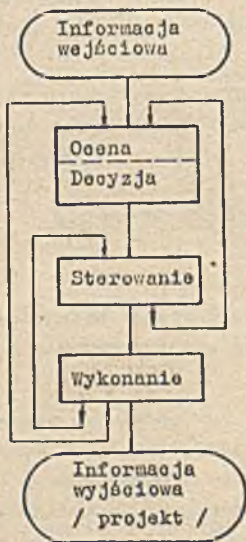
Ze względu na tym działań realizowanych w procesie projektowania programy komputerowe można podzielić na sprawdzające, interpretujące i optymalizujące.

• Programy sprawdzające służą do sprawdzania czy badana wielkość mieści się w dopuszczalnych granicach lub do określania czasu. Obliczenia te mogą być bardzo skomplikowane lub proste. Zarówno obliczenie stanu naprężeń metodą elementów skończonych, jak też przybliżone obliczenie dopuszczalnego momentu skręcającego wał o danej średnicy są przykładami obliczeń sprawdzających.

• Programy interpretujące służą do ustalenia rozmiarów zespołów i elementów odpowiednio do nałożonych na nie wymagań. Programy te zawierają elementy obliczeń sprawdzających. Działają one w ten sposób, że po zakończeniu obliczenia sprawdzającego zmienia się wartość wybranego parametru tak, żeby po powtórzeniu obliczeń sprawdzających otrzymać lepszy wynik. Cykl obliczeń kończy się, gdy nie udaje się już uzyskać poprawy, np. przy obliczaniu krytycznej prędkości kątowej wału zmiennym parametrem może być średnica wału lub rozstaw łożysk. Parametry te będą w toku iteracyjnych obliczeń tak zmieniane, aby osiągnąć wymaganą różnicę między prędkością krytyczną i prędkością roboczą. Programy interpretujące znajdują szerokie zastosowanie w projektowaniu wspomagany komputerowo.

• Programy optymalizujące mogą być uważane za wyższy stopień programów interpretujących. W odróżnieniu od nich, w programach optymalizujących poszukuje się takiej kombinacji wartości parametrów wpływowych, przy której wybrana wielkość (funkcja celu) osiągnie wartość optymalną.

Interesujące wnioski wypływają z cybernetycznej analizy procesu projektowania i z oceny stopnia automatyzacji projektowania według jakości działań wykonywanych przez komputer. W procesie projektowania można wyodrębnić iteracyjnie powiązane czynności oceniające i decyzyjne, sterujące i wykonawcze (rys. 1).



Rys. 1.

Rozpatrując proces projektowania z punktu widzenia podziału tych działań między człowieka i komputer można wyróżnić trzy fazy automatyzacji projektowania:

- zastosowanie komputera do wykonywania pracochłonnych i żmudnych obliczeń,
- zastosowanie komputera do wykonywania czynności sterujących,
- komputeryzacja niektórych czynności wartościująco-decyzyjnych.

Pierwszą fazę automatyzacji projektowania można podzielić na dwa etapy: prymitywny i właściwy. W etapie prymitywnym komputer wykonuje trudniejsze obliczenia inżynierskie w sposób naśladowujący tok obliczeń ręcznych. Przebieg procesu projektowania nie ulega zmianie, skraca się tylko czas obliczeń i zmniejsza się liczba błędów. W literaturze anglosaskiej ten sposób wykorzystania komputera nazywany jest trafnie: "komputer jako duży suwak".

Właściwy etap pierwszej fazy komputeryzacji polega na dostosowaniu procedur obliczeniowych do możliwości maszyny cyfrowej. W obliczeniach ręcznych korzysta się przeważnie z wzorów wyprowadzonych analitycznie, które podają wyniki w formie ostatecznej. Zaprogramowanie tych wzorów daje programy charakterystyczne dla prymitywnych zastosowań komputera. Ten sam wynik można jednak na ogół osiągnąć za pomocą prostych operacji obliczeniowych powtarzanych iteracyjnie. Ilość obliczeń rośnie, ale przy wykorzystaniu komputera nie ma to istotnego znaczenia. Iteracyjne algorytmy obliczeniowe, które są mało popularne w obliczeniach ręcznych, do obliczeń komputerowych są szczególnie przydatne. Znajdują one zastosowanie w wymienionych wcześniej programach interpretujących i optymalizujących.

Istotną cechą drugiego etapu pierwszej fazy automatyzacji projektowania jest także powstawanie złożonych programów, obejmujących kilka lub więcej programów elementarnych, które poprzednio były stosowane oddzielnie. Powoduje to zmniejszanie się liczby danych wejściowych do komputera, maleje także czas i nakład pracy na komunikację z maszyną cyfrową.

W pierwszej fazie komputeryzacji projektowania komputer nie wykracza poza rolę genialnego radmistrza, któremu powierza się do wykonania "czarną robotę" obliczeniową. Druga faza komputeryzacji wymaga daleko posuniętej formalizacji toku projektowania. Projektowanie przebiega według specjalnie opracowanych procedur (systemów algorytmów), które obejmują cały proces projektowania. Procedury projektowania zawierają złożone programy obliczeniowe opracowane i sprawdzone w fazie pierwszej.

Nowa, wyższa jakość drugiej fazy automatyzacji projektowania wynika przede wszystkim z faktu, że system komputerowy przejmuje od człowieka większość funkcji sterujących, pozostawiając mu ocenę wyników i decyzje o wyborze podstawowych danych i metod projektowania. W związku z tym projektant wprowadza do systemu komputerowego głównie decyzje logiczne, gdy w fazie pierwszej przeważały dane liczbowe. Projektant tworzy wspólnie z systemem komputerowym układ cybernetyczny, w którym rośnie znaczenie wzajemnego komunikowania się. Sprawne przekazywanie przez człowieka informacji komputerowi wymaga specjalnych języków konwersacyjnych i problemowo zorientowanych, natomiast do przedstawienia informacji wytworzonej przez komputer w formie dostosowanej do ludzkich możliwości percepcji potrzebne są specjalne urządzenia grafiki komputerowej, urządzenia reprodukcyjne, przesyłania informacji, itp.

Trzecia faza automatyzacji prac projektowych jest prawdopodobnie najwyższą możliwą obecnie do osiągnięcia formą projektowania. W tej fazie system komputerowy wykonuje wszystkie funkcje procesu projektowania, łącznie z oceniającymi i decyzyjnymi. Człowiekowi pozostaje: opracowanie danych wejściowych, kontrola procesu projektowania (np. za pomocą ekranu świetlnego) oraz sprawdzenie i zabranie dokumentacji projektowej. Niektóre decyzje, mające zasadnicze znaczenie dla projektu, również mogą być pozostawione człowiekowi.

Całkowita automatyzacja trudnych i złożonych zadań projektowych może okazać się niemożliwa lub nieopłacalna przy współczesnych środkach technicznych i istniejącym stanie wiedzy o projektowaniu. Stosunkowo najłatwiejsza jest automatyzacja projektowania obiektów dobrze ustrukturalizowanych, np. zbudowanych z elementów typowych.

Jest bardzo prawdopodobne, że trzecia faza automatyzacji projektowania nie jest granicą możliwości w zakresie komputerowego wspomaganie projektowania. Można przypuszczać, że w niedalekiej przyszłości:

- udostępnione będą komputery wykonujące czynności twórcze, które dotychczas przypisywano wyłącznie człowiekowi; otworzy to nowe możliwości przed systemami automatycznego projektowania;
- komputeryzacja opanuje dziedziny projektowania z pogranicza sztuki i techniki, np. projektowanie architektoniczne;
- metody poszukiwania rozwiązań i ich optymalizacja będą udoskonalone do tego stopnia, że dla poprawnie sformułowanego zadania projektowego będzie zapewnione znalezienie optymalnego rozwiązania;
- dalszy rozwój środków komunikacji całkowicie uniezależni wykorzystanie systemu komputerowego od odległości między miejscem pracy projektanta i ośrodkiem obliczeniowym;
- komputery przejmą rolę magazynu ludzkiej wiedzy;
- mikrofilm, taśma magnetyczna i inne nośniki informacji wyeliminują rysunki na papierze. Nowe środki reprograficzne umożliwią tanie i szybkie odtwarzanie dokumentacji projektowej w miejscu wykorzystania.

### 1.1. Strategia procesu automatyzacji projektowania

Opisane w punkcie "Działania projektowe wykonywane na komputerze" trzy fazy automatyzacji projektowania różnią się istotnie jakością działań wykonywanych przez system komputerowy. Warunkiem

przejęcia do wyższej fazy jest opanowanie fazy poprzedniej. Pobieżne potraktowanie lub pominięcie fazy niższej w zasadzie uniemożliwia dalszy rozwój automatyzacji. Postęp komputeryzacji projektowania warunkowany jest nie tylko sprzętem komputerowym i oprogramowaniem, lecz także wiedzą o procesie projektowania, świadomością projektantów i organizacją zespołów projektujących. Harmonijny rozwój tych elementów jest konieczny dla postępu projektowania.

Prace nad automatyzacją projektowania powinno się zaczynać od prostych zadań projektowych o przejrzystej strukturze, łatwych do sformalizowania. Automatyzacja złożonych zadań wymaga bardzo wnikliwej analizy procesu projektowania oraz dużego nakładu pracy i dlatego nie powinna być podejmowana w pierwszej kolejności. Opracowywanie systemów projektowania wspomaganego komputerem jest pracochłonne i kosztowne, a błędy i inne niepowodzenia są niemal nieuchronne. Zbyt duże trudności w opracowywaniu zadań wspomagania komputerowego mogą zniechęcić do kontynuowania prac w tym kierunku, toteż kwalifikacja zadań projektowych do automatyzacji powinna być bardzo wnikliwa.

#### 4. Podsumowanie

Efektywność wspomagania komputerowego powinna być oceniana ze względu na kryteria jakości procesu projektowo-konstrukcyjnego. Z drugiej strony, projektowanie jest integralną częścią procesu wytwarzania i ma w tym procesie szczególne znaczenie, ponieważ decyduje o jakości i kosztach produktu. Proces projektowania wiąże tylko małą część całkowitych kosztów wytwarzania, ale czas projektowania jest porównywalny z czasem reszty cyklu twórczego. Skrócenie tego czasu bez obniżenia jakości projektu jest niemożliwe bez implementacji nowych środków technicznych i metodycznych. Środki te, to systemowe ujęcie procesów projektowo-konstrukcyjnych i ich algorytmizacja oraz automatyzacja niektórych działań projektotwórczych. Systematyzacja i algorytmizacja muszą poprzedzać wprowadzenie komputerowego wspomagania projektowania.

Jakkolwiek realny proces projektowania jest bardzo silnie związany z fizycznymi cechami projektowanego urządzenia, to zadania projektowe dają się klasyfikować za pomocą pewnych interdyscyplinarnych cech, niekoniecznie związanych z fizyczną postacią obiektu. Takie ujęcie lepiej służy poprawie efektywności projektowania, m.in. dlatego, że każde usprawnienie może być zastosowane w wielu zadaniach projektowych.

Biorąc pod uwagę różne cechy procesów projektowania i zadań projektowych otrzymuje się różne klasyfikacje.

Podział zadań projektowych ze względu na stopień oryginalności projektowanego urządzenia ma znaczenie głównie porządkujące, podobnie jak podział programów na sprawdzające, interpretujące i optymalizujące.

Charakterystyka zadań projektowych ze względu na liczbę i rodzaj danych wprowadzanych do i wyprowadzanych z komputera oraz czas obliczeń i stopień ingerencji człowieka w proces obliczeniowy umożliwiają stworzenie relacji między typem zadań projektowych i wyposażeniem systemu komputerowego. Jest to jedno z podstawowych zagadnień, ponieważ środki elektronicznej techniki obliczeniowej powinny być dostosowane do wymagań procesu projektowania.

Na przykład w projektowaniu maszyn i urządzeń mechanicznych należy przede wszystkim usprawnić obliczenia wytrzymałościowe i symulowanie procesów dynamicznych, wprowadzić metody optymalizacji i zautomatyzować prace kreślarskie. Większość obliczeń konstrukcyjnych, to stosunkowo proste operacje, ale jest ich dużo i zawierają dużą liczbę danych, które przygotowuje konstruktor. Aby mógł on sprawnie przeprowadzić tego typu obliczenia, powinien mieć możliwość bezpośredniego dialogu z maszyną cyfrową przez dalekopis zainstalowany przy jego stanowisku pracy, dostępny bez formalności i oczekiwania. Często wystarczy dobry system minikomputerowy wyposażony w urządzenie analogowo-cyfrowe przekazania danych.

Miarą poziomu automatyzacji procesu projektowania jest rodzaj działań wykonywanych przez system komputerowy. Najniższy poziom odpowiada automatyzacji czynności wykonawczych, tj. obliczeniowych. Poziom wyższy, to powierzenie komputerowi również funkcji sterujących toku projektowania. Najwyższy możliwy obecnie do osiągnięcia poziom obejmuje całość procesu projektowania, łącznie z

wartościowaniem i oceną rozwiązań. Człowiekowi pozostaje przygotowanie informacji wejściowej, kontrola przebiegu projektowania oraz sprawdzenie dokumentacji projektu. Dalszy istotny postęp będzie prawdopodobnie wynikiem rozpowszechnienia komputerów zdolnych symulować czynności twórcze przypisywane dotychczas wyłącznie człowiekowi.

Rozwój projektowania wspomaganego komputerem wymaga nie tylko odpowiedniego sprzętu i oprogramowania, zależy on także od postępu wiedzy o projektowaniu i od doświadczenia projektantów. Postęp w tej dziedzinie dokonuje się ewolucyjnie. W celu przejścia do wyższej fazy automatyzacji projektowania konieczne jest solidne opanowanie fazy poprzedniej. Przystąpienie do komputeryzacji określonej klasy zadań projektowych powinno być poprzedzone systemową analizą warunków osiągnięcia celu a realizacja musi być konsekwentna i odporna na okresowe trudności. Jest to jeden z tych procesów, w których powodzenie można osiągnąć tylko systematyczną, żmudną pracą zespołu ludzi, ponieważ próby "przeskoczenia" niższych faz komputeryzacji przynoszą więcej strat niż korzyści.

#### Literatura

- [1] GASPARSKI W.: Z zagadnień metodologii projektowania inżynierskiego. Warszawa: Studium Doskonalenia Organizacji Badań Naukowych 1970
- [2] DIETRYCH J.: System i konstrukcja. Warszawa: WNT 1978
- [3] BRANKAMP K., WIENDAHL H.P.: Rechnergestütztes Konstruieren-Voraussetzungen und Möglichkeiten. Konstruktion 1971 nr 5
- [4] ROHATYŃSKI R. i in.: Algorytmizacja nowych metod projektowania układów hydraulicznych. Praca wykonana w Instytucie Konstrukcji i Eksploatacji Maszyn Politechniki Wrocławskiej. Wrocław 1975/76
- [5] ROHATYŃSKI R.: Systemowe ujęcie procesu projektowania jako podstawa wprowadzenia projektowania wspomaganego komputerem. Materiały I Konferencji: Metody i środki projektowania automatycznego, sekcja 2. Warszawa: Instytut Podstaw Budowy Maszyn Politechniki Warszawskiej 1977
- [6] NEVILL G.E., CROWE R.A.: Computer Augmented Conceptual Design. W: Basic Question of Design Theory. Amsterdam: North Holland Publ. Comp. 1974
- [7] ROHATYŃSKI R.: Koncepcja i założenia centralnego laboratorium projektowania wspomaganego komputerowo w mechanice. Wrocław: Instytut Konstrukcji i Eksploatacji Maszyn Politechniki Wrocławskiej 1977 (praca niepublikowana).

dr inż. Stanisław GRODZICKI  
mgr Janina OLECH  
Instytut Maszyn Matematycznych

## Wybrane aspekty zobrazowania wyników obliczeń komputerowych

### Wstęp

W wielu dziedzinach nauki i techniki, korzystających z obliczeń komputerowych, zachodzi potrzeba graficznego przedstawiania wyników obliczeń w postaci rysunków, wykresów, diagramów lub innych graficznych form prezentacji. Formy graficzne są bardziej przyswajalne i łatwiej je można interpretować, niż równoważne opisy liczbowe i formuły analityczne.

Automatyczne uzyskiwanie rysunków lub innych form graficznych dokonuje się za pomocą odpowiednich urządzeń peryferyjnych, którymi mogą być plottery, grafoskopy, leczy i drukarki. Aby to osiągnąć musi istnieć odpowiedni zestaw programów podstawowych i użytkowych. Do automatycznego rysowania najczęściej są używane plottery, rzadziej drukarki. Tych ostatnich nie należy oczywiście traktować jako formy zastępczej, przy braku klasycznych urządzeń graficznych, a raczej trzeba zdawać sobie sprawę, że w pewnych określonych zastosowaniach ważne jest dla użytkownika otrzymanie jednego łącznego dokumentu, na którym zestawienie liczbowe (lub nawet tekstowe) jest zilustrowane również rysunkowo (bądź wykreślnie).

Niniejsze opracowanie zawiera omówienie możliwości opracowywania oprogramowania realizującego wspomniane sytuacje różnorodnej prezentacji wyników obliczeń komputerowych w wybranych dziedzinach, ze zwróceniem uwagi na problemy związane z tworzeniem takiego oprogramowania. Zostaną również podane: krótka charakterystyka konkretnych przykładów zobrazowania wyników obliczeń komputerowych za pomocą plotterów, opis pewnego programu, który generuje na drukarce wykresy oraz przykłady programów, które ilustrują na drukarce lub monitorze ekranowym (alfanumerycznym) wprowadzaną konwersacyjnie dane do obliczeń, co pozwala użytkownikowi natychmiast ocenić ich merytoryczną poprawność. Ponadto przedstawia się ogólny przegląd urządzeń do automatycznego rysowania. Natomiast w tym opracowaniu nie omawia się mikroplotterów, urządzeń wyjścia mikrofilmowego COM (Computer Output Microfilm), a także grafoskopów z piórem świetlnym oraz urządzeń do przetwarzania danych z postaci graficznej na cyfrową (digitizory).

Wymaga wyjaśnienia jeszcze nazwa urządzenia do przetwarzania danych z postaci cyfrowej na graficzną. Dla urządzeń tych w krajach anglosaskich przyjęła się nazwa plotter, której będziemy używać zasadniczo w dalszej części opracowania w formie spolszczonej. W polskiej literaturze, jak również w różnych ośrodkach komputerowych w kraju, można spotkać także inne nazwy, jak graphplotter, koordynatograf automatyczny, maszyna autokreślarska, autokreślarka, automat kreślący, pisak xy lub czeska nazwa digigraf, a wreszcie samorys. Wszystkie te nazwy zasadniczo oznaczają tę samą klasę urządzeń. Odmienne natomiast znaczenie mają takie określenia, jak monitor graficzny lub końcówka graficzna. Urządzenia tego typu spełniają rolę wejścia i wyjścia, podczas gdy uprzednio wymienione (plotter i drukarka) są wyłącznie urządzeniami wyjściowymi.

### Ogólny przegląd urządzeń do automatycznego rysowania

Automatyczną prezentację postaci graficznej informacji wykonuje się za pomocą ploterów, które są urządzeniami sporządzającymi rysunki w postaci wykreślonej, wrytowanej lub wyświetlonej. Jakich do tego celu używa się narzędzi, będzie podane przy opracowaniu głowicy rysującej plotera [1],[2],[3].

Zazwyczaj wyróżnia się dwie podstawowe zasady działania ploterów.

- Kreślenie krokowo, które polega na rysowaniu dowolnych linii w wyniku wykonania elementarnych przesunięć (kroków) wzdłuż osi X oraz Y, w kierunku dodatnim lub ujemnym. Otrzymuje się w ten sposób przesunięcia w 4 kierunkach podstawowych oraz w 4 kierunkach wypadkowych. Kreślone linie są lamanymi utworzonymi z tych ośmiu kierunków. Długość elementarnego przesunięcia wynosi przeważnie 0,1 - 0,2 mm.

- Kroślenie ciągłe, które dokonuje się wraz ze zmieniającymi się w sposób ciągły szybkościami ruchu względem obydwu osi układu współrzędnych. Precyzyjne automaty działające na tej zasadzie są wyposażone w odpowiedni system pomiarowy, pozwalający określić różnice współrzędnych między położeniem osiągniętym przez element kroślicy i położeniem zadany numerycznie, co z kolei umożliwia ustalenie właściwych szybkości ruchu.

Obecnie produkowane plotery mogą działać jako urządzenia wyjściowe, pracujące na zasadzie bezpośredniego podłączenia do komputera (system on-line) lub pośredniego (system off-line) oraz w systemie zdalnego sterowania. Te trzy podstawowe systemy pracy plotera będą pokrótce scharakteryzowane.

- W systemie bezpośredniego połączenia dane wejściowe są przetwarzane przez komputer, który w sposób ciągły steruje czynnościami plotera. System ten przy zastosowaniu szybkich komputerów nie daje możliwości ich pełnego wykorzystania, ze względu na różnice w szybkości wykonywania operacji komputera i kroślenia plotera.

- W systemie off-line ploter nie jest bezpośrednio połączony z komputerem. Dane wejściowe są przetwarzane przez komputer według odpowiedniego programu, a wyniki "do rysowania" są zapisywane na maszynowych nośnikach informacji (taśma magnetyczna, taśma perforowana). Wyniki te są później wprowadzane do jednostki czytającej plotera, który pracuje według programów standardowych.

- W systemie zdalnym informacja jest przesyłana na odległość przez łącze telekomunikacyjne za pomocą modemów nadawczych i odbiorczych. Ten system przetwarzania informacji może zawierać elementy systemu bezpośredniego lub pośredniego, w zależności od tego, jak wykorzystywana jest przesyłana informacja.

Wartość użytkowa systemu automatycznie kroślicy zależy od wielu różnorodnych czynników, wśród których szybkość maksymalna nie należy do najważniejszych. Istotne są np. osiągane przyspieszenia, określające roboczą szybkość przy krośleniu linii krzywych lub krótkich odcinków linii prostych. Bardzo ważne jest podstawowo i użytkowo oprogramowanie systemu, rodzaje stosowanych głowic i narzędzi, a także wyposażenie w urządzenia peryferyjne np. w urządzenia wejściowe.

Urządzenia do przetwarzania danych z postaci cyfrowej na graficzną można podzielić na 4 podstawowe grupy [2],[1]:

- plotery bębnowe, • mikroplotery,
- plotery płaszczyznowe, zwane także stołowymi, • grafoskopy.

Jak już wspomnieliśmy we wstępie, w tym opracowaniu nie zajmujemy się dwoma ostatnimi grupami urządzeń.

Plotery bębnowe (drum plotter) są urządzeniami dość prostymi, kroślicy na powierzchni papieru przewijającego się wokół bębna. Kroślenie odbywa się krokowo przez pisak przesuwany wzdłuż listwy równoległej do osi bębna. Ruch w drugim, prostokątnym kierunku realizuje się za pomocą obrotu bębna z papierem. Plotery te znalazły duże zastosowanie m.in. w projektowaniu inżynierskim [4],[1].

Plotery bębnowe charakteryzują się następującymi podstawowymi parametrami technicznymi:

- wielkość jednostkowego przemieszczenia (kroku); w ploterach wysokiej precyzji wielkość najmniejszego kroku może wynosić 1,25  $\mu\text{m}$ , a w ploterze o niskiej precyzji największe jednostkowe przemieszczenie może wynosić 625  $\mu\text{m}$  [8]; najczęściej jednak spotyka się plotery, które mają jednostkowo przemieszczenie w granicach 0,1 - 0,05  $\mu\text{m}$ ,
- liczba kroków na sekundę, np. 300 lub maksymalną szybkość rysowania w  $\text{cm/s}$ ,
- szerokość papieru, na którym kroślono są rysunki, np. 78, 74  $\text{cm}$ ,
- liczba wektorów (kierunków rysowania) możliwych do wykorzystania w procesie kroślenia, zazwyczaj 8 lub też 16, gdy rozróżniane są połówki kroków.

Plotery bębnowe mają ograniczoną szerokość rysunku, zależną od szerokości bębna i wynoszącą ok. 500  $\text{mm}$  do 1500  $\text{mm}$ , natomiast długość rysunku może być bardzo duża. Np. długość papieru podkładowego (teoretyczna długość rysunku) w ploterze bębnowym francuskiej firmy HENSON wynosi 100  $\text{m}$ .

W tab. 1 podano przykłady charakterystyk bębnowych ploterów CALCOMP 563 i BENSON 121 [3],[5].

Tab. 1: Przykłady charakterystyk ploterów bębnowych

CALCOMP 563	Charakterystyki	BENSON 121
1219	Wymiary zewnętrzne: długość	1150
533	szerość	400
1168	wysokość	350
101.15	Ciężar	80
67.06	Długość rysunku (długość papieru)	100
78.74	Szerokość rysunku	74
1	Liczba pisaków	1
0,1	Wielkość jednostkowego przemieszczenia (kroku)	0,1
3	Maksymalna szybkość rysowania: wzdłuż osi	9
-	wzdłuż przekątnej	12,7

Plotery płaszczyznowo (flatbed plotter), o powierzchni rysunkowej płaskiej, są urządzeniami sterowanymi automatycznie w sposób cyfrowy lub rzadziej analogowy, przeznaczonymi do sporządzania rysunków na płytach stołów. Istnieje duża różnorodność ploterów płaszczyznowych, począwszy od małych ploterów sterowanych krokowo i używanych jako urządzenia peryferyjne kalkulatorów, a kończąc na dużych systemach automatycznie kreślących, sterowanych przez minikomputery.

Plotery płaszczyznowe wyróżniają się następującymi podstawowymi parametrami technicznymi [1],[2]:

- powierzchnia robocza stołu - jest bardzo zróżnicowana, począwszy od małych ploterów stołowych, o wymiarach 50 x 50 cm, przez najczęściej spotykane o wymiarach ok. 1000 x 1500 mm, aż do znacznych wymiarów, np. norweska firma KONGSBERG produkuje m.in. plotery o wymiarach 2600 x 3700 mm, 1800 x 7500 mm i 1600 x 9000 mm,
- dokładność,
- maksymalna szybkość,
- maksymalne przyspieszenie.

Jednym z najbardziej istotnych parametrów plotera jest jego dokładność. Wyróżnia się tu najczęściej wymieniane przez wytwórców dokładności:

- dokładność wewnętrzna przyrzędu (resolution) - jest to wielkość jednostkowego przesunięcia głowicy kreślącej w jednym z dopuszczalnych kierunków podstawowych, np. może to być wielkość 0,005 mm
- dokładność kreślenia (accuracy) - jest statystycznym błędem dokładności kreślenia, uwzględniającym błędy interpolacji liniowej, gubienie sygnałów sterujących itp., np. może ona wynosić  $\pm 0,100$  mm,
- dokładność powtórzeniowa (repeatability) - określa dokładność powtórzenia, tzn. dokładność z jaką głowica kreśląca powróci do punktu początkowego krzywej zamkniętej, np. może to być  $\pm 0,08$  mm.

Użytkownik zainteresowany jest najczęściej dokładnością kreślenia plotera. Dokładność ta waha się w granicach od  $\pm 0,100$  do  $\pm 0,005$  mm. Pozostałe dokładności są dla poszczególnych przyrządów kilkakrotnie wyższe od dokładności kreślenia. Z ekonomicznego punktu widzenia duże znaczenie ma szybkość kreślenia. Pozostaje ona w pewnym związku z dokładnością, która obniża się przy szybkim kreśleniu. Szybkość ta waha się w granicach od 25 mm/s do 500 mm/s. Szybkość przeciętna ploterów wysokiej klasy waha się w granicach od 100 mm/s do 200 mm/s.

Zapewnienie takich parametrów technicznych można uzyskać przez odpowiednie rozwiązania kons-



trukeyjne elementów plotera. Jednym z ważniejszych elementów konstrukcyjnych jest głowica plotera.

Głowicą plotera nazywa się urządzenie przesuwane względem osi układu współrzędnych i przeznaczone bezpośrednio do kreślenia, rytowania lub naświetlania. Głowicami mogą być [1]:

- Jedno- lub wielonarzędziowe uchwyty piór kulkowych i specjalnych pisaków; uchwyty wielonarzędziowe umożliwiają programowane kreślenie linii w kilku kolorach lub w kilku różnych grubościach,
- uchwyty mikroskopów (do ustawiania głowicy na wybrane punkty rysunku), zerowników, igieł itp.,
- uchwyty ryłców grawerskich; ryłco jest obracany w trakcie grawerowania tak, aby płaszczyzna ostrza była zawsze prostopadła do grawerowanej linii, co umożliwia uzyskiwanie linii o stałej grubości,
- uchwyty noży do nacinania folii "out and pill",
- głowica naświetlająca materiał światłoczuły; głowico to mogą naświetlać linie różnej grubości lub wybrane programowo znaki konwencjonalne (przezroczza tych znaków są umieszczane na obwodzie tarczy obracanej w sposób sterowany tak, aby właściwy znak był wyświetlony).

#### Przebieg oprogramowania

Śśród wielu firm produkujących urządzenia do automatycznego przetwarzania danych cyfrowych na graficznie, których plotory znalazły szersze zastosowanie w Polsce, można wymienić: BENSON (Francja), CALCOMP (Stany Zjednoczone), KONGSBERG (Norwegia), CONTRAVES AG (Szwajcaria). Ponadto niektóre biura projektowe posiadają minikomputery WANG 2200, w których urządzeniami wyjściowymi są również plotory.

Wraz ze sprzętem dostarczane jest również odpowiednie oprogramowanie podstawowe (bazowe) urządzeń rysujących. Niektóre firmy oferują także oprogramowanie użytkowe. Ale zazwyczaj to ostatnio tworzą poszczególni użytkownicy ploterów, zależnie od dziedziny zainteresowań i własnych potrzeb.

Właściwy zestaw oprogramowania podstawowego i użytkowego wraz z odpowiednim sprzętem komputerowym stanowi podstawę efektywnego wykorzystania urządzeń automatycznego przetwarzania danych cyfrowo-graficznych.

Należy zauważyć, że oprogramowanie ploterów cechuje modułowość, łatwość użycia oraz użytkowanie niezależnie od typu komputera. Na przykład firma BENSON powyższe wymagania w zakresie oprogramowania osiąga przez wprowadzenie trzech poziomów oprogramowania. I tak:

- poziom 1 obejmuje oprogramowanie podstawowe, na ogół napisane w języku ASSEMBLER, do rozwiązywania takich problemów jak: interpolacja liniowa, skalowanie, pisanie znaków i cyfr, wybór wymiaru jednostki i wybór pisaka, identyfikacja współrzędnych itp.,
- poziom 2 oprogramowania dotyczy podprogramów w języku FORTRAN przeznaczonych do wykreślenia osi, linii kreskowanych i kropkowanych, symboli przestrzennie centrycznych, linii o różnych grubościach, kół, łuków itp.,
- poziom 3 oprogramowania zawiera wszystkie inne podprogramy, odpowiadające specyficznym wymaganiom użytkownika i stanowią one przeważnie jego własność.

W celu łatwiejszego rozpoznawania funkcji i nazw podprogramów ploterowych w tab.2 dokonano zestawienia oprogramowania podstawowego, biorąc pod uwagę firmy BENSON i CALCOMP. Jak łatwo zauważyć, w zestawieniu tym niektóre podprogramy BENSON-a mają podwójne nazwy. Są to moduły, które spełniają te same funkcje, lecz używane są w różnych układach współrzędnych: absolutnym i użytkownika. Przemieszczenia pisaka mogą być wyrażane w centymetrach, lub jednostkach odwołanych dowolnie przez użytkownika. Jeżeli przemieszczenia są wyrażane w centymetrach, to jest to przestrzeń absolutna, której zdefiniowanie jest związane z wymiarami fizycznymi plotera. Natomiast przestrzeń użytkownika to taka, w której przemieszczenia są wyrażane w jednostkach specyficznych dla opracowywanego problemu. Przestrzeń użytkownika jest związana z przestrzenią absolutną przez skalę. W tym samym programie rysującym można stosować podprogramy odpowiadające dwóm układom współrzędnych: pewne części rysunku mogą być wykreślane w funkcji skali, inne w jednostkach

absolutnych (contymetrach). Poza tym pierwsze cztery podprogramy, przedstawiono w cytowanej tabeli, używa się w układzie absolutnym, natomiast podprogramy, począwszy od Lp. 18 w dół, stosuje się w układzie użytkownika.

Jak widać z tablicy 2, oprogramowanie podstawowe firmy CALCOMP jest trochę obszerniejsze od oprogramowania firmy BENSON. Jednym z najważniejszych modułów jest tu podprogram HG PLOT, który oprócz podstawowych funkcji, jak inicjowanie i zakończenie pracy plotera, może spełniać wiele istotnych zadań związanych z kreśleniem. Za pomocą tego właśnie podprogramu można rysować dowolne linie łamane, dowolne figury symetryczne i nieregularno oraz dowolne krzywe nie zdefiniowane odpowiednimi podprogramami podstawowego oprogramowania. W wielu zagadnieniach, w których wykorzystuje się np. łuki koła lub inne krzywe, łatwiej jest zdefiniować te krzywe jako zbiory kolejno po sobie następujących punktów leżących bardzo blisko siebie, których współrzędne nietrudno określić w przyjętym układzie współrzędnych. Kreślenie takich krzywych przy użyciu podprogramu HG PLOT jest proste i dokładne. Natomiast stosując do rysowania łuku koła odpowiedni podprogram HG PCIRCLE, można mieć kłopoty z określeniem odpowiednich parametrów dla tego modułu, jak promieniu początku i końca krzywej, właściwych kątów nachylenia tych promieni. Takie np. trudności mogą występować przy rysowaniu określonego zespołu różnych krzywych lub zespołu łuków kołowych o zmieniających się promieniach. Połączenie ciągle takich krzywych jest bardzo trudne. Niektóre problemy oprogramowania użytkowego będą przedstawione na konkretnym przykładzie, dotyczącym opracowywania modułów kreślących w oprogramowaniu projektowo-inżynierskim.

Ponadto należy podkreślić, że do wykorzystywania podprogramu HG PLOT użytkownik podaje wszystkie punkty aproksymujące daną krzywą, czyli może użyć do tego celu dowolnego algorytmu aproksymacji, a w szczególności bardziej dokładnego niż te, które zastosowano w innych podprogramach, jak np. w HG PCIRCLE. Należy zauważyć, że procedura HG PLOT jest wykorzystywana przez większość podprogramów, która przedstawiono w tablicy 2. Również algorytm cytowanego podprogramu HG PCIRCLE wykorzystuje ten podprogram. Powną niedogodnością w stosowaniu procedury HG PLOT może być duża liczba przekazywanych współrzędnych punktów, ale tylko dla dużych rysunków, co zwiększa zajętość pamięci programu użytkowego.

Tab. 2. Zestawienie oprogramowania podstawowego (standardowego)

Lp	Funkcja podprogramu	Nazwa podprogramu	
		CALCOMP	BENSON
1	2	3	4
1	Rozpoczęcie i otwarcie zbioru wyjściowego (inicjowanie plotera)	HG PLOT	IDENA
2	Wyznaczenie początku układu współrzędnych. Zamknięcie zbioru wyjściowego (zakończenie rysunku), wytworzenie kodów przesunięć pióra	HG PLOT	PNUMA
3	Wybór skali i początku układu użytkownika	HG PLOT	ECHEL
4	Wybór pióra o danym numerze	-	*) PLUMA
5	Kreślenie prostej lub przemieszczenie podniesionego pióra (łączenie 2 punktów)	HG PLOT HG PDASHLN	TRAA-TRAS
6	Określenie aktualnej pozycji pióra względem początku układu współrzędnych	HG PWHERE	POSA-POSS
7	Rysowanie ciągu znaków alfanumerycznych	HG PSYNBL	PCARA-PCARS
8	Rysowanie liczby zmiennoprzecinkowej w postaci stałej lub zmiennoprzecinkowej	HG PNUMBER	NOMIA-NOMIS
9	Zamiana współrzędnych	-	CVSA, CVSD, CVAS
10	Zamknięcie zbioru wyjściowego	HG PLOT	FTN
11	Przemieszczenie i wykreślenie symbolu (znaku) wokół podanego jego środka	-	DECENA-DECENS
12	Łączenie punktów linią przerywaną i poszerzoną	HG PDASHLN	BETIA-BETIRS
13	Łączenie punktów linią poszerzoną	-	BETEA-BETEPS
14	Wykreślenie okręgu lub łuku okręgu	HG PCIRCLE	BCERCA-BCERCS

1	2	3	4
15	Wykreślenie okręgu lub łuku okręgu linią przerywaną i poszerzoną	-	BCERTA-DCERTS
16	Wykreślenie okręgu lub łuku okręgu poszerzoną linią	-	BCERPA-BCERP5
17	Wykreślenie elipsy lub łuku elipsy	HGPELLIPSE	BLIPSA-BLIPSS
18	Wykreślenie elipsy lub łuku elipsy linią przerywaną i poszerzoną	-	BLIPTA-BLIPTS
19	Wykreślenie elipsy lub łuku elipsy linią poszerzoną	-	BLIPPA-BLIPPS
20	Wykreślenie osi z podziałką liniową	HGPAXIS	BSAXLL
21	Wykreślenie osi z podziałką logarytmiczną dziesiętną	HGPLOAXIS	BSAXLD
22	Wykreślenie zespołu punktów i/lub połączenie ich dowolną linią	HGPLINE HGPDASHLN	BSCURV
23	Wykreślenie zespołu punktów (określonych symboli bez łączenia ich liniami)	HGPSYMBL	BSNUAG
24	Wykreślenie krzywej $Y=F(X)$	HGPSCURVE	BFONXY
25	Wykreślenie półsiatki wg podziałki liniowej	HGPLOT	BSGRLL
26	Wykreślenie półsiatki wg podziałki logarytmicznej	HGPAASHLN	BSGRLD
27	Wykreślenie prostokąta	HGPRECT	-
28	Wykreślenie osi o zadanej podziałce z opisem podziałki w postaci liczby całkowitej lub zmienne-przecinkowej	HGPAXISV	-
29	Optymalizacja układu wartości na wykresie z podziałką	HGPSCALE	-
30	Optymalizacja układu wartości na wykresie w skali logarytmicznej	HGPLOGSCALE	-
31	Wykreślenie krzywej zadanej równaniem $Y = AX^i + BX^j + CX^k + DX^l + E$	HGPXCURVE	-
32	Wykreślenie krzywej zadanej równaniem $X = AY^i + BY^j + CY^k + DY^l + E$	HGPYCURVE	-
33	Wykreślenie odbicia krzywej zadanej równaniem $Y = AX^i + BX^j + CX^k + DX^l + E$ względem prostej o równaniu $Y = GX + H$	HGPXREFL	-
34	Zakładanie zbioru danych na taśmę magnetycznej dla programu # XJGA przy pracy pośredniej	HGPTAPE	-
35	Kopiowanie z taśmy magnetycznej na ploter	# XJGA	-
36	Wykreślenie izometrii i rzutów perspektywicznych	# X2G1	-

\* Kreska pozioma oznacza brak odpowiedniego podprogramu dla opisywanej jego funkcji

#### Przykłady programowych modułów graficznych

##### Użytkowe moduły wyjścia na ploter

Charakteryzowany pokrótce poniżej program ploterowy jest jednym z wielu opracowanych w języku FORTRAN z wykorzystaniem plotera CALCOMP 563, podłączonego w systemie on-line do mc ODRA 1304. Programy te, a właściwie podprogramy ploterowe, gdyż stanowią one jednostki składowe programów, dotyczą projektowania geometrii krzywoliniowych odcinków tras, składających się z różnych zespołów krzywych.

Omawiany podprogram obejmuje zespół krzywych, które tworzą łuk kołowy z dwoma symetrycznymi parabolami stopnia trzeciego.

Rysowanie takiego zespołu krzywych odbywa się najłatwiej za pomocą podprogramu HGPLOT, dla którego generuje się zbiór punktów o znanych współrzędnych  $x, y$ . Taki zbiór punktów przechowywany jest w odpowiednich tablicach. Należy zauważyć, że współrzędne  $x, y$  tego zbioru punktów obli-

oza się w jednym układzie współrzędnych dla krzywych zdefiniowanych przez różne równania. Przyjęcie jednolitego układu współrzędnych dla różnych krzywych zdecydowanie upraszcza pisanie programu, a jednocześnie powoduje zachowanie ciągłości krzywych i dokładności ich rysowania. Tym bardziej jest to godno podkreślenia, że zbiory współrzędnych  $x, y$  można przechowywać w tych samych tablicach, które rezerwowano dla innych zbiorów, we wspólnym obszarze PAO.

Istotną czynnością przy pisaniu podprogramu ploterowego jest przeskalowywanie obliczonych wielkości, które są wykorzystane do rysowania, na układ jednostek ploterowych. Ogólna zasada powinna być taka, że najpierw należy wykonywać wszelkie obliczenia, a następnie je przeskalowywać na układ współrzędnych plotera. W omawianym przykładzie liczone współrzędne były wyrażone w jednostkach metrycznych. Aby przejść na układ jednostek centymetrowych należy współrzędne wyrażone w jednostkach metrycznych pomnożyć przez współczynnik skali, który tworzy się w następujący sposób:

$$WS = S \cdot 100 \text{ cm,}$$

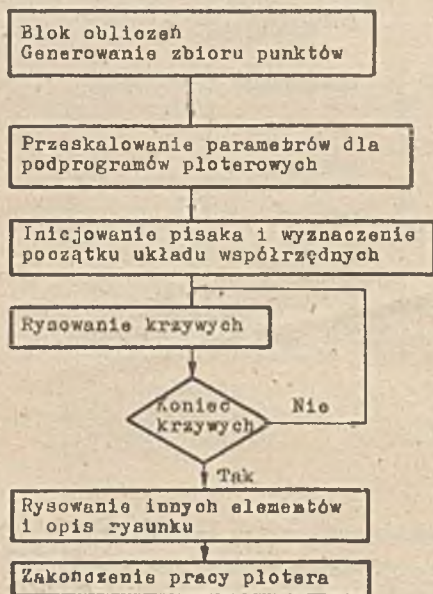
gdzie: WS - współczynnik skali

S - skala rysunku wyrażona w ułamku dziesiętnym.

Np. dla skali rysunku 1:1000 współczynnik skali  $WS = 0,001 \cdot 100 = 0,1$ .

Przy rysowaniu linii łamanych trzeba wziąć pod uwagę fakt, że ze względu na właściwości wywoływanych odpowiednich podprogramów ploterowych, mogą powstać w punktach załamania niedokładności rysowania (niewielkie przerwy). Powodem tego może być nieodpowiednia korelacja ruchów urządzenia i wysuwu pisaka do dolnego położenia. Aby tego uniknąć przed rysowaniem kolejnego odcinka łamanej za pomocą podprogramu HGPLOT należy dać rozkaz opuszczenia pisaka.

Ogólny schemat blokowy charakteryzowanego podprogramu ploterowego przedstawia poniższy rysunek 1.



Rys. 1. Ogólny schemat działania programu ploterowego

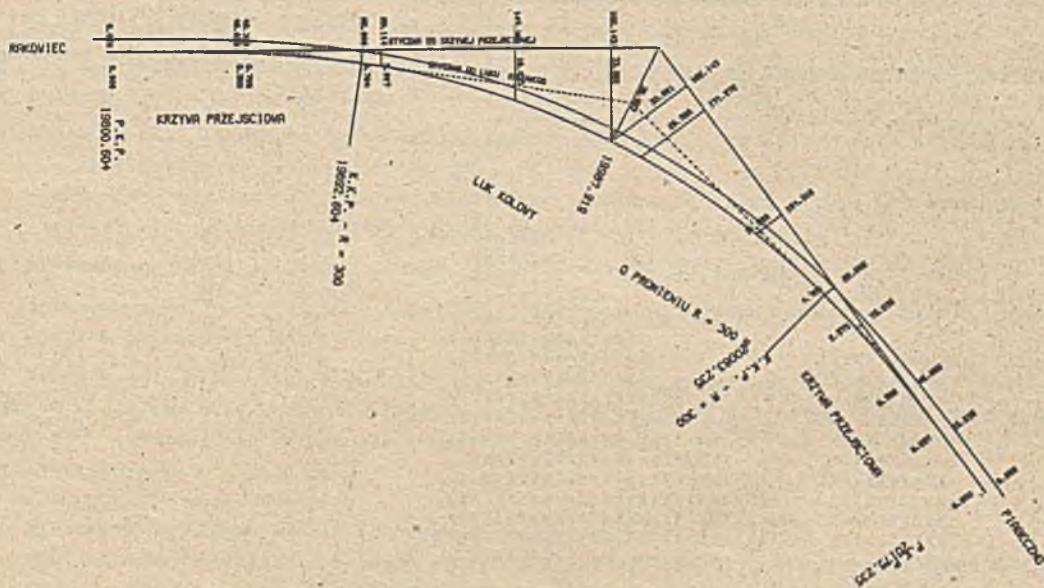
Przykład wizualizacji danych na monitorze alfanumerycznym

Rysunek 2 przedstawia efekt przetwarzania danych liczbowych na postać graficzną. Dotyczy on zagadnienia projektowania geometrii osi dwóch torów kolejowych z uwzględnieniem poszerzenia skrajni budowli, jaką trzeba wykonać na długości krzywych przejściowych (parabola sześcienna) i łuku kółowego. Elementy graficzne i liczbowe tego rysunku powstają w wyniku wykonania podprogramu ploterowego oraz wykorzystania niektórych wielkości liczbowych, obliczonych w innych podprogramach. Np. przedstawione na rysunku wielkości współrzędnych do tyczenia w terenie tego odcinka krzywoliniowego przekazano z innego podprogramu. Podobnie wartości pikietażu punktów głównych tego zespołu krzywych obliczono w innym podprogramie, a następnie przekazano do podprogramu ploterowego. Wszystkie to wielkości odnoszą się do osi toru wewnętrznego. Opis rysunku, jak również pozostałe napisy alfanumeryczne były wprowadzone jako dane do podprogramu ploterowego.

W pewnych typach obliczeń inżynierskich, np. w obliczeniach wytrzymałościowych, ważnym momentem jest bezpośrednia i prawidłowa kontrola wprowadzonych danych początkowych do programu. Szybka i obrazowa kontrola tych danych można uzyskać na monitorze ekranowym w trybie konwersacyjnym. Jest to tzw. wizualizacja danych wejściowych. Polega ona na przedstawieniu, np. projektowanych

# LUK KOŁOWY Z KRZYWAMI PRZEJSCIOWYMI Z UWZGLĘDNIENIEM POSZERZENIA SKRAJNI LINIA KOLEJOWA W-WA - RADOM KM. 20.0

SKALA 1:1000



Rys.2. Przykład rysunku wykonanego przez ploter

obciążeń belki, w postaci schematycznego rysunku w odpowiedniej skali, z którego łatwo można się zorientować o wielkości i prawidłowości projektowanych obciążeń.

Na rysunku 3 podano przykład wizualizacji danych wejściowych na monitorze ekranowym, wykonanej programem w systemie konwersyjnym\*. Przykład pokazany na rys. 3 przedstawia belkę obciążoną siłami równomiernie rozłożonymi i obciążeniami trójkątnymi rosnącymi. Skala takiego rysunku jest determinowana maksymalnym obciążeniem.

Należy zauważyć, że na takim rysunku bardzo łatwo i szybko można wychwycić pewne nieprawidłowości. Na przykład, w sytuacji nieodpowiedniego umiejscowienia lub niewłaściwej wielkości obciążenia, obydwa rodzaje obciążeń mogłyby zachodzić na siebie. Oczywiście można powiedzieć, że wprowadzane wielkości początkowe do obliczeń są również wyświetlane na ekranie w postaci numerycznej.

\* Przykład ten jest zaczerpnięty z prac mgr inż. E. Dobrzyńskiej, która opracowała wiele programów w systemie konwersyjnym z zakresu statyki budowli.

CANE DO OBLICZEN  
 DLUGOSC BELKI = 500.00 (CM)  
 WSPOLCZYNNIK SPREZYSTOSCI = 2000000.00 (KG/CM2)  
 MOMENT BEZWLADNOSCI = 24010.00(CM4)

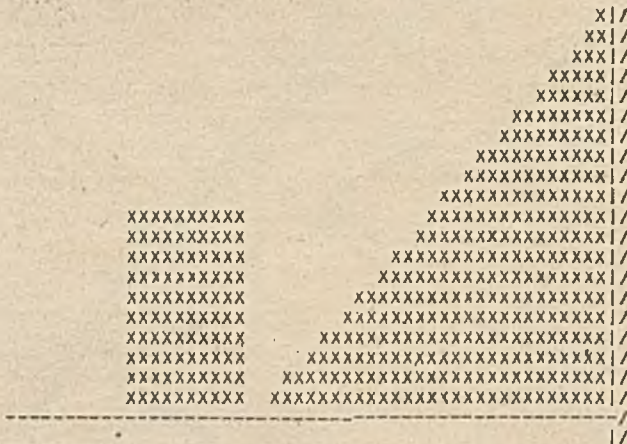
OBCIAZENIA ROWNOMIERNIE ROZLOZONE

Q (KG/CM)	LQ (CM)	A (CM)
30.00	400.00	100.00

OBCIAZENIA TRÓJKATNE ROSNACE

QR (KG/CM)	LQR (CM)	B (CM)
60.00	300.00	300.00

- Q - obciążenie równomiernie rozłożone,
- LQ - odległość początku obciążenia równomiernego od utwierdzonego końca belki,
- A - długość odcinka, na którym występuje obciążenie równomiernie rozłożone,
- QR - obciążenie trójkątne rosnące,
- LQR - odległość początku obciążenia trójkątnego rosnącego od utwierdzonego końca belki,
- B - długość odcinka, na którym występuje obciążenie równomiernie rozłożone.



Rys. 3. Przykład wykresu wykonanego na monitorze graficznym

Tym niemniej poprawność tych danych nie jest oczywista, gdy natomiast postać graficzna tych danych jest bardzo komunikatywna i pozwala od razu zorientować się w błędnych danych.

Reasumując trzeba podkreślić, że kontrola danych wejściowych do programów komputerowych jest zawsze problemem ważnym i skomplikowanym. Dlatego, jeśli jest choćby mała możliwość tej kontroli, należy ją wykonywać.

Wykorzystanie drukarki i dynamicznego formatu wydruku do graficznej prezentacji wyników

Jeszcze jednym, godnym uwagi przykładem graficznej prezentacji wyników obliczeń w niektórych zagadnieniach gospodarczych jest program KONTUR, który wykorzystując dynamiczny format wydruku przedstawia, oprócz zestawień liczbowych, wyniki w postaci wykresu. Jest to szczególnie cenne w takich dziedzinach, gdzie informacja graficzna jest bardzo czytelna, komunikatywna, a nie jest przy tym wymagana zbyt duża dokładność graficznego przedstawienia wyników.

Omówiony poniżej program KONTUR jest przykładem ilustrującym wykorzystanie drukarki do zobrazowania danych wynikowych. Zadaniem, dla którego program opracowano, było przedstawienie na wykresie tendencji rozwoju pewnych zjawisk gospodarczych w perspektywnym modelu gospodarczym. Twórcy programu zależalo, aby przedstawione dane wynikowe byly czytelne, dawaly latwy w interpretacji obraz zmian w zalozonym przedziale czasu, a przy kilku (lub kilkunastu) wariantowych wyliczeniach pozwalaly na szybka analize porownawcza wplywu pewnych czynnikow na kierunki rozwoju okreslonych elementow modelu.

Mając na uwadze powyższe zadanie dla programu przyjęto następujące założenia:

- określono przedział czasowy (IS), dla którego może być sporządzony wykres; przyjęto, że  $5 \leq IS \leq 20$ ;
- wyprowadzane wyniki zobrazowane będą wykresem w I ćwiartce układu współrzędnych; na osiach współrzędnych drukowane będą:
  - na osi rzędnych - wartości skali podziału w odstępach co 10 przesuwów drukarki,
  - na osi odciętych - podział na okresy (IS) co 4 odstęp, a pod osią wypisane wartości tych okresów;
- graficzne odwzorowanie danych (wykres) połączone będzie z tekstową informacją o ilustrowanych danych oraz z wydrukiem poszczególnych zestawów danych liczbowych;
- na jednym wykresie może być zobrazowanych do 5 zestawów danych (ID); współrzędne wykresu zaznaczone będą cyfrą odpowiadającą numerowi zestawu danych liczbowych (z przedziału 1 - 5);
- aby wykres był czytelny i łatwy do wykorzystania, należy go znormalizować; przyjęto, że wydruk zależeć będzie od wartości IS:
  - dla  $IS \leq 10$  będzie to jedna strona papieru z drukarki, na której połowę prawą zajmie wykres, a lewą opis z danymi liczbowymi;
  - dla  $IS > 10$  będą to dwie strony papieru z drukarki. Jedna strona zawierać będzie część opisową oraz liczbowe zestawy danych, druga strona sam wykres;
- wykres może być drukowany z liniami siatki podziału lub bez linii siatki podziału;
- opracowany program powinien pozwolić na otrzymywanie dowolnej liczby wykresów, a wyprowadzanie wyników powinno być jak najmniej czasochłonne.

Wszystkie te założenia są uwzględnione w opracowanym programie KONTUR. Mając ustaloną postać danych wyjściowych, określono jakiego rodzaju informacje wejściowe obowiązany jest podać użytkownik, aby zapewniły one otrzymanie określonego wydruku.

Dane użytkownika

- Ile wykresów chce otrzymać.
- Ile zestawów danych będzie drukowanych na jednym wykresie.
- Dla jakiego przedziału czasowego sporządzany będzie wykres (IS - wartość graniczna) oraz wartości jednostkowo tego przedziału (jeśli będą to lata, a wartość IS = 5, to np. 81, 82, 83, 84, 85).
- Czy wydruk wykresu ma być drukowany z siatką podziału czy bez siatki.
- Słowny opis danych.
- Dane liczbowe dla kolejnych zestawów danych.

Dane te mogą być wprowadzone z kart perforowanych, czytane z dysku, bądź przekazywane w odpowiedniej formie z innego programu.

Wprowadzane dane bądź pełnią w programie rolę danych sterujących, bądź są wpisywane w odpowiednią tabelkę, przetwarzane i przygotowywane do wydruku.

Program korzysta z następujących tabeli zasadniczych:

- OPIS - zawiera słowny opis danych obrazowanych na wykresie,
- ATA - zawiera jednostkowe dane liczbowe okresów,
- DANE - zawiera dane liczbowe poszczególnych zestawów,
- TAB - na podstawie wyliczeń i porównań dokonuje przygotowania wykresu,
- A i B - tablice, w których w zależności od wartości IS przygotowuje się dane do wydruku; zawartość tablicy mieści w sobie dane przenoszone z uprzednio wymienionych tabeli oraz z tabeli roboczych - pomocniczych,
- FA i FB - zawiera podstawowy format wydruku tablicy A lub B,
- FZ - zawiera wszystkie możliwe zmiany w formatach w zależności od wartości danych sterujących IS i ID.

Ponadto program korzysta z tabeli pomocniczych, które nie będą tu omawiane.

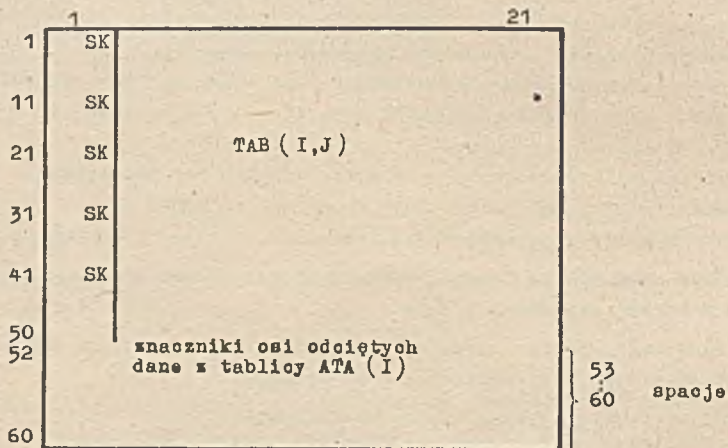
Po wczytaniu danych i rozmieszczeniu ich we właściwych tablicach, program bada wartości liczbowe w zestawach danych i opierając się na tych badaniach wylicza skalę podziału na osi rzędnych.

Przygotowanie wykresu wymaga przebadania całej tablicy TAB, porównania danych w poszczególnych zestawach ze skalą podziału i wstawienie w odpowiednie miejsce tablicy tzw. znacznika wykresu, tzn. cyfry zgodnej z numerem zestawu danych liczbowych.

W wyniku przeprowadzonych operacji otrzymuje się dane wyjściowe dla użytkownika. Wyprowadzenie ich mogłoby być realizowane bezpośrednio z tablic TEKST, ATA, DANE i TAB przez odpowiednio zaprojektowane formaty. Wymagałoby to jednak wielu zatrzymań przesuwu drukarki oraz - w trakcie drukowania - wybierania danych do wydruku z różnych tablic. Powodowałoby to zwolnienie tempa wyprowadzania danych i jednocześnie wymagałoby bardzo częstej komunikacji systemu operacyjnego z urządzeniami pamięciowymi (dysk, taśma), co również wpływałoby na wydłużenie czasu wyprowadzania wyników. Aby temu zapobiec przyjęto, że w specjalnych tablicach A lub B (w zależności od wartości IS), kompletuje się dane w takiej postaci, aby właściwie dobrany format wydruku pozwalał na wyprowadzenie na drukarkę całej tablicy jednym rozkazem. Drugą, niezwykle istotną sprawą przemawiającą za zastosowaniem skompletowanej tablicy do wydruku było to, że przy tak przygotowanych danych wejściowych możliwe jest zastosowanie dynamicznego formatu wydruku, co szerzej będzie omówione w dalszej części tego opracowania, przy omawianiu przygotowania do wydruku tablicy B.

Zgodnie z założeniami rozmiary wydruku (na 1 lub 2 stronach) zależą od wartości IS. Załączone wydruki 1-5 ilustrują zróżnicowaną postać wyjściową danych: pod względem liczby zestawów danych, wielkości zestawów oraz wydruku wykresu z siatką podziału i bez niej.

Z prezentowanych wydruków wynika, że w wypadku tablicy A (gdy  $IS > 10$ ) skompletowanie tablicy do wydruku obejmuje tylko sam wykres. Przygotowanie tablicy wymaga wpisania w odpowiednie miejsca skali podziału osi rzędnych, samej osi odciętych, przeniesienia danych z tablicy TAB i z tablicy ATA. Dane z pozostałych tablic zasadniczych (OPIS, DANE) są wyprowadzane odrębnymi rozkazami. W tym wypadku dynamiczny format wydruku obejmuje wyprowadzanie wyników tylko z tablicy DANE.



Rys. 4. Ogólny obraz rozmieszczenia danych do wydruku w tablicy A (21,60)

Nieco inaczej przedstawia się przygotowanie danych do wydruku w tablicy B. Zgodnie z założeniami wszystkie dane mieszczą się na 1 stronie wydruku, stąd w tablicy B muszą być umieszczone dane opisowe, wartości zestawów danych, wykres i wszystkie dane uzupełniające ten wykres. Tak więc przygotowanie tablicy B (dla  $IS \leq 10$ ) jest w znacznie większym stopniu skomplikowane niż tablicy A. Dlatego w dalszej części dynamiczny format wydruku omówimy na przykładzie tablicy B.

Rysunek 5 pokazuje podział tablicy B na obszary rezerwowane dla danych określonego rodzaju.



1	1	2	...	16
2	OPIS /dane tekstowe przenoszone z tablicy OPIS/			
6				
7	6	7	8	16
	SK			
10	nr zestawów danych			
12	SK			
	Liczbowe wartości zestawów danych /przenoszone z tablicy DANE/			
21				
27	SK	WYKRES /dane przeniesione z tablicy TAB(I,J) /		
37	SK			
47	SK			
57				
58	Dane znakowe- wpisanie osi odc.			
	Przeniesione dane z tablicy ATA(I)			
60				

Rys. 5. Ogólny obraz rozmieszczenia danych do wydruku w tablicy B (16,60)

Z rysunku tego wynika, że w tablicy B znajdują się dane tekstowe, liczbowe i znakowe rozmieszczone w różnych miejscach tablicy. Rozmieszczenie to daje równocześnie wyobrażenie o postaci dokumentu na wydruku. Jednocześnie wskazuje na stopień skomplikowania wyprowadzania na drukarkę tak różnorodnych danych. Przygotowanie tablicy B ma na celu możliwość wyprowadzania na drukarkę tych wszystkich różnorodnych danych jednym rozkazem.

Gdyby program przygotowywany był dla jednego wykresu, lub kilku (czy dowolnej ilości), ale o stałej liczbie zestawów danych dla wykresu i stałym okresie, dla którego sporządzany jest wykres (np. 8), sprawa byłaby prosta i nie zasługiwałaby na uwagę. Jeden, odpowiednio przygotowany, format wyprowadzania wyników zapewniłby prawidłowy ich obraz. Zmienność danych (liczby wykresów, liczby zestawów w wykresie, zmienny okres) przekrośla możliwość zastosowania jednego formatu wydruku. W tej sytuacji zastosowano w programie dynamiczny format wydruku danych wynikowych. Działanie takiego formatu, zgodnie z tym, co powiedziano wcześniej, omówiono będzie na przykładzie tablicy B, przy założeniu zmienności ID. Przyjęcie zmienności jednej danej sterującej jest wystarczające dla pokazania metody przygotowania dynamicznego formatu wydruku.

Za podstawowy format wydruku przyjęto sytuację, w której  $ID = 5$  a  $IS = 10$ , czyli dla tablicy B maksymalno wielkości tych zmiennych sterujących. Fragment formatu przedstawia się następu-

jąco: ( 1H1 // T3ø, 16A4,5 (/T6,16A4)/2x,5A4,44X,F1ø.ø,2H1|/, 1øA4,2 ( 16A4,52X,2H1|,1øA4)/2X,5  
( 7X,F2.ø,3X),1A4,1øX,2H1|,1øA4...)

Ustalony format wydruku wprowadzono do tablicy FB. Umieszczenie formatu w tablicy FB może być dokonane dwoma sposobami: albo wczytane z kart jako dane wejściowe, albo w programie wprowadzone rozkazem DATA jako dane tekstowe.

Pierwszy sposób wprowadzania formatu do tablicy jest mniej pracochłonny, a więc wygodniejszy. Jednak bez względu na sposób wprowadzania istota zagadnienia sprowadza się do podziału całego formatu na elementy zapisywane w komórkach tablicy.

Zgodnie z zasadami języka FORTRAN dla maszyn IBM 370/145 i R-32, na których omawiany program był uruchamiany, standardowa długość danych zmiennych zapisywanych w jednej komórce wynosi 4. Zatem w jednym elemencie tablicy FB zapisywane są 4 znaki z ciągu znaków formatu.

Pamiętając o tej zasadzie, dzielimy format na elementy tablicy FB i w ten sposób wiemy jaka jest zawartość poszczególnych jej elementów. Np. w opisywanym wypadku wcześniej przedstawiony fragment formatu w tablicy FB zapisany jest w sposób następujący:

1	2	3	4	5	6	7	8	9	10	11	12
(1H1	// T3	ø,16	A4,5	(/T6	,16A	4)/2	X,5A	4,44	X,F1	ø.ø,	2H1
13	14	15	16	17	18	19	20	21	22	23	24
,1øA	4,2(	/6A4	,52X	,2H1	,1ø	A4)/	2X,5	(7X,	F2.ø	,3X)	,1A4
25	26	27	28	...	...						
,1øX	,2H1	,1ø	A4/2	....							

Przeprowadzamy następnie analizę zmienności formatu wydruku przy zmieniającym się ID. Ustalamy, jakie elementy tablicy FB i w jaki sposób ulegną zmianie, np. przy ID = 4 w przedstawionym wyżej fragmencie formatu, zmianie ulegną zapisane w tablicy FB elementy 20, 24 i 25. Rodzaj zmian ilustruje poniższa tablica zawierająca fragment takiej analizy.

Tabl. FB	numer elementu	20	24	25
	zawartość elementu	2X,5	,1A4	,1øX
Tabl. FZ	zawartość elementu	2X,4	,2A4	,18X
	numer elementu	1	5	9

Po zakończeniu analizy, czyli po ustaleniu wszystkich możliwych zmian w formacie w zależności od zmiany określonej zmiennej, ustala się tablicę zmian. W omawianym wypadku jest to tablica FZ, do której rozkazem DATA wprowadzono przewidziano zmiany. Fragment tej tablicy wygląda następująco:

Tabl. FZ(27)

1	2	3	4	5	6	7	8
2X,4	2X,3	2X,2	2X,1	,2A4	,3A4	,4A4	,5A4
9	10	11	12	13	14	15	16 ...27*
,18X	,26X	,34X	,42X	X,4F	X,5F	X,2F,	X,1F ...

Ustalenie zawartości tablicy zmian pozwala na zamianę elementów tablicy FB na elementy tablicy FZ czyli na realizację dynamicznego formatu wyprowadzania danych.

Poniżej przedstawiono fragmenty programu, gdzie tworzy się właściwe formaty wydruków.

\* Górne cyfry oznaczają kolejne komórki tablicy FB

I tak, jeżeli  $ID = 1$ , to

FB(20) = FZ(4)  
 FB(24) = FZ(8)  
 FB(25) = FZ(12)

⋮

a jeżeli  $ID = 4$ , to

FB(20) = FZ(1)  
 FB(24) = FZ(5)  
 FB(25) = FZ(9)

⋮

Należy podkreślić, że zastosowanie dynamicznego formatu wydruku wymaga:

- niezwykle dokładnej analizy podstawowej formy danych wyjściowych i wszelkich możliwych zmian zależnych od zmienności danych wejściowych,
- na tej podstawie zaplanowania wszystkich możliwych zmian w podstawowym formacie wyprowadzania danych,
- opracowania zawartości tablicy zmian (wszystkich jej elementów), i wreszcie
- zaprogramowania takich ścieżek logicznych, które realizowałyby wprowadzanie poszczególnych zmian.

Omówione wyżej i fragmentarycznie pokazane przygotowanie dynamicznego formatu wydruku wskazuje, że wymaga to nie tylko dokładnej i szczegółowej analizy sytuacji, ale wskazuje także na pracochłonność przygotowania programu, pozwalającego na otrzymanie żądanych wyników. Daje to jednak określone korzyści. Bowiem tak przygotowany format pozwala użytkownikowi na uzyskanie wielu różnorodnych zobrażeń danych wyjściowych, nie wymaga żadnych dodatkowych danych wejściowych poza tymi, o których była mowa na początku opisu omawianego programu. Szczególnie przy bardzo złożonych wydrukach, zastosowanie takiego formatu zwalnia użytkownika od obowiązku opracowywania bardzo wielu formatów dla danych wyjściowych. Ponadto znacznie skraca czas potrzebny na wyprowadzenie danych.

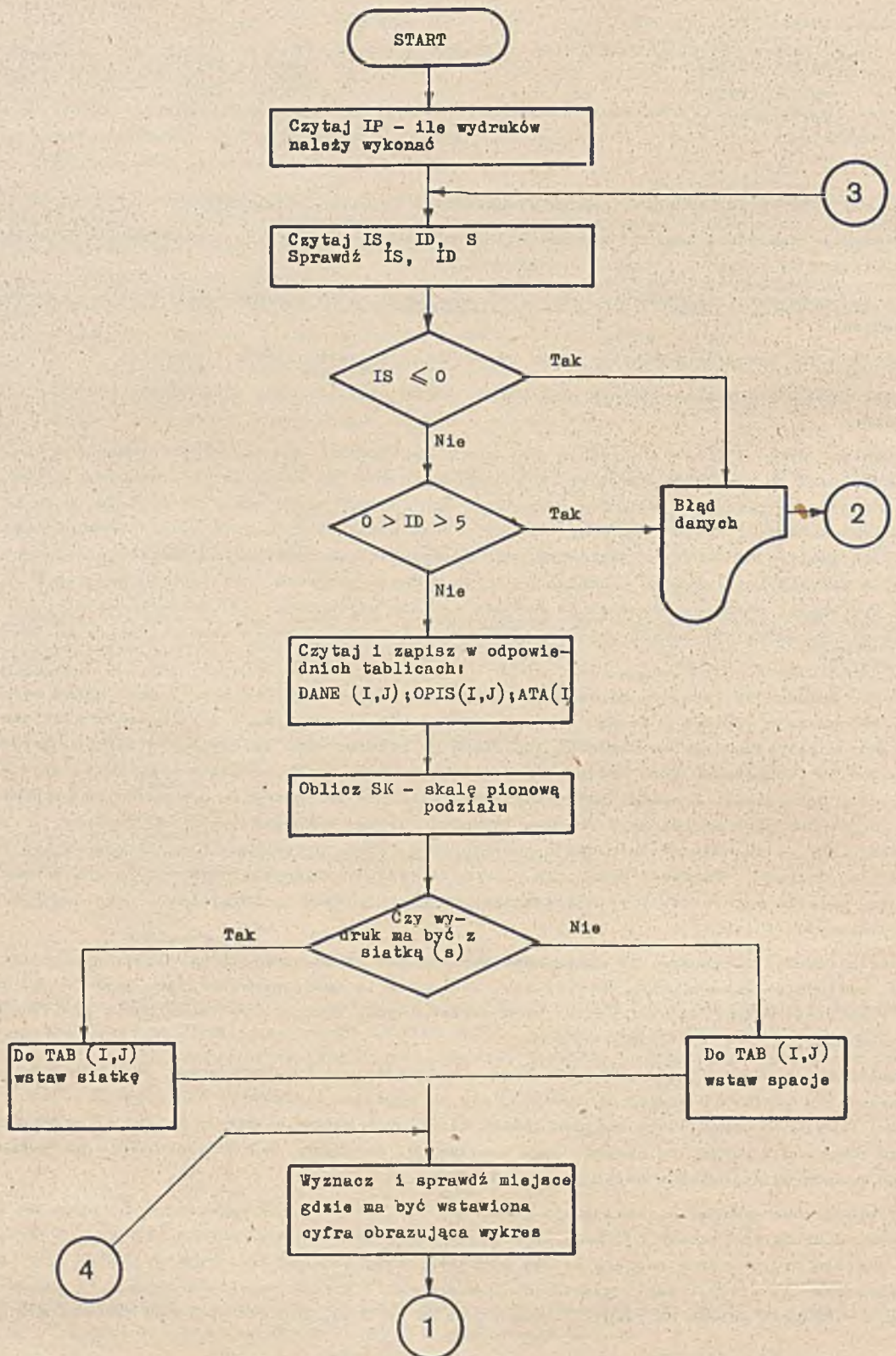
Mając świadomość tego, że zarówno drukarka, jak i język FORTRAN nie są najlepszym narzędziem do wyprowadzania wyników w formie wykresu, wydaje się, że również i takiej możliwości nie należy pomijać. Zapotrzebowanie na wizualną odpowiedź na pytanie może występować w różnych dziedzinach. Omawiane tu rozwiązanie może być przede wszystkim wykorzystane wszędzie tam, gdzie nie jest wymagana duża dokładność, a niezmiernie cenne jest uzyskanie informacji o prawidłowych (lub nieprawidłowych) tendencjach określonego procesu łącznie, zarówno w formie tekstowej (cyfrowej), jak i wykresowej. Np. w zagadnieniach rozwoju gospodarczego (tzn. w zadaniu, dla którego zajęto się opracowaniem programu), kontroli wykonania produkcji, stopnia wykorzystania maszyn, porównania kosztów czy zużycia materiałów przy wielowariantowych projektach inżynierskich, jako jeden z fragmentów dokumentacji itp.

Warto również zaznaczyć, iż dokładność wykresu zależy od rozpiętości między maksymalną a minimalną wartością odwzorowywaną na wykresie. Właściwością omawianego programu jest to, że im mniejsza rozpiętość między ww. wartościami, tym dokładniejszy będzie uzyskiwany obraz graficzny badanego zjawiska.

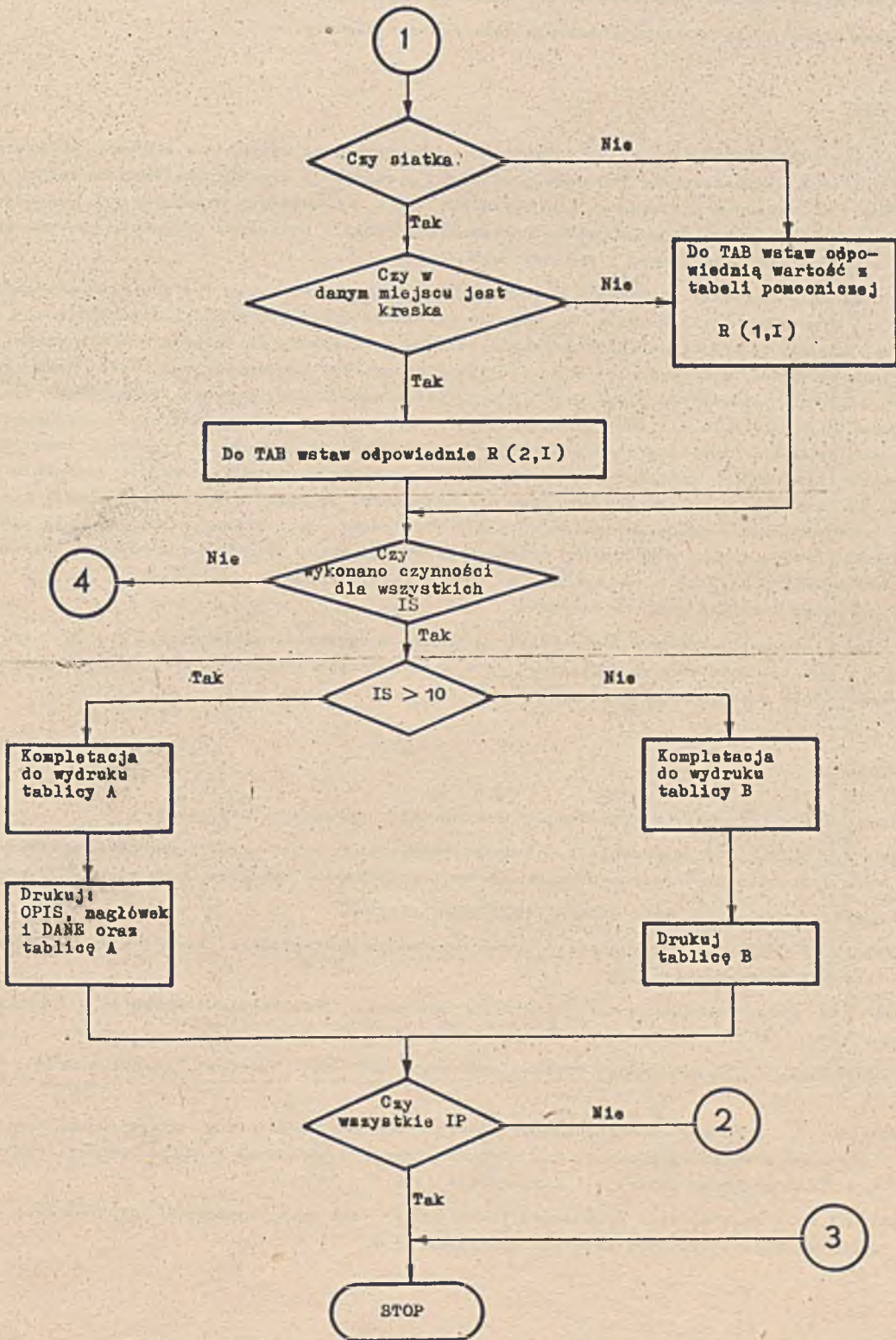
Omówiony problem zilustrowano wydrukami. Dane wejściowe (zarówno opisy danych liczbowych, jak i wartości w zestawach danych liczbowych) nie są wynikiem wyliczeń z konkretnego modelu rzeczywistego. Przyjęcie określonego zestawu danych liczbowych miało na celu sprawdzenie poprawności działania programu i pokazanie efektów jego realizacji. Załączone wydruki pokazują, jak zmienia się obraz danych wyjściowych w zależności od wartości IS i ID.

Pierwsze dwa wydruki realizują sytuację, gdy  $IS > 10$ , czyli wyprowadzają wydruk na 2 strony papieru z drukarki (tabela A), przy czym  $IS = 20$  w pierwszym wydruku i  $IS = 15$  w drugim wydruku. Ponadto drugi wydruk zawiera siatkę podziału. Trzy kolejne realizują sytuację, gdy  $IS \leq 10$  (konkretnie  $IS = 10$ ), czyli wyprowadzają wydruk na 1 stronę papieru z drukarki (tabela B). Różnica w wydrukach polega na różnych wartościach ID (4,3,5), ponadto wydruk 4 zawiera siatkę podziału.

Nie wdając się w merytoryczną interpretację samych wykresów należy stwierdzić, że drukowanie siatki podziału daje mniej czytelny obraz graficzny. Szczególnie wówczas, gdy wykres obrazowany



Rys.6a. Schemat blokowy programu KONTUR



Rys. 6a. Dalszy ciąg schematu blokowego programu KONTUR

jest znakiem = 1 linia wykresu jest prawie niewidoczna. Stąd wniosek, że linię podziału należy drukować tylko wówczas, gdy jest to bezwzględnie konieczne.

Logiczną organizację programu ilustruje jego schemat blokowy.

### Zakończenie

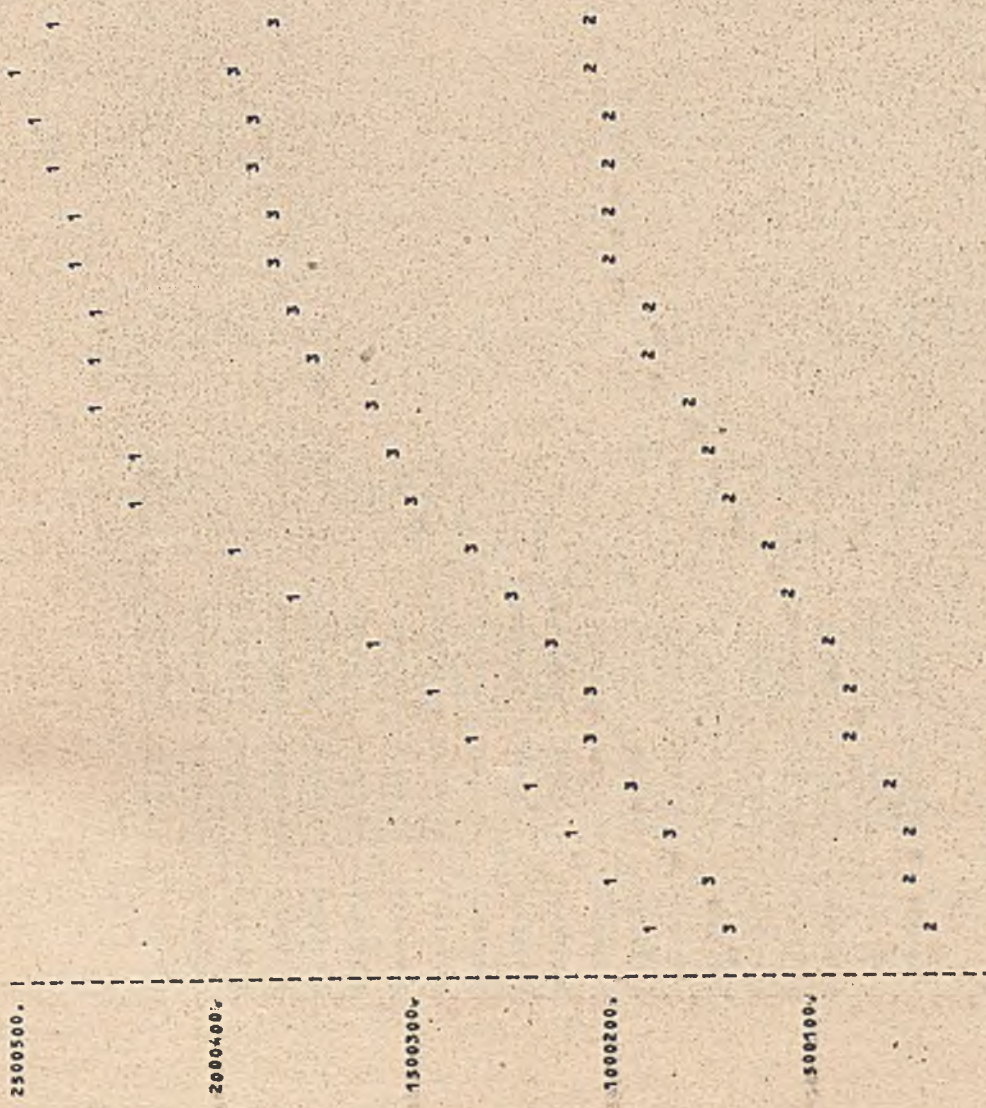
W artykule przedstawiono niektóre zagadnienia wybranych prac własnych z zakresu zobrazowania wyników obliczeń komputerowych. Problemy graficznej prezentacji wyników obliczeń są bardzo istotne zarówno w dziedzinach projektowo-inżynierskich, jak i w niektórych zagadnieniach gospodarczych. Omówiono przykłady trzech form graficznej prezentacji wyników obliczeń. Dotyczyły one wykorzystania plotera, drukarki wierszowej i monitora ekranowego.

Zazwyczaj dotychczasowe rozważania o grafice komputerowej kojarzyły się z monitorami graficznymi, czyli z grafiką interaktywną. Nie negując ogromnego znaczenia grafiki interaktywnej (grafika czynna) w pracach projektowo-inżynierskich, trzeba stwierdzić, że pasywna grafika komputerowa (stosowanie ploterów) może być etapem przejściowym do grafiki konwersacyjnej i jako taka pozwala także na zdobycie doświadczeń niezbędnych przy grafice monitorowej czynnej. Przykładami takich doświadczeń mogą być m.in.: zasady prawidłowego rastrowania rysunku na monitorze, metody stosowania różnych grubości linii i przoswitów między nimi, zmienność skalowania rysunku i inne. Należy także podkreślić, że tak grafika interaktywna, jak i pasywna spełniają określone zadania w projektowaniu inżynierskim w zależności od etapu projektowania. Grafika interaktywna pozwala inżynierowi na przeprowadzenie wielu wariantów obliczeń i na podstawie uzyskanych bezpośrednio wyników, wybór tych, które mogą być przedmiotem rozważań przed podjęciem ostatecznych decyzji. Natomiast w końcowym etapie projektu, kiedy konieczne jest uzyskanie dokumentacji projektu, grafika interaktywna nie może zastąpić grafiki pasywnej.

Tak więc stosowanie obu rodzajów grafiki komputerowej przynosi wiele korzyści i jest wprost niezbędne w pewnych dziedzinach zastosowań, a w szczególności w obiektowych systemach komputerowej automatyzacji przemysłu maszynowego.

### Literatura

- [1] GAŹDZICKI J.: Informatyka w geodezji i kartografii. Warszawa: PPWK, 1975
- [2] ECKES K., HYNTER R.: Urządzenia i metody automatycznego przetwarzania danych graficzno-cyfrowych w procesach geodezyjnych systemów informatycznych. Przegląd Geodezyjny 1973 nr 8-9
- [3] Prospekty i materiały firm: BENSON, KONGSBERG, CALCOMP.
- [4] MAZUR Z., WĘGRZYN M.: Programowanie z wykorzystaniem graf-plotera. Wrocław: Wydawnictwo Politechniki Wrocławskiej 1975
- [5] PISAK XY. Oprogramowanie maszyn cyfrowych ODRA serii 1300. Wrocław: ELWRO 1972 Publikacja nr 13009
- [6] FORTRAN IV-IDIM. Programowanie systemu cyfrowego ODRA 1304. Warszawa: Instytut Dróg i Mostów PW 1975
- [7] GAŹDZICKI J.: Problemy numerycznego opracowania map wielkoskalowych. Sympozjum na temat "Metody numerycznego opracowania map wielkoskalowych". Warszawa: Komitet Geodezji PAN i Centrum Informatyczne Geodezji i Kartografii 1976
- [8] KAMIŃSKI A. i in.: Analiza parametrów technicznych oraz cech użytkowych autokreślarek płaskich. Warszawa: IMM 1977 Archiwum Opracowań nr 4



71. 720 75. 740 750 760 770 78. 79. 800 81. 820 83. 840 850 86. 870 880 890 900

## DANE PORÓWNAWCZE WYNIKAJĄCE Z WYLICZEN Z MODELU

1, DOCHÓD NARODOWY.  
2, INWESTYCJE  
3, KONSUMPCJA

	1	2	3		
	904160,0	207840,0	678720,0	0,0	0,0
	937360,0	233590,0	748580,0	0,0	0,0
	1099100,0	272390,0	828180,0	0,0	0,0
	1286200,0	32210,0	933780,0	0,0	0,0
	1327600,0	378270,0	1039600,0	0,0	0,0
	1485600,0	486950,0	1038700,0	0,0	0,0
	1614700,0	482290,0	1132400,0	0,0	0,0
	1722300,0	530800,0	1231500,0	0,0	0,0
	1970900,0	611530,0	1359300,0	0,0	0,0
	2181900,0	706520,0	1475400,0	0,0	0,0
	2211050,0	741320,0	1530180,0	0,0	0,0
	2280720,0	806200,0	1620100,0	0,0	0,0
	2307380,0	890800,0	1730900,0	0,0	0,0
	2309900,0	911380,0	1810800,0	0,0	0,0
	2330390,0	980500,0	1850950,0	0,0	0,0
	2370400,0	990800,0	1861050,0	0,0	0,0
	2405290,0	1011000,0	1889900,0	0,0	0,0
	2433000,0	1021200,0	1902500,0	0,0	0,0
	2500500,0	1030500,0	1940800,0	0,0	0,0
	2408900,0	1050800,0	1855090,0	0,0	0,0



DANE PORÓWNAUCZE WYNIKAJĄCE Z WYLICZEN Z MODELU

1% DOCHÓD NARODOWY  
 2% INWESTYCJE  
 3% KONSUMPCJA

	1	2	3		
	90416000	20784000	67872000	0,20	0,20
	99736000	23339000	74858000	0,20	0,20
	109910000	27239000	82818000	0,20	0,20
	120620000	32211000	93378000	0,20	0,20
	132740000	37827000	103960000	0,20	0,20
	144360000	40690000	105870000	0,20	0,20
	161670000	46229000	115260000	0,20	0,20
	178230000	53080000	125130000	0,20	0,20
	197090000	61130000	133930000	0,20	0,20
	218190000	70632000	147540000	0,20	0,20
	221105000	76132000	153018000	0,20	0,20
	228072000	80620000	162010000	0,20	0,20
	230738000	89080000	173090000	0,20	0,20
	230990000	91138000	181080000	0,20	0,20
	233039000	98050000	185095000	0,20	0,20



1. СЧЕТЫ НАПОСЧН
2. СРОК ИМБЕИЦЫДНЕ
3. КИСИЛУЦИД
4. ЗАТРУДНЕННЕ

ДАНЕ ПЕРИОДИЧНЕ ЛУЧШАЦЕ З ЛУЦЕН З МДЕЛЛУ

	1.	2.	3.	4.
564350.0	20140.0	676720.0	321800.0	
591300.0	225590.0	748580.0	250600.0	
1099100.0	272590.0	328180.0	391250.0	
226200.0	322110.0	535790.0	370200.0	
1227600.0	378270.0	2059600.0	286120.0	
1428600.0	408590.0	1089700.0	388100.0	
4614700.0	462300.0	2152400.0	400000.0	
1782300.0	520800.0	1281500.0	410500.0	
370500.0	611550.0	1339300.0	415900.0	
218300.0	708520.0	2475400.0	430700.0	

2181900.  
1745520.  
1305140.  
872760.  
436380.



71, 72, 73, 74, 75, 76, 77, 78, 79, 80.

DANE WYNIKOWE PRODUKCJI  
 1. CELEPOD NARODOWY BRUTTO  
 2. ŚRODKI DLA INWESTYCJI PRZEMYSŁOWYCH  
 3. KONTUMPCJA INDYWIDUALNA

2181900.																				
1745520.																				
1309140.																				
872760.																				
436380.																				

1.	2.	3.
504140.0	207840.0	678720.0
557180.0	235590.0	746580.0
665100.0	272390.0	328180.0
1406280.0	222110.0	933780.0
427660.0	376270.0	-039600.0
1465600.0	406590.0	1358700.0
1414700.0	462290.0	1432400.0
1570580.0	611500.0	-359300.0
1782380.0	520800.0	1251500.0
2181560.0	706520.0	-175400.0

71, 72, 73, 74, 75, 76, 77, 78, 79, 80.



mgr inż. Jerzy MOCALA

Instytut Maszyn Matematycznych

Przegląd komputerowych metod projektowania linii elektroenergetycznych

Wprowadzenie

• Przedmiot projektowany - linia elektroenergetyczna

Linia elektroenergetyczna zgodnie z PN-67/E-05100 [1] jest to: "urządzenie napowietrzne, przeznaczone do przesyłania energii elektrycznej, składające się z przewodów, izolatorów, konstrukcji wsporczych i osprzętu". Ze względu na funkcje, jakie spełnia, linia stanowi fragment pewnego układu sieci, którego charakterystykę określa koncepcja sieci obejmująca dany obszar energetyczny. Dlatego też w zadaniu projektowym są już określone punkty początkowe i końcowe linii, przybliżony jej przebieg oraz parametry elektryczne na podstawie analizy technicznej i ekonomicznej układu sieci.

Zatem w procesie projektowania linii elektroenergetycznej linię rozważa się jako konstrukcję, która musi spełniać warunki wytrzymałościowe i gabarytowe. Informacja o linii zawarta w dokumentacji projektowej pozwala traktować linię następująco:

• Definicja 1

Linia elektroenergetyczna jako przedmiot projektowany jest to

$$P_{pr} = \langle (x_1, x_2, \dots, x_m), (T_1, T_2, \dots, T_m), (O_1, O_2, \dots, O_{m-1}), (P_1, P_2, \dots, P_{m-1}), (N_1, N_2, \dots, N_{m-1}) \rangle$$

gdzie:  $x_i$  - współrzędna słupa  $i$   $i = 1, \dots, m$   
 $T_i$  - typ słupa  $i$   $i = 1, \dots, m$   
 $O_j$  - stopień obostrzenia przęsła  $j$   $j = 1, \dots, m-1$   
 $P_j$  - typ przewodu przęsła  $j$   $j = 1, \dots, m-1$   
 $N_j$  - napięcie w przewodzie przęsła  $j$   $j = 1, \dots, m-1$   
 (przęsło - odcinek linii między sąsiednimi słupami [1]).

Cel projektowania - proces projektowania

Proces projektowania, w klasycznym ujęciu [1] jest działaniem doprowadzającym do otrzymania rozwiązania projektowego na podstawie zadania projektowego.

Zadanie projektowe można określić w następujący sposób [3]:

$$Z_{pr} = \langle S_k, \{P_i\}, \{\hat{P}_i\} \rangle$$

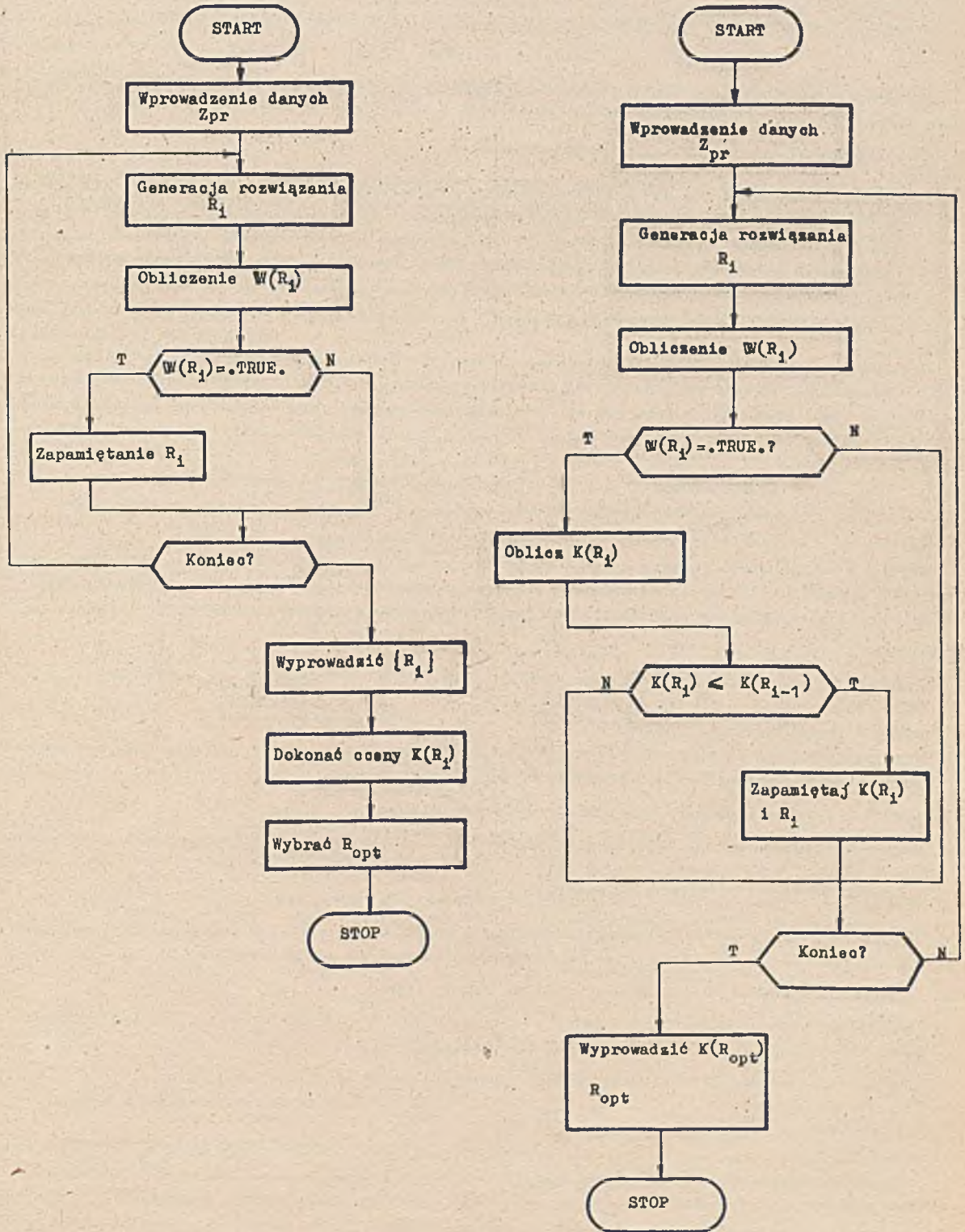
gdzie:  $S_k$  - zbiór koncepcji projektowych (może być jednoelementowy),  
 $\{P_i\}$  - zbiór parametrów zadania  $i = 1, \dots, l_0$   
 $\{\hat{P}_i\}$  - zbiór dopuszczalnych wartości parametrów  $P_i \in \hat{P}_i, i = 1, \dots, l_0$

Zbiór parametrów dzieli się na trzy podzbiory:

$$\{P_i\} = \{c_i\} \cup \{z_i\} \cup \{w_i\}$$

$$\{c_i\} \cap \{z_i\} \cap \{w_i\} = \emptyset$$

gdzie:  $\{c_i\}$  - zbiór stałych rozwiązania (narzucone są przez otoczenie)  $i = 1, \dots, l_1$   
 $\{z_i\}$  - zbiór zmiennych rozwiązania (wartości określone w trakcie wyznaczania rozwiązania)  $i = 1, \dots, l_2$   
 $\{w_i\}$  - charakterystyki techniczno-eksploatacyjne (własności funkcjonalne lub techniczno-eksploatacyjne)  $i = 1, \dots, l_3$ .



Rys. 1 a b . Ilustracja postępowania w metodzie 1

Między charakterystykami a stałymi i zmiennymi rozwiązania zachodzą związki typu:

$$w_i = f_i(c_1, c_2, \dots; z_1, z_2, \dots), \quad i = 1, 2, \dots, l_j$$

Dla procesu projektowania linii elektroenergetycznej interpretacja zadania projektowego ( $Z_{pr}$ ) jest następująca:

$S_k$  - { linia napowietrzna na słupach indywidualnych },

$\{c_i\}$  - { trasa linii łącznie z przeszkodami i innymi informacjami o terenie, typ przewodu, naprężenie przewodu ( $N_{obl}, N_{zm}$ ), współrzędna pierwszego słupa -  $x_1$ , współrzędna ostatniego słupa -  $x_m$  };

$\{z_i\}$  - { współrzędne słupów ( $x_2, x_3, \dots, x_{m-1}$ ), typy słupów ( $T_1, T_2, \dots, T_m$ ), obostrzenia pręseł ( $O_1, O_2, \dots, O_{m-1}$ ), naprężenia w przewodzie ( $N_1, N_2, \dots, N_{m-1}$ ) };

$\{w_i\}$  - { odległości normowe i inne zalecenia budowlane linii }.

Rozwiązanie projektowe ( $R$ ) jest natomiast opisem przedmiotu projektowanego [7],[8]. Ponieważ część wartości parametrów rozwiązania  $R$  (zob. definicja 1) wyznaczona jest w zadaniu projektowym  $Z_{pr}$  (np.  $x_1$ , typ przewodu) w toku procesu projektowania wyznacza się wartości parametrów  $R$ , które w zadaniu zaliczone są do parametrów zmiennych  $\{z_i\}$ . Dla napowietrznej linii elektroenergetycznej wyznacza się:

$m$	- liczbę słupów	$O_1, O_2, \dots, O_{m-1}$	- obostrzenia pręseł
$x_2, \dots, x_{m-1}$	- pozycje, współrzędne słupów	$N_1, N_2, \dots, N_{m-1}$	- naprężenia w przewodzie
$T_1, T_2, \dots, T_m$	- typy słupów		

Wartości parametrów  $R$  powinny należeć do zbioru wartości dopuszczalnych  $\{P_i\}$ . Zbiory dopuszczalnych wartości są najczęściej zadawane w postaci warunków (przy realizacji maszynowej obliczeń jest to nieodzwonne), więc zamiast zbioru  $\{P_i\}$  wprowadza się równoważny zbiór warunków  $W$ .

#### Definicja 2

Rozwiązanie projektowe jest poprawne, gdy  $R$  spełnia zbiór warunków  $W$ , co zapisujemy:

$$W(R) = \text{.TRUE.}$$

Zbiór warunków jest następujący

- słup powinien być zlokalizowany w miejscu dozwolonym na trasie linii (np. nie może być usytuowany na drodze komunikacyjnej krzyżującej się z trasą linii),
- pręseł nie może krzyżować się z niedozwolonymi obiektami,
- krzywa zwisania przewodu musi spełniać wymagania minimalnych prześwitów nad ziemią i obiektami, z którymi krzyżuje się linia,
- rozpiętość pręseł nie może przekraczać największej dopuszczalnej wartości -  $d_{max}$ ,
- typy słupów muszą spełniać wymagania dotyczące krzyżowanych obiektów przez pręseł, wartości kąta załomu trasy itp. oraz należą do ustalonego zbioru dopuszczalnych słupów,
- obostrzenia pręseł są zgodne z przepisami budowy linii,
- naprężenie w przewodzie dla pręseł z obostrzeniem 0,1,2 wynosi  $N_{obl}$ , dla pręseł z obostrzeniem 3 wynosi  $N_{zm}$ .

Celem projektowania jest otrzymanie rozwiązania projektowego  $R$  spełniającego

$$W(R) = \text{.TRUE.}$$

gdzie:  $K$  - kryterium optymalizacji wyboru rozwiązania

$$K(R) = \min_{r \in R} K(r)$$

$R$  - zbiór  $n$  rozwiązań projektowych.

Przejęcie od zadania projektowego ( $Z_{pr}$ ) do rozwiązania projektowego ( $R$ ) odbywać się może trzema zasadniczymi metodami.

#### Metoda 1

Generuje się zbiór rozwiązań  $R$ ,  $\bigwedge_{r \in R} W(r) = \text{.TRUE.}$ . Dla każdego rozwiązania oblicza się wartość kryterium  $K(r)$  i wybiera się to rozwiązanie, które ma "najlepszą" wartość kryterium. Odmianną tej techniki jest wytwarzanie tylko jednego rozwiązania, obliczenie dla niego wartości

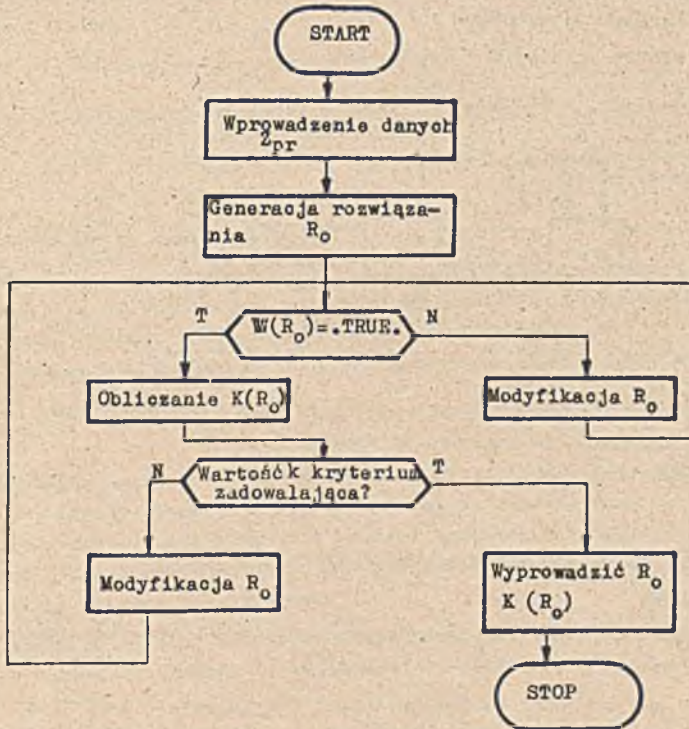


kryterium i porównanie z rozwiązaniem wcześniej uzyskanym. Gdy wartość kryterium jest "gorsza" odrzuca się to rozwiązanie, gdy jest "lepsza" - rozwiązanie zostaje zapamiętane i generuje się nowe rozwiązanie itd.

Dla prostej generacji rozwiązania (przeгляд zupełny - [4]) liczba możliwych rozwiązań (nie tylko takich, że  $W(r)=TRUE.$ ) jest większa od  $2^{3m-2}$ . Metodę tę ilustruje rys.1 [10].

#### Metoda 2

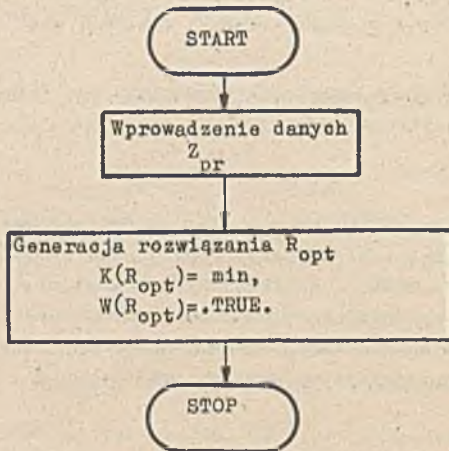
Generuje się rozwiązanie na podstawie oceny rozwiązań poprzednich. Do tej klasy metod zaliczyć można gradientowe metody optymalizacji lub następującą technikę postępowania: rozwiązanie projektowe wytwarza projektant, maszyna cyfrowa sprawdza czy  $W(r)=TRUE.$  i oblicza wartość kryterium oceny rozwiązania  $K(r)$ . Projektant na podstawie uzyskanych wcześniej rozwiązań i wartości kryterium oraz swojego doświadczenia i wiedzy podejmuje decyzje o modyfikacji rozwiązania, aby uzyskać rozwiązanie o "lepszej" wartości kryterium. Na rys. 2 przedstawiono możliwe postępowanie przy realizacji tej techniki.



Rys.2. Ilustracja postępowania w metodzie 2

#### Metoda 3

Generuje się tylko jedno rozwiązanie, które jest "najlepsze", w myśl przyjętego kryterium oceny rozwiązania (rys.3). Oczywiście, przy wytwarzaniu rozwiązania nie można wykorzystać metod 1 i 2. Technika taka jest możliwa, gdy parametry zmienne  $\{z_i\}$  mają rozwiązanie optymalne w postaci jawnej, np. z obliczeń analitycznych wynika, że linia elektroenergetyczna jest rozwiązaniem optymalnym, gdy rozpiętość pręci  $d_i = const, i = 1, \dots, m-1$ . Dla linii elektroenergetycznej nie można opracować metody 3 ze względu na uwiklaną postać zależności między parametrami rozwiązania R. Sytuację ilustruje rys.4. Strzałki pokazują, jakie informacje są potrzebne, aby dokonać wyboru

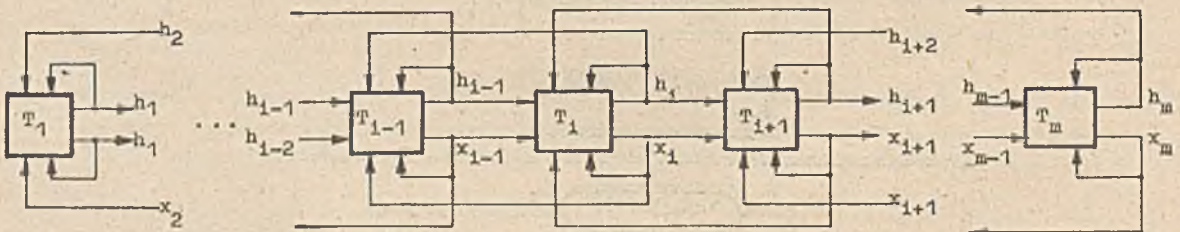


Rys.3. Ilustracja postępowania w metodzie 3

słupa  $T_i$ ,  $i = 1, \dots, m$  i sprawdzenia, czy  $W(R) = \text{TRUE}$ .

Ponieważ istnieją sprzężenia zwrotne, trudno przewidzieć poprawność decyzji w wyborze typu i lokalizacji słupa  $i$ , a także wpływu tej decyzji na wartość kryterium, bo konsekwencje tej decyzji przenoszą się także na słup  $i$  przez słupy  $i-1$  oraz  $i+1$ .

Zatem otrzymanie rozwiązania linii elektroenergetycznej jest możliwe jedynie na drodze wykorzystania metody 1 lub 2, być może z pewnymi modyfikacjami wynikającymi ze specyficznych własności przedmiotu projektowanego.



Rys.4. Charakter zależności między parametrami linii elektroenergetycznej

$h_i$  - wysokość zawieszenia przewodu na słupie  $i$   
 $x_i$  - współrzędna słupa  $i$

#### Komputerowe metody projektowania linii elektroenergetycznych - opis

##### Klasa prezentowanych metod

Teoretyczne ujęcie projektowania linii elektroenergetycznych sprowadza proces do zagadnień programowania dyskretnego (optymalizacji dyskretnej) [4].

Trzy pierwsze metody swój rodowód czerpią z teorii rozwiązywania takich problemów, a przebieg procesu projektowania odbywa się tak, jak na rys.1b. Metoda ozwarta, zaproponowana przez autora artykułu, wywodzi się z przedstawienia procesu projektowania z punktu widzenia takich pojęć informatyki jak język struktury danych [7] i pozwala na prowadzenie procesu projektowania zgodnie ze schematem z rys.2.

##### Metoda kombinatoryczna

Metodę opisano na podstawie pracy Kinsnera i Mrowlińskiego [2]. Polega ona na wariantowaniu wysokości słupów, a pozostałe parametry rozwiązania  $R$  (definicja 1) wyznacza się z zależności występujących między parametrami (np. wiadomo, że jeśli obostrzenie pręgnięcia  $i$  jest 0, to napięcie przewodu  $N_i = N_{obl}$ ).

Wybór pewnych parametrów  $R$  jako zmiennych niezależnych znacznie zmniejsza liczbę rozwiązań w stosunku do generacji prostej opisanej poprzednio.

Metoda składa się z pięciu kroków.

- Krok 1. Przyjęcie liczby słupów  $m$  i zadanie wysokości początkowych  $\bar{h}_0 = (h_1, h_2, \dots, h_m)$ .
- Krok 2. Na podstawie zadanej liczby słupów i ich wysokości ustala się pozycje i typy słupów na trasie linii, tak aby  $W(r_1) = \text{TRUE}$ . Pozycje słupów ustala się sekwencyjnie począwszy od słupa  $i=1$  o zadanym położeniu  $x_1$ , w następujący sposób: dla słupa  $i$ ,  $i=1, 2, \dots, m-1$  "wykreśla się" krzywą zwisu przewodu zachowującą minimalne odległości od terenu i obiektów krzyżowanych. Dla słupa  $i+1$  wyznacza się krzywą słupa, tzn. łukową krzywą, która obrazuje pozycję słupa  $i+1$ , że przy zadanych wysokościach wykreślona krzywa zwisu przewodu jest rozpięta na słupach  $i$ ,  $i+1$ . Przecięcie krzywej słupa z terenem wyznacza lokalizację słupa  $i+1$ . Gdy występuje kilka przecięć, wybiera się punkt najdalej położony od słupa  $i$ . Gdy nie ma punktów przecięcia, ustawia się słup odporowy na końcu dopuszczalnego odcinka i obłożenia prowadzi się dalej.
- Krok 3. Gdy kombinacja słupów zawiera słupy odporowe odrzuca się rozwiązanie, gdy zawiera słupy przelotowe oblicza się wartość kryterium oceny rozwiązania. Kryterium jest koszt użytych słupów.
- Krok 4. Porównuje się wartość kryterium z wartością kryterium rozwiązania poprzedniego i wybiera się wariant tańszy.
- Krok 5. Przyjęcie nowych wartości na wysokości słupów i generacja nowego rozwiązania - krok 2.

Liczba możliwych kombinacji wysokości słupów, dla których sprawdza się rozwiązania jest równa:

$$I_R = n^m$$

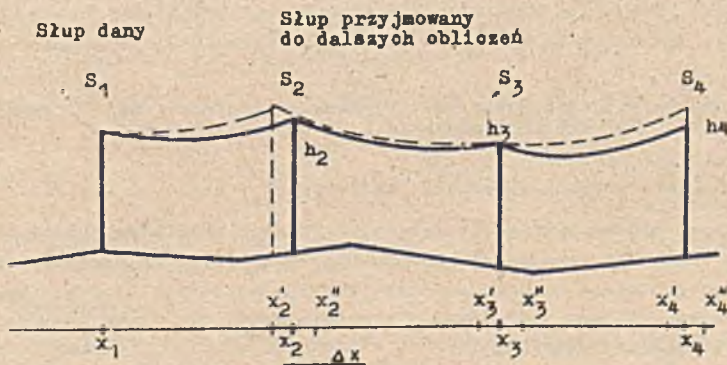
gdzie:  $n$  - liczba różnych wysokości słupów,  
 $m$  - liczba słupów.

Metoda z podziałem na grupy kombinatoryczne

Metodę przedstawiono w pracy Mazanka [5]. Polega ona na rozpatrywaniu wieloprzęsłowych odcinków linii, z których każdy rozpoczyna się od słupa określonego typu o zadanym położeniu i wysokości. Dla ustalonego słupa  $S_1$  rozpatruje się słupy:  $S_{1+1}$ ,  $S_{1+2}$ ,  $S_{1+3}$  i wybiera się słup  $S_{1+1}$ , tj. taki, który występuje w kombinacji dającej najmniejszy koszt jednostkowy trójprzęsłowego odcinka linii.

Kroki metody

- Krok 1. Dobranie słupów narożnych i krańcowych. Słupy te mają wyznaczone już pozycje, na podstawie podanej zasady rozpatruje się dla słupów krańcowych trzy przęsła, dla słupów narożnych sześć przęseł, bada wszystkie możliwe kombinacje i wybiera się taki rozstaw, który daje koszt jednostkowy minimalny. Z tych rozstawów przyjmuje się do dalszych obliczeń jedynie słupy narożne lub krańcowe.
- Krok 2. Mając wyznaczony słup  $S_1$  przeprowadza się rozstawienie trzech kolejnych słupów badając wszystkie możliwe stanowiska i wysokości słupów zaczynając od słupów najniższych (rys.5).



Rys.5. Ilustracja metody z podziałem na grupy kombinatoryczne

Krok 3. Wyznacza się koszt jednostkowy trójprzęsłowego odcinka i wybiera się taki rozstaw, który daje koszt minimalny.

Krok 4. Przyjmuje się pozycję i typ tylko jednego słupa bezpośrednio po początkowym. Słup wybrany i usytuowany w powyższy sposób stanowi początek w kolejnym trójprzęsłowym odcinku (krok 2).

Przyjmując, że  $n$  - różnych wysokości słupów i  $k$  różnych pozycji słupa należy rozpatrzyć, otrzymuje się liczbę możliwych rozwiązań, które należy zbadać:

$$I_R = (n \cdot k)^3 \cdot m$$

gdzie:  $m$

- liczba słupów

$$k = \left\lceil \frac{d_{\max}}{\Delta x} \right\rceil$$

$d_{\max}$

- maksymalna rozpiętość przęsła

$\Delta x$

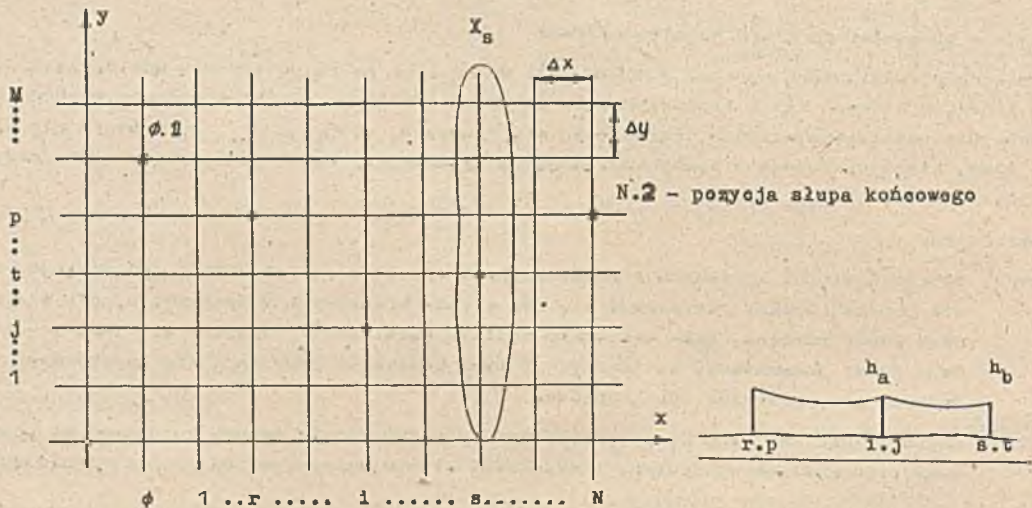
- minimalna odległość między kolejnymi pozycjami słupów

Metoda programowania dynamicznego

Metoda opisana jest na podstawie pracy Olbrychta [9]. Polega ona na dyskretnej wielostadialnej technice podejmowania decyzji.

Kroki metody

Krok 1. Wyznaczenie pozycji testowych słupów, tzn. takich punktów terenu, w których mogą stać słupy i zaznaczenie pozycji dla słupa początkowego i końcowego. Na rys.6 przedstawiono graficzną interpretację punktów przy założeniu, że tworzą siatkę prostokątną.



Rys.6. Ilustracja metody programowania dynamicznego (punkty tworzą siatkę prostokątną)

Krok 2. Wyznaczenie  $N$  podzbiorów pozycji testowych  $X_s$ ,  $s = 1, 2, \dots, N$ .

Krok 3. Dla każdego podzbioru  $s = 1, \dots, N$  wyznacza się cząstkowe rozwiązanie optymalne ( $r_s$ ) dla wszystkich możliwych wysokości słupów ustawionych w pozycjach należących do  $X_s$  w funkcji współrzędnych słupa przodostatniego. Pozycję słupa oznaczać się będzie następująco:  $i.j$  - co oznacza, że jest to pozycja  $j$  z podzbioru  $i$ . Cząstkowe rozwiązanie optymalne otrzymuje się w sposób niżej podany.

- Dla  $s = 1$  lub  $s \geq 2$  i istnieją jednoprzęsłowe połączenia pozycji  $s.t$ ,  $t = 1, \dots, M$ , a  $\beta.1$

Wyznacza się jednoprzęsłowe połączenia pozycji  $s.t$ ,  $t = 1, \dots, M$ , a pozycją słupa po-

czątkowego  $\beta.1$  o zadanej wysokości  $\alpha_1$  takie, aby  $W(r_0) = \text{TRUE}$ . Dla każdej pozycji słupa  $s.t$  i wysokości  $h_b$ ,  $b = 1, \dots, n$  oblicza się wartość kryterium:

$$F_{stb}(\beta.1) = K_0 + K_2 \cdot d_1$$

gdzie:  $K_0$  - koszt początkowego słupa linii  
 $K_2$  - jednostkowy koszt przewodu  
 $d_1$  - odległość pozioma między pozycją  $\beta.1$  a  $s.t$   
 $b$  - indeks wysokości ustawionego słupa w  $s.t$

Kryterium nie obejmuje kosztów słupa w  $s.t$  i przedstawione jest w funkcji pozycji słupa poprzedniego.

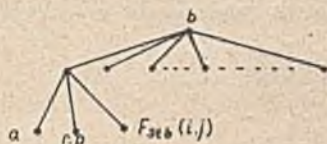
- Dla  $s \geq 2$  i pozycje  $s.t$ ,  $t = 1, \dots, M$  nie mają połączeń jednoprzęsłowych z pozycją  $\beta.1$ . Wyznacza się jednoprzęsłowe połączenia pozycji  $s.t$ ,  $t = 1, \dots, M$  z pozycjami  $i,j$ ,  $i, j = 1, \dots, M$ , takie aby  $W(r_0) = \text{TRUE}$ . Dla każdej wysokości słupa ustawionego w  $s.t$  oblicza się wartość kryterium:

$$F_{stb}(i,j) = \min_{r,p,a} [K_2 \cdot d_2 + F_{ija}(r \cdot p) + K(T,h,q)_{i,j}]$$

gdzie:  $d_2$  - odległość pozioma między  $s.t$  a  $i,j$   
 $K(T,h,q)_{i,j}$  - koszt słupa typu  $T$ , wysokości  $h$  przystosowanego do rodzaju gruntu  $q$  ustawionego w pozycji  $i,j$   
 $F_{ija}(r \cdot p)$  - optymalny koszt budowy linii od pozycji  $\beta.1$  do pozycji  $i,j$ , w której umieszczono słup o indeksie wysokości  $a$ , nie obejmujący kosztu słupa w  $i,j$ . Pozycja przedostatniego słupa wynosi  $r,p$ .

Dla każdej pozycji  $s.t$ ,  $t = 1, \dots, M$  uzyskuje się rodzinę rozwiązań optymalnych uzależnionych od wysokości słupa  $h_b$ ,  $b = 1, \dots, n$  i pozycji słupa poprzedniego  $i,j$ ,  $i < s$ ,  $j = 1, \dots, M$ , takie  $j$ , że  $W(r_0) = \text{TRUE}$ .

Aby móc przeprowadzać obliczenia dla każdej pozycji muszą być pamiętane następujące informacje:



$$b = 1, \dots, n$$

$$i,j \quad i < s, j = 1$$

$b$  - indeks wysokości słupa w pozycji  $s.t$   
 $i,j$  - pozycja słupa poprzedniego

$a$  - indeks wysokości optymalnej w pozycji  $i,j$   
 $r,p$  - optymalna pozycja słupa

Przyjmując, że istnieje  $L = \left[ \frac{d_{\max}}{\Delta x \Delta y} \right]$  poprawnych wariantów rozwiązania przęsła na pozycjach  $i,j$ , a  $s.t$  to dla każdej pozycji  $s.t$  potrzeba rozważyć:

$$1_{r_0} = n^2 \cdot L^2 \quad \text{rozwiązań cząstkowych oraz pamiętać n.6.L liczb.}$$

Krok 4. Postępując od pozycji  $N.2$  (pozycja słupa końcowego) do pozycji  $\beta.1$  odtwarza się te połączenia między pozycjami, które tworzą ogólnie rozwiązanie optymalne. Wyznaczenie pozycji optymalnej słupa przedostatniego odbywa się na drodze poszukiwania takiej pozycji  $i,j$ , która zapewnia:

$$\min_{i,j} F_{N_{s_0} 2}(i,j)$$

gdzie:  $N.2$  - pozycja słupa końcowego  
 $\alpha_2$  - indeks zadanej wysokości słupa końcowego  
 $F_{N_{s_0} 2}(i,j)$  - optymalny koszt budowy linii od pozycji  $\beta.1$  do pozycji  $N.2$  przechodzącej przez pozycję  $i,j$

Ponieważ dla każdej pozycji pamiętane są odpowiednie informacje (zob. krok 3), więc możliwe jest teraz wyznaczenie pozycji i typów następnych słupów.

#### Metoda dekompozycji i idealizacji

Zasady metody prezentowane były w pracy [6]. Metoda polega na stopniowej dekompozycji informacji zagregowanej na informacje szczegółowe, aż do osiągnięcia poziomu wyznaczonego przez definicję przedmiotu projektowanego.

## Kroki metody

Krok 1. Otrzymanie rozwiązania projektowego z definiowanego jako ciąg przęseł. W kroku tym należy wyznaczyć współrzędne przęseł. Rozwiązanie projektowe otrzymuje się w sposób niżej podany.

Tworzy się rozwiązanie projektowe w warunkach idealnych, optymalne w sensie przyjętego kryterium  $k$ :

$$k = \frac{1}{m-1} \sum_{i=1}^{m-1} \frac{kP}{d_i}$$

gdzie:  $m-1$  - liczba przęseł  
 $kP$  - koszt przęśla (koszt słupów i koszt przewodu)  
 $d_i$  - rozpiętość przęśla  $i$

Idealizacja warunków polega na przyjęciu następującego zbioru  $\{o_1\}$ :

- na trasie linii nie ma przeszkód
- przekrój terenu jest płaski na całej długości trasy linii

Z uzyskanego rozwiązania przyjmuje się pozycje słupów  $(x_2, x_3, \dots, x_{m-1})$ , które są początkami i końcami wyznaczonych przęseł, następnie ustala się nowe pozycje uwzględniając przeszkody występujące na trasie i obszary wymagane (tzn. takie odcinki trasy, na których projektant żąda, aby znajdował się słup) i ograniczenie:

$$d_i < d_{\max} \quad i = 1, \dots, m-1$$

Nowe pozycje słupów spełniają:

$$\min \sum_{i=1}^{m-1} (x_{i \text{ stare}} - x_{i \text{ nowe}})^2$$

Krok 2. Wyznaczenie z sytuacji terenowej obostrzeń przęseł.

Krok 3. Wyznaczenie na podstawie obostrzeń przęseł rodzajów słupów (odporowe, krańcowe, przelotowe itp.)

Krok 4. Dobór typów słupów takich, aby pozostałe warunki nakładane na rozwiązanie projektowe były spełnione.

## Opisowanie metod - porównanie

Dla skrócenia zapisu wprowadza się następujące oznaczenia:

- MK - metoda projektowania kombinatoryczna
- MGK - metoda projektowania z podziałem na grupy kombinatoryczne
- MPD - metoda projektowania oparta na programowaniu dynamicznym
- MDI - metoda projektowania dekompozycji i idealizacji

Ogólne porównanie metod ze względu na dane i otrzymywane wyniki przedstawiono w tab.1. Dla metod projektowania informacja wejściowa, to zbiory  $\{o_1\}$  i  $\{w_1\}$ . Ponieważ należy założyć, że przedmioty projektowane spełniają wszystkie wymagania nakładane przez normy i przepisy budowy, więc zbiory  $\{w_1\}$  we wszystkich metodach muszą być takie same. W tab.1 podano więc jedynie zbiór  $\{o_1\}$ . Metoda MPD, biorąc pod uwagę zbiór informacji wejściowej, jest najbardziej ogólna - podaje się pas terenu, na którym planuje się budowę linii. W innych metodach podaje się informację bardziej szczegółową, tj. trasę linii. Ogólność metody MPD pociąga za sobą wzrost kosztów przygotowania danych, ponieważ nie tylko ważny jest kształt terenu, ale także rodzaj gruntu, więc w każdym punkcie testowanym należy wykonać pomiar gruntu ze względu na własności mechaniczne i elektryczne. Gdy w praktyce będzie się przyjmował rodzaj gruntu bez dokładnych badań w terenie (co ma miejsce w biurach projektów), to ogólność MPD nie będzie przynosiła korzyści w porównaniu z pozostałymi metodami.

Podobnie przedstawia się problem czasu obliczeń - zbyt długi okres oczekiwania na wynik zniechęca do korzystania z metody. Czas, po którym otrzymuje się rozwiązanie projektowe optymalne, jest proporcjonalny do liczby rozpatrywanych wariantów rozwiązania lub rozwiązań częściowych. W tab.2 przedstawiono oszacowanie liczby rozpatrywanych wariantów.

Dla wyrobienia sobie poglądu o rzędzie wielkości w tab.3 przedstawiono czas obliczeń i zapotrzebowanie na pamięć dla podstawowych informacji dla następujących przykładów danych:

- długość trasy linii - 10 tys. m
- $\Delta x = 20$  m
- maksymalna rozpiętość przęśla  $d_{\max} = 300$  m
- liczba różnych wysokości słupów  $n = 3$

Tab.1. Porównanie ogólne

Nazwa metody	Informacje wejściowe $\{c_i\}$	Wyniki $\{z_i\}$	Możliwość ignorancji projektanta w rozwiązaniu projektowe
MK	Trasa linii w postaci przekroju terenu, przeszkody, typ przewodu, napięcie obliczeniowe, napięcie zmniejszone	Pozycje słupów przelotowych	Nie przewidziano - istnieje możliwość wpływania na rozwiązanie przez wprowadzenie dodatkowych przeszkód
MGK	Trasa linii w postaci przekroju terenu, przeszkody, typ przewodu, napięcie obliczeniowe, napięcie zmniejszone	Pozycje słupów, typy słupów, długości sekcji odcinkowych	Nie przewidziano - istnieje możliwość wpływania na rozwiązanie przez dobór $\Delta x$
MPD	Pas terenu w postaci punktów testowych ustawienia słupów, rodzaj gruntu, przeszkody, typ przewodu, napięcia obliczeniowe i zmniejszone	Pozycje słupów, typy słupów, kąty załomu trasy	Nie przewidziano - istnieje możliwość wpływania na rozwiązanie przez dobór punktów testowych ustawienia słupów
MDI	Trasa linii w postaci przekroju terenu, przeszkody, obszary wymagane, typ przewodu, napięcie obliczeniowe i zmniejszone	Pozycje słupów obostrzenia przepięć, napięcie w przewodzie	Przewidziano możliwość wpływania na rozwiązanie przez zadawanie abstrakcyjnych przeszkód i obszarów wymaganych

Tab.2. Liczba rozpatrywanych rozwiązań

Nazwa metody	Liczba wariantów	Uwagi
MK	$n^m$	
MGK	$(n \cdot k)^m$	$k = \left[ \frac{d_{\max}}{\Delta x} \cdot 0 \right] \quad 0 < c < 1$
MPD	$n^2 \cdot L^2 \cdot H \cdot N$	$L = \left[ \frac{d_{\max}^2}{\Delta x \Delta y} \right]$
MDI	-	Nie ma wariantowania w takim sensie, jak w pozostałych metodach

$n$  - liczba możliwych wysokości słupów  
 $m$  - liczba słupów  
 $N \cdot M$  - liczba punktów testowych (siatka prostokątna)

Tab.3. Przykłady liczbowo

Nazwa metody	Czas obliczeń	Pojemność pamięci
MK	$2 \cdot 10^5$ lat	2400 bajtów
MGK	45 min	2000 bajtów
MPD	17 min	900 000 bajtów
MDI	1 s	2500 bajtów

MPD jest zredukowana do otrzymywania rozwiązania przy zadanej trasie linii, przyjęto założenie, że badanie poprawności jednego przepięcia trwa  $\sim 1$  ms (dla porównania czas dostępu do pamięci jest ok.  $1 \mu s$ ).

Ze względu na czas obliczeń MK jest praktycznie nieprzydatna, podobnie jak MPD ze względu na zapotrzebowanie na dużą pamięć.

W pracy [9] autor MPD podaje oszacowanie pamięci dla rzeczywistych danych projektowych -  $10^{10}$  bajtów, dla porównania jest to równoważne stu pamięciom dyskowym o pojemnościach 100 Mbajtów.

Na krótko oczekuje się na wynik z MDI, ponieważ w metodzie tej nie korzysta się z wariantowania.

Projektant opierając się na swojej wiedzy może przez odpowiedni dobór przeszkód abstrakcyjnych i obszarów wymaganych otrzymywać różne rozwiązania dla zadanej trasy linii. Dodatkowo, dzięki

obszarom wymagany może uwzględnić żądania władz terenowych dotyczące lokalizacji słupów.

#### Literatura

- [1] DĄBIŃSKI C.: Elementy nauki o projektowaniu. Warszawa: WNT 1972.
- [2] KINSNER K., MROMLIŃSKI R.L.: Zasady projektowania napowietrznych linii elektroenergetycznych. Wrocław: Politechnika Wroclawska 1970
- [3] KOŁODZIEJ R.: Zagadnienia dekompozycji problemów w projektowaniu technicznym. Rozprawa doktorska. Instytut Cybernetyki Politechniki Wroclawskiej. Komunikat 1977 nr 528
- [4] KORBUT A., FINKELSZTEIN J.J.: Programowanie dyskretne. Warszawa: PWN 1974
- [5] MAZANEK Z.: Optymalizacja rozstawu słupów linii W.N. przy zastosowaniu m.c. Energoprojekt - Kraków, PT-3.2.Z - b/69 X-35560
- [6] MOCALA J.: Automatyzacja projektowania napowietrznych linii elektroenergetycznych. Algorytm i implementacja. Referat wygłoszony na I Konferencji Metod i Środków Projektowania Automatycznego. Warszawa, 23-26.XI.1977
- [7] MOCALA J.: Zagadnienia modelu procesu projektowania w komputeryzacji projektowania technicznego. Warszawa: IMM 1977 Archiwum Opracowań nr 35
- [8] MOCALA J.: Pewien warunek realizacji komputerowego wspomaganie projektowania. Ogólnopolska Konferencja, Projektowanie III, Wrocław 20-22.09.1978
- [9] OLBRYCHT L.: Automatyzacja projektowania trasy elektroenergetycznej linii napowietrznej w obrębie określonego pasa terenu. I Konferencja "Metody i Środki Projektowania Automatycznego" Warszawa, 23-26.XI.1977
- [10] SIELICKI A., JELENIEWSKI T.: Podstawy metodologii projektowania. Wrocław: Politechnika Wroclawska 1978
- [11] PN-67/E-05100



## z cyklu: Programowanie obrabiarek

inż. Adam BERTHOLD  
mgr inż. Jorzy GALICKI  
inż. Sławomir TOPOLSKI  
Zakład Doświadczalny  
Instytutu Maszyn Matematycznych

## Automatyczna synteza elementów programu dla obrabiarek sterowanych numerycznie

Obrabiarki sterowane numerycznie (OSN) osiągnęły obecnie tak wysoki poziom rozwoju, że wprowadzane są praktycznie we wszystkich fazach produkcji przemysłowej. Skutkiem tego w ostatnich latach bardzo wydatnie wzrósł popyt na ich oprogramowanie. Colowe stały się prace nad wszelkiego typu generatorami, pozwalającymi na skrócenie czasu tworzenia programu obróbki części.

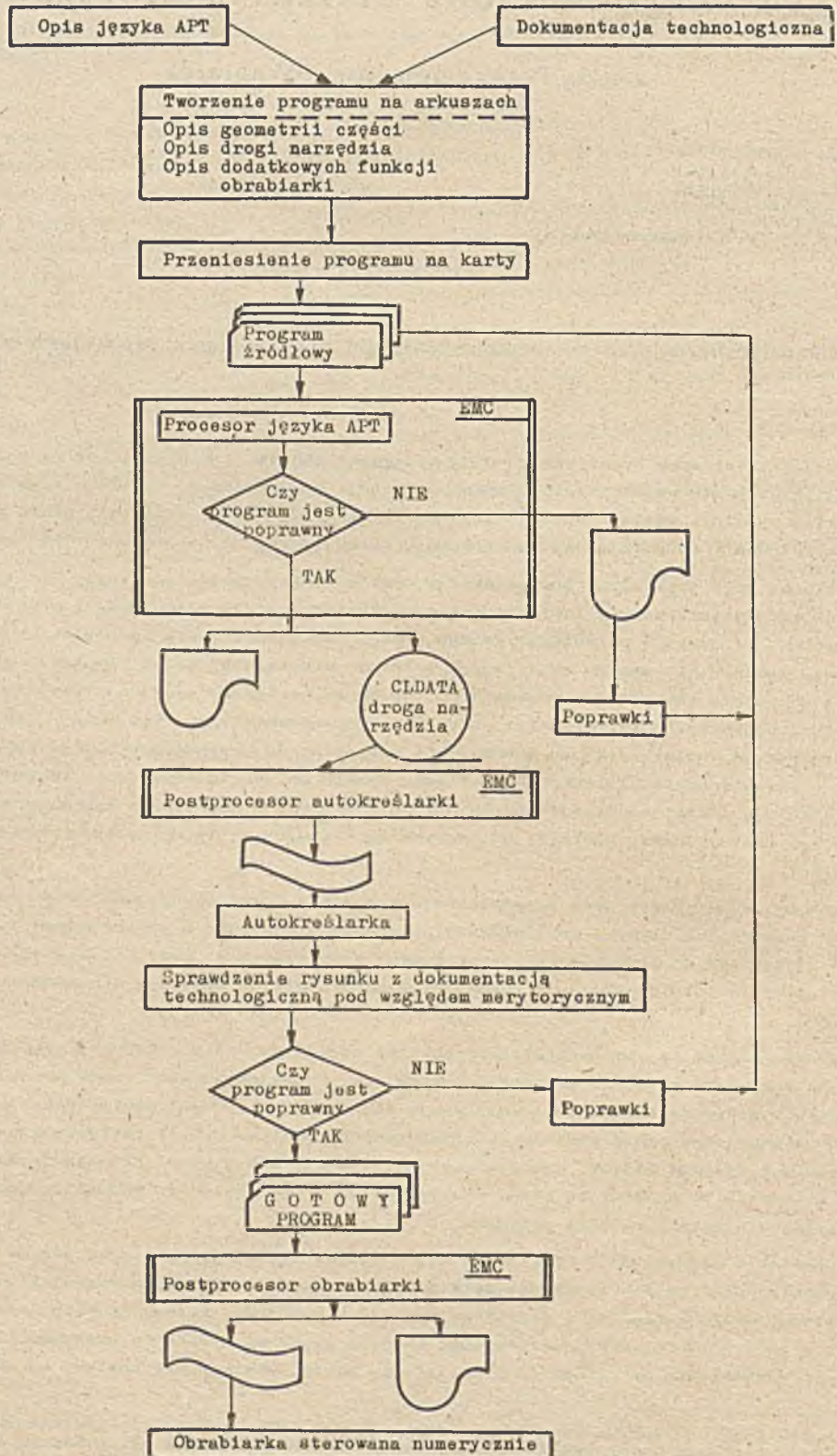
Obrabiarki tego typu można programować w sposób ręczny, półautomatyczny i automatyczny. W zakresie naszych zainteresowań leży automatyczny sposób programowania OSN. W przeciwieństwie do programowania ręcznego i półautomatycznego wymaga on zastosowania specjalnego języka programowania w celu zakodowania danych wejściowych w formie zrozumiałej przez komputer. W języku tym opisyje się geometrię obrabianego przedmiotu, tor ruchu narzędzia oraz dodatkowe funkcje (czynności) obrabiarki. Najbardziej uniwersalnym językiem programowania tego typu jest język APT (Automatically Programmed Tools). Jest on szczególnie przydatny do przygotowywania programów obróbki przedmiotów o przestrzennie złożonych kształtach. Wymaga on jednak stosowania komputerów o pamięci 256Kb. Istnieją liczne adaptacje języka APT na komputery o mniejszych pamięciach. Należą do tej grupy m.in. EXAPT, IFAPT, MINIAPT, CL. Języki te stosuje się do opisywania obróbki mniej złożonych części.

Syntetyzator ABST, który jest przedmiotem opracowania, dostosowany jest do języka APT, może być jednak z powodzeniem używany do tworzenia programów w dowolnym z wymienionych języków. Uklonkowanie na język APT spowodowane było wymaganiami zleconiodawcy (tzn. WSK-Mielec). Części występujące w przemyśle lotniczym mają z reguły złożone kształty i wymagają stosowania tego właśnie języka.

Klasyczna metoda tworzenia programów obróbki części na OSN za pomocą języka APT, lub podobnego, polega na dokładnym opisanu geometrii przedmiotu, zaprojektowaniu ruchów narzędzia obrabiarki, uzupełnieniu programu o dodatkowe instrukcje dla obrabiarki. Cykl postępowania podczas tworzenia programu sterującego pracą obrabiarki przedstawiony jest na rys.1. Postępowanie to opiera się na możliwościach systemu NARVIK, opracowanego na zlecenie WSK-Mielec w Zakładzie Doświadczalnym IMM. Należy zauważyć, że możliwe są pewne odstępstwa od przedstawionego schematu, jednak jak wynika z praktyki, nie są one na ogół znaczne.

Na podstawie dokumentacji konstrukcyjnej i technologicznej, opierając się na zasadach języka APT, technolog-programista sporządza opis geometrii części, a następnie projektuje sposób obróbki, tzn. opisuje drogę narzędzia i dodatkowe funkcje obrabiarki. Należy zauważyć, że czynności te obejmują programiście ok.60% całkowitego nakładu pracy na wykonanie programu. Tak wykonany program po przeniesieniu na nośnik informacji (np. karty) zostaje uruchomiony na EMC, tzn. jest

\*) Język ten jest omówiony w opracowaniu "Opis języka APT - podręcznik programowania". Warszawa: IMM 1977, Archiwum Opracowań IMM nr 18, testy instrukcji języka APT wykonano w Zakładzie Doświadczalnym IMM i opublikowano (dokumentacja) w Archiwum Opracowań IMM nr 44, 45, 46 - Zbiór testów instrukcji języka APT, cz.1, cz.2, Testy instrukcji TOLER, INTOL, OUTTOL, Warszawa: IMM 1978.

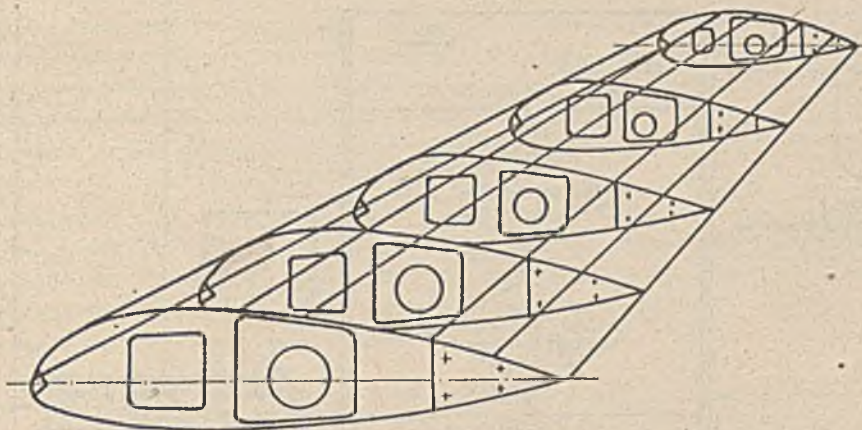


Rys. 1. Klasyczny cykl tworzenia programu sterującego pracą obrabiarki

przetwarzany przez procesor APT. Po usunięciu błędów formalnych - postprocesor autokreślarki przetwarza program i droga narzędzia jest wyrysowywana. Rysunek drogi narzędzia wykonany jest w celu sprawdzenia poprawności programu pod względem merytorycznym z dokumentacją technologiczną.

Po sprawdzeniu poprawności otrzymanego rysunku z dokumentacją technologiczną oraz ewentualnym usunięciu błędów merytorycznych, program przetwarzany jest przez postprocesor konkretnego układu sterowania obrabiarką i służy do sterowania procesem skrawania.

Cykl ten powtarzany jest od nowa dla każdego nowego programu nawet wtedy, gdy wykonujemy serię programów obróbki części bardzo podobnych. Przykładem może być rodzina żebur zbieżnego skrzydła samolotu (rys.2). Żebra te różnią się wielkością i ewentualnie rozmieszczeniem wybrań ulgowych, ale jak łatwo zauważyć geometrycznie są podobne.



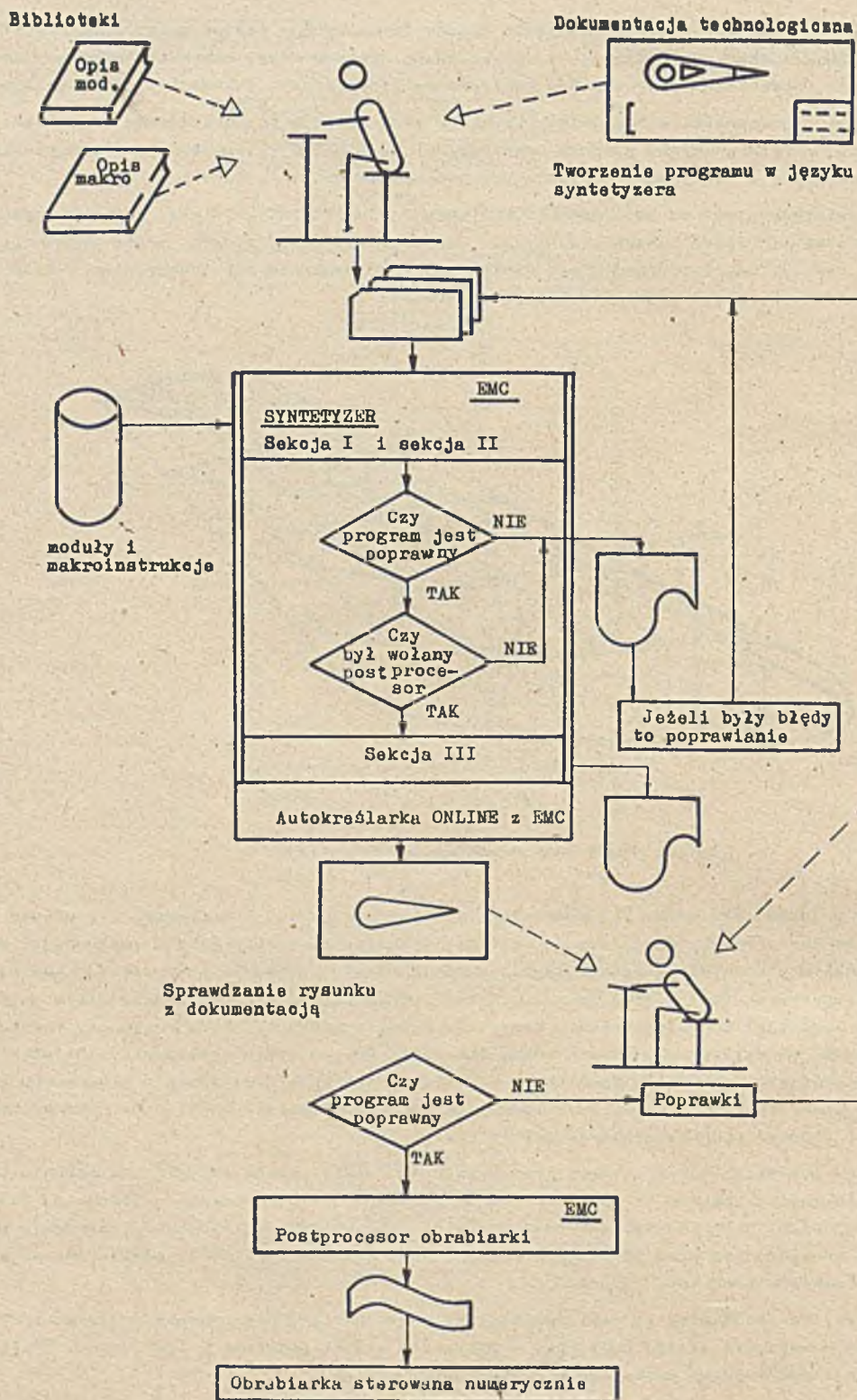
Rys.2. Poglądowy rysunek szkieletu skrzydła

Opisywanie w języku APT geometrii przedmiotu i drogi narzędzia jest prosto, ale bardzo pracochłonne. Wymaga ono z reguły napisania od kilkuset do kilku tysięcy prostych instrukcji. Po napisaniu kilkunastu złożonych programów tego typu można łatwo zauważyć, że zawierają one wiele bardzo podobnych fragmentów. Wynika to oczywiście z podobieństwa geometrycznego wielu części. Jednak nawet w częściach o ogólnie różnym kształcie można wyodrębnić pewne fragmenty podobne. Należy wnioskować, że cykl programowania można znacznie skrócić przez stworzenie biblioteki, zawierającej opisy formalne tych fragmentów i wielokrotne ich wykorzystywanie w programach, z podaniem tylko parametrów aktualnych. Zysk będzie oczywiście większy, jeżeli mamy wykonać programy rodziny części o dużym podobieństwie geometrycznym.

Celem, który postawili sobie autorzy syntetyzera ABST było przede wszystkim skrócenie czasu tworzenia opisu geometrycznego i opisu ruchów narzędzia, a więc skrócenie najbardziej czasochłonnego etapu pracy programisty, jak również pewne usprawnienie całego cyklu tworzenia programu. W efekcie prowadzonych prac powstał procesor o nazwie SYNTETYZER ABST, służący do automatycznej syntezy elementów programu.

W pracy przyjęto założenie, że użytkownik syntetyzera potrafi programować w języku APT, w związku z czym pozostawia się mu możliwość dopisywania części programu w tym języku. Użytkownik ma do dyspozycji podprogramy umieszczone w dwóch bibliotekach:

- w bibliotece zawierającej podprogramy opisujące typowe fragmenty części,
- w bibliotece zawierającej tzw. makroinstrukcje, tzn. podprogramy rozszerzające możliwości języka APT.



Rys. 3. Cykl tworzenia programu za pomocą syntetyzera ABST

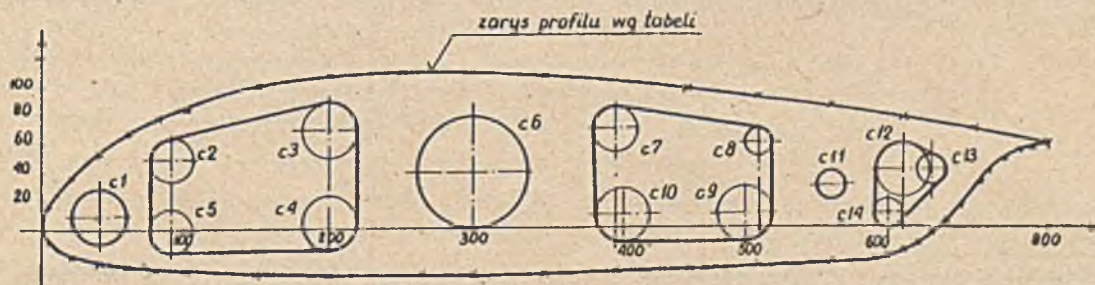
W celu wygenerowania przy użyciu syntetyzera programu opisującego geometrię części, należy utworzyć program składający się z instrukcji w języku syntetyzera ABST, wywołujących odpowiednie podprogramy z biblioteki oraz z fragmentów napisanych przez programistę w języku APT. Sekcja I syntetyzera ABST generuje w pamięci operacyjnej program będący sumą wywołanych i ustawionych w odpowiedniej sekwencji podprogramów z bibliotek oraz wyżej wspomnianych fragmentów programu. Program ten jest w całości w języku APT. Sekcja I kończy się wydrukowaniem programu opisującego geometrię części. Następnie przetwarzanie przechodzi do sekcji II (tzn. do procesora języka APT) i wykonywana jest typowa faza APT. Po jej zakończeniu, zależnie od wybranego przez programistę urządzenia wyjściowego, program przechodzi do sekcji III lub jego przetwarzanie kończy się. W sekcji III dane wyjściowe z sekcji II są przetłumaczone na język autokreślarki i zapisane na taśmie magnetycznej (papierowej) lub/i skierowane na autokreślarkę w celu wyrysowania. Na rys. 3 pokazano w sposób poglądowy cykl tworzenia programu za pomocą omawianego syntetyzera.

W budowie syntetyzera wykorzystano możliwości rozbudowy systemu APT, a mianowicie po stosunkowo niewielkich zmianach w programie sterującym uzyskano możliwość kompletowania programów z fragmentów, znajdujących się na urządzeniach o bezpośrednim dostępie, z jednoczesnym dodawaniem fragmentów programu z urządzenia wejściowego (np. czytnika kart). W celu wygenerowania programu obróbki części należy napisać program w języku syntetyzera ABST. Język ten jest bardzo prosty, zawiera zaledwie dziesięć instrukcji:

- instrukcja TIT - służy do wyboru grupy modułów z biblioteki oraz nadaje programowi tytuł
- instrukcja MOD - służy do wywołania z biblioteki modułu o podanym numerze
- instrukcja DTA - służy do podawania aktualnych danych do wywołanego uprzednio modułu
- instrukcja COM - służy do umieszczenia w programie dowolnego komentarza
- instrukcja MAC - sygnalizuje procesorowi, że program korzysta z danej makroinstrukcji w bibliotece
- instrukcja MAC-R - działa analogicznie jak MAC, z tym, że powoduje umieszczenie wywołanej makroinstrukcji w ściśle określonym miejscu programu
- instrukcja APT - służy do umieszczenia w programie fragmentów opisu części lub drogi narzędzia napisanych w języku APT
- instrukcja PPL - musi być użyta, jeżeli chcemy uzyskać rysunek opisywanej części lub jej fragmentu na autokreślance
- instrukcja KPL - odwołuje działanie instrukcji PPL.
- instrukcja END - kończy program

Program napisany w języku syntetyzera zostaje uruchomiony na EMC. Jeżeli odnalazł on błędy formalne, drukuje specjalne komunikaty (według własnej diagnostyki błędów), jeżeli nie - powoduje skompletowanie programu w języku APT i jego dalsze przetwarzanie aż do wykonania rysunku na autokreślance włącznie. Dalsze postępowanie z programem przebiega jak na uprzednio podanym schemacie (rys. 3).

W celu pokazania korzyści użycia syntetyzera ABST podajemy prosty przykład programu. Mamy do napisania program opisujący geometrię i obróbkę części pokazanej na rys. 4. Jest to typowy przy-



Rys. 4. Schemat żeberka do przykładowego programu

kład kołowa skrzydła samolotu. Program ten składa się z ok. 50 rereków w języku syntetyzera ABST. Wywołuje on z biblioteki trzy różne moduły, a mianowicie moduł opisujący profil wewnętrzny, moduł opisujący wybranie czteronarodne i moduł opisujący wybranie trzynarodne. Otwory okrągłe wykonywane są za pomocą jednej makroinstrukcji. Aby stworzyć program opisujący tę samą część w klasycznym systemie, tzn. bez użycia syntetyzera należałoby napisać ok. 200 instrukcji w języku APT, przy czym możliwość popełnienia błędu jest znacznie większa.

Obecna wersja syntetyzera ABST podlegać będzie dalszym modyfikacjom. Planowane jest stworzenie wersji konwersacyjnej. W wersji tej program będzie wprowadzany do EMC bezpośrednio z klawiatury display'a. Po wywołaniu kolejnego modułu jego rysunek będzie tworzony na monitorze graficznym i jeżeli uzyska się zgodność z życzeniami programisty będzie zapamiętany i dołączony do tworzonego programu. Jeżeli programista zgłosi zastrzeżenie będzie on modyfikowany lub skasowany i zastąpiony innym modułem.

W pierwotnym przeznaczeniu syntetyzer ABST miał służyć tylko generowaniu programów obróbki części. Jego stosowanie jest szczególnie opłacalne wtedy, gdy mamy wygenerować kilka programów opisujących części o podobnym kształcie. Opłacalne jest wtedy tworzenie szczegółowych bibliotek. Jeżeli różnice kształtu opisywanych części są duże, bardziej celowe jest tworzenie bibliotek ogólnych i dopisywanie fragmentów programu w języku APT. W trakcie budowy syntetyzera oraz tworzenia podstawowych bibliotek okazało się, że jego zastosowanie może być znacznie szersze, zwłaszcza w przemyśle lotniczym. Po niewielkiej modyfikacji bibliotek może on służyć tworzeniu dokumentacji plazowej samolotów<sup>M)</sup>. Tworzenie wspomnianej dokumentacji bez użycia techniki komputerowej jest bardzo pracochłonne i może prowadzić do powstawania niedokładności. Metody numeryczne i stosowanie omawianego syntetyzera proces ten skracają a jednocześnie zwiększają dokładność. Możliwe jest równocześnie tworzenie oprzyrządowania kontrolnego.

Syntetyzer ABST może być również pomocny w pracach konstrukcyjnych, zważywszy modułową budowę wielu wyrobów. W tego typu pracach szczególnie wygodna będzie jego wersja konwersacyjna. Stwarza ona konstruktorowi zupełnie nowe możliwości. Za pomocą monitora graficznego będzie on mógł w stosunkowo krótkim czasie zaprojektować kilka wariantów tworzonej konstrukcji i po analizie ich własności wybrać wersję optymalną.

Pełna dokumentacja syntetyzera ABST znajduje się w Archiwum Opracowań IMM nr 66/1978.

<sup>M)</sup> Dokumentacja plazowa - mówiąc bardzo ogólnie - jest to pełny zestaw dokumentów niezbędnych do wytworzenia samolotu. Rysunki w dokumentacji plazowej wykonane są na blachach.

319 Pg

## Automatyzacja projektowania przebiegu procesów technologicznych obróbki detali

### Wprowadzenie

Przy opracowywaniu technologicznych planów obróbki pojedynczych detali lub klasy detali wielo znaczenie ma zautomatyzowanie projektowania przebiegu procesów technologicznych obróbki detali. Można to realizować w następujący sposób. Po pierwsze, przeprowadzić klasyfikację detali według pewnego wybranego przez technologa kryterium. Celowe jest również sformalizowanie zagadnień związanych z przebiegiem procesu technologicznego obróbki. I wreszcie przystosowanie, otrzymanych w wyniku takiej formalizacji, metod tworzenia przebiegu na maszynę cyfrową.

Przy opracowywaniu przebiegu procesu obróbki podstawowym zadaniem jest zawsze ustalenie kolejności operacji, którą zadaje technolog, kierując się przy tym własnym doświadczeniem. Ale zastosowanie maszyn cyfrowych pozwala opracowywać typowe i grupowe procesy obróbcze o lepszej funkcjonalności.

Artykuł zawiera opis formalizacji podstawowych pojęć dotyczących przebiegu procesu technologicznego obróbki detalu, która umożliwi wyprowadzenie wielu wniosków przydatnych w procesie automatyzacji projektowania przebiegu procesów technologicznych obróbki detali. Pokazano też sposób (algorytm) tworzenia uogólnionego przebiegu procesu obróbki dla klasy detali oraz algorytm procesu odwrotnego, czyli wydzielenia indywidualnego przebiegu procesu obróbki konkretnego detalu z przebiegu uogólnionego. Następnie opisano konkretne programy utworzone na podstawie uprzednio podanych algorytmów.

W dużej mierze opracowanie jest oparte na pozycji literaturowej [2], która posłużyła do pewnych modyfikacji i uzupełnień dotyczących formalnego podejścia do spraw związanych z przebiegiem procesów technologicznych obróbki detali. Algorytmy zostały częściowo zmodyfikowane (dotyczy to szczegółów). Natomiast programy napisano tymi metodami są całkowicie własne. Matematyczne podstawy formalnego podejścia można znaleźć w książce K. Kuratowskiego [3]. Natomiast pozycję [1] posłużyła do ujęcia omawianych zagadnień w szerszym aspekcie.

### Definicja przebiegu procesu technologicznego obróbki detali

Procesy technologiczne opracowane na podstawie typowego procesu (zob. punkt "Uogólniony przebieg procesu obróbki klasy detali") charakteryzują się tym, że wszystkie równoważne operacje formułują się jednakowo. Daje to możliwość typowych opisów obróbki powierzchniami lub liniami powierzchni. Operacje obróbki skrawaniem można sklasyfikować i zakodować. Niech kod operacji zawiera 4 znaki cyfrowe (podobnie, jak to przyjął Kapustin [2], może być inny sposób kodowania), pierwsze dwa charakteryzują rodzaj operacji obróbki, następane dwa - okoliczności wykonania operacji, np. kody operacji frezowania zaczynają się od znaków 21, ale

2101 - oznacza operację frezowania w imadle rowków wpustowych według rysunku,  
2102 - frezowanie w pryzmach dwóch segmentowych rowków wpustowych według rysunku,  
2103 - frezowanie w imadle sumocentrycznym rowków wpustowych według rysunku.

Warunki występowania operacji można rozdzielić na wiele grup, m.in.:

- warunki określające konieczność obróbki niektórych specyficznych powierzchni, występujących nie we wszystkich detalach; odnoszą się one do konstrukcyjno-technologicznych cech obrabianych detali;
- warunki określające niedokładności wymiarów, kształtu i wzajemnego położenia powierzchni;
- warunki określające rodzaj powierzchni detali;

- warunki charakteryzujące ohropowatódóó obrabianych powierzchni;
- warunki okreólaające gabaryty detalu.

Poniżej bęódzie przedstawione pewne formalne ujęócie zagadnień związanych z projektowaniem przebiegu procesów technologicznych obróóki detali. Jak to zaznaczono we wstępie, ujęócie to posłuży jako podstawa do opracowania algorytmów projektowania przebiegu procesu i realizacji programów.

Niech dany bęódzie pewien detal. Niech  $C$  bęódzie skończonym zbiorem kodów operacji wystęópujących w procesie obróóki danego detalu oraz  $n$  - liczbą wszystkich operacji obróóki detalu. Oznacza się przez  $o^l$  kod  $l$ -tej operacji w procesie obróóki detalu, gdzie  $l = 1, \dots, n$ .

Przyjmuje się, że dla  $o^p, o^q \in C$

$$o^p = o^q \Leftrightarrow p = q \quad (1)$$

(identycznoóó operacji) oraz

$$o^p \approx o^q \Leftrightarrow \text{kody } p\text{-tej i } q\text{-tej operacji s równe} \quad (2)$$

(róównoważnoóó operacji).

Dla dowolnej operacji  $o \in C$  istnieje co najmniej jeden wskaźnik  $1 \leq p \leq n$  taki, że  $o = o^p$ , czyli

$$\bigvee_{o \in C} \bigexists_{1 \leq p \leq n} o = o^p \quad (3)$$

W zbiorze  $C$  można zadać porządek w nastęópujący sposób:

$o', o'' \in C$ ,  $o' \prec o'' \Leftrightarrow$  operacja z kodem  $o'$  wystęópuje w procesie obróóki danego detalu przed operacją z kodem  $o''$ , niekoniecznie bezpośrednio. (4)

Jeóóli  $o', o'' \in C$ , to z (3) wynika, że istnieją wskaźniki  $p, q$ , gdzie  $1 \leq p, q \leq n$ , takie, że  $o' = o^p$ ,  $o'' = o^q$ . Zatem z (4) wynika, że

$$o' \prec o'' \Leftrightarrow o^p \prec o^q \Leftrightarrow p < q \quad (5)$$

Wynika stąd, że relacja  $\prec$ , podobnie jak relacja mniejszoóóci  $<$ , jest relacją porządkując, tzn. spólnia wlasnoóóci antysymetrii, przechodnioóóci i spójnoóóci.

W wyniku uporządkowania zbioru  $C$  przez relację  $\prec$ , ze zbioru  $C$  otrzymuje się cig kolejnych operacji wchodzących w skłd procesu obróóki danego detalu:

$$C_1 = (o^1, o^2, \dots, o^n) \quad (6)$$

Oznacza się przez  $\tilde{C}$  zbiór wszystkich wyrazów cigu  $C_1$ .

Niech  $A = \{A_r\}_{r=1}^n$  bęódzie zbiorem róónych funkcji zdaniowych (warunków) charakterystycznych dla wszystkich operacji procesu obróóki danego detalu. Oznacza się

$$A(1) = \{A_r(1)\}_{r=1}^n, \quad 1 = 1, \dots, n$$

jako zbiór kodów wystpienia warunków  $A_r$ ,  $r = 1, \dots, n$  w  $l$ -tej operacji.

Niech  $\psi$  bęódzie funkcją okreólon wzorem

$$\begin{aligned} \psi : A \times \tilde{C} &\longrightarrow \{A(1)\}_{1=1}^n \\ \psi(A_r, o^1) &= A_r(1), \end{aligned} \quad (7)$$

dla  $A_r \in A$  i  $o^1 \in \tilde{C}$ .

Funkcja  $\psi$  jest dobrze okreólon, gdyż dla  $1 \leq r_1, r_2 \leq n$  i  $1 \leq p, q \leq n$ :

$$(A_{r_1}, o^p) = (A_{r_2}, o^q) \Rightarrow A_{r_1} = A_{r_2} \wedge o^p = o^q \Rightarrow r_1 = r_2 \wedge p = q \Rightarrow \psi(A_{r_1}, o^p) = A_{r_1}(p) = A_{r_2}(q) = \psi(A_{r_2}, o^q) \quad (8)$$



Kod wystąpienia warunku  $\Lambda_r$  w 1-tej operacji,  $\Lambda_r(1)$ , przeważnie przyjmuje się jako pewną liczbę do dwóch cyfr.  $\Lambda_r(1) = 0$  oznacza, że 1-ta operacja występuje bez spełnienia warunku  $\Lambda_r$ . Oznacza się dla  $1 = 1, \dots, n$  przez

$$r^1 = \bigwedge_{r=1}^n \Lambda_r(1) \quad (9)$$

zakodowany warunek wystąpienia 1-tej operacji jako koniunkcję poszczególnych warunków  $\Lambda_r(1)$ ,  $r = 1, \dots, n$ . Przy tym jeśli wszystkie  $\Lambda_r(1) = 0$ , gdzie  $r = 1, \dots, n$ , to przyjmuje się  $r^1 = 0$ . I odwrotnie.

Niech  $\tilde{F} = \{r^1, \dots, r^n\}$  oraz

$$\phi: \tilde{C} \rightarrow \tilde{F}, \quad \phi(o^1) = r^1, \quad 1 = 1, \dots, n. \quad (10)$$

$\phi$  jest funkcją dobrze określoną, gdyż dla  $1 \leq p, q \leq n$ :

$$\begin{aligned} o^p = o^q &\Rightarrow p = q \Rightarrow \phi(o^p) = r^p = \bigwedge_{r=1}^n \Lambda_r(p) = \\ &= \bigwedge_{r=1}^n \Lambda_r(q) = r^q = \phi(o^q) \end{aligned} \quad (11)$$

W ten sposób otrzymuje się ciąg

$$F_A = (r^1, r^2, \dots, r^n) \quad (12)$$

Za przebieg procesu technologicznego obróbki danego detalu uważa się:

$$(C_j, F_A) \phi \quad (13)$$

Jest on rozumiany jako ciąg par  $\{(o^1, r^1)\}_{1=1}^n$ , tzn.

$$((o^1, r^1), (o^2, r^2), \dots, (o^n, r^n)) \quad (14)$$

Dalej podany jest przykład konkretnego przebiegu procesu technologicznego obróbki detalu w postaci tabeli 1.

Poniższa tabela przebiegu procesu technologicznego obróbki detalu będącego wałem stopniowym, zawiera kody kolejnych operacji obróbki (elementy ciągu  $C_j$ ), kody wystąpienia warunków  $\Lambda_1, \Lambda_2, \Lambda_3$  dla każdej operacji, zakodowane warunki wystąpienia każdej operacji (elementy ciągu  $F_A$ ), a także treść każdej z operacji obróbki. Przy tym treść warunków jest następująca:

- $\Lambda_1$  - w skład zawartości partii wchodzi ponad  $x_1$  sztuk,
- $\Lambda_2$  - liczba stopni jest większa od  $x_2$ ,
- $\Lambda_3$  - występuje zgrubienie wału po środku.

Warunki występowania operacji są zakodowane za pomocą dwóch znaków cyfrowych w ten sposób, aby odróżnić od siebie różne okoliczności wykonania tej samej operacji, zarówno w obrębie jednego, jak i wielu przebiegów procesów obróbki.

Warunki wykonania operacji składają się na zakodowany warunek wystąpienia tej operacji, który jest koniunkcją tych warunków, np. przy wykonaniu 7 operacji oznaczonej kodem  $o^7 = 1601$  spełnione są warunki  $\Lambda_1, \Lambda_2$  i  $\Lambda_3$  oznaczone odpowiednio kodami 0, 18 i 44, tzn.

$$\Lambda_1(7) = 0, \Lambda_2(7) = 18, \Lambda_3(7) = 44 \quad (15)$$

wówczas, zgodnie z wzorem (9) można zapisać

$$r^7 = \bigwedge_{r=1}^3 \Lambda_r(7) = \Lambda_1(7) \wedge \Lambda_2(7) \wedge \Lambda_3(7) = 0 \wedge 18 \wedge 44 = 18 \wedge 44 \quad (16)$$

Przebieg procesu technologicznego obróbki wału stopniowego

Lp.	Warunki (kody) występowania operacji			Kody operacji	Zakodowano warunki występowania operacji	Treść operacji
	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>			
1	2	3	4	5	6	7
1	0	0	0	2001	0	Obcięcie materiału na wymiar
2	0	0	14	1105	14	Operacja tokarska - w uchwycie toczyć prawą powierzchnię ozolową wg szkicu
3	0	0	11	1106	11	Operacja tokarska - w uchwycie podoląć lewą powierzchnię ozolową, wywiercić nakielki, stępić ostre krawędzie
4	0	0	09	1107	09	Operacja tokarska - w uchwycie z podparciem w kle toczyć wzdłuż kopiału prawą stronę detalu wg szkicu
5	0	0	12	1110	12	Operacja tokarska - w uchwycie z podparciem w kle toczyć wzdłuż kopiału lewą stronę detalu wg szkicu
6	0	0	32	1111	32	Operacja tokarska - w kłach toczyć rowki i fazy wg rysunku, stępić ostro krawędzie
7	0	18	44	1601	18 A 44	Operacja szlifowania na okrągło - w kłach szlifować dwie szyjki wg szkicu, powierzchnie czolowych nie obrabiać
8	0	0	17	2602	17	Operacja frezowania - trasować, w przyrządach ustawić detal wzdłuż szyjek, frezować rowek wpustowy wg rysunku
9	0	0	04	1112	04	Operacja tokarska - w kłach naciąć gwint (gwintować) z wykończeniem wg rysunku
10	0	0	33	2103	33	Frezowanie w imadle sumocentryującym rowków wpustowych wg rysunku
11	0	0	0	0201	0	Operacja ślusarska - usunąć zadziory, stępić ostre krawędzie
12	0	16	41	1602	16 A 41	Operacja szlifowania na okrągło - w kłach szlifować wykańczając z wybiegiem i podszlifować powierzchnie czolowe wg rysunku
13	0	0	0	0201	0	Operacja ślusarska - ostrze końce stępić, usunąć zadziory
14	0	0	0	0501	0	Operacja occhowania - occhować numer detalu

1	2	3	4	5	6	7
15	0	0	0	0001	0	Kontrola wszystkich parametrów, zgodnie z warunkami technicznymi
16	0	0	06	0402	06	Oksydowanie
17	0	0	47	0301	47	Konserwacja wg instrukcji

#### Uogólniony przebieg procesu obróbki klasy detali

W przemyśle maszynowym wykorzystuje się ogromną liczbę detali o typowych kształtach, dla których opracowywane są typowe procesy technologiczne.

Istnieją różne metody typizacji procesów technologicznych. Bazują one na klasyfikacji detali. Detale jednego typu charakteryzują się wspólnymi procesami obróbki powierzchni podstawowych. Typowy proces opracowuje się dla typowego detalu, mającego największą liczbę charakterystycznych szczegółów, bądź też dla ogółu detali danej klasy. W tym drugim przypadku typowy proces obejmuje wszystkie operacje charakterystyczne dla danego typu detali, a nie tylko operacje obróbki powierzchni podstawowych.

Zgodnie z tradycyjnymi metodami projektowania, do jednego typu zalicza się detale zbliżone do siebie pod względem kształtu, wymiarów i dokładności. Przebieg ich procesów technologicznych różni się nieznacznie, głównie operacjami drugorzędnymi. Colowe jest połączenie kilku procesów typowych w jeden ogólniejszy, obejmujący całą grupę lub klasę detali.

Kierując się pewnym zadanym kryterium, dany detal można zaliczyć do pewnej klasy, np. wały, dyski, korpusy i in. W zależności od warunków dany detal może być zaliczony do grupy, podgrupy, czy typu detali zgodnie z kryterium klasyfikacji.

Dla danej klasy (grupy) detali określa się tzw. uogólniony przebieg procesu obróbki. Uogólniony przebieg zawiera ogół operacji obróbki charakterystycznych dla określonej klasy detali. Jest on uporządkowany i przedstawia zbiór operacji istniejących dróg indywidualnych.

Niech  $\mathcal{K}$  będzie daną klasą (grupą)  $n$  detali,  $M_1, \dots, M_n$  - indywidualne przebiegi procesów technologicznych obróbki detali z klasy  $\mathcal{K}$ .

Wówczas dla  $i=1, \dots, n$  na podstawie (13)

$$M_i = (C_i, F_i) \phi_i \quad (17)$$

gdzie analogicznie do (6)

$$C_i = (c_{i1}^1, \dots, c_{i1}^{n_1}) \quad (18)$$

jest ciągiem kodów kolejnych operacji przebiegu  $M_i$ , analogicznie do (12)

$$F_i = (f_{i1}^1, f_{i1}^2, \dots, f_{i1}^{n_1}) \quad (19)$$

jest ciągiem kodów warunków wystąpienia kolejnych operacji, przy tym dla  $l=1, 2, \dots, n_1$ , analogicznie do (9)

$$f_{i1}^l = \bigwedge_{r=1}^{s_1} \Lambda_r^l(1) \quad (20)$$

gdzie  $\Lambda^i = \left\{ \Lambda_r^i \right\}_{r=1}^{s_1}$  jest zbiorem różnych funkcji zdaniowych (warunków) charakterystycznych dla wszystkich operacji przebiegu  $M_i$  oraz  $\Lambda^i(1) = \left\{ \Lambda_r^i(1) \right\}_{r=1}^{s_1}$  dla  $l=1, 2, \dots, n_1$  jest zbiorem kodów wystąpienia warunków  $\Lambda_r^i$ ,  $r=1, \dots, s_1$  w  $l$ -tej operacji przebiegu  $M_i$ . Analogicznie do (10) funkcję  $\phi_i$  definiuje się następująco:

$$\phi_1 : \tilde{C}_1 \rightarrow \tilde{F}_1, \quad \phi_1(o_1^1) = f_1^1, \quad 1 = 1, 2, \dots, n_1 \quad (21)$$

gdzie  $\tilde{C}_1$  jest zbiorem wszystkich wyrazów ciągu  $C_1$  oraz  $\tilde{F}_1$  jest zbiorem wszystkich wyrazów ciągu  $F_1$ . Wszystkie podane wyżej wyjaśnienia odnoszą się do  $i = 1, \dots, n$ .

Oznacza się przez  $M_{\text{przec}} = \bigcap_{i=1}^n M_i$ , symboliczne przecięcie wszystkich przebiegów indywidualnych (w sensie przecięcia  $\bigcap_{i=1}^n \tilde{C}_i$ ) detali klasy  $\mathcal{K}$ . Moc przecięcia  $|M_{\text{przec}}|$  jest liczbą równoważnych operacji, występujących we wszystkich przebiegach indywidualnych detali klasy  $\mathcal{K}$ .

Jeśli  $M_u^*$  będzie oznaczać uogólniony przebieg procesu obróbki detali klasy  $\mathcal{K}$ , to moc tego przebiegu  $|M_u^*|$  będzie liczbą wszystkich operacji, występujących we wszystkich przebiegach indywidualnych detali klasy  $\mathcal{K}$ .

Ważnym kryterium efektywności formułowania uogólnionego przebiegu jest moc przecięcia, która powinna dążyć do maksimum:

$$|M_{\text{przec}}| \rightarrow \max \quad (22)$$

Wówczas moc uogólnionego przebiegu będzie dążyć do minimum:

$$|M_u^*| \rightarrow \min \quad (23)$$

Uogólniony przebieg przedstawia się jako zgromadzenie przecinających się przebiegów indywidualnych. Liczby  $|M_{\text{przec}}|$  i  $|M_u^*|$  pozwalają ocenić połączenie detali w grupę.

Niech  $\Lambda = \{ \Lambda_r \}_{r=1}^n$  będzie zbiorem (wykazem) warunków dla detali klasy  $\mathcal{K}$ .

Zatem uogólniony przebieg procesu technologicznego obróbki detali z klasy  $\mathcal{K}$ , to (na podstawie definicji (13)):

$$M_u^* = (C_u^*, F_u^*) \phi_u^* \quad (24)$$

gdzie analogicznie do (6), (12), (10):

$$C_u^* = (c^1, c^2, \dots, c^{n_u}) \quad (25)$$

jest ciągiem kodów kolejnych operacji drogi uogólnionej,

$$F_u^* = (f^1, f^2, \dots, f^{n_u}) \quad (26)$$

jest ciągiem kodów warunków wystąpienia kolejnych operacji uogólnionego przebiegu oraz określona jest funkcja

$$\phi_u^* : \tilde{C}_u^* \rightarrow \tilde{F}_u^*, \quad \phi_u^*(o^1) = f^1, \quad 1=1,2,\dots,n_u \quad (27)$$

gdzie  $\tilde{C}_u^*$  jest zbiorem wszystkich wyrazów ciągu  $C_u^*$  i  $\tilde{F}_u^*$  jest zbiorem wszystkich wyrazów ciągu  $F_u^*$ .

Przy tym

$$\Lambda = \{ \Lambda_r \}_{r=1}^n = \bigcup_{i=1}^n \{ \Lambda_r^i \}_{r=1}^{s_i} = \bigcup_{i=1}^n \Lambda_i \quad (28)$$

$$\tilde{C}_u^* \subseteq \bigcup_{i=1}^n \tilde{C}_i \quad (29)$$

$$\tilde{F}_u^* \subseteq \bigcup_{i=1}^n \tilde{F}_i \quad (30)$$

Niech dla  $l = 1, 2, \dots, n_u$

$$r^l = \bigwedge_{j=1}^n \bigwedge_j^l = \bigwedge_1^l \vee \dots \vee \bigwedge_j^l \vee \dots \vee \bigwedge_n^l \quad (31)$$

Jeśli  $l$ -ta operacja przebiegu uogólnionego występuje także w  $j$ -tym przebiegu indywidualnym  $M_j$ , jako  $k$ -ta operacja, to wówczas

$$\bigwedge_j^l = r_j^k \quad (32)$$

Jeśli natomiast  $l$ -ta operacja przebiegu uogólnionego nie występuje w przebiegu  $M_j$ , to przyjmuje się, że

$$\bigwedge_j^l = \emptyset \quad (33)$$

( $\emptyset$  - symbol, który można pominąć w alternatywie warunków).

Zatem

$$\bigwedge_j^l = \begin{cases} r_j^k, & \text{jeśli } o^l \equiv o_j^k \\ \emptyset, & \text{jeśli } o^l \not\equiv o_j^k \end{cases} \quad (34)$$

dla  $l=1, \dots, n_u$  oraz  $j=1, \dots, n$ .  $\bigwedge_j^l$  nazywa się  $j$ -tym zestawem warunków występowania  $l$ -tej operacji uogólnionego przebiegu.

Zatem dokładnie jeden ze zbioru  $n$  zestawów (34) ( $j=1, \dots, n$ ) pozwala jednoznacznie wybrać operację dla przebiegu indywidualnego ( $M_j$ ) i umieścić ją w odpowiednim miejscu (z numerem  $k$ ).

Jeśli  $r^l$  jest alternatywą  $n$  jednakowych zestawów, oznacza to, że dana  $l$ -ta operacja przebiegu uogólnionego występuje we wszystkich indywidualnych przebiegach procesów technologicznych obróbki dla klasy  $\mathcal{K}$ .

Przy formułowaniu uogólnionego przebiegu procesu obróbki klasy detali należy uwzględnić następujące warunki:

- z dwóch operacji równoważnych jedna wchodzi do przebiegu uogólnionego, czyli

$$o_1^p \in \tilde{C}_1, o_j^q \in \tilde{C}_j \quad (o_1^p \equiv o_j^q \Rightarrow o_1^p \in \tilde{C}_u^*) \quad (35)$$

- dla dowolnych dwóch operacji z kodami  $o_1^p, o_1^q$  wchodzących w skład przecięcia  $\tilde{C}_1, \tilde{C}_j$ , jeśli operacja  $o_1^p$  poprzedza operację  $o_1^q$  w przebiegu  $M_1$ , to operacja  $o_1^p$  poprzedza operację  $o_1^q$  w przebiegu  $M_j$ , czyli

$$o_1^p, o_1^q \in \tilde{C}_1 \cap \tilde{C}_j \quad (o_1^p \prec_{M_1} o_1^q \Rightarrow o_1^p \prec_{M_j} o_1^q) \quad (36)$$

- dla dowolnego przebiegu indywidualnego  $M_1$  i dla dowolnych operacji z kodami  $o_1^p, o_1^q$  występujących w tym przebiegu, jeśli  $o_1^p \prec_{M_1} o_1^q$ , to  $o_1^p \prec_{M_u^*} o_1^q$  w przebiegu uogólnionym  $M_u^*$ , czyli

$$\bigwedge_{M_1} o_1^p, o_1^q \in \tilde{C}_1 \quad (o_1^p \prec_{M_1} o_1^q \Rightarrow o_1^p \prec_{M_u^*} o_1^q) \quad (37)$$

Przedstawiony formalny opis uogólnionego przebiegu procesu obróbki detali daje podstawy opracowania odpowiednich algorytmów tworzenia takich przebiegów dla dowolnej klasy detali. W następnym punkcie będzie przedstawiony najpierw proces tworzenia przebiegu uogólnionego dla pewnej konkretnej klasy detali, a następnie algorytm takiego postępowania, który to algorytm obowiązuje dla bardzo różnych klas detali.

Programowa realizacja budowy przebiegu uogólnionego

Niech będzie dana klasa  $\mathcal{K}$  detali, a  $M_1, \dots, M_n$  - będą indywidualnymi przebiegami procesów obróbki tych detali.

Proces formowania  $M_u^*$  sprowadza się do powtórzenia  $(n-1)$  razy algorytmu, polegającego na łączeniu aktualnego przebiegu uogólnionego (przebiegu bazowego) z następnym przebiegiem indywidualnym (przebiegiem przyłączanym), na przykład

$$\begin{array}{l} M_1, \quad M_2 \rightarrow M_{12}^* \\ M_{12}^*, \quad M_3 \rightarrow M_{123}^* \\ \dots\dots\dots \\ M_{12\dots(n-1)}^*, \quad M_n \rightarrow M_u^* \end{array}$$

W pojedynczym kroku za przebieg bazowy można przyjąć dowolny przebieg indywidualny spośród przebiegów  $M_1, \dots, M_n$ , ale lepiej przyjąć przebieg, zawierający największą liczbę operacji. Przebieg przyłączany może być każdym innym przebiegiem indywidualnym.

Zadanie polega na tym, żeby przyłączyć dany przebieg, zwany przyłączanym do przebiegu bazowego. Przy tym konieczne jest, żeby otrzymany przebieg miał możliwie najmniejszą długość, tzn. zawierał jak najmniejszą liczbę operacji. Należy utożsamiać operacje równoważne i zachować ostrą kolejność operacji (zob. punkt "Uogólniony przebieg procesu obróbki klasy detali").

Niżej podany będzie przykład budowy uogólnionego przebiegu procesu obróbki klasy wałów stopniowych. Rozpatrywano będą dwie odmiany detali z tej klasy. Za przebieg bazowy przyjmuje się przebieg podany w tabeli 1 i oznacza się go przez  $M_1$ . Przebieg przyłączany  $M_2$  podaje tabela 2.

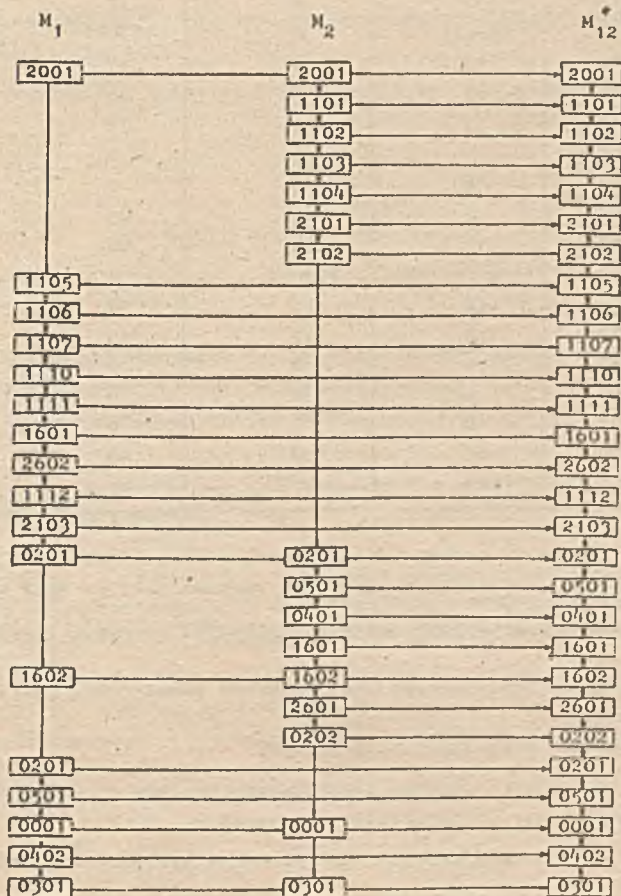
Tabela 2.

Przebieg przyłączany obróbki wału stopniowego w procesie budowy przebiegu uogólnionego

Lp.	Warunki (kody) występowania operacji			Kody operacji	Zakodowane warunki występowania operacji	Treść operacji
	$\Lambda_1$	$\Lambda_2$	$\Lambda_3$			
1	2	3	4	5	6	7
1	0	0	0	2001	0	Obcięcie materiału na wymiar
2	0	0	13	1101	13	Operacja tokarska - w uchwycie splanować powierzchnię czołową na wymiar, nakiełkować z dwóch stron wg szkicu
3	0	0	21	1102	21	Operacja tokarska - w kłach toczyć prawą stronę wg szkicu
4	0	0	27	1103	27	Operacja tokarska - w kłach toczyć lewą stronę wg szkicu
5	0	0	08	1104	08	Operacja tokarska - w kłach nucić gwint z wykończeniem z dwóch stron wg szkicu
6	0	0	0	2101	0	Frezowanie w imadle rowka wpustowego wg rysunku
7	0	0	34	2102	34	Frezowanie w pryzmach dwóch segmentowych rowków wpustowych wg rysunku
8	0	0	0	0201	0	Operacja ślusarska - usunąć zadziory, stępić ostre krawędzie
9	0	0	0	0501	0	Operacja ociekowania - ociekować numer detalu
10	0	0	05	0401	05	Obróbka termiczna

1	2	3	4	5	6	7
11	0	0	40	1601	40	Operacja szlifowania na okrągło - w kłach szlifować wstępnie wg rysunku
12	0	16	41	1602	16 $\wedge$ 41	Operacja szlifowania na okrągło - w kłach szlifować wykańczająco z wybiogiem i podszlifować powierzchnie czołowe wg rysunku
13	0	0	18	2601	18	Operacja wiercenia - trasować na stole ze sprawdzaniem, wiercić wg szkicu
14	0	0	53	0202	53	Operacja ślusarska - naciąg gwint ( gwintować ) wg rysunku
15	0	0	0	0001	0	Kontrola wszystkich parametrów zgodnie z warunkami technicznymi
16	0	0	47	0301	47	Konserwacja wg rysunku

Realizację łączenia przebiegów indywidualnych  $M_1$  i  $M_2$  w przebieg uogólniony pokazuje schemat 1:



Rys. 1. Schemat formowania uogólnionego przebiegu procesu obróbki

W wyniku tej realizacji otrzymuje się ciąg kolejnych operacji przebiegu uogólnionego, który w zestawieniu z warunkami występowania każdej z tych operacji pokazuje tab. 3.

Uogólniony przebieg procesu obróbki dla dwóch odmian walu stopniowego

Lp.	Kody operacji	Pierwszy zestaw warunków występowania operacji	Drugi zestaw warunków występowania operacji	Zakodowany warunek występowania operacji
1	2	3	4	5
1	2001	0	0	0
2	1101	∅	13	13
3	1102	∅	21	21
4	1103	∅	27	27
5	1104	∅	08	08
6	2101	∅	0	0
7	2102	∅	34	34
8	1105	14	∅	14
9	1106	11	∅	11
10	1107	09	∅	09
11	1110	12	∅	12
12	1111	32	∅	32
13	1601	18 ∧ 44	∅	18 ∧ 44
14	2602	17	∅	17
15	1112	04	∅	04
16	2103	33	∅	33
17	0201	∅	0	0
18	0501	∅	0	0
19	0401	∅	05	05
20	1601	∅	40	40
21	1602	16 ∧ 41	16 ∧ 41	16 ∧ 41
22	2601	∅	18	18
23	0202	∅	53	53
24	0201	0	∅	0
25	0501	0	∅	0
26	0001	0	0	0
27	0402	06	∅	06
28	0301	47	47	47

Przedstawiony proces jest ilustracją metody postępowania, według której opracowano program na maszynę cyfrową. W omawianej sytuacji metoda postępowania rozbita jest na trzy kroki:

- program podstawowy - znajdowanie w przebiegu bazowym dla każdej operacji przebiegu przyłączonego operacji analogicznych,
- program rozmieszczenia - rozmieszczenie operacji w przebiegu uogólnionym (otrzymanie tzw. optymalnego sumarycznego przebiegu),
- program zestawienia - zestawianie przebiegu uogólnionego.

Na maszynie cyfrowej IBM 370 w języku FORTRAN zrealizowano program o nazwie BUP budowy uogólnionego przebiegu obróbki klasy detali. Czas realizacji wynosi ok. 30 s.

Danymi wejściowymi są kody kolejnych operacji i ich warunków występowania przebiegów: bazowego i przyłączonego.

Opis oznaczeń:

1.  $C(I)$ , gdzie  $I=1, \dots, N$  - tablica  $1 \times N$  kodów kolejnych operacji przebiegu bazowego  $M$ ,
2.  $F(I)$ , gdzie  $I=1, \dots, N$  - tablica  $1 \times N$  kodów warunków występowania odpowiednich operacji przebiegu bazowego  $M$ ,



3.  $C1(J)$ , gdzie  $J=1, \dots, N1$  - tablica  $1 \times N1$  kodów kolejnych operacji przebiegu przyłączanego  $M1$ ,
4.  $F1(J)$ , gdzie  $J=1, \dots, N1$  - tablica  $1 \times N1$  kodów warunków występowania odpowiednich operacji przebiegu przyłączanego  $M1$ .

Opis oznaczeń występujących w programie BUP:

5.  $NN(J, K)$   $J=1, \dots, N1; K=1, \dots, N$   
- tablica powtórzeń operacji przebiegu  $M1$  w przebiegu  $M$ ;  
 $NN(J, K) = I$  oznacza, że  $J$ -ta operacja przebiegu  $M1$  występuje  $K$ -ty raz w przebiegu  $M$  jako  $I$ -ta operacja;
6.  $KPSD(J)$   $J=1, \dots, N1$   
- sumaryczny przebieg dla  $K$ -tej kolumny tablicy  $NN$ ;
7.  $NNPSD(K, J)$   $K=1, \dots, N; J=1, \dots, N1$   
- tablica wszystkich przebiegów sumarycznych dla każdej kolumny tablicy  $NN$ ;

$$8. LP = \max_{K=1, \dots, N} (LL(K))$$

gdzie  $LL(K)$  dla  $K=1, \dots, N$  jest liczbą różnych od zera wyrazów ciągu  $NNPSD(K, J)$ ,  $J=1, \dots, N1$ ;

9.  $NNOSD(J)$   $J=1, \dots, N1$   
- optymalny sumaryczny przebieg dla tablicy  $NN$ , tzn.  $NNOSD(J) = NNPSD(K, J)$  dla pewnego  $K=1, \dots, N$  takiego, że  $LL(K) = LP$

Wynikiem realizacji programu jest przebieg uogólniony  $MU$  otrzymany z tych przebiegów  $M$  i  $M1$ , przy czym:

10.  $CU(K)$ , gdzie  $K=1, \dots, NU$  - tablica  $1 \times NU$  kodów kolejnych operacji przebiegu uogólnionego  $MU$ ,
11.  $FU(K)$ , gdzie  $K=1, \dots, NU$  - tablica  $1 \times NU$  kodów warunków występowania odpowiednich operacji przebiegu uogólnionego  $MU$ .

Działanie programu przedstawia bardzo ogólnie schemat blokowy z rys. 2.

Natomiast schemat z rys. 3 przedstawia w ogólny sposób działanie programu rozmieszczenia.

Przedstawiona tu za pomocą schematów 2 - 3 metoda postępowania opiera się na algorytmie przedstawionym w pracy [2] w ogólnym zarysie. Natomiast szczegółowe rozpracowanie każdego z trzech kroków tej metody zostało niemal całkowicie zmienione i zmodyfikowane.

#### Wydzielanie technologicznych przebiegów indywidualnych z uogólnionego przebiegu procesu obróbki klasy detali oraz realizacja programu

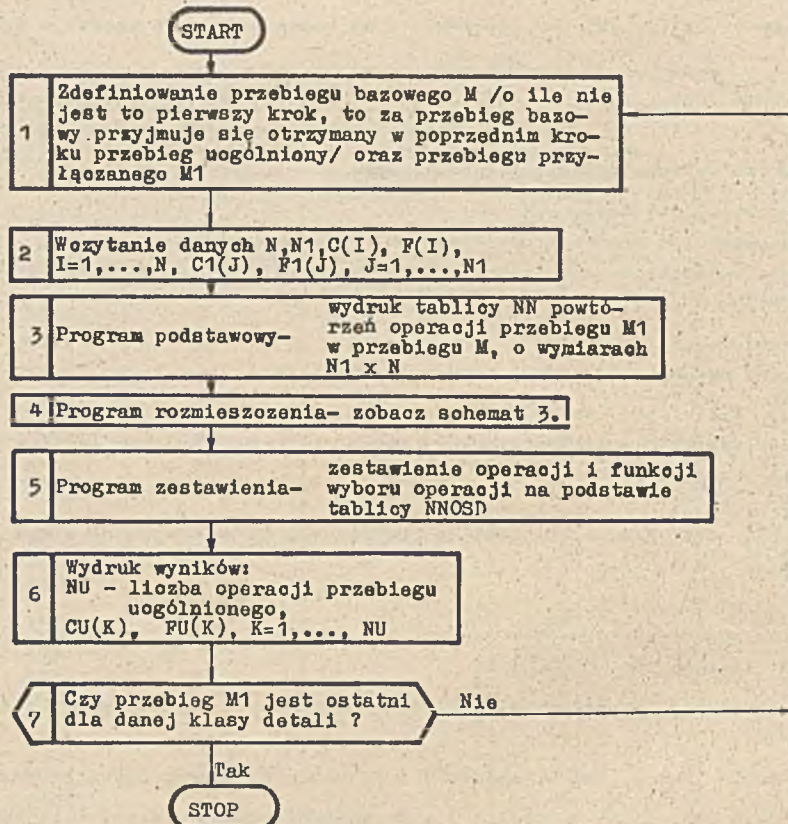
Niech dana będzie klasa  $n$  detali  $\mathcal{K}$ , dla której przebiegiem uogólnionym jest  $M_u^*$ . Na podstawie kodów warunków charakterystycznych dla konkretnego detalu z klasy  $\mathcal{K}$ , można z przebiegu uogólnionego wydzielić indywidualny przebieg procesu obróbki tego detalu (ściślej, kody kolejnych operacji procesu obróbki).

Przebieg uogólniony  $M_u^*$  zawiera kody warunków wystąpienia każdej jego operacji  $r^1$ , gdzie  $l=1, \dots, n_u$ ,  $n_u$  - liczba operacji w przebiegu uogólnionym, zadanej wzorem (31). Niech  $\bigwedge_j^1$  oznacza  $j$ -ty zestaw warunków ( $j=1, \dots, n$ ) występowania  $l$ -tej operacji przebiegu uogólnionego ( $l=1, \dots, n_u$ ), określony za pomocą wzoru (34). Niech dla  $j=1, \dots, n$

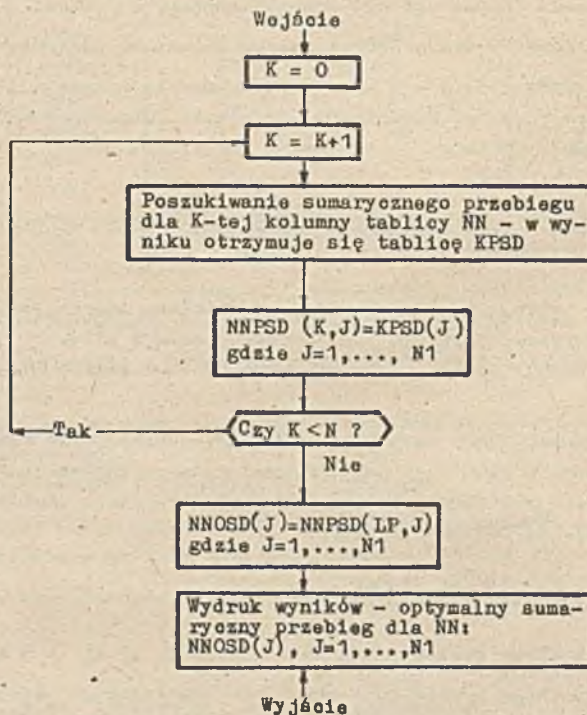
$$\bigwedge_j = \left\{ \bigwedge_{l=1}^{n_u} \right\} = \left( \bigwedge_{l=1}^1, \dots, \bigwedge_{l=1}^{n_u} \right) \quad (38)$$

Na przykład ciąg  $\bigwedge_1$  przedstawiony jest w 3 kolumnie tabeli 3, a  $\bigwedge_2$  w 4 kolumnie tej tabeli.

Metoda wydzielenia indywidualnego przebiegu z przebiegu uogólnionego wygląda następująco. Bierze się  $j$ -ty detal z klasy  $\mathcal{K}$  ( $j=1, \dots, n$ ). Na mocy wzoru (38) określony jest ciąg  $\bigwedge_j$ . Dla każdego  $l=1, \dots, n_u$ , jeśli  $\bigwedge_j^l \neq \emptyset$ , to przyjmuje się jako kolejną operację przebiegu  $M_j$  kod



Rys. 2. Schemat blokowy działania programu BUP



Rys. 3. Schemat blokowy działania programu rozmieszczenia (blok 4 schematu 2)

1-tej operacji  $\alpha^1$  przebiegu uogólnionego; jeśli natomiast  $\bigwedge_j^1 = \emptyset$ , to bada się następnie  $\bigwedge_j^1$ .

Na maszynie RIAD-32 w języku FORTRAN zrealizowano program WTPI wydzielenia technologicznych przebiegów indywidualnych z uogólnionego przebiegu procesu obróbki klasy detali. Na schemacie 4 pokazane jest działanie tego programu. Czas jego realizacji wynosił ok. 9 s.

Danymi wejściowymi do programu są:

$$J \quad (39)$$

- numer detalu z danej klasy (odpowiednik  $j$ );

$$NU \quad (40)$$

- liczba operacji przebiegu uogólnionego (odpowiednik  $n_u$ );

$$CU(L), \quad L=1, \dots, NU \quad (41)$$

- kody kolejnych operacji przebiegu uogólnionego (odpowiednik  $\alpha^1, 1=1, \dots, n_u$ );

$$LA(L, J), \quad L=1, \dots, NU \quad (42)$$

- zestawy warunków występowania każdej operacji przebiegu uogólnionego (odpowiednik  $\bigwedge_j^1, 1 = 1, \dots, n_u$ ).

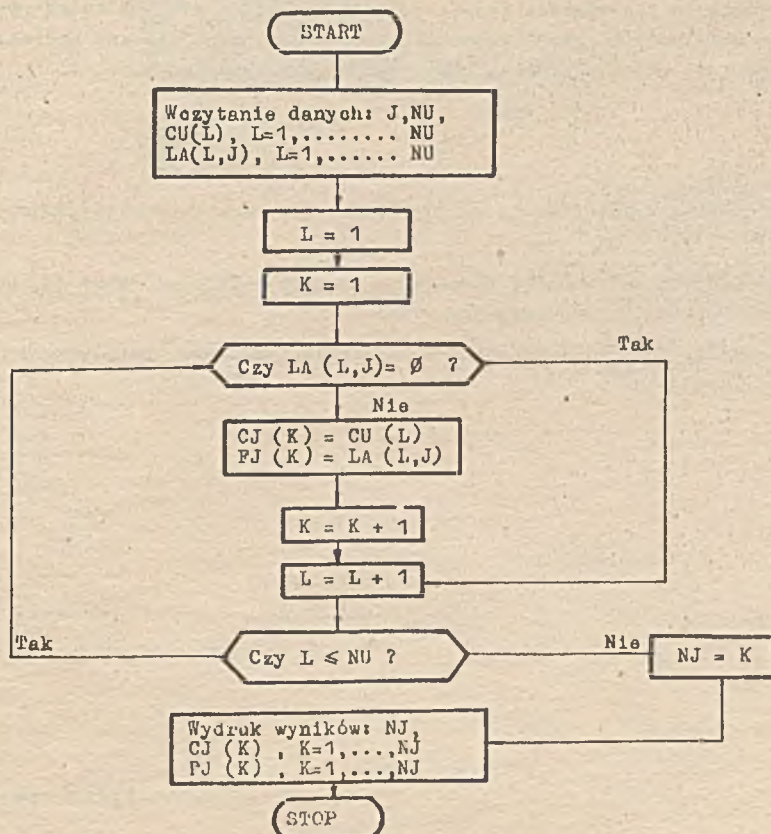
Wynikiem realizacjii programu jest

$$NJ \quad (43)$$

- liczba operacji w przebiegu  $M_j$  (odpowiednik  $n_j$ );

$$CJ(K), \quad K=1, \dots, NJ \quad (44)$$

- kody kolejnych operacji przebiegu procesu obróbki  $j$ -tego detalu (odpowiednik  $\alpha_j^k, k=1, \dots, n_j$ ).



Rys. 4. Schemat blokowy programu WTPI

### Wykorzystanie automatyzacji projektowania technologicznych przebiegów procesów obróbki detali

Budowa uogólnionego przebiegu procesu obróbki dla klasy detali i na jej podstawie wydzielenie technologicznych przebiegów indywidualnych dla detali z tej klasy za pomocą maszyn cyfrowych, przyspiesza w pewnych warunkach produkcji opracowanie technologicznego planu całej obróbki. Plan ten zawiera (w postaci kodów) nazwę i treść operacji, warunki ich występowania, stosowane wyposażenie i technologiczne oprzyrządowanie.

W zautomatyzowanych systemach projektowania technologicznego zastosowanie metod typizacji daje możliwość znacznego skrócenia czasu opracowywania algorytmów i programów, dotyczących obróbki oraz uwalnia technologów od czasochłonnej i mechanicznej pracy nad opracowywaniem procesów technologicznych dla typowych detali.

Automatyzację projektowania procesów technologicznych na podstawie metod typizacji przeprowadza się w następującej kolejności:

- konstrukcyjno-technologiczna analiza obrabianego detalu i określanie cech wpływających na podstawową strukturę przebiegu procesu obróbki detali,
- klasyfikacja detali, zgodnie z możliwościami technologicznymi zakładu,
- grupowanie detali,
- projektowanie przebiegu procesu technologicznego obróbki każdego detalu,
- normowanie operacji (parametry skrawania),
- wydruk kart technologicznych procesu obróbki.

Jak więc widać, przedstawiona w artykule automatyzacja projektowania technologicznych przebiegów procesów obróbki detali jest jednym z podstawowych zadań w automatyzacji projektowania procesów technologicznych.

System automatycznego projektowania daje się dostosowywać do różnych zmian występujących w produkcji oraz daje możliwość korekty technologicznego przebiegu procesu obróbki detali i pozwala na pełną lub częściową zmianę istniejących typowych rozwiązań na nowe.


#### Literatura

- [1] CWIETKOV W.D.: System automatyzacji projektowania procesów technologicznych. Tłum. z ros. Warszawa: PWN 1978
- [2] KAPUSTIN N.M.: Razrabotka technologicznych processov obrabotki detal'ej na stankach s pomošč'ju EVM. Moskva: Mašinostrojenije 1976
- [3] KURATOWSKI K.: Wstęp do teorii mnogości i topologii. Warszawa: PWN 1955


BOINTE udziela informacji  
z zakresu techniki komputerowej

BOINTE wydaje


# informacja ekspresowa

 OBIKTOWE  
SYSTEMY  
KOMPUTEROWE

# przegląd dokumentacyjny

 OBIKTOWE  
SYSTEMY  
KOMPUTEROWE

# biuletyn informacyjny

 OBIKTOWE  
SYSTEMY  
KOMPUTEROWE

Materiały konferencyjne, szkoleniowe, prospekty

BOINTE gromadzi

wydawnictwa zwarte, czasopisma krajowe i zagraniczne, katalogi i prospekty, sprawozdania z prac  
naukowo-badawczych oraz inne materiały informacyjne

BOINTE wykonuje usługi reprodukcyjne i poligraficzne  
kserokopie, mikrofilmy, kserokopie z zakresu posiadanych zbiorów

#### WARUNKI PRENUMERATY

Prenumeratę na kraj przyjmują Oddziały RSW "Prasa-Książka-Ruch" oraz urzędy pocztowe i doręczyciele w terminie do dnia 25 listopada na rok następny.

Cena prenumeraty rocznej zł 840.

Jednostki gospodarki uspołecznionej, instytucje, organizacje i wszelkiego rodzaju zakłady pracy zamawiają prenumeratę w miejscowych Oddziałach RSW "Prasa-Książka-Ruch", w miejscowościach zaś, w których nie ma Oddziałów RSW - w urzędach pocztowych.

Czytelnicy indywidualni opłacają prenumeratę wyłącznie w urzędach pocztowych i u doręczycieli.

Prenumeratę ze zleceniem wysyłki za granicę przyjmuje RSW "Prasa-Książka-Ruch", Centrala Kolportażu Prasy i Wydawnictw, ul. Towarowa 28, 00-958 Warszawa, konto PKO Nr 1153-201045.

Prenumerata ze zleceniem wysyłki za granicę jest droższa od prenumeraty krajowej o 50% dla zlecniodawców indywidualnych i o 100% dla zlecających instytucji i zakładów pracy.