



Politechnika Krakowska

M6 - Instytut Technologii Maszyn i Automatykacji Produkcji

M65 - Zakład Projektowania Procesów Wytwarzania



STUDIA STACJONARNE III STOPNIA

PRACA DOKTORSKA

mgr inż. Paweł Wojakowski

***Metoda projektowania przepływu produkcji
w warunkach zmiennego zapotrzebowania***

Promotor:

dr hab. inż. Jan Duda, prof. PK

Kraków 2012

Składam serdeczne podziękowania wszystkim osobom, których pomoc, cenne uwagi i życzliwość przyczyniły się do powstania niniejszej pracy.

Spis treści

Oznaczenia umowne	5
1. Wprowadzenie	8
1.1. Sformułowanie założeń do problemu naukowego	9
2. Analiza literatury	11
2.1. Klasyczne metody planowania w warunkach zmiennego zapotrzebowania ..	11
2.1.1. Podstawowa metoda MRP I	12
2.1.2. Metoda MRP II	15
2.2. Problemy planowania terminów realizacji zleceń w złożonych systemach produkcyjnych	17
2.3. Metody harmonogramowania zadań w warunkach produkcji małoseryjnej ..	19
2.4. Rozwój komputerowych systemów planowania i harmonogramowania produkcji	24
2.4.1. Systemy planowania zawierające metody MRP I oraz MRP II	24
2.4.2. Systemy klasy ERP	26
2.4.3. Systemy APS	28
2.5. Zastosowanie technologii grupowej do projektowania przepływu produkcji	30
2.6. Rozwój koncepcji kształtowania wirtualnych komórek przedmiotowych	34
2.7. Wnioski z analizy literatury	38
3. Cel, teza i zakres pracy	42
4. Planowanie przepływu produkcji w warunkach zmiennego zapotrzebowania: Algorytm ProduFlow PM	45
4.1. Warunki tworzenia wirtualnych komórek przedmiotowych w algorytmie ProduFlow PM	51
4.1.1. Koncepcja grupowania zleceń dla produkcji małoseryjnej	51
4.1.2. Zastosowanie hierarchicznego aglomeracyjnego algorytmu klastrowania do grupowania zleceń	56
4.2. Planowanie przepływu produkcji z zastosowaniem wirtualnych komórek przedmiotowych	65
4.2.1. Opis funkcjonowania algorytmu genetycznego w nowej metodzie planowania przepływu produkcji	65
4.2.2. Metoda szeregowania zadań dla potrzeb harmonogramowania z wirtualnymi komórkami przedmiotowymi	71
4.2.3. Metoda układania harmonogramu z wirtualnymi komórkami przedmiotowymi	75
4.3. Planowanie okienkowe – określenie długości okienek planistycznych z wykorzystaniem metaheurystyki przeszukiwania tabu	77
4.3.1. Planowanie okienkowe w dynamicznych, stochastycznych warunkach produkcji	78
4.3.2. Teoria planowania okienkowego	78
4.3.3. Szacowanie planowanego terminu realizacji zbioru zleceń z wykorzystaniem metaheurystyki przeszukiwania tabu	85
5. Budowa systemu komputerowego do planowania przepływu produkcji w warunkach zmiennego zapotrzebowania	95
5.1. Założenia do budowy systemu komputerowego	95
5.1.1. Architektura systemu komputerowego	96
5.1.2. Funkcjonalność systemu komputerowego	99
5.2. Schemat budowy bazy danych	101
5.3. Aplikacja kliencka	104
6. Weryfikacja eksperymentalna metody planowania przepływu produkcji w warunkach zmiennego zapotrzebowania	108
6.1. Test eksperymentalny – wyniki grupowania zleceń	111

6.2. Test eksperymentalny – porównanie wyników planowania i harmonogramowania z zastosowaniem algorytmu ProduFlow PM na tle harmonogramowania general job-shop	116
7. Wnioski i podsumowanie.....	124
8. Literatura.....	127
Dodatek A. Przykład zastosowania reguły priorytetu SPT do szeregowania zadań i obiektów.....	134
Dodatek B. Informacje o systemie wytwórczym wykorzystanym jako przykład do testowania działania algorytmu ProduFlow PM	142
Dodatek C. Skrypty procedur składowanych służących do realizacji algorytmu ProduFlow PM	150

OZNACZENIA UMOWNE

$a = 1, \dots, A$	– numer porządkowy wyrobu złożonego,
$e = 1, \dots, E$	– numer porządkowy etapu procesu wytwórczego w wariancie marszruty v części oryginalnej i ,
$f = 1, \dots, F$	– numer porządkowy horyzontu planowania,
$h = 1, \dots, H$	– numer porządkowy okna planistycznego w horyzoncie planowania f ,
$i = 1, \dots, I$	– numer porządkowy wyrobu prostego/części oryginalnej,
$j = 1, \dots, J$	– numer porządkowy operacji technologicznej w wariancie marszruty v wyrobu prostego/części oryginalnej i ,
$k = 1, \dots, K$	– numer porządkowy konfiguracji maszyny m ,
$KP_x^0 = KP_1^0, \dots, KP_x^0$	– numer porządkowy komórki produkcyjnej 0 stopnia w komórce produkcyjnej I stopnia KP_x^1 ,
$KP_x^1 = KP_1^1, \dots, KP_x^1$	– numer porządkowy komórki produkcyjnej I stopnia w komórce produkcyjnej II stopnia KP_x^2 ,
$KP_x^2 = KP_1^2, \dots, KP_x^2$	– numer porządkowy komórki produkcyjnej II stopnia w komórce produkcyjnej wyższego rzędu (w niniejszej pracy w systemie wytwórczym),
$m = 1, \dots, M$	– numer porządkowy maszyny określonego typu,
$t = 1, \dots, T$	– numer porządkowy klastra tymczasowego,
$v = 1, \dots, V$	– numer porządkowy wariantu marszruty technologicznej wyrobu prostego/części oryginalnej i ,
$w = 1, \dots, W$	– numer porządkowy wyrobu finalnego,
$y = 1, \dots, Y$	– numer porządkowy zlecenia produkcyjnego wykonywanego w komórce produkcyjnej II stopnia KP_x^2 , w oknie planistycznym h ,
$z = 1, \dots, Z$	– numer porządkowy klastra zwykłego,
A_{Th}	– dostępny czas pracy maszyn w oknie planistycznym h ,
C	– zbiór klastrów utworzonych na jednym poziomie dendrogramu,
C_{maxh}	– maksymalny termin zakończenia zleceń wygenerowany podczas harmonogramowania bez zleceń priorytetowych,
c_n	– liczba klastrów znajdujących się w zbiorze C .
C_{Pmaxh}	– maksymalny termin zakończenia zleceń wygenerowany po dodaniu zleceń priorytetowych do harmonogramu,
$C_l(c_i, c_k)$	– klastery tymczasowy utworzony na bazie dwóch klastrów C_i , oraz C_k ,
$C_z(KP_x^2, h)$	– identyfikator klastra zwykłego w fazie grupowania zleceń produkcyjnych,
$Cz_y(KP_x^2, h)$	– termin zakończenia zlecenia produkcyjnego z_y ,
$d(KP_x^2, h)$	– planowany termin realizacji zbioru zleceń produkcyjnych,
$Ez_y(KP_x^2, h)$	– kara wcześniejszego wykonania zlecenia z_y ,
$f(\gamma)$	– prawdopodobieństwo wyboru osobnika z populacji rozstawionej w metodzie ruletki algorytmu genetycznego,
$F_{z_y(KP_x^2, h)}$	– czas przepływu zlecenia produkcyjnego z_y przez komórkę produkcyjną KP_x^2 w oknie planistycznym h ,

$K(KP_x^2, h)$	– całkowity koszt realizacji zleceń produkcyjnych w komórce produkcyjnej KP_x^2 w oknie planistycznym h ,
$k_E(KP_x^2, h)$	– koszt jednostkowy wcześniejszego zakończenia dowolnego zlecenia produkcyjnego,
$k_T(KP_x^2, h)$	– koszt jednostkowy późniejszego zakończenia dowolnego zlecenia produkcyjnego,
$Kz_{yE}(KP_x^2, h)$	– koszt wczesnego zakończenia zlecenia produkcyjnego z_y ,
$Kz_{yP}(KP_x^2, h)$	– koszt przetwarzania zlecenia produkcyjnego z_y ,
$Kz_{yT}(KP_x^2, h)$	– koszt późnego zakończenia zlecenia produkcyjnego z_y ,
$Kz_{yW}(KP_x^2, h)$	– koszt składowania produkcji w toku zlecenia produkcyjnego z_y ,
$l_f(KP_x^2)$	– długość każdego okna planistycznego komórki produkcyjnej II stopnia KP_x^2 w horyzoncie planowania f ,
$LCF(c_i, c_k)$	– wartość najdłuższego wspólnego podłańcucha dwóch klastrów C_i , oraz C_k ,
$LC_z(KP_x^2, h)$	– długość łańcucha zadań klastra zwykłego C_z ,
L_{SetC}	– liczba par wyselekcjonowanych klastrów w podzbiorze $SetC$,
$LS_{\max}(KP_x^2, h)$	– najdłuższy łańcuch zadań spośród łańcuchów $LS_y(KP_x^2, h)$ singletonów,
$LS_y(KP_x^2, h)$	– długość łańcucha zadań singletona S_y ,
l_{dend}	– zmienna sterująca pętlą iteracyjną poziomów dendrogramu,
l_{tac}	– stała długość horyzontów planowania,
$N(d)$	– zbiór rozwiązań sąsiednich w metaheurystyce przeszukiwania tabu,
Q_{z_y}	– wielkość produkcji zapisana w zleceniu produkcyjnym z_y ,
$r(KP_x^2, h)$	– termin wspólnego rozpoczęcia zbioru zleceń produkcyjnych,
$r_{z_y}(KP_x^2, h)$	– termin rozpoczęcia zlecenia produkcyjnego,
$s(d, d_i)$	– zapis ruchu z rozwiązania bieżącego do rozwiązania sąsiedniego w metaheurystyce przeszukiwania tabu C_i , oraz C_k ,
$SetC$	– podzbiór będący dwuelementową kombinacją bez powtórzeń zbioru klastrów C ,
$S_{mk-1,k}$	– czas przebrojenia maszyny m z konfiguracji $k-1$ w konfigurację k ,
$S_y(KP_x^2, h)$	– identyfikator singletona w fazie grupowania zleceń produkcyjnych,
$T(d)$	– zbiór ruchów zapisanych na liście tabu,
$Tabu_{Length}$	– parametr służący do określenia długości listy tabu,
$Tabu_{Loop}$	– parametr służący do określenia liczby iteracji w pętli optymalizacyjnej metaheurystyki przeszukiwania tabu,
T_{Downh}	– czas trwania nieplanowanych przestojów w oknie planistycznym h ,
t_{ivjmk}	– czas jednostkowy na maszynie m w konfiguracji k dla operacji technologicznej j wyrobu prostego/części oryginalnej i wykonywanej w wariacie marszruty technologicznej v ,
$Tz_y(KP_x^2, h)$	– kara późniejszego wykonania zlecenia z_y ,

$tz_{yj-l,j}$	– czas składowania zapasu produkcji w toku zlecenia produkcyjnego $z_{y,}$, pomiędzy zadaniami $j-l$, a j ,
tz_{yjmk}	– uproszczony zapis czasu jednostkowego operacji j , traktowanej jako zadanie j dla zlecenia produkcyjnego $z_{y,}$,
$z(a)$	– zlecenie produkcyjne na wyrób złożony a ,
$z(a,i)$	– zlecenie produkcyjne na część oryginalną i wchodzącą w skład wyrobu złożonego a ,
$z(i)$	– zlecenie produkcyjne na część oryginalną i ,
$z(KP_x^2, h)_{a=1, \dots, n}$	– zlecenia kompletacyjne dla zlecenia $z(w)$,
$z_y(KP_x^2, h)$	– numer porządkowy zlecenia etapowego, wykonywanego w komórce produkcyjnej II stopnia KP_x^2 , w oknie planistycznym h ,
$z(w)$	– zlecenie produkcyjne na wyrób finalny w ,
η_A	– wskaźnik dostępności,
η_P	– wskaźnik priorytetowości,
λ_{cross}	– punkt krzyżowania w algorytmie genetycznym,
λ_{mut}	– punkt mutacji w algorytmie genetycznym,
$\xi(KP_x^2, h)$	– termin wspólnego zakończenia zbioru zleceń produkcyjnych,
$\zeta(KP_x^2, h)$	– dolna granica rozwiązań dopuszczalnych w metaheurystyce przeszukiwania tabu.

1. WPROWADZENIE

W dzisiejszym mocno konkurencyjnym środowisku biznesu decydenci przedsiębiorstw produkcyjnych, których myśli ukierunkowane są na badania potrzeb kształtowania wartości oraz uzyskanie wymiernego wyniku ekonomicznego, są zmuszeni do ciągłego doskonalenia metod zarządzania produkcją. Jest to związane ze zmiennością otoczenia przedsiębiorstwa produkcyjnego, wynikającą zarówno z intensywnych działań konkurencji, jak i ograniczeń czasowych, majątkowych i kapitałowych przedsiębiorstwa. Ciągły nacisk na ograniczanie kosztów produkcji przy jednoczesnym wzroście produktywności systemu wytwórczego jest wyzwaniem, zwłaszcza w systemach wytwórczych charakteryzujących się typem produkcji małoseryjnej. Złożony proces zarządzania produkcją wieloasortymentową, w szczególności w dziedzinie planowania produkcji małoseryjnej wstępnie ukierunkowały obszar badań naukowych autora pracy.

Podstawową przesłanką wyboru problemu naukowego, tj. badania w dziedzinie planowania przepływu produkcji w warunkach zmiennego zapotrzebowania (badania głównie dotyczą problemu planowania produkcji małoseryjnej), była potrzeba opracowania wydajnych instrumentów zarządzania produkcją w odpowiedzi na problemy zgłaszane przez decydentów przedsiębiorstw przemysłu maszynowego. Głównym celem prowadzonych prac badawczych jest poprawa relacji na linii producent-odbiorca. W tym miejscu należy uzasadnić tak obrany cel pracy.

Z rozmów autora pracy z przedstawicielami lokalnych przedsiębiorstw produkcyjnych (głównie jest to bezpośrednie kierownictwo produkcji) borykających się z problemami produkcji małoseryjnej wynika chęć poprawy wartości wskaźników: szybkości realizacji zamówień, a także wiarygodności potwierdzonych terminów realizacji zamówień. Największe problemy tych przedsiębiorstw produkcyjnych wynikają z błędów podejmowania decyzji na szczeblu planowania zarówno średniookresowego, jak i krótkookresowego (czyli szczegółowego harmonogramowania produkcji na poziomie operacyjnym). Powszechnie stosowana jest praktyka odraczania terminów realizacji zamówień co wpływa na obniżenie wartości wskaźnika wiarygodności producenta.

Należy podkreślić również fakt, iż większość przedsiębiorstw produkcyjnych nie stać na zakup nowej, wydajnej, niezawodnej, ale również drogiej technologii celem poprawy produktywności systemu produkcji wieloasortymentowej. Co więcej, ośrodki badawcze zajmujące się problematyką organizacji i zarządzania produkcją (*Kaizen Institute*, *Lean Enterprise Institute*, czy *Goldratt Institute*) udowodniły, że wdrożenie nowej technologii może skutkować pogorszeniem wskaźnika produktywności systemu. Bez właściwego narzędzia do zarządzania produkcją koszty produkcji, zwłaszcza przy zastosowaniu zaawansowanej technologii nie ulegną obniżeniu, czasami nawet mogą się zwiększyć.

Gwałtowna globalizacja i rozwój w obszarze technologii informacyjnej sprawiają, że klienci mają lepszy wgląd nie tylko w ofertę producentów, ale również w sposób pracy przedsiębiorstw produkcyjnych. W wyniku tego oczekiwania klientów w stosunku do jakości zamawianych wyrobów oraz terminowości dostaw niezmiennie rosną od kilku lat. Są to więc

decydujące czynniki mogące zaważyć na decyzji zakupu produktu przez klienta. Im więcej zamówień w danym przedsiębiorstwie produkcyjnym jest realizowanych na czas, tym większe zaufanie klientów. Czynnikiem terminowej realizacji zamówienia nabiera szczególnego znaczenia, gdy przedsiębiorstwo jest poddostawcą wyrobów dla kolejnego producenta w łańcuchu dostaw.

Innym aspektem globalizacji i rozwoju technologii informacyjnych jest fakt, iż klienci żądają zindywidualizowanych produktów [40]. W dzisiejszych czasach coraz mniej znacząca jest produkcja masowa. Wiele firm produkcyjnych boryka się z problemem przejścia z produkcji masowej na produkcję średnioseryjną lub z produkcji średnioseryjnej na produkcję małoseryjną w wyniku zmieniającego się zapotrzebowania [14]. Niedogodności uwidaczniają się szczególnie w przemyśle maszynowym, gdzie przy produkcji małoseryjnej lub średnioseryjnej zindywidualizowanych produktów, częste przestawienia maszyn w celu poprawy produktywności są nieuzasadnione ekonomicznie. Ponadto, fluktuacja produkowanego asortymentu wyrobów, charakterystyczna w warunkach produkcji małoseryjnej, powoduje trudność w dotrzymaniu terminów realizacji zamówień. Wynika stąd potrzeba opracowywania nowych metod zarządzania produkcją, które są w stanie wspomóc proces planowania przepływu produkcji w przedsiębiorstwie, aby podnieść wydajność systemu wytwórczego oraz zmniejszyć koszty produkcji [130].

Niniejsza praca poświęcona jest zagadnieniu planowania przepływu produkcji w warunkach zmiennego zapotrzebowania z wyszczególnieniem przypadku produkcji małoseryjnej. Dodatkowym uzasadnieniem podjęcia tematu pracy jest obecne zainteresowanie tą tematyką na świecie w kręgach uczelnianych i biznesowych. W celu szczegółowego wyjaśnienia problemu naukowego poniżej sformułowano założenia.

1.1. Sformułowanie założeń do problemu naukowego

Dany jest system wytwarzania, w którym realizuje się wieloasortymentową produkcję o charakterze dyskretnym. Asortyment produkowanych wyrobów jest zróżnicowany i może w zwiększyć się w czasie. Występuje nieregularne zapotrzebowanie na poszczególne wyroby w zależności od bieżących potrzeb klientów. Klienci zamawiają wyroby w niewielkich ilościach mieszczących się w zakresie kilkudziesięciu do kilkuset sztuk na jedno zamówienie. Oznacza to, że w systemie wytwórczym dominuje produkcja małoseryjna.

W pracy skupiono uwagę na rozpatrywaniu konwencjonalnych systemów wytwórczych, w których możliwe jest wydzielenie komórek produkcyjnych II stopnia (np. wydziałów produkcyjnych). W komórkach produkcyjnych II stopnia występują maszyny uniwersalne wymagające obsługi pracowników. Automatyzacja występuje jedynie na poziomie realizacji zabiegów technologicznych. Przez stanowiska robocze komórki produkcyjnej II stopnia przebiegają marszruty technologiczne wyrobów w różnych kierunkach, przy czym istnieją grupy ukierunkowanych marszrut technologicznych. Ze względu na małoseryjny typ produkcji, wydzielenie komórek produkcyjnych I stopnia o specjalizacji przedmiotowej jest nieuzasadnione ekonomicznie. Zlecenia produkcyjne, wystawiane na wyrób w ramach realizacji produkcji w komórce produkcyjnej II stopnia, pojawiają się w nierównych odstępach czasu. Ponadto liczba sztuk wyrobu, na które wystawiono zlecenie produkcyjne jest zróżnicowana (waha się w

zakresie 50 – 500 sztuk na zlecenie produkcyjne). Warunki produkcji małoseryjnej skutkują koniecznością podziału horyzontu planowania na pewne przedziały czasu. Poszczególne przedziały czasu nie muszą mieć jednakowej długości. Istotne jest natomiast, aby każde zlecenie produkcyjne było wykonane w jednym przedziale czasu. Podział horyzontu planowania na przedziały czasu pozwala wyeliminować zjawisko zmiennego zapotrzebowania. Pojawia się jednak problem z ustaleniem odpowiedniej długości trwania przedziałów czasu tak, aby czas realizacji zleceń był jak najkrótszy, a z drugiej strony, aby terminy realizacji zleceń były dotrzymane.

Marszruty technologiczne są nieukierunkowane, co wymusza konieczność współdziałania maszyn w zależności od bieżącego zapotrzebowania. Może to doprowadzić do wydłużenia czasu realizacji poszczególnych zleceń produkcyjnych z powodu obniżenia efektywności produkcji. W związku z tym poszukuje się odpowiedzi na pytanie, czy istnieje sposób na skrócenie czasu realizacji zleceń produkcyjnych w warunkach produkcji małoseryjnej, przy zmieniającym się zapotrzebowaniu.

Uzyskanie odpowiedzi na tak sformułowane pytanie wymaga opracowania metody, która pozwoli pogrupować prognozowane zlecenia produkcyjne w celu ich wspólnej realizacji. Następnie umożliwi podjęcie decyzji o długości przedziałów czasu, w których prognozowane zbiory zleceń będą realizowane bez naruszenia wyznaczonych terminów ich realizacji. Ponadto, metoda powinna umożliwić optymalizację przepływu produkcji według kryterium minimalizacji terminów realizacji zleceń. Ze względu na złożoność zjawisk występujących w tak sformułowanym problemie postanowiono zawęzić zakres analizowanych systemów wytwórczych do takich, w których występują procesy technologiczne obróbki i montażu. Za asortyment produkowanych wyrobów przyjęto typowe części maszyn.

W kolejnym punkcie wykazano aktualny stan wiedzy dotyczący metod stosowanych do rozwiązania sformułowanego problemu. Przegląd analizy literatury podsumowano wnioskami, które uzasadniają podjętą tematykę i sposób rozwiązania problemu.

2. ANALIZA LITERATURY

Z uwagi na sformułowany problem badawczy dokonano przeglądu literatury światowej zarówno w dziedzinie planowania produkcji jak i w dziedzinie projektowania systemów wytwórczych. Celem analizy literatury jest uzyskanie informacji na temat aktualnych sposobów radzenia sobie z problemem planowania produkcji małoseryjnej. Należy więc z jednej strony zapoznać się z metodami zarządzania produkcją z uwagi na planowanie zleceń produkcyjnych w warunkach zmiennego zapotrzebowania oraz alokację zasobów do celu realizacji tych zleceń produkcyjnych. Z drugiej strony należy określić warunki konfiguracyjne oraz funkcjonalne systemów wytwórczych, w których produkcja małoseryjna jest sprawnie zarządzana.

Biorąc pod uwagę sformułowany problem, analizę literatury podzielono na sześć części. W pierwszej części dokonano przeglądu metod powszechnie stosowanych w praktyce do planowania produkcji. W drugiej części nawiązano do problemu bezpośrednio związanego z planowaniem przepływu produkcji. Dokonano przeglądu literatury w zakresie planowania terminów realizacji zleceń na podstawie zaplanowanego przepływu produkcji. W trzeciej części dokonano analizy metod harmonogramowania, z których bezpośrednio wynika charakter przepływu produkcji. W części czwartej przedstawiono zarys rozwoju komputerowych systemów planowania produkcji. Piąta część opisuje zastosowanie technologii grupowej do formowania komórek produkcyjnych w warunkach produkcji seryjnej. Szósta część ujmuje koncepcję budowy wirtualnych komórek przedmiotowych, jako formy organizacji produkcji z okresowo dedykowanymi maszynami w celu zaspokojenia zmieniającego się zapotrzebowania. Ostatnią część stanowią wnioski z analizy literatury nakreślające kierunek rozwoju planowania przepływu produkcji w warunkach zmiennego zapotrzebowania.

2.1. Klasyczne metody planowania w warunkach zmiennego zapotrzebowania

Zmienność zamawianych wyrobów w horyzoncie planowania oraz związana z tym potrzeba szybkiej reakcji systemu wytwarzania doprowadziły do rozwoju systemów produkujących na zlecenie klienta. Strategia produkcji na zlecenie klienta nosi nazwę MTO (ang. *make-to-order*), przy czym musi być spełnionych 14 reguł implementacji, aby strategia MTO przynosiła wymierne korzyści finansowe [101][180]. Reguły implementacji strategii MTO stały się podstawą do budowy modeli decyzyjnych wyboru produkcji na zlecenie, bądź produkcji na magazyn wyrobów gotowych MTS (ang. *make-to-stock*) [115][143][185].

Przyjęcie strategii MTO pociąga za sobą konieczność zastosowania odpowiedniej metody zarządzania produkcją. Z przeprowadzonych w przemyśle maszynowym sondaży [7] wynika, że dwa na trzy przedsiębiorstwa borykające się z problemami produkcji małoseryjnej wykorzystują systemy ERP (ang. *Enterprise Resource Planning*) z zaimplementowaną metodą MRP (ang. *Materials Requirements Planning*) do zarządzania produkcją. Popularność metody MRP jest skutkiem kampanii prowadzonej przez stowarzyszenie APICS (ang. *American Production and Inventory Control Society*, obecnie *The Association for Operations Management*), czyli organizację wspierającą narodowy program propagowania i szkolenia w

Stanach Zjednoczonych. Metoda MRP została opracowana w celu sterowania zapasami w sytuacji produkowania wyrobów złożonych. Jednakże, cytując słowa twórcy metody MRP, J. Orlicky'ego [117]:

„...stosowanie metody MRP ogranicza się na ogół do produkcji o charakterze dyskretnym. Do niedawna sądzono, że zastosowanie metody MRP jest uzasadnione tylko w warunkach produkcji obejmującej operacje kończące się wytworzeniem względnie złożonego wyrobu, ale rozwój metody zaprzeczył temu. Metoda MRP może być i jest stosowana także przez przedsiębiorstwa wytwarzające bardzo proste wyroby, włączając w to wyroby jednoczesne.”

MRP również jest stosowana z powodzeniem w zarządzaniu wieloasortymentową produkcją, niekoniecznie małoseryjną, gdzie przepływ produkcji każdego wyrobu podzielony jest na etapy, wytwarzane w kolejnych wydziałach zakładu produkcyjnego [18][106].

2.1.1. Podstawowa metoda MRP I

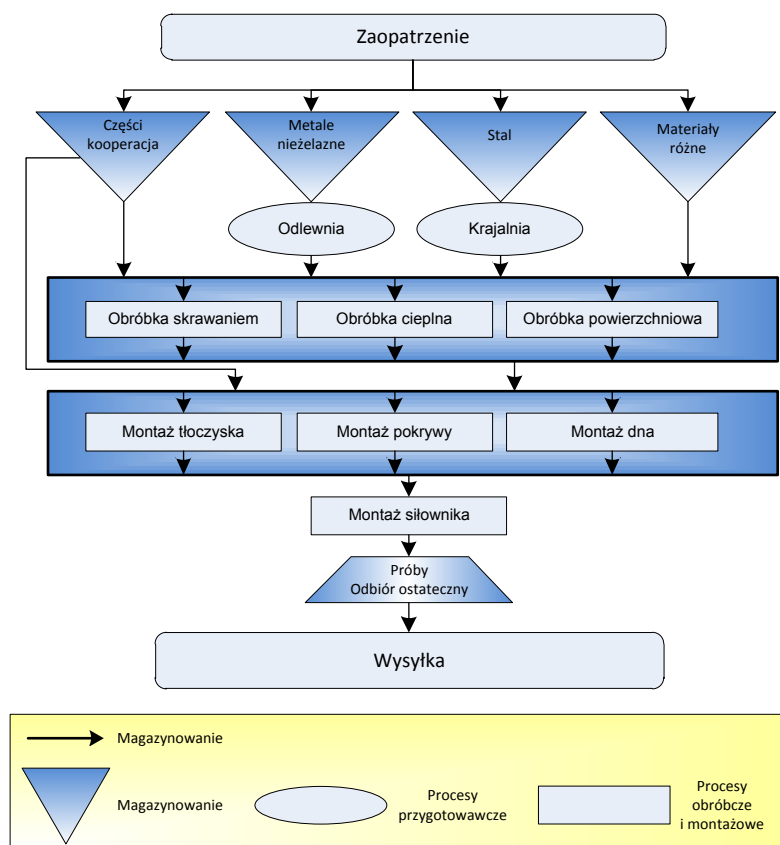
Podstawowa metoda MRP I może być stosowana do zarządzania produkcją w warunkach zmiennego zapotrzebowania z kilku powodów [150]:

- Metoda MRP I uwzględnia prognozę zapotrzebowania. Natomiast problem stanowi przeplanowanie, gdy zapotrzebowanie ulegnie zmianie (zjawisko nerwowości metody MRP) [103].
- Metoda MRP I uwzględnia zależności zachodzące w strukturze procesu produkcyjnego określonego wyrobu złożonego. Na podstawie tych zależności planowane jest zapotrzebowanie na poszczególne składowe wyroby. Zapotrzebowanie to określane jest mianem popytu zależnego.
- Dane wyjściowe metody MRP I mogą być dalej wykorzystane do sprawdzenia obciążenia komórek produkcyjnych. Podstawową daną wyjściową jest wielkość produkcji. W metodzie MRP wielkość produkcji jest wyznaczana w stałych odstępach czasu, tzw. okresach planowania. Długości okresów planowania najczęściej są ustanawiane jako tygodniowe lub dwutygodniowe przedziały czasu.

Do przeprowadzenia obliczeń wielkości produkcji na każdy komponent wyrobu złożonego metodą MRP I wymagana jest znajomość struktury procesu produkcyjnego tego wyrobu według faz technologicznych [20]. Termin faza technologiczna obowiązuje w polskiej terminologii [20]. Natomiast światowy standard ustanowiony przez APICS definiuje tożsame pojęcie określone terminem etap wytwarzania. Przez etap wytwarzania rozumie się zbiór operacji wykonywanych na stanowiskach roboczych w komórce produkcyjnej co najmniej I stopnia. Materiały do produkcji w etapie wytwarzania dostarczane są z magazynu centralnego, następnie przechodzą przez wszystkie operacje zdefiniowane w etapie wytwarzania, po czym jako materiały przetworzone z powrotem trafiają do magazynu centralnego [7]. Produkcja w liniach produkcyjnych lub gniazdach o specjalizacji przedmiotowej jest traktowana jako etap wytwarzania. Przykładowy schemat struktury procesu produkcyjnego wyrobu złożonego podzielonej na etapy wytwarzania zilustrowano na Rys. 1.

Strukturę procesu produkcyjnego wyrobu podzieloną na etapy wytwarzania dzieli się na trzy rodzaje [7]:

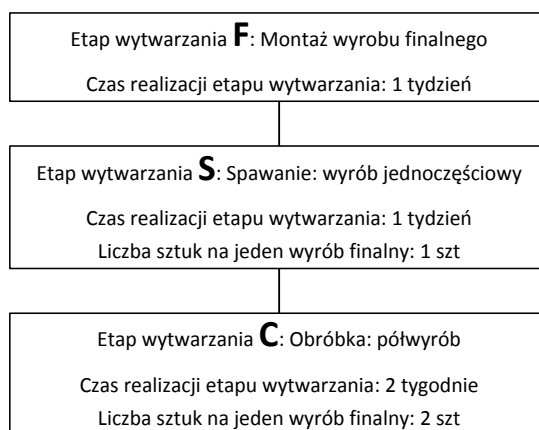
- 1) struktura szeregową, która składa się z szeregu następujących po sobie etapów wytwarzania, tzn. występują wyłącznie wyroby jednoczęściowe, wyroby te przepływają pomiędzy kolejnymi etapami wytwarzania w sposób ukierunkowany,
- 2) struktura montażowa, w której materiały łączą się w kolejnych etapach wytwarzania wraz z postępowaniem realizacji procesu produkcyjnego, ostatni etap wytwarzania musi być montażem finalnym wyrobu,
- 3) struktura sieciowa, w której przepływ materiału przebiega pomiędzy różnymi etapami wytwarzania (strukturę sieciową przedstawiono na Rys. 1), występuje w sytuacji, gdy pewna część wchodzi w skład listy materiałowej kilku różnych wyrobów złożonych.



Rys. 1. Przykładowy schemat struktury sieciowej procesu produkcyjnego wyrobu złożonego podzielonej na etapy wytwarzania [20].

Z terminem etap wytwarzania jest związany czas niezbędny do jego realizacji. Czas realizacji etapu wytwarzania jest utożsamiany ze stałą długością okresu planowania, dzięki czemu proces planowania produkcji jest niezależny od wielkości produkcji. Czas zależy jedynie od przyjętej długości okresu planowania. Dzięki temu, niezależnie od zmieniającego się zapotrzebowania, znając wymagany termin zakończenia produkcji wyrobu złożonego, oblicza się terminy rozpoczęcia produkcji poszczególnych składowych wyrobu na zasadzie planowania wstecz.

Przykład wykorzystania metody MRP I do kalkulacji zapotrzebowania na poszczególne składowe wyroby, realizowanego zgodnie z szeregową strukturą procesu produkcyjnego (jak na Rys. 2), pokazano na Rys. 3 [167].



Rys. 2. Przykład struktury szeregowej procesu produkcyjnego wyrobu [167].

W metodzie MRP I rozpatruje się jedynie składowe wyrobu, które są produkowane w systemie wytwórczym. Pozostałe komponenty wchodzące w skład wyrobu (zakupywane lub wykonywane w kooperacji) nie są brane pod uwagę, dlatego struktura procesu produkcyjnego, przedstawiona na Rys. 2, może być traktowana jako struktura szeregowa.

W analizowanym przykładzie wyodrębniono trzy etapy wytwórcze. Każdy etap wytwórczy jest związany z inną technologią wytwarzania. Można zatem wywnioskować, że każdy etap wytwarzania jest wykonywany w innym wydziale produkcyjnym (w wydzielonej komórce produkcyjnej II stopnia). Każdy wydział produkcyjny dysponuje indywidualnym kalendarzem planowania, stąd czasy realizacji w etapach wytwarzania różnią się. Ponadto, każdy wydział produkcyjny realizuje zamówienia według przyjętej strategii sterowania zapasami co zaznaczono na Rys. 3. Wyrób finalny jest montowany według strategii partia-na-partię (ang. *lot-for-lot*), która z uwagi na minimalizację poziomu zapasów może być stosowana w przypadku produkcji małoseryjnej. Zapas wyrobu jednoczęściowego, powstały w wyniku realizacji produkcji w etapie wytwarzania S, jest gromadzony co stały okres zamawiania według strategii POQ (ang. *Periodic Order Quantity*). Z kolei zapas półwyrobu, powstały w wyniku realizacji produkcji w etapie wytwarzania C, jest uzupełniany w stałej wielkości zamówienia według strategii EOQ (ang. *Economic Order Quantity*).

Szczegółowy opis funkcjonowania metody MRP I oraz sposób kalkulacji parametrów znajdujących się w tabelach składowych wyrobu można znaleźć w publikacjach książkowych [7][18][117].

Kalkulacja zapotrzebowania na poszczególne wyroby metodą MRP I wymaga obliczenia następujących parametrów [84]:

- 1) zapotrzebowania netto, w celu możliwości wygenerowania zamówień na składowe wyrobu,
- 2) wielkości produkcji (na Rys. 3 oznaczonej parametrem planowane dostawy), czyli ilości materiału przypadającego na wyprodukowanie każdej składowej wyrobu,
- 3) terminów rozpoczęcia produkcji na każdą wielkość produkcji wygenerowaną na podstawie zapotrzebowania netto,
- 4) zapotrzebowania brutto, czyli zapotrzebowania na składową podrzędną w strukturze procesu produkcyjnego wyrobu (np. na Rys. 3 składowa C jest składową podrzędną w stosunku do składowej S).

Okres planowania	0	1	2	3	4	5	6	7	8	9
Plan nadrzędny		30	30	40	40	50	50	50	30	30

Wyrób finalny <i>F</i>				partia-na-partię							czas realizacji: 1						
Okres planowania	0	1	2	3	4	5	6	7	8	9							
Zapotrzebowanie brutto		30	30	40	40	50	50	50	30	30							
Uzgodnione dostawy		30															
Dostępny zapas	0	0	0	0	0	0	0	0	0	0							
Zapotrzebowanie netto		0	30	40	40	50	50	50	30	30							
Planowane dostawy			30	40	40	50	50	50	30	30							
Planowane zamówienia		30	40	40	50	50	50	30	30								

Wyrób jednoczęściowy <i>S</i>				okresowa wielkość zamówienia		POQ: 2		czas realizacji: 1			
Okres planowania	0	1	2	3	4	5	6	7	8	9	
Zapotrzebowanie brutto		30	40	40	50	50	50	30	30		
Uzgodnione dostawy		70									
Dostępny zapas	0	40	0	50	0	50	0	30	0		
Zapotrzebowanie netto		0	0	40	0	50	0	30	0		
Planowane dostawy				90		100		60			
Planowane zamówienia			90		100		60				

Półwyrób <i>C</i>				ekonomiczna wielkość zamówienia			EOQ: 120		czas realizacji: 2			
Okres planowania	0	1	2	3	4	5	6	7	8	9		
Zapotrzebowanie brutto			180		200		120					
Uzgodnione dostawy			120									
Dostępny zapas	60	60	0	120	40	40	40	40				
Zapotrzebowanie netto		0	0	0	80	0	80	0				
Planowane dostawy				120	120		120					
Planowane zamówienia		120	120		120							

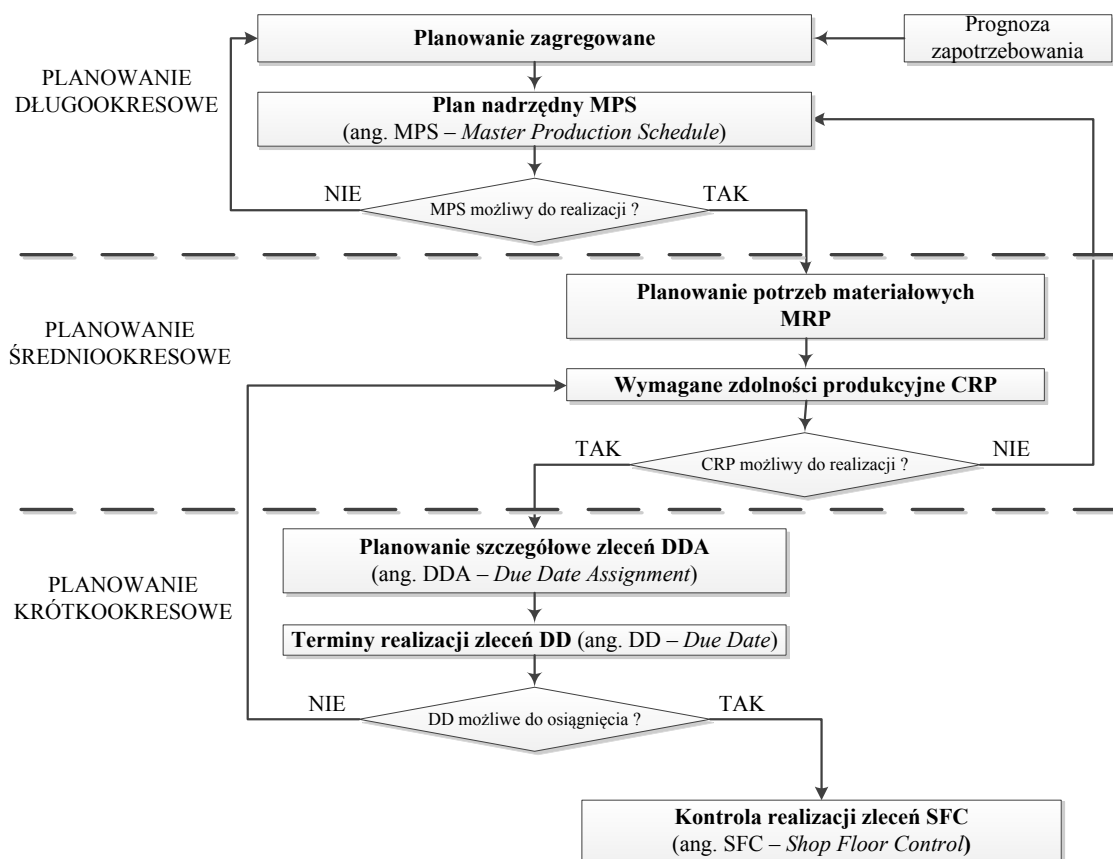
Rys. 3. Kalkulacja zapotrzebowania na składowe wyrobu z Rys. 2 metodą MRP I [167].

W metodzie MRP zakłada się, że każda faza technologiczna (każdy etap wytwarzania) jest wykonywana w odrębnej komórce produkcyjnej. Dla każdej komórki produkcyjnej generowany jest plan potrzeb materiałowych. Wymaga się więc dodatkowego sprawdzenia, czy poziom obciążenia komórek produkcyjnych, wynikający z planu potrzeb materiałowych, nie przekracza dostępnych zdolności produkcyjnych komórek produkcyjnych. Aby możliwe było sprawdzenie obciążenia komórek produkcyjnych, podstawowa metoda MRP I ewoluowała do metody MRP II. Metodę MRP II opisano w kolejnym podrozdziale.

2.1.2. Metoda MRP II

Metoda MRP II (ang. *Manufacturing Resource Planning*) jest rozbudowana w stosunku do metody MRP I o bilansowanie zapotrzebowania ze zdolnościami produkcyjnymi CRP (ang. *Capacity Requirements Planning*). Głównym celem rozszerzenia możliwości podstawowej metody MRP I jest redukcja kosztów stałych produkcji ze szczególnym naciskiem na eliminację kosztów z tytułu występowania nadgodzin [37]. Dodatkowo, w metodzie MRP II

w horyzoncie planowania wyodrębniono trzy poziomy uszczegółowienia w procesie planowania produkcji [136]. Od momentu wprowadzenia metody MRP II do użytku praktycznego, planowanie realizowane jest na następujących poziomach uszczegółowienia: planowanie długookresowe, do którego zaliczane jest planowanie zagregowane (poziom ogólny nieomawiany w pracy), planowanie średniookresowe czyli planowanie potrzeb materiałowych według metody podstawowej MRP I w połączeniu z planowaniem CRP oraz planowanie krótkookresowe jako szczegółowe planowanie zleceń produkcyjnych na poziomie operacyjnym. Według twórców metody II wszystkie poziomy są zintegrowane tworząc tzw. zamkniętą pętlę MRP [128][181]. Schematycznie pokazano to na Rys. 4.



Rys. 4. Schemat ideowy zamkniętej pętli MRP [128].

Obciążenie stanowisk roboczych jest sprawdzane na poziomie planowania średniookresowego. Opracowany bilans zapotrzebowania ze zdolnościami produkcyjnymi pozwala zgrubnie oszacować realność przygotowanego planu potrzeb materiałowych MRP. Jeżeli istnieje możliwość dokładnej kalkulacji zdolności produkcyjnych poszczególnych komórek produkcyjnych, wówczas można stwierdzić czy plan MRP wymaga korekty. Dokładne oszacowanie zdolności produkcyjnych jest możliwe w przypadku występowania prostych systemów produkcyjnych. Za prosty system produkcyjny uważa się taki system, w którym występują komórki produkcyjne II stopnia, w których przepływ produkcji jest ukierunkowany (marszruty technologiczne różnych wyrobów przechodzą przez kolejne stanowiska robocze w tej samej kolejności). W literaturze proste systemy produkcyjne nazywa się systemami *flow-shop* [36][172].

W przypadku prostych systemów produkcyjnych, zasadniczy problem stanowi obliczenie właściwej wielkości produkcji (ang. *multi-level capacitated lot-sizing problem*) [4][42][61]. Jest to problem optymalizacyjny, którego rozwiązanie zostało zaimplementowane w systemach APS [71] opisywanych w dalszej części analizy literatury. Niemniej jednak problem nie stanowi celu niniejszej pracy naukowej, dlatego nie będzie dalej opisywany.

Zasadniczym problemem jest kalkulacja zdolności produkcyjnych w systemach złożonych. Do takich systemów zalicza się systemy zadaniowe znane w literaturze jako systemy *job-shop* [91][172]. Systemem zadaniowym może być zakład obróbki mechanicznej. Zakład obróbki mechanicznej (głównie skrawaniem) jest uważany za system złożony ze względu na mnogość technologii obróbki zwykle stosowanych w celu wytworzenia określonych wyrobów. Z punktu widzenia przepływu produkcji, największy problem stanowi nieukierunkowanie marszrut technologicznych. W związku z tym wymagane jest szczegółowe planowanie zleceń DDA (ang. *due date assignment problem*). Kalkulacja zdolności produkcyjnych odbywa się na zasadzie szacowania terminów realizacji zleceń. Dotychczas opracowane metody szacowania terminów realizacji zleceń zostały opisane w kolejnym rozdziale.

2.2. Problemy planowania terminów realizacji zleceń w złożonych systemach produkcyjnych

Szczegółowe planowanie zleceń służy do oszacowania terminów, kiedy poszczególne zlecenia produkcyjne mogą najwcześniej być zrealizowane w danym systemie wytwórczym (termin realizacji zlecenia produkcyjnego: ang. *due date*). Szacowanie terminu realizacji zleceń odbywa się co najwyżej na szczeblu jednego wydziału produkcyjnego (w którym realizowany jest etap wytwarzania). Jeżeli proces produkcyjny wyrobu dzieli się na kilka etapów wytwarzania, wówczas szacowanie terminów realizacji zleceń wykonuje się dla każdego etapu wytwarzania z osobna [146].

Z punktu widzenia niniejszej pracy, istotne jest w jaki sposób szacuje się terminy realizacji zleceń w etapie wytwarzania. Analiza literatury światowej wskazuje, że problem szacowania terminów realizacji zleceń jest bezpośrednio związany z harmonogramowaniem produkcji [29][56]. Klasyczny problem harmonogramowania produkcji został opisany w kolejnym rozdziale analizy literatury. Natomiast problem planowania terminów realizacji zleceń zmienia sposób harmonogramowania produkcji. W tym przypadku znane są terminy realizacji zleceń, a zagadnienie polega na ustanowieniu takiej kolejki realizowanych zadań na maszynach, aby dodatkowe koszty tworzone w wyniku wcześniejszego lub późniejszego wykonania zadań w stosunku do założonych terminów realizacji zadań był minimalny [139]. Problem planowania terminów realizacji zleceń dzieli się na trzy grupy:

- Problem harmonogramowania produkcji oraz planowania realizacji zleceń wykonywanych na pojedynczych maszynach (ang. *single-machine scheduling problem with due date assignment*). W tym przypadku spotyka się dwa podejścia do rozwiązania problemu planowania realizacji zleceń. Pierwsze z nich dotyczy planowania ze wspólnym terminem realizacji zleceń (ang. *common due date assignment problem*) [8][23][24][113].

- Problem harmonogramowania produkcji oraz planowania realizacji zleceń wykonywanych na maszynach równoległych (ang. *parallel-machine scheduling problem with due date assignment*) [149][157][163].
- Problem harmonogramowania produkcji oraz planowania realizacji zleceń w systemach złożonych (ang. *job-shop scheduling problem with due date assignment*) [160].

W pracy założono występowanie złożonych systemów produkcyjnych. W związku z tym najistotniejszym jest przeanalizowanie, w jaki sposób rozwiązywane są problemy harmonogramowania produkcji oraz planowania realizacji zleceń w systemach złożonych. Okazuje się, że literatura obfituje w prace skupiające się na szacowaniu terminów realizacji zleceń produkcyjnych w systemach złożonych o seryjnym typie produkcji. Analizie poddaje się systemy pracujące przy wysokim obciążeniu zleceniami, sięgającym 80÷90%. Analizuje się czas przepływu poszczególnych zleceń w komórce produkcyjnej. Najczęściej stosowanym narzędziem do analizy przepływu produkcji oraz ustalania terminów realizacji zleceń jest eksploracja danych. Wśród metod stosowanych do eksploracji danych można wyróżnić algorytmy drzew decyzyjnych do poszukiwania wartości czasu przepływu [119][138], algorytm szeregów czasowych do szacowania czasu oczekiwania zleceń produkcyjnych przed maszynami w komórce produkcyjnej [12] oraz algorytmy regresji wielorakiej do analizy zmienności czasu przepływu zleceń produkcyjnych w zależności od liczby zleceń, liczby zadań oraz wielkości produkcji [129][160].

W problemie planowania realizacji zleceń produkcyjnych kluczową kwestią stanowi potwierdzenie terminu realizacji zlecenia produkcyjnego. W kontekście zarządzania produkcją w systemach złożonych, za termin realizacji zlecenia przyjmuje się datę dostawy produktu do klienta. Istnieją dwie grupy metod służących do określenia, czy założony termin realizacji zlecenia może zostać potwierdzony [185]:

- 1) Metody egzogeniczne. W metodzie egzogenicznej termin realizacji zlecenia uważany jest jako niezależny od harmonogramu produkcji. Harmonogram jest tworzony zwykle z założeniem nieprzekraczalnego terminu zakończenia zleceń (np. poprzez harmonogramowanie wstecz), a do klienta przesyłana jest informacja, czy dane zlecenie może być wykonane z założonym terminem czy też nie.
- 2) Metody endogeniczne. Korzystając z metody endogenicznej, najpierw tworzony jest harmonogram według pewnej metody harmonogramowania. Następnie z harmonogramu odczytuje się wynikłe terminy zakończenia zleceń. Do klienta, oprócz potwierdzenia terminu realizacji zlecenia trafia informacja, na kiedy dane zlecenie może zostać najwcześniej zrealizowane.

W literaturze wiele metod zarówno egzogenicznych jak i endogenicznych zostało opisanych [29][56]. W praktyce najczęściej wykorzystywanymi metodami egzogenicznymi są: metoda DDA ze wspólnym terminem realizacji wszystkich zleceń CON (ang. *Common Due Date*) oraz metoda zmiennej zawartości pracy (RWK – ang. *Random Work Content*). Jednak metody egzogeniczne są zdecydowanie rzadziej stosowane niż metody endogeniczne. Do metod endogenicznych zalicza się m. in.: metoda dynamiczna czasu przetwarzania i oczekiwania

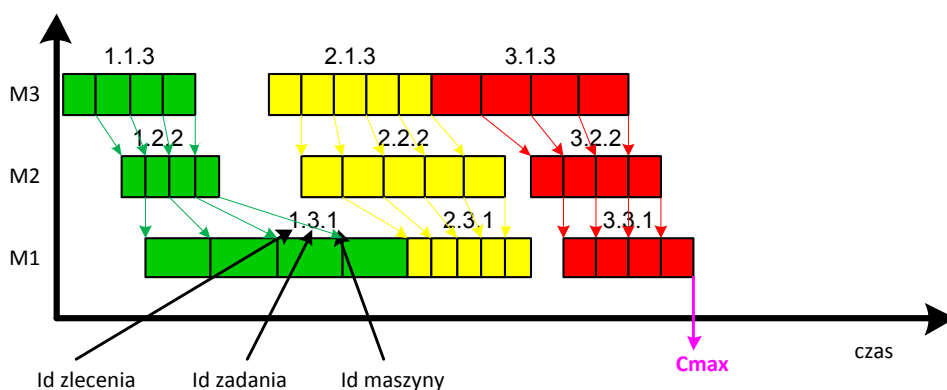
(DPPW – ang. *Dynamic Processing and Waiting Time*), metoda całkowitej zawartości pracy (TWK – ang. *Total Work Content*), metoda dynamiczna całkowitej zawartości pracy (DTWK – ang. *dynamic total work content*) a także metody kolejki zadań (JIQ – ang. *Job in Queue*).

Cechą charakterystyczną wymienionych metod jest to, że do obliczeń wykorzystuje się wskaźniki, których wartości wyznacza się na drodze doświadczalnej za pomocą narzędzi eksploracji danych. Wartości wskaźników wpływają na wartość odsunięcia (opóźnienia) terminu realizacji zleceń w czasie tak, aby wyeliminować w pewnym stopniu zjawisko tzw. zapchania (ang. *congestion*) systemu złożonego. Przez zapchanie systemu złożonego rozumie się stan, gdy zbyt duża liczba zleceń jest realizowana w określonym przedziale czasu. Skutkiem zapchania systemu złożonego są częste przestoje maszyn wynikłe z nieplanowanych zdarzeń, takich jak oczekiwanie, drobne usterki maszyn, spadek wydajności pracy. Wskaźniki te są kluczowe z punktu widzenia wiarygodności systemu wytwarzania w dotrzymywaniu terminów realizacji zleceń. Wiarygodność systemu wytwarzania jest kluczowym czynnikiem brany pod uwagę przy tworzeniu nowych metod planowania przepływu produkcji. Kolejnym zagadnieniem wymagającym analizy literatury jest klasyczny proces harmonogramowania produkcji. Został on opisany w kolejnym rozdziale.

2.3. Metody harmonogramowania zadań w warunkach produkcji małoseryjnej

Problem harmonogramowania produkcji jest od wielu lat przedmiotem intensywnych badań. Jest to jeden z najczęściej poruszanych problemów w literaturze. Można to udowodnić przeglądając platformy naukowe, na których zamieszczone są artykuły poświęcone tematyce z dziedziny inżynierii produkcji i badań operacyjnych. Z punktu widzenia problemu naukowego, problem harmonogramowania produkcji dzieli się na dwie grupy.

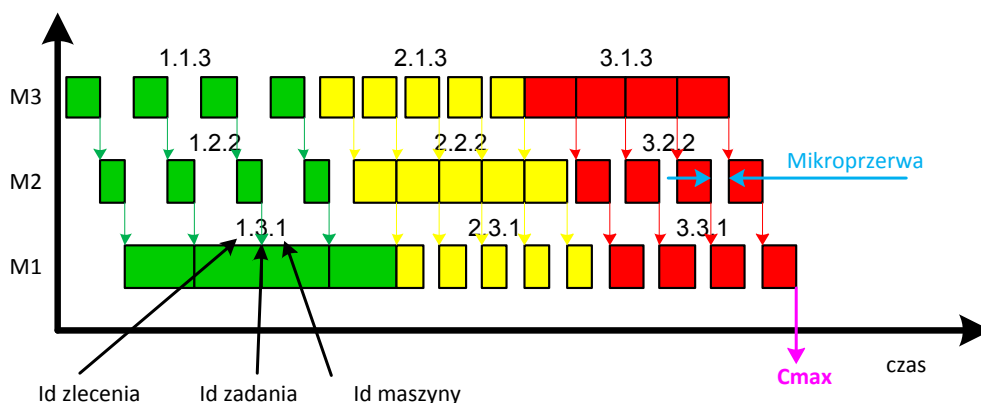
Pierwsza grupa poświęcona jest problemowi harmonogramowania produkcji o przepływie ukierunkowanym (ang. *flow-shop scheduling problem*). Harmonogramowanie produkcji *flow-shop* odbywa się w systemach wytwórczych przepływowych. Harmonogram produkcji w systemach przepływowych najczęściej przybiera postać zaprezentowaną na Rys. 5. Jest to harmonogram z przepływem szeregowo-równoległym, charakteryzujący się przesyłaniem między kolejnymi maszynami partii transportowych.



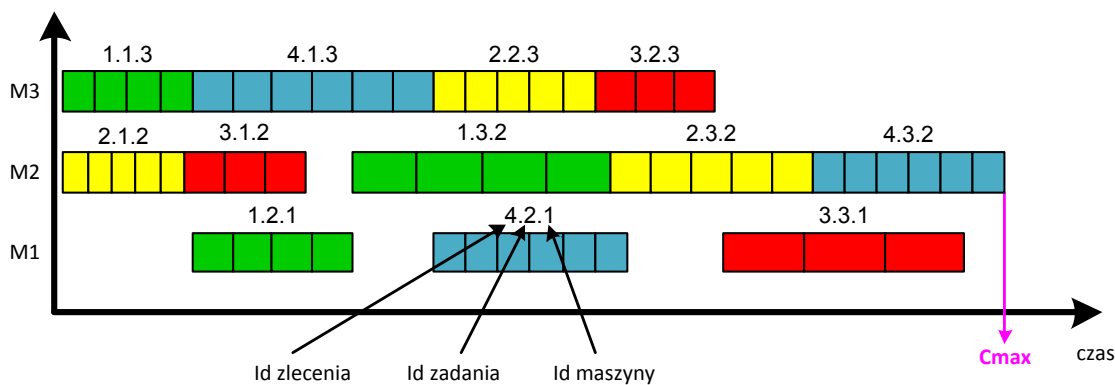
Rys. 5. Przykład harmonogramu z przepływem szeregowo-równoległym stosowanego w systemach przepływowych [opracowanie własne na podstawie [172]].

W systemach przepływowych spotykany jest również harmonogram z przepływem równoległym (Rys. 6). Wadą tego podejścia jest występowanie mikroprzerw pomiędzy realizacją kolejnych partii transportowych na maszynach o większej przepustowości. Z drugiej jednak strony przepływ równoległy jest łatwiejszy do implementacji w system komputerowego wspomaganie harmonogramowania produkcji, czego autor pracy doświadczył w trakcie realizacji badań wstępnych nad tematyką niniejszej pracy.

Druga, zdecydowanie liczniejsza grupa, dotyczy problemu harmonogramowania produkcji nieukierunkowanej, tzw. problem harmonogramowania zadań (ang. *general job-shop* lub w uproszczeniu *job-shop scheduling problem*). Problem ten występuje w systemach wytwórczych zadaniowych. Harmonogram produkcji w systemach zadaniowych najczęściej wygląda tak jak zaprezentowano na Rys. 7.



Rys. 6. Przykład harmonogramu z przepływem równoległym stosowanego w systemach przepływowych [opracowanie własne na podstawie [172]].



Rys. 7. Przykład harmonogramu z przepływem szeregowym stosowanego w systemach zadaniowych [opracowanie własne na podstawie [172]].

W celu potwierdzenia znaczenia problemu harmonogramowania *general job-shop*, sporządzono listę czasopism dostępnych na platformie naukowej Elsevier, w których najczęściej publikowano artykuły z dziedziny problemu harmonogramowania zadań (ang. *general job-shop scheduling problem*). Listę podzielono na okresy czteroletnie, w których dla każdego czasopisma podliczono liczbę publikacji. Liczbę publikacji zestawiono w Tab. 1.

Liczbę publikacji, podzieloną na okresy czteroletnie w Tab. 1, przedstawiono graficznie na wykresie słupkowym. Wykres słupkowy zilustrowano na Rys. 8.

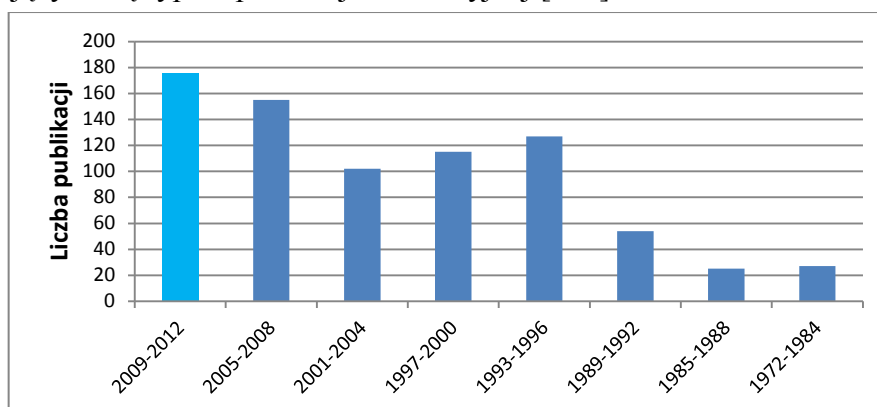
Z wykresu widać ciągły wzrost zainteresowania tematyką *general job-shop scheduling*. Należy zaznaczyć, że liczba dostępnych publikacji na jednej platformie naukowej jest tak

duża (np. w latach 2005-2008 liczba publikacji przekroczyła poziom 150), że niemożliwym staje się szczegółowe przeanalizowanie wszystkich publikacji w tym temacie.

Tytuł czasopisma	Liczba publikacji w czteroletnich przedziałach czasu								Razem
	2009-2012	2005-2008	2001-2004	1997-2000	1993-1996	1989-1992	1985-1988	1972-1984	
Applied Mathematics and Computation	3	10	0	0	0	0	0	0	13
Applied Soft Computing	11	1	3	0	0	0	0	0	15
CIRP Annals - Manufacturing Technology	2	0	1	2	4	0	0	2	11
Computers & Industrial Engineering	38	20	13	19	20	14	9	4	137
Computers & Operations Research	38	24	25	6	7	0	1	2	103
Discrete Applied Mathematics	2	2	6	8	8	0	3	1	30
European Journal of Operational Research	26	48	30	36	42	16	5	5	208
Expert Systems with Applications	29	6	0	1	1	0	0	0	37
International Journal of Production Economics	11	24	10	16	16	1	0	0	78
Journal of Manufacturing Systems	3	3	2	5	6	4	3	3	29
Journal of Operations Management	0	0	0	3	7	7	3	6	26
Mathematical and Computer Modelling	4	5	1	3	7	2	0	0	22
Omega	4	1	4	5	5	7	1	4	31
Operations Research Letters	0	4	4	10	3	2	0	0	23
Robotics and Computer-Integrated Manufacturing	5	7	3	1	1	1	0	0	18
Razem	176	155	102	115	127	54	25	27	781

Tab. 1. Liczba publikacji na platformie naukowej Elsevier zwróconych na hasło: „job shop scheduling” [opracowanie własne].

W pracy główną uwagę poświęcono tematyce harmonogramowania zadań (*general job-shop scheduling*), gdyż taki problem będzie głównie występował w systemach wytwórczych charakteryzujących się typem produkcji małoseryjnej [142].



Rys. 8. Prezentacja liczby publikacji na temat harmonogramowania zadań [opracowanie własne].

Chociaż problem harmonogramowania *general job-shop* jest jednym z najczęściej poruszanych i rozwiązywanych problemów w literaturze, to zgodnie z tezą M. Pawlaka stwierdza się, że problem jest niemal wyłącznie przedmiotem badań teoretycznych [121]. Głównie rozpowszechnione są tzw. problemy testowe, które służą za podstawę odniesienia przy ocenie uzyskanych wyników z prowadzonych badań. Do najbardziej znanych problemów testowych należą te sformułowane przez Fishera oraz Thomsona w roku 1963. Są to przykłady testowe o wymiarach 10×10 . Wymiar 10×10 oznacza 10 zleceń składających się z 10 zadań. Realizacja każdego zlecenia wymaga zastosowania 10 różnych maszyn. Maszyny nie powtarzają się w ramach wykonania jednego zlecenia. W przykładzie należy wyznaczyć taką kolejność wykonania poszczególnych zadań na maszynach, aby uzyskać najmniejszy czas zakończenia wszystkich zleceń.

Problem harmonogramowania *general job-shop* jest problemem optymalizacyjnym NP-trudnym. Jak wynika z powyższego tekstu, za kryterium optymalizacyjne przyjmuje się zwykle czas zakończenia wszystkich zleceń. Można jednak posługiwać się innymi kryteriami oceny harmonogramów. Kryteria oceny harmonogramów szczegółowo sklasyfikowano w książkach polskich autorów [91][121][143].

Z punktu widzenia tematu pracy bardziej istotne jest w jakich warunkach podejmowane są działania optymalizacyjne podczas harmonogramowania *general job-shop*. Okazuje się, że rozwiązania teoretyczne bazują na dawno sformułowanych warunkach ograniczających, do których należą [48]:

1. Zadania pochodzące od jednego zlecenia nie mogą być wykonywane równolegle. Ograniczenie to dyskryminuje możliwość wytwarzania wyrobów złożonych, w których najpierw wytwarza się poszczególne części a następnie przystępuje się do montażu tych części w wyrób.
2. Na jednej maszynie nie można wykonywać dwóch zadań równocześnie. Założenie powoduje eliminację możliwości realizacji procesów wsadowych (np. obróbka cieplna części z różnych zleceń).
3. Każde zlecenie posiada stałą liczbę zadań. Założenie jest całkowicie niezgodne z rzeczywistością i nie wymaga komentarza.
4. Każde zlecenie musi być wykonane do końca. Założenie zgodne z praktyką, gdyż rzadko występują sytuacje, w których przerwanie realizacji zlecenia jest jednoznaczne z jego zaniechaniem.
5. Czasy wykonania zadań są niezależne od harmonogramu. Założenie słuszne, gdyż w warunkach przemysłowych nie występują sytuacje, w których opóźnienie terminu rozpoczęcia zadania może spowodować pogorszenie się warunków pracy i tym samym wydłużenie czasu przetwarzania zadania.
6. Dopuszczalne jest oczekiwanie zadania na zwolnienie maszyny, na której ma być wykonane.
7. Jest tylko jedna maszyna danego typu. Założenie jest sprzeczne z rzeczywistością. Przeważnie zakłady przemysłowe posiadają park maszynowy składający się z wielu maszyn z czego można pogrupować maszyny według typu technologicznego (maszyny tego samego typu technologiczne są w stanie podjąć określone zadanie).
8. Maszyny nigdy się nie psują i są dostępne w ciągu całego czasu realizacji zleceń. Założenie zupełnie nie odpowiada rzeczywistości, w której zdarzają się zarówno awarie maszyn, absencje pracowników, jak i planowane przestoje z powodu remontów.
9. Nie występuje czynnik losowości, w szczególności znane i ustalone są: liczba maszyn, liczba zleceń, czasy gotowości. Wątpliwość budzi ustalona liczba zleceń. W trakcie wizyt autora w zakładach przemysłowych charakteryzujących się typem produkcji małoseryjnej wyraźnie podkreślano wpływ zleceń nagłych (tzw. priorytetowych), które powodują zaburzenie wygenerowanego harmonogramu zadań.

10. Niedopuszczalne jest wariantowe zapisanie technologii. Kolejne założenie, które jest niezgodne z rzeczywistością. Istnieje możliwość stosowania alternatywnych marszrut technologicznych. Planista powinien mieć możliwość podjęcia decyzji, którą marszrutę wybrać w celu realizacji zlecenia produkcyjnego.

Powyżej sformułowano 10 kluczowych założeń silnie ograniczających możliwość implementacji proponowanych rozwiązań w praktyce. W ostatnich latach zostały podjęte próby eliminowania poszczególnych ograniczeń w problemie harmonogramowania *general job-shop*. Powstały nowe nurty naukowe rozwijane na świecie, do których zalicza się:

1. *Job shop scheduling with alternative process plans*. Nurt badań ma za zadanie eliminację ograniczenia nr 10. Zakłada się, że istnieją alternatywne procesy technologiczne, a zadanie polega na wybraniu takich procesów, aby zminimalizować wartość określonej funkcji celu. Zwykle dąży się do minimalizacji terminu zakończenia wszystkich zadań [77][118][156].
2. *Flexible job-shop scheduling problem*. Jest to nurt badań eliminujący ograniczenie nr 7. W problemie *flexible job-shop scheduling* każde zadanie może zostać przydzielone do maszyny ze zbioru kilku maszyn. Problem ten jest bardziej złożonym problemem optymalizacyjnym od tradycyjnego problemu *general job-shop scheduling*. Do rozwiązania problemu *flexible job-shop scheduling* stosuje się wszelkiego rodzaju metaheurystyki, takie jak algorytm genetyczny [58], algorytm roju cząstek [100], algorytm mrówkowy [177], przeszukiwanie tabu [88], poszukiwania za zmiennym otoczeniem [178].
3. *Stage shop scheduling problem*. Ten nurt badań pozwala wyeliminować pierwsze ograniczenie. W problemie *stage shop scheduling*, głównym założeniem jest istnienie wyrobów złożonych, które są produkowane w systemie wytwórczym. Problem *stage shop scheduling* jest rozwiązywany za pomocą różnych metod, m. in. algorytmów genetycznych [107], programowania z ograniczeniami [182], programowania całkowitoliczbowego [81].
4. *Job shop scheduling with deteriorating jobs problem*. Nurt badań dotyczy zjawiska pogarszania się warunków pracy wraz z upływem czasu. Eliminuje się więc ograniczenie nr 5. Choć pojawiło się kilka prac w zakresie harmonogramowania *job-shop* z wykorzystaniem zjawiska pogarszających się warunków pracy [99][162], to nurt badań jest bardziej wykorzystywany przy harmonogramowaniu *flow-shop* oraz harmonogramowaniu pojedynczych maszyn (*ang. single machine scheduling problem*).

Przedstawione nowe nurty badań nad problemem harmonogramowania *general job-shop* dają możliwość ich zastosowania w praktyce. Jednakże odnosząc się do 10 wymienionych warunków ograniczających zadań optymalizacyjnych harmonogramowania, nie wszystkie z tych ograniczeń zostały wyeliminowane. W przypadku problemu harmonogramowania *general job-shop* w przemyśle maszynowym, szczególnie istotne jest wyeliminowanie ograniczeń nr 8 oraz 9. W ostatnim czasie w literaturze pojawiły się publikacje, w których autorzy próbują eliminować ograniczenie nr 8 [2][43].

Ponadto, istnieje jeszcze jedno ukryte założenie problemu harmonogramowania *general-job shop*. Mówi ono, że pewne zadanie musi być zakończone, aby zadanie po nim następujące w marszrucie technologicznej mogło być rozpoczęte. Harmonogramowanie *general job-shop* umożliwia więc wyłącznie tworzenie harmonogramów szeregowych, gdyż z założenia metody harmonogramowania *general job-shop* w pierwszej kolejności dążą do zaszeregowania oddzielnych zadań. Nie ma możliwości budowy harmonogramów z przepływem równoległym, które pozwalają skrócić czas realizacji zleceń produkcyjnych.

2.4. Rozwój komputerowych systemów planowania i harmonogramowania produkcji

W niniejszym rozdziale omówiono historię rozwoju oraz bieżącą funkcjonalność komputerowych systemów planowania produkcji w celu dokonania analizy ich przydatności do planowania produkcji małoseryjnej. Treść niniejszego rozdziału zaczerpnięto z artykułów publikowanych w czasopiśmie listy filadelfijskiej oraz z ogólnodostępnych prospektów i dokumentacji technicznych komercyjnych systemów.

Rozdział podzielono na trzy podrozdziały. W pierwszym podrozdziale przedstawiono zarys historyczny rozwoju systemów komputerowych planowania produkcji, w których wykorzystano metody MRP I, MRP II [69][92]. W drugim podrozdziale omówiono zintegrowane systemy planowania ERP [15][39][69][79][80][98]. W trzecim podrozdziale opisano najnowszą, obecnie rozwijaną generację systemów planowania zwaną APS [38][63][147].

2.4.1. Systemy planowania zawierające metody MRP I oraz MRP II

Pierwsze próby zastosowania systemów komputerowych do planowania produkcji datuje się na lata pięćdziesiąte ubiegłego wieku. Wówczas panowało przekonanie o słuszności produkcji na wielką skalę według koncepcji ekonomii skali zaproponowanej przez H. Forda [45]. Pierwsze komputerowe systemy planowania produkcji służyły do obliczania ekonomicznej wielkości produkcji i punktu ponownego zamawiania.

Koncepcja ekonomii skali nie nadawała się do wdrożenia w przemyśle maszyn rolniczych, mocno narażonym na wahania popytu. Za sprawą dr Orlickiego pracującego w zakładach J.I. Case rozpoczęto pracę nad budową systemu komputerowego planowania według metody MRP. W porozumieniu z IBM w roku 1961 powstał prototyp systemu MRP o nazwie 'PICS' wykorzystujący do pracy jednostkę IBM RAMAC 305 z zapisem danych na taśmach magnetycznych. Na taśmach magnetycznych zapisywano dane o stanie zapasów w tygodniowym okresie planowania. Rola systemu 'PICS' sprowadzała się do planowania produkcji wyrobu według jego listy materiałowej BOM (ang. *Bill of Materials*), przeliczenia wielkości produkcji oraz ustalenia poziomu zapasów w kolejnym okresie planowania na podstawie danego zapotrzebowania i stanu zapasów z poprzedzającego okresu planowania.

Lata 70-te zaowocowały znaczącym postępem w dziedzinie miniaturyzacji komputerów oraz wzrostem mocy obliczeniowej i pojemności nośników informacji. Rozwój komputerów spowodował wzrost zainteresowania informatyką. W efekcie w roku 1970 E. Codd zaproponował relacyjny model danych. Na bazie tego modelu w roku 1976 powstały pierwsze prototypy relacyjnych baz danych (Ingres, System R). Relacyjne bazy danych, tj. ich rozwiązania

komercyjne stały się później centralnymi jednostkami gromadzenia i przetwarzania informacji komputerowych systemów planowania.

W latach 70-tych narodziły się największe korporacje informatyczne produkujące komputerowe systemy planowania. W roku 1972 grupa pięciu inżynierów oprogramowania z Mannheim w Niemczech założyła SAP (niem. *Systemanalyse und Programmentwicklung*). Celem istnienia SAP było pierwotnie tworzenie standardowych i zintegrowanych rozwiązań informatycznych wspierających działalność biznesową. W roku 1975 założono Lawson Software. Celem działalności firmy miało być wspieranie przedsiębiorstw produkcyjnych poprzez tworzenie i wdrażanie dedykowanych rozwiązań biznesowych. W roku 1977 powstały J. D. Edwards oraz Oracle Corporation. Oracle jako pierwszy zaoferował w roku 1979 komercyjny system do zarządzania relacyjną bazą danych, który 20 lat później zdominował rynek wdrażając platformę systemów komputerowych planowania w ponad 40000 przedsiębiorstw na świecie. W roku 1978 Jan Baan założył w Holandii The Baan Corporation dostarczając rozwiązania z zakresu zarządzania administracją i serwisem w przedsiębiorstwach produkcyjnych. Ponadto w roku 1975 IBM zaoferował system MMAS (ang. Manufacturing Management and Account System), pierwszy system modułowy integrujący prognozowanie zapotrzebowania z podstawową metodą MRP I.

W roku 1983 J. D. Edwards opracował system pracujący pod IBM System/38, który jako pierwszy na świecie posiadał zaimplementowaną metodę MRP II. Oprogramowanie przeznaczone było dla producentów należących do grupy małych i średnich przedsiębiorstw. System dokonywał planowania produkcji na różnych poziomach uszczegółowienia, począwszy od planowania zagregowanego przez planowanie potrzeb materiałowych, które następnie podlegały bilansowaniu zapotrzebowania ze zdolnościami produkcyjnymi a skończywszy na szczegółowym planowaniu zleceń. Dodatkowo poszerzono funkcjonalność systemu o zarządzanie jakością włączając narzędzia jakościowe promowane przez Deminga, Jurana, Crosby'ego czy też Ishikawę.

W roku 1988 IBM zaktualizował oprogramowanie, wypuszczając na rynek system CO-PICS. System został zbudowany według nowej koncepcji o akronimie CIM (ang. Computer-Integrated Manufacturing), która miała oferować w pełni zintegrowane narzędzia komputerowe do zarządzania przepływem informacji w przedsiębiorstwie. System posiadał trzy warstwy obsługi. Pierwsza warstwa wspomagała pracę w zakresie marketingu, technicznego przygotowania produkcji, planowania przepływu produkcji, zarządzania operacjami oraz zarządzania zamówieniami. Drugą warstwę stanowiły procedury i algorytmy wspomaganie decyzji. Trzecia warstwa zawierała bazę danych oraz mechanizmy zarządzania danymi oraz komunikacji użytkowników w systemie. Koncepcja CIM powodowała znaczny wzrost wielkości systemów komputerowych planowania. Nie było to korzystne, gdyż nie zawsze w przedsiębiorstwach wymagano stosowania kompleksowego narzędzia informatycznego. Niedogodność ta przyczyniła się do powstania nowej generacji systemów komputerowych planowania. Powstały zintegrowane systemy informatyczne o akronimie ERP, służące do zarządzania przedsiębiorstwem zarówno w zakresie produkcyjnym jak i finansowym.

2.4.2. Systemy klasy ERP

Termin ERP (ang. *enterprise resource planning*) użyto po raz pierwszy w roku 1990 w Gartner Group, firmie doradczo-opiniującej o wysokiej renomie w zakresie technologii informacyjnych. Termin ten miał identyfikować systemy charakteryzujące się wysoką integracją działań w zakresie planowania dostaw, produkcji, dystrybucji oraz kompleksową kontrolą zapasów w ogniwie łańcucha dostaw. Pierwszym systemem spełniającym te wymagania był SAP R/3 wypuszczony na rynek w roku 1992. SAP R/3 stał się szybko wiodącym programem w dziedzinie komputerowych systemów planowania produkcji i zarządzania zasobami przedsiębiorstw przemysłowych. W roku 1998 ilość wdrożeń przekroczyła 15000 przynosząc dochód firmie w wysokości ponad 4 mld €.

SAP R/3 został opracowany w nowej architekturze *client/server* co pozwoliło instalować system na różnych platformach operacyjnych. Ponadto system zbudowano w architekturze otwartej, co umożliwiło firmom zewnętrznym opracowanie dodatków poszerzających funkcjonalność systemu w zakresie zarządzania produkcją (w szczególności zarządzania zleceniami produkcyjnymi oraz szacowaniem terminów realizacji zleceń) w zależności od indywidualnego zapotrzebowania.

Wysoka elastyczność systemów ERP objawiająca się możliwością wdrożenia wybranych modułów i dodatków (w szczególności z zakresu zarządzania i organizacji produkcji) nie zawsze była odpowiednio wykorzystana przez klientów tych systemów. Główne przyczyny niepowodzenia przy wdrażaniu systemu ERP identyfikuje się ze źle przygotowanym procesem przystosowywania przedsiębiorstwa do sposobu przetwarzania danych biznesowych w systemie ERP. Ogólnie uważa się, że proces przystosowywania przedsiębiorstwa dziesięciokrotnie przewyższa koszt zakupu samego systemu ERP. Ponadto, przyjęte terminy i koszty realizacji projektu i wdrożenia systemu ERP częstokroć są przekraczane, co skłania przedsiębiorstwa do wycofania się z projektu nawet w trakcie jego realizacji.

Przykładowo, niepowodzenia przy wdrażaniu systemu ERP doświadczyły takie korporacje produkcyjne jak *Dell Computer*, *Boeing*, *Mobil Europe*, *Kellogg's*, *Hershey*, *Dow Chemical* czy też *FoxMeyer Drug* oraz *Shane*. Dwie ostatnie ogłosiły upadłość w wyniku źle zaplanowanego wdrożenia. *FoxMeyer Drug* wytoczył proces sądowy opiewający na 500 mln \$, obwiniając SAP za błędy systemu SAP R/3 w zakresie planowania dostaw i kosztów produkcji i w konsekwencji o doprowadzenie do bankructwa [13].

W literaturze wyodrębniono najważniejsze czynniki wpływające na osiągnięcie sukcesu przy wdrażaniu systemu ERP. Opracowano pięcioetapowy proces implementacji ERP. Kolejne etapy procesu implementacji to [158]:

- 1) przygotowanie projektu – obszerny proces planowania projektu wdrożenia systemu ERP obejmujący dobór kadry biorącej udział przy wdrożeniu, ustalenie docelowego budżetu, opracowanie harmonogramu prac projektowych,
- 2) przyjęcie strategii przystosowywania przedsiębiorstwa – szczegółowa analiza reguł biznesowych panujących w przedsiębiorstwie, która ma na celu wybór najlepiej dopasowanego systemu ERP,

- 3) testowanie wybranego systemu ERP – skupienie uwagi na opracowaniu instrukcji i procedur do dalszego szkolenia kadry przedsiębiorstwa w zakresie nowej funkcjonalności systemu ERP,
- 4) przygotowanie końcowe wdrożenia – sprawdzenie poziomu integracji w obiegu i przetwarzaniu dokumentów na bazie danych rzeczywistych przedsiębiorstwa,
- 5) przedłożenie sprawozdania końcowego, rozpoczęcie przeprowadzania szkoleń, wsparcie techniczne.

Dynamiczny rozwój systemów ERP oraz powstanie wielu przedsiębiorstw będących partnerami biznesowymi zajmującymi się kwestiami wdrażania i serwisowania systemów ERP spowodowało, że w roku 2000 poziom udanych implementacji tych systemów osiągnął 33% [25] (wyjątek stanowiły Chiny, gdzie ze względów kulturowych poziom udanych implementacji systemów ERP wynosił zaledwie 10% [183]). Przykładowo udane wdrożenia systemu SAP R/3 zanotowały największe korporacje w dziedzinie oprogramowania, tj. Microsoft oraz IBM.

Wielomodułowa struktura systemów ERP doprowadziła do stanu, w którym liczba wdrożeń w przedsiębiorstwach produkcyjnych zrównała się z liczbą wdrożeń w przedsiębiorstwach usługowych. Systemy ERP stały się systemami zorientowanymi na rachunkowość, w których moduł produkcyjny przestał odgrywać decydującą rolę. Za podstawową funkcjonalność systemu ERP zaczęto uważać księgowość i zarządzanie kadrami (najlepszym przykładem jest połączenie J.D.Edwards oraz PeopleSoft, a następnie wrogie przejęcie tych dwóch przedsiębiorstw przez Oracle).

Dalszy rozwój systemów ERP był związany z ich rozbudową o zarządzanie łańcuchem dostaw (moduł SCM) oraz zarządzanie relacjami z klientem (moduł CRM). Systemy ERP ewoluowały do ERP II. W systemie ERP II przedsiębiorstwo jest postrzegane jako ogniwo łańcucha dostaw, które wraz z innymi przedsiębiorstwami tworzy sieć producentów, hurtowników i dystrybutorów. Relacje pomiędzy przedsiębiorstwami w łańcuchu dostaw nazywane są B2B (ang. *Business to Business*). System ERP II odpowiada za komunikację przedsiębiorstw w łańcuchu dostaw. Komunikacja taka (na podstawie zdefiniowanych relacji) odbywa się drogą elektroniczną poprzez internet i określa się ją często mianem e-biznesu.

Od roku 2005 rynek światowy w zakresie komputerowych systemów planowania produkcją klasy ERP II został zdominowany przez dwa systemy: SAP R/3 oraz Oracle Enterprise One. W Polsce dwa wyżej wymienione systemy zajmują dwa czołowe miejsca w rankingu udziału systemów ERP w rynku, chociaż znaczny udział w rynku mają również Comarch CDN XL oraz IFS Application. Szczególnie zauważalny jest wzrost liczby wdrożeń systemu Comarch CDN XL/Optima w sektorze małych i średnich przedsiębiorstw.

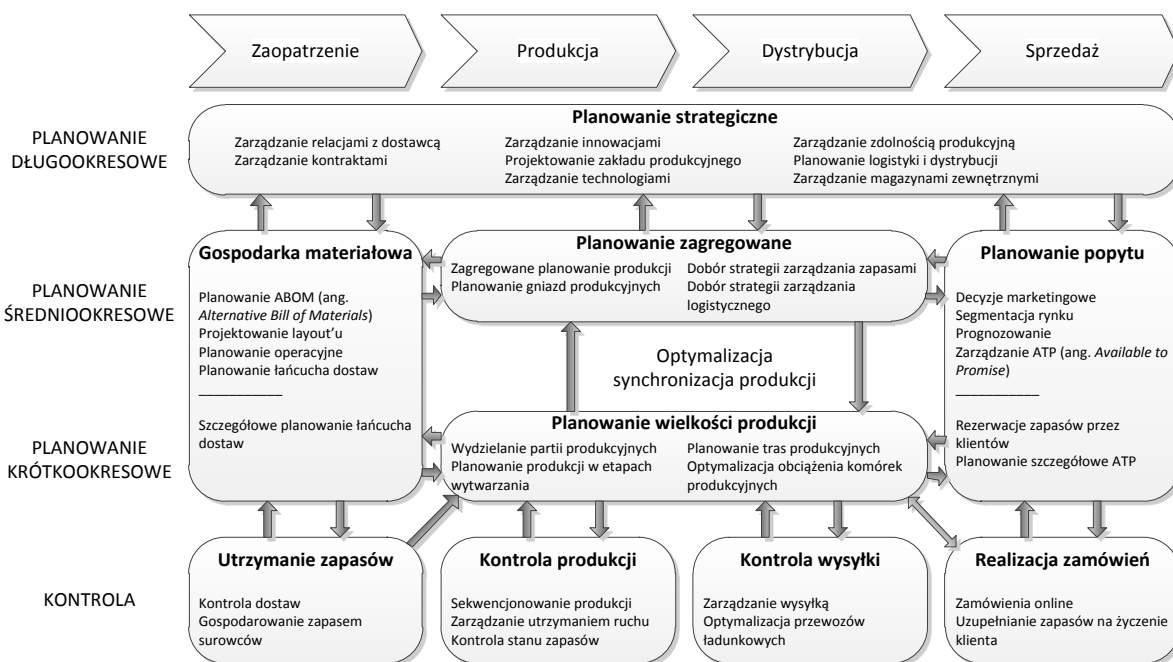
Chociaż systemy klasy ERP / ERP II ciągle są rozwijane to w zakresie planowania produkcji aż do roku 2000 standardowo były instalowane moduły produkcyjne działające w oparciu o planowanie w zamkniętej pętli MRP (pokazanej na Rys. 4). Zamknięta pętla MRP sprawdza się jako metoda planowania produkcji w wielu branżach przemysłowych. Natomiast w warunkach produkcji w systemach job-shop działanie systemu MRP II, zwłaszcza na poziomie planowania krótkookresowego, jest bezskuteczne. Powodem jest brak modułów harmonogramowania produkcji job-shop zaimplementowanych w system ERP. Taki stan rzeczy

zgadza się z praktyczną obserwacją wdrożeń systemów ERP w polskich zakładach przemysłu maszynowego. Okazuje się, że moduły planowania szczegółowego produkcji, zaimplementowane w systemy ERP, są wycofywane w fazie wdrożenia systemu ERP do przedsiębiorstw produkcyjnych.

W odpowiedzi na brak zainteresowania przedsiębiorstw produkcyjnych modułami do szczegółowego planowania produkcji, około roku 2000 powstała nowa koncepcja systemu zintegrowanego planowania produkcji. Została zaprezentowana koncepcja systemu zaawansowanego planowania APS, który ewoluował z MRP II [38].

2.4.3. Systemy APS

System zaawansowanego planowania APS (ang. *Advanced Planning Systems*) jest zbiorem zaawansowanych narzędzi symulacyjnych i optymalizacyjnych do analizy i wspomagania decyzji w planowaniu dostaw, produkcji i dystrybucji. System APS nie ma wbudowanych modułów finansowych, więc powinien być integrowany z systemami ERP. Zasadniczym celem działania APS jest zautomatyzowanie pracy w zakresie planowania produkcji w systemie ERP [147]. Ogólną architekturę systemu APS (podział na moduły funkcjonalne) ilustruje Rys. 9 [41].



Rys. 9. Architektura systemu APS na przykładzie SAP APO [41].

Systemy APS są systemami analitycznymi. Do obliczeń wykorzystują zaawansowane algorytmy optymalizacyjne bazujące na programowaniu liniowym (np. z zastosowaniem metody podziału i ograniczeń) [27] oraz algorytmach genetycznych [26][87]. Zadaniem systemu APS jest ustalenie, dla zadanego zapotrzebowania, ograniczeń systemu wytwórczego. APS potrafi rozróżnić wewnętrzne i zewnętrzne ograniczenia. Na zewnętrzne ograniczenia systemu wytwórczego składają się ograniczenia łańcucha dostaw. Wewnętrznymi ograniczeniami systemu wytwórczego są zdolność produkcyjna wyrażona przez wąskie gardła systemu, dostępność materiałów do produkcji oraz czas realizacji zleceń produkcyjnych. Na podstawie

ustalonych ograniczeń, system APS generuje plan produkcji a następnie doskonalą go za pomocą wybranych metod optymalizacyjnych i symulacyjnych.

Rozwój systemów APS rozpoczął się w latach 90-tych. Natomiast w roku 2000 powstały pierwsze wersje systemów APS stanowiące moduły zintegrowane z systemami ERP. Największymi dostawcami systemów APS są SAP, twórca oprogramowania SAP APO (APO – ang. *Advanced Planning Optimizer*) oraz Oracle, twórca oprogramowania Oracle VCP (VCP – ang. *Value Chain Planning*).

Wdrożenie systemu APS w niektórych przedsiębiorstwach (np. Kellogg) spowodowało wzrost dokładności w procesie planowania realizacji zamówień z 79% do nawet 99%, redukcję czasu realizacji zleceń o 50% oraz redukcję kadry planistów o 30%. Jednak z przeprowadzonych badań empirycznych [63] wynika, że jedynie 20% wdrożeń systemów APS zakończyło się sukcesem. Za główną przyczynę niepowodzeń przy wdrożeniu systemu APS uważa się brak wsparcia ze strony producentów systemów APS pod względem przeprowadzania szkoleń w zakresie właściwego sposobu gromadzenia i przetwarzania informacji.

Ogólna architektura systemów APS, pokazana na Rys. 9, obowiązuje w systemach dwóch największych dostawców oprogramowania tego typu. Architektura w zaprezentowanej formie obejmuje planowanie w szerokim zakresie horyzontu planowania (od planowania długoterminowego do planowania szczegółowego oraz kontroli). Szeroki zakres zastosowania systemów APS w planowaniu produkcji wpływa na wygórowaną cenę, przy czym cena zakupu samej licencji oprogramowania (cena zakupu licencji wynosi ok. \$500.000) stanowi ok 10% całkowitego kosztu zakupu i wdrożenia systemu do przedsiębiorstwa [122].

W celu ograniczenia ceny zakupu oraz kosztów wdrożenia systemów APS powstały tańsze wersje tych systemów zawierające wybrane moduły. Z punktu widzenia tematyki pracy, najważniejszą grupę stanowią uproszczone systemy APS służące do harmonogramowania produkcji. Najpopularniejszymi systemami z tej grupy są: brytyjski Preactor APS, japoński Asprova APS oraz amerykański Taylor APS. Głównym zadaniem stawianym przed tymi systemami jest budowa harmonogramów produkcji dla systemów ERP.

Podstawowym zadaniem uproszczonych systemów APS jest harmonogramowanie produkcji. Wszystkie systemy APS tego typu stosują do tego celu standardową metodę harmonogramowania produkcji. Jest to metoda szeregowania zadań wykorzystująca podstawowe reguły priorytetu zestawione w Tab. 2.

W kontekście systemów APS, zastosowanie wymienionych reguł priorytetu ogranicza się do wyboru zadań, które mają być wykonane na określonej maszynie z zachowaniem ograniczeń technologicznych. Oznacza to, że harmonogram produkcji jest tworzony sekwencyjnie na kolejnych maszynach występujących w systemie wytwarzania.

Należy stwierdzić, że sposób budowy harmonogramów w uproszczonych systemach APS jest bardzo prosty. Systemy tego typu nie mają możliwości optymalizacji harmonogramów produkcji ze względu na wybrane kryterium optymalizacyjne. Jak podkreślają producenci wymienionych systemów APS, ich programy są systemami komputerowego wspomaganie w harmonogramowaniu produkcji. Ich dwie podstawowe funkcje to: integracja z ERP oraz czytelność interfejsu w celu szybkiej i przyjemnej obsługi przez planistów przedsiębiorstw przemysłowych.

Akronim reguły priorytetu	Nazwa reguły priorytetu (ang.)	Opis
FIFO	First In First Out	Najpierw wykonaj zadanie należące do zlecenia wstawionego jako pierwsze do zbioru zleceń
LIFO	Last In First Out	Najpierw wykonaj zadanie należące do zlecenia wstawionego jako ostatnie do zbioru zleceń
SPT	Shortest Processing Time	Najpierw wykonaj zadanie posiadające najkrótszy czas przetwarzania
LPT	Longest Processing Time	Najpierw wykonaj zadanie posiadające najdłuższy czas przetwarzania
MWR	Most Work Remaining	Najpierw wykonaj zadanie należące do zlecenia, w którym pozostało najwięcej (w sensie czasu potrzebnego na przetworzenia zlecenia w pozostałych zadaniach) do zrobienia
RAN	Random Selection	Wykonaj zadanie wybrane losowo spośród dostępnych zadań
EDD	Earliest Due Date	Najpierw wykonaj zadanie należące do zlecenia a najwcześniejszym terminie realizacji, wynikającym z zamówienia klienta

Tab. 2. Wykaz najczęściej stosowanych reguł priorytetów w uproszczonych systemach APS [121].

2.5. Zastosowanie technologii grupowej do projektowania przepływu produkcji

Systemy ERP/APS są systemami przyszłościowymi. Liczba wdrożeń tych systemów zakończonych powodzeniem stanowi o sile tej technologii informacyjnej. Istnieją nikłe szanse na zastąpienie systemów ERP/APS nowymi systemami informatycznymi z dziedziny zarządzania produkcją. Korporacje informatyczne będą raczej dążyć do integracji systemów ERP z nowymi systemami informatycznymi z zakresu planowania produkcji, umożliwiającymi poprawę wskaźników kosztowych, czasowych oraz jakościowych przedsiębiorstwa produkcyjnego. Przykładowo, ciągłe dążenie do skrócenia czasu realizacji zlecenia produkcyjnego zmusza do poszukiwania nowych rozwiązań metodycznych w dziedzinie harmonogramowania produkcji. Szczególnie istotne jest to w przypadku produkcji małoseryjnej jeżeli istnieje możliwość stwierdzenia, że wybrane zlecenia produkcyjne mogą być produkowane wspólnie w celu osiągnięcia korzyści jakie daje technologia grupowa. W niniejszym rozdziale dokonano analizy literatury z zakresu koncepcji technologii grupowej oraz możliwości jej zastosowania do produkcji seryjnej. Natomiast kolejny rozdział jest wyodrębniony ze względu na poszukiwanie sposobu zastosowania technologii grupowej na potrzeby produkcji małoseryjnej.

Technologia grupowa pierwotnie była stosowana do projektowania komórek o specjalizacji przedmiotowej [83]. Zaprojektowane komórki produkcyjne o specjalizacji przedmiotowej (zwane dalej komórkami przedmiotowymi) są składowymi całego systemu wytwórczego. Praca poszczególnych komórek przedmiotowych była niezależna, tj. każda z nich produkowała wyodrębnioną grupę wyrobów zwanych dalej rodzinami wyrobów.

Wyodrębnione spośród produkowanego asortymentu rodziny wyrobów składają się z wyrobów posiadających podobne cechy konstrukcyjno-technologiczne. Do produkcji poszczególnych rodzin wyrobów dedykuje się maszyny. Zadedykowane maszyny zostają rozlokowane zgodnie z przebiegiem marszruty technologicznej rodziny wyrobów lub jej reprezentanta. Zadedykowane maszyny są rozlokowane w zamkniętej komórce przedmiotowej, w której wytwarza się zwykle tylko jedną rodzinę wyrobów.

Komórka przedmiotowa charakteryzuje się tym, że maszyny wykonujące kolejne operacje marszrutu technologicznej rodziny wyrobów znajdują się w bezpośrednim sąsiedztwie, blisko siebie. Taki układ pozwala zredukować czas poświęcony na transport przedmiotów pomiędzy maszynami. Pozwala również zredukować poziom zapasów produkcji w toku. Ponadto, produkowanie wyrobów o cechach podobieństwa konstrukcyjno-technologicznego pozwala skrócić lub całkowicie wyeliminować czas poświęcony na przebrojenia maszyn. Eliminację czasu przebrojenia można uzyskać stosując specjalne oprzyrządowanie technologiczne dedykowane dla całej rodziny wyrobów [57][82].

Wysoki poziom autonomii komórki przedmiotowej w systemie wytwórczym jest bodźcem motywacyjnym dla operatorów obsługujących maszyny. Operatorzy pracujący w komórce przedmiotowej odpowiadają za utrzymanie zarówno odpowiednio wysokiego poziomu produktywności komórki jak i odpowiednio wysokiego poziomu jakości produkowanych w niej wyrobów. Biorą również czynny udział w ciągłym doskonaleniu pracy komórki przedmiotowej tworząc zespół ściśle kooperujących pracowników [65].

Przypadek systemu wytwórczego, w którym komórki przedmiotowe są zamknięte, tj. wytwarzają w całości wydzielone rodziny wyrobów, jest wyidealizowany i rzadko spotykany w rzeczywistości. Przeważnie występują wyroby, które wymagają wykonania operacji na maszynach przydzielonych do co najmniej dwóch komórek przedmiotowych (zwane dalej niedopasowanymi wyrobami). Z drugiej strony, mogą występować maszyny, na których należy wykonać operacje na wyrobach należących do co najmniej dwóch rodzin wyrobów (zwane dalej niedopasowanymi maszynami). W takiej sytuacji dąży się do minimalizacji ilości niedopasowanych wyrobów i niedopasowanych maszyn, aby zastosowanie technologii grupowej było efektywne [28].

Zadanie to w literaturze przedstawia się jako problem grupowania technologicznego (ang. *GT problem*), który wymaga realizacji procedury grupowania składającej się z następujących kroków [20][52]:

- 1) wydzielanie komórek przedmiotowych (ang. *cell formation problem*),
- 2) projektowanie rozmieszczenia maszyn w wydzielonych komórkach przedmiotowych (ang. *facility layout problem*),
- 3) utworzenie harmonogramów produkcji w wydzielonych komórkach przedmiotowych (ang. *cell scheduling problem*),
- 4) alokacja zasobów takich jak pracownicy, oprzyrządowanie, urządzenia transportowe, magazyny buforowe, do pracy w wydzielonych komórkach przedmiotowych.

Za najważniejszy krok w procedurze grupowania uważa się krok pierwszy, tj. wydzielanie komórek produkcyjnych I stopnia [168]. W dalszej części rozdziału dokonano przeglądu literatury dotyczącej wydzielania komórek przedmiotowych. Pozostałe kroki, tzn. od drugiego do czwartego nie zostały dalej opisane z powodu małej ich przydatności w warunkach zmiennego zapotrzebowania.

Należy podkreślić, iż problem wydzielania komórek przedmiotowych jest problemem optymalizacyjnym, w którym za kryterium optymalizacyjne przyjmuje się:

- 1) minimalizację kosztu transportu międzykomórkowego (minimalizacja ilości maszyn nieprzydzielonych do komórek przedmiotowych, ang. *exceptional elements*),
- 2) minimalizację kosztu transportu wewnątrzkomórkowego (minimalizacja ilości operacji nieprzydzielonych do realizacji na maszynach wewnątrz komórek przedmiotowych, ang. *voids*).

Do wydzielania komórek produkcyjnych I stopnia najczęściej stosuje się podejścia oparte o wykorzystanie macierzy zależności maszyn-wyrobów $[a_{mi}]$ (ang. *machine-part incidence matrix*) [1][33][68]. Macierz $[a_{mi}]$ składa się z elementów binarnych '1' lub '0'. Wartość '1' ('0') elementu macierzy $[a_{mi}]$ oznacza, że maszyna m wykonuje operację (nie wykonuje operacji) na wyrobie i . Wstępne ułożenie maszyn i wyrobów w macierzy $[a_{mi}]$ zazwyczaj nie pozwala zidentyfikować rodzin wyrobów. Elementy macierzy o wartościach '1' są rozproszone po całej macierzy (Rys. 10). Wymagane jest przegrupowanie maszyn i wyrobów tak, aby zgromadzić wyroby o podobnych cechach konstrukcyjno-technologicznych oraz maszyny wytwarzające te wyroby. Stąd, wymaga się dekompozycji elementów macierzy o wartościach '1', aby skupiska na przekątnej macierzy tworzyły układ diagonalny (Rys. 11). Utworzone w ten sposób bloki elementów macierzy o wartościach '1' są kandydatami do budowy komórek przedmiotowych.

		Wyroby j				
		1	2	3	4	5
Maszyny i	1	0	1	0	1	1
	2	1	0	1	0	0
	3	0	1	0	1	0
	4	1	0	1	0	0

Rys. 10. Wygląd macierzy zależności maszyn-wyrobów przed przegrupowaniem [28].

		Wyroby j					
		RW-1		RW-2			
		1	3	2	4	5	
Maszyny i	KP-1	2	1	1	0	0	0
	KP-2	4	1	1	0	0	0
		1	0	0	1	1	1
		3	0	0	1	1	0

Legenda:

RW-1, RW-2 – rodziny wyrobów

KP-1, KP-2 - komórki przedmiotowe

Rys. 11. Wygląd macierzy zależności maszyn-wyrobów po przegrupowaniu [28].

W celu rozwiązania problemu grupowania technologicznego powstały metody, które mogą być zaklasyfikowane do pięciu kategorii [168]:

- 1) metody klastrowania z wykorzystaniem wskaźników podobieństwa, opisane w artykule przeglądowym [179],
- 2) metody podziału grafu, w tym metody sieciowe [47],
- 3) metody programowania matematycznego,
- 4) metody heurystyczne i metaheurystyki, przegląd tych metod można znaleźć w literaturze [50][120],
- 5) metody sztucznej inteligencji.

W Tab. 3 znajduje się wykaz publikacji z ostatnich 10 lat dotyczących problemu wydzielenia komórek produkcyjnych I stopnia. Dla każdej publikacji wykazano, oprócz zastosowanej metody obliczeniowej, czynniki uwzględnione podczas tworzenia modelu optymalizacyjnego [169].

W wykazie zamieszczonym w Tab. 3 zawarto publikacje, w których w sposób klasyczny wydziela się komórki przedmiotowe stosując macierz zależności maszyn wyrobów. Cecha charakterystyczną wykazanych rozwiązań jest to, iż nie uwzględnia się w nich czynnik zmienności zapotrzebowania.

Autorzy opracowania	Metoda obliczeniowa	Czynniki produkcyjne													
		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14
Selim i in., 1998 [137]		✓		✓		✓			✓		✓		✓		✓
Caux i in., 2000 [21]	T31, T45	✓		✓				✓		✓					
Sarker, Xu, 2000 [134]	T14	✓		✓	✓										
Chan i in., 2002 [22]	T41		✓											✓	
Mukattash i in., 2002 [102]	T41		✓	✓				✓							
Ohta, Nakamura, 2002 [114]	T12	✓	✓			✓									
Foulds, Neumann, 2003 [47]	T21		✓					✓	✓						
Zolfaghari, Liang, 2003 [184]	T46							✓		✓					
Kim, Baek, 2004 [76]	T41	✓						✓							
Won, Lee, 2004 [171]	T32			✓											
Boulif, Atif, 2006 [16]	T31, T46	✓	✓	✓						✓					
Defersha, Chen, 2006 [33]	T33, T41	✓		✓		✓	✓	✓	✓	✓		✓			
Foulds i in., 2006 [46]	T33, T41, T44				✓										
Stawowy, 2006 [148]	T41, T46			✓											
Won, Currie, 2006 [170]	T14	✓	✓								✓				
Alhourani, Seifoddini, 2007 [3]	T14	✓	✓					✓							
Vitanov i in., 2007 [161]	T41		✓				✓	✓							
Wu i in., 2007 [176]	T46	✓	✓					✓		✓					
Boulif, Atif, 2008 [17]	T31, T46, T51	✓	✓	✓							✓				
Defersha, Chen, 2008 [34]	T46		✓					✓	✓					✓	
Mahdavi, Mahadevan, 2008 [94]	T41	✓						✓	✓						
Ah kioon i in., 2009 [1]	T33, T41	✓		✓				✓	✓	✓			✓	✓	
Mahdavi i in., 2009 [95]	T46										✓				
Oliveira i in., 2009 [116]	T13			✓						✓					
Tavakkoli-M. i in., 2009 [155]	T45			✓										✓	
Wu i in., 2009 [174]	T41, T45, T46			✓				✓							
Ghosh i in., 2010 [51]	T11, T41		✓												
Naadimuthu i in., 2010 [105]	T12, T51										✓	✓			
Neto, Filho, 2010 [108]	T46	✓								✓					
Nokdehdan i in., 2010 [109]	T46			✓											
Nouri i in., 2010 [112]	T41, T42	✓													
Wu i in., 2010 [175]	T41, T43			✓				✓							
Arkat i in., 2011 [5]	T46			✓											
Chung i in., 2011 [30]	T44	✓	✓						✓	✓					
Rezaeian i in., 2011 [125]	T46, T52			✓											

Czynniki brane pod uwagę w modelach projektowania struktury produkcyjnej:
P1 – sekwencja operacji, P2 – wielkość produkcji, P3 – Ograniczenie wielkości komórek produkcyjnych, P4 – limitowany budżet, P5 – przebrojenia, P6 – wielkość partii produkcyjnej, P7 – alternatywne marszruty technologiczne, P8 – przepustowość maszyn, P9 – powiązanie procesowe maszyn, P10 – oprzyrządowanie, P11 – poziom zapasów, P12 – outsourcing, P13 – częstotliwość dostaw, P14 – umiejętności operatorów

Metody obliczeniowe zastosowane do rozwiązania problemu projektowania struktury produkcyjnej:
Algorytmy klastrowania: T11 – hierarchiczne, T12 – iteracyjne, T13 – widmowe, T14 – na bazie własnej definicji wskaźnika podobieństwa technologicznego wyrobu

Metody podziału grafu: T21 –sieci przepływu produkcji

Metody programowania matematycznego: T31 – metoda podziału i ograniczeń, T32 – programowanie całkowitoliczbowe, T33 – mieszane programowanie całkowitoliczbowe

Algorytmy heurystyczne i metaheurystyki: heurystyki: T41 – własne definicje heurystyk; metaheurystyki: T42 – algorytm bakteryjnej techniki zerowania, T43 – algorytm przepływu wody, T44 – algorytm przeszukiwania tabu, T45 – algorytm symulowanego wywarzania, T46 – algorytm genetyczny, T47 – algorytm przeszukiwania rozproszonego, T48 – rozkład Bendera, T49 – algorytm roju cząstek

Metody sztucznej inteligencji: T51 – z wykorzystaniem logiki rozmytej, T52 – z wykorzystaniem sieci neuronowych

Tab. 3. Wykaz publikacji opracowanych w celu rozwiązania problemu wydzielenia komórek przedmiotowych [169].

W zakresie wydzielenia komórek przedmiotowych, w wymienionych pozycjach literatury w Tab. 3 przyjmuje się podstawowe założenie. Produkowany asortyment oraz wielkość produkcji są ustalone w czasie a wahania zapotrzebowania na wyroby są na tyle niskie, że nie wpływają na zaplanowany przebieg produkcji. Wydajność tak zaprojektowanych komórek przedmiotowych znacząco ulega pogorszeniu przy pojawieniu się rotacji produkowanego asortymentu.

Zmiany zapotrzebowania na produkowane wyroby wymuszają konieczność stosowania nowych metod umożliwiających praktyczne zastosowanie komórek przedmiotowych. Problem wydzielania komórek przedmiotowych w warunkach zmiennego zapotrzebowania został zainicjowany w roku 1995 [127]. Natomiast dopiero w roku 2000 opracowano podstawy metodyczne dynamicznego projektowania struktury produkcyjnej [104]. Wypracowano treści merytoryczne problemu oraz zdefiniowano zasadniczy model matematyczny dynamicznej struktury produkcyjnej. W zamyśle twórcy koncepcji dynamicznych komórek przedmiotowych jest okresowe rekonfigurowanie systemu wytwórczego. Układ i kształt komórek przedmiotowych może ulegać zmianie z okresu planowania na kolejny okres tak, aby dopasować przepływ produkcji do bieżącego zapotrzebowania.

Praca [104] znacząco wpłynęła na dalszy rozwój tej koncepcji. Wszystkie nowe metody w tym temacie nawiązują do podstaw merytorycznych z roku 2000. W Tab. 4 zestawiono publikacje, które pojawiły się po roku 2000.

Autorzy opracowania	Metoda obliczeniowa	Czynniki produkcyjne
Mungwatanna, 2000 [104]	T41, T45	D1, D2, D3
Tavakkoli-Moghaddam et al., 2005 [154]	T44, T45, T46	D1, D2, D3, D4
Safaei et al., 2008 [132]	T41, T45	D1, D2, D4, D10, D11
Aryanezhad et al., 2009 [6]	T41	D1, D2, D5, D6, D7
Wang et al., 2009 [164]	T47	D1, D2, D4, D8, D9
Deljoo et al., 2010 [35]	T46	D1, D2, D3, D4
Mahdavi et al., 2010 [93]	T31	D1, D2, D3, D4, D5, D12
Ghotboddini et al., 2011 [52]	T48	D1, D2, D5, D6
Rezazadeh et al., 2011 [126]	T49	D1, D2, D3, D5, D6
Metody obliczeniowe wykorzystane w problemie dynamicznego projektowania struktury produkcyjnej: jak w tabeli 1 Czynniki brane pod uwagę w modelach dynamicznego projektowania struktury produkcyjnej: D1 – asortyment, D2 – wielkość produkcji, D3 – elastyczność procesu, D4 – elastyczność maszyn, D5 – przydział pracowników, D6 – trening krzyżowy, D7 – niezawodność maszyn, D8 – żywotność komórek produkcyjnych, D9 wielkość komórek produkcyjnych, D10 – sekwencja operacji, D11 – alternatywne marszruty technologiczne, D12 – przepustowość maszyn		

Tab. 4. Wykaz publikacji, w których rozwiązuje się problem wydzielania dynamicznych komórek przedmiotowych [169].

Należy zaznaczyć, że zastosowanie koncepcji dynamicznych komórek przedmiotowych jest możliwe jedynie w przypadku użycia mobilnych maszyn i urządzeń. Niekiedy jednak konieczność przestawienia maszyn może być czasochłonne i kosztowne. Ponadto, jeżeli maszyny wymagają odosobnienia ze względu na specyficzny charakter pracy (np. ze względu na uwarunkowania ergonomiczne, specjalne fundamentowanie) to ich częste przestawienia są niemożliwe. Niedogodności produkcji w tego typu komórkach przedmiotowych doprowadziły do rozwoju nowej formy organizacji produkcji, tj. wirtualnych komórek przedmiotowych.

2.6. *Rozwój koncepcji kształtowania wirtualnych komórek przedmiotowych*

Ze względu na bezpośrednie sąsiedztwo maszyn wykonujących kolejne operacje marszrut technologicznych rodzin wyrobów, komórki przedmiotowe są fizycznie wydzielonymi podsystemami zamkniętymi, wewnątrz których produkuje się kompleksowo wyroby o podobnych cechach konstrukcyjno-technologicznych. Układ komórki przedmiotowej traci swoje zalety w przypadku produkcji małoseryjnej, gdyż częste zmiany produkowanego asortymentu powodują konieczność ciągłego przestawiania maszyn w celu rekonfigurowania komórek przedmiotowych. Powstałe niedogodności przyczyniają się do rozwoju nowej formy organizacji produkcji, zwanej wirtualną komórką przedmiotową.

Wirtualna komórka przedmiotowa jest hybrydową formą organizacji produkcji, w której maszyny są dedykowane do produkowania rodziny wyrobów (tak jak w przypadku komórki przedmiotowej) natomiast maszyny te nie znajdują się w bezpośrednim sąsiedztwie. Maszyny są zwykle zgrupowane według specjalizacji technologicznej (tworzą gniazda o specjalizacji technologicznej zwane dalej komórkami technologicznymi), chociaż mogą również tworzyć układ rozproszony, holoniczny lub fraktalny [10][11][59]. Ze względu na wymienione cechy, wirtualne komórki przedmiotowe uważa się za logicznie wydzielone podsystemy zamknięte, w których produkuje się kompleksowo wyroby o podobnych cechach konstrukcyjno-technologicznych.

Koncepcja kształtowania wirtualnych komórek przedmiotowych pojawiła się w roku 1982 i pierwotnie była rozwijana przez organizację NBS (ang. *National Bureau of Standards*) w celu rozwiązania problemów sterowania w zautomatyzowanych systemach obróbkowych wytwarzających małe partie różnorodnych wyrobów [141]. Maszyny w zautomatyzowanym systemie obróbkowym były lokalizowane w formie rozproszonej. Podsystem sterowania miał za zadanie wydzielić maszyny w celu utworzenia logicznych (wirtualnych) komórek przedmiotowych zaadoptowanych do wytwarzania poszczególnych rodzin wyrobów w zależności od bieżącego zapotrzebowania. Zmiana logicznej konfiguracji systemu wytwórczego wymagała przeprowadzenia czasochłonnych przebrojeń maszyn zautomatyzowanych. Częste zmiany zapotrzebowania były kłopotliwe ze względu na konieczność wykonywania złożonych czynności przygotowawczo-zakończeniowych, dlatego żywotność wirtualnej komórki przedmiotowej wynosiła co najmniej miesiąc.

W roku 1993 zaproponowano nowe podejście do kształtowania wirtualnych komórek przedmiotowych [66]. Założono, że grupa logicznie wydzielonych maszyn zautomatyzowanych nie powinna być zadedykowana do produkowania jednej rodziny wyrobów przez tak długi czas jak miesiąc. Maszyny powinny być ogólnodostępne, natomiast skupiono się nad opracowaniem metody sterowania podsystemem przepływu materiałów. Według autorów pracy [66], podsystem przepływu materiałów ma decydujący wpływ na kształt wirtualnych komórek przedmiotowych. Wydajne planowanie i sterowanie przepływu produkcji jest możliwe dzięki zautomatyzowaniu podsystemu przepływu materiałów. W celu planowania i sterowania przepływem produkcji w zautomatyzowanym podsystemie przepływu materiałów, zaproponowano dwuetapową procedurę stanowiącą połączenie programowania liniowego z teorią grafów.

W roku 1994 opracowano podstawy zastosowania technologii grupowej w harmonogramowaniu produkcji [151]. Opracowano nowe metody pozwalające projektować wirtualne komórki przedmiotowe w systemach zautomatyzowanych, zorganizowanych w komórki technologiczne [96][124][131]. Ponadto, w pracy [124] opracowano procedurę kształtowania wirtualnych komórek przedmiotowych dla systemów zautomatyzowanych, realizowaną w czterech krokach postępowania:

- 1) generowania wielowariantowych marszrut technologicznych,
- 2) opracowania założeń kształtowania wirtualnych komórek przedmiotowych, takich jak: liczby wirtualnych komórek, liczby przydzielanych typów maszyn do wirtualnych komórek, liczby wykonywanych operacji w wirtualnych komórkach itd.,

- 3) selekcji egzemplarzy maszyn danego typu do pracy w wirtualnych komórkach przedmiotowych,
- 4) oceny wskaźników obciążenia, dotrzymania terminów zakończenia zleceń, itp.

Dalsze prace związane z rozwojem metod pozwalających na planowanie przepływu produkcji w systemach zautomatyzowanych związane są z powstaniem koncepcji zwiniętych systemów wytwarzania AMS (ang. *agile manufacturing systems*) [49][62][123].

W roku 1998 po raz pierwszy zwrócono uwagę na możliwość użycia koncepcji kształtowania wirtualnych komórek przedmiotowych w systemach niezautomatyzowanych (w systemach, w których podstawowym czynnikiem produkcji jest praca ludzka) [137]. Zdefiniowano pojęcie wirtualnej grupy roboczej, którą tworzą pracownicy przydzieleni do pracy w wirtualnej komórce przedmiotowej. W pracy [64] ostatecznie określono warunki jakie muszą być spełnione w celu zapewnienia żywotności wirtualnej komórki przedmiotowej. Stwierdzono, że „*wirtualna komórka przedmiotowa istnieje wtedy i tylko wtedy, gdy ograniczony zbiór maszyn i pracowników jest zadedykowany do wykonania rodziny wyrobów oraz wewnątrz wirtualnej komórki przedmiotowej występuje skoordynowany przepływ materiału i informacji w czasie, niekoniecznie w ograniczonej przestrzeni*”.

Bazując na definicji żywotności wirtualnej komórki przedmiotowej, w pracy [159] dokonano porównania stopnia obciążenia oraz wydajności wirtualnych komórek z tradycyjnymi komórkami przedmiotowymi w wieloetapowym systemie wytwórczym produkującym wyroby o szeregowej strukturze procesu produkcyjnego. Dla celów porównawczych opracowano model analityczny, w którym zmiennymi decyzyjnymi są liczba etapów procesu produkcyjnego, liczba maszyn występujących w jednym etapie, wielkość partii produkcyjnej oraz wskaźnik czasu przebrojenia do czasu operacyjnego.

W pracy [133] zaprezentowano odmienne podejście do kształtowania wirtualnych komórek przedmiotowych. Zasugerowano, aby skupić się na kształtowaniu wirtualnych komórek w oparciu o ukierunkowany przepływ produkcji wyrobów zamiast dedykować maszyny i pracowników do całościowego produkowania rodziny wyrobów. Cechą szczególną tego podejścia jest identyfikacja wyrobów posiadających taką samą sekwencję operacji. Korzyścią jest skrócenie czasu przepływu produkcji wyrobów w systemie wytwórczym składającym się z kilku wydziałów produkcyjnych.

W pracy [123] zaproponowano metodę będącą rozbudowaną wersją analizy przepływu produkcji PFA (ang. *production flow analysis*), opisaną dokładnie w pracy [60]. Metoda umożliwia projektowanie wirtualnych komórek przedmiotowych, w których maszyny mogą być współdzielone pomiędzy różne rodziny wyrobów w jednym okresie planowania. Metodę wdrożono w dwóch przedsiębiorstwach produkcyjnych. Metoda ta została dalej rozwinięta w pracy [78]. Powstały dwa algorytmy projektowania wirtualnych komórek przedmiotowych. Pierwszy algorytm to tradycyjna metoda wydzielania komórek przedmiotowych, wykorzystująca wskaźnik podobieństwa. W drugim algorytmie badany jest poziom współdzielenia pogrupowanych typów maszyn pomiędzy rodziny wyrobów oraz ostatecznie kształtuje się wirtualne komórki.

W pracy [144] sformułowano podstawy projektowania wirtualnych komórek przedmiotowych z uwzględnieniem ograniczonej dostępności pracowników oraz ich ograniczonych

umiejętności. Opracowano metodę programowania wielokryterialnego, w której najpierw dokonuje się grupowania maszyn w wirtualne komórki a następnie przydziela się pracowników do pracy w utworzonych wirtualnych komórkach. Za cel metody optymalizacyjnej przyjęto maksymalizację poziomu obciążenia maszyn i pracowników oraz minimalizację stopnia współdzielenia maszyn.

W roku 2006 pojawił się artykuł przeglądowy, w którym przedstawiono dotychczasowe osiągnięcia w dziedzinie projektowania, analizy oraz badań empirycznych wirtualnych komórek przedmiotowych [110]. Przeprowadzono wyczerpujący przegląd literatury oraz zasugerowano kierunki dalszych prac związanych z projektowaniem, analizą oraz badaniami empirycznymi wirtualnych komórek przedmiotowych.

W dziedzinie projektowania wirtualnych komórek zauważono, że dalsze prace powinny skupić się na rozwijaniu zaawansowanych metod uwzględniających ograniczenia dostępności i umiejętności pracowników. Metoda powinna być ukierunkowana na tworzenie wirtualnych grup pracowników o określonej wydajności pracy. Następnie, projektowanie wirtualnych komórek wymaga uwzględnienia uwarunkowań podsystemu przepływu materiałów (transportu oraz składania materiałów). Ponadto w trakcie projektowania powinno uwzględnić się ograniczenia zasobowe. Przez ograniczenia zasobowe rozumie się ograniczoną liczbę narzędzi, oprzyrządowania, oraz możliwości systemu wytwórczego. Autorzy twierdzą, że tak złożone zagadnienia wymagają użycia zaawansowanych metod sztucznej inteligencji oraz algorytmów metaheurystycznych.

Wymienione uwagi i kierunki dalszych prac w dziedzinie projektowania wirtualnych komórek przedmiotowych przyczyniły się do opracowania nowej metody projektowania [126]. Do projektowania wykorzystano programowanie liniowe z wbudowanym algorytmem optymalizacji rojem cząstek. Skupiono się na projektowaniu wirtualnych komórek przedmiotowych w systemie wytwórczym produkującym wyroby, dla których marszruty technologiczne są nieukierunkowane. W systemie wytwórczym może znajdować się kilka maszyn tego samego typu technologicznego. Występują również pracownicy o różnych kwalifikacjach zawodowych. Układ maszyn w systemie jest rozproszony. W pracy dąży się do takiego projektowania wirtualnych komórek przedmiotowych, aby zminimalizować całkowite koszty wytwarzania, transportu międzystanowiskowego oraz utrzymania zapasów w horyzoncie planowania, przy założeniu losowo napływających zleceń produkcyjnych.

W pracy [74] zwrócono uwagę na konieczność harmonogramowania produkcji w wirtualnych komórkach przedmiotowych. Opracowano model programowania całkowitoliczbowego dla problemu harmonogramowania maksymalnie 6×6 (tzn. 6 zleceń produkcyjnych, realizacja każdego zlecenia jest związana z sekwencyjnym przejściem przez 6 maszyn) oraz algorytm genetyczny dla problemów o rozmiarach większych niż 6×6 . Za cel postawiono minimalizację terminu zakończenia wszystkich zleceń w poszczególnych wirtualnych komórkach przedmiotowych.

W dziedzinie analizy wykorzystania wirtualnych komórek przedmiotowych, autorzy pracy [110] zauważają, że dotychczas skupiono się głównie na porównywaniu stopnia wydajności wirtualnych komórek z komórkami technologicznymi lub tradycyjnymi komórkami przedmiotowymi. Do analizy wykorzystywano metody symulacyjne. Testowano stopień wy-

dajności porównywanych form organizacji produkcji dla dobranych wartości parametrów. W zależności od metody badawczej, parametrami były: czasy przebrojenia, czasy jednostkowe, wielkości partii produkcyjnych, odmienne reguły priorytetów lub zmienne poziomy obciążenia systemu zleceniami produkcyjnymi. Analizie podlegały: czas przepływu produkcji, poziom zapasu produkcji w toku, wskaźnik obciążenia, wskaźnik opóźnienia realizacji zleceń lub długość pokonywanej drogi transportowej. Autorzy zasugerowali, aby w dalszych pracach związanych z analizą wirtualnych komórek produkcyjnych wykorzystywać rzeczywiste dane pochodzące z przedsiębiorstw przemysłowych. Główny zarzut, jaki pojawia się przy ocenie prac związanych z analizą wirtualnych komórek dotyczy stosowania danych hipotetycznych.

Postawiony zarzut przyczynił się do powstania pracy, w której na podstawie dwóch przypadków przedsiębiorstw przemysłowych dokonano analizy porównawczej trzech konfiguracji wirtualnych komórek przedmiotowych [111].

Zostały również przeprowadzone badania symulacyjne mające na celu sprawdzenie korzyści stosowania wirtualnych komórek przedmiotowych w stosunku do tradycyjnych komórek przedmiotowych oraz tradycyjnych komórek technologicznych [75]. Przyjęto założenie, że symulacja zostanie przeprowadzona w rzeczywistych warunkach produkcji małoseryjnej. Badaniu podlegały wskaźniki, m. in. średni czas przepływu oraz średnie opóźnienie w stosunku do założonego terminu zakończenia zleceń. Do budowy komórek produkcyjnych oraz do tworzenia harmonogramów zadań wykorzystano algorytm mrówkowy. W wyniku przeprowadzonej symulacji otrzymano rezultaty wskazujące na wirtualną komórkę przedmiotową jako najodpowiedniejszą formę organizacji produkcji dla produkcji małoseryjnej.

W dziedzinie badań empirycznych zauważono mały postęp ze względu na niską liczbę opracowań zadedykowanych do wirtualnych komórek przedmiotowych. Dotychczasowe badania były prowadzone na potrzeby tradycyjnych komórek przedmiotowych. W literaturze pojawiają się jedynie wzmianki na temat badań empirycznych prowadzonych dla wirtualnych komórek przedmiotowych, które mają charakter opisowy. Z przeprowadzonych badań empirycznych wynika, że w większości przedsiębiorstw przemysłowych nie wydziela się tradycyjnych komórek przedmiotowych. W rzeczywistości występują formy mieszane lub hybrydowe, które mogą przypominać wirtualne komórki przedmiotowe pod względem logicznego powiązania maszyn [135].

2.7. Wnioski z analizy literatury

Na podstawie przeprowadzonej analizy literatury wyciągnięto następujące wnioski.

Po przeprowadzeniu analizy literatury stwierdzono, że obecnie do planowania przepływu produkcji w warunkach zmiennego zapotrzebowania najczęściej w praktyce stosowane są metody MRP. Są to najpopularniejsze metody planowania produkcji wyrobów charakteryzujących się wieloetapowym procesem produkcyjnym. Ich implementacja jest powszechnie dostępna w systemach klasy ERP. Stanowią one również fundament nowej klasy systemów planowania APS, które wzbogacają metodę MRP o dodatkowe tryby symulacyjne i optymalizacyjne. Chociaż systemy APS dzięki wbudowanym trybom optymalizacyjnym wydajniej przeprowadzają proces planowania dając zasadniczą korzyść finansową (minimalizacja kosztów realizacji zamówień), to liczba wdrożeń systemów APS jest znikoma. Przyczyny upatruje się

w wysokiej cenie systemu APS oraz niskim wskaźniku wdrożeń zakończonych powodzeniem. Występujące niepowodzenia przy wdrażaniu systemu APS wynikają ze specyficznych uwarunkowań obsługowych takich systemów. Systemy te posiadają wbudowane solvery optymalizacyjne. Ich prawidłowa obsługa wymaga dużych nakładów szkoleniowych oraz koordynacji planowania na wszystkich poziomach uszczegółowienia, począwszy od planowania długookresowego do planowania krótkookresowego. Szczególnie jest to istotne w przypadku małych i średnich przedsiębiorstw, które nie mogą pozwolić sobie na zatrudnienie rzeszy pracowników do obsługi wyłącznie systemu APS. Przedsiębiorstwa z sektora MŚP będą poszukiwać rozwiązań prostszych w obsłudze, ukierunkowanych bardziej na sposób planowania zgodny ze standardem panującym w przedsiębiorstwie, ale dający się integrować z systemami ogólnodostępnymi, jakimi są systemy ERP.

W odpowiedzi na żądania sektora MŚP powstały uproszczone systemy APS charakteryzujące się ograniczoną funkcjonalnością w stosunku do dużego systemu APS klasy SAP APO. Uproszczone systemy APS to głównie systemy harmonogramowania zadań. Jak przytoczono w rozdziale poświęconym systemom APS, są to systemy z założenia proste i przyjazne w obsłudze. Mają stanowić narzędzia komputerowego wspomaganie planisty, którego funkcją jest układanie harmonogramu zadań. Uproszczone systemy APS posiadają proste narzędzia do automatycznego szeregowania zadań oparte o wykorzystanie reguł priorytetów. Tak, więc systemy mają bardzo ograniczone możliwości wspomaganie decyzji planistów. Nie ma żadnej gwarancji, że przygotowany harmonogram zadań w uproszczonym systemie APS jest zoptymalizowany.

Systemy klasy ERP/APS pracujące w oparciu o funkcjonalność metody MRP dokonują planowania produkcji na trzech poziomach uszczegółowienia. Na poszczególnych poziomach planowania dokonuje się planowania nadrzędnego, planowania potrzeb materiałowych z bilansowaniem zapotrzebowania ze zdolnościami produkcyjnymi oraz szczegółowego harmonogramowania produkcji. Cechą wspólną planowania na każdym poziomie uszczegółowienia jest przyjęcie pewnej stałej długości okresów planowania. Długość okresu planowania jest kluczowym parametrem gdyż stanowi podstawę do obliczenia obciążenia komórek produkcyjnych zleceniami i w przypadku produkcji wieloetapowej pośrednio przyczynia się do określenia terminów realizacji zleceń produkcyjnych. Obecnie do szacowania terminów realizacji zleceń produkcyjnych wykorzystuje się metody bazujące na technikach eksploracji danych. W tym miejscu pojawiają się dwie wątpliwości. Po pierwsze, literatura podaje, że w metodach szacowania terminów realizacji zleceń wykorzystuje się dane historyczne. Oznacza to, że takie dane muszą być gromadzone oraz odpowiednio przetwarzane, aby można było przeprowadzić proces szacowania terminów realizacji zleceń. Do tego celu wymagane są drogie rozwiązania komercyjne, np. narzędzie MS Business Intelligence. Nowe podejście powinno być ukierunkowane na możliwość szacowania terminów realizacji zleceń w oparciu o wskaźniki, obliczane na podstawie danych gromadzonych w łatwy sposób wprost z systemu wytwórczego (np. wskaźnik efektywności stanowisk roboczych, wskaźnik występowania zleceń priorytetowych). Po drugie, metody szacowania terminów realizacji zleceń zostały opracowane dla produkcji seryjnej. Założono przy tym, że poziom obciążenia komórki produkcyjnej jednakowymi wyrobami jest stały i nie ulega zmianie w czasie. Dla produkcji małoseryjnej, przy

często rotującym asortymencie wyrobów wymaga się opracowania nowego podejścia, w którym zmienność w zakresie rotujących wyrobów, wielkości produkcji i napływających zleceń jest rozpatrywana.

Inną kwestią podlegającą ocenie jest dążenie do skrócenia długości trwania okresu planowania. Autor pracy sformułował problem poszukiwania sposobu skrócenia czasu realizacji zleceń w warunkach produkcji małoseryjnej [166][167]. Drogą do osiągnięcia celu jest utworzenie sposobu planowania przepływu produkcji, który na etapie harmonogramowania produkcji jest w stanie zapewnić skrócenie terminu zakończenia wszystkich zleceń. Ważne jest, aby nowa metoda, pozwalająca skrócić termin zakończenia wszystkich zleceń, była niezależna od przyjętej metody harmonogramowania zadań.

Po przeprowadzeniu analizy literatury stwierdzono, że istnieje taka możliwość (tj. skrócenia terminu realizacji wszystkich zleceń), która jest upatrywana w zastosowaniu technologii grupowej na etapie harmonogramowania produkcji. W rozdziale analizy literatury poświęconym zastosowaniu technologii grupowej do projektowania przepływu produkcji wyjaśniono problem grupowania technologicznego. Standardowa procedura grupowania technologicznego jest przeznaczona do projektowania tradycyjnych komórek przedmiotowych w celu uzyskania wydajnego przepływu w warunkach produkcji seryjnej. Wymaga modyfikacji, aby technologię grupową można było zastosować w warunkach produkcji małoseryjnej. Macierz zależności maszyn-wyrobów z elementami binarnymi nie zawiera informacji o konieczności wykonywania przezbrojeń maszyn. Ponadto przegrupowanie elementów macierzy nie informuje o sekwencji wykonywanych operacji w marszrucie technologicznej. W nowym podejściu powinno uwzględnić się kwestie konfiguracji maszyn do każdej operacji. Aby umożliwić uzyskanie korzyści krótszego przepływu produkcji, macierz zależności maszyn-wyrobów powinna być zbudowana w taki sposób, aby zachować sekwencję wykonywanych operacji.

Zastosowanie dynamicznych komórek przedmiotowych w warunkach produkcji małoseryjnej wydaje się być nieprzydatne. Biorąc pod uwagę fakt, iż z założenia w koncepcji dynamicznych komórek przedmiotowych dokonuje się okresowych przestawień maszyn, takie podejście jest nieekonomiczne w przypadku rozpatrywanego w niniejszej pracy systemu job-shop. Dużo lepszym rozwiązaniem wydaje się być zastosowanie koncepcji wirtualnych komórek przedmiotowych.

Przeprowadzona analiza literatury z zakresu kształtowania wirtualnych komórek przedmiotowych wykazała, że jest to najodpowiedniejsza forma organizacji w przypadku planowania przepływu produkcji w warunkach zmiennego zapotrzebowania z wyszczególnieniem produkcji małoseryjnej. Koncepcja kształtowania wirtualnych komórek przedmiotowych bazuje na zasadach wyżej wymienionej technologii grupowej. Dotychczas opracowane podejścia zostały przeanalizowane, wymieniono kierunki dalszych prac. Złożoność problemu projektowania wirtualnych komórek przedmiotowych powoduje, że w celu racjonalnego zaplanowania przepływu produkcji małoseryjnej wymaga się zastosowania technik sztucznej inteligencji lub algorytmów metaheurystycznych. Obecny stan zaawansowania metod projektowania wirtualnych komórek przedmiotowych jest zbieżny z przedstawionym spostrzeżeniem. Należy jednak zwrócić uwagę na dwie kwestie. Po pierwsze, w rozwijanych metodach wykorzystuje się hipotetyczne dane o charakterze deterministycznym. Spostrzeżenie nabiera

istotnego znaczenia przy uwzględnianiu czynnika ludzkiego, który decyduje o wydajności wirtualnej komórki przedmiotowej lub całego systemu wytwórczego. W nowym podejściu powinno pojawić się założenie, aby dane do projektowania wirtualnych komórek przedmiotowych miały charakter stochastyczny. Rozkład prawdopodobieństwa zmiennych losowych powinien być dopasowany do danych historycznych określonego przedsiębiorstwa produkcyjnego. Po drugie, metody projektowania ukierunkowane są na planowanie przepływu produkcji tylko poprzez kształtowanie wirtualnych komórek przedmiotowych. W rzeczywistości, przy często rotującym asortymencie wyrobów zaplanowanie racjonalnego przepływu produkcji jest utrudnione, mając do dyspozycji jedynie możliwość projektowania wirtualnych komórek. Nowe podejście powinno umożliwić planowanie przepływu produkcji w oparciu o różne formy organizacji produkcji, ale w szczególności przez kształtowanie wirtualnych komórek przedmiotowych.

3. CEL, TEZA I ZAKRES PRACY

Mając na uwadze wnioski z analizy literatury, dla sformułowanego problemu skrócenia czasu zakończenia zleceń produkcyjnych w warunkach produkcji małoseryjnej, ustalono cel prowadzonych badań.

Celem niniejszej pracy jest opracowanie algorytmu do racjonalnego planowania przepływu produkcji w warunkach zmiennego zapotrzebowania. Racjonalizacja planowania przepływu produkcji dotyczy możliwości skrócenia czasu realizacji zleceń produkcyjnych z założeniem, że raz potwierdzony termin realizacji zleceń produkcyjnych nie zostanie przekroczony nawet w przypadku wystąpienia nieoczekiwanych zdarzeń.

Za nieoczekiwane zdarzenia uważa się wszelkie awarie maszyn, absencje pracowników, oczekiwanie na materiał do produkcji oraz pojawienie się zleceń priorytetowych, które w pierwszej kolejności muszą być wykonane niezależnie od wcześniej utworzonego planu. W praktyce, nieoczekiwane zdarzenia powodują konieczność przeprojektowania harmonogramu co wiąże się z naruszeniem zaplanowanych terminów realizacji zleceń. Nowy algorytm ma przeciwdziałać zjawisku naruszania terminów realizacji zleceń produkcyjnych.

W tym celu należy opracować metodę ustalania terminów realizacji wszystkich zleceń. W metodzie tej dąży się do minimalizacji terminów zakończenia wszystkich zleceń. Następnie do zminimalizowanego terminu zakończenia wszystkich zleceń dodaje się pewien fundusz czasu. Minimalny termin zakończenia wszystkich zleceń oraz dodany fundusz czasu określają jednoznacznie termin realizacji wszystkich zleceń. Jest to informacja przesyłana do klientów systemu wytwórczego. Fundusz czasu ma być obliczony w celu wytłumienia wszelkich nieoczekiwanych zdarzeń. Bardzo ważne jest zdefiniowanie wskaźników pozwalających oszacować długość funduszu czasu stanowiącego narzut na termin zakończenia wszystkich zleceń produkcyjnych.

We wnioskach z analizy literatury stwierdzono, że nowa metoda planowania przepływu produkcji ma być uniezależniona od zastosowanej metody harmonogramowania zadań. Oznacza to, że posługując się pewną metodą harmonogramowania należy opracować taki sposób planowania przepływu produkcji, aby umożliwić osiągnięcie lepszego rozwiązania niż w sposób tradycyjny określony przez problem *general job-shop*. Do tego celu została wykorzystana koncepcja formowania wirtualnych komórek przedmiotowych w sposób niespotykany do tej pory w literaturze. Powstała metoda planowania przepływu produkcji, która jest ściśle związana z zagadnieniem harmonogramowania *general job-shop*. Uwzględniając powyższe stwierdzenie sformułowano tezę pracy:

Metoda planowania przepływu produkcji, bazująca na kształtowaniu wirtualnych komórek przedmiotowych tworzonych na poziomie planowania krótkookresowego, umożliwia skrócenie terminu realizacji zbioru zleceń produkcyjnych w porównaniu do tradycyjnego sposobu planowania i harmonogramowania *general job-shop*, niezależnie od dobranej metody szeregowania zadań.

Opracowana metoda planowania przepływu produkcji ma zastosowanie w warunkach produkcji małoseryjnej. Ten typ produkcji cechuje duża losowość zapotrzebowania. Cechą charakterystyczną metody jest postępowanie zgodnie z zasadami technologii grupowej.

W celu udowodnienia tezy pracy, posłużono się wskaźnikiem czasowym, którym jest termin zakończenia wszystkich zleceń produkcyjnych C_{max} [121]. Ważne jest, że wartość tego wskaźnika obliczana jest co pewien przedział czasu, w każdym oknie planistycznym dla zbioru losowo napływających zleceń. Zakłada się przy tym, że planowanie przepływu produkcji jest procesem powtarzanym co pewien odstęp czasu, a nie na zasadzie dynamicznego harmonogramowania zadań. Zlecenia produkcyjne są najpierw gromadzone, a następnie tworzony jest harmonogram dla wszystkich nagromadzonych zleceń. W przypadku dynamicznego harmonogramowania zadań, zlecenie produkcyjne jest rozplanowane w istniejącym harmonogramie w momencie jego pojawiania się w komórce produkcyjnej.

Opracowana metoda planowania przepływu produkcji została wykorzystana do budowy algorytmu noszącego nazwę ProduFlow PM. Model decyzyjny zaimplementowany w algorytmie ukierunkowany jest na kształtowanie wirtualnych komórek przedmiotowych w oknach planistycznych. Zadania przewidziane do pracy w wirtualnych komórkach przedmiotowych, wraz z pozostałymi zadaniami podlegają dalszemu harmonogramowaniu produkcji. O przydziale zadań do pracy w wirtualnych komórkach przedmiotowych decyduje wbudowany algorytm genetyczny, który optymalizuje harmonogram według kryterium minimalizacji terminu zakończenia zbioru zleceń.

Istotną cechą algorytmu ProduFlow PM jest jego możliwość decydowania o przyjęciu terminu realizacji zbioru zleceń. Model decyzyjny w tym przypadku szacuje długość funduszu czasu, bazując na wskaźnikach zarówno efektywności poszczególnych komórek produkcyjnych jak i liczby zleceń priorytetowych. Zgodnie z założeniem, znany jest wspólny czas rozpoczęcia realizacji grupy zleceń. Algorytm dąży do oszacowania terminów realizacji zbioru zleceń, czyli inaczej, algorytm dąży do ustalenia długości okna planistycznego. Do szacowania długości okna planistycznego zastosowano metaheurystykę przeszukiwania tabu.

Dla potwierdzenia tezy pracy zbudowano algorytm ProduFlow PM przyjmując następujące założenia:

1. Planowaniu podlegają jedynie systemy produkcyjne typu job-shop charakteryzujące się wieloasortymentową produkcją nierytmiczną. Zlecenia produkcyjne napływają nieregularnie (losowo). Liczba wyrobów przypadających na jedno zlecenie mieści się w zakresie produkcji typu małoseryjnego.
2. W systemie produkcyjnym realizuje się procesy wieloetapowe. W strukturze systemu produkcyjnego identyfikuje się elementy składowe tego systemu będące komórkami produkcyjnymi II stopnia. W komórce produkcyjnej II stopnia wykonuje się jeden etap procesu produkcyjnego wyrobu. Nowa metoda pozwala na planowanie przepływu produkcji w poszczególnych komórkach produkcyjnych II stopnia. Dlatego w pracy ograniczono się do weryfikacji metody w pojedynczej komórce produkcyjnej II stopnia.
3. Rozpatruje się systemy produkcyjne z przyjętą strategią MTO. Są to systemy, w których utrzymywane są zapasy surowców. Zapasy surowców pochodzą od do-

stawców systemu produkcyjnego, ich poziom może być regulowany np. za pomocą metody kontroli ciągłej. Wyroby systemu produkuje się wyłącznie na zamówienie klienta. Oznacza to, że w związku z występowaniem popytu zależnego obowiązującą strategią zarządzania zapasami jest *lot-for-lot*.

4. W systemie produkcyjnym istnieje określony park maszynowy. Układ maszyn w tym systemie jest ustalony. Maszyny są rozmieszczone w formie komórek technologicznych (są to komórki produkcyjne I stopnia), dopuszcza się rozmieszczenie rozproszone, holoniczne lub fraktalne. Mogą pojawić się nowe maszyny zwiększające zdolność produkcyjną danej komórki produkcyjnej II stopnia.
5. W metodzie nie rozpatruje się wpływu czynnika ludzkiego na planowanie przepływu produkcji. Ograniczono się jedynie do harmonogramowania produkcji na maszynach. Spadek efektywności stanowisk roboczych, spowodowany spadkiem wydajności pracy operatorów został uwzględniony pośrednio we wskaźniku efektywności stanowisk roboczych.
6. Nie dysponuje się danymi historycznymi o dotychczasowej realizacji zleceń produkcyjnych. W związku z tym, wartości wskaźników nieoczekiwanych zdarzeń są wartościami hipotetycznymi. W pracy nie dokonuje się obliczenia wartości tych wskaźników.
7. Planowanie przepływu produkcji na podstawie napływających zleceń produkcyjnych jest wykonywane w każdym oknie planistycznym. Częstość planowania zależy od przyjętej długości okna planistycznego w określonej komórce produkcyjnej II stopnia. Wyjątek stanowią zlecenia priorytetowe. Pojawienie się takiego zlecenia może spowodować konieczność przeplanowania istniejącego planu przepływu produkcji.
8. W pracy nie rozpatruje się możliwości zastosowania alternatywnych marszrut technologicznych. Zakłada się, że w fazie poprzedzającej działanie algorytmu układającego harmonogram produkcji zostały wybrane marszrut technologiczne dla każdego zlecenia produkcyjnego.

4. **PLANOWANIE PRZEPŁYWU PRODUKCJI W WARUNKACH ZMIENNEGO ZAPO- TRZEBOWANIA: ALGORYTM PRODUFLOW PM**

Opracowanie nowego algorytmu do planowania przepływu produkcji, który pozwoli skrócić czas realizacji zleceń produkcyjnych w warunkach produkcji małoseryjnej, wymaga przyjęcia obowiązujących definicji oraz stosowanych oznaczeń. W niniejszej pracy przyjęto (głównie korzystając ze źródeł [20][31]) następujące definicje:

Wyrób finalny – jest to produkt stanowiący końcowy wynik procesu produkcyjnego przedsiębiorstwa. Każdy wyrób posiadający unikalne cechy użytkowe, estetyczne, jakościowe itd., stanowi odrębną pozycję wśród produkowanego asortymentu. Dla przedsiębiorstwa produkcyjnego może on być:

- a) wyrobem złożonym - zwanym konwencjonalnie zespołem lub celowo utworzonym zbiorem zespołów i podzespołów, jest to złożenie składające się przynajmniej z dwóch elementów stanowiących wyroby proste, połączonych ze sobą w sposób rozłączny lub nierozłączny.
- b) wyrobem prostym (detalem lub częścią) – jest to jednolity element konstrukcyjny otrzymany w wyniku przetwarzania jednego gatunku materiału i nieposiadający żadnych powiązań z innymi elementami.

Z założenia w niniejszej pracy przyjmuje się, że wyrobem finalnym, będącym jednocześnie wyrobem prostym może być jedynie część oryginalna. Części oryginalne to części wytwarzane w zakresie działalności przedsiębiorstwa produkcyjnego. Do wykonania tych części niezbędne jest przygotowanie odpowiedniej dokumentacji konstrukcyjno-technologicznej. Części oryginalne, choć stanowią odrębne wyroby finalne, mogą być wykorzystane do produkcji wyrobów złożonych.

Na wyrób finalny będący jednocześnie wyrobem złożonym, oprócz części oryginalnych, mogą składać się również:

- części znormalizowane – części, których parametry konstrukcyjne określa odpowiednia norma europejska (EN), polska (PN), branżowa (BN) lub zakładowa,
- części zunifikowane – części produkowane przez kooperantów, w oparciu o ich własną dokumentację konstrukcyjno-technologiczną, zakupywane do projektowanego wyrobu w ramach odpowiednich umów o dostawę.

Powyższe definicje przyczyniły się do przyjęcia następujących oznaczeń:

$w = 1, \dots, W$ – numery porządkowe wyrobów finalnych (pozycji asortymentu) systemu produkcyjnego,

$i = 1, \dots, I$ – numery porządkowe części oryginalnych będących jednocześnie wyrobami prostymi,

$a = 1, \dots, A$ – numery porządkowe wyrobów złożonych.

Parametry W , I oraz A są zbiorami skończonymi, na których zachodzi zależność:

$$A \cup I = W \tag{1}$$

Zlecenia produkcyjne z są tworzone na poszczególne wyroby finalne w , co zapisuje się $z(w)$. Zleceniem produkcyjnym jest dokument lub grupa dokumentów upoważniające do wyprodukowania wyszczególnionego wyrobu finalnego w określonej ilości, z nadanym terminem realizacji. Zlecenie produkcyjne może być utworzone dla wyrobu finalnego będącego wyrobem złożonym a . Dla takiego wyrobu tworzy się zbiór podzleceń $z(a)$ zwanych potocznie zleceniami kompletacyjnymi. Zlecenia takie służą do uruchomienia montażu zarówno wyrobu finalnego jak i jego podzespołów i zespołów. Dodatkowo wyrób złożony może składać się z pewnej liczby części oryginalnych i . Dla każdej części oryginalnej i należy utworzyć podzlecenie $z(i)$ zwane potocznie zleceniem komponentowym.

Do zlecenia produkcyjnego $z(w)$ utworzonego dla wyrobu złożonego przyporządkowuje się zbiór podzleceń:

$$z(w) = \{z(a=1, \dots, n), z(i=1, \dots, n)\}, \quad a \in A, i \in I \quad (2)$$

Każdy wyrób finalny w jest produkowany w systemie wytwórczym, którego struktura jest hierarchicznie uporządkowana. Składowymi struktury systemu produkcyjnego są komórki produkcyjne o różnym stopniu uszczegółowienia. Wyróżnia się komórki produkcyjne 0 stopnia KP_x^0 . Są to stanowiska robocze. Stanowiska robocze KP_x^0 są grupowane w gniazda o specjalizacji technologicznej. Gniazda o specjalizacji technologicznej są komórkami produkcyjnymi I stopnia KP_x^1 . Komórki KP_x^1 są tworzone w obrębie wydziału produkcyjnego. Na wydziale produkcyjnym wykonuje się zbiór operacji składających się na etap wytwórczy e (np. wydział kucia, wydział obróbki skrawaniem, wydział obróbki cieplnej, itd.). Wydział produkcyjny jest komórką produkcyjną II stopnia KP_x^2 . Inne stopnie komórek produkcyjnych nie są rozpatrywane w niniejszej pracy. Obowiązują oznaczenia:

$KP_x^0 = KP_1^0, \dots, KP_x^0$ – numer porządkowy komórki produkcyjnej 0 stopnia w komórce produkcyjnej I stopnia KP_x^1 ,

$KP_x^1 = KP_1^1, \dots, KP_x^1$ – numer porządkowy komórki produkcyjnej I stopnia w komórce produkcyjnej II stopnia KP_x^2 ,

$KP_x^2 = KP_1^2, \dots, KP_x^2$ – numer porządkowy komórki produkcyjnej II stopnia w komórce produkcyjnej wyższego rzędu (w niniejszej pracy w systemie wytwórczym).

W pojedynczej komórce produkcyjnej II stopnia KP_x^2 może być wykonywane zlecenie produkcyjne $z(w)$ lub jego część. Istnieje możliwość wydzielenia montażu wyrobów złożonych a , jeżeli istnieje wyodrębniony wydział montażu będący komórką produkcyjną II stopnia. Wówczas zachodzi zależność:

$$z(a) = z(KP_{MNT}^2, h) \quad (3)$$

gdzie:

KP_{MNT}^2 – wydział montażu, h – numer okresu planowania opisany w dalszej części rozdziału.

Jeżeli wyroby złożone a są montowane w obrębie komórki produkcyjnej II stopnia, w której wytwarza się części oryginalne i , wówczas zachodzi zależność:

$$z(w) = z(KP_x^2, h) = \{z(KP_x^2, h)_{a=1, \dots, n}, z(KP_x^2, h)_{i=1, \dots, n}\} \quad (4)$$

gdzie:

$z(KP_x^2, h)_{a=1, \dots, n}$ – zlecenia kompletacyjne dla zlecenia $z(w)$ wykonywane w komórce produkcyjnej x II stopnia w oknie planistycznym h ,

$z(KP_x^2, h)_{i=1, \dots, n}$ – zlecenia komponentowe dla zlecenia $z(w)$ wykonywane w komórce produkcyjnej x II stopnia w oknie planistycznym h . Opis koncepcji planowania okienkowego oraz definicja okna planistycznego znajduje się w rozdziale 4.3.

Części oryginalne i mogą być wytwarzane w kilku etapach wytwórczych e , jeżeli istnieje pewna liczba wyodrębnionych komórek produkcyjnych II stopnia KP_1^2, \dots, KP_x^2 , przez które przebiega marszruta technologiczna części i :

$e = 1, \dots, E$ – numer porządkowy etapu dla części oryginalnej i .

Do każdego etapu wytwórczego e części oryginalnej i tworzy się odrębne podzlecenie $z(e)$ zwane potocznie zleceniem etapowym. Zlecenie etapowe musi być wykonane w komórce produkcyjnej II stopnia. Wówczas na jedno zlecenie komponentowe $z(i)$ składa się określoną liczbą zleceń etapowych:

$$z(i) = \{z(e=1), \dots, z(e=E)\}, \quad (5)$$

Przyjęto kolejne założenie w pracy, w którym każdy etap wytwórczy e części oryginalnej i jest wykonywany w odrębnym oknie planistycznym h :

$h = 1, \dots, H$ – numer porządkowy okna planistycznego w horyzoncie planowania f .

Jeżeli przyjąć, że realizacja zlecenia komponentowego $z(i)$ wymaga wykonania sekwencji zleceń etapowych $z(e=1), z(e=2), \dots, z(e=E)$, co z kolei wymaga przejścia przez kolejne komórki produkcyjne $KP_{x=1}^2, KP_{x=2}^2, \dots, KP_{x=n}^2$ w kolejnych oknach planistycznych $h, h+1, \dots, h+E-1$, to dla takiego zlecenia komponentowego $z(i)$ zachodzi zależność:

$$z(i) = \{z(KP_1^2, h), z(KP_2^2, h+1), \dots, z(KP_n^2, h+E-1)\}, \quad (6)$$

Z kolei w określonej komórce produkcyjnej II stopnia KP_x^2 , w jednym oknie planistycznym h , może być wykonywanych wiele zleceń etapowych. W celu ich odróżnienia, dla każdego zlecenia etapowego przyjęto oznaczenie $z_y(KP_x^2, h)$, gdzie:

$y=1, \dots, Y$ – numer porządkowy zlecenia etapowego wykonywanego w komórce produkcyjnej II stopnia w oknie planistycznym h .

Długość okna planistycznego w horyzoncie planowania dla określonej komórki produkcyjnej II stopnia $l_f(KP_x^2)$ jest zmienna. Rozpatruje się horyzont planowania f przyjmując, że długość horyzontu planowania $l_{horizon}$ jest zawsze stała. Jeżeli przyjąć oznaczenie:

$f = 1, \dots, F$ – numer porządkowy horyzontu planowania,

to długości okien planistycznych $l_f(KP_x^2), l_{f+1}(KP_x^2)$ w kolejnych horyzontach planowania $f, f+1$ mogą się różnić.

Każde zlecenie etapowe $z(e)$ może być wykonane na kilka sposobów. Do etapu wytwórczego e części oryginalnej i przyporządkowany jest zbiór alternatywnych marszrut technologicznych:

$v = 1, \dots, V$ – numer porządkowy alternatywnej marszrut technologicznej w etapie wytwórczym e części oryginalnej i .

Każda alternatywna marszruta technologiczna v stanowi zbiór operacji technologicznych j :

$j = 1, \dots, J$ – numer porządkowy operacji technologicznej alternatywnej marszrutę technologicznej v . Rosnąca numeracja informuje o sekwencji wykonywania operacji technologicznych.

Każda operacja technologiczna j jest wykonywana na maszynie określonego typu m w jej konfiguracji k (przez konfigurację maszyny rozumie się zbiór narzędzi i oprzyrządowania do wykonania operacji j):

$m = 1, \dots, M$ – numer porządkowy typu technologicznego maszyny,

$k = 1, \dots, K$ – numer porządkowy konfiguracji maszyny m .

Przyjmuje się, że operację technologiczną j opisuje parametr:

t_{ievjmk} – jest to czas jednostkowy.

Podjęciem decyzję wyboru wariantu marszrutę technologicznej v w celu realizacji zlecenia etapowego z_y dla części oryginalnej i , czas jednostkowy dla każdej operacji j dla tego zlecenia można opisać w uproszczeniu symbolem tz_{yjm} .

Każdej maszynie m przyporządkowano zbiór konfiguracji k . Przy zmianie konfiguracji z $k-1$ na k zachodzi konieczność wystąpienia przebrojenia na maszynie m . Przebrojenie jest opisane parametrem:

$S_{mk-1,k}$ – jest to czas przebrojenia maszyny m z konfiguracji $k-1$ w konfigurację k .

Dodatkowo zakłada się, że czas przebrojenia maszyny m z konfiguracji $k-1$ w konfigurację k wynosi tyle samo, co czas przebrojenia z konfiguracji k w konfigurację $k-1$, tzn.:

$$S_{mk-1,k} = S_{mk,k-1} \quad (7)$$

Pomiędzy kolejnymi operacjami technologicznymi $j-1$ oraz j może wystąpić konieczność składowania zapasu produkcji w toku pewnego zlecenia etapowego $z_y(KP_x^2, h)$ w magazynie buforowym. Składowanie zapasu produkcji w toku jest konieczne, jeżeli po zakończeniu operacji $j-1$ nie można rozpocząć wykonywania operacji j ze względu na zajętość maszyny, na której ma być wykonana operacja j . Czas składowania zapasu produkcji w toku zlecenia etapowego pomiędzy operacjami oznacza się symbolem $tz_{yj-1,j}$. Za czas składowania $tz_{y0,1}$ przyjmuje się czas od momentu rozpoczęcia okna planistycznego do momentu rozpoczęcia pierwszej operacji z marszrutę technologicznej zlecenia z_y .

W niniejszej pracy skupiono się na pokazaniu zasady działania algorytmu *ProduFlow PM* w pojedynczej komórce produkcyjnej II stopnia. Ze względu na fakt, iż praca jest ukierunkowana na tworzenie wirtualnych komórek przedmiotowych, największe efekty działania algorytmu występują w komórce produkcyjnej II stopnia charakteryzującej się produkcją typu job-shop.

Celem działania algorytmu *ProduFlow PM* jest otrzymanie wartości minimalnej terminu realizacji zbioru zleceń produkcyjnych w warunkach występowania nieoczekiwanych zdarzeń. Proces racjonalizacji przepływu produkcji, polegający na poszukiwaniu dopuszczalnego terminu realizacji zbioru zleceń jest niezależny od zastosowanej metody harmonogramowania produkcji. W związku z tym, osiągnięcie krótszego terminu realizacji zbioru zleceń jest moż-

liwe przez wykorzystanie koncepcji kształtowania wirtualnych komórek przedmiotowych na poziomie harmonogramowania produkcji.

Racjonalizacja przepływu produkcji jest przeprowadzana z założeniem występowania nieoczekiwanych zdarzeń w fazie realizacji przyjętego planu przepływu produkcji. Dlatego funkcjonalność algorytmu ProduFlow PM jest rozszerzona o zadanie minimalizacji kosztu realizacji zbioru zleceń, mając na uwadze możliwość ponoszenia kar z tytułu niedotrzymania przyjętego terminu realizacji zbioru zleceń. Przyjęto ostatecznie dwa czynniki stanowiące zmienne decyzyjne w algorytmie ProduFlow PM w celu racjonalizacji przepływu produkcji. Należą do nich:

- $Cz_y(KP_x^2, h)$ – termin zakończenia zlecenia produkcyjnego $z_y(KP_x^2, h)$,
- $d(KP_x^2, h)$ – planowany termin realizacji zbioru zleceń $z_y(KP_x^2, h)$, $y = 1, \dots, Y$.

Algorytm ProduFlowPM działa w celu realizacji następującego zadania:

$$\min K(KP_x^2, h) = \sum_{y=1}^Y Kz_y(KP_x^2, h) \quad (8)$$

gdzie:

$$Kz_y(KP_x^2, h) = f(Cz_y(KP_x^2, h), d(KP_x^2, h)) \quad (9)$$

$Kz_y(KP_x^2, h)$ jest kosztowym modelem matematycznym realizacji zlecenia produkcyjnego, w którym zmiennymi decyzyjnymi są $Cz_y(KP_x^2, h)$ oraz $d(KP_x^2, h)$. Szczegółowy opis formalny modelu matematycznego został przedstawiony w podrozdziale 4.3 niniejszej pracy.

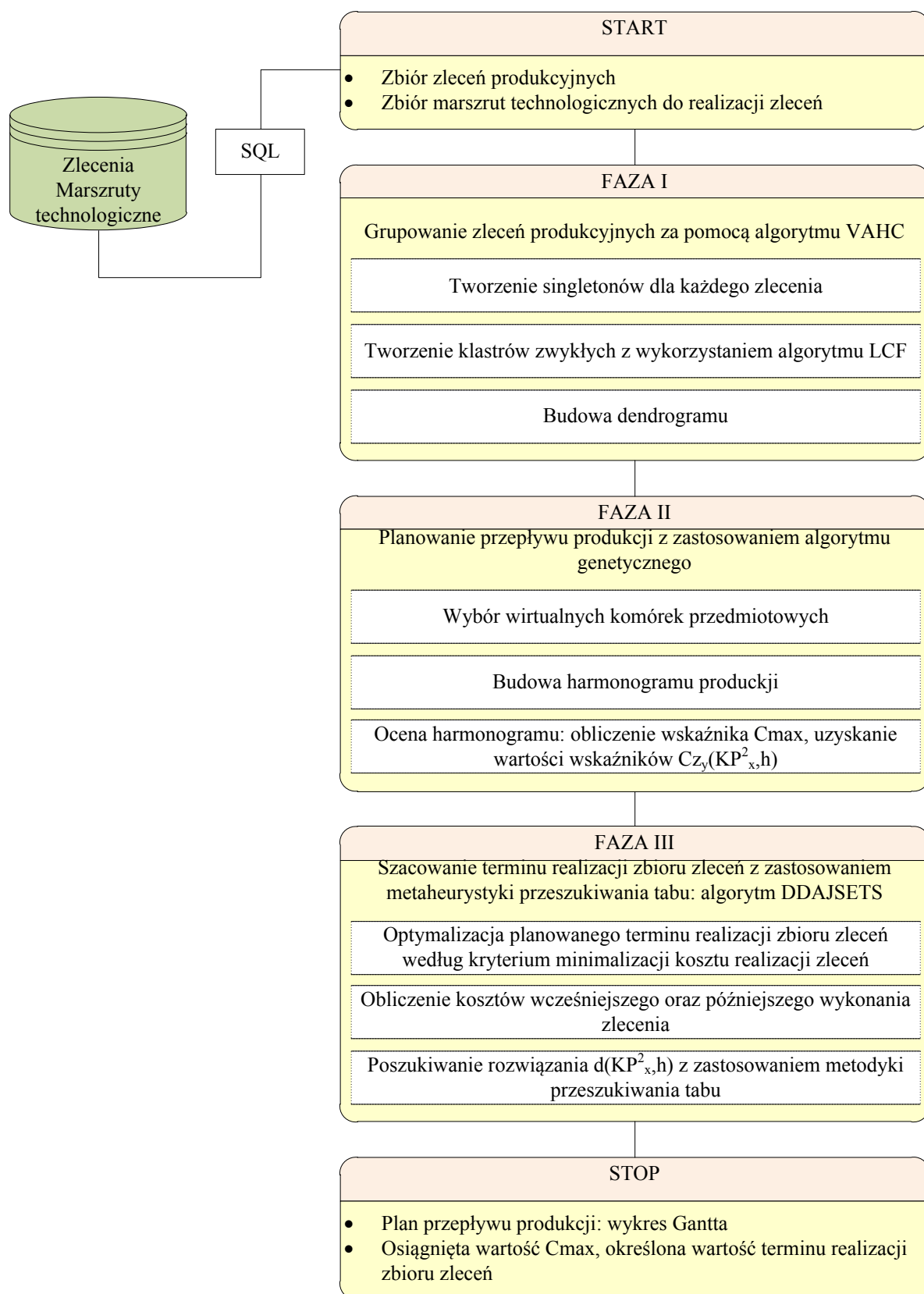
Bazując na powyższych definicjach oraz przyjętych oznaczeniach opracowano algorytm ProduFlow PM, w którym zaimplementowano zadanie (8). Schemat algorytmu przedstawiono na Rys. 12.

Algorytm ProduFlow PM składa się z trzech zasadniczych faz. Wartości poszczególnych zmiennych decyzyjnych są ustalane w fazach drugiej (dotyczy zmiennej $Cz_y(KP_x^2, h)$) oraz trzeciej (dotyczy zmiennej $d(KP_x^2, h)$).

Faza pierwsza służy do grupowania zleceń produkcyjnych w myśl zasad technologii grupowej. Sposób grupowania zleceń produkcyjnych oraz zasady tworzenia wirtualnych komórek przedmiotowych dla grupowanych zleceń zostały szczegółowo opisane w podrozdziale 4.1.

W fazie drugiej dokonuje się planowania przepływu produkcji z zastosowaniem wirtualnych komórek przedmiotowych. Proces planowania wykorzystujący algorytm genetyczny do planowania przepływu produkcji został wyjaśniony w podrozdziale 4.2.

Faza trzecia służy do szacowania długości okienek planistycznych, równoważnych z terminem realizacji zbioru zleceń produkcyjnych. Koncepcja planowania okienkowego oraz sposób szacowania terminu realizacji zbioru zleceń z zastosowaniem metaheurystyki przeszukiwania tabu zostały opisane w podrozdziale 4.3.



Rys. 12. Algorytm ProduFlow PM do planowania przepływu produkcji w warunkach produkcji małoseryjnej [opracowanie własne].

4.1. Warunki tworzenia wirtualnych komórek przedmiotowych w algorytmie ProduFlow PM

Szeroki asortyment produkcji, a równocześnie niewielka powtarzalność produkowanych wyrobów wymuszają konieczność stosowania systemów harmonogramowania *general job-shop* do planowania zleceń produkcyjnych. Pojedyncze zlecenie produkcyjne jest rozbijane na szereg zadań. Zadanie jest utożsamiane z operacją technologiczną. Budowa harmonogramu *general job-shop* charakteryzuje się tym, że każde zadanie, które ma być zrealizowane w systemie wytwórczym, jest planowane oddzielnie. Wymaga się jedynie spełnienia warunku uporządkowania technologicznego. Zadania pochodzące z jednego zlecenia produkcyjnego muszą być wykonywane w ściśle określonej kolejności. Zadania muszą być ustawione w sekwencji zgodnej z wybraną spośród alternatywnych marszrutą technologiczną wyrobu.

Harmonogramowanie typu *general job-shop* charakteryzuje się tym, że zadanie musi być wykonane bez przerw na wykonywanie innych zadań. Kolejne zadanie występujące w sekwencji zadań danego zlecenia produkcyjnego może się rozpocząć po zakończeniu zadania poprzedzającego z tej sekwencji zadań. W harmonogramie typu *general job-shop* występuje szeregowy przepływ produkcji. Jeżeli czasy trwania zadań są długie oraz liczba zadań do wykonania w zleceniu produkcyjnym jest duża to w efekcie uzyskuje się długi czas przepływu zlecenia $F_{z_y(KP_x^2, h)}$, niezależnie od stosowanej metody poszukiwania dokładnego rozwiązania problemu harmonogramowania. Czas przepływu zlecenia oblicza się ze wzoru:

$$F_{z_y(KP_x^2, h)} = C_{z_y(KP_x^2, h)} - r_{z_y(KP_x^2, h)} \quad (10)$$

gdzie:

$C_{z_y(KP_x^2, h)}$ - termin zakończenia zlecenia produkcyjnego $z_y(KP_x^2, h)$,

$r_{z_y(KP_x^2, h)}$ - termin rozpoczęcia zlecenia produkcyjnego $z_y(KP_x^2, h)$.

W pracy zaproponowano inne podejście do planowania szczegółowych zadań w komórce produkcyjnej II stopnia. Zamiast traktować zadania, jako wyodrębnione części zleceń produkcyjnych, poszukuje się możliwości grupowania określonych zleceń (lub fragmentów łańcuchów zadań wybranych zleceń) w myśl zasad technologii grupowej. Zgrupowane zlecenia są następnie wykonywane na wydzielonych (dedykowanych) stanowiskach roboczych. Dedykowane stanowiska robocze stanowią logicznie wydzielone komórki produkcyjne, tzw. wirtualne komórki przedmiotowe. W wirtualnych komórkach przedmiotowych stosuje się harmonogramowanie *permutation flow-shop*. W wirtualnych komórkach przedmiotowych występuje równoległy przepływ produkcji. Poniżej wyjaśniono koncepcję grupowania zleceń produkcyjnych.

4.1.1. Koncepcja grupowania zleceń dla produkcji małoseryjnej

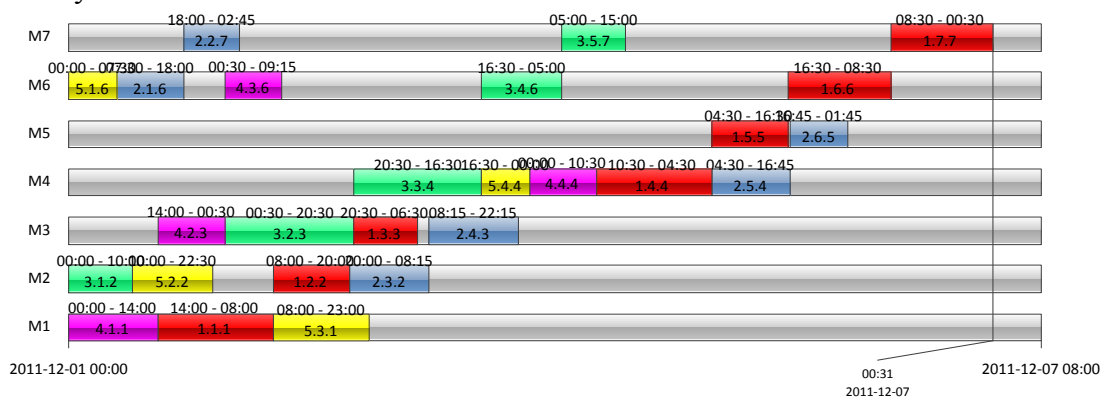
W celu wyjaśnienia koncepcji grupowania zleceń produkcyjnych w warunkach produkcji małoseryjnej zademonstrowano przykład. W przykładzie występuje pięć zleceń produkcyjnych. Dla każdego zlecenia produkcyjnego są dostępne następujące dane wejściowe, zestawione w Tab. 5.

Wykaz danych wejściowych dla zleceń produkcyjnych										
Id Zlecenia	$z_1(KP_x^2, h)$		$z_2(KP_x^2, h)$		$z_3(KP_x^2, h)$		$z_4(KP_x^2, h)$		$z_5(KP_x^2, h)$	
Wielkość zlecenia	120		105		150		105		150	
Partia transportowa	20		15		25		35		50	
Wykaz marszrut technologicznych i czasów przetwarzania dla każdego zlecenia										
Id Zlecenia	$z_1(KP_x^2, h)$		$z_2(KP_x^2, h)$		$z_3(KP_x^2, h)$		$z_4(KP_x^2, h)$		$z_5(KP_x^2, h)$	
Parametr	Maszyna	Czas [min]	Maszyna	Czas [min]	Maszyna	Czas [min]	Maszyna	Czas [min]	Maszyna	Czas [min]
Zadanie 1	M1	9	M6	6	M2	4	M1	8	M6	3
Zadanie 2	M2	6	M7	5	M3	8	M3	6	M2	5
Zadanie 3	M3	5	M2	7	M4	8	M6	5	M1	6
Zadanie 4	M4	9	M3	8	M6	5	M4	6	M4	3
Zadanie 5	M5	6	M4	7	M7	4				
Zadanie 6	M6	8	M5	6						
Zadanie 7	M7	8								

Tab. 5. Wykaz danych wejściowych dla zleceń produkcyjnych [opracowanie własne].

Dane wejściowe podzielone są na dwie części. W pierwszej części podaje się liczbę sztuk wyrobów (wielkość zlecenia) do wykonania dla każdego zlecenia produkcyjnego. Wielkość zlecenia jest równa partii produkcyjnej. Dodatkowo podaje się partię transportową przemieszczaną pomiędzy stanowiskami roboczymi w wirtualnych komórkach przedmiotowych. W drugiej części, dla każdego zlecenia określono marszrutę technologiczną. Każde zadanie w marszrucie technologicznej jest określone na maszynie, na której zostanie wykonane oraz czas jednostkowy. Dla uproszczenia rozważań w przykładzie pominięto przebrojenia maszyn.

Wstępnie opracowano harmonogram typu *general job-shop* z zaszeregowaniem zadań według reguły priorytetu SPT. Taki sposób zaszeregowania zadań występuje w systemach planowania APS. Szczegółowy opis sposobu szeregowania zadań według reguły SPT został zamieszczony w dodatku A, pkt 1. Na Rys. 13 przedstawiono harmonogram *general job-shop* utworzony według zaszeregowania zadań regułą priorytetu SPT. Harmonogram utworzono w postaci wykresu Gantta.



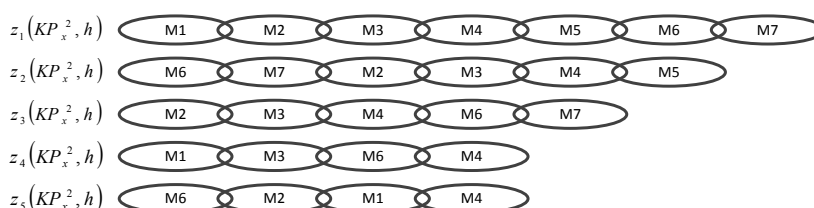
Rys. 13. Harmonogram job-shop z zaszeregowanymi zadaniami według reguły SPT [opracowanie własne].

Dla danego harmonogramu typu job-shop SPT otrzymano następujące czasy przebiegu, zestawione w Tab. 6.

Czasy przepływu zleceń produkcyjnych				
Nr zlecenia	Termin rozpoczęcia	Termin zakończenia	Czas przepływu [dz]	Czas przepływu [h]
Z1	01-12-11 14:00	07-12-11 0:30	5,438	130,5
Z2	01-12-11 7:30	06-12-11 1:45	4,760	114,25
Z3	01-12-11 0:00	04-12-11 5:00	3,208	77
Z4	01-12-11 0:00	04-12-11 10:30	3,438	82,5
Z5	01-12-11 0:00	04-12-11 0:00	3,000	72
			Średni F	95,25

Tab. 6. Czasy przepływu zleceń produkcyjnych dla harmonogramu job-shop SPT [opracowanie własne].

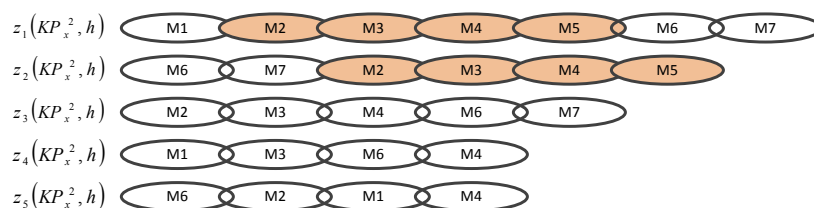
W celu grupowania zleceń produkcyjnych w warunkach produkcji małoseryjnej marszrut technologiczne traktowane są, jako łańcuchy zadań. Ogniwami łańcucha zadań są maszyny, na których realizowane są zadania. Rys. 14 przedstawia łańcuchy zadań marszrut technologicznych z Tab. 5.



Rys. 14. Łańcuchy zadań marszrut technologicznych [opracowanie własne].

Grupowanie zleceń produkcyjnych polega na poszukiwaniu najdłuższego wspólnego podłańcucha (*ang. longest common substring*) zadań spośród marszrut technologicznych. Nie ma ograniczenia liczby marszrut technologicznych, które równocześnie podlegają sprawdzeniu najdłuższego wspólnego podłańcucha.

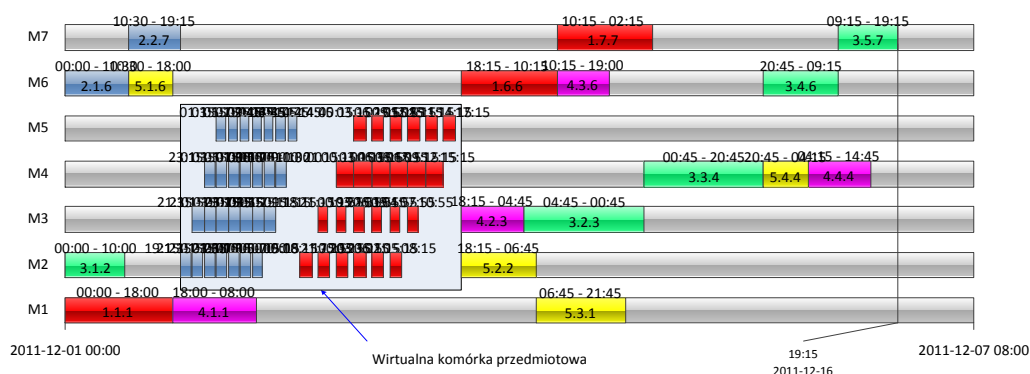
Porównując ze sobą pary łańcuchów zadań, najdłuższy wspólny podłańcuch posiadają zlecenia $z_1(KP_x^2, h)$ oraz $z_2(KP_x^2, h)$. Najdłuższy wspólny podłańcuch dla tej pary zleceń to: $M2 \rightarrow M3 \rightarrow M4 \rightarrow M5$. Długość tego podłańcucha wynosi 4. Rys. 15 przedstawia najdłuższy wspólny podłańcuch zleceń $z_1(KP_x^2, h)$ oraz $z_2(KP_x^2, h)$.



Rys. 15. Najdłuższy wspólny podłańcuch zleceń $z_1(KP_x^2, h)$ oraz $z_2(KP_x^2, h)$ [opracowanie własne].

Dla najdłuższego wspólnego podłańcucha zadań pochodzących od zleceń $z_1(KP_x^2, h)$ oraz $z_2(KP_x^2, h)$ można utworzyć wirtualną komórkę przedmiotową, w której występuje przepływ równoległy. Pozostałe zadania są traktowane rozłącznie i planowane tradycyjnie zachowując jedynie ograniczenia technologiczne. Rys. 16 przedstawia harmonogram job-shop SPT z wydzieloną wirtualną komórką przedmiotową, dla której zadedykowano maszyny M2, M3, M4 oraz M5. Należy zwrócić uwagę, że na czas istnienia wirtualnej komórki przedmiotowej maszyny są logicznie wydzielone z systemu wytwórczego. Nie ma możliwości wstawiania

innych zadań na maszyny zadedykowane do pracy dla wirtualnej komórki przedmiotowej niż te, które są zawarte w najdłuższym wspólnym podłańcuchu.



Rys. 16. Harmonogram job-shop SPT z wirtualną komórką przedmiotową dla najdłuższego wspólnego podłańcucha pochodzącego od zleceń $z_1(KP_x^2, h)$ oraz $z_2(KP_x^2, h)$ [opracowanie własne].

Tak uzyskany harmonogram typu job-shop SPT z wydzieloną wirtualną komórką przedmiotową cechuje inny rozkład zadań w czasie. Czasy przepływu poszczególnych zleceń dla harmonogramu job-shop SPT z Rys. 16 zestawiono w Tab. 7.

Czasy przepływu zleceń produkcyjnych				
Nr zlecenia	Termin rozpoczęcia	Termin zakończenia	Czas przepływu [dz]	Czas przepływu [h]
Z1	01-12-11 0:00	05-12-11 2:15	4,094	98,25
Z2	01-12-11 0:00	02-12-11 15:15	1,635	39,25
Z3	01-12-11 0:00	06-12-11 19:15	5,802	139,25
Z4	01-12-11 18:00	06-12-11 14:45	4,865	116,75
Z5	01-12-11 10:30	06-12-11 4:15	4,740	113,75
			Średni F	101,45

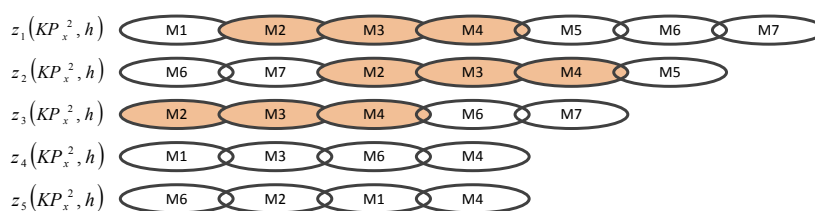
Tab. 7. Czasy przepływu zleceń produkcyjnych dla harmonogramu job-shop SPT z wydzieloną wirtualną komórką przedmiotową z Rys. 16 [opracowanie własne].

Cechą charakterystyczną harmonogramowania *job-shop* z wirtualnymi komórkami przedmiotowymi jest to, że zlecenia, które przepływają przez wirtualną komórkę przedmiotową są szeregowane, jako pierwsze. W efekcie, tworząc harmonogram z wykorzystaniem tej samej reguły priorytetu SPT uzyskuje się skrócenie czasu realizacji zleceń produkcyjnych, kosztem zwiększenia średniego czasu przepływu zleceń produkcyjnych. Przyczyną zwiększenia średniego czasu przepływu zleceń produkcyjnych jest oczekiwanie niektórych zadań na zwolnienie maszyn zadedykowanych do pracy w wirtualnej komórce przedmiotowej.

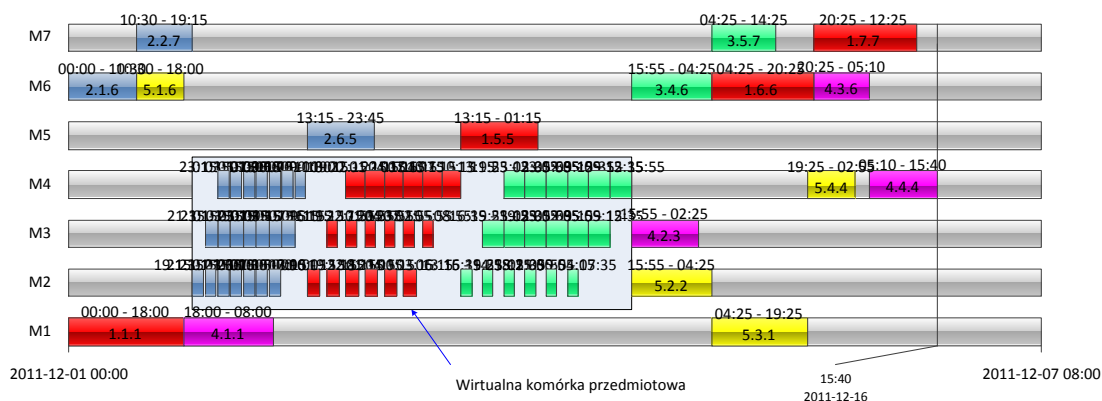
Poszukując możliwości dalszego grupowania zleceń można zmniejszyć długość wspólnego podłańcucha. Przykładowo, jest możliwe utworzenie grupy składającej się ze zleceń $z_1(KP_x^2, h)$, $z_2(KP_x^2, h)$ oraz $z_3(KP_x^2, h)$, której najdłuższy wspólny podłańcuch to: $M2 \rightarrow M3 \rightarrow M4$. Długość tego podłańcucha wynosi 3. Rys. 17 przedstawia najdłuższy wspólny podłańcuch zleceń $z_1(KP_x^2, h)$, $z_2(KP_x^2, h)$ oraz $z_3(KP_x^2, h)$.

Ponownie dla wybranych zleceń tworzy się wirtualną komórkę przedmiotową. Tym razem do wirtualnej komórki przedmiotowej zostają zadedykowane trzy maszyny M2, M3 oraz M4. Na kolejnych maszynach M2, M3 oraz M4 jest wykonywany najdłuższy wspólny podłańcuch zadań pochodzących od zleceń $z_1(KP_x^2, h)$, $z_2(KP_x^2, h)$ oraz $z_3(KP_x^2, h)$. Rys. 18

przedstawia harmonogram *job-shop* z wirtualną komórką przedmiotową z zadedykowanymi maszynami M2, M3 oraz M4.



Rys. 17. Najdłuższy wspólny podłańcuch zleceń $z_1(KP_x^2, h)$, $z_2(KP_x^2, h)$ i $z_3(KP_x^2, h)$ [opracowanie własne].



Rys. 18. Harmonogram *job-shop* SPT z wirtualną komórką przedmiotową dla najdłuższego wspólnego podłańcucha pochodzącego od zleceń $z_1(KP_x^2, h)$, $z_2(KP_x^2, h)$ oraz $z_3(KP_x^2, h)$ [opracowanie własne].

W wyniku utworzenia wirtualnej komórki przedmiotowej z zadedykowanymi maszynami M2, M3 oraz M4 powstał harmonogram, na podstawie zaszergowanych zadań regułą SPT. Szczegółowy opis metody szeregowania zadań z wirtualnymi komórkami przedmiotowymi regułą SPT zamieszczono w dodatku A, pkt 2. Czasy przepływu poszczególnych zleceń dla harmonogramu *job-shop* z Rys. 18 zestawiono w Tab. 8.

Wykorzystanie wirtualnej komórki przedmiotowej z zadedykowanymi maszynami M2, M3 oraz M4 pozwoliło jeszcze bardziej skrócić czas realizacji zleceń produkcyjnych. Średni czas przepływu zleceń również uległ skróceniu.

Czasy przepływu zleceń produkcyjnych				
Nr zlecenia	Termin rozpoczęcia	Termin zakończenia	Czas przepływu [dz]	Czas przepływu [h]
Z1	01-12-11 0:00	06-12-11 12:25	5,517	132,42
Z2	01-12-11 0:00	02-12-11 23:45	1,990	47,75
Z3	03-12-11 13:15	05-12-11 14:25	2,049	49,17
Z4	01-12-11 18:00	06-12-11 15:40	4,903	117,67
Z5	01-12-11 10:30	06-12-11 2:55	4,684	112,42
			Średni F	91,88

Tab. 8. Czasy przepływu zleceń produkcyjnych dla harmonogramu *job-shop* SPT z wydzieloną wirtualną komórką przedmiotową z Rys. 18 [opracowanie własne].

Przykład ukazuje możliwości skrócenia czasu realizacji zleceń produkcyjnych dzięki grupowaniu niektórych zleceń. Zgrupowane zlecenia, dla których określono najdłuższy wspólny podłańcuch zadań mogą być wytwarzane w wirtualnej komórce przedmiotowej. W wirtualnej komórce przedmiotowej są wykonywane zadania składające się na najdłuższy wspólny podłańcuch określonych zleceń. Daje to możliwość skrócenia czasu realizacji zbioru

zleceń produkcyjnych. Natomiast problemem staje się czas przepływu zleceń produkcyjnych. Wykonywanie zadań w wirtualnych komórkach przedmiotowych, choć wpływa na skrócenie czasu realizacji zleceń produkcyjnych, może doprowadzić do znacznego wydłużenia czasu przepływu tych zleceń przez system wytwórczy. Jest to zjawisko niepożądane. Dlatego średni czas przepływu zleceń produkcyjnych powinien stanowić kryterium optymalizacyjne (dążenie do osiągnięcia jego minimum) przy głównym ograniczeniu, którym jest ustalony wspólny czas zakończenia wszystkich zleceń produkcyjnych (*ang. due date*). Ustalenie wspólnego czasu zakończenia wszystkich zleceń produkcyjnych jest możliwe, jeżeli zastosuje się planowanie okienkowe. Planowanie okienkowe jest opisane w rozdziale 4.3.

4.1.2. Zastosowanie hierarchicznego aglomeracyjnego algorytmu klastrowania do grupowania zleceń

Podstawowym zadaniem algorytmu ProduFlow PM jest tworzenie przepływu produkcji z zastosowaniem wirtualnych komórek przedmiotowych. Wirtualna komórka przedmiotowa może być utworzona wtedy i tylko wtedy, gdy znane są grupy zleceń z określonym najdłuższym wspólnym podłańcuchem zadań. W rozdziale opisano sposób grupowania zleceń oraz sposób wyznaczania najdłuższego wspólnego podłańcucha zadań. Opracowano algorytm grupowania zleceń o nazwie VAHC (*ang. virtual agglomerative clustering algorithm*). Algorytm posiada wbudowany mechanizm hierarchicznego aglomeracyjnego klastrowania obiektów. W wyniku działania algorytmu powstają grupy zleceń na różnych poziomach. O poziomie, na którym zapisano grupę zleceń decyduje długość wspólnego podłańcucha zadań grupy. Grupy zleceń produkcyjnych prezentuje się na specjalnym dendrogramie. Dendrogram ten służy do wskazania zleceń, dla których mogą być tworzone wirtualne komórki przedmiotowe.

4.1.2.1. Budowa i działanie algorytmu VAHC do grupowania zleceń w warunkach produkcji małoseryjnej

Grupowanie zleceń produkcyjnych jest procesem czteroetapowym. Z tyłu też części zbudowany jest algorytm VAHC. Rys. 19 przedstawia schemat grupowania zleceń produkcyjnych za pomocą algorytmu VAHC.

W części pierwszej algorytmu VAHC tworzone są klastry podstawowe $S_y(KP_x^2, h)$, tzw. singletony. Singleton różni się od zwykłego klastra tym, że nie grupuje on w sobie kilku obiektów a zostaje w nim umieszczony jeden wybrany obiekt. W przypadku grupowania zleceń dla produkcji małoseryjnej, w singletonie umieszcza się pojedyncze zlecenie produkcyjne $z_y(KP_x^2, h)$. Singletony charakteryzują się tym, że posiadają łańcuch zadań równy liczbie zadań zlecenia produkcyjnego umieszczonego w singletonie. Singletony tworzy się wyłącznie dla poszczególnych zleceń produkcyjnych, stąd ten sam parametr wyliczeniowy $y=1, \dots, Y$ zarówno dla singletonów jak i zleceń produkcyjnych.

Spośród utworzonych singletonów zostaje wyszukany ten o najdłuższym łańcuchu zadań $LS_{\max}(KP_x^2, h)$:

$$LS_{\max}(KP_x^2, h) = \max \{LS_y(KP_x^2, h)\}, \quad y = 1, \dots, Y \quad (11)$$

gdzie: $LS_y(KP_x^2, h)$, $y = 1, \dots, Y$, - długości łańcuchów zadań poszczególnych singletonów.

Liczba zadań singletona o najdłuższym łańcuchu zadań stanowi górną granicę klastrowania. Od tej liczby rozpoczyna się proces grupowania w pętli iteracyjnej. Pętla iteracyjna klastrowania jest sterowana zmienną dekrementowaną l_{dend} od górnej granicy klastrowania do wartości 2 (bo minimalna liczba zadań podlegających realizacji w wirtualnych komórkach przedmiotowych wynosi 2):

$$l_{dend} = LS_{\max}(KP_x^2, h) \quad (12)$$

Wewnątrz pętli iteracyjnej znajdują się trzy pozostałe części algorytmu VAHC, w których tworzone są zwykłe klastry $C_z(KP_x^2, h)$. Klastry zwykłe to klastry utworzone na bazie co najmniej dwóch singletonów lub innych klastrów posiadających dłuższe łańcuchy zadań niż aktualna wartość zmiennej sterującej l_{dend} .

W drugiej części algorytmu VAHC dokonuje się selekcji klastrów lub singletonów, których długość łańcucha zadań jest większa lub równa wartości zmiennej sterującej l_{dend} w pętli iteracyjnej. Wyselekcjonowane singletony muszą spełniać następujący warunek:

$$LS_y(KP_x^2, h) \geq l_{dend} \quad (13)$$

Natomiast wyselekcjonowane klastry zwykłe muszą spełniać następujący warunek:

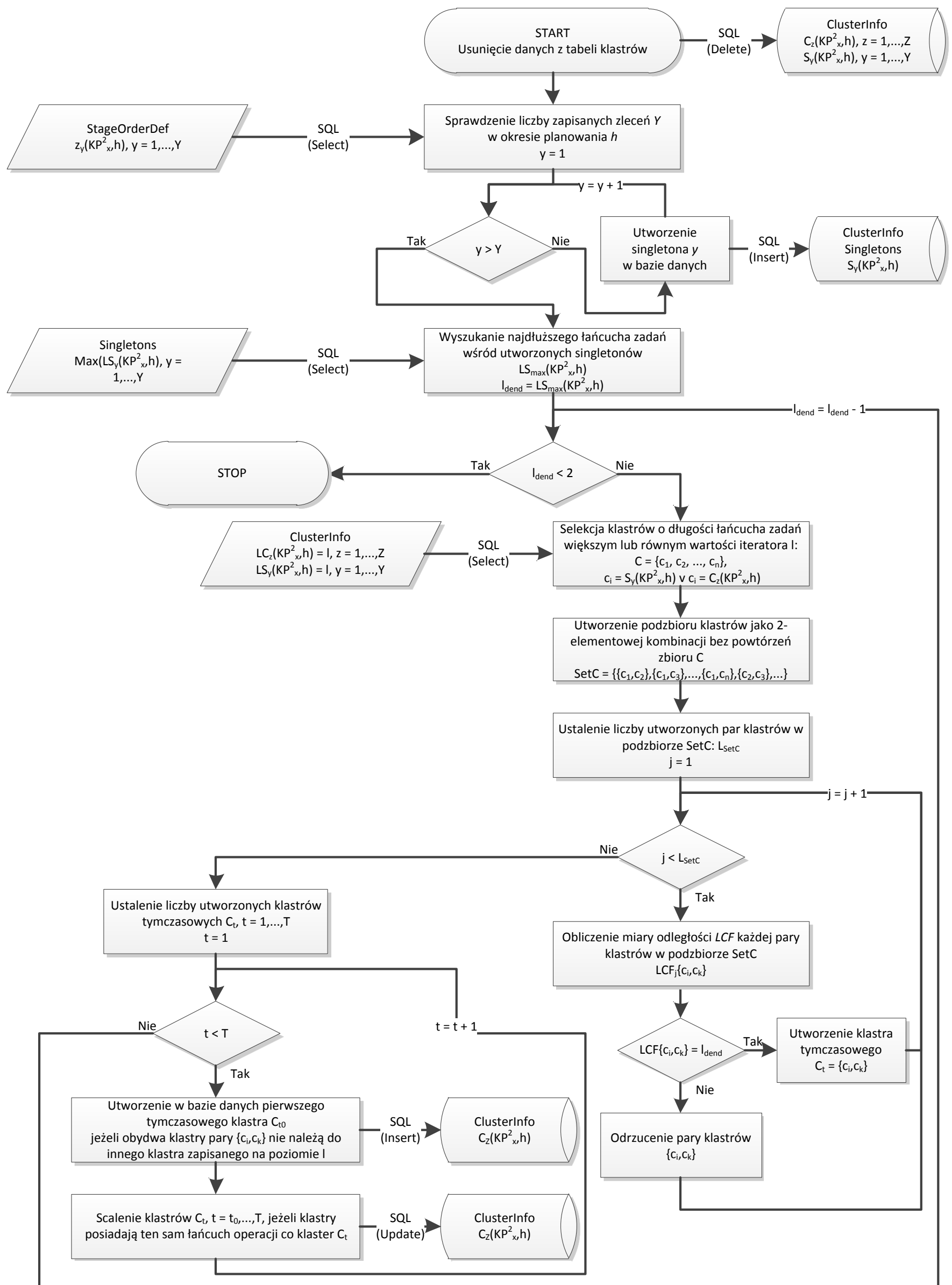
$$LC_z(KP_x^2, h) > l_{dend} \quad (14)$$

gdzie: $LC_z(KP_x^2, h)$, $z = 1, \dots, Z$, - długości łańcuchów zadań poszczególnych klastrów zwykłych.

Wybierane są jedynie te klastry i singletony, które nie posiadają klastra nadrzędnego (tj. klastra utworzonego na bazie klastrów wcześniej zdefiniowanych w hierarchii, klastru nadrzędnego posiada krótszy podłańcuch zadań od łańcuchów klastrów, z których został utworzony). Wyselekcjonowane klastry tworzą zbiór klastrów $C = \{c_1, c_2, \dots, c_i, \dots, c_k, \dots, c_n\}$. Ze zbioru C tworzony jest podzbiór $SetC = \{(c_1, c_2), \dots, (c_1, c_i), \dots, (c_1, c_n), \dots, (c_i, c_k), \dots, (c_k, c_n)\}$. Podzbiór $SetC$ jest dwuelementową kombinacją bez powtórzeń zbioru C . Podzbiór $SetC$ gromadzi pary wyselekcjonowanych klastrów. Liczbę par wyselekcjonowanych klastrów oblicza się ze wzoru:

$$L_{SetC} = \binom{c_n}{2} = \frac{c_n!}{2 \cdot (c_n - 2)!} \quad (15)$$

gdzie: c_n - liczba klastrów znajdujących się w zbiorze C .



Rys. 19. Schemat grupowania zleceń produkcyjnych za pomocą algorytmu VAHC [opracowanie własne]

Każda para wyselekcjonowanych klastrów ze zbioru $SetC$ podlega obliczeniu miary odległości między nimi (część trzecia algorytmu). Miarą odległości pomiędzy parą klastrów jest wartość najdłuższego wspólnego podłańcucha $LCF(c_i, c_k)$. Dla celów grupowania zleceń w planowaniu przepływu produkcji zmodyfikowano ogólnie dostępny algorytm wyszukiujący najdłuższy wspólny podłańcuch dwóch obiektów. Zmodyfikowany algorytm nazwano LCF (*ang. longest common flow*). Algorytm LCF służy do wyznaczenia najdłuższego wspólnego przepływu dwóch wyselekcjonowanych klastrów. Rys. 20 przedstawia schemat algorytmu LCF.

Definicja najdłuższego wspólnego przepływu dwóch klastrów brzmi następująco:

Definicja 1:

Jeżeli ciąg zadań pierwszego klastra przepływa przez takie same maszyny jak ciąg zadań drugiego klastra bez względu na to, które zadanie jest pierwszym zadaniem ciągu w stosunku do całego łańcucha zadań oraz jeżeli liczba zadań w danym ciągu zadań jest największa spośród innych ciągów zadań jednakowych dla obydwu klastrów, wówczas taki ciąg zadań uważa się za najdłuższy wspólny przepływ dla obydwu klastrów.

Zgodnie z definicją 1, do wyznaczenia najdłuższego wspólnego przepływu wymagana jest znajomość łańcuchów zadań dwóch porównywanych klastrów oraz typ technologiczny maszyny dla każdego zadania z obydwu łańcuchów zadań. Wyznaczenie najdłuższego wspólnego przepływu dwóch klastrów polega na porównaniu typów technologicznych maszyn, na których wykonywane są kolejne zadania obydwu klastrów.

Porównywanie dwóch ciągów zadań rozpoczyna się w momencie, gdy zostaną zidentyfikowane zadania, które są wykonywane na tym samym typie technologicznej maszyny. Wówczas sprawdzane są kolejne zadania z obydwu ciągów zadań do momentu zidentyfikowania zadań wykonywanych na różnych typach technologicznych maszyn. Liczba kolejnych jednakowych zadań w obydwu ciągach staje się najdłuższym wspólnym przepływem $LCF(c_i, c_k)$ pod warunkiem, że jest to najdłuższy napotkany ciąg zadań jednakowy dla obydwu klastrów.

Obliczona miara odległości $LCF(c_i, c_k)$ decyduje, czy dana para klastrów zostaje zachowana na liście kandydatów do utworzenia klastrów. Aby dana para klastrów nie została odrzucona musi być spełniony warunek:

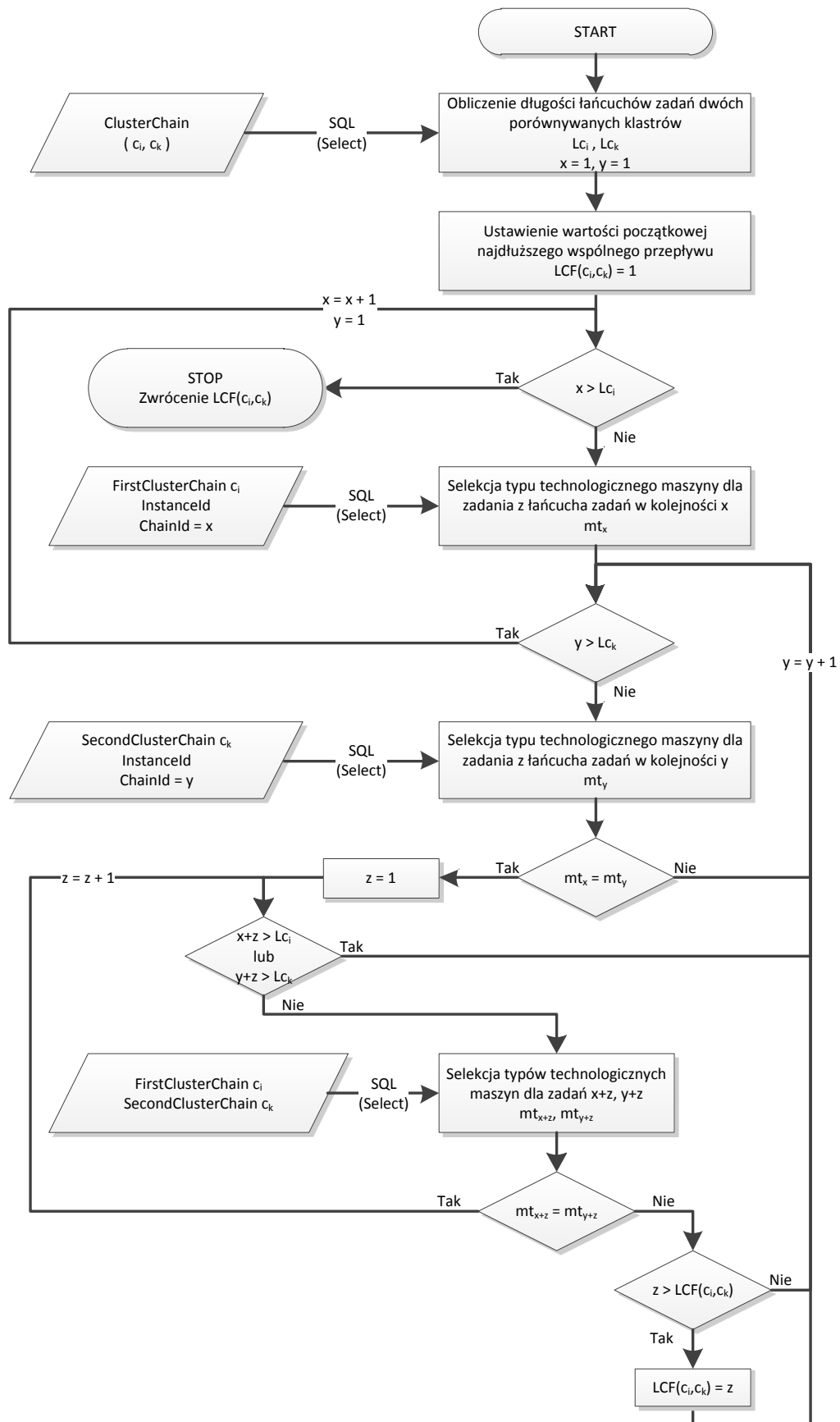
$$LCF(c_i, c_k) = l_{dend} \quad (16)$$

gdzie: l_{dend} – zmienna sterująca pętli iteracyjnej.

Oprócz miary odległości, dana para klastrów posiada dodatkowy parametr wskazujący początek najdłuższego wspólnego przepływu $LCB(c_i, c_k)$. Początek jest określany na pierwszym klastrze z pary klastrów. Jest to liczba wskazująca numer kolejnego zadania w ciągu zadań pierwszego klastra, od której rozpoczyna się najdłuższy wspólny przepływ $LCF(c_i, c_k)$.

Jeżeli powyższy warunek jest spełniony, wówczas na bazie danej pary klastrów zostaje utworzony klaster tymczasowy $C_i\{c_i, c_k\}$. Najdłuższy wspólny przepływ $LCF(c_i, c_k)$ staje się

łańcuchem zadań dla klastra tymczasowego $C_t \{c_i, c_k\}$. Zbiór klastrów tymczasowych $C_t, t = 1, \dots, T$ poddawany jest sprawdzeniu w ostatniej, czwartej części algorytmu.



Rys. 20. Schemat wyszukiwania najdłuższego wspólnego przepływu – algorytm LCF [opracowanie własne].

Czwarta część algorytmu służy do tworzenia klastrów na stałe spośród klastrów tymczasowych. Klasytry mogą być tworzone w różny sposób w zależności od statusów klastrów tymczasowych. Dopuszczalne są cztery statusy klastrów tymczasowych: ‘N’, ‘C’, ‘V’, ‘R’. Gdy dany klaster tymczasowy C_t posiada status ‘N’, wówczas jest to nowy nie sprawdzony jeszcze klaster. Jeżeli jest to pierwszy klaster na liście klastrów tymczasowych, to klaster ten zostanie przekształcony w klaster stały $C_z \{KP_x^2, h\}$. Przekształcenie takie powoduje zmianę statusu klastra tymczasowego na ‘C’. Jeżeli natomiast dany klaster tymczasowy C_t posiada status ‘N’, ale jest zapisany jako kolejny klaster na liście klastrów tymczasowych, wówczas zostaje uruchomiony ponownie algorytm LCF porównujący łańcuch zadań danego klastra tymczasowego z łańcuchami zadań klastrów przekształconych już w klasytry stałe. Jeżeli w wyniku działania algorytmu LCF okaże się, że warunek (12) jest spełniony, tzn. że dany klaster tymczasowy posiada łańcuch zadań identyczny z łańcuchem zadań pewnego klastra stałego. Klaster tymczasowy zostaje wchłonięty przez klaster stały i przyjmuje status ‘V’. Ten sposób postępowania pozwala tworzyć klasytry składające się z dwóch lub więcej składników, tj. klastrów podrzędnych, z których utworzony jest dany klaster stały. Ostatni status ‘R’ dotyczy przypadku, gdy w wyniku porównywania dwóch klastrów (tymczasowego i stałego) okaże się, że co najmniej jeden składnik klastra tymczasowego jest składnikiem klastra stałego. Wówczas taki klaster tymczasowy jest odrzucany, gdyż na bazie klastra podrzędnego można utworzyć jeden i tylko jeden klaster nadrzędny. Klaster tymczasowy przyjmuje status ‘R’.

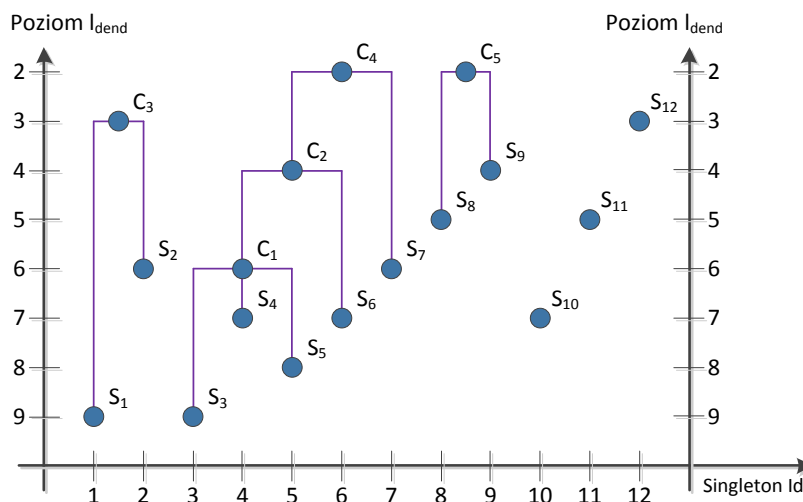
Algorytm VAHC kończy działanie w momencie osiągnięcia wartości $l_{dend} = 2$. W wyniku działania algorytmu otrzymuje się klasytry na różnych poziomach w zależności od obliczonych długości łańcuchów zadań klastrów. Wynik działania algorytmu VAHC jest przedstawiany graficznie w formie specjalnie sformatowanego dendrogramu. Wygląd oraz opis dendrogramu zamieszczono w kolejnym punkcie pracy.

4.1.2.2. Graficzna prezentacja wyników działania algorytmu VAHC - dendrogram

Formą prezentacji wyników działania algorytmu VAHC jest generowany programowo dendrogram. Pełni on dwojaką funkcję. Po pierwsze pozwala śledzić rezultaty klastrowania algorytmem VAHC, ułatwiając w ten sposób odczytanie wyników klastrowania. Po drugie wskazuje potencjalne miejsca i zakres wykorzystania wirtualnych komórek przedmiotowych dla dynamicznie napływających zleceń produkcyjnych. Rys. 21 przedstawia przykładowy dendrogram.

Osie dendrogramu są opisane następująco. Na osi odciętych znajdują się identyfikatory wszystkich singletonów utworzonych w części pierwszej algorytmu VAHC. Na osi rzędnych umieszczane są poziomy dendrogramu. Poziomy dendrogramu, na których tworzone kolejne klasytry są tożsame z wartością zmiennej sterującej pętli iteracyjnej l_{dend} . Ponadto w dendrogramie umieszcza się identyfikatory klastrów. Klasytry są podzielone na dwie grupy. Do pierwszej grupy zalicza się wszystkie singletony. Singletony oznaczono symbolami S_y (na Rys. 21: $y=1, \dots, 12$). Do drugiej grupy zalicza się wszystkie klasytry „właściwe” (tj. klasytry utworzone z singletonów lub innych klastrów podrzędnych). Klasytry takie oznaczono symbolami C_z (na Rys. 21: $z=1, \dots, 5$).

Cechą charakterystyczną dendrogramu prezentującego wyniki działania algorytmu VAHC jest to, że bezpośrednio umożliwia odczytanie liczby zadań oraz liczby zleceń, które mają być wykonywane w wirtualnej komórce przedmiotowej. Np. klastera C_2 powstał z połączenia klastra C_1 oraz singletona S_6 , z kolei klastera C_1 powstał z połączenia singletonów S_3 , S_4 oraz S_5 . Oznacza to, że może powstać wirtualna komórka przedmiotowa, w której będą wykonywane zlecenia związane z singletonami S_3 , S_4 , S_5 oraz S_6 . Z osi rzędnych odczytuje się ile zadań będzie wykonywanych w wirtualnej komórce przedmiotowej. Klastera C_2 znajduje się na czwartym poziomie, czyli cztery zadania z najdłuższego wspólnego podłańcucha zadań będą realizowane w wirtualnej komórce przedmiotowej.



Rys. 21. Przykładowy dendrogram reprezentujący wyniki działania algorytmu VAHC [opracowanie własne].

W kolejnym podrozdziale zaprezentowano przykład działania algorytmu dla pięciu zleceń zamieszczonych w Tab. 5. Wyniki zaprezentowano w formie graficznej na dendrogramie opisanym powyżej.

4.1.2.3. Algorytm VAHC – przykład obliczeniowy

Mając dostępne informacje (Tab. 5) zarówno o łańcuchach zadań oraz typach technologicznych maszyn, na których te zadania występują, w pierwszej części algorytmu tworzy się singletony. Liczba singletonów jest równa liczbie zleceń produkcyjnych i wynosi 5. Singletonom przyporządkowuje się identyfikatory: $S_1(KP_x^2, h) = z_1(KP_x^2, h)$, $S_2(KP_x^2, h) = z_2(KP_x^2, h)$, $S_3(KP_x^2, h) = z_3(KP_x^2, h)$, $S_4(KP_x^2, h) = z_4(KP_x^2, h)$, $S_5(KP_x^2, h) = z_5(KP_x^2, h)$. W uproszczeniu identyfikatory singletonów przyjmują symbole S_1, S_2, S_3, S_4, S_5 . Spośród singletonów wyszukuje się tego, który posiada najdłuższy łańcuch zadań. Na Rys. 14 widać, że najdłuższy łańcuch zadań posiada singleton S_1 . Długość łańcucha zadań singletona S_1 wynosi: $LS_1(KP_x^2, h) = LS_{\max}(KP_x^2, h) = 7$. Zmienna sterująca pętli iteracyjnej l_{dend} również przyjmuje wartość 7. Algorytm kończy pracę w pierwszej części i przechodzi do pętli iteracyjnej sterowanej dekrementowaną zmienną l_{dend} .

W części drugiej algorytmu, wyselekcjonowane klastry i singletony łączone są w pary klastrów. Selekcji dokonuje się zgodnie z warunkami (13) oraz (14). W pierwszej iteracji warunek spełnia jedynie singleton S_1 . Pary nie zostaną utworzone, więc część trzecia i czwarta algorytmu zostaje opuszczona. Zmienna sterująca l_{dend} jest dekrementowana do wartości 6.

Przy $l_{\text{dend}} = 6$ warunki (13) oraz (14) spełniają dwa singletony: S_1 oraz S_2 . Powstaje zbiór klastrów $C = \{c_1, c_2\}$, gdzie: $c_1=S_1$, $c_2=S_2$. Ze zbioru klastrów C tworzony jest podzbiór $SetC = \{(c_1, c_2)\}$. Liczba par klastrów w podzbiorze $SetC$ wynosi $L_{SetC}=1$ (zgodnie z zależnością (15)).

Algorytm przechodzi do części trzeciej, w której dla każdej pary klastrów z podzbioru $SetC$ obliczana jest miara odległości LCF. Miara odległości dla pary klastrów (c_1, c_2) wynosi $LCF(c_1, c_2)=4$. Jest to zgodne z zawartością Rys. 15, na którym zaznaczono najdłuższy wspólny przepływ dla pary klastrów (c_1, c_2) . Początek najdłuższego wspólnego przepływu pary (c_1, c_2) wynosi $LCB(c_1, c_2)=2$. Początek najdłuższego wspólnego przepływu jest określony na klastrze c_1 w parze klastrów (c_1, c_2) .

Para klastrów nie spełnia warunku (16) dlatego zostaje odrzucona. Zostaje opuszczona część czwarta algorytmu. Zmienna sterująca jest dekrementowana do wartości $l_{\text{dend}} = 5$. Warunki (13) oraz (14) są spełnione dla trzech singletonów: S_1 , S_2 oraz S_3 . Powstaje zbiór klastrów $C = \{c_1, c_2, c_3\}$, gdzie: $c_1=S_1$, $c_2=S_2$, $c_3=S_3$. Ze zbioru klastrów C tworzony jest podzbiór $SetC = \{(c_1, c_2), (c_1, c_3), (c_2, c_3)\}$. Liczba par klastrów w podzbiorze $SetC$ wynosi $L_{SetC}=3$. Dla każdej pary klastrów zostaje obliczona miara odległości: $LCF(c_1, c_2)=4$, $LCF(c_1, c_3)=3$, $LCF(c_2, c_3)=3$. Dla każdej pary klastrów zostaje również wyznaczony początek najdłuższego wspólnego przepływu: $LCB(c_1, c_2)=2$, $LCB(c_1, c_3)=2$, $LCB(c_2, c_3)=3$. Wszystkie trzy pary nie spełniają warunku (16) dlatego zostają odrzucone. Część czwarta algorytmu ponownie zostaje pominięta. Zmienna sterująca jest dekrementowana do wartości $l_{\text{dend}} = 4$.

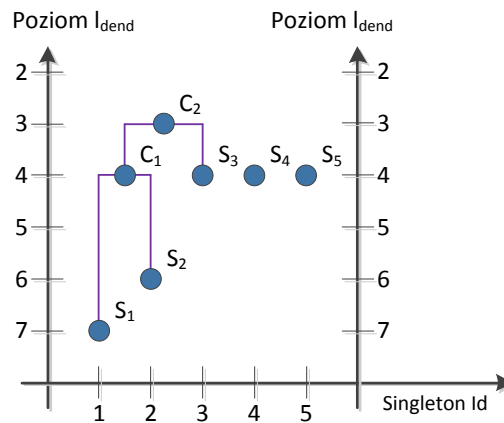
Przy wartości zmiennej sterującej $l_{\text{dend}} = 4$ wszystkie singletony S_1, S_2, S_3, S_4, S_5 spełniają warunek (13). Powstaje zbiór klastrów $C = \{c_1, c_2, c_3, c_4, c_5\}$, gdzie: $c_1=S_1$, $c_2=S_2$, $c_3=S_3$, $c_4=S_4$, $c_5=S_5$. Ze zbioru klastrów C tworzony jest podzbiór $SetC = \{(c_1, c_2), (c_1, c_3), (c_1, c_4), (c_1, c_5), (c_2, c_3), (c_2, c_4), (c_2, c_5), (c_3, c_4), (c_3, c_5), (c_4, c_5)\}$. Liczba par klastrów w podzbiorze $SetC$ wynosi $L_{SetC}=10$. Dla każdej pary klastrów zostaje obliczona miara odległości: $LCF(c_1, c_2)=4$, $LCF(c_1, c_3)=3$, $LCF(c_1, c_4)=1$, $LCF(c_1, c_5)=1$, $LCF(c_2, c_3)=3$, $LCF(c_2, c_4)=1$, $LCF(c_2, c_5)=1$, $LCF(c_3, c_4)=1$, $LCF(c_3, c_5)=1$, $LCF(c_4, c_5)=1$. Dla każdej pary klastrów zostaje również wyznaczony początek najdłuższego wspólnego przepływu: $LCB(c_1, c_2)=2$, $LCB(c_1, c_3)=2$, $LCB(c_1, c_4)=1$, $LCB(c_1, c_5)=1$, $LCB(c_2, c_3)=3$, $LCB(c_2, c_4)=1$, $LCB(c_2, c_5)=1$, $LCB(c_3, c_4)=2$, $LCB(c_3, c_5)=1$, $LCB(c_4, c_5)=1$. Warunek (16) spełnia jedna para klastrów: (c_1, c_2) . Dla niej zostaje utworzony klaster tymczasowy $C_{i=1} = \{c_1, c_2\}$. Pozostałe 9 par klastrów zostaje odrzuconych. Następuje przejście do części czwartej algorytmu. W części czwartej zostaje utworzone klastry stałe. Początkowo jedyny klaster tymczasowy posiada status 'N', a ponieważ jest to jedyny klaster w zbiorze klastrów tymczasowych zostaje on przekształcony w klaster zwykły $C_1\{KP_x^2, h\}$. Klaster zwykły $C_1\{KP_x^2, h\}$ posiada łańcuch zadań składający się z czterech zadań wykony-

wanych na maszynach $M2 \rightarrow M3 \rightarrow M4 \rightarrow M5$. Ten zbiór zadań stanowi najdłuższy wspólny przepływ dwóch połączonych singletonów: S_1 oraz S_2 . Klaster tymczasowy $C_{t=1}$ zmienia status na 'C'. Zmienna sterująca jest dekrementowana: $l_{\text{dend}} = 3$.

Przy wartości zmiennej sterującej $l_{\text{dend}} = 3$ warunki (13) oraz (14) są spełnione dla następujących klastrow: S_3, S_4, S_5 oraz $C_1 \{KP_x^2, h\}$ oznaczanym w skrócie C_1 . Singletony S_1 oraz S_2 nie mogą być wzięte pod uwagę, gdyż są klastrami podrzędnymi w klastrze zwykłym C_1 . Powstaje zbiór klastrow $C = \{c_1, c_2, c_3, c_4\}$, gdzie: $c_1=S_3, c_2=S_4, c_3=S_5, c_4=C_1$. Ze zbioru klastrow C tworzony jest podzbiór $SetC = \{(c_1, c_2), (c_1, c_3), (c_1, c_4), (c_2, c_3), (c_2, c_4), (c_3, c_4)\}$. Liczba par klastrow w podzbiorze $SetC$ wynosi $L_{SetC}=6$. Dla każdej pary klastrow zostaje obliczona miara odległości: $LCF(c_1, c_2)=1, LCF(c_1, c_3)=1, LCF(c_1, c_4)=3, LCF(c_2, c_3)=1, LCF(c_2, c_4)=1, LCF(c_3, c_4)=1$. Dla każdej pary klastrow zostaje również wyznaczony początek najdłuższego wspólnego przepływu: $LCB(c_1, c_2)=2, LCB(c_1, c_3)=1, LCB(c_1, c_4)=1, LCB(c_2, c_3)=1, LCB(c_2, c_4)=2, LCB(c_3, c_4)=2$. Warunek (16) spełnia jedna para klastrow: (c_1, c_4) . Dla niej zostaje utworzony klaster tymczasowy $C_{t=1} = \{c_1, c_4\}$. Pozostałe 5 par klastrow zostaje odrzuconych. Algorytm przechodzi do części czwartej. Ponownie, jedyny klaster tymczasowy posiada status 'N'. Jest to jedyny klaster w zbiorze klastrow tymczasowych, więc zostaje on przekształcony w klaster zwykły $C_2 \{KP_x^2, h\}$. Klaster zwykły $C_2 \{KP_x^2, h\}$ posiada łańcuch zadań składający się z trzech zadań wykonywanych na maszynach $M2 \rightarrow M3 \rightarrow M4$. Ten zbiór zadań stanowi najdłuższy wspólny przepływ połączonych singletona S_3 oraz klastra zwykłego C_1 . Klaster tymczasowy $C_{t=1}$ zmienia status na 'C'. Zmienna sterująca jest dekrementowana: $l_{\text{dend}} = 2$.

Przy wartości zmiennej sterującej $l_{\text{dend}} = 2$ warunki (13) oraz (14) są spełnione dla następujących klastrow: S_4, S_5 oraz C_2 . Tym razem oprócz singletonów S_1 oraz S_2 , dodatkowo nie są brane pod uwagę singleton S_3 oraz klaster zwykły C_1 , gdyż są one elementami podrzędnymi w klastrze zwykłym C_2 . Powstaje zbiór klastrow $C = \{c_1, c_2, c_3\}$, gdzie: $c_1=S_4, c_2=S_5, c_3=C_2$. Ze zbioru klastrow C tworzony jest podzbiór $SetC = \{(c_1, c_2), (c_1, c_3), (c_2, c_3)\}$. Liczba par klastrow w podzbiorze $SetC$ wynosi $L_{SetC}=3$. Dla każdej pary klastrow zostaje obliczona miara odległości: $LCF(c_1, c_2)=1, LCF(c_1, c_3)=1, LCF(c_2, c_3)=1$. Dla każdej pary klastrow zostaje również wyznaczony początek najdłuższego wspólnego przepływu: $LCB(c_1, c_2)=1, LCB(c_1, c_3)=2, LCB(c_2, c_3)=2$. Wszystkie trzy pary nie spełniają warunku (16) dlatego zostają odrzucone. Część czwarta algorytmu ponownie zostaje pominięta. Zmienna sterująca jest dekrementowana do wartości $l_{\text{dend}} = 1$, co oznacza wyjście z pętli iteracyjnej i zakończenie działania algorytmu VAHC.

Wynik działania algorytmu VAHC dla przytoczonego przykładu przedstawiono graficznie w formie dendrogramu. Dendrogram zamieszczono na Rys. 22. Zgodnie z przytoczonym przykładem, klaster C_1 powstał z połączonych singletonów S_1 oraz S_2 , natomiast klaster C_2 powstał z połączenia singletonów S_1, S_2 oraz S_3 .



Rys. 22. Graficzna prezentacja wyników – dendrogram dla danych z Tab. 5 [opracowanie własne].

Wirtualne komórki przedmiotowe mogą być tworzone jedynie dla klastrów zwykłych $C_z \{KP_x^2, h\}$. Jak widać z Rys. 22 istnieje możliwość utworzenia dwóch wariantów wirtualnych komórek przedmiotowych. Możliwe jest albo utworzenie wirtualnej komórki przedmiotowej na poziomie klastra zwykłego $C_1 \{KP_x^2, h\}$, albo na poziomie klastra zwykłego $C_2 \{KP_x^2, h\}$. Istnienie jednocześnie dwóch wirtualnych komórek przedmiotowych utworzonych dla klastrów zwykłych $C_1 \{KP_x^2, h\}$ oraz $C_2 \{KP_x^2, h\}$ jest niemożliwe, gdyż w skład obydwu klastrów wchodzi singletony S_1 oraz S_2 .

4.2. Planowanie przepływu produkcji z zastosowaniem wirtualnych komórek przedmiotowych

W drugiej fazie działania algorytmu ProduFlow PM dokonuje się planowania przepływu produkcji z wykorzystaniem wirtualnych komórek przedmiotowych. Zgodnie z przyjętymi założeniami w fazie pierwszej algorytmu ProduFlow PM (grupowania zleceń), nie jest możliwe wskazanie jednego rozwiązania pozwalającego zbudować harmonogram z wirtualnymi komórkami produkcyjnymi. Jest to zagadnienie o dużej złożoności obliczeniowej, tzw. problem NP-trudny. Złożoność obliczeniowa podejmowania decyzji o wyborze wirtualnych komórek przedmiotowych w planowaniu przepływu produkcji sprawia, że nie jest skuteczne zastosowanie metod poszukujących dokładne rozwiązanie. W zamian zastosowano algorytm genetyczny do poszukiwania racjonalnego rozwiązania problemu planowania przepływu produkcji z wykorzystaniem wirtualnych komórek przedmiotowych.

4.2.1. Opis funkcjonowania algorytmu genetycznego w nowej metodzie planowania przepływu produkcji

Podstawowym założeniem funkcjonowania algorytmu genetycznego w nowej metodzie planowania przepływu produkcji jest istnienie klastrów (grup zleceń utworzonych w pierwszej fazie algorytmu ProduFlowPM) podlegających realizacji w pewnym oknie planistycznym. Możliwe jest utworzenie wirtualnych komórek przedmiotowych zarówno dla singletona (łańcucha zadań pojedynczego zlecenia) jak i dla klastra zwykłego (wydzielonego łańcucha zadań grupy zleceń). Mając określony zbiór singletonów i klastrów dokonuje się planowania przepływu produkcji algorytmem genetycznym według klasycznego schematu działania tego

algorytmu. Specyfiką działania algorytmu genetycznego jest jego cykl pracy podzielony na następujące etapy:

- określenie reprezentacji problemu – budowa chromosomu,
- inicjalizacja populacji,
- tworzenie nowej generacji osobników - krzyżowanie,
- mutacja chromosomów potomnych,
- tworzenie planu przepływu oraz ocena przystosowania osobników z populacji,
- selekcja osobników w populacji.

Dwa pierwsze etapy wykonywane są jednorazowo przy każdorazowym uruchomieniu algorytmu genetycznego. Natomiast pozostałe etapy są wykonywane w pętli optymalizacyjnej. Są one powtarzane aż do osiągnięcia warunku wyjścia z pętli optymalizacyjnej. Po przeprowadzeniu serii prób stwierdzono, że wykonanie 20 iteracji pozwala osiągnąć skrócenie terminów realizacji zbioru zleceń. Dlatego w pracy przyjęto warunek wyjścia w pętli, którym jest stała liczba przeprowadzonych iteracji w pętli.

4.2.1.1.Reprezentacja wirtualnych komórek przedmiotowych w algorytmie genetycznym

Kluczową kwestią w planowaniu przepływu produkcji jest liczba możliwych do utworzenia wirtualnych komórek przedmiotowych. Patrząc na dendrogram zaprezentowany na Rys. 21 nie da się wykorzystać wszystkich klastrow jednocześnie. Można np. utworzyć wirtualną komórkę przedmiotową z klastra C_4 , nie można wówczas utworzyć wirtualnej komórki z klastra C_2 ani C_1 . Jednakże można w zamian utworzyć wirtualną komórkę z klastra C_2 , wówczas zabronione będzie utworzenie wirtualnych komórek z klastra C_1 lub C_4 . Wynika to stąd, że klastry wyższych rzędów są budowane na bazie tych samych singletonów (zleceń) co klastry podrzędne. Ponadto nie da się stwierdzić, czy wybór określonych klastrow do utworzenia wirtualnych komórek przedmiotowych przyczyni się do uzyskania krótszego terminu realizacji zbioru zleceń.

W celu wskazania klastrow, z których zostaną utworzone wirtualne komórki przedmiotowe opracowano sposób ich reprezentacji w algorytmie genetycznym. Zdefiniowano odwzorowanie między punktem w przestrzeni rozwiązań (zbiór wirtualnych komórek przedmiotowych) a punktem w przestrzeni reprezentacji (chromosom). Przyjęto binarną formę reprezentacji, tj. chromosom składa się z genów (jest łańcuchem genów) przyjmujących wartości 0 lub 1. Wartość 0 genu informuje, że dla danego klastra nie zostanie utworzona wirtualna komórka przedmiotowa. Wartość 1 genu informuje o utworzeniu wirtualnej komórki przedmiotowej dla danego klastra.

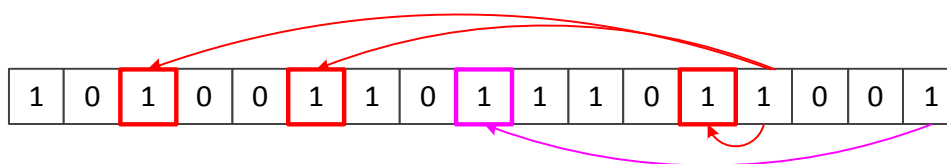
Liczba genów w chromosomie jest uzależniona od liczby utworzonych singletonów i klastrow zwykłych w pierwszej fazie algorytmu ProduFlow PM. Przykładowo na Rys. 21 istnieje 12 singletonów od S_1 do S_{12} oraz 5 klastrow zwykłych od C_1 do C_5 . Chromosom składa się w tym przypadku z 17 genów ułożonych w sposób pokazany na Rys. 23.

S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	C_1	C_2	C_3	C_4	C_5
-------	-------	-------	-------	-------	-------	-------	-------	-------	----------	----------	----------	-------	-------	-------	-------	-------

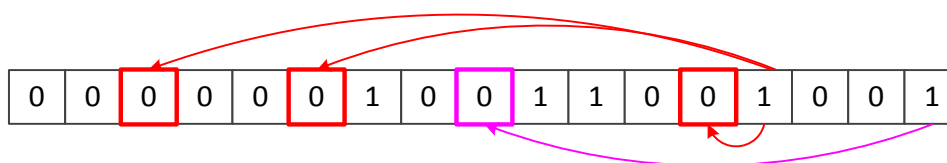
Rys. 23. Przykładowy chromosom reprezentujący zbiór klastrow z Rys. 21 [opracowanie własne].

Poszczególne geny w chromosomie przyjmują wartości 0 lub 1, stąd chromosom z Rys. 23 może przyjąć wygląd np. tak jak pokazano na Rys. 24.

Należy zwrócić uwagę na geny zaznaczone kolorem. Jak stwierdzono powyżej, pojedyncze zlecenie może przepływać przez jedną i tylko jedną wirtualną komórkę przedmiotową w oknie planistycznym. Jeżeli w wyniku działania algorytmu genetycznego zostanie otrzymany chromosom z wartościami 1 genów reprezentujących singletony i klastry dla tych samych zleceń, to taki chromosom podlega naprawie. W wyniku naprawy wszystkie geny reprezentujące klastry podrzędne (w stosunku do genu reprezentującego klastra nadrzędnego o wartości 1) przyjmują wartości 0. W takim przypadku chromosom z Rys. 24 przyjmie wygląd jak na Rys. 25.



Rys. 24. Kodowanie binarne chromosomu reprezentującego zbiór klastrów [opracowanie własne].



Rys. 25. Wygląd łańcucha genów chromosomu po procesie naprawy [opracowanie własne].

Oznacza to, że w procesie tworzenia wirtualnych komórek przedmiotowych pierwszeństwo mają klastry na najwyższych poziomach w dendrogramie.

4.2.1.2. Inicjalizacja populacji

Inicjalizacja populacji to etap wstępnego utworzenia chromosomów. Głównym jego zadaniem jest przygotowanie populacji do właściwej realizacji kolejnych etapów algorytmu genetycznego, wykonywanych w pętli optymalizacyjnej. Literatura podaje, że wielkość populacji nie jest ograniczona w żaden sposób. Dobór wielkości populacji jest sprawą dowolną i zależy jedynie od twórcy danej metody optymalizacyjnej. Natomiast co do populacji początkowej to ma ona spełniać podstawową cechę: uzyskane rozwiązanie dla populacji początkowej musi być rozwiązaniem dopuszczalnym [152].

Istnieją dwie metody inicjalizacji populacji: inicjalizacja losowa lub inicjalizacja adaptacyjna. W inicjalizacji losowej populacja początkowa pokrywa cały obszar rozwiązań dopuszczalnych. Natomiast w inicjalizacji adaptacyjnej stosuje się pewne metody, zwykle heurystyki, służące do wstępnego ukierunkowania rozwiązania ku rozwiązaniu optymalnemu. W drugiej metodzie istnieje niebezpieczeństwo skierowania procesu optymalizacji w stronę optimum lokalnego, dlatego w pracy przyjęto inicjalizację losową.

W rozpatrywanym problemie planowania przepływu produkcji z zastosowaniem wirtualnych komórek przedmiotowych przyjęto stałą wielkość populacji wejściowej, wynoszącą 10 osobników. Chromosomy z populacji wejściowej są rodzicami w pierwszej iteracji optymalizacyjnej.

4.2.1.3. Tworzenie nowej generacji osobników - krzyżowanie

Operator krzyżowania jest najważniejszym operatorem wykorzystywanym w ramach algorytmów genetycznych. Zadaniem operatora krzyżowania jest rekombinacja chromosomów poprzez wymianę materiału genetycznego (fragmentów łańcuchów genów) pomiędzy osobnikami z populacji rodziców. W wyniku krzyżowania powstaje nowa generacja potomna osobników, których chromosomy to zrekombinowane chromosomy rodziców.

W celu przeprowadzenia procesu krzyżowania, osobniki z populacji rodziców dobierane są losowo w pary. Każda para rodziców podlega procesowi krzyżowania z wykorzystaniem operatora krzyżowania w pojedynczym punkcie (ang. *one-point crossover*). W tym celu losowana jest liczba X w zakresie $\langle 0;1 \rangle$, pozwalająca ustalić punkt krzyżowania λ_{cross} . Punkt krzyżowania λ_{cross} jest obliczany ze wzoru:

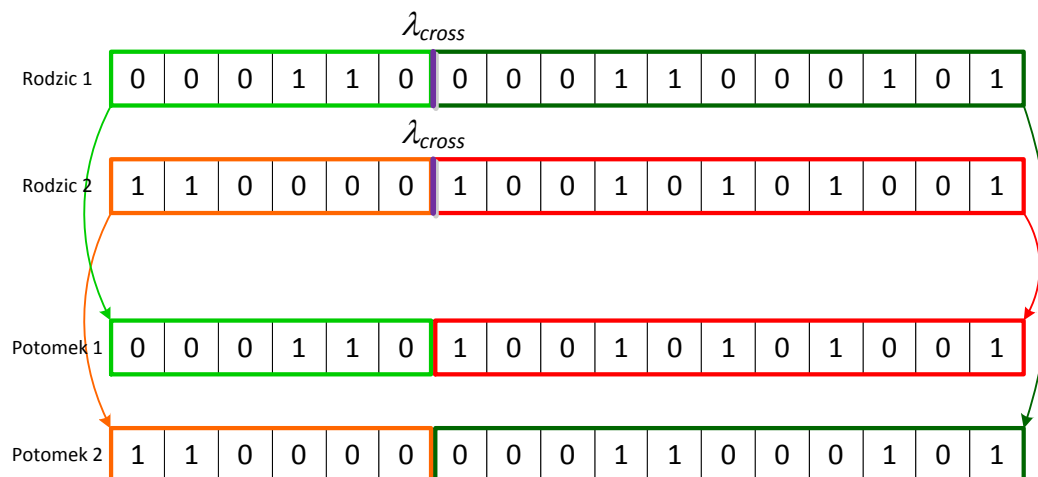
$$\lambda_{cross} = \lfloor (L_{Chrom} - 3) \cdot X \rfloor + 2 \quad (17)$$

gdzie: L_{Chrom} – długość łańcucha genów rodzica.

Z powyższego wzoru wynika, że punkt krzyżowania λ_{cross} może wystąpić po co najmniej drugim genie oraz przed co najwyżej przedostatnim genem chromosomu rodzica.

Dla każdej pary chromosomów rodziców tworzona jest para chromosomów potomnych. Pierwszy chromosom potomny powstaje w ten sposób, że fragment łańcucha genów od początku chromosomu do punktu krzyżowania jest brany z chromosomu rodzica pierwszego, natomiast fragment łańcucha genów od punktu krzyżowania do końca chromosomu jest brany z chromosomu rodzica drugiego. Analogicznie, drugi chromosom potomny powstaje w ten sposób, że fragment łańcucha genów od początku chromosomu do punktu krzyżowania jest brany z chromosomu rodzica drugiego, natomiast fragment łańcucha genów od punktu krzyżowania do końca chromosomu jest brany z chromosomu rodzica pierwszego. Poniżej zaprezentowano przykład krzyżowania pary chromosomów reprezentujących zbiór klastrów jak na Rys. 23.

Dwa chromosomy rodzice, których długość wynosi $L_{Chrom} = 17$, zostały poddane krzyżowaniu. Ustalono punkt krzyżowania λ_{cross} , który występuje po 6 genie w chromosomach rodziców. Na bazie skrzyżowanych chromosomów rodziców powstały dwa chromosomy potomne. Przebieg procesu krzyżowania zaprezentowano na Rys. 26.



Rys. 26. Krzyżowanie chromosomów za pomocą operatora one-point crossover [opracowanie własne].

4.2.1.4. Mutacja chromosomów potomnych

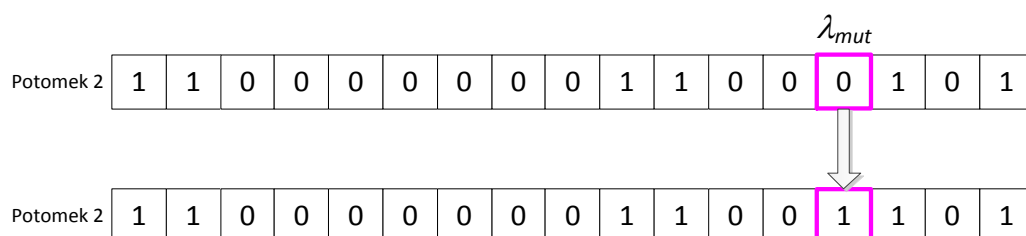
Mutacja osobników potomnych powoduje niewielkie modyfikacje łańcuchów genów. Założeniem leżącym u podstaw algorytmów genetycznych jest, aby mutacja zachodziła stosunkowo rzadko (około 1 raz na 1000 genów) [55]. W algorytmie genetycznym mutacja zachodzi, więc z małym prawdopodobieństwem rzędu $0,01 \div 0,05$.

W pracy przyjęto, że proces mutacji polega na zamianie wartości pojedynczego genu na wartość przeciwną. Mutacja jest mutacją punktową, przy czym losowana jest liczba X w zakresie $\langle 0;1 \rangle$, pozwalająca ustalić punkt mutacji λ_{mut} . Punkt mutacji λ_{mut} jest obliczany ze wzoru:

$$\lambda_{mut} = \lceil L_{Chrom} \cdot X \rceil \quad (18)$$

gdzie: L_{Chrom} – długość łańcucha genów rodzica.

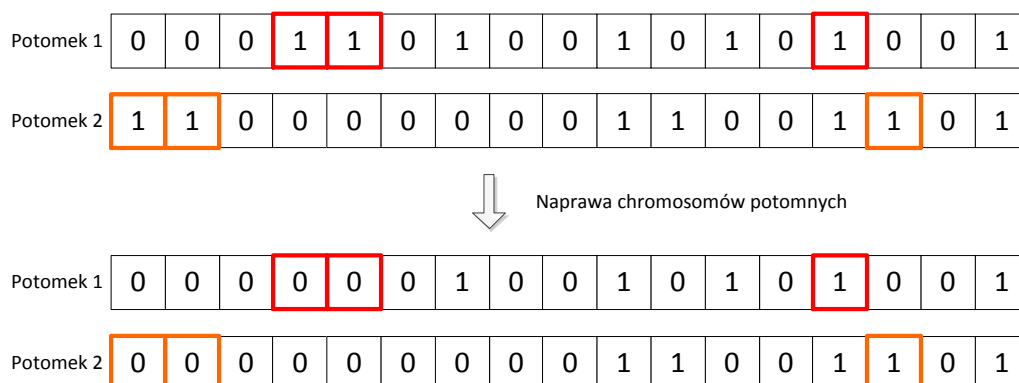
Przykładowo, patrząc na Rys. 26, mutacja zachodzi w chromosomie potomnym nr 2. Ustalono punkt mutacji $\lambda_{mut} = 14$. Gen czternasty posiada wartość 0, więc w wyniku mutacji zostanie on zamieniony na wartość 1. Proces mutacji zilustrowano na Rys. 27.



Rys. 27. Mutacja chromosomu potomnego za pomocą operatora mutacji punktowej [opracowanie własne].

4.2.1.5. Tworzenie planu przepływu oraz ocena przystosowania osobników

Chromosomy potomne reprezentują zbiór wariantów planu przepływu produkcji dla napływających zleceń. Zdekodowanie pojedynczego chromosomu potomnego pozwala określić wystąpienie wirtualnych komórek przedmiotowych w planie przepływu produkcji. Należy zaznaczyć, że na tym etapie chromosomy potomne wymagają naprawy zgodnie z założeniami podanymi w rozdziale 4.2.1.1. Kontynuując przykład, po procesie naprawy chromosomów potomnych nr 1 i 2 otrzymano łańcuchy genów zilustrowane na Rys. 28.



Rys. 28. Proces naprawy chromosomów potomnych [opracowanie własne].

Wartości 1 w poszczególnych genach informują, dla których singletonów i klastrów zwykłych zostaną utworzone wirtualne komórki przedmiotowe. Taka informacja zostaje

przekazana do określonej metody harmonogramowania produkcji, która umożliwia zaszeregowanie zadań i wirtualnych komórek przedmiotowych na liście zadań i wirtualnych komórek oczekujących na włączenie do harmonogramu. Opis wybranej metody harmonogramowania produkcji z wirtualnymi komórkami produkcji znajduje się w kolejnym rozdziale.

Informacją zwrotną z metody harmonogramowania produkcji wraz z wirtualnymi komórkami jest maksymalny termin zakończenia zlecenia spośród wszystkich zleceń $C_{z,y,max}(KP_x^2, h)$ (ang. *makespan*). Na jej podstawie dokonuje się przekształcenia minimalizowanego kryterium C_{max} do postaci maksymalizowanej funkcji przystosowania. Ocena przystosowania osobników jest wykonywana na populacjach rodziców oraz potomków. Na podstawie kryterium C_{max} sporządzany jest ranking wartości funkcji celu od największej do najmniejszej. Im dany chromosom znajduje się niżej w rankingu, tym przyjmuje on większą wartość funkcji przystosowania. Wartość funkcji przystosowania dla danego chromosomu jest równa jego pozycji γ w rankingu.

Pojedynczy chromosom może być zaszeregowany w rankingu na pozycjach od 1 do 20. Wynika to stąd, że w procesie pozycjonowania chromosomów biorą udział zarówno chromosomy potomne jak i ich rodzice. W pracy przyjęto założenie, że krzyżowaniu każdorazowo podlega 10 chromosomów rodziców a po krzyżowaniu powstaje dodatkowo 10 chromosomów potomnych. Sumarycznie daje to 20 chromosomów pozycjonowanych w rankingu.

4.2.1.6. Selekcja osobników w populacji

Selekcja jest ostatnim krokiem wykonywanym w pętli optymalizacyjnej algorytmu genetycznego. Jest ona wykorzystywana do wyboru osobników, na bazie, których będzie utworzona kolejna generacja potomna. Najpowszechniejszą metodą selekcji osobników jest selekcja bazująca na przystosowaniu osobników określana mianem selekcji z wykorzystaniem koła ruletki. Taką metodę wykorzystano w pracy do selekcji osobników podlegających krzyżowaniu.

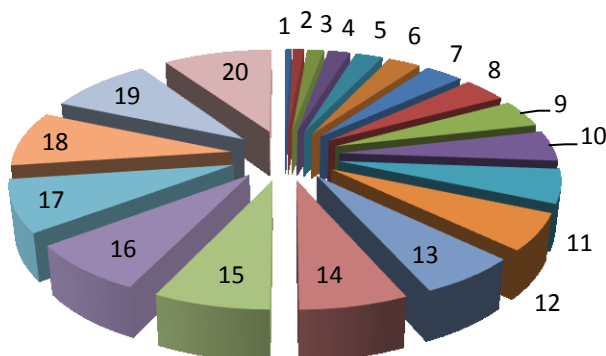
Selekcja z wykorzystaniem koła ruletki polega na tym, że najpierw dla każdego osobnika w populacji liczone jest prawdopodobieństwo jego wyboru według zależności:

$$f(\gamma) = \frac{\gamma}{\sum_{\gamma=1}^{20} \gamma} \quad (19)$$

gdzie: γ – numer pozycji danego chromosomu w rankingu przystosowania chromosomów.

Następnie buduje się koło ruletki, w którym każdemu osobnikowi przypisywane jest pole (część obwodu ruletki) o wielkości zgodnej z prawdopodobieństwem jego wyboru. Wielkości pól dla chromosomów spozycjonowanych w rankingu są stałe, gdyż prawdopodobieństwo wyboru chromosomu na określonej pozycji w rankingu również jest stałe. Im osobnik niżej w rankingu tym większe jego pole w kole ruletki. Natomiast miejsce lokalizacji spozycjonowanego osobnika jest identyfikowane poprzez zsumowanie wartości prawdopodobieństwa wyboru osobników występujących na pozycjach niższych w rankingu. Lokalizację kolejnych osobników znajdujących się na pozycjach od 1 do 20 w rankingu przedstawiono na Rys. 29.

Selekcja osobników odbywa się poprzez wielokrotne uruchomienie koła ruletki. W pracy przyjęto następujące założenia selekcji osobników. Po pierwsze reprezentacją koła ruletki jest zmienna losowa przyjmująca wartości w zakresie $\langle 0;1 \rangle$. Wylosowanie określonej liczby jest równoznaczne z wyborem osobnika znajdującego się w określonym miejscu koła ruletki. Osobnik znajdujący się na danej pozycji w rankingu może być wylosowany jeden raz. Liczba wyselekcjonowanych osobników wynosi zawsze 10.



Rys. 29. Lokalizacja osobników w ruletce według rankingu [opracowanie własne].

Uruchomienie koła ruletki jest realizowane tyle razy, aż zostanie wyselekcjonowanych 10 różnych osobników zarówno z populacji rodziców jak i z populacji potomnej. Wśród 10 wyselekcjonowanych osobników tworzy się losowo 5 par. Następnie pary osobników podlegają procesowi krzyżowania. Pętla optymalizacyjna jest wykonywana do momentu osiągnięcia warunku wyjścia z pętli. Warunkiem wyjścia z pętli jest zrealizowanie określonej liczby generacji.

4.2.2. Metoda szeregowania zadań dla potrzeb harmonogramowania z wirtualnymi komórkami przedmiotowymi

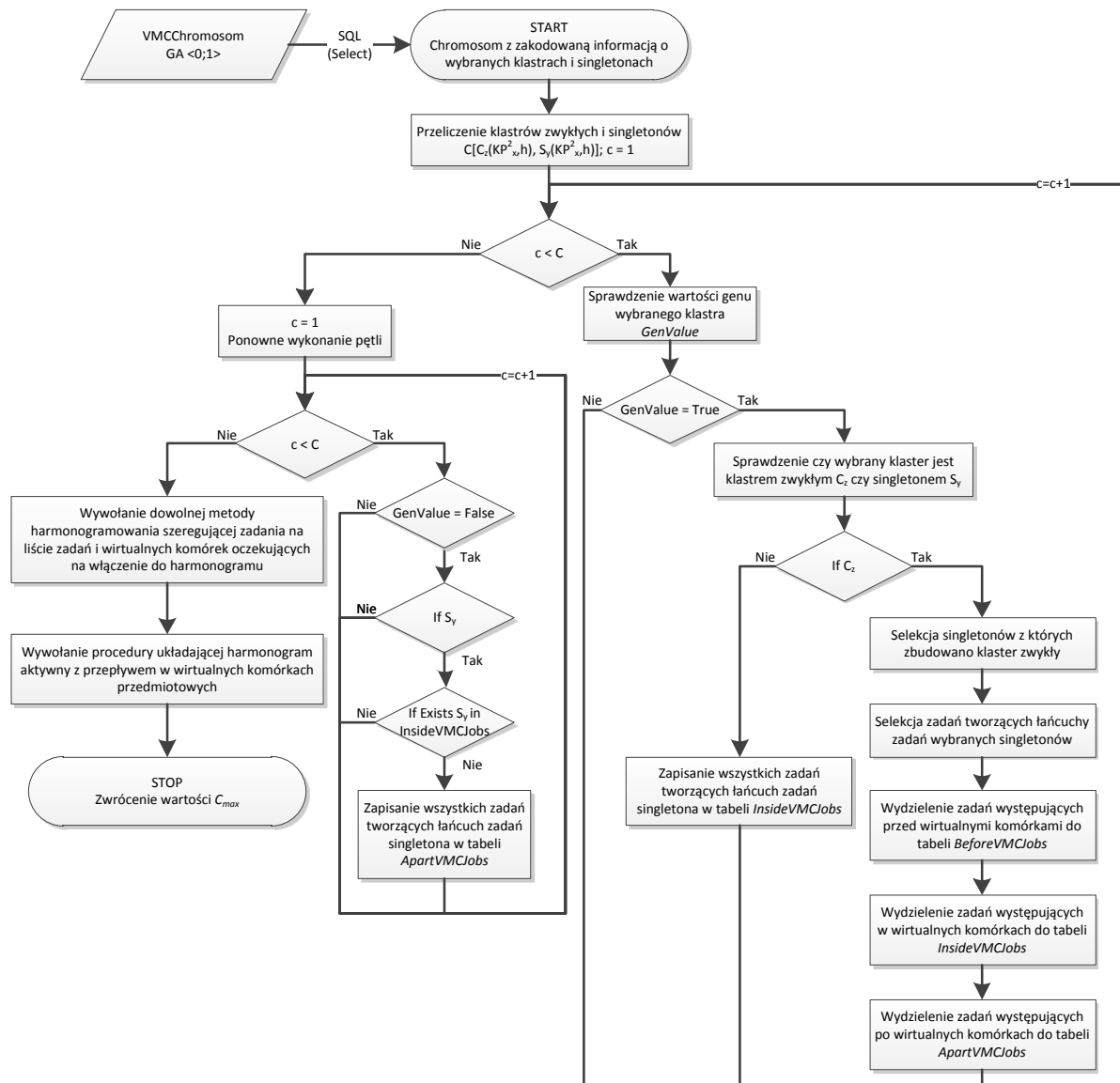
W rozdziale 4.2.1.5 zostało napisane, że chromosom z zakodowaną informacją o wybranych klastrach i singletonach do utworzenia wirtualnych komórek przedmiotowych jest przekazywany do pewnej metody harmonogramowania. W przypadku tworzenia harmonogramu z wirtualnymi komórkami przedmiotowymi wymagana jest pewna modyfikacja klasycznych metod harmonogramowania. W metodzie harmonogramowania z wirtualnymi komórkami proces szeregowania zadań nie może polegać na prostym układaniu zadań na liście zadań oczekujących na włączenie ich do harmonogramu. Pojawia się dodatkowe ograniczenie technologiczne wymuszające pierwszeństwo szeregowania zadań poprzedzających zadania wykonywane w wirtualnych komórkach przedmiotowych.

Nowe ograniczenie technologiczne powoduje konieczność klasyfikowania zadań do trzech grup:

- zadania związane z realizacją zgrupowanych zleceń, poprzedzające zadania wykonywane w wirtualnych komórkach przedmiotowych,
- zadania wykonywane w wirtualnych komórkach przedmiotowych,

- wszystkie zadania pozostałe (występujące po zadaniach z wirtualnych komórek oraz zadania pochodzące od zleceń nie przechodzących przez wirtualne komórki przedmiotowe).

Klasyfikację należy dokonać każdorazowo po przekazaniu chromosomu z zakodowaną informacją o wybranych klastrach i singletonach. Proces dekodowania chromosomu oraz klasyfikacji zadań przebiega według określonego algorytmu, którego schemat przedstawiono na Rys. 30.



Rys. 30. Schemat procesu dekodowania chromosomu, klasyfikacji zadań do trzech grup i wywołania metody szeregowania zadań z wirtualnymi komórkami przedmiotowymi [opracowanie własne].

Proces przedstawiony schematycznie na Rys. 30 przebiega w następujący sposób. Najpierw dokonuje się przeliczenia wszystkich klastrów i singletonów występujących w zakodowanym chromosomie. Długość chromosomu stanowi liczbę wszystkich klastrów zwykłych i singletonów. Następnie każdy klastrowy zakodowany w chromosomie podlega dwukrotnemu sprawdzeniu w dwóch pętlach iteracyjnych.

W pierwszej pętli iteracyjnej dokonuje się selekcji wyłącznie klastrów zwykłych i singletonów, których reprezentacja genu w chromosomie przyjmuje wartość 1. Oznacza to, że w

pierwszej kolejności zdekodowaniu polegają klastry zwykłe i singletony, na bazie, których zostaną utworzone wirtualne komórki przedmiotowe. Następnie sprawdza się, czy wybrano klaster zwykły czy też singleton. Jeżeli został wybrany singleton, wówczas wszystkie zadania występujące w łańcuchu zadań singletona będą wykonywane w wirtualnej komórce przedmiotowej. Zostają więc one zaklasyfikowane do grupy zadań wykonywanych w wirtualnych komórkach przedmiotowych. Dla tej grupy zadań przyjęto nazwę *InsideVMCJobs*. Natomiast jeżeli został wybrany klaster zwykły wówczas dokonuje się selekcji singletonów, z których powstał dany klaster. Singletony mają dłuższe łańcuchy zadań od łańcucha zadań danego klastra zwykłego. Należy sprawdzić dla każdego singletona, z których zadań w ramach jego łańcucha zadań utworzono łańcuch zadań klastra zwykłego. Jeżeli w łańcuchu singletona zidentyfikowano zadania występujące przed łańcuchem zadań klastra zwykłego wówczas takie zadania zostają sklasyfikowane do grupy zadań o nazwie *BeforeVMCJobs*. Zadania należące do łańcucha zadań klastra zwykłego klasyfikuje się do grupy zadań *InsideVMCJobs*. Natomiast jeżeli w łańcuchu singletona zidentyfikowano zadania występujące po łańcuchu zadań klastra zwykłego wówczas takie zadania zostają sklasyfikowane do grupy zadań o nazwie *ApartVMCJobs*.

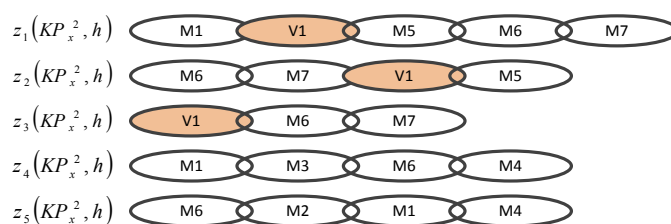
W drugiej pętli iteracyjnej dokonuje się selekcji wyłącznie singletonów, których reprezentacja genu w chromosomie przyjmuje wartość 0. Dla takich singletonów nie zostaną utworzone wirtualne komórki przedmiotowe. Zadania składające się na łańcuchy zadań tych singletonów klasyfikuje się do grupy zadań *ApartVMCJobs*.

Po wyjściu z drugiej pętli iteracyjnej zostaje wywołana metoda harmonogramowania produkcji. Metoda harmonogramowania posiada specjalny moduł służący do szeregowania zadań i wirtualnych komórek przedmiotowych. Na listę zadań oczekujących na włączenie do harmonogramu trafiają obiekty (łańcuchy zadań klastrów zwykłych oraz pozostałe zadania znajdujące się w grupach zadań *BeforeVMCJobs* oraz *ApartVMCJobs*) ze zmodyfikowanych łańcuchów zadań singletonów.

Przykładowo zakładając, że chromosom przybrał wartości genów tak, aby utworzyć wirtualną komórkę przedmiotową w sposób jak przedstawiono na Rys. 17, najpierw sklasyfikowano zadania do trzech grup zadań. Grupa zadań *BeforeVMCJobs* to zbiór następujących zadań: z_{1j_1} , z_{2j_1} , z_{2j_2} , gdzie z_1 oznacza numer zlecenia 1, j_1 oznacza numer zadania 1w określonym zleceniu. Grupa zadań *InsideVMCJobs* to zbiór następujących zadań: $v_1z_{1j_2}$, $v_1z_{1j_3}$, $v_1z_{1j_4}$, $v_1z_{2j_3}$, $v_1z_{2j_4}$, $v_1z_{2j_5}$, $v_1z_{3j_1}$, $v_1z_{3j_2}$, $v_1z_{3j_3}$, gdzie dodatkowo v_1 oznacza numer wirtualnej komórki przedmiotowej, przez którą przepływają określone zlecenia, w tym przypadku z_1 , z_2 oraz z_3 . Grupa zadań *ApartVMCJobs* to zbiór następujących zadań: z_{1j_5} , z_{1j_6} , z_{1j_7} , z_{2j_6} , z_{3j_4} , z_{3j_5} , z_{4j_1} , z_{4j_2} , z_{4j_3} , z_{4j_4} , z_{5j_1} , z_{5j_2} , z_{5j_3} , z_{5j_4} . Aby możliwe było uszeregowanie zadań występujących w trzech grupach zadań dokonano modyfikacji łańcuchów zadań z Rys. 17. Zmodyfikowane łańcuchy zadań dla pięciu zleceń przedstawiono na Rys. 31.

Modyfikacja łańcucha zadań polega na zastąpieniu zadań wykonywanych w wirtualnej komórce przedmiotowej jednym zadaniem reprezentującym przepływ produkcji przez wirtualną komórkę przedmiotową (zadania oznaczone kolorem na Rys. 31). Dzięki takiej modyfikacji upraszcza się proces szeregowania zadań i wirtualnych komórek. Wymagane jest jednak

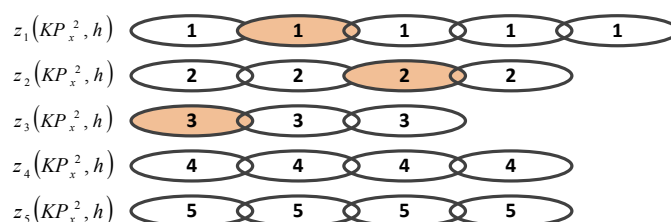
ustandaryzowanie postępowania prowadzącego do rozszyfrowania kolejności zapisanych obiektów na liście zadań oczekujących na włączenie do harmonogramu.



Rys. 31. Modyfikacja łańcuchów zadań do celu uszeregowania zadań i wirtualnych komórek przedmiotowych w procesie harmonogramowania produkcji [opracowanie własne].

W pracy przyjęto zasadę, że niezależnie od uszeregowania obiektów na liście zadań oczekujących na włączenie do harmonogramu, w pierwszej kolejności do harmonogramu trafiają zadania z grupy *BeforeVMCJobs*. Dla tych zadań tworzy się przepływ szeregowy. Następnie do harmonogramu włącza się wirtualne komórki przedmiotowe, wewnątrz których występuje przepływ równoległy. Oznacza to, że kolejne zadania występujące w wirtualnych komórkach przedmiotowych (występujące w grupie *InsideVMCJobs*) są wykonywane partiami transportowymi. Na końcu do harmonogramu trafiają zadania z grupy *ApartVMCJobs*. Zadania z tej grupy połączone są przepływem szeregowym.

Aby możliwe było przeprowadzenie szeregowania zadań i wirtualnych komórek produkcyjnych, zmodyfikowane łańcuchy zadań zostają zaszyfrowane w sposób pokazany na Rys. 32 .



Rys. 32. Szyfrowanie zadań i wirtualnych komórek przedmiotowych do celu ich szeregowania dowolną metodą [opracowanie własne].

Poszczególne zadania i wirtualne komórki przedmiotowe są zastępowane cyfrą reprezentującą numer odpowiadającego im zlecenia produkcyjnego. Liczba wystąpień danej cyfry jest równa liczbie zadań i wirtualnych komórek przedmiotowych w zmodyfikowanym łańcuchu zadań zlecenia. Przykładowo, liczba wystąpień cyfry 1, reprezentującej zadania i wirtualną komórkę zlecenia 1, wynosi 5.

Zaszyfrowanie zadań i wirtualnych komórek przedmiotowych umożliwia przeprowadzenie szeregowania dowolną metodą. W wyniku szeregowania zaszyfrowanych zadań i wirtualnych komórek przedmiotowych z Rys. 32 otrzymuje się listę obiektów oczekujących na włączenie do harmonogramu. Przykład uszeregowania zadań i wirtualnych komórek przedmiotowych z Rys. 32 na liście obiektów oczekujących na włączenie do harmonogramu pokazano na Rys. 33.

5	2	1	1	4	5	4	3	2	3	1	5	1	2	4	3	2	5	4	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Rys. 33. Lista zaszyfrowanych obiektów (zadań i wirtualnych komórek przedmiotowych) oczekujących na włączenie do harmonogramu [opracowanie własne].

Z Rys. 33 widać, że aby zaszyfrowany ciąg obiektów został poprawnie ułożony w harmonogramie, wymagane jest zastosowanie klucza szyfrowania. W przypadku szeregowania zadań i wirtualnych komórek przedmiotowych kluczem szyfrowania jest uporządkowany zbiór zadań w trzech grupach zadań: *BeforeVMCJobs*, *InsideVMCJobs* oraz *ApartVMCJobs*. Zbiór zadań sklasyfikowanych do wymienionych trzech grup zadań oraz listę uszeregowanych obiektów na liście obiektów oczekujących na ułożenie w harmonogramie produkcji przekazuje się do metody układającej harmonogram aktywny produkcji. Opis procesu deszyfrowania obiektów oraz układania harmonogramu aktywnego opisano w kolejnym podrozdziale.

4.2.3. Metoda układania harmonogramu z wirtualnymi komórkami przedmiotowymi

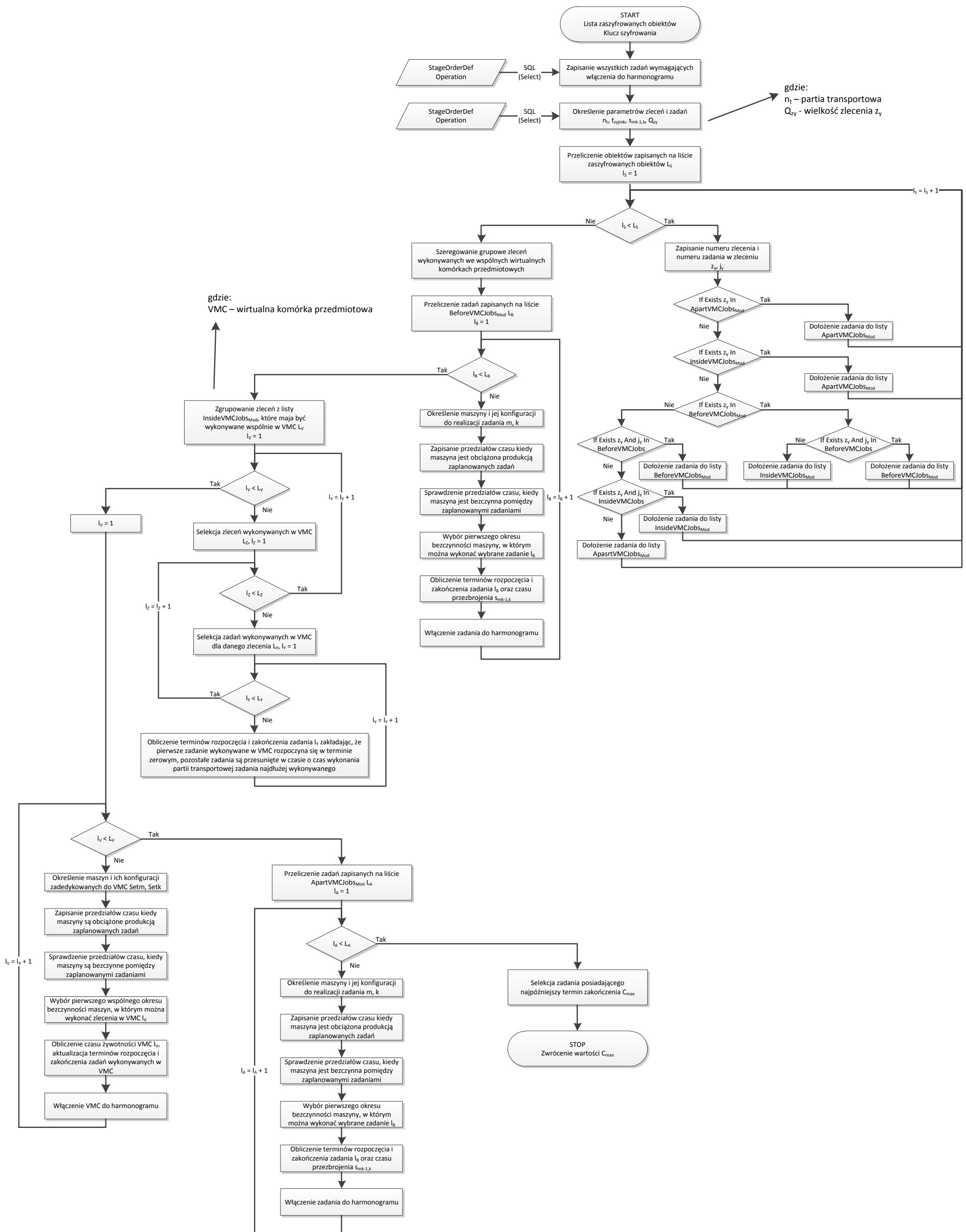
Znając uszeregowanie na liście obiektów oczekujących do ułożenia w harmonogramie oraz zbiory zadań sklasyfikowane w trzech grupach zadań można przystąpić do budowy harmonogramu aktywnego. Zgodnie z definicją przytoczoną w książce [121], za harmonogram aktywny uważa się taki harmonogram, że pewien obiekt zaszeregowany na liście obiektów oczekujących może być włączony do harmonogramu w najwcześniejszym możliwym czasie pod warunkiem zachowania ograniczeń technologicznych (sekwencja operacji) oraz ograniczeń zasobowych (warunek niewywłaszczania maszyn).

Przy budowie harmonogramu aktywnego z wirtualnymi komórkami przedmiotowymi należy wprowadzić pewne modyfikacje. Pierwszą z nich jest wprowadzenie fazy przygotowawczej. W fazie przygotowawczej należy w odpowiedni sposób odszyfrować listę obiektów oczekujących na wprowadzenie do harmonogramu. Należy rozdzielić obiekty z listy do trzech grup zadań posługując się kluczem szyfrowania, zachowując jednocześnie ich kolejność na liście. Przykładowo, korzystając z listy obiektów przedstawionej na Rys. 33 oraz klucza szyfrowania przedstawionego na Rys. 32, dokonano odszyfrowania listy. Zmodyfikowane zbiory obiektów uszeregowanych przedstawiono na Rys. 34.

Grupa <i>BeforeVMCJobs_{Mod}</i> :													
2	1	2											
Grupa <i>InsideVMCJobs_{Mod}</i> :													
1	3	2											
Grupa <i>ApartVMCJobs_{Mod}</i> :													
5	4	5	4	3	1	5	1	4	3	2	5	4	1

Rys. 34. Uszeregowanie odszyfrowanych obiektów na trzech listach: *BeforeVMCJobs_{Mod}*, *InsideVMCJobs_{Mod}* oraz *ApartVMCJobs_{Mod}* [opracowanie własne].

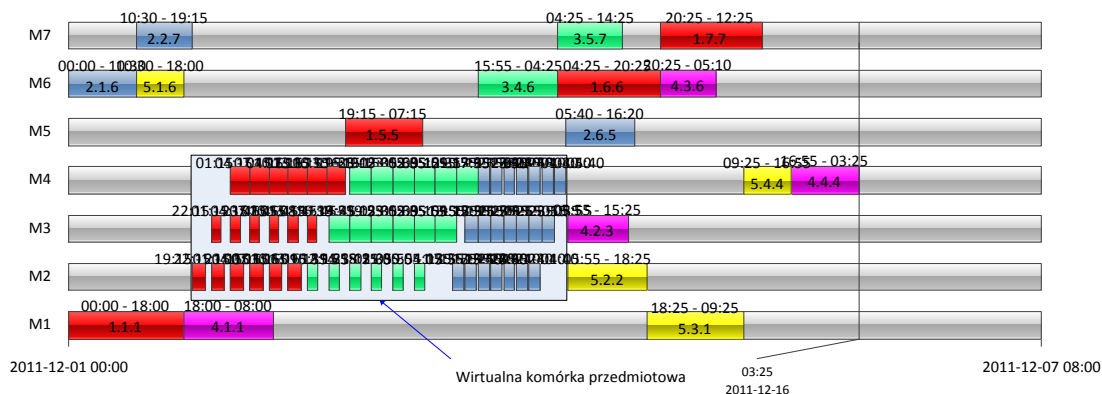
Uszeregowane zadania wprowadza się do harmonogramu aktywnego według schematu przedstawionego na Rys. 35. Najpierw do harmonogramu włączane są zadania z grupy *BeforeVMCJobs_{Mod}*. Następnie tworzy się wirtualne komórki produkcyjne, w których kolejność wykonywania poszczególnych zleceń jest ustanowiona na liście *InsideVMCJobs_{Mod}*. Na końcu włączane są do harmonogramu zadania z listy *ApartVMCJobs_{Mod}*.



Rys. 35. Schemat działania algorytmu służącego do budowy harmonogramu aktywnego z wirtualnymi komórkami przedmiotowymi [opracowanie własne].

Lista uszeregowanych zadań/zleceń $InsideVMCJobs_{Mod}$ jest dodatkowo modyfikowana w taki sposób, aby zadania/zlecenia wykonywane w jednej wirtualnej komórce przedmiotowej były uszeregowane po sobie. Stosuje się zasadę, że w ramach pojedynczej wirtualnej komórki przedmiotowej zadania/zlecenia są uszeregowane w kolejności pierwotnego ich występowania na liście $InsideVMCJobs$.

Przykładowo, posługując się schematem z Rys. 35 zbudowano harmonogram aktywny dla obiektów uszeregowanych jak na Rys. 33. Zbudowany harmonogram aktywny zaprezentowano na Rys. 36.



Rys. 36. Harmonogram aktywny job-shop dla uszeregowanych obiektów z przykładu omówionego w rozdziale 4.2.3 [opracowanie własne].

Otrzymany harmonogram job-shop uwzględniający występowanie wirtualnych komórek przedmiotowych posiada ustalony parametr C_{max} . Oznacza to, że poszczególne zlecenia produkcyjne również posiadają ustalone wartości parametrów $Cz_y(KP_x^2, h)$, będących zmiennymi decyzyjnymi w trakcie działania algorytmu ProduFlow PM. Zakończenie fazy drugiej algorytmu oznacza, że parametry te nie zmieniają się w trakcie dalszego działania algorytmu. W trzeciej części działania algorytmu pozostaje do ustalenia wartość parametru $d(KP_x^2, h)$ – wspólny termin realizacji zbioru zleceń $z_y(KP_x^2, h)$.

4.3. Planowanie okienkowe – określenie długości okienek planistycznych z wykorzystaniem metaheurystyki przeszukiwania tabu

Istotnym problemem w planowaniu przepływu produkcji w warunkach zmiennego zapotrzebowania jest zachowanie zaplanowanych terminów realizacji zleceń produkcyjnych. Problem jest szczególnie zauważalny w przypadku harmonogramowania *general job-shop*. W dynamicznie zmieniających się warunkach zapotrzebowania mogą pojawić się nieoczekiwane zlecenia produkcyjne. Problem nasila się, jeżeli weźmie się pod uwagę losowy charakter produkcji wynikający z nieoczekiwanych zdarzeń powodujących np. opóźnienie realizacji poszczególnych zadań na maszynach. Stosowanie harmonogramowania ciągłego (na zasadzie ciągłego doplanowywania napływających zleceń produkcyjnych) staje się nieefektywne. Wystarczy, że termin zakończenia jednego z zadań zostanie opóźniony. Wszystkie zlecenia produkcyjne, których zadania występują po zadaniu opóźnionym na tej samej maszynie, zostaną opóźnione. Spowoduje to opóźnienie terminów realizacji tych zleceń produkcyjnych.

W niniejszym rozdziale zaproponowano nowy sposób harmonogramowania produkcji z wykorzystaniem teorii planowania okienkowego. Planowanie okienkowe jest skuteczne dla

dynamicznego oraz stochastycznego harmonogramowania produkcji. Szczególnie przydatne staje się przy planowaniu przepływu produkcji z wykorzystaniem wirtualnych komórek przedmiotowych. W kolejnych podrozdziałach przedstawiono koncepcję planowania okienkowego.

Najistotniejszym czynnikiem planowania okienkowego, decydującym o jak największej wartości wskaźnika dotrzymania terminów realizacji zleceń produkcyjnych, jest długość okienek planistycznych. W kolejnych podrozdziałach wytłumaczono, dlaczego warto decydować o długości okienek planistycznych podczas harmonogramowania okienkowego. Opisano również metodę szacującą długość okienek planistycznych na podstawie prognozowanej produkcji.

4.3.1. Planowanie okienkowe w dynamicznych, stochastycznych warunkach produkcji

Planowanie okienkowe w warunkach produkcji małoseryjnej jest uzasadnione z kilku powodów. Po pierwsze, jeżeli w klasycznym problemie harmonogramowania job-shop wyeliminuje się ograniczenie braku nieplanowanych przestojów i opóźnień. To restrykcyjne ograniczenie powodowało brak możliwości implementacji metody harmonogramowania w praktyce. Zakładano, że maszyny się nie psują. Choć ograniczenie jest dużym ułatwieniem optymalizacyjnym to w warunkach rzeczywistych jest niemożliwe do wyeliminowania.

Po drugie, jeżeli w procesie harmonogramowania uwzględni się zlecenia priorytetowe mogące wywłaszczać maszyny zajęte wykonywaniem zaplanowanych wcześniej zadań. Jak wykazano w analizie literatury, brak możliwości wywłaszczania maszyn jest jednym z podstawowych założeń w teorii harmonogramowania. W praktyce nie ma systemów wytwórczych, które nie byłyby obciążone obsługą zleceń priorytetowych.

Trzeci powód jest związany z koniecznością integralności z systemami ERP. Postępująca ekspansja informatyzacyjna przedsiębiorstw produkcyjnych w kierunku stosowania systemów ERP wymusza stosowanie narzędzi dających się integrować z tymi systemami. Planowanie okienkowe jest doskonałym sposobem harmonogramowania general job-shop umożliwiającym integrację z systemem ERP poprzez dołączenie, jako moduł planowania szczegółowego job-shop, do zamkniętej pętli MRP (standard MRP wykorzystuje ideę planowania okresowego, przy czym okres planowania ma ustaloną, niezmienną długość).

4.3.2. Teoria planowania okienkowego

Powyżej wymieniono powody, dla których powstała koncepcja planowania okienkowego. Planowanie okienkowe jest skuteczne pod warunkiem posiadania prognozy zapotrzebowania na określony czas. Prognoza zapotrzebowania jest konieczna, gdyż planowanie okienkowe bazuje na pojęciu okienka planistycznego. Definicja okienka planistycznego brzmi następująco:

Definicja 2:

Jeżeli:

1. istnieje zbiór zleceń produkcyjnych z ustalonym wspólnym terminem rozpoczęcia zbioru zleceń, który poddano procesowi harmonogramowania job-shop wybraną metodą harmonogramowania,

2. znane są wskaźniki losowych zdarzeń podczas realizacji procesu harmonogramowania, to:

Możliwe jest określenie takiego wspólnego terminu zakończenia zbioru zleceń, że wpływ losowych zdarzeń nie wymusi odsunięcia tego terminu w czasie. Przedział czasu od terminu rozpoczęcia zbioru zleceń do terminu zakończenia zbioru zleceń nazywany jest oknem planistycznym. Różnica pomiędzy terminem zakończenia zbioru zleceń a terminem rozpoczęcia zbioru zleceń stanowi długość okna planistycznego.

W powyższej definicji wymagane jest wyjaśnienie kilku użytych pojęć. Po pierwsze za wspólny termin rozpoczęcia zbioru zleceń przyjmuje się najwcześniejszy czas, kiedy wszystkie zlecenia ze zbioru zleceń produkcyjnych $z(KP_x^2, h) = \{z_1(KP_x^2, h), z_2(KP_x^2, h), \dots, z_Y(KP_x^2, h)\}$ są dostępne, aby można było rozpocząć proces harmonogramowania. Dla wspólnego terminu rozpoczęcia zbioru zleceń przyjęto symbol $r(KP_x^2, h)$. Zbiór zleceń produkcyjnych wynika z prognozy zapotrzebowania. Prognoza zapotrzebowania może być sporządzona na podstawie napływających zamówień klientów, choć powinna być szacowana na podstawie danych historycznych (np. za pomocą technik eksploracji danych) o realizacji zleceń produkcyjnych [166]. Opracowanie prognozy zapotrzebowania wykracza poza zakres pracy. Przyjmuje się, że prognoza zapotrzebowania jest znana.

Należy również wyjaśnić pojęcie losowych zdarzeń. Za losowe zdarzenie uważa się nieoczekiwane działanie skutkujące dezorganizacją przygotowanego harmonogramu. Wyróżnia się następujące źródła nieoczekiwanych zdarzeń losowych:

- Nieplanowane przestoje. Są to awarie, usterki maszyn a także nieplanowane przerwy w realizacji zadań na maszynach. Powodują opóźnienia czasów zakończenia zadań. Nieplanowane przestoje opisane są za pomocą wskaźnika η_A . Zakres wartości wskaźnika η_A mieści się w przedziale $(0;1)$, gdzie 0 oznaczałoby totalny brak dostępności maszyny w określonym czasie, 1 oznacza brak nieplanowanych przestojów. Wskaźnik nieplanowanych przestojów oblicza się tak samo jak wskaźnik dostępności w standardzie OEE [32]:

$$\eta_A = \frac{\sum_{h=-1}^{-H} - \left(\frac{A_{Th} - T_{Downh}}{A_{Th}} \right)}{-H} \quad (20)$$

gdzie: $h=-1, -2, \dots, -H$ – identyfikatory okien planistycznych, w których zawarte są dane historyczne o realizacji zleceń produkcyjnych, A_{Th} – dostępny czas pracy maszyn w oknie planistycznym, T_{Downh} – czas trwania nieplanowanych przestojów w oknie planistycznym.

- Zlecenia priorytetowe. Są to zlecenia napływające w trakcie realizacji harmonogramu. Ich pojawienie się skutkuje koniecznością przebudowy harmonogramu, gdyż zlecenie priorytetowe jest zaliczane do grupy zleceń produkcyjnych (posiada wspólny termin zakończenia grupy zleceń). Terminy zakończenia niektórych zleceń mogą ulec wydłużeniu. Może również zmienić się wartość maksymalnego terminu zakończenia zleceń, co zostało uwzględnione we wskaźniku

priorytetowości η_P . Wartości wskaźnika η_P mieszają się w przedziale $(0;1)$.

Wskaźnik priorytetowości oblicza się ze wzoru:

$$\eta_P = \frac{\sum_{h=1}^{-H} \left(\frac{C_{\max h}}{C_{P\max h}} \right)}{-H} \quad (21)$$

gdzie: $C_{\max h}$ – maksymalny termin zakończenia zleceń wygenerowany podczas harmonogramowania bez zleceń priorytetowych, $C_{P\max h}$ – maksymalny termin zakończenia zleceń wygenerowany po dodaniu zleceń priorytetowych do harmonogramu.

W definicji okna planistycznego jest zapis, że określenie wspólnego terminu zakończenia zbioru zleceń (wspólny termin zakończenia zbioru zleceń jest dalej opisany symbolem $\xi(KP_x^2, h)$) jest możliwe po przeprowadzeniu harmonogramowania wybraną metodą. Wynika to z prostej przyczyny. Harmonogramowanie general job-shop jest problemem optymalizacyjnym NP-trudnym. W zależności od przyjętej metody harmonogramowania otrzyma się inną wartość wspólnego terminu zakończenia zbioru zleceń. Dlatego najpierw należy przeprowadzić proces harmonogramowania job-shop aby uzyskać wartości parametrów $C_{z,y}(KP_x^2, h)$ dla poszczególnych zleceń produkcyjnych, a następnie należy przeprowadzić szacowanie wartości wspólnego terminu zakończenia zbioru zleceń $\xi(KP_x^2, h)$, który zależy od występowania nieoczekiwanych zdarzeń w komórce produkcyjnej. Takie podejście do określenia wspólnego terminu zakończenia zbioru zleceń jest kontrowersyjne. Jednakże z punktu widzenia planowania przepływu produkcji jest bezpieczne ponieważ uodparnia plan przepływu produkcji od niebezpieczeństwa konieczności przesunięcia terminów zakończenia zbioru zleceń. Chociaż należy podkreślić, że jeżeli w trakcie realizacji planu przepływu produkcji wystąpią losowe zdarzenia wykraczające poza zakres dobranych wskaźników η_A oraz η_P , wówczas istnieje prawdopodobieństwo niedotrzymania wspólnego terminu zakończenia zbioru zleceń. Jednak ze względu na założenia do niniejszej pracy, wartości tych wskaźników nie są obliczane.

W celu określenia wspólnego terminu zakończenia zbioru zleceń należy przyjąć dodatkowo następujące parametry:

- Planowany termin realizacji zbioru zleceń $d(KP_x^2, h)$. Jest to wartość teoretyczna mniejsza od wspólnego terminu zakończenia zbioru zleceń $\xi(KP_x^2, h)$. Ustalenie wartości planowanego terminu realizacji zbioru zleceń pozwala stwierdzić czy poszczególne zlecenia ze zbioru zleceń są zakończone wcześniej bądź później w stosunku do tej wartości.
- Kara $Ez_y(KP_x^2, h)$ wcześniejszego wykonania zlecenia produkcyjnego $z_y(KP_x^2, h)$ w stosunku do planowanego terminu realizacji zbioru zleceń, liczona w jednostkach czasowych np. godzinowych. Im termin zakończenia zlecenia

produkcyjnego występuje wcześniej w stosunku do planowanego terminu realizacji zbioru zleceń, tym kara jest większa. Parametr oblicza się z zależności:

$$Ez_y(KP_x^2, h) = \max\{0, (d(KP_x^2, h) - Cz_y(KP_x^2, h))\} \quad (22)$$

gdzie: $Cz_y(KP_x^2, h)$ – termin zakończenia zlecenia produkcyjnego $z_y(KP_x^2, h)$.

Jest to zmienna decyzyjna algorytmu ProduFlow PM, której wartość została ustalona w drugiej fazie działania algorytmu.

- Koszt jednostkowy $k_E(KP_x^2, h)$ wcześniejszego zakończenia, liczony np. w złotych na godzinę wcześniejszego zakończenia zlecenia $z_y(KP_x^2, h)$. Koszt ten jest stały dla komórki produkcyjnej II stopnia w horyzoncie planowania.
- Kara $Tz_y(KP_x^2, h)$ późniejszego wykonania zlecenia produkcyjnego $z_y(KP_x^2, h)$ w stosunku do planowanego terminu realizacji zbioru zleceń, liczona w jednostkach czasowych, np. godzinowych. Im termin zakończenia zlecenia produkcyjnego występuje później w stosunku do planowanego terminu realizacji zbioru zleceń, tym kara jest większa. Parametr oblicza się z zależności:

$$Tz_y(KP_x^2, h) = \frac{[\max\{0, Cz_y(KP_x^2, h) - d(KP_x^2, h)\}]^2}{\xi(KP_x^2, h) - Cz_y(KP_x^2, h)} \quad (23)$$

- Koszt jednostkowy $k_T(KP_x^2, h)$ późniejszego zakończenia, liczony np. w złotych na godzinę późniejszego zakończenia zlecenia $z_y(KP_x^2, h)$. Koszt ten jest stały dla komórki produkcyjnej II stopnia w horyzoncie planowania.

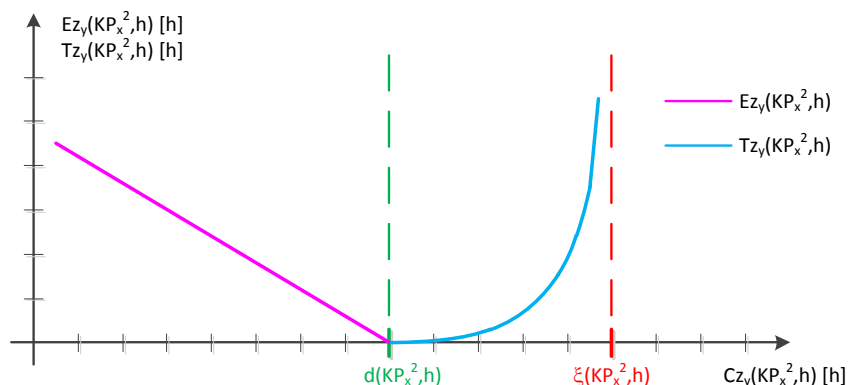
Ostatnim parametrem wymagającym wyjaśnienia sposobu obliczania jest wspólny termin zakończenia zbioru zleceń $\xi(KP_x^2, h)$. Parametr ten obliczany jest ze wzoru:

$$\xi(KP_x^2, h) = 8 \cdot \left\lceil \frac{d(KP_x^2, h)}{8 \cdot \eta_A \cdot \eta_P} \right\rceil \quad (24)$$

Wartość wspólnego terminu zakończenia zbioru zleceń jest zawsze dopasowywana do osmiogodzinnego trybu pracy. Wynik dzielenia wewnątrz nawiasów kwadratowych domkniętych u góry (funkcja sufit) jest zaokrąglany do najbliższej wartości całkowitej w górę. Wewnątrz nawiasów kwadratowych, planowany termin realizacji zbioru zleceń jest dzielony przez wskaźniki losowych zdarzeń. Wskaźniki losowych zdarzeń przyjmują wartości ułamkowe nie większe, niż 1, więc wynik działania funkcji zwróci wartość mniejszą lub równą wartości $d(KP_x^2, h)$. W skrajnym przypadku, gdy obydwa wskaźniki losowych zdarzeń przyjmują wartość 1, planowany termin realizacji zbioru zleceń jest równy terminowi zakończenia zbioru zleceń.

Dla pojedynczego zlecenia produkcyjnego można obliczyć wprost parametry kar wcześniejszego oraz późniejszego wykonania zlecenia. Zakładając, że planowany termin realizacji grupy zleceń został ustalony i wynosi $d(KP_x^2, h)$, wartości parametrów $Ez_y(KP_x^2, h)$ oraz $Tz_y(KP_x^2, h)$ będą zależeć od odsunięcia w czasie terminu zakończenia zlecenia $Cz_y(KP_x^2, h)$ w

stosunku do planowanego terminu realizacji grupy zleceń $d(KP_x^2, h)$. Można to zilustrować na wykresie pokazanym na .



Rys. 37. Wykres zależności kar wcześniejszego i późniejszego wykonania zlecenia od terminu zakończenia zlecenia, przy ustalonych wartościach planowanego terminu oraz terminu zakończenia zbioru zadań [opracowanie własne].

Z wykresu widać, że jeżeli termin zakończenia zlecenia produkcyjnego $Cz_y(KP_x^2, h)$ wypada wcześniej od wartości $d(KP_x^2, h)$, wówczas powstaje kara wcześniejszego wykonania zlecenia o wartości $Ez_y(KP_x^2, h)$, natomiast kara $Tz_y(KP_x^2, h)$ przyjmuje wartość 0. Natomiast jeżeli termin zakończenia zlecenia produkcyjnego $Cz_y(KP_x^2, h)$ wypada później od wartości $d(KP_x^2, h)$, wówczas powstaje kara późniejszego wykonania zlecenia o wartości $Tz_y(KP_x^2, h)$, natomiast kara $Ez_y(KP_x^2, h)$ przyjmuje wartość 0. Jeżeli termin zakończenia zlecenia produkcyjnego jest równy planowanemu terminowi realizacji grupy zleceń, tj. $Cz_y(KP_x^2, h) = d(KP_x^2, h)$ wówczas wartości obydwu kar przyjmują wartość 0.

Jeżeli zgodnie z definicją 2, zostanie przeprowadzony proces harmonogramowania, to wartości $Cz_y(KP_x^2, h)$, $y = 1, \dots, Y$ są znane i nie zmieniają się w harmonogramie wyznaczonym jedną metodą harmonogramowania. Jedynym parametrem, którego wartość można zmieniać jest, więc planowany termin realizacji zbioru zleceń $d(KP_x^2, h)$. W celu podjęcia decyzji o wartości planowanego terminu realizacji zbioru zleceń, należy opracować model matematyczny realizacji zlecenia. Model matematyczny realizacji zlecenia zawiera następujące składowe:

- Koszt wczesnego zakończenia zlecenia $Kz_{yE}(KP_x^2, h)$, obliczany ze wzoru:

$$Kz_{yE}(KP_x^2, h) = k_E(KP_x^2, h) \cdot Ez_y(KP_x^2, h) \quad (25)$$

- Koszt późnego zakończenia zlecenia $Kz_{yT}(KP_x^2, h)$, obliczany ze wzoru:

$$Kz_{yT}(KP_x^2, h) = k_T(KP_x^2, h) \cdot Tz_y(KP_x^2, h) \quad (26)$$

- Koszt składowania zapasu produkcji w toku $Kz_{yW}(KP_x^2, h)$. Koszt jest stały dla harmonogramu utworzonego określoną metodą. Zmienia się w zależności od wybranej metody harmonogramowania. Koszt obliczany jest ze wzoru:

$$Kz_{yW}(KP_x^2, h) = k_W(KP_x^2, h) \cdot \sum_{j=1}^J (tz_{yj-1,j} \cdot wz_y) \quad (27)$$

gdzie: $k_W(KP_x^2, h)$ - koszt jednostkowy składowania zapasu produkcji w toku, liczony np. w złotych na godzinę składowania zapasu razy jedną sztukę zapasu (koszt ten jest stały dla komórki produkcyjnej II stopnia w horyzoncie planowania), wz_y - wielkość zlecenia $z_y(KP_x^2, h)$.

- Koszt przetwarzania zlecenia produkcyjnego $Kz_{yP}(KP_x^2, h)$. Koszt jest stały dla harmonogramu utworzonego określoną metodą. Zmienia się w zależności od wybranej metody harmonogramowania ze względu na możliwość wystąpienia innych czasów przebrojenia. Koszt obliczany jest ze wzoru:

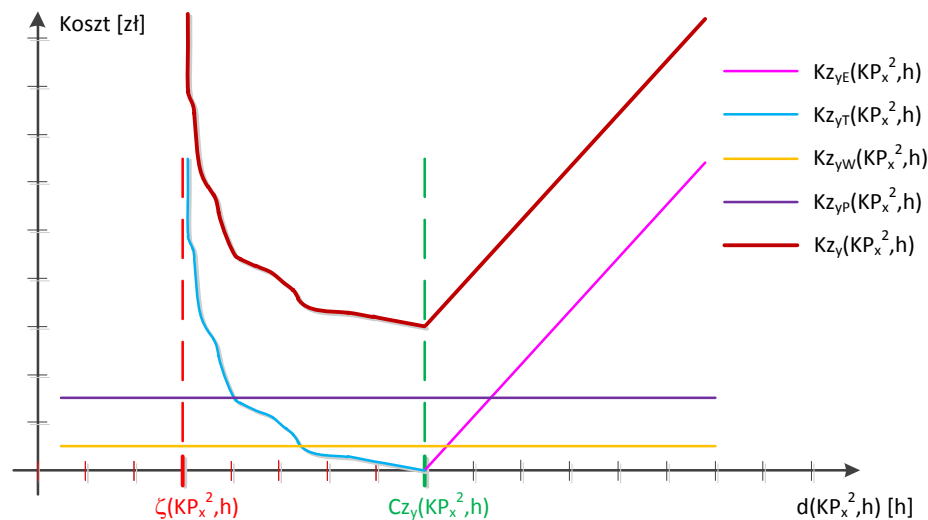
$$Kz_{yP}(KP_x^2, h) = \sum_{j=1}^J (s_{mk-1,k} + tz_{jmk} \cdot wz_y) \cdot k_m \quad (28)$$

gdzie: k_m - koszt maszynogodziny.

Model matematyczny realizacji zlecenia jest modelem kosztowym powstałym w wyniku zsumowania powyższych składowych i stanowi koszt całkowity realizacji zlecenia produkcyjnego:

$$Kz_y(KP_x^2, h) = Kz_{yE}(KP_x^2, h) + Kz_{yT}(KP_x^2, h) + Kz_{yW}(KP_x^2, h) + Kz_{yP}(KP_x^2, h) \quad (29)$$

Rys. 38 pokazuje w jaki sposób koszt realizacji zlecenia produkcyjnego $Kz_y(KP_x^2, h)$ reaguje na zmiany planowanego terminu realizacji zbioru zleceń $d(KP_x^2, h)$.



Rys. 38. Wykres zależności kosztu realizacji zlecenia produkcyjnego od planowanego terminu realizacji zbioru zleceń [opracowanie własne].

Z Rys. 38 widać, że planowany termin realizacji zbioru zleceń ma wpływ tylko na dwie pierwsze składowe równania (29). Dwie ostatnie składowe są stałe dla harmonogramu wyznaczonego określoną metodą harmonogramowania. Jednakże warto wprowadzić składowe kosztów składowania zapasu produkcji w toku oraz przetwarzania zlecenia produkcyjnego,

gdyż zmiana metody harmonogramowania wpływa na wartość obydwu kosztów (ma to sens w przypadku porównywania kilku metod harmonogramowania).

Z Rys. 38 widać również, że funkcja kosztu posiada jedną asymptotę pionową stanowiącą ograniczenie modelu kosztowego. Ograniczenie to należy rozpatrywać z punktu widzenia zbioru zleceń. Jeżeli znany jest zbiór zleceń produkcyjnych to istnieje możliwość znalezienia zlecenia, którego termin zakończenia jest największy $Cz_{y,\max}(KP_x^2, h)$. Dla takiego zlecenia musi być spełniony warunek:

$$\xi(KP_x^2, h) > Cz_{y,\max}(KP_x^2, h) \quad (30)$$

Podstawiając za $\xi(KP_x^2, h)$ równanie (26) otrzymuje się:

$$8 \cdot \left[\frac{d(KP_x^2, h)}{8 \cdot \eta_A \cdot \eta_P} \right] > Cz_{y,\max}(KP_x^2, h) \quad (31)$$

Przekształcając równanie (31) tak aby otrzymać warunek ograniczający dla $d(KP_x^2, h)$ uzyskuje się ostatecznie:

$$d(KP_x^2, h) > 8 \cdot \eta_A \cdot \eta_P \cdot \left[\frac{Cz_{y,\max}(KP_x^2, h)}{8} \right] = \zeta(KP_x^2, h) \quad (32)$$

Jest to dolny warunek ograniczający wartość planowanego terminu realizacji zbioru zleceń $d(KP_x^2, h)$ pokazany na Rys. 38 jako $\zeta(KP_x^2, h)$. Górny warunek ograniczający stanowi nierówność:

$$d(KP_x^2, h) \leq Cz_{y,\max}(KP_x^2, h) \quad (33)$$

Oznacza to, że wartość planowanego terminu realizacji zbioru zleceń nie może być mniejsza od maksymalnego terminu zakończenia zleceń pomnożonego przez wskaźniki losowych zdarzeń (wartość ta jest sprowadzona do ośmiogodzinnego trybu pracy), z drugiej strony nie może być większa niż maksymalny termin zakończenia zleceń.

Znając model matematyczny realizacji zlecenia produkcyjnego (równanie (29)) można określić zadanie optymalizacyjne dla zbioru zleceń. Zadaniem optymalizacyjnym jest równanie (8), które w tym miejscu jest powtórzone poniżej:

$$\min K(KP_x^2, h) = \sum_{y=1}^Y Kz_y(KP_x^2, h) \quad (\text{powt. 8})$$

gdzie $K(KP_x^2, h)$ jest zadaną funkcją celu (jest to koszt całkowity realizacji zbioru zleceń), parametr $d(KP_x^2, h)$ jest zmienną decyzyjną. Należy znaleźć taką wartość zmiennej decyzyjnej $d_0(KP_x^2, h)$ aby funkcja celu osiągnęła minimum. Warunkami ograniczającymi zadania optymalizacyjnego (8) są zależności (32) oraz (33).

Definicja matematyczna modelu optymalizacyjnego jest prosta, natomiast praktyczne wyznaczenie minimum funkcji celu już nie jest. Dzieje się tak, ponieważ dla kolejnych zleceń produkcyjnych $z_y(KP_x^2, h)$ wartości $Cz_y(KP_x^2, h)$ są różne. Powoduje to znaczne zniekształce-

nie funkcji celu oraz powstanie wielu minimów lokalnych. Przeszukiwanie przestrzeni rozwiązań algorytmami przeszukiwania sąsiedztwa lub algorytmami zachłannymi może doprowadzić do znalezienia minimum lokalnego, które nie musi być minimum globalnym. Sposobem na rozwiązanie problemu jest wykorzystanie metod przeszukiwania zupełnego. Jednakże w przypadku problemów optymalizacyjnych o dużej przestrzeni rozwiązań dopuszczalnych (tutaj wydłużone okno planistyczne prowadzi do konieczności przeszukiwania rozwiązań w szerokim zakresie wartości $d(KP_x^2, h)$) jest czasochłonne. W związku z powyższym w celu znalezienia minimum funkcji celu (8) wykorzystano prostą implementację metaheurystyki przeszukiwania tabu. W kolejnym rozdziale na przykładzie pokazano działanie algorytmu z wbudowanym mechanizmem przeszukiwania tabu.

4.3.3. Szacowanie planowanego terminu realizacji zbioru zleceń z wykorzystaniem metaheurystyki przeszukiwania tabu

W celu wyjaśnienia w jaki sposób została zaimplementowana metaheurystyka przeszukiwania tabu do rozwiązania zadania optymalizacyjnego (8) posłużono się następującym przykładem.

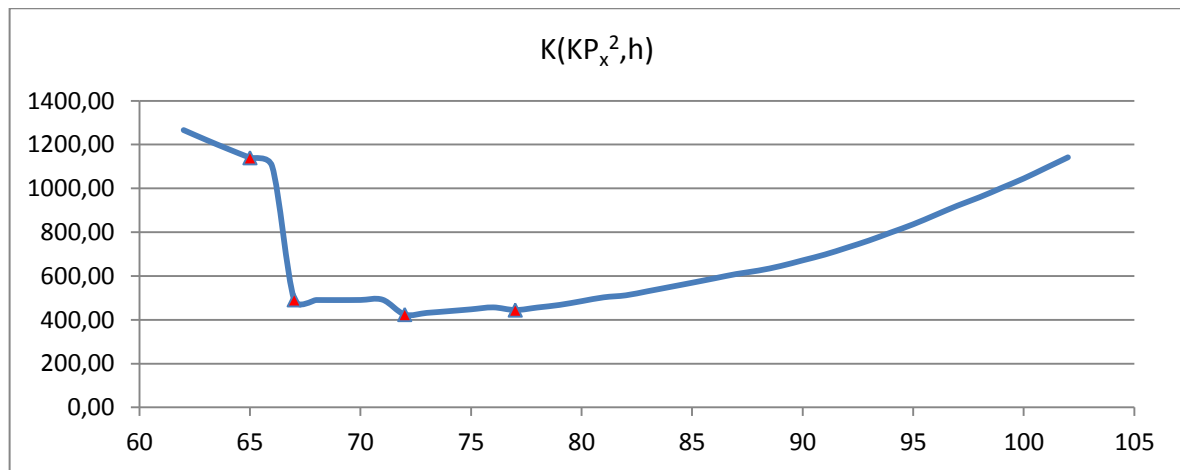
Danych jest 10 zleceń produkcyjnych, które poddano procesowi harmonogramowania wybraną metodą. Informacje o realizacji tych zleceń zestawiono w Tab. 9. W przykładzie nie uwzględniono kosztów stałych realizacji zleceń (tj. kosztu składowania zapasów produkcji w toku oraz kosztu przetwarzania).

Spośród 10 zleceń jedno z nich posiada wartość maksymalną terminu zakończenia $Cz_{y,max}(KP_x^2, h) = 102$ [h]. Znając wartość $Cz_{y,max}(KP_x^2, h)$ można obliczyć wartość dolnego ograniczenia $\zeta(KP_x^2, h) = 61,44$ [h]. Szukane rozwiązanie $d_0(KP_x^2, h)$ znajduje się więc w granicach $\zeta(KP_x^2, h) < d(KP_x^2, h) \leq Cz_{y,max}(KP_x^2, h)$, czyli $61,44 [h] < d(KP_x^2, h) \leq 102 [h]$. Jeżeli przyjmąc dokładność przeszukiwania 0,01 [h], wówczas w ograniczonej przestrzeni znajduje się 4056 rozwiązań dopuszczalnych. Wykres funkcji celu przedstawiono na Rys. 39.

Nr zlecenia	Czas zakończenia $Cz_y(KP_x^2, h)$
$z_1(KP_x^2, h)$	100
$z_2(KP_x^2, h)$	$102 = Cz_{y,max}(KP_x^2, h)$
$z_3(KP_x^2, h)$	92
$z_4(KP_x^2, h)$	58
$z_5(KP_x^2, h)$	91
$z_6(KP_x^2, h)$	50
$z_7(KP_x^2, h)$	48
$z_8(KP_x^2, h)$	89
$z_9(KP_x^2, h)$	95
$z_{10}(KP_x^2, h)$	79

Wskaźnik nieplanowanych przestołów	0,8
Wskaźnik priorytetowości	0,8
koszt jednostkowy wcześniejszego zakończenia	5
koszt jednostkowy późniejszego zakończenia	0,75

Tab. 9. Wykaz danych wejściowych do szacowania planowanego terminu realizacji zbioru zleceń [opracowanie własne].



Rys. 39. Wykres zależności kosztu całkowitego realizacji zbioru zleceń od przyjętej wartości planowanego terminu realizacji zbioru zleceń dla przykładu zawartego w Tab. 9 [opracowanie własne].

Funkcja celu posiada cztery minima lokalne, pokazane na Rys. 39. Im więcej zleceń produkcyjnych tym większa liczba minimów lokalnych. W celu wyznaczenia minimum globalnego opracowano algorytm DDAJSETS (ang. *Due Date Assignment Job Shop with Embedded Tabu Search*) z wbudowaną metaheurystyką przeszukiwania tabu. W kolejnych podrozdziałach opisano podstawowe działanie metaheurystyki tabu do celów wyznaczenia wartości $d_0(KP_x^2, h)$. Następnie przedstawiono budowę i działanie algorytmu DDAJSETS do szacowania planowanego terminu realizacji zbioru zleceń. Na koniec zaprezentowano działanie algorytmu DDAJSETS na przykładzie zawartym w Tab. 9.

4.3.3.1. Implementacja metaheurystyki przeszukiwania tabu w zadaniu minimalizacji kosztu całkowitego realizacji zbioru zleceń

Przeszukiwanie tabu jest metaheurystyką realizującą heurystyczną procedurę przeszukiwania lokalnego, wychodzącą jednak poza obszar lokalnego optimum. Metaheurystyka przeszukiwania tabu operuje na pojedynczym rozwiązaniu, które w kolejnych iteracjach jest poprawiane ze względu na optymalizację funkcji celu.

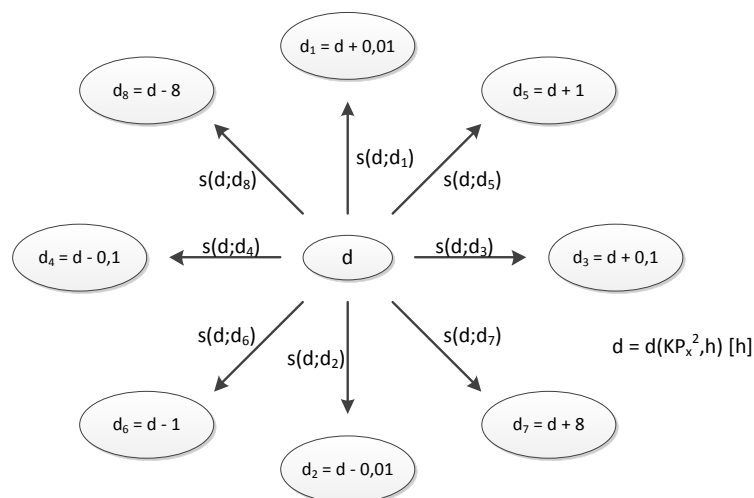
Rozwiązanie początkowe $d_{Start}(KP_x^2, h)$, od którego rozpoczyna się heurystyczna procedura przeszukiwania, może być losowe lub wyznaczone za pomocą innych heurystyk. W algorytmie DDAJSETS zastosowano sposób wyznaczenia rozwiązania początkowego, co opisano w kolejnym podrozdziale.

Jednym z najważniejszych czynników w metaheurystyce przeszukiwania tabu jest zdefiniowanie sąsiedztwa $N(d)$. Każde rozwiązanie należące do przestrzeni rozwiązań musi posiadać sąsiedztwo, którego osiągnięcie jest możliwe przez operację zwaną ruchem $s(d; d_n)$. Ruch ma na celu zmianę w obrębie rozwiązania. Do celów opracowania algorytmu DDAJSETS sąsiedztwo zdefiniowano w sposób pokazany na Rys. 40.

Sąsiedztwo pewnego rozwiązania d (d oznacza w uproszczeniu $d(KP_x^2, h)$) jest zbiorem następujących rozwiązań:

$$N(d) = \{d_1, d_2, d_3, \dots, d_8\} \quad (34)$$

gdzie bezpośrednimi sąsiadami pewnego rozwiązania d są rozwiązania, których osiągnięcie jest możliwe po wykonaniu ruchu dodawania lub odejmowania określonej wartości czasu. W pracy założono istnienie 8 bezpośrednich sąsiadów $d_n, n=1, \dots, 8$ każdego rozwiązania d .



Rys. 40. Definicja sąsiedztwa rozwiązania funkcji kosztu całkowitego realizacji zbioru zleceń [opracowanie własne].

Jeżeli istnieje 8 bezpośrednich sąsiadów rozwiązania d , to istnieje również 8 ruchów $s(d; d_n)$. W procesie optymalizacji z wykorzystaniem metaheurystyki przeszukiwania tabu, wybierany jest taki ruch spośród 8 ruchów, który prowadzi do maksymalnej poprawy wartości funkcji celu tj. zmniejszenia kosztu całkowitego realizacji zbioru zleceń $K(KP_x^2, h)$. Jeżeli w wyniku przeanalizowania 8 ruchów nie otrzyma się poprawy wartości funkcji celu to wybierany jest ruch prowadzący do minimalnego pogorszenia jej wartości (bo po osiągnięciu minimum lokalnego kolejny ruch może spowodować jedynie pogorszenie bieżącego kryterium optymalizacyjnego). Łatwo w ten sposób cofnąć się do obszaru wcześniej przeszukanego. W konsekwencji łatwo zapętlić poszukiwania rozwiązania w obrębie pewnego obszaru.

W celu uniknięcia możliwości zapętlenia się wokół minimum lokalnego stosuje się listę tabu $T(d)$. Lista tabu gromadzi realizowane ruchy $s(d; d_n)$. Jeżeli pewien ruch $s(d; d_n)$ zostanie zrealizowany oraz zapamiętany na liście tabu to do momentu jego istnienia na tej liście (przez zadaną liczbę iteracji) nie będzie mógł być ponownie osiągnięty. Jeżeli rozpatrywać bieżące rozwiązanie d oraz jego sąsiedztwo $N(d)$, to istnieje dopuszczalne sąsiedztwo $N^*(d)$ rozwiązania d , dla którego zachodzi zależność:

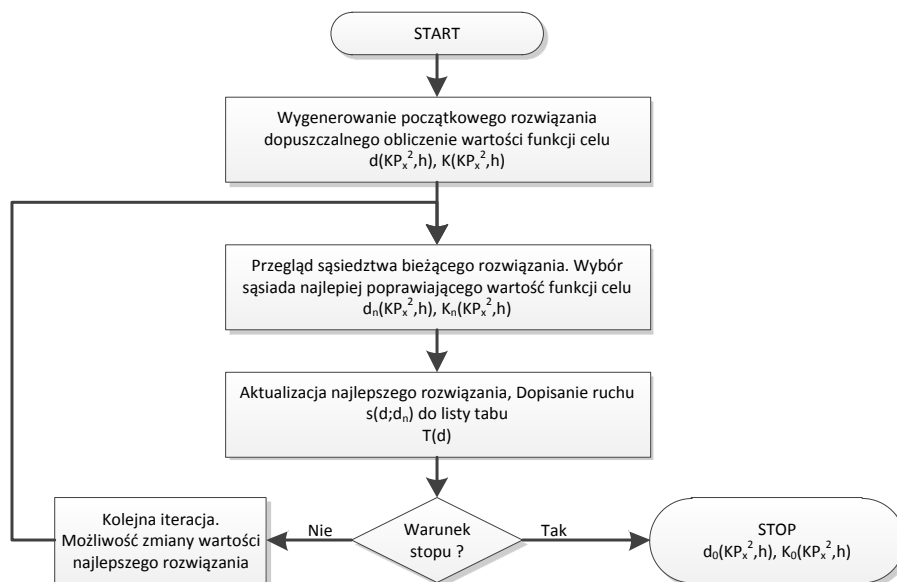
$$N^*(d) = N(d) \setminus T(d) \quad (35)$$

gdzie: $T(d)$ - są to rozwiązania ze statusem tabu

Rozwiązania są uszeregowane na liście tabu za pomocą kolejki FIFO. Długość listy tabu jest stała. Nie ulega zmianie w trakcie działania algorytmu. Jeżeli na liście tabu są zgromadzone ruchy w liczbie równej długości listy tabu, to dopisanie kolejnego ruchu do tej listy spowoduje usunięcie z niej najdłuższej zapamiętanego ruchu.

Pętla optymalizacyjna wykorzystująca metaheurystykę przeszukiwania tabu kończy działanie, jeżeli zostanie osiągnięty warunek wyjścia z pętli. Warunkiem wyjścia z pętli jest uzyskanie określonej liczby rozwiązań prowadzących do pogorszenia wartości funkcji celu.

Ogólny schemat działania metaheurystyki przeszukiwania tabu pokazano na Rys. 41. Schemat ten został opracowany w roku 1986 a następnie sformalizowany w 1989 przez twórcę koncepcji przeszukiwania tabu – Freda Glovera [53][54].



Rys. 41. Podstawowy schemat działania metaheurystyki przeszukiwania tabu [opracowanie własne na podst. [53][54]]

W kolejnym podrozdziale przedstawiono budowę algorytmu DDAJSETS, w którym zaimplementowano metaheurystykę przeszukiwania tabu. Opisano również sposób działania tego algorytmu.

4.3.3.2. Budowa i zasada działania algorytmu DDAJSETS z wbudowaną metaheurystyką przeszukiwania tabu

Algorytm DDAJSETS służy do szacowania planowanego terminu realizacji zbioru zleceń. Algorytm działa w oparciu o teorię planowania okienkowego. Ze względu na to, iż problem szacowania planowanego terminu realizacji zbioru zleceń jest problemem optymalizacyjnym, w którym za kryterium przyjmuje się minimalizację kosztu całkowitego realizacji zbioru zleceń, do poszukiwania rozwiązania optymalnego wykorzystano metaheurystykę przeszukiwania tabu.

Schemat algorytmu DDAJSETS przedstawiono na Rys. 42. Algorytm DDAJSETS zbudowany jest z pięciu części. Część pierwsza algorytmu DDAJSETS dotyczy tworzenia rozwiązania początkowego. Znając liczbę zleceń produkcyjnych na podstawie prognozy zapotrzebowania, po przeprowadzeniu harmonogramowania dowolną metodą można obliczyć maksymalny termin zakończenia grupy zleceń $Cz_{y,max}(KP_x^2, h)$. Maksymalny termin zakończenia grupy zleceń jest składową ograniczeń modelu matematycznego. Parametr ten pozwala obliczyć dolną granicę przestrzeni rozwiązań dopuszczalnych $\zeta(KP_x^2, h)$. Wartość $Cz_{y,max}(KP_x^2, h)$ stanowi jednocześnie górną granicę przestrzeni rozwiązań dopuszczalnych. Po ustaleniu dolnej i górnej granicy przestrzeni rozwiązań dopuszczalnych algorytm przechodzi

do obliczenia rozwiązania początkowego. Rozwiązanie początkowe jest obliczane z zależności:

$$d(KP_x^2, h) = \frac{\zeta(KP_x^2, h) + Cz_{y\max}(KP_x^2, h)}{2} \quad (36)$$

Rozwiązaniem początkowym $d(KP_x^2, h)$ jest, więc wartość średnia z dolnej i górnej granicy rozwiązań dopuszczalnych.

Znając wartość rozwiązania początkowego oblicza się w części drugiej początkową wartość kosztu całkowitego realizacji zbioru zleceń $K(KP_x^2, h)$. W celu obliczenia kosztu $K(KP_x^2, h)$ najpierw zostają obliczone w pętli iteracyjnej poszczególne składowe koszty dla każdego zlecenia. W kolejnych krokach obliczane są: w kroku pierwszym koszty składowania zapasu produkcji w toku (równanie (27)) oraz przetwarzania (równanie (28)), w kroku drugim koszty wczesnego oraz późnego zakończenia zlecenia (równania (25) oraz (26)), w kroku trzecim kosztu całkowitego realizacji zlecenia produkcyjnego według zależności (29). Po obliczeniu kosztów całkowitych realizacji poszczególnych zleceń algorytm przechodzi do obliczenia kosztu całkowitego realizacji zbioru zleceń według zależności (8).

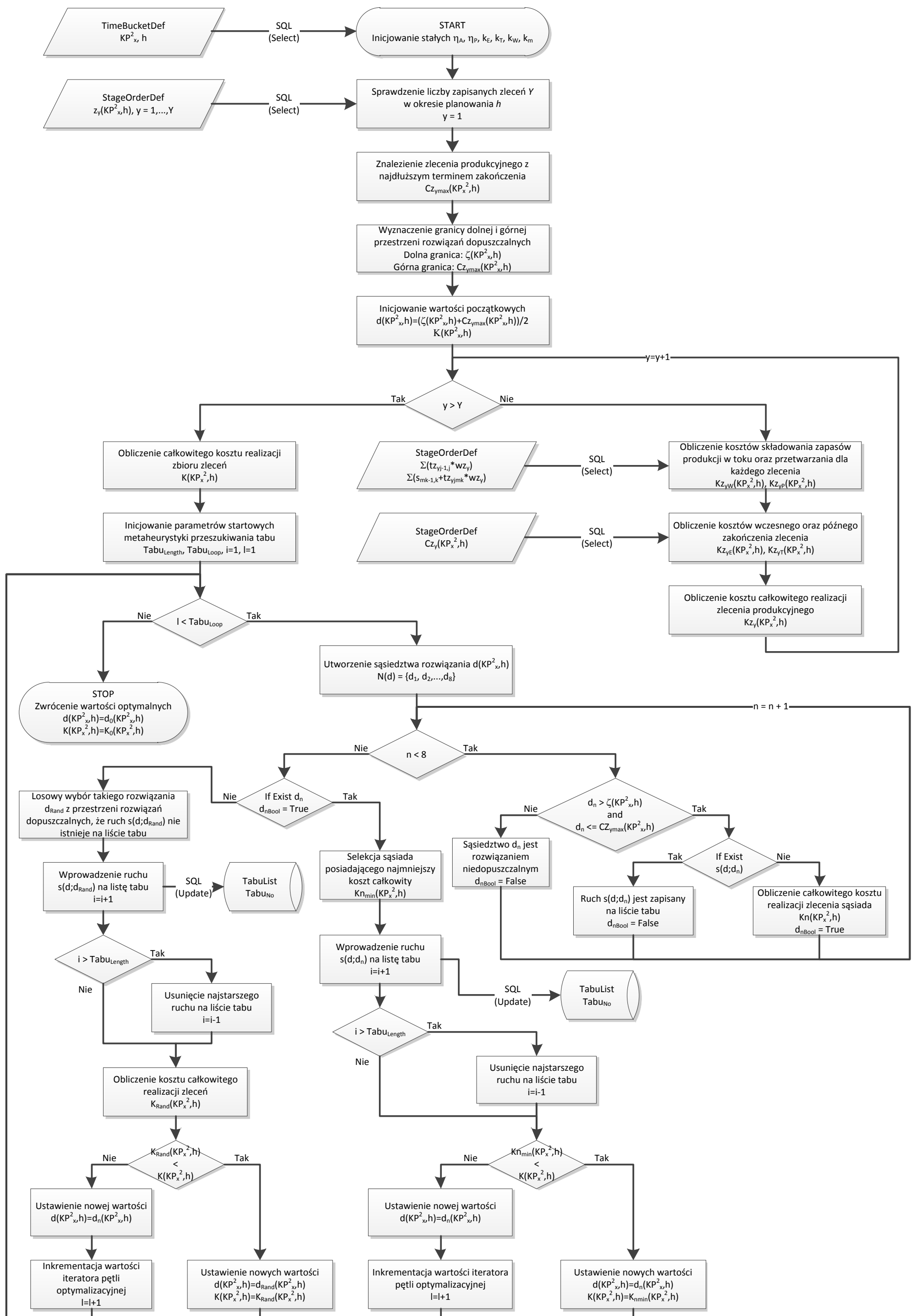
Część trzecia algorytmu poprzedzona jest inicjowaniem wartości parametrów niezbędnych do prawidłowego przeprowadzenia optymalizacji z wykorzystaniem metaheurystyki przeszukiwania tabu. Inicjuje się następujące parametry: i – zmienna lokalna (występująca tylko w trakcie działania algorytmu DDAJSETS) sterująca liczbą ruchów zapisanych na liście tabu, l – zmienna sterująca liczbą iteracji w pętli optymalizacyjnej przeszukiwania tabu, $Tabu_{Length}$ – parametr służący do określenia długości listy tabu, którego wartość obliczana jest ze wzoru:

$$Tabu_{Length} = \lceil 20 \cdot (Cz_{y\max}(KP_x^2, h) - \zeta(KP_x^2, h)) \rceil \quad (37)$$

co stanowi wartość 20% wszystkich rozwiązań dopuszczalnych, zaokrągloną w górę do liczby całkowitej, $Tabu_{Loop}$ – parametr służący do określenia liczby iteracji w pętli optymalizacyjnej, którego wartość oblicza się ze wzoru:

$$Tabu_{Loop} = \lceil 5 \cdot (Cz_{y\max}(KP_x^2, h) - \zeta(KP_x^2, h)) \rceil \quad (38)$$

co stanowi wartość 5% wszystkich rozwiązań dopuszczalnych, zaokrągloną w górę do liczby całkowitej. Warunkiem wyjścia z pętli optymalizacyjnej jest osiągnięcie wartości zmiennej $l = Tabu_{Loop}$. Zmienna l jest inkrementowana, jeżeli w danej iteracji uzyska się rozwiązanie pogarszające wartość funkcji celu $K(KP_x^2, h)$.



Rys. 42. Schemat procesu optymalizacji kosztu całkowitego realizacji zbioru zleceń [opracowanie własne].

W trzeciej części algorytmu DDAJSETS podejmowane są działania sprawdzające sąsiedztwo danego rozwiązania $d(KP_x^2, h)$. Każde rozwiązanie spośród 8 rozwiązań sąsiednich d_n sprawdzane jest, czy znajduje się w przestrzeni rozwiązań dopuszczalnych. Sprawdza się również czy ruch $s(d; d_n)$ nie znajduje się na liście tabu. Jeżeli powyższe warunki są spełnione, wówczas rozwiązanie sąsiednie d_n poddawane jest obliczeniu kosztu całkowitego realizacji zbioru zleceń $Kn(KP_x^2, h)$. Jeżeli spośród 8 sąsiadów co najmniej jedno ma obliczony koszt $Kn(KP_x^2, h)$ wówczas algorytm przechodzi do części czwartej. W przeciwnym razie algorytm przejdzie do części piątej.

Część czwarta algorytmu DDAJSETS służy do sprawdzenia sąsiada z najmniejszą wartością obliczonego kosztu całkowitego realizacji zbioru zleceń $Kn_{\min}(KP_x^2, h)$. Ruch $s(d; d_n)$ dla którego zachodzi $Kn_{\min}(KP_x^2, h)$, jest zapisywany na liście tabu. Jeżeli lista tabu jest pełna, tj. spełniony jest warunek $i = Tabu_{Length}$, wówczas usuwany jest ruch najdłużej zapamiętany na liście tabu. Następnie sprawdzany jest warunek czy koszt $Kn_{\min}(KP_x^2, h)$ obliczony dla ruchu sąsiedniego d_n jest mniejszy od dotychczasowego kosztu minimalnego zapisanego w zmiennej globalnej $K(KP_x^2, h)$. Jeżeli tak to za koszt minimalny $K(KP_x^2, h)$ przyjmuje się wartość kosztu $Kn_{\min}(KP_x^2, h)$. Jeżeli nie to zmienna l jest inkrementowana o 1. Algorytm przystępuje do kolejnej iteracji pętli optymalizacyjnej, jeżeli spełniony jest warunek $l < Tabu_{Loop}$. W kolejnej iteracji pętli optymalizacyjnej rozwiązanie d_n staje się rozwiązaniem początkowym d .

Część piąta algorytmu DDAJSETS służy do losowego generowania rozwiązania d_{Rand} z przestrzeni rozwiązań dopuszczalnych. Jeżeli zostanie znalezione rozwiązanie d_{Rand} , które dodatkowo spełnia warunek, że ruch $s(d; d_{Rand})$ nie istnieje na liście tabu, wówczas takie rozwiązanie jest sprawdzane pod kątem minimalizacji wartości funkcji celu $K_{Rand}(KP_x^2, h)$ w stosunku do dotychczasowej wartości $K(KP_x^2, h)$. Jeżeli wartość kosztu $K_{Rand}(KP_x^2, h)$ jest mniejsza od kosztu $K(KP_x^2, h)$ to za koszt minimalny $K(KP_x^2, h)$ przyjmuje się wartość kosztu $Kn_{\min}(KP_x^2, h)$. Jeżeli nie to zmienna l jest inkrementowana o 1. Algorytm przystępuje do kolejnej iteracji pętli optymalizacyjnej, jeżeli spełniony jest warunek $l < Tabu_{Loop}$. W kolejnej iteracji pętli optymalizacyjnej rozwiązanie d_{Rand} staje się rozwiązaniem początkowym d .

W wyniku działania algorytmu DDAJSETS zwracane jest rozwiązanie z przestrzeni rozwiązań dopuszczalnych, dla którego funkcja celu (koszt całkowity realizacji grupy zleceń) przyjmuje wartość minimalną. Rozwiązaniem jest wartość planowanego terminu realizacji grupy zleceń $d_0(KP_x^2, h)$. Posiadając wiedzę o zoptymalizowanej wartości planowanego terminu realizacji grupy zleceń, można przystąpić do obliczenia wspólnego terminu zakończenia zbioru zleceń $\xi_0(KP_x^2, h)$ zgodnie ze wzorem (24).

Wspólny termin zakończenia zbioru zleceń $\xi_0(KP_x^2, h)$ wiąże realizację zleceń produkcyjnych z terminami zamówień klientów. Jeżeli termin zamówień klientów zostanie potwier-

dzony to oznacza, że żadne zlecenie ze zbioru zleceń nie może być zakończone później niż wskazuje parametr $\xi_0(KP_x^2, h)$ bez poniesienia kary za opóźnienie realizacji zamówienia.

Obliczenie parametru $\xi_0(KP_x^2, h)$ oznacza równocześnie ustanowienie okna planistycznego, którego długość wynosi:

$$l_h(KP_x^2) = \xi_0(KP_x^2, h) - r(KP_x^2, h) \quad (39)$$

W kolejnym podrozdziale zaprezentowano działanie algorytmu DDAJSETS na przykładzie zamieszczonym w Tab. 9.

4.3.3.3. Algorytm DDAJSETS – przykład obliczeniowy

W niniejszym podrozdziale kontynuowane są rozważania dotyczące przykładu zamieszczonego w Tab. 9. Wiedząc, że dla 10 zleceń produkcyjnych znane są parametry $\zeta(KP_x^2, h) = 61,44$ oraz $Cz_{y\max}(KP_x^2, h)$, przystępuje się do obliczenia rozwiązania początkowego $d(KP_x^2, h)$. Korzystając ze wzoru otrzymuje się rozwiązanie początkowe $d(KP_x^2, h) = 81,72$. Algorytm DDAJSETS przechodzi do obliczenia kosztów całkowitych realizacji poszczególnych zleceń $Kz_y(KP_x^2, h)$ (część druga algorytmu). Jeżeli przyjąć dla każdego zlecenia stałe koszty składowania zapasów produkcji w toku $Kz_{yW}(KP_x^2, h) = 15$ oraz koszty przetwarzania $Kz_{yP}(KP_x^2, h) = 50$ to pozostaje do obliczenia koszt wczesnego $Kz_{yE}(KP_x^2, h)$ oraz późnego $Kz_{yT}(KP_x^2, h)$ zakończenia każdego zlecenia. Na końcu cztery wymienione koszty są sumowane dając całkowity koszt realizacji określonego zlecenia $Kz_y(KP_x^2, h)$. Wszystkie koszty dla poszczególnych zleceń zebrano w Tab. 10.

$z_y(KP_x^2, h)$	$Kz_{yE}(KP_x^2, h)$	$Kz_{yT}(KP_x^2, h)$	$Kz_{yW}(KP_x^2, h)$	$Kz_{yP}(KP_x^2, h)$	$Kz_y(KP_x^2, h)$
$z_1(KP_x^2, h)$	0,00	8,95	15,00	50,00	73,95
$z_2(KP_x^2, h)$	0,00	11,86	15,00	50,00	76,86
$z_3(KP_x^2, h)$	0,00	2,20	15,00	50,00	67,20
$z_4(KP_x^2, h)$	118,60	0,00	15,00	50,00	183,60
$z_5(KP_x^2, h)$	0,00	1,75	15,00	50,00	66,75
$z_6(KP_x^2, h)$	158,60	0,00	15,00	50,00	223,60
$z_7(KP_x^2, h)$	168,60	0,00	15,00	50,00	233,60
$z_8(KP_x^2, h)$	0,00	1,02	15,00	50,00	66,02
$z_9(KP_x^2, h)$	0,00	4,01	15,00	50,00	69,01
$z_{10}(KP_x^2, h)$	13,60	0,00	15,00	50,00	78,60
Koszt całkowity realizacji zbioru zleceń $K(KP_x^2, h)$					1139,19

Tab. 10. Zestawienie kosztów całkowitych realizacji poszczególnych zleceń [opracowanie własne].

W Tab. 10 na żółto zaznaczono całkowity koszt realizacji zbioru zleceń $K(KP_x^2, h)$, który został obliczony po wyjściu z pętli iteracyjnej w części drugiej algorytmu. Koszt ten trak-

towany jest jako wartość optymalna w danym momencie działania algorytmu. Algorytm przystępuje do optymalizacji z wykorzystaniem metaheurystyki przeszukiwania tabu, której celem jest poprawa (zmniejszenie) wartości kosztu $K(KP_x^2, h)$.

Przed rozpoczęciem przeszukiwania rozwiązań w pętli optymalizacyjnej, algorytm inicjuje wartości zmiennych sterujących $i=1, l=1$. Zostają również obliczone wartości zmiennych przechowujących długość listy tabu oraz warunkującej wyjście z pętli optymalizacyjnej: $Tabu_{Length} = 812$ oraz $Tabu_{Loop} = 203$.

Rozpoczyna się pętla optymalizacyjna. Algorytm przechodzi do części trzeciej. Sprawdzane jest sąsiedztwo rozwiązania $d(KP_x^2, h)$. W kolejnych krokach dla każdego sąsiada $d_n, n=1, \dots, 8$ sprawdzane są warunki dopuszczalności rozwiązania oraz istnienia ruchu na liście tabu, a także obliczenia wartości funkcji celu. Wyniki działania algorytmu z części trzeciej zestawiono w Tab. 11.

Z Tab. 11 widać, że całe sąsiedztwo rozwiązania spełnia $d_n(KP_x^2, h)$ warunki, tzn. każde rozwiązanie sąsiednie jest rozwiązaniem dopuszczalnym oraz żaden ruch $s(d; d_n)$ nie jest zapisany na liście tabu. Dla każdego rozwiązania obliczono, więc wartość funkcji celu.

Po wyjściu z części trzeciej algorytmu, sprawdzany jest warunek, czy istnieje co najmniej jeden sąsiad rozwiązania optymalnego, dla którego obliczono wartość funkcji celu. Warunek jest spełniony, więc algorytm przystępuje do działania z czwartej części. Zostaje wyselekcjonowany sąsiad posiadający najmniejszą wartość funkcji celu (w Tab. 11 jest on zaznaczony kolorem żółtym). Do dalszych działań zostaje wybrane rozwiązanie sąsiednie $d_8(KP_x^2, h) = 73,72[h]$, ponieważ $K_8(KP_x^2, h) = K_{n_{\min}}(KP_x^2, h) = 1071,44$. Ruch $s(d; d_8) = s(81,72; 73,72)$ zostaje umieszczony na liście tabu. Zmienna sterująca jest tylko inkrementowana: $i=2$. Żaden ruch nie zostaje usunięty z listy tabu (aby najwcześniej zapisany ruch został usunięty z listy tabu to wartość zmiennej sterujących musi osiągnąć wartość $Tabu_{Length} + 1 = 813$). Algorytm przystępuje do porównania optymalnej wartości funkcji celu $K(KP_x^2, h)$ z wartością obliczoną dla sąsiada $K_8(KP_x^2, h)$. Wartość funkcji celu jest mniejsza od dotychczasowej wartości optymalnej. Zostają, więc ustawione nowe wartości rozwiązania optymalnego $d(KP_x^2, h) = d_8(KP_x^2, h)$ oraz kosztu $K(KP_x^2, h) = K_8(KP_x^2, h)$. Zmienna sterująca l nie jest inkrementowana w tym przypadku, ponieważ nowe rozwiązanie jest lepsze od poprzedniego. Algorytm powraca do początku pętli optymalizacyjnej. Rozpoczyna się nowa iteracja.

$d_n(KP_x^2, h)$	Wartość	Ruch	$\zeta < d_n \leq Cz_{y \max}$	Lista tabu	Koszt
$d_1(KP_x^2, h)$	81,73	$s(d; d_1)$	Warunek spełniony	Brak ruchu	1139,35
$d_2(KP_x^2, h)$	81,71	$s(d; d_2)$	Warunek spełniony	Brak ruchu	1139,03
$d_3(KP_x^2, h)$	81,82	$s(d; d_3)$	Warunek spełniony	Brak ruchu	1140,81
$d_4(KP_x^2, h)$	81,62	$s(d; d_4)$	Warunek spełniony	Brak ruchu	1137,57
$d_5(KP_x^2, h)$	82,72	$s(d; d_5)$	Warunek spełniony	Brak ruchu	1149,82

$d_6(KP_x^2, h)$	80,72	$s(d; d_6)$	Warunek spełniony	Brak ruchu	1123,17
$d_7(KP_x^2, h)$	89,72	$s(d; d_7)$	Warunek spełniony	Brak ruchu	1278,02
$d_8(KP_x^2, h)$	73,72	$s(d; d_8)$	Warunek spełniony	Brak ruchu	1071,44

Tab. 11. Zestawienie wyników sprawdzających sąsiedztwo rozwiązania $d(KP_x^2, h)$ [opracowanie własne].

W nowej iteracji rozwiązaniem początkowym jest $d(KP_x^2, h) = 73,72 [h]$. Ponownie tworzone jest sąsiedztwo dla tego rozwiązania oraz wybierany jest sąsiad z najmniejszą wartością funkcji celu. W Tab. 12 zamieszczono 7 kolejnych iteracji oraz wyniki uzyskane w kolejnych pętłach optymalizacyjnych.

Lp	Rozwiązanie pocz.	Sąsiad	Koszt	Stan	i	l
1	81,72	73,72	1139,19	Początkowy	1	1
2	73,72	72,72	1071,44	Poprawa	2	1
3	72,72	71,72	1059,08	Poprawa	3	1
4	71,72	71,71	1059,02	Poprawa	4	1
5	71,71	71,7	1058,96	Poprawa	5	1
6	71,7	71,69	1058,9	Poprawa	6	1
7	71,69	71,7	1058,96	Pogorszenie	7	2
8	71,7	71,71	1059,02	Pogorszenie	8	3
9	71,71	71,72	1059,08	Pogorszenie	9	4
10	71,72	71,73	1059,13	Pogorszenie	10	5

Tab. 12. Kolejne iteracje w pętli optymalizacyjnej algorytmu DDAJSETS [opracowanie własne].

Proces optymalizacji rozwiązania przebiega do momentu osiągnięcia warunku wyjścia, tj. $l = Tabu_{Loop} + 1 = 204$. Warto zwrócić uwagę, że algorytm DDAJSETS doszedł do rozwiązania optymalnego w szóstej iteracji (wiersz zaznaczony kolorem żółtym w Tab. 11). Po osiągnięciu minimum lokalnego, które w tym przypadku jest również minimum globalnym, algorytm rozpoczął wyjście z tego minimum po ścieżce schodzenia do minimum. Ze względu na fakt, iż każdy wykonany ruch trafia na listę tabu, nie ma niebezpieczeństwa zapętlenia poszukiwań wokół wyznaczonego minimum.

Zakończenie działania algorytmu oznacza zwrócenie rozwiązania optymalnego. W przykładzie jest to $d_0(KP_x^2, h) = 71,69 [h]$, dla którego koszt wynosi $K_0(KP_x^2, h) = 1058,9$. Dla tak przyjętego rozwiązania $d_0(KP_x^2, h)$, wspólny termin zakończenia zbioru zleceń wynosi $\xi_0(KP_x^2, h) = 120 [h]$. Okno planistyczne zostaje ustanowione. Zakładając w uproszczeniu, że wspólny termin rozpoczęcia zbioru zleceń wypada w czasie $r(KP_x^2, h) = 0 [h]$, to długość okna planistycznego wynosi $l_h(KP_x^2) = 120 [h]$. Jest to nieprzekraczalny termin realizacji wszystkich 10 zleceń produkcyjnych.

Teoretyczny opis metody planowania przepływu produkcji w warunkach zmiennego zapotrzebowania, przedstawiony w rozdziale 4, posłużył do budowy systemu komputerowego. System komputerowy został zbudowany w celu weryfikacji opracowanej metody planowania przepływu produkcji pod względem postawionej tezy pracy. Opis systemu komputerowego do planowania przepływu produkcji w warunkach zmiennego zapotrzebowania znajduje się w kolejnym rozdziale.

5. BUDOWA SYSTEMU KOMPUTEROWEGO DO PLANOWANIA PRZEŁYWU PRODUKCJI W WARUNKACH ZMIENNEGO ZAPOTRZEBOWANIA

Rozdział poświęcony jest opisowi oprogramowania służącego do planowania przepływu produkcji według przedstawionej w rozdziale 4 koncepcji. Dzięki funkcjonalności tego oprogramowania możliwe jest porównanie otrzymanych wyników w zakresie harmonogramowania produkcji biorąc pod uwagę kryterium terminu zakończenia zbioru zleceń C_{max} . Opracowane oprogramowanie służy więc do sprawdzenia czy wypracowana teoria planowania przepływu produkcji z wykorzystaniem wirtualnych komórek przedmiotowych daje rozwiązania potwierdzające postawioną tezę pracy. Jest to zintegrowany system komputerowy umożliwiający planowanie przepływu produkcji oraz harmonogramowanie zadań w systemach job-shop.

Opis zintegrowanego systemu komputerowego służącego do planowania przepływu produkcji i harmonogramowania podzielono na trzy części. W pierwszej części przyjęto założenia funkcjonalne oprogramowania. W drugiej części przedstawiono logiczny model bazy danych systemu komputerowego. W części trzeciej zaprezentowano interfejs graficzny systemu komputerowego służący do komunikacji z użytkownikiem.

5.1. Założenia do budowy systemu komputerowego

Problem planowania przepływu produkcji i harmonogramowania job-shop jest problemem złożonym z punktu widzenia funkcjonalności i przetwarzania danych w systemie komputerowym. W procesach planowania przepływu produkcji i harmonogramowania zachodzi potrzeba opracowania systemu informatycznego, który jest w stanie zarządzać dużymi zbiorami danych w krótkim czasie. Szczególnie istotne jest, aby system komputerowy mógł być integrowany z systemami zarządzania zasobami przedsiębiorstwa takimi jak ERP. W związku z tym najodpowiedniejszym jest przyjęcie architektury klient-serwer. Tego typu systemy komputerowe to obecnie standard na rynku informatycznym w zakresie stosowania systemów ERP czy APS. Sercem systemu planowania przepływu produkcji i harmonogramowania staje się relacyjna baza danych, natomiast do komunikacji z użytkownikiem wykorzystuje się aplikację kliencką opracowaną w wybranym języku programowania.

Na rynku istnieje duża różnorodność narzędzi programistycznych zarówno w zakresie stosowania relacyjnych baz danych jak i niezależnych programów użytkownika. Spośród nich wyróżnia się jeden z najprężniej rozwijanych systemów zarządzania bazami danych SQL Server oraz środowisko programistyczne Visual Studio. Warty uwagi jest fakt, że narzędzia zaimplementowane w środowisko Visual Studio (głównie chodzi o platformę .NET) umożliwiają bezpośredni dostęp do bazy danych SQL Server.

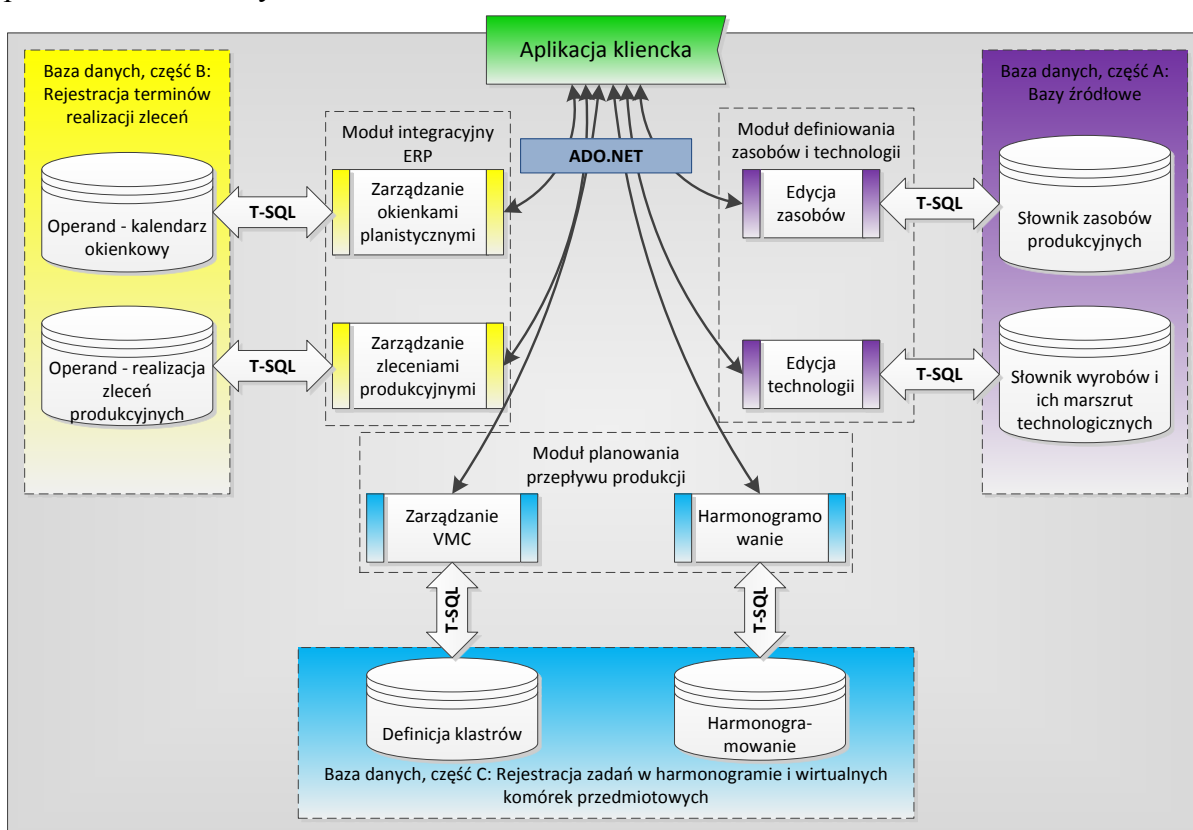
Ze względu na to, iż praca ma charakter naukowy z zakresu zarządzania produkcją, nie jest najistotniejszym wybór wysoko wydajnego narzędzia programistycznego. Dlatego postanowiono skorzystać z możliwości łatwej budowy systemów komputerowych w środowiskach Visual Studio i SQL Server. Do budowy bazy danych posłużono się wersją SQL Server 2008 R2, natomiast aplikację kliencką zbudowano w środowisku Visual Studio 2010 Ultimate, ko-

rzyszając z języka programowania Visual Basic. Do komunikacji pomiędzy bazą danych SQL Server 2008 R2 a aplikacją kliencką wykorzystano kontrolki ADO.NET.

5.1.1. Architektura systemu komputerowego

Podstawowym założeniem do budowy systemu komputerowego jest umożliwienie weryfikacji nowej metody planowania przepływu produkcji w warunkach zmiennego zapotrzebowania. System komputerowy umożliwia przetwarzanie i gromadzenie informacji w celu grupowania zleceń tak jak przedstawiono w podrozdziale 4.1. Została również zaimplementowana metoda budowy harmonogramu przedstawiona w podrozdziale 4.2. Należy podkreślić, że system komputerowy umożliwia tworzenie harmonogramu według klasycznych metod szeregowania zadań (zaimplementowano metody szeregowania zadań z wykorzystaniem reguł priorytetu FIFO, LIFO, SPT, LPT, MWR, LWR oraz szeregowanie zadań algorytmem genetycznym), a także według metody szeregowania zadań i wirtualnych komórek przedmiotowych zaprezentowanej w rozdziale 4.2. System komputerowy umożliwia również dobór długości okienek planistycznych według metody zaprezentowanej w rozdziale 4.3.

Biorąc pod uwagę postawione wymagania funkcjonalne dotyczące planowania przepływu produkcji z zastosowaniem wirtualnych komórek przedmiotowych, zbudowano architekturę systemu komputerowego. Ogólny schemat architektury systemu komputerowego przedstawiono na Rys. 43.



Rys. 43. Ogólny schemat architektury systemu komputerowego do planowania przepływu produkcji w warunkach zmiennego zapotrzebowania [opracowanie własne].

Na schemacie z Rys. 43 wyraźnie wyróżniono trzy części bazy danych. Pierwsza część, oznaczona jako A (kolor fioletowy), zawiera słowniki. W części A bazy danych znajdują się dwa słowniki. W pierwszym słowniku definiuje się zasoby produkcyjne. Na potrzeby pracy,

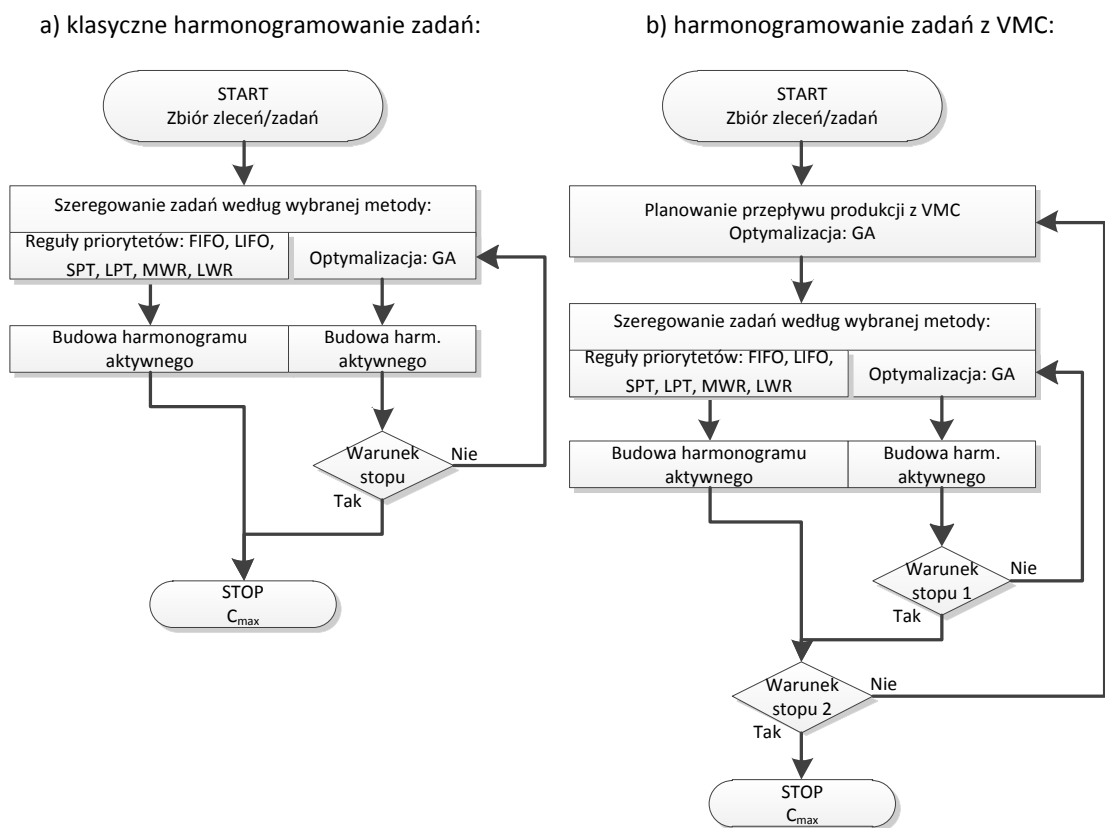
definicja zasobów ogranicza się do maszyn m oraz ich konfiguracji k . W słowniku zasobów produkcyjnych zapisywane są czasy trwania przebrojeń dla różnych konfiguracji maszyn $s_{mk-l,k}$. W drugim słowniku definiowane są wyroby finalne w , w tym wyroby złożone a oraz części oryginalne i wykonywane w komórce produkcyjnej II stopnia KP_x^2 . Każdemu wyrobowi złożonemu a oraz części oryginalnej i przypisuje się zbiór alternatywnych marszrut technologicznych v . W każdej alternatywnej marszrucie technologicznej wyróżnia się zbiór operacji j , a dla każdej operacji j definiuje się jej czas jednostkowy t_{ievjmk} .

System komputerowy jest opracowany w celu planowania przepływu produkcji w pojedynczej komórce produkcyjnej II stopnia KP_x^2 . Oznacza to, że w przypadku występowania systemów wytwórczych składających się z kilku komórek produkcyjnych II stopnia $KP_x^2 = KP_1^2, \dots, KP_x^2$, planowanie produkcji na szczeblu międzykomórkowym jest zadaniem systemu ERP, integrowanego z opracowywanym systemem komputerowym. W związku z tym wyróżniono drugą część bazy danych, oznaczoną jako B (kolor żółty). W części B bazy danych znajdują się dwa operandy jako zbiory obiektów podlegających działaniom aktualizacyjnym. Pierwszy operand służy do gromadzenia informacji przekazanych z modułu integracyjnego ERP, który z kolei wykonuje operacje na oknach planistycznych h w kalendarzu okienkowym. Operand kalendarza okienkowego podlega działaniu procedury szacującej długość okienek planistycznych $l_j(KP_x^2)$. Procedura posiada zaimplementowaną metodę DDAJSETS, opisaną w rozdziale 4.3.3.2. Drugi operand służy do gromadzenia informacji przekazanych bezpośrednio z systemu ERP. Są to informacje dotyczące realizacji zleceń produkcyjnych. Do operandu realizacji zleceń produkcyjnych trafiają zlecenia etapowe $z(e) = z(KP_x^2, h)$. W trakcie realizacji procesu planowania przepływu produkcji zgodnie z algorytmem ProduFlow PM, parametry zleceń etapowych $z(e)$ (np. termin zakończenia zlecenia, koszt realizacji zlecenia) są aktualizowane za pomocą kilku procedur składowanych (zgodnie z teorią baz danych, procedura składowana jest obiektem elementarnym bazy danych).

Trzecia część bazy, oznaczona jako C (kolor niebieski), służy do gromadzenia informacji o utworzonym planie przepływu produkcji oraz harmonogramowaniu. Składa się z dwóch części bezpośrednio związanych z modułem planowania przepływu produkcji. Jednym z zadań modułu planowania przepływu produkcji jest tworzenie klastrów (grup zleceń produkcyjnych) na podstawie danych o zleceniach produkcyjnych przechowywanych w części B bazy danych. Proces klastrowania jest zaimplementowany w procedury składowane zgodnie z algorytmami VAHC oraz LCF. Pierwsza z części C bazy danych służy do gromadzenia informacji o utworzonych klastrach oraz hierarchii występującej między nimi. Druga z części C bazy danych przechowuje informacje o zbiorze zadań ułożonych w harmonogramie według jednej z dostępnych metod harmonogramowania. Moduł planowania przepływu produkcji umożliwia harmonogramowanie na dwa sposoby, przedstawione na Rys. 44.

Pierwszy sposób harmonogramowania polega na wywołaniu procedur odpowiedzialnych za tworzenie harmonogramu zadań w sposób klasyczny. Jest to proces dwuetapowy (Rys. 44a). W pierwszym etapie dokonuje się szeregowania zadań za pomocą dostępnej metody szeregowania. Uszeregowane zadania trafiają na listę zadań oczekujących na włączenie do harmonogramu. W oparciu o przygotowaną listę zadań budowany jest w drugim etapie harmonogram aktywny za pomocą procedury składowanej.

Drugi sposób harmonogramowania polega na wywołaniu procedur odpowiedzialnych za tworzenie harmonogramu zadań oraz wirtualnych komórek przedmiotowych. Jest to proces trzyetapowy (Rys. 44b). W pierwszym etapie wywoływana jest procedura odpowiedzialna za planowanie przepływu produkcji, zgodnie z założeniami przedstawionymi w rozdziale 4.2.1. W wyniku działania procedury z pierwszego etapu otrzymuje się chromosom klastrów, którego zapis jest zgodny z zapisem z Rys. 23. Chromosom klastrów jest przekazywany do drugiego etapu. W drugim etapie wywołuje się procedurę odpowiedzialną za dekodowanie chromosomu zadań oraz szeregowanie zadań i wirtualnych komórek przedmiotowych (Rys. 30). Szeregowanie zadań i wirtualnych komórek przedmiotowych odbywa się poprzez zaszyfrowanie obiektów na liście obiektów oczekujących na włączenie do harmonogramu. Tym razem, do trzeciego etapu przekazuje się zaszyfrowaną listę obiektów oraz klucz szyfrowania. Natomiast w trzecim etapie, procedura układająca harmonogram aktywny z wirtualnymi komórkami przedmiotowymi najpierw wykonuje proces deszyfrowania obiektów z przekazanej listy według klucza szyfrowania.



Rys. 44. Sposoby harmonogramowania produkcji zaimplementowane w systemie komputerowym [opracowanie własne].

Odczyt, wstawianie, aktualizacja oraz usuwanie danych z bazy danych odbywa się za pośrednictwem procedur składowanych. Ze względu na korzystanie z systemu zarządzania bazą danych SQL Server, do wykonywania wyżej wymienionych czynności służy predefiniowany dialekt strukturalnego języka zapytań SQL (ang. *Structured Query Language*). Nosi on nazwę *Transact-SQL* (w skrócie T-SQL). T-SQL został zastosowany we wszystkich procedurach składowanych systemu komputerowego.

5.1.2. Funkcjonalność systemu komputerowego

System komputerowy planowania przepływu produkcji został opracowany w taki sposób, aby można było prawidłowo przeprowadzić proces planowania i harmonogramowania w kolejnych fazach algorytmu ProduFlow PM, zamieszczonym na Rys. 12.

Etap normalnej pracy z systemem komputerowym zawsze poprzedza etap wdrożenia. Wymaga się wcześniejszego przygotowania (uzupełnienia) bazy danych informacjami zarówno o dostępnych zasobach produkcyjnych (maszyny oraz ich możliwe konfiguracje) jak i o wyrobach i ich alternatywnych marszrutach technologicznych. Informacje te są gromadzone w części A bazy danych. Dzięki uzupełnieniu danych zasobowych system komputerowy automatycznie przelicza czas przebrojeń $s_{mk-1,k}$. Działa to w ten sposób, że dla określonego typu technologicznego maszyny m definiuje się wiele możliwych konfiguracji k . Dla każdej pary konfiguracji $k-1$ oraz k definiuje się czas przebrojenia. Jeżeli w wyniku układania harmonogramu znajdzie potrzeba realizacji dwóch zadań na maszynie m , a do realizacji zadań wymaga się różnych konfiguracji maszyny ($k-1$ dla zadania pierwszego oraz k dla zadania drugiego), wówczas zachodzi potrzeba przebrojenia maszyny m . System komputerowy wyselekcjonuje odpowiednią wartość zaistniałego czasu przebrojenia $s_{mk-1,k}$ i uwzględni ją podczas układania harmonogramu.

Z kolei wymóg wcześniejszego uzupełnienia danych o wyrobach oraz ich alternatywnych marszrutach technologicznych stwarza możliwości kontroli kolejnościowej zadań wstawianych do harmonogramu. System komputerowy sprawdza, czy tworzone zlecenia są wystawiane dla wyrobów złożonych, czy też dla wyrobów prostych. Jeżeli zostanie utworzone zlecenie na wyrób złożony $z(w)$, wówczas system komputerowy automatycznie wystawia zlecenia na elementy podrzędne w stosunku do tego wyrobu $z(w) = \{z(KP_x^2, h)_{a=1, \dots, n}, z(KP_x^2, h)_{i=1, \dots, n}\}$. W takim przypadku istnieje kontrola kolejnościowa wykonywania poszczególnych zleceń składających się na pierwotne zlecenie $z(w)$. Kontrola kolejnościowa występuje na etapie szeregowania zadań. Polega ona na tym, aby w harmonogramie najpierw były układane zadania należące do zleceń na elementy podrzędne wyrobu złożonego.

Ponadto istnieje możliwość wyboru jednej marszruty spośród zbioru alternatywnych marszrut technologicznych v dla każdego elementu wchodzącego w skład wyrobu złożonego. Ze względu na przyjęte założenie w pracy, system komputerowy nie ma zaimplementowanej procedury podejmowania decyzji o wyborze marszruty technologicznej. Użytkownik podejmuje taką decyzję w momencie definiowania nowych zleceń produkcyjnych, przechodzących przez komórkę produkcyjną II stopnia KP_x^2 .

Po wprowadzeniu do systemu komputerowego planowania przepływu produkcji powyższych informacji przystępuje się do wstępnego definiowania okna planistycznego oraz zleceń, które mają być w nim wykonywane (część B bazy danych). Wstępnie zdefiniowane okno planistyczne nie ma określonej długości w czasie. Długość okna planistycznego jest określana po podjęciu decyzji co do wyboru wariantu przepływu produkcji.

Po zdefiniowaniu zleceń produkcyjnych, które mają być wykonane w oknie planistycznym oraz po wybraniu marszrut technologicznych dla każdego zlecenia produkcyjnego przystępuje się do realizacji algorytmu ProduFlow PM.

W pierwszej kolejności przeprowadza się proces grupowania zleceń produkcyjnych. Proces grupowania zleceń jest wywoływany przez użytkownika systemu komputerowego. Wszystkie zlecenia zdefiniowane wcześniej przez użytkownika podlegają procesowi grupowania. Wywoływane są procedury odpowiedzialne za budowę dendrogramu za pomocą hierarchicznego aglomeracyjnego algorytmu klastrowania. W wyniku działania procedur otrzymuje się zbiór danych o zgrupowanych zleceniach produkcyjnych. W bazie danych przechowywane są informacje zarówno o singletonach reprezentujących poszczególne zlecenia jak i o klastrach z wydzielonymi łańcuchami zadań.

Zbiór zapisanych informacji o singletonach i klastrach zwykłych, zapisanych w bazie danych, umożliwia prawidłowe przeprowadzenie procesu generowania przepływu produkcji z zastosowaniem wirtualnych komórek przedmiotowych. W tym celu system komputerowy automatycznie wywołuje kolejne procedury składowane. Są to trzy następujące procedury składowane:

- procedura służąca do ustalenia zbioru wirtualnych komórek przedmiotowych podlegających harmonogramowaniu,
- procedura szeregująca zadania oraz wirtualne komórki przedmiotowe na liście obiektów oczekujących na włączenie do harmonogramu,
- procedura budująca harmonogram aktywny *job-shop* z wirtualnymi komórkami przedmiotowymi.

Istnieje również możliwość budowy klasycznego harmonogramu zadań, z pominięciem procedury generowania przepływu produkcji. O wyborze metody szeregowania obiektów na liście obiektów oczekujących na włączenie do harmonogramu decyduje użytkownik systemu komputerowego. Dla każdej metody szeregowania obiektów do celu tworzenia harmonogramu zarówno w sposób klasyczny *job-shop* jak i z wirtualnymi komórkami przedmiotowymi, została opracowana osobna procedura. To, w jaki sposób ma być zbudowany harmonogram zależy od użytkownika, którego zadaniem jest wywołanie jednego z dwóch dostępnych scenariuszy: budowa klasycznego harmonogramu general *job-shop* lub harmonogramu z wirtualnymi komórkami przedmiotowymi. Użytkownik ma do dyspozycji procedury szeregujące obiekty według reguł priorytetów FIFO, LIFO, SPT, LPT, MWR oraz LWR. Zostały również zaimplementowane metody optymalizacyjne szeregowania zadań i układania harmonogramu z zastosowaniem algorytmu genetycznego do optymalizacji terminu zakończenia zbioru zleceń produkcyjnych. Jako przykład, w dodatku A zamieszczono szczegółowy opis funkcjonowania procedur tworzących harmonogram zarówno w sposób klasyczny *job-shop* jak i z wirtualnymi komórkami przedmiotowymi, za pomocą reguły priorytetu SPT dla określonego zbioru zleceń produkcyjnych.

Po wykonaniu procedur związanych z planowaniem przepływu produkcji i harmonogramowaniem, w bazie danych zostają zapisane wszystkie zadania i wirtualne komórki przedmiotowe ze zdefiniowanymi terminami rozpoczęcia oraz zakończenia. Uruchamiana jest

również procedura służąca do aktualizacji informacji o planowanej realizacji zleceń produkcyjnych. Każde zlecenie zostaje zaktualizowane wartościami terminu zakończenia. Pozwala to na uruchomienie procedury szacującej długość okienka planistycznego.

Dostępna funkcjonalność systemu komputerowego stwarza możliwości porównania różnych wariantów przepływu produkcji. Użytkownik jest w stanie wybrać spośród dostępnych wariantów ten najlepszy, posiadający, z punktu widzenia tezy pracy, minimalną wartość C_{max} . Ponadto, zgodnie z postawionym celem pracy, system komputerowy umożliwia na szacowanie długości okienek planistycznych, co znacząco wpływa na możliwości integracji tego systemu z dowolnym systemem klasy ERP.

W kolejnych rozdziałach przedstawiono schemat budowy bazy danych po stronie serwera oraz zaprezentowano graficzny interfejs aplikacji klienckiej, służącej do komunikacji z bazą danych oraz umożliwiający wywoływanie procedur związanych z realizacją algorytmu ProduFlow PM, a znajdujących się po stronie serwera bazodanowego.

5.2. Schemat budowy bazy danych

Baza danych systemu komputerowego służącego do planowania przepływu produkcji w warunkach zmiennego zapotrzebowania została opracowana w celu gromadzenia oraz przetwarzania proceduralnego informacji związanych z realizacją algorytmu ProduFlow PM. Składa się z kilku rodzajów obiektów bazodanowych ułatwiających przetwarzanie danych po stronie serwera oraz wysyłanie pakietów danych do aplikacji klienckiej. Do obiektów tych należą: tabele, widoki, procedury składowane oraz zmienne tabelaryczne.

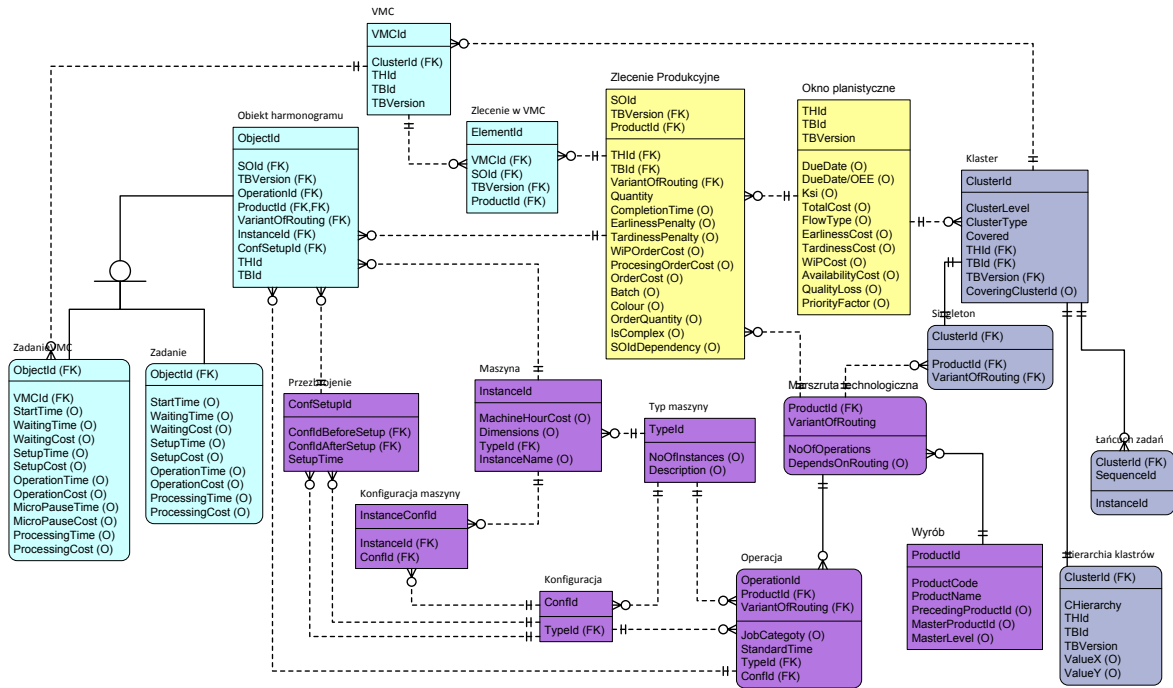
Model bazy danych został zaprojektowany w sposób umożliwiający odczyt danych za pomocą narzędzi ADO.NET oraz ich prezentację graficzną za pomocą aplikacji klienckiej. Jednocześnie zostały opracowane procesy biznesowe (w formie procedur składowanych) związane z realizacją algorytmu ProduFlow PM po stronie serwera bazodanowego.

Na Rys. 45 przedstawiono logiczny model danych dla systemu komputerowego planowania przepływu produkcji. Model danych został przygotowany w standardzie IDEF1X wspomaganym notacją ptasiej nogi.

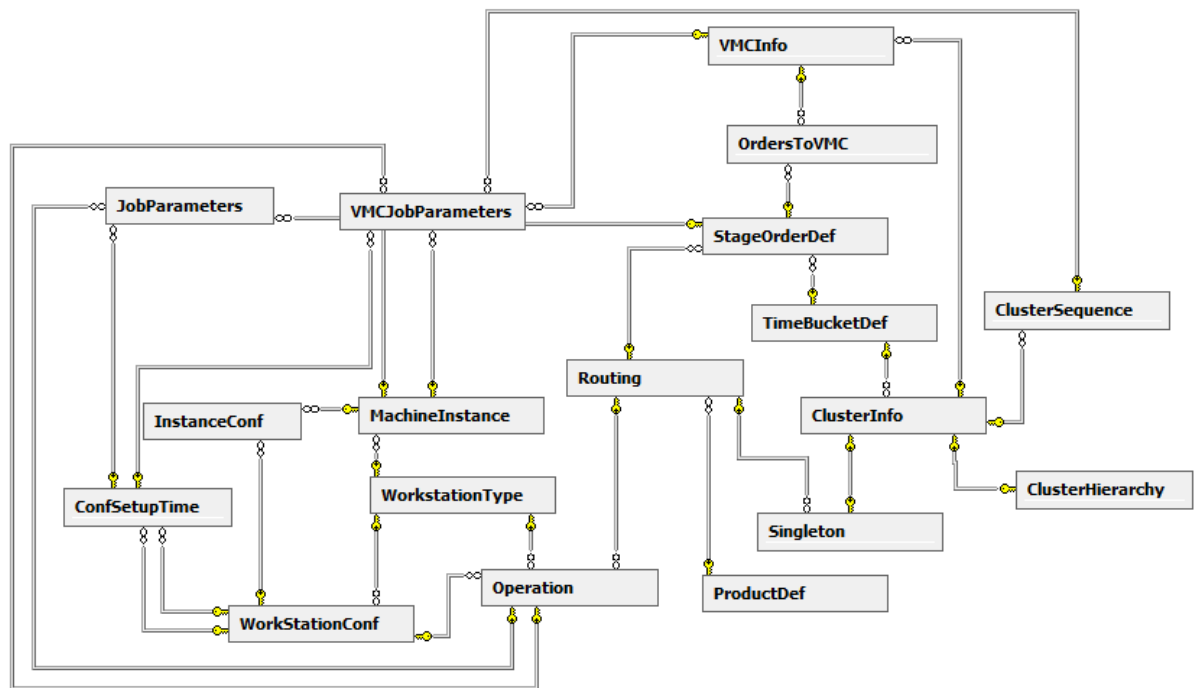
W modelu logicznym bazy danych zostały wydzielone cztery grupy tabel (Rys. 45):

- grupa tabel przechowująca dane źródłowe o zasobach produkcyjnych i marszrutach technologicznych (kolor fioletowy),
- grupa tabel przechowująca dane o realizacji zleceń produkcyjnych (kolor żółty),
- grupa tabel przechowująca dane o klastrach, tj. zgrupowanych zleceniach (kolor granatowy),
- grupa tabel przechowująca dane o obiektach harmonogramu (kolor niebieski).

Model logiczny bazy danych wskazuje kolejność uzupełniania danych w bazie danych systemu komputerowego. Dobrze opracowany model logiczny pozwala na łatwą implementację schematu logicznego w środowisko zarządzania bazą danych. W przypadku bazy danych SQL Server 2008 R2, przechodząc z modelu logicznego na model fizyczny wykorzystano wbudowane narzędzie ERD do tworzenia modelu fizycznego bazy danych. Model fizyczny oraz jego implementację w SQL Server 2008 R2 w postaci uproszczonej przedstawiono na Rys. 46.



Rys. 45. Schemat logiczny bazy danych systemu komputerowego planowania przepływu produkcji z zastosowaniem standardu IDEF1X wspomaganego notacją ptasiej nogi [opracowanie własne].



Rys. 46. Uproszczony schemat fizyczny bazy danych systemu komputerowego planowania przepływu produkcji z zastosowaniem standardu ERD zaimplementowanego w bazie SQL SERVER 2008 R2 [opracowanie własne].

Wdrożenie systemu komputerowego odbywa się według następującego planu. Najpierw zostają uzupełnione tabele bez oznaczenia ptasiej nogi w encjach modelu logicznego (są to tabele nie zawierające kluczy obcych w modelu fizycznym). Jak wynika z modelu logicznego bazy danych (Rys. 45), najpierw zostaną uzupełnione tabele *ProductDef* (encja *Wyrób*) i *WorkstationType* (encja *Typ maszyny*). Istnieje również możliwość uzupełnienia tabeli *TimeBucketDef* (encja *Okno planistyczne*), jednakże wykonuje się to później w trakcie realizacji algorytmu *ProduFlow PM*. Po uzupełnieniu tabel *ProductDef* i *WorkstationType* przystępuje

się do wypełniania tabel *Routing* (encja *Marszruta technologiczna*), *MachineInstance* (encja *Maszyna*) i *WorkstationConf* (encja *Konfiguracja*). Następnie uzupełnia się tabelę *Operation* (encja *Operacja*), tabelę pośrednią pomiędzy tabelami *MachineInstance* oraz *WorkstationConf* o nazwie *InstanceConf* (encja *Konfiguracja maszyny*) oraz tabelę *ConfSetupTime* (encja *Przebrojenie*). Jest to zgodne z założeniami przyjętymi w pracy, tj. informacje o asortymencie wyrobów, marszrutach technologicznych wyrobów oraz parku maszynowym są znane. Wymaga się jedynie wprowadzenia danych o wymienionych zasobach do systemu komputerowego.

Po wprowadzeniu wyżej wymienionych informacji do systemu komputerowego przystępuje się do realizacji algorytmu ProduFlow PM. W pierwszej kolejności tworzy się okna planistyczne w tabeli *TimeBucketDef* (encja *Okno planistyczne*). Tabela ta zawiera atrybuty umożliwiające wstawianie wartości *Null*. Jest to zrobione celowo, gdyż nie wszystkie informacje w tabeli *TimeBucketDef* są dostępne w momencie tworzenia okna planistycznego. Uwaga dotyczy atrybutów związanych z określeniem długości okna planistycznego. Algorytm ProduFlow PM powraca do uaktualnienia wartości tych atrybutów na samym końcu swojego działania (zgodnie z założeniem przyjętym podczas opracowania algorytmu ProduFlow PM, po zakończeniu realizacji fazy III algorytmu otrzymuje się szacowaną długość okna planistycznego).

W utworzonym oknie planistycznym definiuje się kolejno zlecenia produkcyjne w tabeli *StageOrderDef* (encja *Zlecenie produkcyjne*), które mają być w nim realizowane. Z założenia przyjmuje się, że zdefiniowanie zlecenia produkcyjnego jest równoznaczne z jego gotowością do realizacji od początku okna planistycznego. Założenie dotyczy wszystkich zleceń produkcyjnych definiowanych w tym samym oknie planistycznym. Nie istnieje ograniczenie co do liczby zleceń produkcyjnych, jakie mogą wystąpić w jednym oknie planistycznym. Określenie zbioru zleceń produkcyjnych dopełnia wymagane informacje wejściowe na starcie algorytmu ProduFlow PM.

Rozpoczęcie działania algorytmu ProduFlow PM jest związane z realizacją fazy I - grupowania zleceń produkcyjnych. Za przechowywanie informacji o zgrupowanych zleceniach produkcyjnych metodą klastrowania z wykorzystaniem hierarchicznego aglomeracyjnego algorytmu klastrowania jest odpowiedzialny zbiór tabel *ClusterInfo*, *Singleton*, *ClusterSequence* oraz *ClusterHierarchy* (są to encje odpowiednio *Klaster*, *Singleton*, *Łańcuch zadań* oraz *Hierarchia klastrów*, zaznaczone kolorem granatowym na Rys. 45). Sam proces grupowania, przedstawiony na Rys. 19 oraz Rys. 20 realizuje się za pomocą procedur składowanych po stronie serwera bazodanowego. Procedury składowane nazywają się odpowiednio *spDendrogram* oraz *spLCF*. Ze względu na wykorzystanie serwera bazodanowego SQL Server 2008 R2, wszystkie procedury składowane utworzone w celu realizacji algorytmu ProduFlow PM zostały opracowane w dialekcie języka SQL o nazwie Transact-SQL. Kod T-SQL znajduje się w dodatku C do pracy. W dodatku C zamieszczono kod przykładowej procedury składowanej służącej do budowy dendrogramu (o nazwie *spDendrogram*).

Wypełnienie tabel odpowiedzialnych za gromadzenie danych o klastrach (zgrupowanych zleceniach) umożliwia przejście do fazy II algorytmu ProduFlow PM. W drugiej fazie algorytmu ProduFlow PM tworzony jest plan przepływu oraz harmonogram w zależności od

ustawień wykonanych przez użytkownika systemu komputerowego. Utworzenie planu przepływu oraz harmonogramu skutkuje uzupełnieniem tabel *VMCInfo*, *OrdersToVMC*, *VMCJobParameters* oraz *JobParameters* (są to encje odpowiednio *VMC*, *Zlecenie w VMC* oraz *Obiekt harmonogramu*, który jest kategoryzowany albo, jako zadanie zwykłe albo, jako zadanie wykonywane w wirtualnej komórce przedmiotowej). Do realizacji planowania przepływu produkcji oraz harmonogramowania służy szereg procedur składanych z zaimplementowanymi algorytmami opisanymi w rozdziale 4.2. Są to kolejno procedury *spGAVMCSselection*, służące do utworzenia planu przepływu produkcji z wirtualnymi komórkami przedmiotowymi (z wbudowaną pętlą optymalizacyjną, w której wykorzystano algorytm genetyczny do uzyskania minimalnego parametru C_{max}), *spGAVMCScheduling*, służąca do szeregowania obiektów (zadań wykonywanych w wirtualnych komórkach przedmiotowych i zadań zwykłych wykonywanych poza wirtualnymi komórkami przedmiotowymi) w pętli optymalizacyjnej z wykorzystaniem ponownie algorytmu genetycznego (do minimalizacji ponownie parametru C_{max}), na końcu procedura *spCreateVMCSchedule* odpowiedzialna za tworzenie harmonogramu według uszeregowanych obiektów z listy obiektów oczekujących na włączenie do harmonogramu.

Utworzenie harmonogramu pozwala określić terminy zakończenia poszczególnych zleceń produkcyjnych. Aktualizacji terminów zakończenia poszczególnych zleceń produkcyjnych dokonuje się w tabeli *StageOrderDef* (atrybut *CompletionTime*). Do tego celu służy procedura składana o nazwie *spCalculateThreeOrderParams*, która oprócz wspomnianego atrybutu aktualizuje również wartości atrybutów *ProcessingOrderCost* (koszt przetwarzania zlecenia) oraz *WiPOrderCost* (koszt oczekiwania zlecenia w magazynie buforowym).

Aktualizacja trzech atrybutów w tabeli *StageOrderDef* pozwala przejść do III fazy algorytmu ProduFlow PM. W niej dokonuje się oszacowania wspólnego terminu zakończenia zbioru zleceń, na podstawie przyjętych przez użytkownika w tabeli *TimeBucketDef* wartości parametrów nieplanowanych zdarzeń (wartości nieplanowanych zdarzeń, tj. wskaźnika nieplanowanych przestoju jako *AvailabilityLoss* oraz wskaźnika zleceń priorytetowych jako *PriorityFactor*, są ustawiane przez użytkownika). Uruchamiana jest procedura składana *spFindOptimalDueDate*, która ma zaimplementowany algorytm DDAJSETS, przedstawiony na Rys. 42.

Opracowana baza danych służy do przechowywania oraz przetwarzania informacji związanych z realizacją algorytmu ProduFlow PM. Do celu komunikacji oraz graficznej prezentacji wyników zapisanych w bazie danych opracowano aplikacje kliencką. Opis aplikacji klienckiej znajduje się w kolejnym rozdziale.

5.3. Aplikacja kliencka

Aplikacja kliencka została opracowana w języku programowania Visual Basic 2010 .NET, w środowisku programistycznym Visual Studio 2010 Ultimate. Rolą aplikacji klienckiej jest prawidłowa komunikacja użytkownika z bazą danych SQL Server 2008 R2. Aplikacja kliencka ma ułatwić obsługę systemu komputerowego podczas realizacji kolejnych faz algorytmu ProduFlow PM.

Za komunikację aplikacji z bazą danych odpowiedzialne są mechanizmy ADO.NET, wbudowane w platformę .NET 4.5 środowiska Visual Studio 2010. Służą one m. in. do ustanawiania połączenia, transferu danych z serwera (głównie z widoków bazy danych) do aplikacji oraz w drugą stronę, wywoływania procedur składowanych po stronie serwera oraz gromadzenie danych z bazy po stronie aplikacji w celu ich dalszego prezentowania w formie graficznej. Są to obiekty *SqlConnection* do komunikacji z bazą danych, *SqlDataAdapter*, *DataSet* do transferu danych do aplikacji klienckiej, *SqlCommand* do wywoływania procedur składowanych po stronie serwera oraz *DataTable* i *DataRow* do gromadzenia danych po stronie aplikacji klienckiej.

W celu nawiązania połączenia z bazą danych wymagane jest podanie określonego łańcucha znaków, zawierającego nazwę serwera bazodanowego, nazwę bazy danych, login oraz hasło do loginu. W przypadku korzystania z bazy danych do systemu komputerowego planowania przepływu produkcji, jedynym uprawnionym użytkownikiem jest administrator systemu SQL Server 2008 R2. Podany łańcuch danych jest przechowywany w obiekcie *SqlConnection*. Jakikolwiek przepływ informacji z i do serwera bazodanowego odbywa się przy wykorzystaniu obiektu *SqlConnection*. Przykładowo, transfer danych z serwera bazodanowego z tabeli *TimeBucketDef* (wszystkie atrybuty okna planistycznego) do aplikacji klienckiej odbywa się z wykorzystaniem obiektów *SqlDataAdapter* oraz *DataSet* w sposób zaprezentowany na listingu poniżej:

```
'Deklaracja zmiennej do ustanowienia połączeń z bazą danych wewnątrz programu
Dim objConnection As New SqlConnection("Server=" & txtServer.Text & ";Database=" & txtDatabase.Text &
";User ID=" & txtUser.Text & ";Password=" & txtPassword.Text)

'Obiekty do nawiązania połączenia z tabelą TimeBucketDef
Dim objDataAdapter As SqlDataAdapter
Dim objDataSet As DataSet
Dim objDataTable As DataTable
Dim objDataRow As DataRow
Dim intCount As Integer
Dim objTimeBucketInfo As TimeBucketInfo

'Utworzenie pomostu między bazą danych a aplikacją oraz przypisanie instrukcji SQL
objDataAdapter = New SqlDataAdapter("SELECT * FROM TimeBucketDef WHERE THId = " & stringTHId & " AND
TBId = " & stringTBId & " AND TBVersion = " & stringTBVersion, objConnection)

'Inicjowanie nowego egzemplarza obiektu DataSet
objDataSet = New DataSet()

'Zapełnienie objDataSet danymi
objDataAdapter.Fill(objDataSet, "TimeBucketDef")

'Inicjowanie objDataTable
objDataTable = New DataTable("TimeBucketDef")

'Przekazanie danych z objDataSet do tabeli objDataTable
objDataTable = objDataSet.Tables("TimeBucketDef")

'Przeliczenie wierszy w tabeli objDataTable
intCount = objDataTable.Rows.Count
```

Podobnie, transfer danych oraz wywołanie procedury składowanej po stronie serwera odbywa się za pośrednictwem obiektu *SqlConnection*. Tym razem wykorzystuje się jednak obiekt *SqlCommand*. Przykładowo, wstawienie nowego zlecenia produkcyjnego za pomocą obiektu *SqlCommand* przedstawiono na listingu poniżej:

```

'Deklaracja i inicjowanie obiektu służącego do połączenia się z bazą danych
Dim objConnection As New SqlConnection("Server=" & txtServer.Text & ";Database=" & txtDatabase.Text &
";User ID=" & txtUser.Text & ";Password=" & txtPassword.Text)

'Deklaracja i inicjowanie parametrów wywołujących procedurę spInsertMainStageOrder wstawiającą nowe
zlecenie produkcyjne do tabeli StageOrderDef
Dim objCommand As SqlCommand = New SqlCommand()
'Ustawienie połączenia w obiekcie objCommand
objCommand.Connection = objConnection
'Ustawienie typu komendy w obiekcie objCommand na procedurę składowaną
objCommand.CommandType = CommandType.StoredProcedure

'Ustawienie daty wystawienia zlecenia: zlecenie jest zapisane w bazie jako ciąg znaków char(16)
Dim strDate As String = CType(Date.Today, String)
strDate = strDate.Substring(0, 4) & strDate.Substring(5, 2) & strDate.Substring(8, 2)

'Wyczyszczenie obiektu objCommand
objCommand.Parameters.Clear()

'Ustawienie nazwy procedury składowanej w obiekcie objCommand
objCommand.CommandText = "spInsertMainStageOrder"
'Inicjowanie obiektu objStageOrderInfo
objStageOrderInfo = New StageOrderInfo(strDate & ModifySProductIdToChar(CType(cbxProductId.Text, Integer)) & "01", CType(cbxTHId.Text, Integer), CType(cbxTBId.Text, Integer), CType(cbxTBVersion.Text, Integer), CType(cbxProductId.Text, Integer), CType(cbxAlternativeRouting.Text, Integer), CType(txtOrderQuantity.Text, Integer), strColor, False)

'Ustawienie wartości atrybutów wprowadzanych do procedury spInsertMainStageOrder
objCommand.Parameters.AddWithValue("@SOId", objStageOrderInfo.SOId)
objCommand.Parameters.AddWithValue("@THId", objStageOrderInfo.THId)
objCommand.Parameters.AddWithValue("@TBId", objStageOrderInfo.TBId)
objCommand.Parameters.AddWithValue("@TBVersion", CType(stringTBVersion, Integer))
objCommand.Parameters.AddWithValue("@SProductId", objStageOrderInfo.SProductId)
objCommand.Parameters.AddWithValue("@VariantOfRouting", objStageOrderInfo.VariantOfRouting)
objCommand.Parameters.AddWithValue("@Quantity", objStageOrderInfo.Quantity)
objCommand.Parameters.AddWithValue("@CompletionTime", objStageOrderInfo.CompletionTime)
objCommand.Parameters.AddWithValue("@EarlinessPenalty", objStageOrderInfo.EarlinessPenalty)
objCommand.Parameters.AddWithValue("@TardinessPenalty", objStageOrderInfo.TardinessPenalty)
objCommand.Parameters.AddWithValue("@WaitingCostPerOrder", objStageOrderInfo.WiPCost)
objCommand.Parameters.AddWithValue("@ProcessingCostPerOrder", objStageOrderInfo.ProcessingCost)
objCommand.Parameters.AddWithValue("@OrderZK", objStageOrderInfo.OrderZK)
objCommand.Parameters.AddWithValue("@AssignedToCluster", objStageOrderInfo.AssignedToCluster)
objCommand.Parameters.AddWithValue("@ClusterId", objStageOrderInfo.ClusterId)
objCommand.Parameters.AddWithValue("@ClusterLevel", objStageOrderInfo.ClusterLevel)
objCommand.Parameters.AddWithValue("@Batch", objStageOrderInfo.Batch)
objCommand.Parameters.AddWithValue("@StageColour", objStageOrderInfo.SOColor)
objCommand.Parameters.AddWithValue("@StageOrderQuantity", objStageOrderInfo.SOQuantity)
objCommand.Parameters.AddWithValue("@IsComplex", objStageOrderInfo.IsComplex)

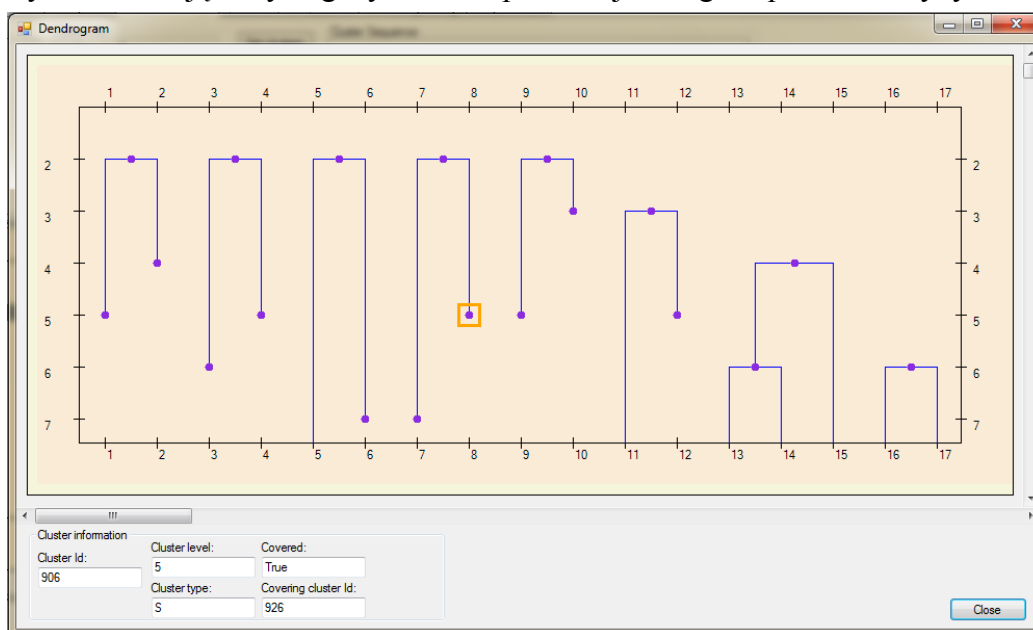
'Otwarcie połączenia z bazą danych
objConnection.Open()
'Wykonanie polecenia - wywołanie procedury składowanej spInsertMainStageOrder
Try
    objCommand.ExecuteNonQuery()
Catch ex As Exception
    MessageBox.Show("Data error. Please check your data")
Finally
    'Zamknięcie połączenia
    objConnection.Close()
End Try

```

Oprócz normalnej komunikacji z bazą danych, aplikacja kliencka spełnia również rolę graficznej prezentacji wyników zapisanych w bazie danych. Po pierwsze, jest ona wykorzystywana do prezentacji dendrogramu klastrów, zgodnie z założeniami i opisem przedstawionym w rozdziale 4.1.2.2. Przykładowy widok okna z narysowanym programowo dendrogramem zilustrowano na Rys. 47.

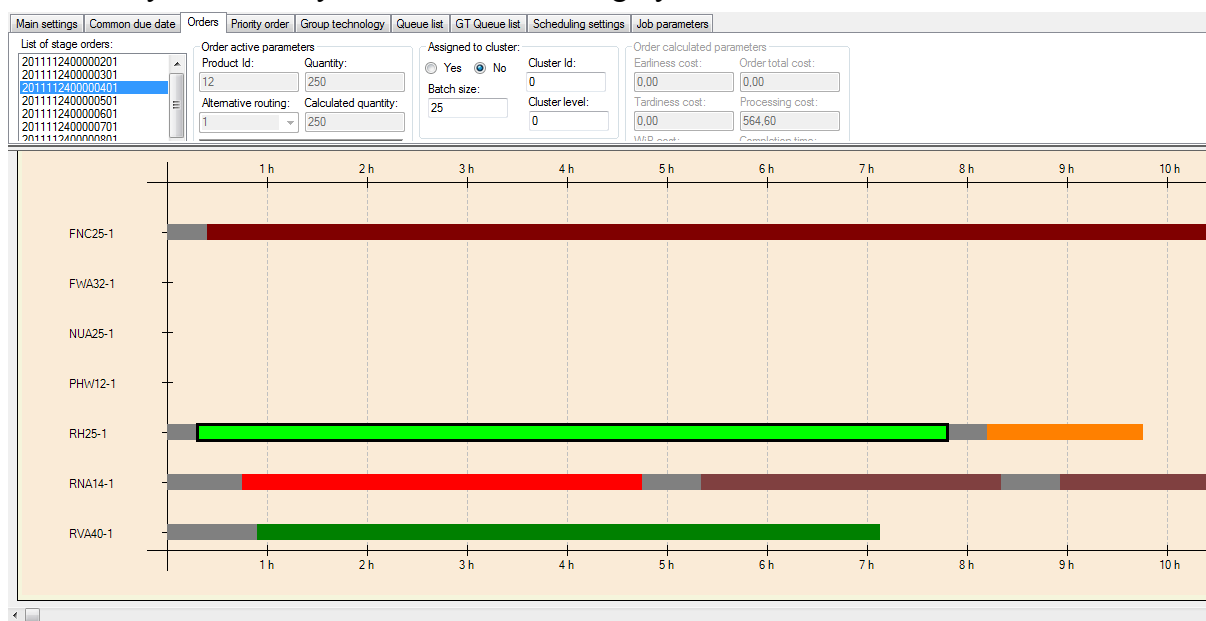
Największe korzyści daje możliwość podglądu utworzonego harmonogramu. Harmonogram jest prezentowany graficznie w oknie głównym aplikacji klienckiej. Jest on przedstawiony w postaci wykresu Gantta. Przykładowy widok okna z narysowanym harmonogramem zaprezentowano na Rys. 48.

Na wykresie Gantta, rysowanym programowo przez aplikację kliencką, poszczególnymi kolorami wyodrębniono kolejne zlecenia produkcyjne. Szare prostokątki pomiędzy dwoma kolorowymi informują o wymaganym czasie przezbrojenia egzemplarza maszyny.



Rys. 47. Widok okna z narysowanym programowo dendrogramem [opracowanie własne].

W systemie komputerowym planowania przepływu produkcji i harmonogramowania przyjęto skalę godzinową czasu. Dzięki temu nie ma konieczności przeskalowywania wartości czasowych w celu wykonania obliczeń w algorytmie ProduFlow PM.



Rys. 48. Widok okna z narysowanym programowo harmonogramem w postaci wykresu Gantta [opracowanie własne].

Opracowany system komputerowy posłużył do weryfikacji koncepcji teoretycznej planowania przepływu produkcji z wykorzystaniem wirtualnych komórek przedmiotowych. W wyniku przeprowadzenia szeregu testów na specjalnie przygotowanym przykładzie systemu wytwórczego, otrzymano porównanie, jak nowa metoda planowania przepływu produkcji wypada w świetle klasycznego sposobu planowania i harmonogramowania typu job-shop produkcji. Wyniki porównawcze zaprezentowano w kolejnym rozdziale pracy.

6. WERYFIKACJA EKSPERYMENTALNA METODY PLANOWANIA PRZEPŁYWU PRODUKCJI W WARUNKACH ZMIENNEGO ZAPOTRZEBOWANIA

Dla potrzeb testowania przedstawionej metody planowania przepływu produkcji w warunkach zmiennego zapotrzebowania przygotowano przykład systemu wytwórczego charakteryzujący się typem produkcji małoseryjnej. Sprawdzenie poprawności przedstawionej metody oraz skuteczności działania algorytmu ProduFlow PM na tle klasycznego sposobu harmonogramowania produkcji zrealizowano w dwóch krokach. W pierwszym kroku opracowano przykład harmonogramowania z wykorzystaniem reguły priorytetu SPT, według dwóch scenariuszy (dodatek A):

- klasyczne harmonogramowanie general job-shop,
- harmonogramowanie z wirtualnymi komórkami przedmiotowymi według działania algorytmu ProduFlow PM.

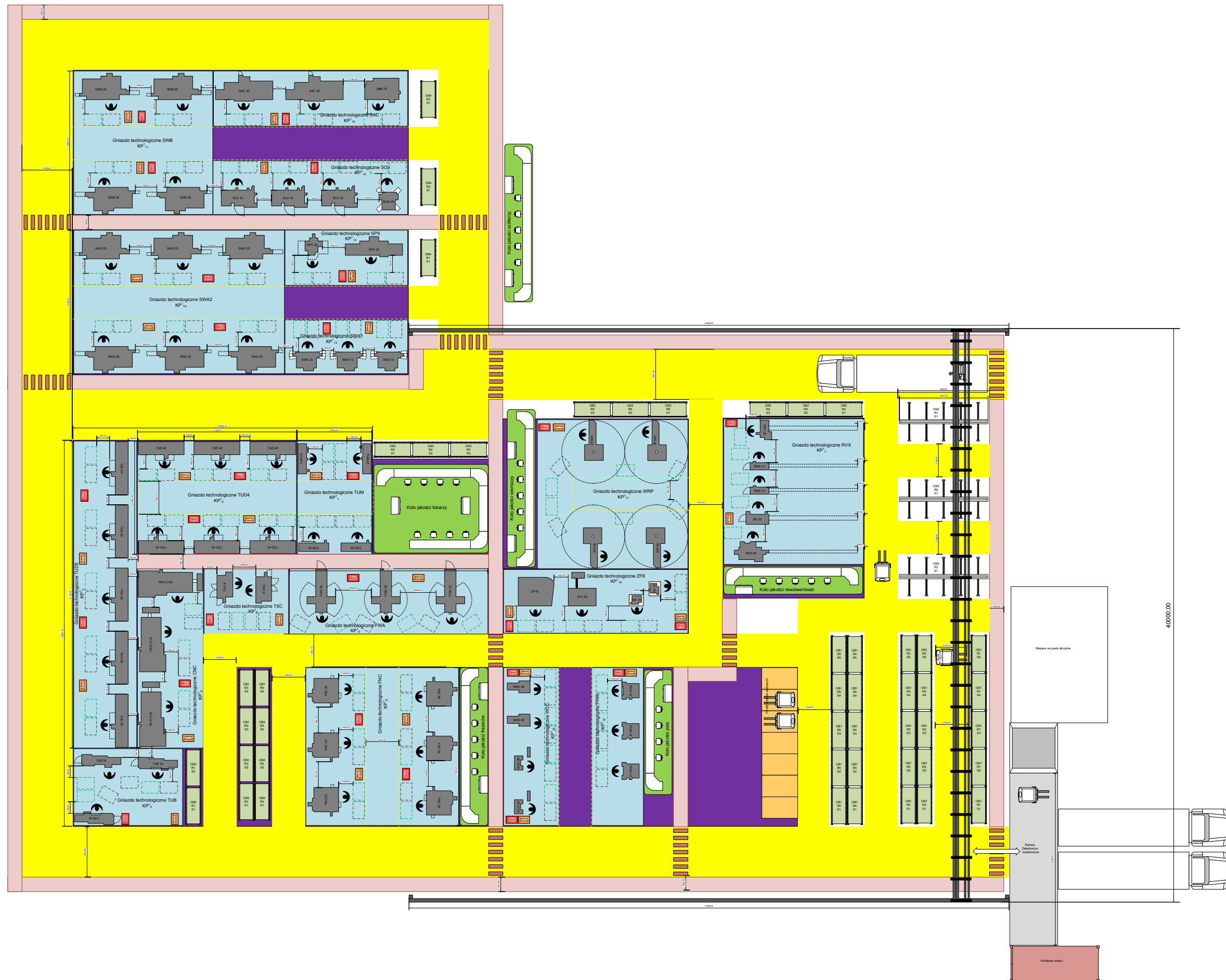
Przykład posłużył do weryfikacji poprawności metody w zakresie tworzenia harmonogramu w sposób klasyczny oraz z wykorzystaniem wirtualnych komórek przedmiotowych.

W drugim kroku dokonano porównania wyników (porównaniu podlegają parametry: termin zakończenia wszystkich zleceń produkcyjnych C_{max} oraz długość okna planistycznego $l_h(KP_x^2)$) obliczany na podstawie ustalonej wartości zmiennej decyzyjnej $d(KP_x^2, h)$) otrzymanych ponownie dla dwóch scenariuszy: klasycznego harmonogramowania general job-shop oraz harmonogramowania z wirtualnymi komórkami przedmiotowymi.

Do przeprowadzenia planowania przepływu produkcji w drugim kroku testowym opracowano model eksperymentalny systemu wytwórczego (w oparciu o źródła literaturowe [19][67][97]) ujmujący dane o wyrobach i marszrutach technologicznych. W modelu systemu wytwórczego wyodrębniono komórkę produkcyjną II stopnia, w której realizuje się produkcję małoseryjną. Plan rozmieszczenia maszyn w komórce produkcyjnej II stopnia przedstawiono na Rys. 49. W komórce produkcyjnej II stopnia wydzielono gniazda produkcyjne (komórki produkcyjne I stopnia) o specjalizacji technologicznej. Układ organizacyjny komórki produkcyjnej II stopnia został przygotowany w taki sposób aby odzwierciedlić przepływ produkcji w systemie typu job-shop.

Zgodnie z przyjętym w pracy założeniem park maszynowy jest ustalony a rozmieszczenie maszyn w komórce produkcyjnej II stopnia nie ulega zmianie. Ponadto przygotowano następujące informacje dotyczące realizacji produkcji w komórce produkcyjnej II stopnia:

- zestawienie maszyn, realizujących zadania w ramach komórki produkcyjnej II stopnia (dodatek B, punkt 1),
- zestawienie marszrut technologicznych wyrobów przepływających przez komórkę produkcyjną II stopnia (dodatek B, punkt 2).



Rys. 49. Plan rozmieszczenia maszyn w eksperymentalnej komórce produkcyjnej II stopnia [opracowanie własne].

Testowanie skuteczności działania algorytmu ProduFlow PM na tle klasycznego sposobu harmonogramowania produkcji zrealizowano według następującego planu:

- Dane charakteryzujące system wytwórczy zostały zapisane w bazie danych systemu komputerowego. Zostały wprowadzone informacje o parku maszynowym (29 różnych typów technologicznych maszyn, w ramach każdego typu technologicznego znajduje się kilka instancji maszyn co daje w sumie 73 maszyny w parku maszynowym).
- Zostały również zdefiniowane wyroby. W celu uproszczenia procesu testowania w drugim kroku, ograniczono asortyment produkowanych wyrobów wyłącznie do części oryginalnych (wyroby proste). Jest ich w sumie zapisanych 101 wyrobów prostych, przy czym do testowania wykorzystano kolejne wyroby z identyfikatorami od 16 do 30 (ze względu na najdłuższe marszruty technologiczne – nawet do 20 operacji w pojedynczej marszrucie technologicznej).
- Pierwszy test polega na utworzeniu zleceń produkcyjnych na wyroby o identyfikatorach od 16 do 20 (5 zleceń produkcyjnych). Składa się on z dwóch części. W części pierwszej, dla utworzonych zleceń produkcyjnych uruchamiane są procedury składowane budujące klasyczny harmonogram aktywny dla różnych metod szeregowania zadań. Procedury składowane generują wartości parametrów (C_{max} oraz $l_h(KP_x^2)$) podlegających porównaniu. Do celów testowych wykorzystano 7 metod szeregowania zadań: FIFO, LIFO, SPT, LPT, MWR i LWR oraz metodę szeregowania zadań za pomocą algorytmu genetycznego. Otrzymano łącznie 7 zestawów parametrów porównawczych dla przypadku klasycznego harmonogramowania general job-shop.
- Druga część pierwszego testu polega na uruchomieniu procedur składowanych budujących harmonogram z wirtualnymi komórkami przedmiotowymi, dla tego samego zbioru zleceń na wyroby o identyfikatorach od 16 do 20. Są to procedury składowane przetwarzające dane w oparciu o działanie algorytmu ProduFlow PM. Procedury składowane ponownie generują wartości parametrów porównawczych (C_{max} oraz $l_h(KP_x^2)$). Ponownie wykorzystano 7 metod szeregowania zadań: FIFO, LIFO, SPT, LPT, MWR i LWR oraz metodę szeregowania zadań za pomocą algorytmu genetycznego. Otrzymano łącznie 7 zestawów parametrów porównawczych dla przypadku harmonogramowania z wirtualnymi komórkami przedmiotowymi.
- Drugi test polega na dodaniu nowego zlecenia na kolejny wyrób o identyfikatorze 21 do zbioru zleceń wykorzystanych w pierwszym teście. Drugi test również składa się z dwóch części, w których uruchamiane są procedury składowane zgodnie z opisem powyżej. Procedury składowane generują łącznie 14 zestawów parametrów porównawczych (C_{max} oraz $l_h(KP_x^2)$) z dwóch części testów.
- Proces testowania powtarzany jest 9 razy, za każdym razem dodając nowe zlecenie na wyrób z kolejnym identyfikatorem (np. test trzeci to dodanie zlecenia na wyrób 22 do zbioru zleceń na wyroby 16-21, itd.).

- Ilustrację grupowanych zleceń produkcyjnych (dendrogramy) dla zbiorów zleceń z kolejnych testów zaprezentowano na rysunkach od Rys. 50 do Rys. 60, natomiast wyniki, tj. otrzymane wartości parametrów porównawczych (C_{max} oraz $l_h(KP_x^2)$), dla kolejnych metod szeregowania zadań zamieszczono w kolejnych tabelach (od Tab. 14 do Tab. 20) oraz na wykresach (od Rys. 61 do Rys. 74).

Proces testowania ma na celu sprawdzenie, jak na tle tradycyjnego sposobu planowania i harmonogramowania różnymi metodami szeregowania zadań, przedstawiają się wyniki otrzymane po wykonaniu planowania przepływu produkcji i harmonogramowania z wirtualnymi komórkami przedmiotowymi za pomocą algorytmu ProduFlow PM. Przeprowadzenie testu eksperymentalnego pozwoli odpowiedzieć na pytanie, czy dzięki zastosowaniu nowej metody planowania przepływu produkcji w warunkach zmiennego zapotrzebowania można doprowadzić do skrócenia terminu realizacji zbioru zleceń w porównaniu z klasycznym sposobem planowania oraz harmonogramowania, niezależnie od dobranej metody szeregowania zadań.

Do przeprowadzenia testu eksperymentalnego wymagane jest ustawienie parametrów, związanych z oszacowaniem długości okna planistycznego. Zgodnie z przedstawionym w rozdziale 4 opisem, w celu prawidłowego działania algorytmu ProduFlow PM należy ustalić wartości następujących wskaźników: wielkość produkcji na każde zlecenie charakteryzujący typ produkcji małoseryjnej, wskaźniki nieplanowanych zdarzeń oraz wartości kosztów jednostkowych wcześniejszego i późniejszego zakończenia zlecenia w stosunku do planowanego terminu realizacji zbioru zleceń. W teście eksperymentalnym przyjęto następujące wartości parametrów, zebrane w Tab. 13.

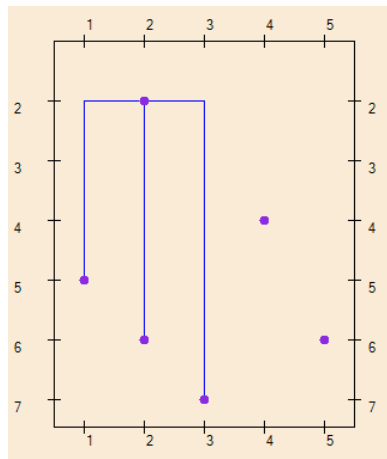
Stałe wartości parametrów	
Wielkość produkcji [szt/zlecenie]	100
Wskaźnik dostępności η_A	0,6
Wskaźnik priorytetowości η_P	0,7
Koszt jednostkowy wcześniejszego zakończenia $k_E(KP_x^2, h)$	2
Koszt jednostkowy późniejszego zakończenia $k_T(KP_x^2, h)$	500

Tab. 13. Przyjęte stałe wartości parametrów wykorzystywanych w trakcie działania algorytmu ProduFlow PM [opracowanie własne].

W kolejnym podrozdziale przedstawiono wyniki grupowania zleceń w postaci narysowanych programowo dendrogramów dla kolejnych prób.

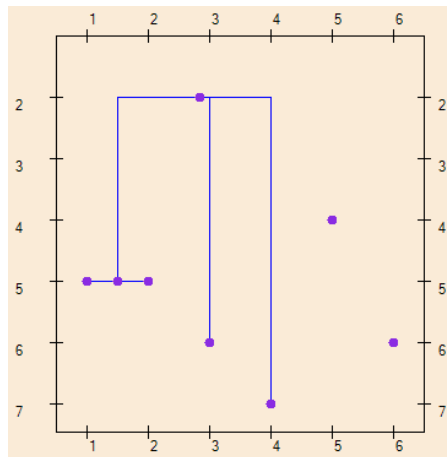
6.1. Test eksperymentalny – wyniki grupowania zleceń

W pierwszej próbie występuje 5 zleceń produkcyjnych dla wyrobów o identyfikatorach 16-20, zgrupowanych w sposób przedstawiony na dendrogramie z Rys. 50.



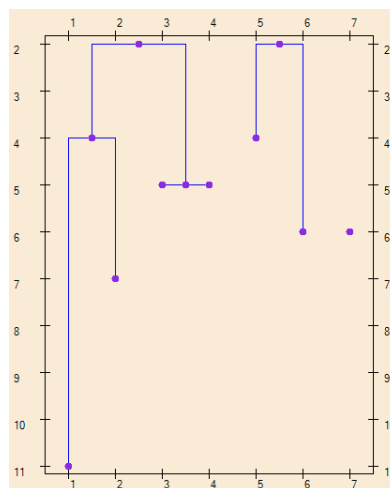
Rys. 50. Dendrogram dla zgrupowanych zleceń wyrobów 16-20 [opracowanie własne].

W drugiej próbie do zbioru 5 zleceń dodaje się nowe zlecenie produkcyjne na wyrób o identyfikatorze 22. Po fazie klastrowania powstał dendrogram (Rys. 51).



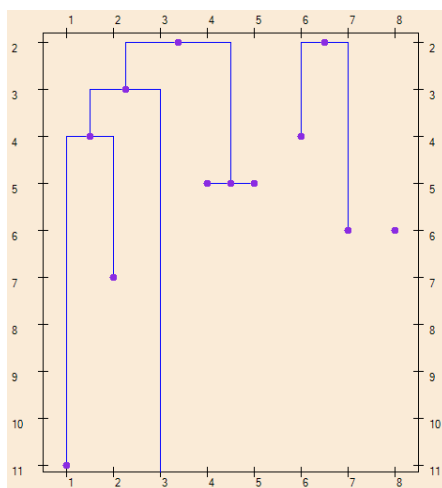
Rys. 51. Dendrogram dla zgrupowanych zleceń wyrobów 16-21 [opracowanie własne].

W trzeciej próbie występuje 7 zleceń produkcyjnych dla wyrobów o identyfikatorach 16-22, zgrupowanych w sposób przedstawiony na dendrogramie z Rys. 52.



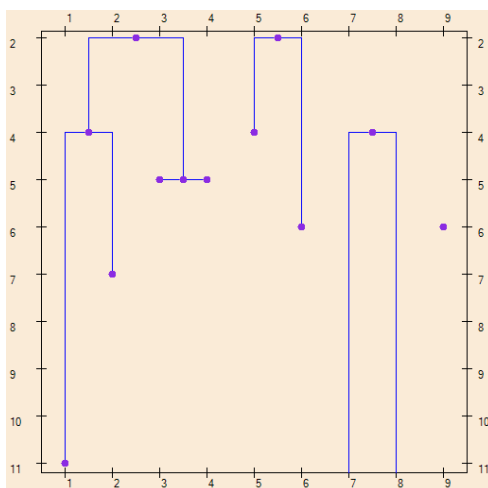
Rys. 52. Dendrogram dla zgrupowanych zleceń wyrobów 16-22 [opracowanie własne].

W czwartej próbie występuje 8 zleceń produkcyjnych dla wyrobów o identyfikatorach 16-23, zgrupowanych w sposób przedstawiony na dendrogramie z Rys. 53.



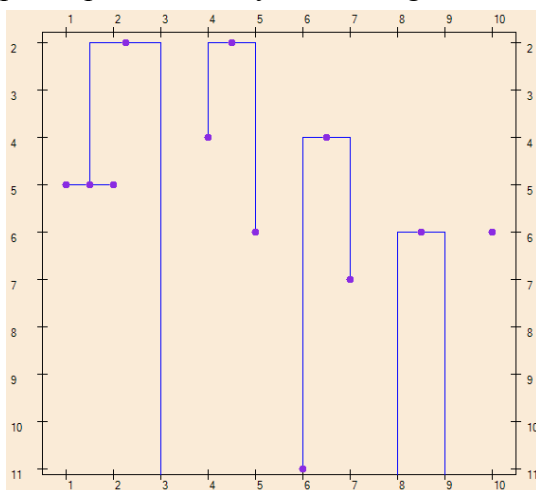
Rys. 53. Dendrogram dla zgrupowanych zleceń wyrobów 16-23 [opracowanie własne].

W piątej próbie występuje 9 zleceń produkcyjnych dla wyrobów o identyfikatorach 16-24, zgrupowanych w sposób przedstawiony na dendrogramie z Rys. 54.



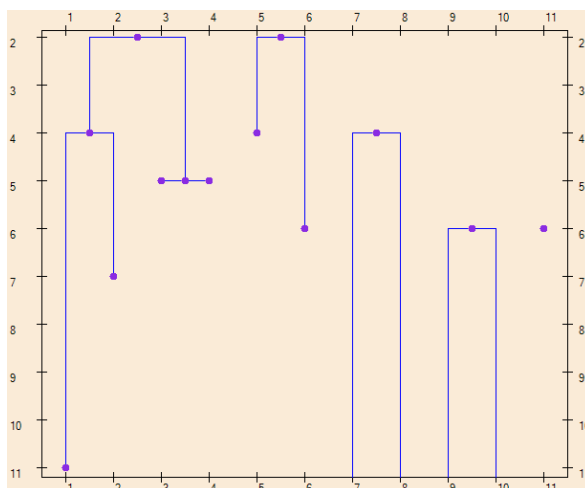
Rys. 54. Dendrogram dla zgrupowanych zleceń wyrobów 16-24 [opracowanie własne].

W szóstej próbie występuje 10 zleceń produkcyjnych dla wyrobów o identyfikatorach 16-25, zgrupowanych w sposób przedstawiony na dendrogramie z Rys. 55.



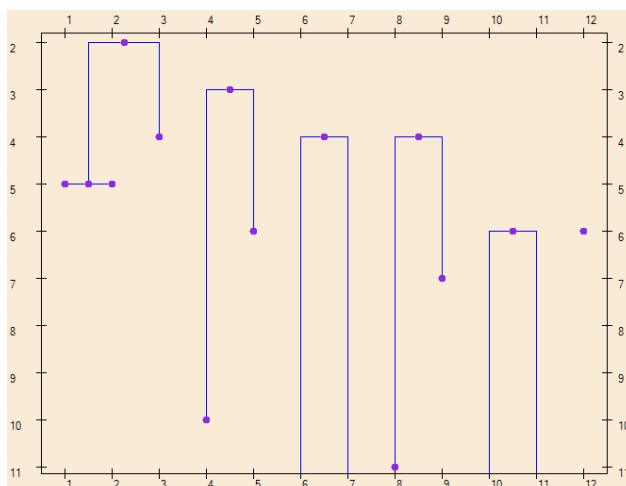
Rys. 55. Dendrogram dla zgrupowanych zleceń wyrobów 16-25 [opracowanie własne].

W siódmej próbie występuje 11 zleceń produkcyjnych dla wyrobów o identyfikatorach 16-26, zgrupowanych w sposób przedstawiony na dendrogramie z Rys. 56.



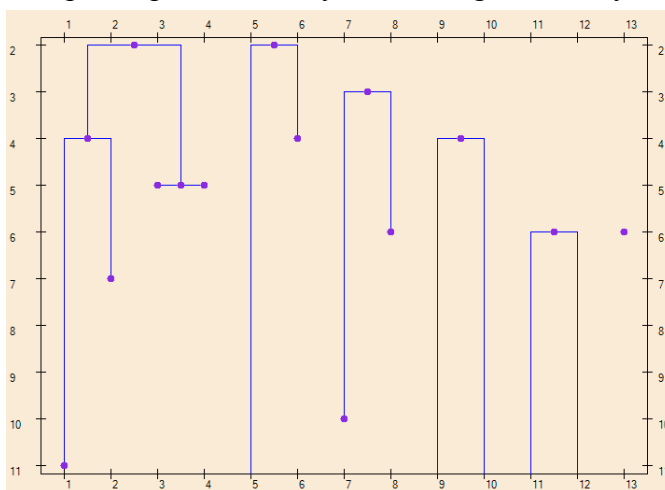
Rys. 56. Dendrogram dla zgrupowanych zleceń wyrobów 16-26 [opracowanie własne].

W ósmej próbie występuje 12 zleceń produkcyjnych dla wyrobów o identyfikatorach 16-27, zgrupowanych w sposób przedstawiony na dendrogramie z Rys. 57.



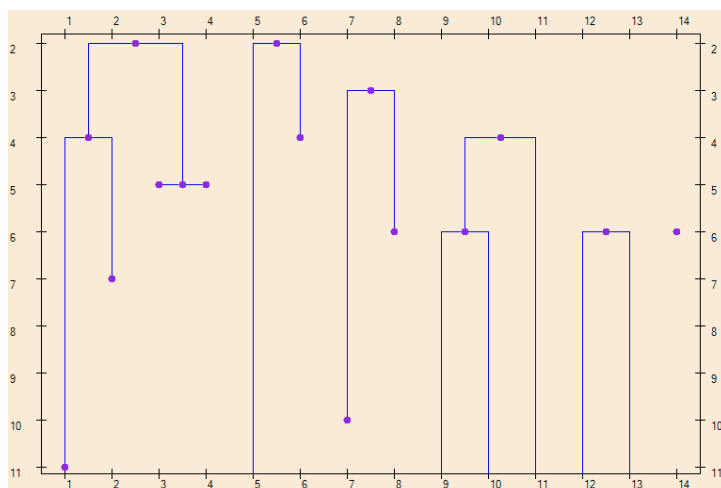
Rys. 57. Dendrogram dla zgrupowanych zleceń wyrobów 16-27 [opracowanie własne].

W dziewiątej próbie występuje 13 zleceń produkcyjnych dla wyrobów o identyfikatorach 16-28, zgrupowanych w sposób przedstawiony na dendrogramie z Rys. 58.



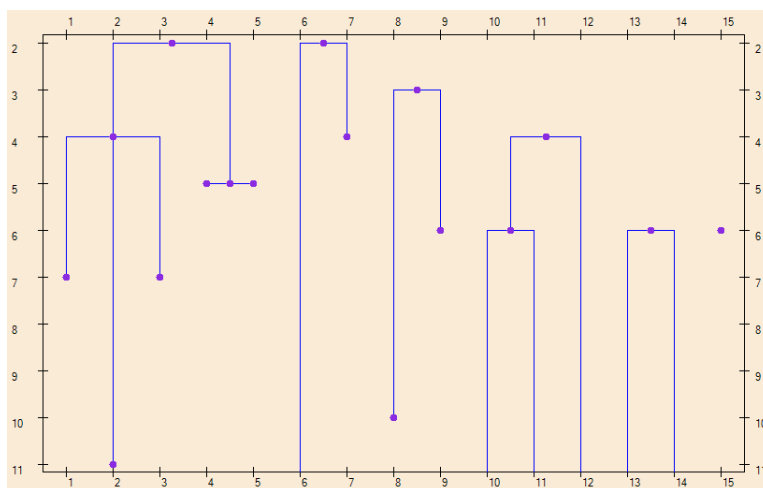
Rys. 58. Dendrogram dla zgrupowanych zleceń wyrobów 16-28 [opracowanie własne].

W dziesiątej próbie występuje 14 zleceń produkcyjnych dla wyrobów o identyfikatorach 16-29, zgrupowanych w sposób przedstawiony na dendrogramie z Rys. 59.



Rys. 59. Dendrogram dla zgrupowanych zleceń wyrobów 16-29 [opracowanie własne].

W jedenastej próbie występuje 15 zleceń produkcyjnych dla wyrobów o identyfikatorach 16-30, zgrupowanych w sposób przedstawiony na dendrogramie z Rys. 60.



Rys. 60. Dendrogram dla zgrupowanych zleceń wyrobów 16-30 [opracowanie własne].

Podczas realizacji testów eksperymentalnych zbudowano łącznie 77 harmonogramów, w 11 testach. W pojedynczym teście zbudowano dwa harmonogramy zgodnie z przyjętym planem. Po utworzeniu pojedynczego harmonogramu obliczono parametry C_{max} oraz $l_h(KP_x^2)$. Zestawienie wyników oraz wykresy zależności terminów zakończenia zbioru zleceń od liczby zleceń a także zależności szacowanych długości okna planistycznego od liczby zleceń przedstawiono w kolejnym podrozdziale.

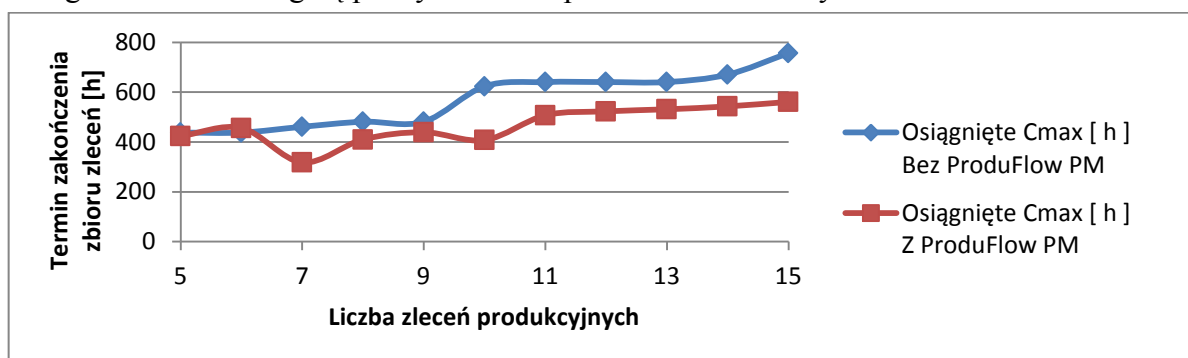
6.2. Test eksperymentalny – porównanie wyników planowania i harmonogramowania z zastosowaniem algorytmu ProduFlow PM na tle harmonogramowania general job-shop

Wyniki otrzymane dla szeregowania zadań z zastosowaniem reguły priorytetu FIFO zestawiono w Tab. 14.

Test eksperymentalny FIFO							
Liczba zleceń	Identyfikator wyrobów	Osiągnięte C_{max} [h]		Osiągnięte $I_h(KP_x^2)$ [h]		Liczba zmian 8-godz.	
		Bez ProduFlow PM	Z ProduFlow PM	Bez ProduFlow PM	Z ProduFlow PM	Bez ProduFlow PM	Z ProduFlow PM
5	16 - 20	437,8	423,9	504	424	63	53
6	16 - 21	437,8	456	440	464	55	58
7	16 - 22	461	318,2	464	528	58	66
8	16 - 23	481,3	410,2	488	712	61	89
9	16 - 24	481,3	439,14	488	440	61	55
10	16 - 25	623,32	407,72	624	568	78	71
11	16 - 26	640,48	507,08	648	728	81	91
12	16 - 27	640,48	522,94	648	704	81	88
13	16 - 28	640,48	531,12	648	728	81	91
14	16 - 29	670,48	542,87	672	584	84	73
15	16 - 30	756,68	560,77	760	720	95	90

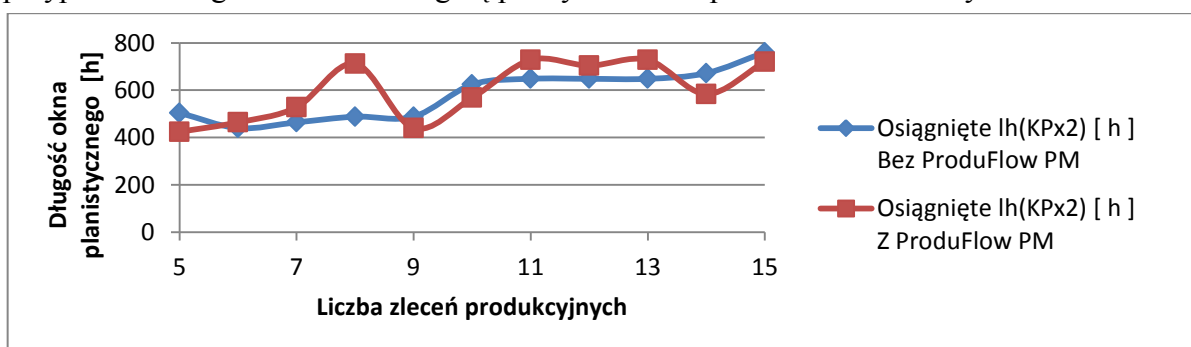
Tab. 14. Wyniki testu eksperymentalnego dla metody szeregowania: reguła priorytetu FIFO [opracowanie własne].

Wykres zależności terminów zakończenia zbioru zleceń od liczby zleceń dla przypadku szeregowania zadań regułą priorytetu FIFO przedstawiono na Rys. 61.



Rys. 61. Wykres zależności terminów zakończenia zbioru zleceń od liczby zleceń (reguła priorytetu FIFO) [opracowanie własne].

Wykres zależności szacowanych długości okna planistycznego od liczby zleceń dla przypadku szeregowania zadań regułą priorytetu FIFO przedstawiono na Rys. 62.



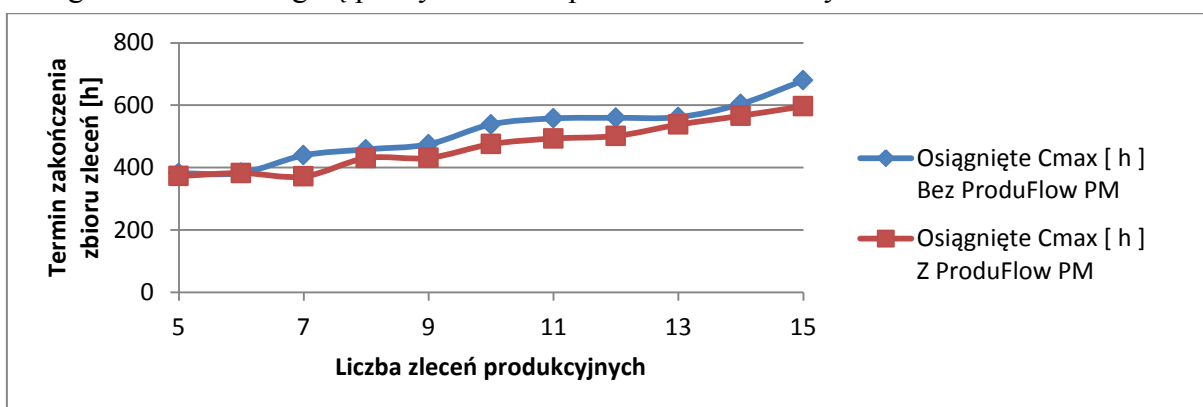
Rys. 62. Wykres zależności szacowanych długości okna planistycznego od liczby zleceń (reguła priorytetu FIFO) [opracowanie własne].

Wyniki otrzymane dla szeregowania zadań z zastosowaniem reguły priorytetu LIFO zestawiono w Tab. 15.

Test eksperymentalny LIFO							
Liczba zleceń	Identyfikator wyrobów	Osiągnięte C_{max} [h]		Osiągnięte $I_h(KP_x^2)$ [h]		Liczba zmian 8-godz.	
		Bez ProduFlow PM	Z ProduFlow PM	Bez ProduFlow PM	Z ProduFlow PM	Bez ProduFlow PM	Z ProduFlow PM
5	16 - 20	381,7	372,4	384	560	48	70
6	16 - 21	382,7	382,3	384	384	48	48
7	16 - 22	439,12	371,4	440	632	55	79
8	16 - 23	458,32	430,3	464	712	58	89
9	16 - 24	474,82	430,9	480	632	60	79
10	16 - 25	538,43	475,09	544	536	68	67
11	16 - 26	557,59	493,19	560	536	70	67
12	16 - 27	559,59	501,31	560	576	70	72
13	16 - 28	561,79	537,7	568	608	71	76
14	16 - 29	603,91	565,95	608	616	76	77
15	16 - 30	679,48	596,69	680	656	85	82

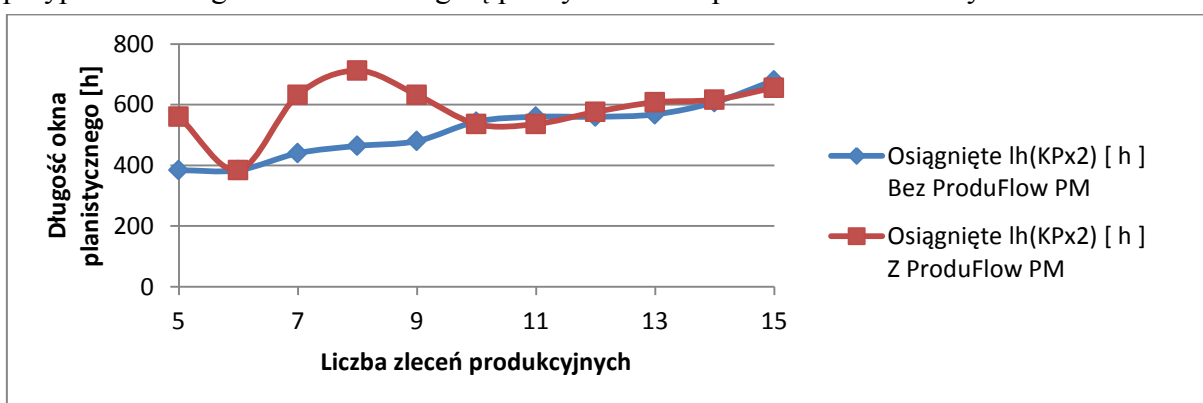
Tab. 15. Wyniki testu eksperymentalnego dla metody szeregowania: reguła priorytetu LIFO [opracowanie własne].

Wykres zależności terminów zakończenia zbioru zleceń od liczby zleceń dla przypadku szeregowania zadań regułą priorytetu LIFO przedstawiono na Rys. 63.



Rys. 63. Wykres zależności terminów zakończenia zbioru zleceń od liczby zleceń (reguła priorytetu LIFO) [opracowanie własne].

Wykres zależności szacowanych długości okna planistycznego od liczby zleceń dla przypadku szeregowania zadań regułą priorytetu LIFO przedstawiono na Rys. 64.



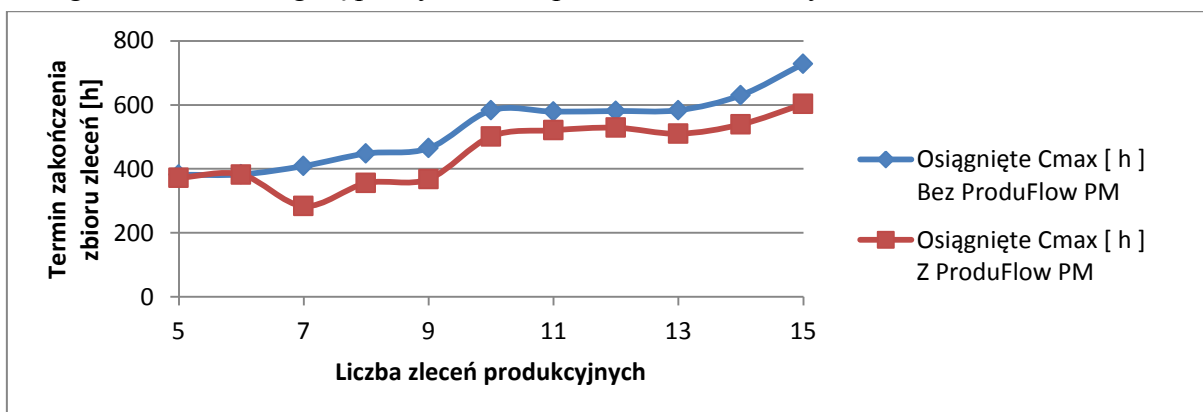
Rys. 64. Wykres zależności szacowanych długości okna planistycznego od liczby zleceń (reguła priorytetu LIFO) [opracowanie własne].

Wyniki otrzymane dla szeregowania zadań z zastosowaniem reguły priorytetu SPT zestawiono w Tab. 16.

Test eksperymentalny SPT							
Liczba zleceń	Identyfikator wyrobów	Osiągnięte C_{max} [h]		Osiągnięte $I_h(KP_x^2)$ [h]		Liczba zmian 8-godz.	
		Bez ProduFlow PM	Z ProduFlow PM	Bez ProduFlow PM	Z ProduFlow PM	Bez ProduFlow PM	Z ProduFlow PM
5	16 - 20	381,7	372,4	384	416	48	52
6	16 - 21	382,7	382,3	384	384	48	48
7	16 - 22	409,02	283,6	416	448	52	56
8	16 - 23	448,04	355,5	456	384	57	48
9	16 - 24	464,54	368,2	472	616	59	77
10	16 - 25	582,99	501,38	584	568	73	71
11	16 - 26	578,85	520,68	584	576	73	72
12	16 - 27	580,85	528,62	584	608	73	76
13	16 - 28	583,35	509,88	584	568	73	71
14	16 - 29	630,15	538,98	632	632	79	79
15	16 - 30	728,45	602,64	736	608	92	76

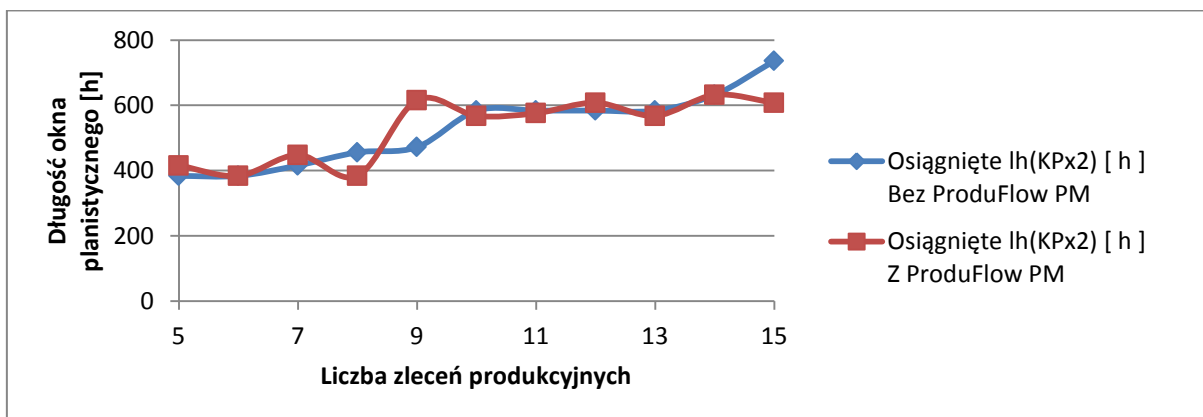
Tab. 16. Wyniki testu eksperymentalnego dla metody szeregowania: reguła priorytetu SPT [opracowanie własne].

Wykres zależności terminów zakończenia zbioru zleceń od liczby zleceń dla przypadku szeregowania zadań regułą priorytetu SPT przedstawiono na Rys. 65.



Rys. 65. Wykres zależności terminów zakończenia zbioru zleceń od liczby zleceń (reguła priorytetu SPT) [opracowanie własne].

Wykres zależności szacowanych długości okna planistycznego od liczby zleceń dla przypadku szeregowania zadań regułą priorytetu SPT przedstawiono na Rys. 66.



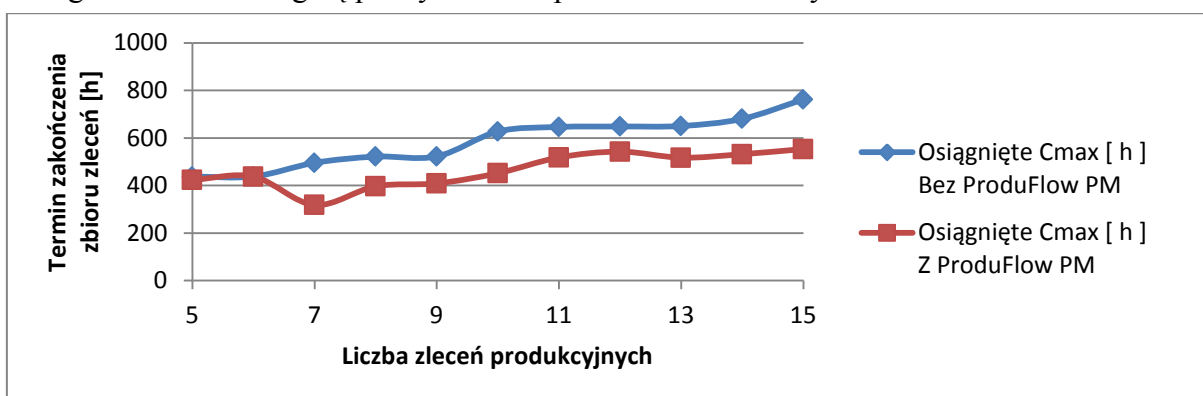
Rys. 66. Wykres zależności szacowanych długości okna planistycznego od liczby zleceń (reguła priorytetu SPT) [opracowanie własne].

Wyniki otrzymane dla szeregowania zadań z zastosowaniem reguły priorytetu LPT zestawiono w Tab. 17.

Test eksperymentalny LPT							
Liczba zleceń	Identyfikatory wyrobów	Osiągnięte C_{max} [h]		Osiągnięte $I_h(KP_x^2)$ [h]		Liczba zmian 8-godz.	
		Bez ProduFlow PM	Z ProduFlow PM	Bez ProduFlow PM	Z ProduFlow PM	Bez ProduFlow PM	Z ProduFlow PM
5	16 - 20	437,8	423,9	512	424	64	53
6	16 - 21	437,8	437,3	528	520	66	65
7	16 - 22	495,52	318,2	552	568	69	71
8	16 - 23	521,54	396,6	568	568	71	71
9	16 - 24	522,44	408,95	560	664	70	83
10	16 - 25	627,01	451,8	632	504	79	63
11	16 - 26	646,17	518,07	648	680	81	85
12	16 - 27	648,17	542,42	656	560	82	70
13	16 - 28	650,37	517,17	656	680	82	85
14	16 - 29	680,37	531,77	696	696	87	87
15	16 - 30	762,07	553,47	768	568	96	71

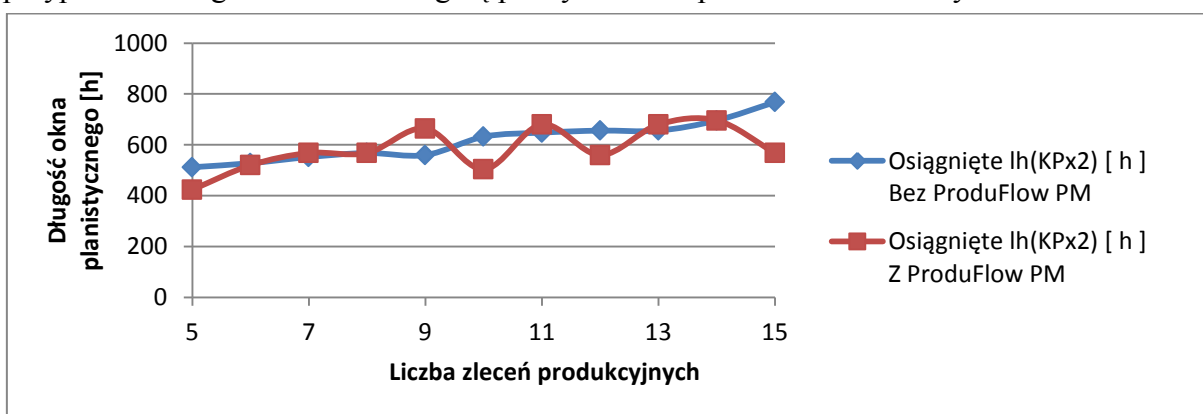
Tab. 17. Wyniki testu eksperymentalnego dla metody szeregowania: reguła priorytetu LPT [opracowanie własne].

Wykres zależności terminów zakończenia zbioru zleceń od liczby zleceń dla przypadku szeregowania zadań regułą priorytetu LPT przedstawiono na Rys. 67.



Rys. 67. Wykres zależności terminów zakończenia zbioru zleceń od liczby zleceń (reguła priorytetu LPT) [opracowanie własne].

Wykres zależności szacowanych długości okna planistycznego od liczby zleceń dla przypadku szeregowania zadań regułą priorytetu LPT przedstawiono na Rys. 68.



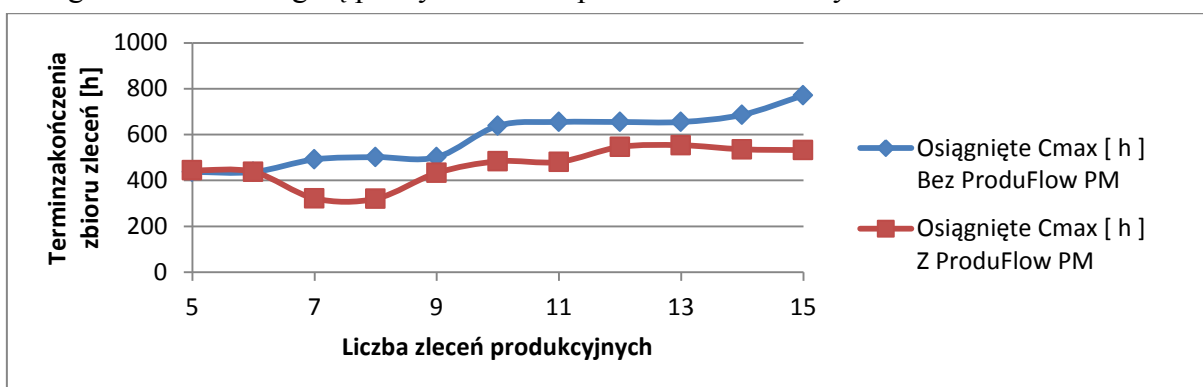
Rys. 68. Wykres zależności szacowanych długości okna planistycznego od liczby zleceń (reguła priorytetu LPT) [opracowanie własne].

Wyniki otrzymane dla szeregowania zadań z zastosowaniem reguły priorytetu MWR zestawiono w Tab. 18.

Test eksperymentalny MWR							
Liczba zleceń	Identyfikator wyrobów	Osiągnięte C_{max} [h]		Osiągnięte $I_h(KP_x^2)$ [h]		Liczba zmian 8-godz.	
		Bez ProduFlow PM	Z ProduFlow PM	Bez ProduFlow PM	Z ProduFlow PM	Bez ProduFlow PM	Z ProduFlow PM
5	16 - 20	437,2	444,4	504	448	63	56
6	16 - 21	437,2	437,3	520	440	65	55
7	16 - 22	491,85	321,85	544	552	68	69
8	16 - 23	501,85	320,6	568	376	71	47
9	16 - 24	501,85	432,8	680	656	85	82
10	16 - 25	638,52	483,55	704	552	88	69
11	16 - 26	654,98	481	704	624	88	78
12	16 - 27	654,98	546,25	656	640	82	80
13	16 - 28	654,98	553,71	704	640	88	80
14	16 - 29	686,48	535,9	992	624	124	78
15	16 - 30	771,18	532,39	776	688	97	86

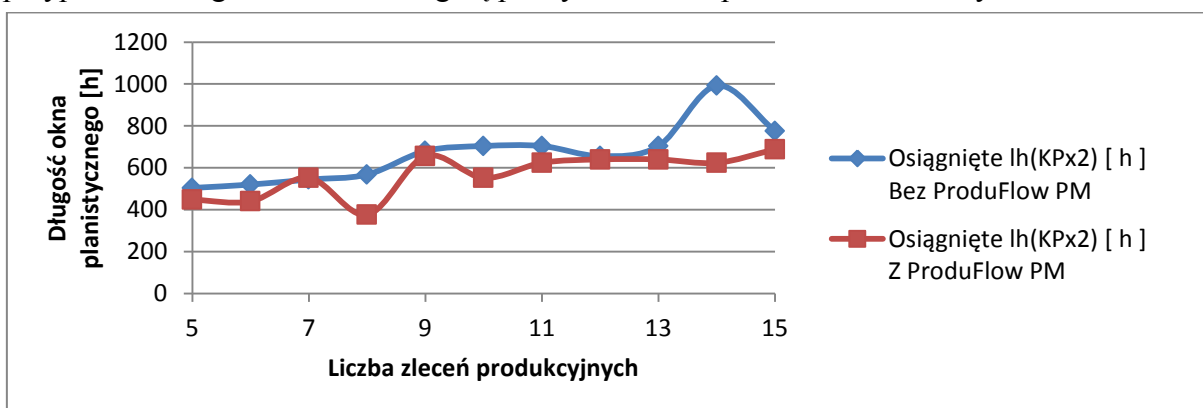
Tab. 18. Wyniki testu eksperymentalnego dla metody szeregowania: reguła priorytetu MWR [opracowanie własne].

Wykres zależności terminów zakończenia zbioru zleceń od liczby zleceń dla przypadku szeregowania zadań regułą priorytetu MWR przedstawiono na Rys. 69.



Rys. 69. Wykres zależności terminów zakończenia zbioru zleceń od liczby zleceń (reguła priorytetu MWR) [opracowanie własne].

Wykres zależności szacowanych długości okna planistycznego od liczby zleceń dla przypadku szeregowania zadań regułą priorytetu MWR przedstawiono na Rys. 70.



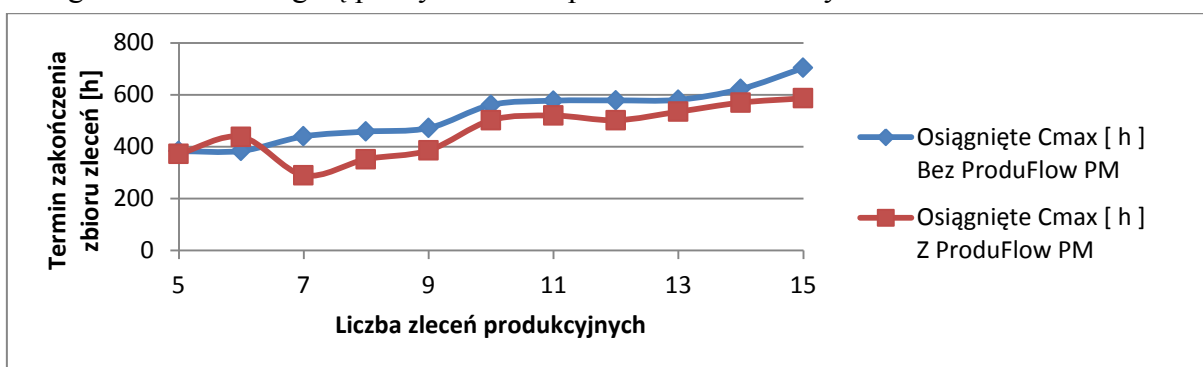
Rys. 70. Wykres zależności szacowanych długości okna planistycznego od liczby zleceń (reguła priorytetu MWR) [opracowanie własne].

Wyniki otrzymane dla szeregowania zadań z zastosowaniem reguły priorytetu LWR zestawiono w Tab. 19.

Test eksperymentalny LWR							
Liczba zleceń	Identyfikator wyrobów	Osiągnięte C_{max} [h]		Osiągnięte $I_h(KP_x^2)$ [h]		Liczba zmian 8-godz.	
		Bez ProduFlow PM	Z ProduFlow PM	Bez ProduFlow PM	Z ProduFlow PM	Bez ProduFlow PM	Z ProduFlow PM
5	16 - 20	382,7	372,4	384	416	48	52
6	16 - 21	383,7	437,3	384	440	48	55
7	16 - 22	439,52	290	440	504	55	63
8	16 - 23	458,1	351,7	464	480	58	60
9	16 - 24	472,1	386	480	392	60	49
10	16 - 25	560,66	501,98	568	568	71	71
11	16 - 26	577,12	519,78	584	728	73	91
12	16 - 27	578,37	501,9	584	568	73	71
13	16 - 28	580,57	534,74	584	600	73	75
14	16 - 29	622,27	569,82	624	576	78	72
15	16 - 30	703,32	586,36	704	592	88	74

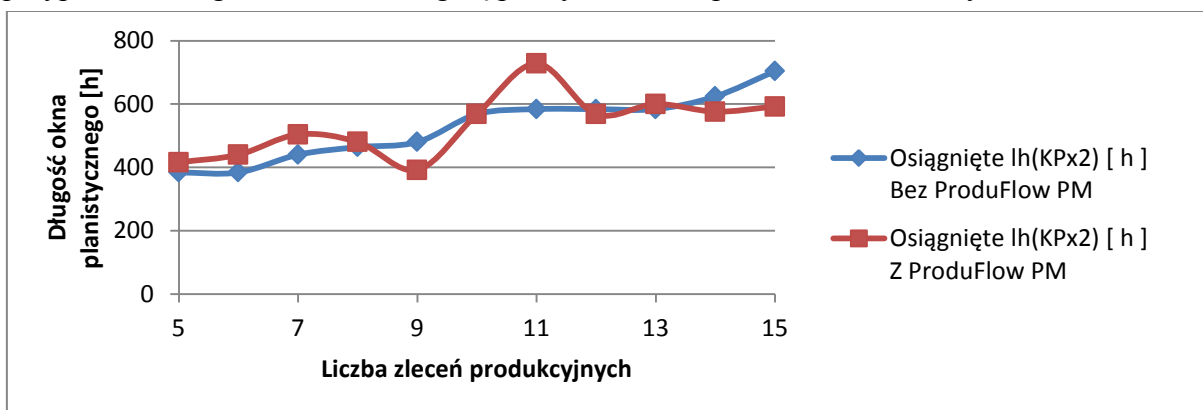
Tab. 19. Wyniki testu eksperymentalnego dla metody szeregowania: reguła priorytetu LWR [opracowanie własne].

Wykres zależności terminów zakończenia zbioru zleceń od liczby zleceń dla przypadku szeregowania zadań regułą priorytetu LWR przedstawiono na Rys. 71.



Rys. 71. Wykres zależności terminów zakończenia zbioru zleceń od liczby zleceń (reguła priorytetu LWR) [opracowanie własne].

Wykres zależności szacowanych długości okna planistycznego od liczby zleceń dla przypadku szeregowania zadań regułą priorytetu LWR przedstawiono na Rys. 72.



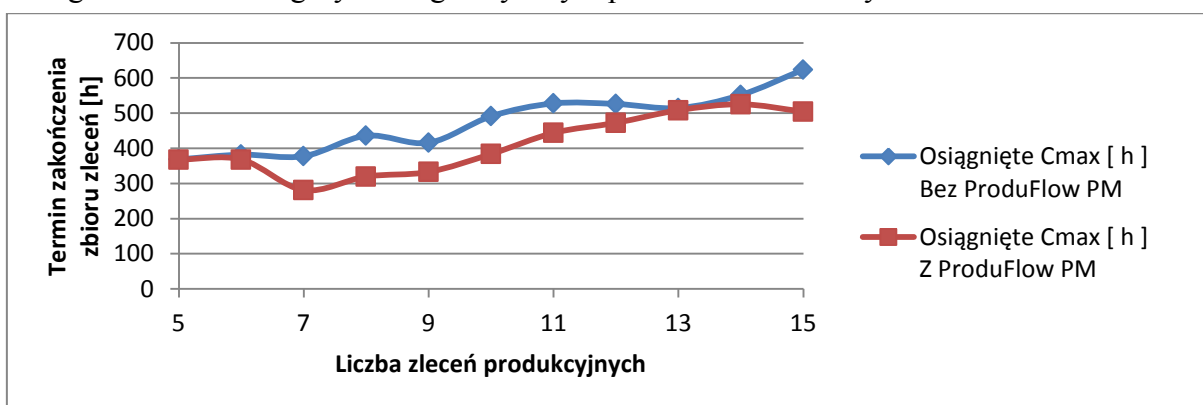
Rys. 72. Wykres zależności szacowanych długości okna planistycznego od liczby zleceń (reguła priorytetu LWR) [opracowanie własne].

Wyniki otrzymane dla szeregowania zadań z zastosowaniem algorytmu genetycznego zestawiono w Tab. 20.

Test eksperymentalny GA - algorytm genetyczny							
Liczba zleceń	Identyfikator wyrobów	Osiągnięte C_{max} [h]		Osiągnięte $I_h(KP_x^2)$ [h]		Liczba zmian 8-godz.	
		Bez ProduFlow PM	Z ProduFlow PM	Bez ProduFlow PM	Z ProduFlow PM	Bez ProduFlow PM	Z ProduFlow PM
5	16 - 20	368,5	368	464	472	58	59
6	16 - 21	381,9	368	384	472	48	59
7	16 - 22	377,6	281,2	424	536	53	67
8	16 - 23	435,8	319,7	584	504	73	63
9	16 - 24	415,82	332,8	736	456	92	57
10	16 - 25	491,17	383,75	496	576	62	72
11	16 - 26	527,7	443,81	800	600	100	75
12	16 - 27	526,11	472,48	608	824	76	103
13	16 - 28	514,61	508,31	752	520	94	65
14	16 - 29	551,86	525,3	1000	624	125	78
15	16 - 30	623,87	504,4	688	624	86	78

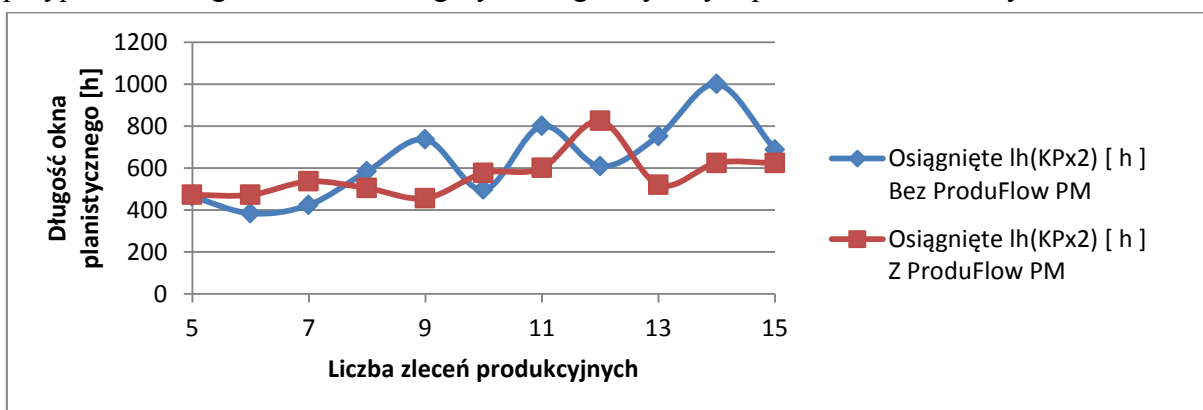
Tab. 20. Wyniki testu eksperymentalnego dla metody szeregowania: algorytm genetyczny [opracowanie własne].

Wykres zależności terminów zakończenia zbioru zleceń od liczby zleceń dla przypadku szeregowania zadań algorytmem genetycznym przedstawiono na Rys. 73.



Rys. 73. Wykres zależności terminów zakończenia zbioru zleceń od liczby zleceń (algorytm genetyczny) [opracowanie własne].

Wykres zależności szacowanych długości okna planistycznego od liczby zleceń dla przypadku szeregowania zadań algorytmem genetycznym przedstawiono na Rys. 74.



Rys. 74. Wykres zależności szacowanych długości okna planistycznego od liczby zleceń (algorytm genetyczny) [opracowanie własne].

Na poszczególnych wykresach zależności terminu zakończenia zbioru zleceń od liczby zleceń produkcyjnych zaznaczono osiągnięte wartości parametru C_{max} dwoma kolorami. Kolorem niebieskim zaznaczono wartości parametru C_{max} dla pierwszej części testów eksperymentalnych (dla przypadku klasycznego harmonogramowania general job-shop). Kolorem czerwonym zaznaczono wartości parametru C_{max} dla drugiej części testów eksperymentalnych (dla przypadku harmonogramowania z wirtualnymi komórkami przedmiotowymi). Przedstawione na wykresach wyniki wskazują na skuteczność działania algorytmu ProduFlow PM pod względem skrócenia terminu zakończenia zbioru zleceń w stosunku do klasycznego harmonogramowania general job-shop. Przebieg linii niebieskich oraz czerwonych na kolejnych wykresach jest zbliżony do siebie. Jedyna różnica pojawia się na wykresie przedstawiającym wyniki otrzymane dla harmonogramowania z szeregowaniem zadań z zastosowaniem algorytmu genetycznego. Może być to spowodowane błędnie przyjętymi założeniami w konstrukcji algorytmu genetycznego (wybór sposobu krzyżowania w jednym punkcie, wartość prawdopodobieństwa mutacji oraz postać funkcji przystosowania osobników i sposób selekcji osobników do krzyżowania w kolejnej iteracji).

Na wykresach zależności długości okienek planistycznych od liczby zleceń produkcyjnych zaznaczono osiągnięte wartości parametru $l_h(KP_x^2)$ również dwoma kolorami. Przyjęta kolorystyka jest taka sama jak na wykresach opisanych powyżej. Z wykresów widać znaczne wahania długości okienek planistycznych. Wahania są bardziej nasilone w przypadku zastosowania algorytmu ProduFlow PM do planowania przepływu produkcji, choć również występują w przypadku klasycznego harmonogramowania general job-shop. Z wykresów nie wynika bezpośrednia przyczyna występowania tego zjawiska co wskazuje na brak bezpośredniego związku pomiędzy parametrami C_{max} oraz $l_h(KP_x^2)$. Bezpośredni wpływ na wahania długości okienek planistycznych ma zwiększona koncentracja terminów zakończenia poszczególnych zleceń $Cz_y(KP_x^2, h)$ w pobliżu wartości parametru C_{max} . Koncentracja terminów zakończenia zbioru zleceń $Cz_y(KP_x^2, h)$ w pobliżu C_{max} wpływa na opóźnienie planowanego terminu realizacji zbioru zleceń $d(KP_x^2, h)$, co w konsekwencji prowadzi do wydłużenia okienka planistycznego. W połączeniu z ustalonymi wartościami parametrów nieplanowanych zdarzeń oraz wartości kosztów jednostkowych wcześniejszego i późniejszego zakończenia zlecenia w stosunku do planowanego terminu realizacji zbioru zleceń dało to efekt niepożądany w postaci zmieniającej się długości okna planistycznego.

7. WNIOSKI I PODSUMOWANIE

Całokształt prowadzonych przez autora prac badawczych w dziedzinie planowania przepływu produkcji w warunkach zmiennego zapotrzebowania miał na celu przygotowanie podstaw pod sformułowanie nowej metody, umożliwiającej uzyskanie skróconego czasu realizacji zbioru zleceń. W efekcie prowadzenia tych prac badawczych powstała metoda, oparta o wykorzystanie wirtualnych komórek przedmiotowych wewnątrz komórek produkcyjnych II stopnia. Chociaż w analizie literatury przedstawiono zarys rozwoju zarówno metod planowania MRP/APS jak i systemów bazujących na tych metodach, z założeń do pracy wynika, że ich podstawową rolą w stosunku do opracowanej metody planowania przepływu produkcji jest generowanie (dostarczanie) zleceń produkcyjnych. Sama metoda planowania przepływu produkcji, opracowana przez autora pracy, integruje proces planowania przepływu produkcji i harmonogramowania w stosunku do systemów APS, których rolą jest wyłącznie generowanie harmonogramów typu general job-shop.

Przedstawiona w pracy nowa metoda planowania przepływu produkcji w warunkach zmiennego zapotrzebowania została sformalizowana w postaci algorytmu o nazwie ProduFlow PM. Schemat ogólny algorytmu ProduFlow PM przedstawia trzy fazy, do których należą grupowanie zleceń, planowanie przepływu produkcji z harmonogramowaniem oraz szacowanie długości okresu planowania. Każda z faz została szczegółowo opisana w osobnym rozdziale. Wyjaśniono schematycznie oraz opisowo wszystkie algorytmy wewnętrzne faz zawartych w podstawowym algorytmie ProduFlow PM. Zostały również zaprezentowane przykłady użycia poszczególnych faz wyjaśniające ich funkcjonalność.

Algorytm ProduFlow PM został zaimplementowany w postaci systemu komputerowego, którego sercem jest baza danych SQL Server 2008 R2. Dzięki temu nie ma ograniczenia co do ilości informacji (np. liczby zleceń produkcyjnych, długości marszrut technologicznych, liczby maszyn) przetwarzanych przez algorytm ProduFlow PM. System komputerowy pozwolił na przeprowadzenie szeregu testów eksperymentalnych mających na celu weryfikację opracowanej metody planowania przepływu produkcji.

Przeprowadzono testowanie operacowanego algorytmu ProduFlow PM z punktu widzenia skuteczności jego działania prowadzącego do skrócenia terminu zakończenia zbioru zleceń C_{max} , niezależnie od wybranej metody harmonogramowania produkcji. Testowano również wpływ wypracowanego planu przepływu produkcji na długość okna planistycznego $l_h(KP_x^2)$. Zebrano otrzymane wyniki oraz przedstawiono je na dwóch wykresach zależności terminów zakończenia zbioru zleceń od liczby zleceń a także zależności szacowanych długości okna planistycznego od liczby zleceń. W wyniku przeprowadzonych testów eksperymentalnych możliwe było sformułowanie szeregu wniosków dotyczących nowej metody planowania przepływu produkcji:

1. Dążenie do minimalizacji terminu zakończenia zbioru zleceń C_{max} nie zawsze jest korzystne z punktu widzenia potwierdzenia terminu realizacji klientom, jeżeli weźmie się pod uwagę wpływ nieplanowanych zdarzeń na realizację zleceń produkcyjnych oraz możliwość wprowadzania zleceń priorytetowych w trakcie realizacji har-

monogramu według przyjętego przepływu produkcji. Im poziom obciążenia systemu wytwórczego jest większy, lub im więcej terminów zakończenia zleceń jest zbliżonych do terminu zakończenia zbioru zleceń, tym zwiększa się prawdopodobieństwo opóźnień. W związku z tym wartość szacowanej długości okna planistycznego $l_h(KP_x^2)$, równoznaczna z przyjętym terminem realizacji zbioru zleceń, jest wrażliwa na losowo występujące zlecenia produkcyjne. Taki efekt zaobserwowano na wykresach związanych z szeregowaniem zadań regułami priorytetu FIFO, LPT oraz algorytmem genetycznym (charakterystyczne falowanie linii czerwonej, a w przypadku algorytmu genetycznego również niebieskiej).

2. Uzyskanie minimalnej wartości terminu zakończenia zbioru zleceń C_{max} jest silnie uzależnione od przyjętej metody grupowania zleceń produkcyjnych. Potwierdza to powszechne przekonanie autorów publikacji związanych z badaniami n.t. wydzielania komórek przedmiotowych, mówiące o tym, że jest to najważniejszy etap projektowania systemów produkcyjnych. W pracy wykorzystano hierarchiczny aglomeracyjny algorytm klastrowania. Za jego pomocą uzyskuje się grupy zleceń w sposób ściśle określony. Pierwszeństwo w tworzeniu grup zleceń mają zlecenia z dłuższymi marszrutami technologicznymi. Choć do celów pracy jest to wystarczające, ponieważ uzyskuje się skrócenie terminu zakończenia zbioru zleceń w porównaniu do klasycznego harmonogramowania general job-shop, to proces klastrowania powinien być ukierunkowany na poszukiwanie rozwiązań (poszukiwanie grup zleceń), które w efekcie przyczynią się do minimalizacji terminu zakończenia zbioru zleceń C_{max} .
3. Planowanie przepływu produkcji w sposób zaprezentowany w pracy polega na kombinatorycznym wyłanianiu grup zleceń, których fragmenty (wydzielone wspólne łańcuchy zadań charakteryzujące się ukierunkowaniem oraz sekwencją zadań na kolejnych maszynach) są wykonywane w wirtualnych komórkach przedmiotowych. Do kombinatorycznego wyłaniania grup zleceń produkowanych w wirtualnych komórkach przedmiotowych wykorzystano algorytm genetyczny, gdyż zadanie potraktowano jako problem optymalizacyjny z kryterium C_{max} . Wyniki zaprezentowane na wykresie zależności terminów zakończenia zbioru zleceń od liczby zleceń (Rys. 73, wyznaczony C_{max} dla liczby zadań 12, 13 oraz 14 jest zbliżony zarówno przy klasycznym harmonogramowaniu general job-shop jak i przy harmonogramowaniu z użyciem algorytmu ProduFlow PM) świadczą o błędach implementacji algorytmu genetycznego w algorytmie ProduFlow PM.
4. Przepływ produkcji oparty o wykorzystanie wirtualnych komórek przedmiotowych wymaga dobrej synchronizacji pracy na maszynach dedykowanych do pracy w takich komórkach. Synchronizacja produkcji w wirtualnych komórkach przedmiotowych jest uzależniona od czynnika ludzkiego, nie branego pod uwagę w pracy. Wymagany jest więc rozwój metody w zakresie uwzględnienia pracy ludzkiej w tego typu przypadkach.

W aspekcie przedstawionych wyników, weryfikujących działanie metody planowania przepływu produkcji w warunkach zmiennego zapotrzebowania stwierdza się słuszność po-

stawionej tezy pracy. W porównaniu do tradycyjnego sposobu planowania i harmonogramowania w systemach job-shop, uzyskano skrócenie terminu realizacji zbioru zleceń C_{max} dzięki nowej metodzie planowania przepływu produkcji.

Jednocześnie cel pracy został osiągnięty. Opracowano algorytm do racjonalnego planowania przepływu produkcji w oknach planistycznych, w warunkach zmiennego zapotrzebowania. Racjonalizacja planowania przepływu produkcji umożliwiła skrócenie czasu realizacji zleceń produkcyjnych z założeniem, że raz ustalony termin realizacji zleceń produkcyjnych nie ulega zmianie nawet w przypadku wystąpienia nieoczekiwanych zdarzeń.

Z przedstawionego toku obliczeniowego oraz w wyniku implementacji metody do opracowanego przez autora systemu komputerowego wynika, że nowa metoda planowania przepływu produkcji może z powodzeniem stanowić moduł uzupełniający systemów ERP, stanowiący narzędzie wykorzystywane na poziomie krótkookresowego planowania produkcji. W tym zakresie powinny być kontynuowane prace badawcze prowadzące do rozwoju funkcjonalności algorytmu ProduFlow PM. Rozwój powinien być również ukierunkowany na uwzględnienie fazy szczegółowej kontroli produkcji (ang. Shop Floor Control) aby możliwa było stwierdzenie realizowalności wypracowanego planu przepływu produkcji.

Na zakończenie należy stwierdzić, że proces budowy założeń metody planowania przepływu produkcji, opracowania algorytmu ProduFlow PM oraz jego implementacji do systemu komputerowego jest niezwykle pracochłonny. Świadczy o tym chociażby fakt, iż przygotowanie aplikacji klienckiej wymagało napisania 50000 linii kodu w języku Visual Basic, a przygotowanie bazy danych do działania według algorytmu ProduFlow PM wymagało napisania ok. 150000 linii kodu w języku T-SQL. Jednakże ze względu na spodziewane korzyści w wyniku zastosowania tej metody planowania przepływu produkcji w warunkach produkcji małoseryjnej, dalsze prace rozwojowe powinny być prowadzone.

8. LITERATURA

- [1] Ah Kioon S., Bulgak A., Bektas T.: *Integrated cellular manufacturing systems design with production planning and dynamic system reconfiguration*. European Journal of Operational Research 192, 2009, p.414-428.
- [2] Al-Hinai N., ElMekkawy T.: *Robust and stable flexible job-shop scheduling with random machine breakdowns using a hybrid genetic algorithm*. International Journal of Production Economics 132, 2011, p. 279-291.
- [3] Alhourani F., Seifoddini H.: *Machine cell formation for production management in cellular manufacturing systems*. International Journal of Production Research 45, 2007, 913-934.
- [4] Almeder C.: *A hybrid optimization approach for multi-level capacitated lot-sizing problems*. European Journal of Operational Research 200, 2010, p.599-606.
- [5] Arkat J., Hosseini L., Farahani M.: *Minimization of exceptional elements and voids in the cell formation problem using a multi-objective genetic algorithm*. Expert Systems with Applications 38, 2011, 9597-9602.
- [6] Aryanezhad M., Deljoo V., Mirzapour S.: *Dynamic cell formation and the worker assignment problem: a new model*. International Journal of Advanced Manufacturing Technology 41, 2009, p.329-342.
- [7] Askin R., Goldberg J.: *Design and analysis of Lean Production Systems*. John Wiley & Sons Inc., New York, USA 2002.
- [8] Baker K., Trietsch D.: *Safe scheduling: setting due dates in single machine problems*. European Journal of Operational Research 196, 2009, p.69-77.
- [9] Balakrishnan J., Cheng C.: *Dynamic cellular manufacturing under multi-period planning horizons*. Journal of Manufacturing Technology Management 16, 2005, p.516-530.
- [10] Balakrishnan J., Cheng C.: *Multi-period planning and uncertainty issues in cellular manufacturing: A review and future directions*. European Journal of Operational Research 177, 2007, p.281-309.
- [11] Baykasoglu A.: *Capability-based distributed layout approach for virtual manufacturing cells*. International Journal of Production Research 41, 2003, p.2597-2618.
- [12] Baykasoglu A., Gocken M., Unutmaz Z.: *New approaches to due date assignment in job shops*. European Journal of Operational Research 187, 2008, p.31-45.
- [13] Beard, J., Sumner M.: *Seeking strategic advantage in the post-net era: Viewing ERP systems from the resource-based perspective*. Journal of Strategic Information Systems 13, 2004, p.129-150.
- [14] Berry W., Cooper M.: *Manufacturing flexibility: methods for measuring the impact of product variety on performance in process industry*. Journal of Operations Management 17, 1999, p.163-178.
- [15] Botta-Genoulaz V., Millet P., Grabot B.: *A survey on the recent research literature on ERP systems*. Computers in Industry 56, 2005, p.510-522.
- [16] Boulif M., Atif K.: *A new branch-and-bound enhanced genetic algorithm for the manufacturing cell formation problem*. Computers & Operations Research 33, 2006, 2219-2245.
- [17] Boulif M., Atif K.: *A new fuzzy genetic algorithm for the dynamic bi-objective cell formation problem considering passive and active strategies*. International Journal of Approximate Reasoning 47, 2008, 141-165.
- [18] Bozarth C., Handfield R.: *Wprowadzenie do zarządzania operacjami i łańcuchem dostaw*. Prentice Hall, New Jersey, USA 2006, tłum. Helion, Gliwice 2007.
- [19] Brzeziński M.: *Organizacja produkcji*. Wydawnictwo Politechniki Lubelskiej, Lublin 2000.
- [20] Brzeziński M.: *Organizacja i sterowanie produkcją: projektowanie systemów produkcyjnych i procesów sterowania produkcją*. Placet, Warszawa 2002.
- [21] Caux C., Bruniaux R., Pierreval H.: *Cell formation with alternative process plans and machine capacity constraints: A new combined approach*. International Journal of Production Economics 64, 2000, 179-284.
- [22] Chan W., Chan C., Ip W.: *A heuristic algorithm for machine assignment in cellular manufacturing*. Computers & Industrial Engineering 44, 2002, 49-73.
- [23] Chang P., HsinChen S., Mani V.: *A note on due date assignment and single machine scheduling with a learning/aging effect*. International Journal of Production Economics 117, 2009, p.142-149.
- [24] Chen D., Li S., Tang G.: *Single machine scheduling with common due date assignment in a group technology environment*. Mathematical and Compute Modelling 25, 1997, p.81-90.
- [25] Chen I.: *Planning for ERP systems: Analysis and future trend*. Business Process Management Journal 7, 2001, p.374-386.
- [26] Chen K., Ji P.: *A genetic algorithm for dynamic advanced planning and scheduling (DAPS) with frozen interval*. Expert Systems with Applications 33, 2007, p.1004-1010.
- [27] Chen K., Ji P.: *A mixed integer programming model for advanced planning and scheduling (APS)*. European Journal of Operational Research 181, 2007, p.515-522.

- [28] Cheng Ch., Goh Ch., Lee A.: *Designing group technology manufacturing systems using heuristics branching rules*. Computers & Industrial Engineering 40, 2001, p.117-131.
- [29] Cheng T.C., Gupta M.C.: *Survey of scheduling research involving due date determination decisions*. European Journal of Operational Research 38, 1989, p.156-166.
- [30] Chung S., Wu T., Chang C.: *An efficient tabu search algorithm to the cell formation problem with alternative routings and machine reliability considerations*. Computers & Industrial Engineering 60, 2011, 7-15.
- [31] Cox J., Blackstone J.: *APICS Dictionary – 12th Edition*. The Association for Operations Management, Chicago, USA, 2010.
- [32] Czerska J.: *Doskonalenie strumienia wartości*. Wydawnictwo Difin, Warszawa 2009.
- [33] Defersha F., Chen M.: *A comprehensive mathematical model for the design of cellular manufacturing systems*. International Journal of Production Economics 103, 2006, p.767-783.
- [34] Defersha F., Chen M.: *A linear programming embedded genetic algorithm for an integrated cell formation and lot sizing considering product quality*. European Journal of Operational Research 187, 2008, p.46-69.
- [35] Deljoo V., Mirzapour S., Deljoo F., Aryanezhad M.: *Using genetic algorithm to solve dynamic cell formation problem*. Applied Mathematical Modelling 34, 2010, p.1078-1092.
- [36] Donselaar K., Gubbels B.: *How to release orders in order to minimise system inventory and system nervousness?*. International Journal of Production Economics 78, 2002, p.335-343.
- [37] Durlik I.: *Inżynieria zarządzania cz.I. Strategie organizacji produkcji. Nowe koncepcje zarządzania*. Wydawnictwo Placet, Warszawa 2004.
- [38] van Eck M.: *Advanced planning and scheduling. Is logistics everything?*. Paper for Business Mathematics and Informatics, Amsterdam, Netherlands 2003.
- [39] Ehie I., Madsen M.: *Identifying critical issues in enterprise resource planning (ERP) implementation*. Computers in Industry 56, 2005, p.545-557.
- [40] ElMaraghy H., Azab A., Schuh G., Pulz C.: *Managing variations in products, processes and manufacturing systems*. CIRP Annals – Manufacturing Technology 58, 2009, p.441-446.
- [41] Eulalia L., D'Amours S., Frayret J., Menegusso C., Azevedo R.: *Advanced Supply Chain Planning Systems (APS) today and tomorrow*. Rozdział w książce: pt: "Supply Chain Management – Pathways for research and practice", edytor: Dilek Onkal, InTech Open Access Book, 2011.
- [42] Fandel G., Stammen-Hegene C.: *Simultaneous lot sizing and scheduling for multi-product multi-level production*. International Journal of Production Economics 104, 2006, p.308-316.
- [43] Fattahi P., Fallahi A.: *Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability*. CIRP Journal of Manufacturing Science and Technology 2, 2010, p.114-123.
- [44] Filho E., Tiberti A.: *A group genetic algorithm for the machine cell formation problem*. International Journal of Production Economics 102, 2006, p.1-21.
- [45] Ford H.: *Dziś i jutro*. Wydawnictwo ProdPress.com, Wrocław 2007, tłumaczenie z oryginału: Doubleday. Page & Company, New York, USA 1926.
- [46] Foulds L., French A., Wilson J.: *The sustainable cell formation problem: manufacturing cell creation with machine modification costs*. Computer & Operations Research 33, 2006, 1010-1032.
- [47] Foulds L., Neumann K.: *A network flow model of group technology*. Mathematical and Computer Modelling 38, 2003, p.623-635.
- [48] French S.: *Sequencing and scheduling: an introduction to the mathematics of the job shop*. John Wiley & Sons Inc., New York, USA 1982.
- [49] Fung Y., Liang F., Jiang Z.: *A multi-stage methodology for virtual cell formation oriented agile manufacturing*. International Journal of Advanced Manufacturing Technology 36, 2008, p.798-810.
- [50] Ghosh T., Sengupta S., Chattopadhyay M., Dan P.: *Meta-heuristics in cellular manufacturing: A state-of-the-art review*. International Journal of Industrial Engineering Computation 2, 2011, 87-122.
- [51] Ghosh T., Sengupta S., Dan P.: *A hybrid heuristic based clustering algorithm to design manufacturing cell*. Management and Production Engineering Review 1, 2010, 26-37.
- [52] Ghotboddini M., Rabbani M., Rahimian H.: *A comprehensive dynamic cell formation design: Benders' decomposition approach*. Expert Systems with Applications 38, 2011, p.2478-2488.
- [53] Glover F.: *Tabu Search: Part I*. Operations Research Society of America Journal on Computing 1, 1989, p.190-206.
- [54] Glover F.: *Tabu Search: Part II*. Operations Research Society of America Journal on Computing 2, 1990, p.4-32.
- [55] Goldberg D.: *Algorytmy genetyczne i ich zastosowania*. Addison-Wesley Publishing Company Inc., 1989, tłum. WNT, Warszawa 2003.
- [56] Gordon V., Proth J.M., Chu C.B.: *A survey of the state-of-the-art of common due date assignment and scheduling research*. European Journal of Operational Research 139, 2002, p.1-25.
- [57] Green T., Sadowski R.: *A review of cellular manufacturing assumptions, advantages and design techniques*. Journal of Operations Management 4, 1984, p.85-97.

- [58] Gutierrez C., Garcia-Magarino I.: *Modular design of hybrid genetic algorithm for a flexible job-shop scheduling problem*. Knowledge-Based Systems 24, 2011, p.102-112.
- [59] Habel J., Wojakowski P.: *Virtual cellular manufacturing – A review*. Proceedings of The 8th International Conference CEuRSIS, Baia Mare, Romania, 2010, p.111-116.
- [60] Hameri A.: *Production flow analysis – cases from manufacturing and service industry*. International Journal of Production Economics 129, 2011, p.233-241.
- [61] Helber S., Sahling F.: *A fix-and-optimize approach for the multi-level capacitated lot sizing problem*. International Journal of Production Economics 123, 2010, p.247-256.
- [62] Huang H.: *Integrated production model in agile manufacturing systems*. International Journal of Advanced Manufacturing Technology 20, 2002, p.515-525.
- [63] Hvolby H., Steger-Jensen K.: *Technical and industrial issues of Advanced Planning and Scheduling (APS) systems*. Computers in Industry 61, 2010, p.845-851.
- [64] Hyer N., Brown K.: *The discipline of real cells*. Journal of Operations Management 17, 1999, p.557-574.
- [65] Imai M.: *Gemba Kaizen*. Wydawnictwo MT Biznes, Warszawa 2006.
- [66] Irani S., Cavalier T., Cohen P.: *Virtual manufacturing cells: exploiting layout design and intercell flows for the machine sharing problem*. International Journal of Production Research 31, 1993, p.791-810.
- [67] Jackowicz R., Lis S., Wagner K.: *Projektowanie organizacji komórki produkcyjnej w warunkach produkcji powtarzalnej*. Wydawnictwa Politechniki Warszawskiej, Warszawa 1980.
- [68] Jackowicz R., Santarek K.: *Organizacja podstawowych procesów produkcyjnych*. Wydawnictwa Politechniki Warszawskiej, Warszawa 1982.
- [69] Jacobs F., Weston Jr. F.: *Enterprise resource planning (ERP) – A brief history*. Journal of Operations Management 25, 2007, p.357-363.
- [70] Jensen J., Malhotra M., Philipoom P.: *Machine dedication and process flexibility in a group technology environment*. Journal of Operations Management 14, 1996, p.19-39.
- [71] Jeunet J., Jonard N.: *Measuring the performance of lot sizing techniques in uncertain environments*. International Journal of Production Economics 64, 2000, p.197-208.
- [72] Kannan V., Ghosh S.: *A virtual cellular manufacturing approach to batch production*. Decision Sciences 27, 1996, p.519-539.
- [73] Kernighan B., Lin S.: *An efficient heuristic procedure for partitioning graphs*. Bell Systems Technical Journal 49, 1970, p.291-307.
- [74] Kesen S., Das S., Güngör Z.: *A genetic algorithm based heuristic for scheduling of virtual manufacturing cells (VCMs)*. Computers & Operations Research 37, 2010, 1148-1156.
- [75] Kesen S., Toksari M., Güngör Z., Güner E.: *Analyzing the behaviors of virtual cells (VCs) and traditional manufacturing systems: Ant colony optimization (ACO)-based metamodels*. Computers & Operations Research 36, 2009, p.2275-2285.
- [76] Kim C., Baek J.: *A two-phase heuristic algorithm for cell formation problems considering alternative part routes and machine sequences*. International Journal of Production Research 42, 2004, 3911-3927.
- [77] Kis T.: *Job-shop scheduling with processing alternatives*. European Journal of Operational Research 151, 2003, p.307-332.
- [78] Ko K., Egbelu P.: *Virtual cell formation*. International Journal of Production Research 41, 2003, p.2365-2389.
- [79] Koh S., Gunasekaran A., Rajkumar D.: *ERP II: The involvement, benefits, and impediments of collaborative information sharing*. International Journal of Production Economics 113, 2008, p.245-268.
- [80] Koh S., Saad S.: *Managing uncertainty in ERP-controlled manufacturing environments in SMEs*. International Journal of Production Economics 101, 2006, p.109-127.
- [81] Kopanos G., Mendez C., Puigjander L.: *MIP-based decomposition strategies for large-scale scheduling problems in multiproduct multistage batch plants. A benchmark scheduling problem of the pharmaceutical industry*. European Journal of Operational Research 207, 2010, p.644-655.
- [82] Kornberger Z.: *Technologia obróbki skrawaniem i montażu*. Wydawnictwa Naukowo-Techniczne, Warszawa, 1974.
- [83] Kusiak A.: *The generalized group technology concept*. International Journal of Production Research 25, 1987, p.561-569.
- [84] Lee H., Park N., Jeong H., Park J.: *Grid enabled MRP process improvement under distributed database environment*. The Journal of Systems and Software 82, 2009, p.1087-1097.
- [85] Lee W., Wu Ch.: *A note on single-machine group scheduling problems with position-based learning effect*. Applied Mathematical Modelling 33, 2009, p.2159-2163.
- [86] Lee W., Wu Ch.: *A note on optimal policies for two group scheduling problems with deteriorating setup and processing times*. Computers & Industrial Engineering 58, 2010, p.646-650.
- [87] Lee Y., Jeong C., Moon C.: *Advanced planning and scheduling with outsourcing in manufacturing supply chain*. Computers & Industrial Engineering 43, 2002, p.351-374.
- [88] Li J., Pan Q., Liang Y.: *An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems*. Computers & Industrial Engineering 59, 2010, p.647-662.

- [89] Li S., Ng C., Yuan J.: *Group scheduling and due date assignment on a single machine*. International Journal of Production Economics 130, 2011, p.230-235.
- [90] Lis S., Santarek K.: *Projektowanie rozmieszczenia stanowisk roboczych*. Wydawnictwo Naukowe PWN, Warszawa 1980.
- [91] Ławrynowicz A.: *Zarządzanie produkcją w ogniwie sieci dostaw z zastosowaniem systemu eksperckiego i algorytmu genetycznego*. Rozprawa Naukowa nr 119, Wydawnictwa Uczelniane Akademii Techniczno-Rolniczej, Bydgoszcz 2006.
- [92] Mabert V.: *The early road material requirements planning*. Journal of Operations Management 25, 2007, p.346-356.
- [93] Mahdavi I., Aalaei A., Paydar M., Solimanpur M.: *Designing a mathematical model for dynamic cellular manufacturing systems considering production planning and worker assignment*. Computers and Mathematics with Application 60, 2010, 1014-1025.
- [94] Mahdavi I., Mahadevan B.: *CLASS: An algorithm for cellular manufacturing system and layout design using sequence data*. Robotics and Computer-Integrated Manufacturing 24, 2008, 488-497.
- [95] Mahdavi I., Paydar M., Solimanpur M., Heidarzade A.: *Genetic algorithm approach for solving a cell formation problem in cellular manufacturing*. Expert Systems with Applications 36, 2009, 6598-6604.
- [96] Mak K., Wang X.: *Production scheduling and cell formation for virtual cellular manufacturing systems*. International Journal of Advanced Manufacturing Technology 20, 2002, p.144-152.
- [97] Mazurczak J.: *Projektowanie struktur systemów produkcyjnych*. Wydawnictwo Politechniki Poznańskiej, Poznań 2002.
- [98] McKay K., Black G.: *The evolution of production planning system: A 10-year case study*. Computers in Industry 58, 2007, p.756-771.
- [99] Mosheiov G.: *A note: Multi-machine scheduling with general position-based deterioration to minimize total load*. International Journal of Production Economics 135, 2012, p.523-525.
- [100] Moslehi G., Mahnem M.: *A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search*. International Journal of Production Economics 129, 2011, p.14-22.
- [101] Muda S., Hendry L.: *Developing a new world class model for small and medium sized make-to-order companies*. International Journal of Production Economics 78, 2002, p.295-310.
- [102] Mukattash A., Adil M., Tahboub K.: *Heuristic approaches for part assignment in cell formation*. Computers & Industrial Engineering 42, 2002, p.329-341.
- [103] Mula J., Poler R., Garcia-Sabater J., Lario F.: *Models for production planning under uncertainty: A review*. International Journal of Production Economics 103, 2006, p.271-285.
- [104] Mungwatanna A.: *Design of cellular manufacturing systems for dynamic and uncertain production requirement with presence of routing flexibility*. Ph.D. thesis, State University of Virginia, Blacksburg 2000.
- [105] Naadimuthu G., Gultom P., Lee E.: *Fuzzy clustering in cell formation with multiple attributes*. Computers and Mathematics with Applications 59, 2010, 3137-3147.
- [106] Nakagiri D., Kuriyama S.: *A study of production management system with MRP*. International Journal of Production Economics 44, 1996, p.27-33.
- [107] Nasiri M., Kianfar F.: *A GA/TS algorithm for the stage shop scheduling problem*. Computers & Industrial Engineering 61, 2011, p.161-170.
- [108] Neto A., Filho E.: *A simulation-based evolutionary multi-objective approach to manufacturing cell formation*. Computers & Industrial Engineering 59, 2010, 64-74.
- [109] Noktehdan A., Karimi B., Kashan A.: *A differential evolution algorithm for the manufacturing cell formation problem using group based operators*. Expert Systems with Applications 37, 2010, 4822-4829.
- [110] Nomden G., Slomp J., Suresh N.: *Virtual manufacturing cells: taxonomy of past research and identification of future research issues*. International Journal of Flexible Manufacturing Systems 17, 2006, p.71-92.
- [111] Nomden G., van der Zee D.: *Virtual cellular manufacturing: configuring routing flexibility*. International Journal of Production Economics 112, 2008, p.439-451.
- [112] Nouri H., Tang S., Hang Tuah B., Anuar M.: *BASE: A bacteria foraging algorithm for cell formation with sequence data*. Journal of Manufacturing Systems 29, 2010, 102-110.
- [113] Oguz C., Salman F., Yalcin Z.: *Order acceptance and scheduling decisions in make-to-order systems*. International Journal of Production Economics 125, 2010, p.200-211.
- [114] Ohta H., Nakamura M.: *Cell formation with reduction in setup times*. Computers & Industrial Engineering 42, 2002, 317-327.
- [115] Olhager J.: *Strategic positioning of the order penetration point*. International Journal of Production Economics 85, 2003, p.319-329.
- [116] Oliveira S., Ribeiro J., Seok S.: *A spectral clustering algorithm for manufacturing cell formation*. Computers & Industrial Engineering 57, 2009, 1008-1014.
- [117] Orlicky J.: *Materials requirement planning*. McGraw-Hill, New York, USA 1976.

- [118] Özgüven C., Özbakir L., Yavuz Y.: *Mathematical models for job-shop scheduling problems with routing and process plan flexibility*. Applied Mathematical Modelling 34, 2010, p.1539-1548.
- [119] Öztürk A., Kayaligil S., Özdemirel N.: *Manufacturing lead time estimation using data mining*. European Journal of Operational Research 173, 2006, p.683-700.
- [120] Papaioannou G., Wilson J.M.: *The evolution of cell formation problem methodologies based on recent studies (1997-2008): Review and directions for future research*. European Journal of Operational Research 206, 2010, 509-521.
- [121] Pawlak M.: *Algorytmy ewolucyjne jako narzędzie harmonogramowania produkcji*. Wydawnictwo Naukowe PWN, Warszawa 1999.
- [122] Pekny J.: *A brief guide to purchasing Advanced Planning and Scheduling (APS) System*. Whitepaper: Scheduling Systems Buyer's Guide, Advanced Process Combinatorics Inc., 2005.
- [123] Prince J., Kay J.: *Combining lean and agile characteristics: creating of virtual groups by enhanced production flow analysis*. International Journal of Production Economics 85, 2003, p.305-318.
- [124] Ratchev S.: *Concurrent process and facility prototyping for formation of virtual manufacturing cells*. Integrated Manufacturing Systems 12, 2001, p.306-315.
- [125] Rezaeian J., Javadian N., Tavakkoli-Moghaddam R., Jolai F.: *A hybrid approach based on the genetic algorithm and neural network to design an incremental cellular manufacturing system*. Applied Soft Computing 11, 2011, 4195-4202.
- [126] Rezazadeh H., Mahini R., Zarei M.: *Solving a dynamic cell formation problem by linear programming embedded particle swarm optimization algorithm*. Applied Soft Computing 11, 2011, p.3160-3169.
- [127] Rheault M., Drolet J., Abdounour G.: *Physically reconfigurable virtual cells: A dynamic model for a highly dynamic environment*. Computers & Industrial Engineering 29, 1995, 221-225.
- [128] Riane F., Artiba A., Iassinovski S.: *An integrated production planning and scheduling system for hybrid flowshop organizations*. International Journal of Production Economics 74, 2001, p.33-48.
- [129] Ruben R., Mahmoodi F.: *Lead time prediction in unbalanced production systems*. International Journal of Production Research 38, 2000, p.1711-1729.
- [130] Saad S.: *The reconfiguration issues in manufacturing systems*. Journal of Materials Processing Technology 138, 2003, p.277-283.
- [131] Saad S., Baykasoglu A., Gindy N.: *An integrated framework for reconfiguration of cellular manufacturing systems using virtual cells*. Production Planning & Control 13, 2002, p.381-393.
- [132] Safaei N., Saidi-Mehrabad M., Jabal-Ameli M.: *A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system*. European Journal of Operational Research 185, 2008, 563-592.
- [133] Sarker B., Li Z.: *Job routing and operation scheduling: a network-based virtual cell formation approach*. Journal of the Operational Research Society 52, 2001, p.673-681.
- [134] Sarker B., Xu Y.: *Designing multi-product lines: Job routing in cellular manufacturing systems*. IIE Transactions 32, 2000, 219-235.
- [135] Satoglu S., Suresh N.: *A goal-programming approach for design of hybrid cellular manufacturing systems in dual resource constrained environment*. Computers & Industrial Engineering 56, 2009, p.560-575.
- [136] Segerstedt A.: *Formulas of MRP*. International Journal of Production Economics 46-47, 1996, p.127-136.
- [137] Selim H., Askin R., Vakharia A.: *Cell formation in group technology: review, evaluation and directions for future research*. Computers & Industrial Engineering 34, 1998, p.3-20.
- [138] Sha D., Liu C.: *Using data mining for due-date assignment in a dynamic job shop environment*. International Journal of Advanced Manufacturing Technology 25, 2005, p.1164-1174.
- [139] Shabtay D.: *Scheduling and due date assignment to minimize earliness, tardiness, holding, due date assignment and batch delivery costs*. International Journal of Production Economics 123, 2010, p.235-242.
- [140] Shabtay D., Itskovitch Y., Yedidsion L., Oron D.: *Optimal due date assignment and resource allocation in a group technology scheduling environment*. Computers & Operations Research 37, 2010, p.2218-2228.
- [141] Simpson J., Hocken R., Albus J.: *The automated manufacturing research facility of the national bureau of standards*. Journal of Manufacturing Systems 1, 1982, p.17-32.
- [142] Skołud B.: *Planowanie wielosortymentowej produkcji rytmicznej*. Rozprawa Naukowa s. Mechanika 136, Wydawnictwo Politechniki Śląskiej, Gliwice 2000.
- [143] Skołud B.: *Zarządzanie operacyjne. Produkcja w małych i średnich przedsiębiorstwach*. Wydawnictwo Politechniki Śląskiej, Gliwice 2006.
- [144] Slomp J., Chowdary B., Suresh N.: *Design of virtual manufacturing cells: a mathematical programming approach*. Robotics and Computer Integrated Manufacturing 21, 2005, p.273-288.
- [145] Solimanpur M., Vrat P., Shankar R.: *A heuristic to minimize makespan of scheduling problem*. International Journal of Production Economics 88, 2004, p.231-241.
- [146] Song D., Hicks C., Earl C.: *Product due date assignment for complex assemblies*. International Journal of Production Economics 76, 2002, p.243-256.

- [147] Stadler H.: *Supply chain management and advanced planning – basics, overview and challenges*. European Journal of Operational Research 163, 2005, p.575-588.
- [148] Stawowy A.: *Evolutionary strategy for manufacturing cell design*. Omega 34, 2006, 1-18.
- [149] Su L., Lien C.: *Scheduling parallel machines with resource-dependent processing times*. International Journal of Production Economics 117, 2009, p.256-266.
- [150] Sum Ch., Quek S., Lim H.: *Analysing interaction effects on MRP implementation using ACE*. International Journal of Production Economics 58, 1999, p.303-318.
- [151] Suresh N., Meredith J.: *Coping with the loss of pooling synergy in cellular manufacturing systems*. Management Science 40, 1994, p.466-483.
- [152] Syswerda G.: *Schedule optimization using genetic algorithms*. The genetic algorithm handbook, Van Nostrand Reinhold, New York, USA, 1991.
- [153] Tariq A., Hussain I., Ghafoor A.: *A hybrid genetic algorithm for machine-part grouping*. Computers & Industrial Engineering 56, 2009, p.347-356.
- [154] Tavakkoli-Moghaddam R., Aryanezhad M., Safaei N., Azaron A.: *Solving a dynamic cell formation problem using metaheuristics*. Applied Mathematics and Computation 170, 2005, 761-780.
- [155] Tavakkoli-Moghaddam R., Rahimi-Vahed A., Ghodrathnama A., Siadat A.: *A simulated annealing method for solving a new mathematical model of multi-criteria cell formation problem with capital constraints*. Advances in Engineering Software 40, 2009, 268-273.
- [156] Thomalla C.: *Job-shop scheduling with alternative process plans*. International Journal of Production Economics 74, 2001, p.125-134.
- [157] Toksari M., Günter E.: *The common due date early/tardy scheduling problem on a parallel machine under the effects of time-dependent learning and linear and nonlinear deterioration*. Expert Systems with Applications 37, 2010, p.92-112.
- [158] Umble E., Haft R., Umble M.: *Enterprise resource planning: Implementation procedures and critical success factors*. European Journal of Operational Research 146, 2003, p.241-257.
- [159] Vakharia J., Moily J., Huang Y.: *Evaluating virtual cells and multistage flow shops: an analytical approach*. International Journal of Flexible Manufacturing Systems 11, 1999, p.291-314.
- [160] Vinod V., Sridharan R.: *Simulation modeling and analysis of due date assignment methods and scheduling decision rules in a dynamic job shop production system*. International Journal of Production Economics 129, 2011, p.127-146.
- [161] Vitanov V., Tjahjono B., Marghalany I.: *A decision support tool to facilitate the design of cellular manufacturing layouts*. Computers & Industrial Engineering 52, 2007, 380-403.
- [162] Wang J., Xia Z.: *Scheduling jobs under decreasing linear deterioration*. Information Processing Letters 94, 2005, p.63-69.
- [163] Wang X., Cheng T.: *Heuristics for parallel machine scheduling with job class setups and delivery to multiple customers*. International Journal of Production Economics 119, 2009, p.199-206.
- [164] Wang X., Tang J., Yung K.: *Optimization of the multi-objective dynamic cell formation problem using a scatter search approach*. International Journal of Advanced Manufacturing Technology 44, 2009, 318-329.
- [165] Wei C., Wang J.: *Single machine quadratic penalty function scheduling with deteriorating jobs and group technology*. Applied Mathematical Modelling 34, 2010, p.3642-3647.
- [166] Wojakowski P.: *Metodyka szacowania okresu planowania z zastosowaniem technik eksploracji danych*. Czasopismo Techniczne 108, 2011, p. 523-529.
- [167] Wojakowski P.: *The idea of lead time estimation for multistage small-lot production systems based on MRP*. Monograph: Improvement of production processes, Łódź 2011, p.215-228.
- [168] Wojakowski P.: *Cell formation problem – state of the art*. Rozdział w monografii pod red. Marka Fertscha pt.: “Production Systems-Selected Issues-Theory and Practice”. Wydawnictwo Politechniki Poznańskiej, 2011, str. 69-84.
- [169] Wojakowski P.: *Przegląd literatury na temat projektowania struktury produkcyjnej*. Gospodarka Materiałowa i Logistyka, 11/2011, str. 56-59.
- [170] Won Y., Currie K.: *An effective p-median model considering production factors in machine cell/part family formation*. Journal of Manufacturing Systems 25, 2006, 58-64.
- [171] Won Y., Lee K.: *Modified p-median approach for efficient GT Cell formation*. Computers & Industrial Engineering 46, 2004, 495-510.
- [172] Wróblewski K.: *Podstawy sterowania przepływem produkcji*. WNT, Warszawa 1993.
- [173] Wu Ch., Lee W.: *Single-machine group scheduling problems with deteriorating setup times and job processing times*. International Journal of Production Economics 115, 2008, p.128-133.
- [174] Wu T., Chang C., Yeh J.: *A hybrid heuristic algorithm adopting both Boltzmann function and mutation operator for manufacturing cell formation problems*. International Journal of Production Economics 120, 2009, 669-688.
- [175] Wu T., Chung S., Chang C.: *A water flow-like algorithm for manufacturing cell formation problems*. European Journal of Operational Research 205, 2010, 346-360.

- [176] Wu X., Chu C., Wang Y., Yan W.: *A genetic algorithm for cellular manufacturing design and layout*. European Journal of Operational Research 181, 2007, 156-167.
- [177] Xing L., Chen Y., Wang P., Zhao Q., Xiong J.: *A knowledge-based ant colony optimization for flexible job-shop scheduling problems*. Applied Soft Computing 10, 2010, p.888-896.
- [178] Yazdani M., Amiri M., Zandieh M.: *Flexible job-shop scheduling with parallel variable neighborhood search algorithm*. Expert systems with Applications 37, 2010, p.678-687.
- [179] Yin Y., Yasuda K.: *Similarity coefficient methods applied to the cell formation problem: A taxonomy and review*. International Journal of Production Economics 101, 2006, 329-352.
- [180] Zaerpour N., Rabbani M., Gharehgozli A., Tavakkoli-Moghaddam R.: *Make-to-order or make-to-stock decision by novel hybrid approach*. Advanced Engineering Informatics 22, 2008, p.186-201.
- [181] Zäpfel G.: *Production planning in the case of uncertain individual demand. Extension for an MRP II concept*. International Journal of Production Economics 46-47, 1996, p.153-164.
- [182] Zeballos L., Novas J., Henning G.: *A CP formulation for scheduling multiproduct multistage batch plants*. Computers & Chemical Engineering 35, 2011, p.2973-2989.
- [183] Zhang Z., Lee M., Huang P., Zhang L., Huang X.: *A framework of ERP systems implementation success in China: An empirical study*. International Journal of Production Economics 98, 2005, p.56-80.
- [184] Zolfaghari S., Liang M.: *A new genetic algorithm for the machine/part grouping problem involving processing times and lot sizes*. Computers & Industrial Engineering 45, 2003, p.713-731.
- [185] Zorzini M., Corti D., Pozzetti A.: *Due date (DD) quotation and capacity planning in make-to-order companies: results from an empirical analysis*. International Journal of Production Economics 112, 2008, p.919-933.

Dodatek A. PRZYKŁAD ZASTOSOWANIA REGUŁY PRIORYTETU SPT DO SZEREGOWANIA ZADAŃ I OBIEKTÓW

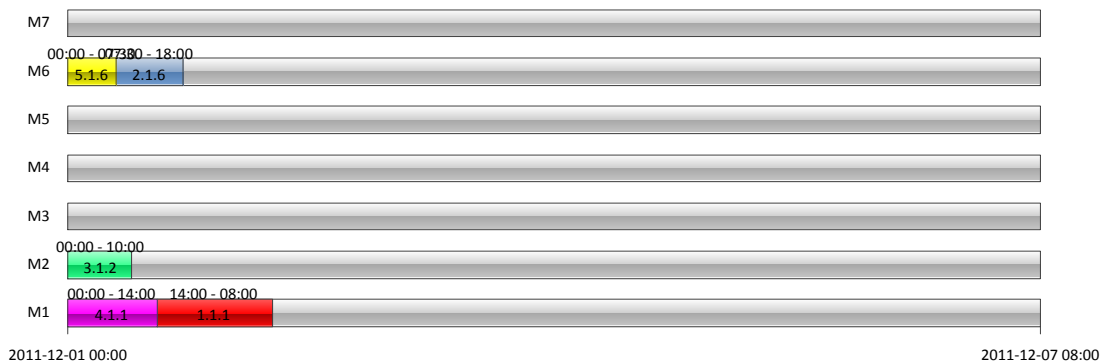
Niniejszy dodatek został przygotowany w celu rozwinięcia przykładu zaprezentowanego w rozdziale 4.1.1. W dodatku wydzielono dwie części. Pierwsza część szczegółowo opisuje proces szeregowania zadań i tworzenia harmonogramu zadań według klasycznej metody wykorzystującej regułę priorytetu SPT. Druga część szczegółowo opisuje proces kategoryzacji zadań do trzech grup zadań oraz ich dalszego zaszeregowania regułą SPT a następnie utworzenia harmonogramu, uwzględniając przepływ produkcji w wirtualnej komórce przedmiotowej powstałej w wyniku grupowania zleceń pokazanym na Rys. 17. W obydwu częściach, do szeregowania zadań i tworzenia harmonogramu wykorzystano dane zamieszczone w Tab. 5.

1. Zastosowanie reguły priorytetu SPT do szeregowania zadań

W pierwszej kolejności szeregowaniu poddano tylko pierwsze zadania z łańcuchów zadań poszczególnych zleceń produkcyjnych. Poniżej zamieszczono wyniki szeregowania:

SPT Zadania1							
Przed szeregowaniem		Po szeregowaniu		Maszyna	Kod	Start	Stop
Z1	1080	Z5	7:30	M6	5.1.6	01-01-12 0:00	01-01-12 7:30
Z2	630	Z3	10:00	M2	3.1.2	01-01-12 0:00	01-01-12 10:00
Z3	600	Z2	10:30	M6	2.1.6	01-01-12 7:30	01-01-12 18:00
Z4	840	Z4	14:00	M1	4.1.1	01-01-12 0:00	01-01-12 14:00
Z5	450	Z1	18:00	M1	1.1.1	01-01-12 14:00	02-01-12 8:00

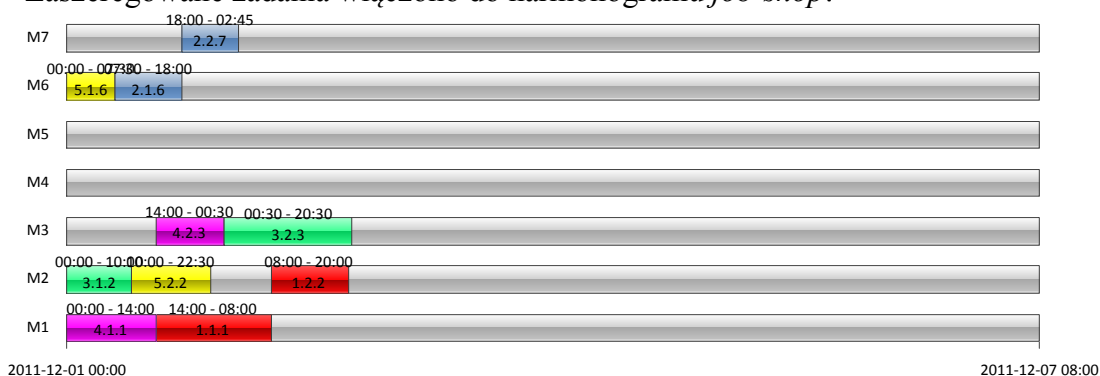
Zaszeregowane zadania włączono do harmonogramu *job-shop*:



Następnie szeregowaniu poddano drugie zadania z łańcuchów zadań poszczególnych zleceń produkcyjnych. Poniżej zamieszczono wyniki szeregowania:

SPT Zadania2							
Przed szeregowaniem		Po szeregowaniu		Maszyna	Kod	Start	Stop
Z1	720	Z2	8:45	M7	2.2.7	01-01-12 18:00	02-01-12 2:45
Z2	525	Z4	10:30	M3	4.2.3	01-01-12 14:00	02-01-12 0:30
Z3	1200	Z1	12:00	M2	1.2.2	02-01-12 8:00	02-01-12 20:00
Z4	630	Z5	12:30	M2	5.2.2	01-01-12 10:00	01-01-12 22:30
Z5	750	Z3	20:00	M3	3.2.3	02-01-12 0:30	02-01-12 20:30

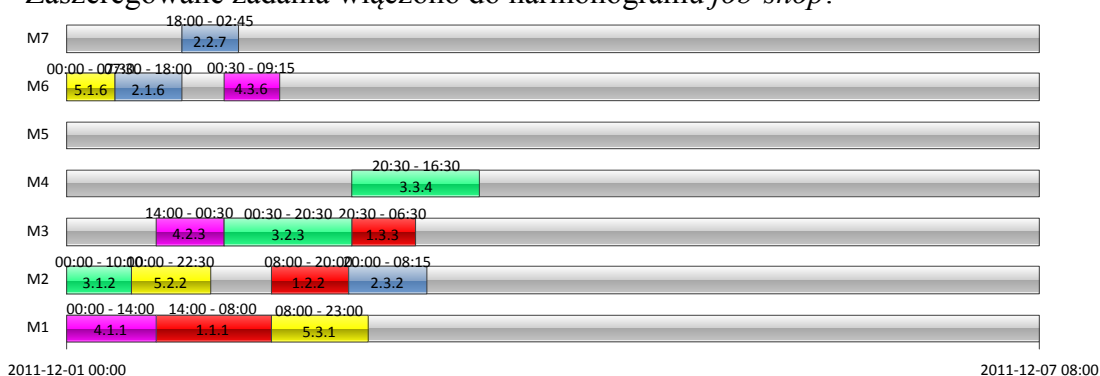
Zaszeregowane zadania włączono do harmonogramu *job-shop*:



Dalej szeregowaniu poddano trzecie zadania z łańcuchów zadań poszczególnych zleceń produkcyjnych. Poniżej zamieszczono wyniki szeregowania:

SPT Zadania3							
Przed szeregowa- niem		Po szeregowa- niu		Maszyna	Kod	Start	Stop
Z1	600	Z4	8:45	M6	4.3.6	02-01-12 0:30	02-01-12 9:15
Z2	735	Z1	10:00	M3	1.3.3	02-01-12 20:30	03-01-12 6:30
Z3	1200	Z2	12:15	M2	2.3.2	02-01-12 20:00	03-01-12 8:15
Z4	525	Z5	15:00	M1	5.3.1	02-01-12 8:00	02-01-12 23:00
Z5	900	Z3	20:00	M4	3.3.4	02-01-12 20:30	03-01-12 16:30

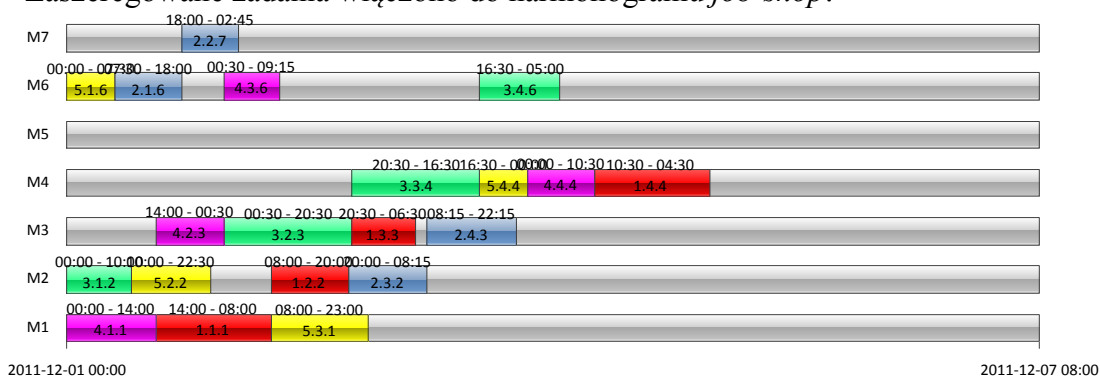
Zaszeregowane zadania włączono do harmonogramu *job-shop*:



Dalej szeregowaniu poddano czwarte zadania z łańcuchów zadań poszczególnych zleceń produkcyjnych. Poniżej zamieszczono wyniki szeregowania:

SPT Zadania4							
Przed szeregowa- niem		Po szeregowa- niu		Maszyna	Kod	Start	Stop
Z1	1080	Z5	7:30	M4	5.4.4	03-01-12 16:30	04-01-12 0:00
Z2	840	Z4	10:30	M4	4.4.4	04-01-12 0:00	04-01-12 10:30
Z3	750	Z3	12:30	M6	3.4.6	03-01-12 16:30	04-01-12 5:00
Z4	630	Z2	14:00	M3	2.4.3	03-01-12 8:15	03-01-12 22:15
Z5	450	Z1	18:00	M4	1.4.4	04-01-12 10:30	05-01-12 4:30

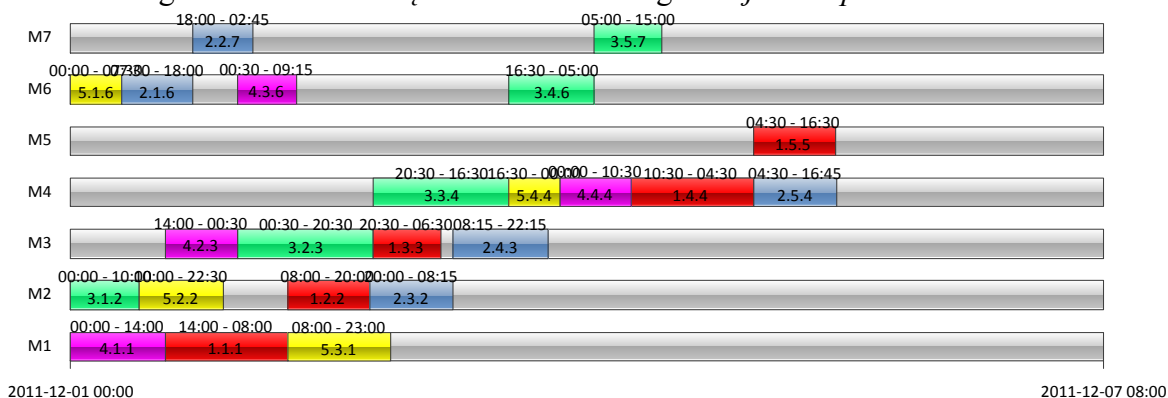
Zaszeregowane zadania włączono do harmonogramu *job-shop*:



Dalej szeregowaniu poddano piąte zadania z łańcuchów zadań trzech zleceń produkcyjnych. Poniżej zamieszczono wyniki szeregowania:

SPT Zadania5							
Przed szeregowaniem		Po szeregowaniu		Maszyna	Kod	Start	Stop
Z1	720	Z3	10:00	M7	3.5.7	04-01-12 5:00	04-01-12 15:00
Z2	735	Z1	12:00	M5	1.5.5	05-01-12 4:30	05-01-12 16:30
Z3	600	Z2	12:15	M4	2.5.4	05-01-12 4:30	05-01-12 16:45

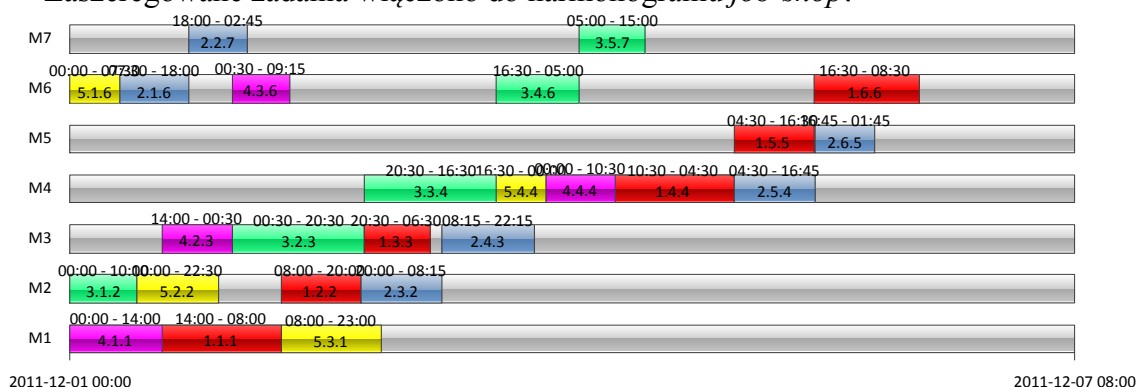
Zaszeregowane zadania włączono do harmonogramu *job-shop*:



Dalej szeregowaniu poddano szóste zadania z łańcuchów zadań dwóch pozostałych zleceń produkcyjnych. Poniżej zamieszczono wyniki szeregowania:

SPT Zadania6							
Przed szeregowaniem		Po szeregowaniu		Maszyna	Kod	Start	Stop
Z1	960	Z2	9:00	M5	2.6.5	05-01-12 16:45	10-01-12 1:45
Z2	630	Z1	16:00	M6	1.6.6	05-01-12 16:30	12-01-12 8:30

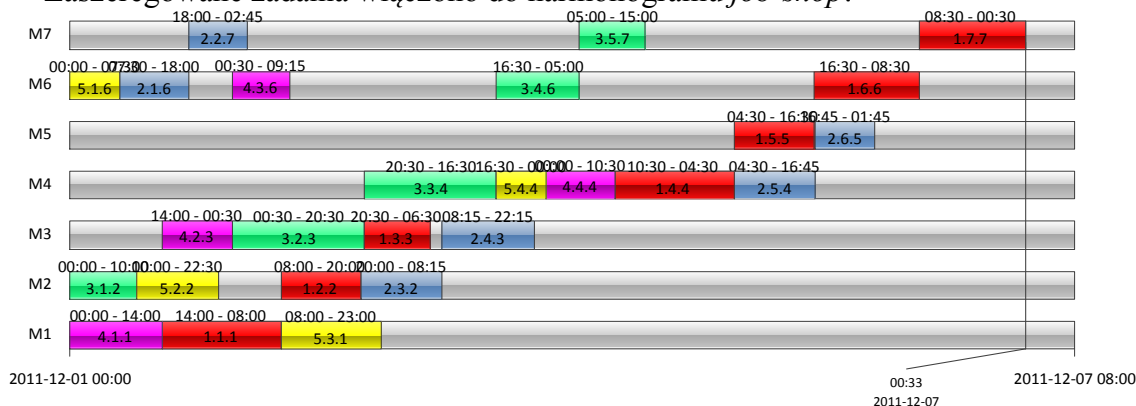
Zaszeregowane zadania włączono do harmonogramu *job-shop*:



Na końcu szeregowaniu poddano ostatnie siódme zadanie z łańcucha zadań pierwszego zlecenia produkcyjnego. Poniżej zamieszczono wyniki szeregowania:

SPT Zadania7							
Przed szeregowaniem	Po szeregowaniu	Maszyna	Kod	Start	Stop		
Z1	960	Z1	16:00	M7	1.7.7	12-01-12 8:30	13-01-12 0:30

Zaszeregowane zadania włączono do harmonogramu *job-shop*:

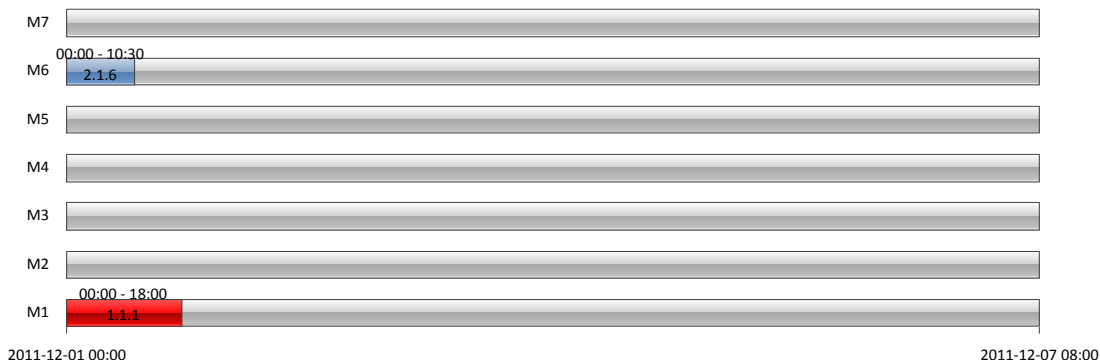


2. Zastosowanie reguły priorytetu SPT do szeregowania zadań z uwzględnieniem wirtualnych komórek przedmiotowych

W pierwszej kolejności szeregowaniu poddano zadania oznaczone numerem 1, znajdujących się na liście zadań *BeforeVMCJobs*. Poniżej zamieszczono wyniki szeregowania:

BeforeVMCJobs SPT Zadania1							
Przed szeregowaniem	Po szeregowaniu	Maszyna	Kod	Start	Stop		
Z1	1080	Z2	10:30	M6	2.1.6	01-01-12 0:00	01-01-12 10:30
Z2	630	Z1	18:00	M1	1.1.1	01-01-12 0:00	01-01-12 18:00

Zaszeregowane zadania włączono do harmonogramu *job-shop* z VMC:



Następnie szeregowaniu poddano zadanie oznaczone numerem 2, znajdujące się na liście zadań *BeforeVMCJobs*. Poniżej zamieszczono wyniki szeregowania:

BeforeVMCJobs SPT Zadania2							
Przed szeregowaniem	Po szeregowaniu	Maszyna	Kod	Start	Stop		
Z2	525	Z2	8:45	M7	2.2.7	01-01-12 10:30	01-01-12 19:15

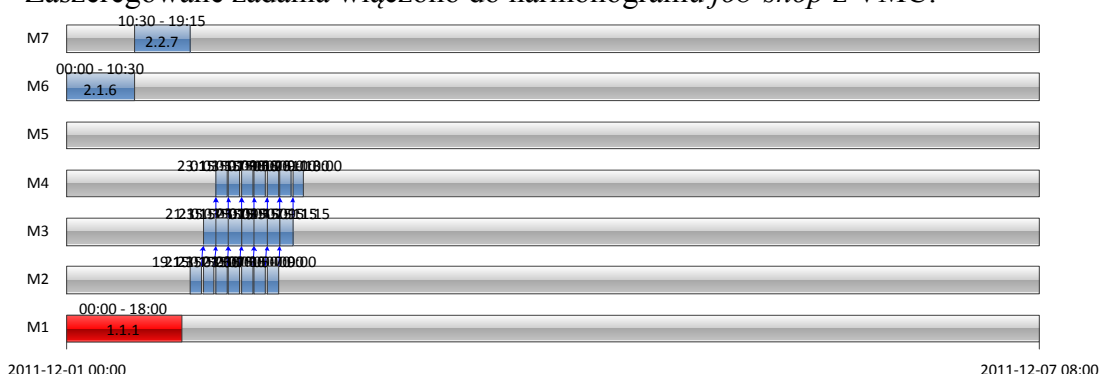
Zaszeregowane zadania włączono do harmonogramu *job-shop* z VMC:



W drugiej kolejności szeregowaniu poddano zadania składające się na najdłuższy wspólny podłańcuch zadań wykonywanych w wirtualnej komórce przedmiotowej (należą do listy *InsideVMCJobs*). Najpierw do harmonogramu włącza się zlecenie, które w ramach zleceń wykonywanych w wirtualnej komórce przedmiotowej ma najmniejsza wartość T_{jmax} . Poniżej zamieszczono wyniki szeregowania:

InsideVMCJobs SPT Parallel Z2							
Przed szeregowaniem		Po szeregowaniu		Maszyna	Kod	Start	Stop
Z2O3	105	Z2O3	1:45	M2	2.3.2	01-01-12 19:15	01-01-12 21:15
Z2O4	120	Z2O4	2:00	M3	2.4.3	01-01-12 21:15	01-01-12 23:15
Z2O5	105	Z2O5	1:45	M4	2.5.4	01-01-12 23:15	02-01-12 1:15
Tjmax	120	Last	12:00	M4	2.5.4	02-01-12 11:15	02-01-12 13:15

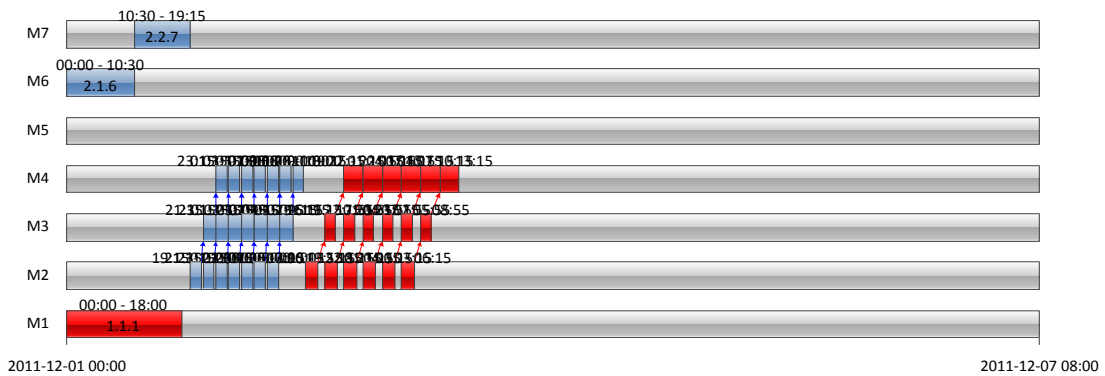
Zaszeregowane zadania włączono do harmonogramu *job-shop* z VMC:



Następnie z listy *InsideVMCJobs* dodano zadania należące do kolejnego zlecenia wykonywanego w wirtualnej komórce przedmiotowej. Poniżej zamieszczono wyniki szeregowania:

InsideVMCJobs SPT Parallel Z1							
Przed szeregowaniem		Po szeregowaniu		Maszyna	Kod	Start	Stop
Z1O2	120	Z1O2	2:00	M2	1.2.2	02-01-12 13:15	02-01-12 16:15
Z1O3	100	Z1O3	1:40	M3	1.3.3	02-01-12 16:15	02-01-12 19:15
Z1O4	180	Z1O4	3:00	M4	1.4.4	02-01-12 19:15	02-01-12 22:15
Tjmax	180	Last	15:00	M4	1.4.4	03-01-12 10:15	03-01-12 13:15

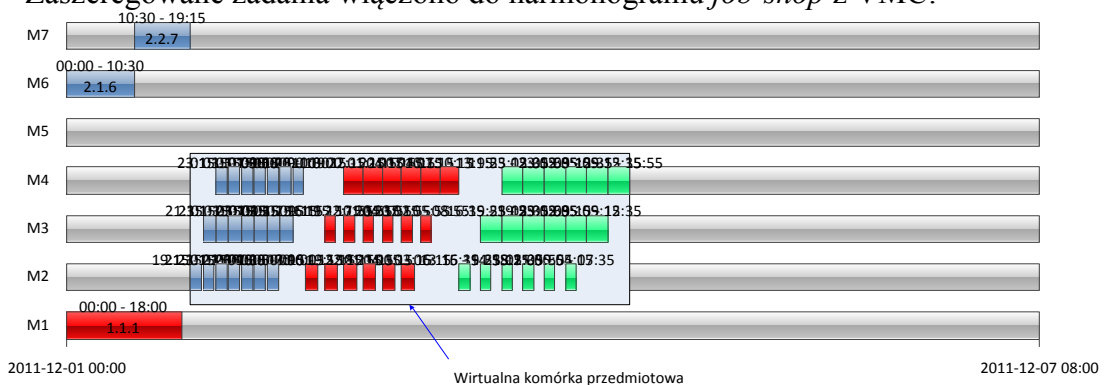
Zaszeregowane zadania włączono do harmonogramu *job-shop* z VMC:



Na końcu z listy *InsideVMCJobs* dodano zadania należące do ostatniego zlecenia wykonywanego w wirtualnej komórce przedmiotowej. Poniżej zamieszczono wyniki szeregowania:

InsideVMCJobs SPT Paralel Z3							
Przed szeregowaniem		Po szeregowaniu		Maszyna	Kod	Start	Stop
Z3O1	100	Z3O1	1:40	M2	3.1.2	03-01-12 13:15	03-01-12 16:35
Z3O2	200	Z3O2	3:20	M3	3.2.3	03-01-12 16:35	03-01-12 19:55
Z3O3	200	Z3O3	3:20	M4	3.3.4	03-01-12 19:55	03-01-12 23:15
Tjmax	200	Last	16:40	M4	3.3.4	04-01-12 12:35	04-01-12 15:55

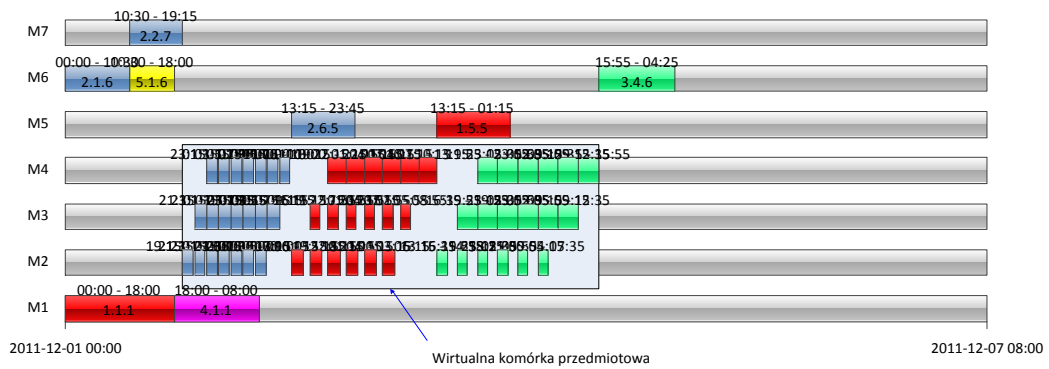
Zaszeregowane zadania włączono do harmonogramu *job-shop* z VMC:



Dalej z listy *ApartVMCJobs* dodano pierwsze zadania. Poniżej zamieszczono wyniki szeregowania:

ApartVMCJobs SPT Zadania1							
Przed szeregowaniem		Po szeregowaniu		Maszyna	Kod	Start	Stop
Z1	720	Z5	7:30	M6	5.1.6	12-01-01 10:30	12-01-01 18:00
Z2	630	Z2	10:30	M5	2.x1.5	12-01-02 13:15	12-01-02 23:45
Z3	750	Z1	12:00	M5	1.x1.5	12-01-03 13:15	12-01-04 1:15
Z4	840	Z3	12:30	M6	3.x1.6	12-01-04 15:55	12-01-05 4:25
Z5	450	Z4	14:00	M1	4.1.1	12-01-01 18:00	12-01-02 8:00

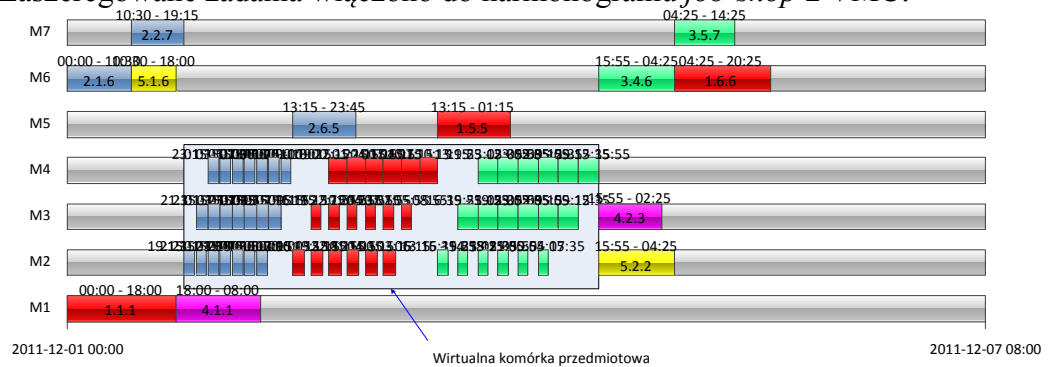
Zaszeregowane zadania włączono do harmonogramu *job-shop* z VMC:



Dalej z listy *ApartVMCJobs* dodano drugie zadania. Poniżej zamieszczono wyniki szeregowania:

ApartVMCJobs SPT Zadania2							
Przed szeregowaniem		Po szeregowaniu		Maszyna	Kod	Start	Stop
Z1	960	Z3	10:00	M7	3.x2.7	12-01-05 4:25	12-01-05 14:25
Z3	600	Z4	10:30	M3	4.2.3	12-01-04 15:55	12-01-05 2:25
Z4	630	Z5	12:30	M2	5.2.2	12-01-04 15:55	12-01-05 4:25
Z5	750	Z1	16:00	M6	1.x2.6	12-01-05 4:25	12-01-05 20:25

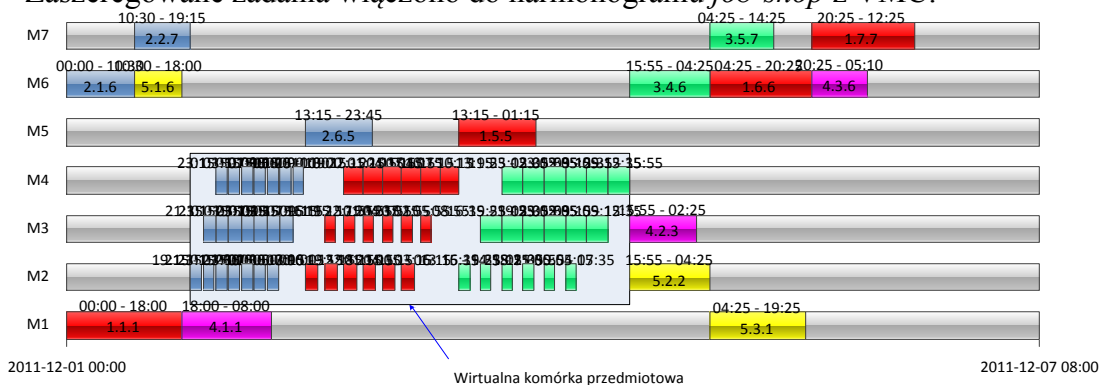
Zaszeregowane zadania włączono do harmonogramu *job-shop* z VMC:



Dalej z listy *ApartVMCJobs* dodano trzecie zadania. Poniżej zamieszczono wyniki szeregowania:

ApartVMCJobs SPT Zadania3							
Przed szeregowaniem		Po szeregowaniu		Maszyna	Kod	Start	Stop
Z1	960	Z4	8:45	M6	4.3.6	12-01-05 20:25	12-01-06 5:10
Z4	525	Z5	15:00	M1	5.3.1	12-01-05 4:25	12-01-05 19:25
Z5	900	Z1	16:00	M7	1.x3.7	12-01-05 20:25	12-01-06 12:25

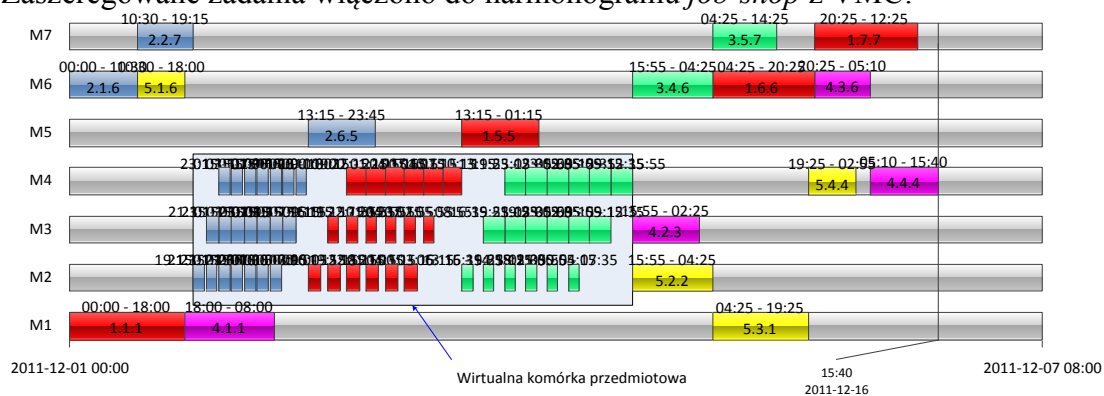
Zaszeregowane zadania włączono do harmonogramu *job-shop* z VMC:



Dalej z listy *ApartVMCJobs* dodano ostatnie, czwarte zadania. Poniżej zamieszczono wyniki szeregowania:

ApartVMCJobs SPT Zadania4							
Przed szeregowaniem		Po szeregowaniu		Maszyna	Kod	Start	Stop
Z3	750	Z5	7:30	M4	5.4.4	12-01-06 12:25	12-01-06 19:55
Z4	630	Z4	10:30	M4	4.4.4	12-01-06 5:10	12-01-06 15:40

Zaszeregowane zadania włączono do harmonogramu *job-shop* z VMC:



**Dodatek B. INFORMACJE O SYSTEMIE WYTWÓRCZYM WYKORZYSTANYM
JAKO PRZYKŁAD DO TESTOWANIA DZIAŁANIA ALGORYTMU PRODUFLOW
PM**

1. Zestawienie maszyn rozmieszczonych w komórce produkcyjnej II stopnia

Lp	Identyfikator maszyny	Godzinowy koszt pracy maszyny	Wymiary	Identyfikator typu technologicznego maszyny	Nazwa typu technologicznego maszyny
1	FNC25-1	5.5000	2000x2150x1900	FV	Frezarka pionowa
2	FNC25-2	5.5000	2000x2150x1900	FV	Frezarka pionowa
3	FNC25-3	5.2000	2000x2150x1900	FV	Frezarka pionowa
4	FNC25-4	5.2000	2000x2150x1900	FV	Frezarka pionowa
5	FNC25-5	5.1000	2000x2150x1900	FV	Frezarka pionowa
6	FNC25-6	5.1000	2000x2150x1900	FV	Frezarka pionowa
7	FWA32-1	7.5000	3000x2000x1800	FH	Frezarka pozioma
8	FWA32-2	7.2500	3000x2000x1800	FH	Frezarka pozioma
9	FWA32-3	7.2500	3000x2000x1800	FH	Frezarka pozioma
10	NUA25-1	6.0000	1350x1250x1250	SN	Szlifierka narzędziowa
11	PHW12-1	8.5000	1200x1200x2500	P	Prasa hydrauliczna wysięgowa
12	PHW12-2	8.5000	1200x1200x2500	P	Prasa hydrauliczna wysięgowa
13	PHW12-3	8.5000	1200x1200x2500	P	Prasa hydrauliczna wysięgowa
14	PM120-1	2.4000	1350x1100x1150	PM	Piła ramowa
15	RH25-1	6.5000	1600x900x1200	TR	Tokarka rewolwerowa
16	RNA14-1	7.2000	1300x600x1200	TR	Tokarka rewolwerowa
17	RNA14-2	7.2000	1300x600x1200	TR	Tokarka rewolwerowa
18	RVA40-1	9.6000	2400x1400x1500	TR	Tokarka rewolwerowa
19	SAC30-1	12.5000	4000x1750x2400	SV	Szlifierka do płaszczyzn z osią pionową
20	SAC30-2	12.5000	4000x1750x2400	SV	Szlifierka do płaszczyzn z osią pionową
21	SBB75-1	18.5000	2500x1700x1750	SB	Szlifierka bezkłowa
22	SOJ10-1	9.0000	2500x1500x1750	SO	Szlifierka bezkłowa
23	SOJ10-2	9.0000	2500x1500x1750	SO	Szlifierka bezkłowa
24	SOJ10-3	9.0000	2500x1500x1750	SO	Szlifierka bezkłowa
25	SPC20-1	7.8000	1200x1400x1600	SH	Szlifierka do płaszczyzn z osią poziomą
26	SPD30-1	8.8000	4000x1950x2300	SH	Szlifierka do płaszczyzn z osią poziomą
27	SWA10-1	11.3000	2050x1250x1550	S	Szlifierka do wałków
28	SWA10-2	11.3000	2050x1250x1550	S	Szlifierka do wałków
29	SWA10-3	13.4000	2050x1250x1550	S	Szlifierka do wałków
30	SWA25-1	16.5000	3500x1850x1800	S	Szlifierka do wałków
31	SWA25-2	16.5000	3500x1850x1800	S	Szlifierka do wałków
32	SWA25-3	15.9000	3500x1850x1800	S	Szlifierka do wałków
33	SWA25-4	15.9000	3500x1850x1800	S	Szlifierka do wałków
34	SWA25-5	14.0000	3500x1850x1800	S	Szlifierka do wałków
35	SWA25-6	14.0000	3500x1850x1800	S	Szlifierka do wałków
36	SWB25-1	14.0000	3500x1850x1500	S	Szlifierka do wałków
37	SWB25-2	14.0000	3500x1850x1500	S	Szlifierka do wałków

38	SWB25-3	13.3000	3500x1850x1500	S	Szlifierka do wałków
39	SWB25-4	13.3000	3500x1850x1500	S	Szlifierka do wałków
40	TGC8-1	18.5000	2150x1100x1800	TG	Tokarka kopiarka
41	TR10C1N-1	15.2000	3800x1950x1350	TP	Tokarka produkcyjna
42	TR10C1N-2	15.2000	3800x1950x1350	TP	Tokarka produkcyjna
43	TR10C1N-3	16.7000	3800x1950x1350	TP	Tokarka produkcyjna
44	TUB32-1	6.2000	2750x900x1250	TU	Tokarka uniwersalna
45	TUB32-2	6.2000	2750x900x1250	TU	Tokarka uniwersalna
46	TUB32-3	6.2000	2750x900x1250	TU	Tokarka uniwersalna
47	TUD40-1	7.5000	3200x1250x1600	TU	Tokarka uniwersalna
48	TUD40-2	7.5000	3200x1250x1600	TU	Tokarka uniwersalna
49	TUD40-3	8.0000	3200x1250x1600	TU	Tokarka uniwersalna
50	TUD40-4	8.0000	3200x1250x1600	TU	Tokarka uniwersalna
51	TUD40-5	8.5000	3200x1250x1600	TU	Tokarka uniwersalna
52	TUD40-6	8.5000	3200x1250x1600	TU	Tokarka uniwersalna
53	TUD50-1	7.3000	3700x1250x1650	TU	Tokarka uniwersalna
54	TUD50-2	7.3000	3700x1250x1650	TU	Tokarka uniwersalna
55	TUD50-3	7.3000	3700x1250x1650	TU	Tokarka uniwersalna
56	TUD50-4	8.9000	3700x1250x1650	TU	Tokarka uniwersalna
57	TUD50-5	8.9000	3700x1250x1650	TU	Tokarka uniwersalna
58	TUM25-1	5.5000	2100x800x1180	TU	Tokarka uniwersalna
59	TUM25-2	5.5000	2100x800x1180	TU	Tokarka uniwersalna
60	TUM25-3	5.5000	2100x800x1180	TU	Tokarka uniwersalna
61	TUM25-4	5.5000	2100x800x1180	TU	Tokarka uniwersalna
62	TWC8-1	11.9000	2150x1100x1700	TW	Tokarka wielonożowa
63	WCC80-1	6.0000	1000x600x1000	WC	Wiertarka centrówka
64	WCC80-2	6.0000	1000x600x1000	WC	Wiertarka centrówka
65	WKA40-1	5.1000	1600x850x3200	WK	Wiertarka kadłubowa
66	WKA40-2	5.1000	1600x850x3200	WK	Wiertarka kadłubowa
67	WR50-1	5.0000	2800x1300x3300	WR	Wiertarka promieniowa
68	WR50-2	5.0000	2800x1300x3300	WR	Wiertarka promieniowa
69	WR50-3	5.0000	2800x1300x3300	WR	Wiertarka promieniowa
70	WR50-4	5.0000	2800x1300x3300	WR	Wiertarka promieniowa
71	WS15-1	2.0000	750x550x750	WS	Wiertarka stołowa
72	WS15-2	2.0000	750x550x750	WS	Wiertarka stołowa
73	ZFC20-1	17.0000	1950x1250x2000	ZC	Frezarka obwiedniowa do uzębień
74	ZFTK-1	19.0000	2500x2000x1800	ZG	Strugarka Gleasona do uzębień

2. Wykaz marszrut technologicznych wyrobów przepływających przez komórkę produkcyjną II stopnia WOSK

Lp.	Nazwa i numer części	Typy technologiczne maszyn																												
		TUM25	TUD40	TUD50	TUB32	TR10CIN	ZFC20	SOJ10	SAC30	ZFTK	WR50	FNC25	NUA25	WS15	SWA25	SWB25	SWA10	FWA32	SPC20	SPD30	TWC8	SBB75	TGC8	PM120	WKA40	RNA14	RH25	WCC80	PHW12	RVA40
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	Sworzeń																					30/5 0,3/0,01				10/4 0,58/0,01 20/4 0,58/0,01				
2	Sworzeń											20/4 0,4/0,03 30/4 0,4/0,04 40/4 0,4/0,2										50/5 0,25/0,01			10/4 0,6/0,01					
3	Sworzeń											20/4 0,4/0,02										30/5 0,3/0,01			10/4 0,75/0,02					
4	Sworzeń											30/4 0,36/0,03										40/5 0,3/0,01			10/4 0,59/0,02 20/4 0,59/0,01					
5	Sworzeń											30/4 0,38/0,02 40/4 0,38/0,01										50/5 0,3/0,01			10/4 0,59/0,02 20/4 0,59/0,01					
6	Sworzeń	20/4 0,4/0,02										30/4 0,4/0,02										40/5 0,3/0,01			10/4 0,75/0,01					
7	Sworzeń											20/4 0,4/0,03		30/4 0,25/0,04			50/5 0,15/0,01					40/5 0,3/0,04			10/4 0,6/0,01					
8	Wkręt											30/4 0,4/0,03		20/4 0,4/0,01											10/4 0,6/0,01					
9	Sworzeń											20/4 0,4/0,02 30/4 0,4/0,03		40/4 0,4/0,02 50/4 0,4/0,02								60/5 0,3/0,01			10/4 0,9/0,03					
10	Sworzeń	20/4 0,4/0,02										30/5 0,5/0,02 40/4 0,5/0,04 50/5 0,4/0,03		60/4 0,4/0,02											10/4 0,9/0,02					
11	Trzpień		20/4 0,4/0,07									30/4 0,4/0,04 40/4 0,4/0,06 50/4 0,4/0,03		60/4 0,4/0,1								70/6 0,5/0,06			10/4 0,4/0,02					
12	Wał z gwintem		30/5 0,4/0,05 40/5 0,4/0,13									50/5 0,4/0,05										20/5 0,7/0,02			10/4 0,3/0,03					
13	Wał		20/5 0,36/0,2									30/6 0,4/0,13				40/7 0,46/0,48									10/4 0,3/0,02					
14	Wał z gwintem		30/6 0,4/0,33 60/7 0,4/0,35	20/5 0,4/0,25												50/6 0,5/0,06									10/4 0,3/0,03					
15	Wał z gwintem		30/7 1,5/0,2	20/5 0,4/0,28 40/5 0,5/0,08								50/5 0,5/0,05				60/6 0,3/0,09									10/4 0,5/0,08	70/8 1,5/1,15				
16	Wał z gwintem		50/7 1,5/0,45	20/5 0,4/0,1 30/5 0,4/0,11 40/5 0,4/0,25												60/7 0,5/0,29									10/4 0,5/0,58	70/8 1,5/1,9				
17	Wał z gwintem		60/7 1,5/0,18	20/5 0,4/0,12								30/5 0,3/0,02 40/5 0,5/0,01				50/6 0,5/0,2									10/4 0,3/0,03					
18	Wał z gwintem	20/5 0,4/0,55	50/9 0,5/0,15									30/5 0,5/0,04		40/5 0,4/0,04		60/6 0,5/0,15										10/4 0,3/0,03				
19	Wał		20/4 0,4/0,06									30/5 0,5/0,04				40/7 0,5/0,18										10/4 0,4/0,03				
20	Wał			20/4 0,4/0,15								40/5 0,5/0,5				50/6 0,5/0,45									30/4 0,4/0,11		10/4 0,4/0,02			
21	Wał			20/5 0,4/0,06								40/5 0,5/0,03				50/7 0,45/0,25									30/4 0,45/0,07		10/4 0,4/0,01			
22	Wrzeciono		90/5 0,5/0,1	20/5 0,5/0,33 30/5 0,5/0,25 40/5 0,5/0,3 50/5 0,5/0,2												70/7 0,6/0,12 100/8 0,6/0,11 110/8 0,6/0,2	80/7 0,6/0,2								60/4 0,4/0,2		10/5 0,12/0,01			

Lp.	Nazwa i numer części	Stanowiska robocze																															
		TUM25	TUD40	TUD50	TUB32	TRI0CIN	ZFC20	SOJ10	SAC30	ZFTK	WR50	FNC25	NUA25	WS15	SWA25	SWB25	SWA10	FWA32	SPC20	SPD30	TWC8	SBB75	TGC8	PM120	WKA40	RNA14	RH25	WCC80	PHW12	RVA40			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
23	Wrzeciono		40/6 0,4/0,08 60/5 0,34/0,08 70/5 0,3/0,07 100/5 0,3/0,1 170/5 0,5/0,2	20/5 0,5/0,1 30/5 0,5/0,1 50/5 0,4/0,28 80/5 0,5/0,1 90/5 0,5/0,1								120/5 0,27/0,1 130/5 0,4/0,1			140/7 0,6/0,1 150/7 0,6/0,1 160/7 0,6/0,1 180/8 0,6/0,1 190/8 0,6/0,1 200/8 0,6/0,1										110/5 0,6/0,17			10/5 0,12/0,05					
24	Wrzeciono		20/5 0,5/0,03 30/5 0,4/0,06 50/5 0,35/0,02 60/6 0,4/0,13 130/6 0,28/0,1 150/5 0,5/0,1	40/5 0,4/0,14 80/5 0,5/0,06								70/5 0,4/0,1 110/5 0,28/0,12			120/7 0,5/0,14 140/7 0,6/0,1 160/8 0,6/0,1 170/8 0,5/0,1			100/5 0,7/0,12							90/5 0,6/0,12			10/4 0,5/0,02					
25	Wrzeciono		130/5 0,5/0,2	20/6 0,4/0,1 30/5 0,4/0,18 50/5 0,36/0,1								60/5 0,4/0,1			100/7 0,5/0,15 110/7 0,6/0,03 120/7 0,6/0,2 140/8 0,6/0,05 150/8 0,6/0,2			90/5 0,75/0,54							70/5 0,6/0,23 80/4 0,4/0,01			10/5 0,33/0,08					
26	Wrzeciono		20/6 0,4/0,13 40/5 0,3/0,07 50/5 0,36/0,06 60/6 0,4/0,1 70/5 0,3/0,1 110/5 0,5/0,04	30/5 0,46/0,16								80/5 0,4/0,06			90/7 0,6/0,13 100/6 0,6/0,07 120/8 0,6/0,07 130/8 0,5/0,11												10/5 0,3/0,02						
27	Wrzeciono		20/5 0,5/0,04 30/5 0,5/0,1 80/5 0,5/0,1									40/5 0,5/0,05 50/5 0,25/0,04			100/8 0,6/0,02			60/6 0,6/0,05 70/6 0,6/0,5 90/8 0,6/0,05										10/5 0,3/0,02					
28	Wrzeciono		20/5 0,5/0,02 30/5 0,4/0,2 40/6 0,4/0,1 100/5 0,5/0,18				60/7 0,6/0,12 70/7 0,2/0,01 120/8 0,6/0,12					50/5 0,25/0,25			110/8 0,6/0,04			80/7 0,6/0,05 90/7 0,6/0,2										10/5 0,2/0,02					
29	Wrzeciono		20/6 0,4/0,2 40/5 0,3/0,1 50/6 0,4/0,14 60/6 0,4/0,11	30/5 0,4/0,3 90/5 0,5/0,1 100/6 0,4/0,08								70/5 0,26/0,1 80/5 0,5/0,04			110/7 0,6/0,05 120/7 0,6/0,01 130/7 0,5/0,01 140/8 0,5/0,14													10/5 0,3/0,01					
30	Wał z gwintem			20/5 0,4/0,26 30/7 1,5/0,5 40/5 0,3/0,08	50/7 0,4/0,3 70/7 0,2/0,3							60/5 0,3/0,05																10/4 0,5/0,05					
31	Wał		20/4 0,4/0,08 30/4 0,4/0,05	40/5 0,4/0,6 70/5 0,4/0,13								50/5 0,3/0,3 60/5 0,3/0,1			80/7 0,5/0,1 90/7 0,5/0,5													10/4 0,3/0,02					
32	Tuleja sprzęgłowa			10/4 0,4/0,25 20/6 0,4/0,4 30/7 0,4/0,3								40/6 0,4/0,2													60/5 0,2/0,1				50/6 0,3/0,05				
33	Koło pasowe		10/6 0,4/0,42	20/7 0,4/0,58																				30/6 0,2/0,09				40/6 0,3/0,07					

Lp.	Nazwa i numer części	Stanowiska robocze																												
		TUM25	TUD40	TUD50	TUB32	TR10C1N	ZFC20	SOJ10	SAC30	ZFTK	WR50	FNC25	NUA25	WS15	SWA25	SWB25	SWA10	FWA32	SPC20	SPD30	TWC8	SBB75	TGC8	PM120	WKA40	RNA14	RH25	WCC80	PHW12	RVA40
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
34	Płyta dociskowa			10/5 0,4/0,11 20/6 0,4/0,09																					40/5 0,2/0,12			30/6 0,3/0,06		
35	Tarcza prawa			10/6 0,4/0,37 20/6 0,4/0,2																					30/5 0,2/0,18 40/6 0,2/0,12			50/6 0,3/0,15		
36	Dźwignia											10/5 0,3/0,19 20/5 0,3/0,05 50/5 0,3/0,07													30/6 0,24/0,18 60/6 0,2/0,03			40/6 0,3/0,04		
37	Obudowa sprężyn			10/6 0,4/0,18 20/6 0,5/0,12	30/7 0,4/0,08																				50/6 0,2/0,15			40/6 0,3/0,1		
38	Dźwignia				10/7 0,3/0,2																				20/6 0,23/0,05 40/6 0,2/0,03			30/6 0,3/0,04		
39	Pokrywa										40/5 0,2/0,43	10/5 0,4/0,15 30/5 0,4/0,1													20/6 0,2/0,16					
40	Trzpień			20/4 0,4/0,03								40/4 0,4/0,03 50/5 0,25/0,06		70/7 0,5/0,2											60/4 0,4/0,01		10/4 0,4/0,01	30/4 0,6/0,15		
41	Wrzeciono		20/5 0,4/0,1 30/6 0,4/0,1 40/5 0,4/0,05	50/5 0,5/0,02 80/5 0,5/0,08			90/7 0,6/0,08 130/8 0,6/0,08					60/5 0,25/0,02 70/5 0,5/0,1					100/6 0,5/0,03 110/6 0,6/0,04 120/8 0,6/0,07											10/4 0,3/0,02		
42	Trzpień		20/4 0,4/0,02									40/4 0,4/0,02 50/5 0,25/0,08					60/6 0,4/0,1									30/4 0,6/0,03	10/4 0,4/0,02			
43	Trzpień		20/5 0,4/0,04									40/4 0,4/0,06 40/4 0,4/0,03 50/5 0,25/0,04				60/6 0,5/0,06											10/4 0,4/0,01			
44	Wrzeciono			20/5 0,4/0,2 30/6 0,4/0,12 40/5 0,4/0,01 50/5 0,5/0,1 120/5 0,5/0,03						90/4 0,15/0,01		70/5 0,25/0,25 80/5 0,25/0,06					100/6 0,6/0,05 110/6 0,6/0,05 130/8 0,6/0,05 140/8 0,6/0,06							60/4 0,4/0,08			10/4 0,4/0,3			
45	Trzpień		20/4 0,4/0,05									40/5 0,5/0,03		50/7 0,5/0,2			30/5 0,5/0,05										10/4 0,4/0,04			
46	Wał		20/4 0,4/0,07 30/5 0,4/0,2													50/7 0,6/0,9		40/6 0,5/0,23										10/4 0,4/0,05		
47	Wał		20/4 0,4/0,08 30/5 0,4/0,24												50/7 0,6/1,0		40/6 0,5/0,25											10/4 0,4/0,05		
48	Dźwignia sprężynowa				30/8 0,4/0,5						40/6 0,2/0,18 50/6 0,2/0,1	10/7 0,4/0,3 20/6 0,4/0,16																		
49	Łącznik										20/6 0,2/0,14 30/6 0,2/0,08 40/6 0,2/0,06 50/6 0,2/0,07	10/6 0,4/0,12																		
50	Dźwignia regulacji										30/6 0,2/0,16 40/6 0,2/0,04 50/6 0,2/0,08	10/6 0,4/0,06 20/6 0,4/0,04																		

Lp.	Nazwa i numer części	Stanowiska robocze																														
		TUM25	TUD40	TUD50	TUB32	TR10C1N	ZFC20	SOJ10	SAC30	ZFTK	WR50	FNC25	NUA25	WS15	SWA25	SWB25	SWA10	FWA32	SPC20	SPD30	TWC8	SBB75	TGC8	PM120	WKA40	RNA14	RH25	WCC80	PHW12	RVA40		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
51	Dźwignia napędowa										40/5 0,2/0,18 50/6 0,2/0,1							10/5 0,4/0,25 20/5 0,4/0,25 30/6 0,4/0,21														
52	Dźwignia								30/6 0,3/0,05		20/5 0,2/0,45 40/5 0,2/0,03	30/5 0,3/0,06						10/5 0,3/0,11														
53	Nakrętka posuwu				20/7 0,5/0,44 40/8 0,3/0,39						60/5 0,2/0,03	30/5 0,3/0,27 50/5 0,4/0,08												10/4 0,15/0,05								
54	Ślimak	20/5 0,4/0,15 30/7 0,4/0,4 60/8 0,4/0,5										40/5 0,4/0,15					50/8 0,5/0,4												10/4 0,4/0,05			
55	Tłoczysko	20/5 0,4/0,18 30/6 0,4/0,05 50/7 0,4/0,12										40/5 0,4/0,15					60/8 0,5/0,3 5												10/4 0,4/0,05			
56	Korpus pompki	20/6 0,5/0,46									40/5 0,2/0,12				50/7 0,4/0,23 60/7 0,45/0,12			30/6 0,4/0,12							10/4 0,15/0,07							
57	Podstawa	10/6 0,4/0,22 20/6 0,4/0,2 40/7 0,4/0,3												50/6 0,2/0,2 1				30/5 0,4/0,15														
58	Tarcza lewa	10/6 0,4/0,28 20/6 0,4/0,18	30/7 0,4/0,15											50/6 0,2/0,2 3				40/6 0,4/0,2														
59	Korbowód		20/7 0,4/0,35 30/7 0,4/0,45 40/7 0,4/0,5									10/6 0,4/0,36		50/6 0,3/0,1																		
60	Sworzeń				20/6 0,4/0,1							30/5 0,4/0,1 40/5 0,4/0,06		50/5 0,2/0,0 5			60/7 0,5/0,1										10/5 0,5/0,18					
61	Wodzik				10/5 0,4/0,16 20/6 0,4/0,2 40/7 0,4/0,26							30/6 0,4/0,2		50/6 0,2/0,1 6			60/8 0,5/0,3															
62	Walek z mimośrodem			10/5 0,4/0,2	20/7 0,4/0,5 30/7 0,4/0,4						40/5 0,4/0,12			60/8 0,5/0,2		50/8 0,5/0,4																
63	Walek z mimośrodem			10/5 0,4/0,2	20/7 0,4/0,59 30/7 0,4/0,4									60/8 0,5/0,2		50/8 0,5/0,4	40/7 0,5/0,42															
64	Tuleja															30/8 0,5/0,0 8	40/6 0,4/0,13										20/4 0,4/0,21				10/4 0,4/0,17	
65	Skala			10/7 0,4/0,3 20/7 0,4/0,28										50/5 0,2/0,0 8				30/6 0,4/0,38 40/6 0,4/0,21														
66	Sworzeń	20/5 0,4/0,5 30/5 0,4/0,3 40/6 0,4/0,05 70/6 0,6/0,1																50/7 0,4/0,02 60/7 0,5/0,03													10/4 0,6/0,02	
67	Sworzeń	20/4 0,4/0,02										30/5 0,26/0,02													40/6 0,5/0,04						10/5 0,6/0,05	
68	Sworzeń	20/5 0,4/0,02										30/5 0,3/0,03	40/4 0,4/0,01												50/6 0,46/0,08						10/5 0,9/0,07	

Lp.	Nazwa i numer części	Stanowiska robocze																														
		TUM25	TUD40	TUD50	TUB32	TR10C1N	ZFC20	SOJ10	SAC30	ZFTK	WR50	FNC25	NUA25	WS15	SWA25	SWB25	SWA10	FWA32	SPC20	SPD30	TWC8	SBB75	TGC8	PM120	WKA40	RNA14	RH25	WCC80	PHW12	RVA40		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
69	Wrzeciono			30/5 0,35/0,1	110 /6 0,5/ 0,15	20/5 1,0/0,2 40/5 1,0/0,1		80/7 0,6/0,1				60/5 0,3/0, 03				90/7 0,6/0,04 100/7 0,6/0,09 120/8 0,6/0,04 130/8 0,6/0,08		50/5 0,75/0,32 60/5 0,5/0,04							70/5 0,05/0,1			10/4 0,3/0,05				
70	Sworzeń	20/4 0,4/0,01										30/5 0,4/0, 05		40/5 0,15/0,01					30/7 0,2/0,01								10/4 0,6/0,01					
71	Koło zębate walcowe wielowienicowe								20/4 0,3/0,08					40/3 0,5/0,05				30/7 0,9/1,29												10/4 1/0,15		
72	Koło zębate		10/4 1/0,3 20/4 0,5/0,1				60/6 1,5/1,5				30/4 0,5/0,1																		40/4 0,4/0,05 50/4 0,4/0,05			
73	Koło zębate z wyłączeniem		10/4 0,4/0,2 20/4 0,5/0,1				50/6 0,5/1,2																							40/4 0,4/0,05 40/4 0,4/0,05		
74	Koło zębate płaskie		10/4 0,4/0,2 20/4 0,3/0,1				50/6 0,8/1,2																							30/4 0,4/0,05 40/4 0,4/0,05		
75	Koło zębate płaskie	10/4 0,45/0,2 20,4 0,25/0,2					50/6 0,5/0,4							40,4 0,5/0,1																30/4 0,24/0,05		
76	Koło zębate z piastą	10/4 0,45/0,3 20/4 0,3/0,2					40/6 0,3/0,4											50/5 0,4/0,35													30/4 0,4/0,05	
77	Koło zębate z piastą jednostronną		30/4 0,5/0,18	10/4 0,4/0,1 20/4 0,5/0,1			40/6 1/0,7																							30/4 0,3/0,05 50/5 0,2/0,01		
78	Koło zębate płaskie						30/6 0,9/0,1																							20/4 0,4/0,01	10/4 0,9/0,1	
79	Koło zębate z piastą jednostronną	40/4 0,4/0,03	10/4 0,4/0,05 20,4 0,5/0,03				70/6 0,8/0,6							60/4 0,5/0,03					50/5 0,3/0,03											30/4 0,17/0,02 80/4 0,15/0,01		
80	Koło zębate stożkowe			10/5 0,46/0,02 20/5 0,46/0,14							60/7 0,6/0,22	50/4 0,8/0,01																		30/4 0,18/0,01 40/4 0,18/0,01 70/4 0,1/0,01		
81	Koło zębate stożkowe			10/5 0,4/0,02 20/5 0,4/0,2							60/7 0,6/0,3	50/4 0,3/0,02																		30/4 0,2/0,02 40/4 0,2/0,02 70/4 0,1/0,02		
82	Koło zębate stożkowe	10/4 0,4/0,06 20/4 0,5/0,09									40/7 0,6/0,3																				30/4 0,5/0,04 50/4 0,1/0,01	
83	Koło płaskie zębate			50/4 0,5/0,1		10/4 0,5/0,1 20/4 0,5/0,1 30/4 0,5/0,2	60/6 1/3,5																								40/4 0,4/0,01	
84	Dźwignia wodząca					10/7 0,4/0,3 20/7 0,4/0,28 30/8 0,4/0,21	60/8 0,5/0,5					40/6 0,2/0,23 50/6 0,2/0,08																				
85	Koło zębate stożkowe		10/4 0,4/0,07 20/4 0,5/0,06								50/7 0,5/0,16	40/4 0,4/0,01																			30/4 0,15/0,01	
86	Koło zębate stożkowe		10/4 0,4/0,06 20/4 0,5/0,05								50/7 0,5/0,15	40/4 0,4/0,01																			30/4 0,15/0,01	
87	Koło zębate stożkowe			10/5 1,0/0,3 20/4 1,0/0,25					70/6 1,0/0,2 80/7 1,0/0,3	50/6 0,5/0,3						30/6 1,0/0,1 40/5 1,0/0,05															60/4 0,5/0,1	

Lp.	Nazwa i numer części	Stanowiska robocze																													
		TUM25	TUD40	TUD50	TUB32	TR10CIN	ZFC20	SOJ10	SAC30	ZFTK	WR50	FNC25	NUA25	WS15	SWA25	SWB25	SWA10	FWA32	SPC20	SPD30	TWC8	SBB75	TGC8	PM120	WKA40	RNA14	RH25	WCC80	PHW12	RVA40	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
88	Koło zębate stożkowe	10/4 0,5/0,2 20/4 0,5/0,15								50/7 0,5/0,4						40/5 1,0/0,05													30/4 0,5/0,05		
89	Koło zębate stożkowe	10/4 0,5/0,2 20/4 0,5/0,15								50/7 0,5/0,4						40/5 1,0/0,05													30/4 0,5/0,05		
90	Koło zębate stożkowe					10/5 0,5/0,16 40/5 0,4/0,19				50/7 0,5/0,58																		20/4 0,3/0,01 30/4 0,3/0,01			
91	Koło zębate walcowe		10/4 0,5/0,2 20/4 0,5/0,15	50/4 0,5/0,1			60/7 0,5/0,5																						30/4 0,6/0,06 40/4 0,4/0,1		
92	Koło zębate z piastą jednostronną	10/4 0,5/0,1 20/5 0,5/0,1					40/7 0,5/0,3																						30/4 0,5/0,05		
93	Koło zębate z piastą dwustronną	10/4 0,5/0,5 20/4 0,5/0,1					30/7 0,3/0,3																								
94	Koło zębate stożkowe			10/4 0,5/0,15 20/4 0,5/0,15						50/7 1,0/0,4					40/5 0,8/0,05														30/4 0,4/0,05		
95	Koło zębate stożkowe			10/4 0,5/0,2 20/4 0,5/0,2						50/7 1,0/0,4					40/5 0,8/0,1														30/4 0,4/0,05		
96	Koło zębate stożkowe		10/5 0,5/0,05 20/5 0,4/0,16					50/6 0,3/0,1		40/7 0,8/0,2					60/6 0,3/0,05		30/5 0,7/0,15														
97	Koło zębate stożkowe		10/5 0,4/0,05 20/5 0,4/0,16					40/6 0,3/0,1		30/7 0,8/0,2					50/6 0,3/0,05																
98	Koło zębate stożkowe						60/5 1,0/0,2		20/4 0,3/0,04					30/4 0,4/0,01															40/4 0,1/0,03 50/4 0,1/0,08	10/4 0,9/0,06	
99	Koło zębate z piastą dwustronną			10/4 0,5/0,4			50/5 0,6/0,5		20/4 0,3/0,06																				30/4 0,4/0,02 40/4 0,4/0,03		
100	Koło zębate stożkowe	10/4 0,5/0,2 20/4 0,5/0,15								60/6 1,0/0,3		40/6 0,5/0,2					50/5 0,5/0,2 70/5 0,8/0,1													30/4 0,5/0,06	
101	Koło zębate stożkowe					10/5 0,5/0,45 20/5 0,6/0,77		50/7 0,5/0,3		40/7 1,0/0,7																			30/5 0,3/0,12		

W poszczególnych polach tabeli znajdują się podstawowe informacje o realizacji zadań. Przyjato następujące oznaczenia:

Nr zadania / Grupa zaszeregowania zadania
Czas przebrojenia / Czas jednostkowy

Przykładowo oznaczenie:

20 / 4
0,4 / 0,07

informuje o realizacji zadania nr 20 z 4 stopniem zaszeregowania (parametr nie uwzględniany pracy, dotyczy zaszeregowania robót ze względu na przydział pracownika do pracy), dla którego czas przebrojenia wynosi 0,4 [h], czas jednostkowy wynosi 0,07 [h].

Dodatek C. SKRYPTY PROCEDUR SKŁADOWANYCH SŁUŻĄCYCH DO REALIZACJI ALGORYTMU PRODUFLOW PM

W dodatku zamieszczono kod programowy języka T-SQL, który posłużył do budowy jednej z procedur składowanych, służącej do grupowania zleceń (*spDendrogram*).

1. Kod T-SQL procedury składowanej *spDendrogram*

```
CREATE PROCEDURE spDendrogram
@THId          int,
@TBId          int,
@TBVersion     int
AS
BEGIN
/*Deklaracja tablicy przechowującej zlecenia, których marszruty podlegają klastrowaniu*/
DECLARE @Orders table
(
  RowNumber          int,
  SOId               char(16),
  SProductId         int,
  VariantOfRouting  int
)
/*Zmienna zliczająca i indeksująca rekordy w tabeli @Orders*/
DECLARE @CountOrders int
DECLARE @iOrders int
/*Deklaracja zmiennych pozwalających znaleźć marszruty technologiczne dla zadanych zleceń*/
DECLARE @SProductId int
DECLARE @VariantOfRouting int
/*Deklaracja tablicy przechowującej marszruty dla zadanych zleceń*/
DECLARE @Singletons table
(
  SingletonNumber    int,
  SProductId         int,
  VariantOfRouting  int,
  CountIV           int
)
/*Deklaracja zmiennej zliczającej ilość rekordów zapisanych w tabeli @Singletons*/
DECLARE @CountSingletons int
/*Deklaracja zmiennej przechowującej ilość operacji danej marszruty*/
DECLARE @NumberOfOperations int
/*Deklaracja zmiennych pozwalających ustalić najdłuższy singleton spośród rekordów zapisanych w tabeli @Singletons,
Pozwala to ustalić ilość poziomów dendrogramu*/
DECLARE @MaxLevel int
DECLARE @CountIV int
/*Deklaracja zmiennej przechowującej identyfikator automatycznie przypisany do klastra*/
DECLARE @ClusterIdentity int
/*Deklaracja zmiennej iterującej przejścia po poszczególnych poziomach dendrogramu*/
DECLARE @iLevel int
/*Deklaracja tablicy przechowującej wszystkie klastry dendrogramu na określonym poziomie*/
DECLARE @ClustersLevel table
(
  RowNumber          int,
  ClusterId         int,
  ClusterLevel      int,
  ClusterType       char(1)
)
/*Zmienna zliczająca i indeksująca rekordy w tabeli @ClustersLevel*/
DECLARE @CountClustersLevel int
DECLARE @iClustersLevel int
/*Deklaracja tablicy przechowującej łańcuchy klastrów*/
DECLARE @ClustersChain table
(
  ChainNumber        int,
  ClusterId         int,
  ChainId           int,
  InstanceId        varchar(20)
)
/*Deklaracja zmiennych pozwalających sprawdzić, w której tabeli znajduje się zapisana sekwencja łańcuch klastra*/
DECLARE @ClusterId int
DECLARE @ClusterType char(1)
```

```

/*Deklaracja iteratorów pętli pozwalających porównać ze sobą dwa klastry*/
DECLARE @FirstCluster int
DECLARE @SecondCluster int
/*Deklaracja zmiennej przechowującej identyfikator pierwszego porównywanego klastra*/
DECLARE @ClusterId1 int
/*Deklaracja typu udtSequence przechowującego łańcuch pierwszego klastra*/
DECLARE @FirstClusterChain udtSequence
/*Deklaracja zmiennej przechowującej identyfikator drugiego porównywanego klastra*/
DECLARE @ClusterId2 int
/*Deklaracja typu udtSequence przechowującego łańcuch drugiego klastra*/
DECLARE @SecondClusterChain udtSequence
/*Deklaracja zmiennej przechowującej długość najdłuższego wspólnego podłańcucha*/
DECLARE @ChainLength int
/*Deklaracja zmiennej przechowującej początek najdłuższego wspólnego podłańcucha na łańcuchu
pierwszego porównywanego klastra*/
DECLARE @ChainBegin int
/*Deklaracja tabeli tymczasowej przechowującej nowo tworzone klastry na określonym poziomie,
wymagające dalszego porównania*/
DECLARE @TemporaryClusters table
(
    RowNumber int,
/*ClusterId nie ma bo nowo tworzone klastry mają automatycznie przydzielane identyfikatory za
pomocą polecenia IDENTITY*/
    ClusterLevel int,
    ClusterType char(1),
    Covered bit,
    THId int,
    TBId int,
    TBVersion int,
    CoveredClusterId1 int,
    CoveredClusterId2 int
)
/*Zmienna zliczająca i indeksująca rekordy w tabeli @TemporaryClusters*/
DECLARE @CountTemporaryClusters int
DECLARE @iTemporaryClusters int
/*Deklaracja tabeli tymczasowej przechowującej łańcuchy nowo utworzonych klastrów*/
DECLARE @TemporarySequences table
(
    RowNumber int,
    ClusterRowNumber int,
    SequenceId int,
    InstanceId varchar(20)
)
/*Zmienna zliczająca i indeksująca rekordy w tabeli @TemporarySequences*/
DECLARE @CountTemporarySequences int
DECLARE @iTemporarySequences int
/*Deklaracja zmiennych przechowujących parametry pozwalające utworzyć łańcuch dla nowo
tworzonego klastra*/
DECLARE @TemporaryChainId int
DECLARE @TemporaryInstanceId varchar(20)
/*Deklaracja tabeli przechowującej informacje o stanie klastrów tymczasowych*/
DECLARE @StateClusters table
(
    RowNumber int,
    ClusterRowNumber int,
    ClusterState char(1)
/*ClusterState przyjmuje następujące wartości:
C - gdy klaster został utworzony na stałe w bazie danych
V - gdy klaster został wchłonięty przez klaster wcześniej zapisany w bazie danych, w przypadku
gdy klaster miał taki sam łańcuch jak klaster wcześniej wprowadzony
R - gdy klaster został odrzucony z powodu posiadania skądś należącego do innego wcześniej
wprowadzonego klastra*/
)
/*Zmienna zliczająca i indeksująca rekordy w tabeli @StateClusters*/
DECLARE @CountStateClusters int
DECLARE @iStateClusters int
/*Zmienna przechowująca stan 1 porównywanego klastra w tabeli @StateClusters*/
DECLARE @ClusterState1 char(1)
/*Zmienna przechowująca stan 2 porównywanego klastra w tabeli @StateClusters*/
DECLARE @ClusterState2 char(1)
/*Deklaracja tabeli przechowującej skądś klastrów zapisanych na stałe w bazie danych*/
DECLARE @Components table
(
    RowNumber int,
    ClusterRowNumber int,
    CoveredClusterId int
)
/*Zmienna zliczająca i indeksująca rekordy w tabeli @Components*/
DECLARE @CountComponents int

```

```

DECLARE @iComponents int
/*Deklaracja zmiennych przechowuj¹cych identyfikatory klastrów przykrytych przez pierwszy
porównywany klaster*/
DECLARE @FirstClusterId1 int
DECLARE @FirstClusterId2 int
/*Deklaracja zmiennych przechowuj¹cych identyfikatory klastrów przykrytych przez drugi
porównywany klaster*/
DECLARE @SecondClusterId1 int
DECLARE @SecondClusterId2 int
/*****
/*****Algorytm*****/
/*****
/*Usunięcie kodów hierarchicznych klastrów z tabeli ClusterHierarchy*/
DELETE FROM ClusterHierarchy WHERE THID = @THID AND TBID = @TBID AND TBVersion = @TBVersion
/*Usunięcie klastrów z tabel w określonym oknie planistycznym*/
DELETE FROM ClusterInfo WHERE THID = @THID AND TBID = @TBID AND TBVersion = @TBVersion
/* ##### STAGE 1 - analiza zleceń podlegaj¹cych klastrowaniu #####*/
/*Uzup³nienie tabeli @Orders zleceniami, których marszruty podlegaj¹ klastrowaniu*/
INSERT INTO @Orders
SELECT ROW_NUMBER() OVER(ORDER BY SOId) AS RowNumber, SOId, SProductId, VariantOfRouting FROM
StageOrderDef WHERE THID = @THID AND TBID = @TBID AND TBVersion = @TBVersion
/*Przeliczenie iloœci zapisanych rekordów*/
SELECT @CountOrders = COUNT(*) FROM @Orders
/* ##### STAGE 2 - utworzenie singletonów #####*/
/* Uzup³nienie tabeli @Singletons marszrutami podlegaj¹cymi klastrowaniu */
SELECT @iOrders = 1
WHILE @iOrders < @CountOrders+1
BEGIN
/*Ustawienie wartoœci parametrów pozwalaj¹cych znaleŹæ odpowiedni¹ marszrutê w tabeli
Routing*/
SELECT @SProductId = SProductId FROM @Orders WHERE RowNumber = @iOrders
SELECT @VariantOfRouting = VariantOfRouting FROM @Orders WHERE RowNumber = @iOrders
/*Ustawienie wartoœci parametru przechowuj¹cego iloœæ operacji danej marszrut*/
SELECT @NumberOfOperations = NumberOfOperations FROM Routing WHERE SProductId = @SProductId
AND VariantOfRouting = @VariantOfRouting
/*Zliczenie iloœci rekordów zapisanych w tabeli @Singletons*/
SELECT @CountSingletons = COUNT(*) FROM @Singletons
/*Uzup³nienie tabeli @Singletons danymi*/
INSERT INTO @Singletons
VALUES( @CountSingletons+1, @SProductId, @VariantOfRouting, @NumberOfOperations)
SELECT @iOrders = @iOrders + 1
END
/*Znalezienie najd³u¿szego singletona zapisanego w tabeli @Singletons*/
SELECT @iOrders = 1
SELECT @MaxLevel = 0
WHILE @iOrders < @CountOrders+1
BEGIN
/*Ustawienie wartoœci parametru przechowuj¹cego iloœæ operacji danej marszrut*/
SELECT @CountIV = CountIV FROM @Singletons WHERE SingletonNumber = @iOrders
/*Nadpisanie parametru @MaxLevel gdy d³ugoœæ marszruty jest wiêksza ni¿ wartoœæ @MaxLevel*/
IF @MaxLevel < @CountIV
SELECT @MaxLevel = @CountIV
SELECT @iOrders = @iOrders + 1
END
/*Wpisanie singletonów na sta³e do tabel ClusterInfo i Singleton w bazie danych*/
SELECT @iOrders = 1
WHILE @iOrders < @CountOrders+1
BEGIN
/*Ustawienie wartoœci parametru przechowuj¹cego iloœæ operacji danej marszrut*/
SELECT @CountIV = CountIV FROM @Singletons WHERE SingletonNumber = @iOrders
/*Ustawienie wartoœci parametrów identyfikuj¹cych marszrutê*/
SELECT @SProductId = SProductId FROM @Singletons WHERE SingletonNumber = @iOrders
SELECT @VariantOfRouting = VariantOfRouting FROM @Singletons WHERE SingletonNumber = @iOrders
/*Wstawienie rekordu do tabeli ClusterInfo*/
INSERT INTO ClusterInfo(ClusterLevel, ClusterType, Covered, THID, TBID, TBVersion)
VALUES(@CountIV, 'S', 'False', @THID, @TBID, @TBVersion)
/*Wpisanie do zmiennej @ClusterIdentity identyfikatora automatycznie przypisanego do klastra*/
SELECT @ClusterIdentity = @@IDENTITY
/*Wstawienie rekordu do tabeli Singleton*/
INSERT INTO Singleton
VALUES(@ClusterIdentity, @SProductID, @VariantOfRouting)
SELECT @iOrders = @iOrders + 1
END
/* ##### STAGE 3 - utworzenie klastrów na poziomach od @MaxLevel do 2 #####*/
/*Ustawienie wartoœci iteratora przejœcia po poziomach dendrogramu na max*/
SELECT @iLevel = @MaxLevel
/*Wykonanie w p³tli wi¹zania klastrów po kolejnych poziomach*/
WHILE @iLevel > 1
BEGIN

```



```

/*Wyczyszczenie tabeli @ClustersLevel z danych zapisanych w poprzedniej iteracji*/
DELETE FROM @ClustersLevel
/*Wyczyszczenie tabeli tymczasowej @TemporaryClusters przechowuj¹cej klastry podlegaj¹ce
dlaszemu sprawdzeniu na okreœlonym poziomie*/
DELETE FROM @TemporaryClusters
/*Wyczyszczenie tabeli tymczasowej @TemporarySequences przechowuj¹cej ³aŃcuchy klastrów*/
DELETE FROM @TemporarySequences
/*Wyszukanie obiektów w tabeli ClusterInfo, które s¹ zapisane na poziomie co najmniej @iLevel
i nie s¹ oznaczone etykiet¹ Covered*/
INSERT INTO @ClustersLevel
SELECT ROW_NUMBER() OVER(ORDER BY ClusterId) AS RowNumber, ClusterId, ClusterLevel,
ClusterType FROM ClusterInfo WHERE THId = @THId AND TBId = @TBId AND TBVersion = @TBVersion
AND ClusterLevel >= @iLevel AND Covered = 'False'
/*Przeliczenie iloœci rekordów zapisanych w tabeli @ClustersLevel*/
SELECT @CountClustersLevel = COUNT(*) FROM @ClustersLevel
/*Ustawienie licznika klastrów*/
SELECT @iClustersLevel = 1
/*Zapisanie ³aŃcuchów ka¿dego klastra w tabeli @ClustersChain pod warunkiem, ¿e w tabeli
@ClustersLevel znajduj¹ siê co najmniej dwa rekordy*/
IF @CountClustersLevel > 1
BEGIN
/*Wyczyszczenie tabeli @ClustersChain z danych zapisanych w poprzedniej iteracji*/
DELETE FROM @ClustersChain
WHILE @iClustersLevel < @CountClustersLevel+1
BEGIN
/*Ustawienie parametrów pozwalaj¹cych zlokalizowaæ odpowiednie rekordy w okreœlonej tabeli*/
SELECT @ClusterId = ClusterId FROM @ClustersLevel WHERE RowNumber = @iClustersLevel
SELECT @ClusterType = ClusterType FROM @ClustersLevel WHERE RowNumber = @iClustersLevel
/*Sprawdzenie w której tabeli zlokalizowana jest marszruta klastra*/
IF @ClusterType = 'S'
/*Gdy typ klastra ma wartoœæ 'S', tzn. ¿e nale¿y odnaleŹæ operacje marszruty zapisane w tabeli
Operation*/
BEGIN
/*Ustawienie parametrów SProductId oraz VariantOfRouting aby wyselekcjonowaæ wszystkie
operacje marszruty*/
SELECT @SProductId = SProductId FROM Singleton WHERE ClusterId = @ClusterId
SELECT @VariantOfRouting = VariantOfRouting FROM Singleton WHERE ClusterId = @ClusterId
/*Znalezienie i wpisanie wszystkich operacji marszruty do tabeli @ClusterChain*/
INSERT INTO @ClustersChain
SELECT ROW_NUMBER() OVER(ORDER BY OperationId) AS ChainNumber, @ClusterId, OperationId,
SUBSTRING(ConfId, PATINDEX('%;',ConfId)+1, PATINDEX('%-;',ConfId)-PATINDEX('%;',ConfId)-1)
AS InstanceId FROM Operation WHERE SProductId = @SProductId AND VariantOfRouting =
@VariantOfRouting
END
ELSE
/*Gdy typ klastra ma wartoœæ 'K', tzn. ¿e nale¿y odnaleŹæ operacje marszruty zapisane w tabeli
ClusterSequence*/
BEGIN
/*Znalezienie i wpisanie wszystkich operacji klastra do tabeli @ClusterChain*/
INSERT INTO @ClustersChain
SELECT ROW_NUMBER() OVER(ORDER BY ClusterId, SequenceId) AS ChainNumber, @ClusterId,
SequenceId, InstanceId FROM ClusterSequence WHERE ClusterId = @ClusterId
END
SELECT @iClustersLevel = @iClustersLevel + 1
END
/*Ustawienie wartoœci pierwszego iteratora porównawczego klastrów na 1*/
SELECT @FirstCluster = 1
/*Wykonanie pêtli sprawdzaj¹cych klastry ze sob¹*/
WHILE @FirstCluster < @CountClustersLevel
BEGIN
/*Usuniêcie zapisanego ³aŃcucha pierwszego klastra*/
DELETE FROM @FirstClusterChain
/*Ustawienie identyfikatora pierwszego porównywanego klastra*/
SELECT @ClusterId1 = ClusterId FROM @ClustersLevel WHERE RowNumber = @FirstCluster
/*Zapisanie ³aŃcucha pierwszego porównywanego klastra*/
INSERT INTO @FirstClusterChain
SELECT ChainId, ClusterId, InstanceId FROM @ClustersChain WHERE ClusterId = @ClusterId1 ORDER
BY ChainId
/*Ustawienie wartoœci drugiego iteratora porównawczego klastrów na wartoœæ o 1 wiêksz¹ ni¿
wartoœæ iteratora @FirstCluster*/
SELECT @SecondCluster = @FirstCluster + 1
WHILE @SecondCluster < @CountClustersLevel+1
BEGIN
/*Usuniêcie zapisanego ³aŃcucha drugiego klastra*/
DELETE FROM @SecondClusterChain
/*Ustawienie identyfikatora drugiego porównywanego klastra*/
SELECT @ClusterId2 = ClusterId FROM @ClustersLevel WHERE RowNumber = @SecondCluster
/*Zapisanie ³aŃcucha drugiego porównywanego klastra*/
INSERT INTO @SecondClusterChain

```

```

SELECT ChainId, ClusterId, InstanceId FROM @ClustersChain WHERE ClusterId = @ClusterId2 ORDER
BY ChainId
/* Uruchomienie procedury obliczaj¹cej odlegoœæ dwóch klastrów wed³ug regu³y Longest Common
Flow (LCF) */
EXECUTE dbo.spLCF
@FirstR = @FirstClusterChain,
@SecondR = @SecondClusterChain,
@ChainMaxLength = @ChainLength OUTPUT,
@ChainFirstBegin = @ChainBegin OUTPUT
/*Wpisanie nowego klastra do tabeli tymczasowej pod warunkiem, ¿e d³ugoœæ jego ³ańcucha jest
równa poziomowi dendrogramu*/
IF @ChainLength = @iLevel
BEGIN
/*Przeliczenie iloœci rekordów zapisanych w tabeli @TemporaryClusters*/
SELECT @CountTemporaryClusters = COUNT(*) FROM @TemporaryClusters
/*Utworzenie klastra tymczasowego*/
INSERT INTO @TemporaryClusters
VALUES(@CountTemporaryClusters+1, @iLevel, 'K', 'False', @THId, @TBId, @TBVersion,
@ClusterId1, @ClusterId2)
/*Ustawienie iteratora sekwencji klastra tymczasowego na 1*/
SELECT @iTemporarySequences = 1
WHILE @iTemporarySequences < @ChainLength+1
BEGIN
/*Przeliczenie iloœci rekordów w tabeli @TemporarySequences*/
SELECT @CountTemporarySequences = COUNT(*) FROM @TemporarySequences
/*Wstawienie parametrów z rekordu w tabeli @FirstClusterChain*/
SELECT @TemporaryChainId = ChainId FROM @FirstClusterChain WHERE ChainId =
(@ChainBegin+@iTemporarySequences-1)
SELECT @TemporaryInstanceId = InstanceId FROM @FirstClusterChain WHERE ChainId =
(@ChainBegin+@iTemporarySequences-1)

/*Utworzenie rekordów przechowuj¹cych ³ańcuch klastra, który stanowi fragment ³ańcucha klastra
zapisanego w typie tabelarycznym @FirstClusterChain*/
INSERT INTO @TemporarySequences
VALUES(@CountTemporarySequences+1, @CountTemporaryClusters+1, @TemporaryChainId,
@TemporaryInstanceId)
SELECT @iTemporarySequences = @iTemporarySequences + 1
END
END
SELECT @SecondCluster = @SecondCluster + 1
END
SELECT @FirstCluster = @FirstCluster + 1
END
/*Wykonanie operacji sprawdzaj¹cych, czy nowo utworzone klastry maj¹ takie same ³ańcuchy*/
/*Przeliczenie iloœci rekordów w tabeli @TemporaryClusters*/
SELECT @CountTemporaryClusters = COUNT(*) FROM @TemporaryClusters
/*Usuniêcie danych z tabel @StateClusters oraz @Components*/
DELETE FROM @StateClusters
DELETE FROM @Components
/*Ustawienie wartoœci iteratora pierwszego porównywanego klastra na 1*/
SELECT @FirstCluster = 1
/*Wykonanie w pierwszej pêtli porównania klastrów - pierwsza pêtla jest odpowiedzialna za
uzupe³nienie pierwszego porównywanego klastra na lew¹ stronê porównania*/
WHILE @FirstCluster < @CountTemporaryClusters+1
BEGIN
/*Usuniêcie ³ańcucha operacji ze zmiennej przechowuj¹cej ³ańcuchy pierwszego porównywanego
klastra*/
DELETE FROM @FirstClusterChain
/*Sprawdzenie, czy wybrany klaster zosta³ wczeœniej zapisany w tabeli tymczasowej
@StateCluster*/
IF EXISTS (SELECT * FROM @StateClusters WHERE ClusterRowNumber = @FirstCluster)

SELECT @ClusterState1 = ClusterState FROM @StateClusters WHERE ClusterRowNumber =
@FirstCluster
ELSE
SELECT @ClusterState1 = 'N' /*N - oznacza, ¿e danego klastra nie ma zapisanego w bazie
danych*/
/*Dany klaster bêdzie wpisany na sta³e do bazy, je¿eli parametr @ClusterState1 ma wartoœæ 'N'
!!!! w przeciwnym wypadku, tj. gdy klaster ma ustawiony stan 'V' lub 'R' dany klaster bêdzie
pomijany - pêtla zwiêkszy wartoœæ iteratora i przejdzie do analizy nastêpnego klastra*/
IF @ClusterState1 = 'N'
BEGIN
/*Sprawdzenie czy klastry przykryte przez wybrany klaster tymczasowy nie wystêpuj¹ w innym
istniej¹cym klastrze*/
/*Wpisanie pierwszego sk³adnika*/
SELECT @FirstClusterId1 = CoveredClusterId1 FROM @TemporaryClusters WHERE RowNumber =
@FirstCluster
/*Wpisanie drugiego sk³adnika*/

```

```

SELECT @FirstClusterId2 = CoveredClusterId2 FROM @TemporaryClusters WHERE RowNumber =
@FirstCluster
/*Sprawdzenie czy klastry przykryte przez pierwszy porównywany klastery występują jako klastry
przykryte przez jakieś istniejące klastry*/
/*Jeżeli tak, to klastery zostają odrzucone, gdyż co najmniej jeden z jego składników należy do
innego istniejącego klastra*/
IF (EXISTS (SELECT * FROM @Components WHERE CoveredClusterId = @FirstClusterId1)) OR (EXISTS
(SELECT * FROM @Components WHERE CoveredClusterId = @FirstClusterId2))
BEGIN
/*Wpisanie parametru do tabeli @StateClusters*/
SELECT @CountStateClusters = COUNT(*) FROM @StateClusters
INSERT INTO @StateClusters
VALUES (@CountStateClusters+1, @FirstCluster, 'R')
END
/*Jeżeli nie, to pierwszy porównywany klastery zostanie wpisany do bazy danych na stałe*/
ELSE
BEGIN
/*Wpisanie parametru do tabeli @StateClusters*/
SELECT @CountStateClusters = COUNT(*) FROM @StateClusters
INSERT INTO @StateClusters
VALUES (@CountStateClusters+1, @FirstCluster, 'C')
/*Wpisanie składników nowego tworzonego klastra do tabeli @Components*/
SELECT @CountComponents = COUNT(*) FROM @Components
INSERT INTO @Components
VALUES (@CountComponents+1, @FirstCluster, @FirstClusterId1)
INSERT INTO @Components
VALUES (@CountComponents+2, @FirstCluster, @FirstClusterId2)
/*Utworzenie rekordu w bazie danych*/
INSERT INTO ClusterInfo(ClusterLevel, ClusterType, Covered, THId, TBId, TBVersion)
VALUES (@iLevel, 'K', 'False', @THId, @TBId, @TBVersion)
/*Utworzenie łańcucha utworzonego klastra*/
SELECT @ClusterIdentity = @@IDENTITY
/*Przeliczenie ilości rekordów łańcucha określonego klastra w tabeli @TemporarySequences*/
SELECT @CountTemporarySequences = COUNT(*) FROM @TemporarySequences WHERE ClusterRowNumber =
@FirstCluster
/*Ustawienie iteratora łańcucha na 1*/
SELECT @iTemporarySequences = 1
WHILE @iTemporarySequences < @CountTemporarySequences+1
BEGIN
/*Wstawienie parametrów z rekordu w tabeli @TemporarySequences*/
SELECT @TemporaryChainId = SequenceId FROM @TemporarySequences WHERE ClusterRowNumber =
@FirstCluster AND RowNumber = (@iLevel*(@FirstCluster-1)+@iTemporarySequences)
SELECT @TemporaryInstanceId = InstanceId FROM @TemporarySequences WHERE ClusterRowNumber =
@FirstCluster AND RowNumber = (@iLevel*(@FirstCluster-1)+@iTemporarySequences)
/*Wstawienie rekordu na stałe do tabeli ClusterSequence*/
INSERT INTO ClusterSequence
VALUES (@ClusterIdentity, @TemporaryChainId, @TemporaryInstanceId)
SELECT @iTemporarySequences = @iTemporarySequences + 1
END
/*Ustawienie wartości Covered klastrów przykrytych na True oraz wartości CoveringClusterId na
@ClusterIdentity*/
/*Aktualizacja danych pierwszego klastra*/
UPDATE ClusterInfo
SET
Covered = 'True',
CoveringClusterId = @ClusterIdentity
WHERE ClusterId = @FirstClusterId1
/*Aktualizacja danych drugiego klastra*/
UPDATE ClusterInfo
SET
Covered = 'True',
CoveringClusterId = @ClusterIdentity
WHERE ClusterId = @FirstClusterId2
/*Zapisanie łańcucha pierwszego porównywanego klastra*/
INSERT INTO @FirstClusterChain
SELECT SequenceId, ClusterRowNumber, InstanceId FROM @TemporarySequences WHERE
ClusterRowNumber = @FirstCluster ORDER BY SequenceId
/*Ustawienie wartości drugiego iteratora porównawczego klastrów na wartość o 1 większą niż
wartość iteratora @FirstCluster*/
SELECT @SecondCluster = @FirstCluster + 1
WHILE @SecondCluster < @CountClustersLevel+1
BEGIN
/*Usunięcie zapisanego łańcucha drugiego klastra*/
DELETE FROM @SecondClusterChain
/*Sprawdzenie, czy wybrany klastery został już wpisany do tabeli tymczasowej @StateCluster*/
IF EXISTS (SELECT * FROM @StateClusters WHERE ClusterRowNumber = @SecondCluster)

SELECT @ClusterState2 = ClusterState FROM @StateClusters WHERE ClusterRowNumber =
@SecondCluster

```

```

ELSE
SELECT @ClusterState2 = 'N' /*N - oznacza, że danego klastra nie ma zapisanego w bazie
danych*/
/*Dany klaster będzie porównany z klastrem pierwszym, gdy parametr @ClusterState2 ma wartość
'N'
!!! w przeciwnym wypadku, tj. gdy klaster ma ustawiony stan 'V' lub 'R' dany klaster będzie
pomijany - pętla zwiększy wartość iteratora i przejdzie do analizy następnego klastra*/
IF @ClusterState2 = 'N'
BEGIN
/*Ustalenie identyfikatorów klastrów które zostały przykryte przez drugi porównywany klaster*/
SELECT @SecondClusterId1 = CoveredClusterId1 FROM @TemporaryClusters WHERE RowNumber =
@SecondCluster
SELECT @SecondClusterId2 = CoveredClusterId2 FROM @TemporaryClusters WHERE RowNumber =
@SecondCluster
/*Sprawdzenie czy klastry przykryte przez drugi porównywany klaster występują jako klastry
przykryte przez jakiś istniejący klaster inny niż ten pierwszy porównywany w pętli*/
/*Jeżeli tak, to klaster zostaje odrzucony, gdyż co najmniej jeden z jego składników należy do
innego istniejącego klastra*/
IF (EXISTS (SELECT * FROM @Components WHERE CoveredClusterId = @SecondClusterId1 AND
ClusterRowNumber <> @FirstCluster)) OR (EXISTS (SELECT * FROM @Components WHERE
CoveredClusterId = @SecondClusterId2 AND ClusterRowNumber <> @FirstCluster))
BEGIN
/*Wpisanie parametru do tabeli @StateClusters*/
SELECT @CountStateClusters = COUNT(*) FROM @StateClusters
INSERT INTO @StateClusters
VALUES (@CountStateClusters+1, @SecondCluster, 'R')
END
/*Jeżeli nie, to ścieżka klastra jest porównana z ścieżką klastra pierwszego*/
ELSE
BEGIN
/*Zapisanie ścieżki drugiego porównywanego klastra*/
INSERT INTO @SecondClusterChain
SELECT SequenceId, ClusterRowNumber, InstanceId FROM @TemporarySequences WHERE
ClusterRowNumber = @SecondCluster ORDER BY SequenceId
/* Uruchomienie procedury obliczającej odległość dwóch klastrów według reguły Longest Common
Flow (LCF) */
EXECUTE dbo.spLCF
@FirstR = @FirstClusterChain,
@SecondR = @SecondClusterChain,
@ChainMaxLength = @ChainLength OUTPUT,
@ChainFirstBegin = @ChainBegin OUTPUT
/*Jeżeli porównywane klastry mają jednakowy ścieżkę, tj. znaleziona długość jest równa
poziomowi dendrogramu wówczas klastry są ze sobą złączone*/
IF @ChainLength = @iLevel
BEGIN
/*Sprawdzenie, czy obydwa klastry mają takie same składniki*/
/*Jeżeli tak, to zostaje dopisany jedynie jeden składnik do istniejącego klastra*/
IF @FirstClusterId1 = @SecondClusterId1 OR @FirstClusterId1 = @SecondClusterId2 OR
@FirstClusterId2 = @SecondClusterId1 OR @FirstClusterId2 = @SecondClusterId2
BEGIN
/*Wpisanie parametru do tabeli @StateClusters*/
SELECT @CountStateClusters = COUNT(*) FROM @StateClusters
INSERT INTO @StateClusters
VALUES (@CountStateClusters+1, @SecondCluster, 'V')
/*Sprawdzenie czy pierwszy składnik jest inny od tych już zapisanych w klastrze*/
IF @SecondClusterId1 <> @FirstClusterId1 AND @SecondClusterId1 <> @FirstClusterId2
BEGIN
/*Dopisanie pierwszego składnika do tabeli @Components*/
SELECT @CountComponents = COUNT(*) FROM @Components
INSERT INTO @Components
VALUES (@CountComponents+1, @FirstCluster, @SecondClusterId1)
/*Aktualizacja danych pierwszego klastra*/
UPDATE ClusterInfo
SET
Covered = 'True',
CoveringClusterId = @ClusterIdentity
WHERE ClusterId = @SecondClusterId1
END
/*Inaczej drugi składnik jest inny od tych zapisanych w klastrze*/
ELSE
BEGIN
/*Dopisanie drugiego składnika do tabeli @Components*/
SELECT @CountComponents = COUNT(*) FROM @Components
INSERT INTO @Components
VALUES (@CountComponents+1, @FirstCluster, @SecondClusterId2)
/*Aktualizacja danych drugiego klastra*/
UPDATE ClusterInfo
SET
Covered = 'True',

```

```

CoveringClusterId = @ClusterIdentity
WHERE ClusterId = @SecondClusterId2
END
END
/*Jeżeli nie to klaster zostaje całkowicie wcielony do pierwszego porównywanego klastra*/
ELSE
BEGIN
/*Wpisanie parametru do tabeli @StateClusters*/
SELECT @CountStateClusters = COUNT(*) FROM @StateClusters
INSERT INTO @StateClusters
VALUES (@CountStateClusters+1, @SecondCluster, 'V')
/*Dopisanie obydwu składowych do tabeli @Components*/
SELECT @CountComponents = COUNT(*) FROM @Components
INSERT INTO @Components
VALUES (@CountComponents+1, @FirstCluster, @SecondClusterId1)
INSERT INTO @Components
VALUES (@CountComponents+2, @FirstCluster, @SecondClusterId2)
/*Aktualizacja danych obydwu klastrów*/
UPDATE ClusterInfo
SET
Covered = 'True',
CoveringClusterId = @ClusterIdentity
WHERE ClusterId = @SecondClusterId1
UPDATE ClusterInfo
SET
Covered = 'True',
CoveringClusterId = @ClusterIdentity
WHERE ClusterId = @SecondClusterId2
END
END
END
END
SELECT @SecondCluster = @SecondCluster + 1
END
END
END
SELECT @FirstCluster = @FirstCluster + 1
END
END
SELECT @iLevel = @iLevel - 1
END
/*Uruchomienie procedury spInsertClustersHierarchy tworzącej kody hierarchiczne dla każdego
klastra utworzonego za pomocą procedury spDendrogram*/
EXEC spInsertClustersHierarchy
@THid = @THid,
@TBid = @TBid,
@TBVersion = @TBVersion
END

```