XV Krajowa Konferencja Automatyki

Tom II

KKA 2005

Redaktorzy:
Zdzisław Bubnicki
Roman Kulikowski
Janusz Kacprzyk

XV Krajowa Konferencja Automatyki
Tom II

KKA
2005

Redaktorzy:
Zdzisław BUBNICKI
Roman KULIKOWSKI
Janusz KACPRZYK

ORGANIZATOR
Komitet Automatyki i Robotyki Polskiej Akademii Nauk
Instytut Badań Systemowych Polskiej Akademii Nauk

WSPÓŁORGANIZATORZY
Politechnika Warszawska
Przemysłowy Instytut Automatyki i Pomiarów
Polskie Stowarzyszenie Pomiarów, Automatyki i Robotyki

ORGANIZATOR

Komitet Automatyki i Robotyki Polskiej Akademii Nauk
Instytut Badań Systemowych Polskiej Akademii Nauk

WSPÓŁORGANIZATORZY

Politechnika Warszawska
Przemysłowy Instytut Automatyki i Pomiarów
Polskie Stowarzyszenie Pomiarów, Automatyki i Robotyki

KOMITET PROGRAMOWY

Przewodniczący	Zdzisław BUBNICKI
Zastępca Przewodniczącego	Roman KULIKOWSKI

CZŁONKOWIE

Stanisław BAŃKA
Mikołaj BUSŁOWICZ
Ryszard GESSING
Jakub GUTENBAUM
Stanisław KACZANOWSKI
Janusz KACPRZYK
Józef KORBICZ
Krzysztof KOZŁOWSKI
Krzysztof KUŹMIŃSKI
Krzysztof MALINOWSKI
Antoni NIEDERLIŃSKI
Tadeusz PUCHAŁKA
Stanisław SKOCZOWSKI
Jerzy ŚWIĄTEK
Ryszard TADEUSIEWICZ
Krzysztof TCHOŃ
Jan WĘGLARZ
Michał BIAŁKO
Władysław FINDEISEN
Henryk GÓRECKI
Jerzy JÓZEFCZYK
Tadeusz KACZOREK
Jerzy KLAMKA
Zbigniew KOWALSKI
Juliusz L. KULIKOWSKI
Kazimierz MALANOWSKI
Wojciech MITKOWSKI
Władysław PEŁCZEWSKI
Leszek RUTKOWSKI
Roman SŁOWIŃSKI
Andrzej ŚWIERNIAK
Piotr TATJEWSKI
Leszek TRYBUS
Andrzej P. WIERZBICKI

KOMITET ORGANIZACYJNY

Przewodniczący	Roman KULIKOWSKI
Zastępcy Przewodniczącego	Janusz KACPRZYK Stanisław KACZANOWSKI Tadeusz KACZOREK Krzysztof MALINOWSKI
Członkowie	Roman OSTROWSKI Tadeusz PUCHAŁKA Dariusz WAGNER
Sekretarze naukowi	Jan STUDZIŃSKI Jan W. OWSIŃSKI

ISBN 83-89475-01-4

Druk: ARGRAF, Warszawa

STEROWANIE
KOMPLEKSAMI OPERACJI

ZASTOSOWANIE TECHNIK PROGRAMOWANIA Z OGRANICZENIAMI W ZADANIACH ROZSTRZYGANIA KONFLIKTÓW ZASOBOWYCH

Robert WÓJCIK*, Krzysztof BZDYRA, Zbigniew BANASZAK*****

* Politechnika Wrocławska, Instytut Cybernetyki Technicznej
ul. Janiszewskiego 11/17, Wrocław, 50-372 Wrocław, e-mail: wojcik@ict.pwr.wroc.pl

** Politechnika Koszalińska, Wydział Elektroniki i Informatyki
ul. Śniadeckich 2, 75-453 Koszalin e-mail: kbzdyra@tu.koszalin.pl

*** Pracownia Systemów Wiedzy i Sztucznej Inteligencji, IBS PAN,
ul. Podwale 75, Wrocław, e-mail: banaszak@tu.koszalin.pl

Streszczenie: W pracy przedstawiono zastosowanie metody programowania logicznego z ograniczeniami (CLP) do wyznaczania przepływów materiałowych w systemach współbieżnych procesów produkcyjnych. Reguły rozstrzygania konfliktów zasobowych wyrażono za pomocą zbioru predykatów definiujących ograniczenia na zmienne decyzyjne problemu. Model predykatowy wyrażono w dziedzinie czasu i w dziedzinie zdarzeń, a następnie zaimplementowano w procedurach propagacji i podstawiania wartości zmiennych, które umożliwiają szybkie wyszukiwanie rozwiązań dopuszczalnych w drzewie poszukiwań problemu CP. Zdefiniowane predykaty stanowią warunki wystarczające dla wyznaczenia harmonogramów operacji wolnych od blokad i spełniających zadane ograniczenia jakościowe i ilościowe.

Słowa kluczowe: Współbieżne procesy produkcyjne, rozstrzyganie konfliktów zasobowych, harmonogramowanie operacji, programowanie z ograniczeniami

1. WSTĘP

Problem rozstrzygania konfliktów zasobowych w systemach procesów współbieżnych należy do klasy NP trudnych i rozwiązywany jest zwykle przy pomocy metod zapobiegania przed blokadami lub metod unikania blokad [1], [6]. Metody tego typu implementują różne reguły przydziału zasobów nie dopuszczające do powstawania stanów blokady, charakteryzujących się istnieniem cyklicznych żądań zasobowych procesów, z których żadne nie może zostać zrealizowane. Reguły wykorzystywane w tych metodach mają charakter warunków wystarczających, tzn. eliminują z rozważań część rozwiązań (stanów) dopuszczalnych. Oznacza to możliwość pomijania rozwiązań ekstremalizujących wykorzystywane dalej funkcje celu.

Naturalne w tej sytuacji pytanie dotyczy zależności łączącej stosunek obejmowanych analizą stanów do wszystkich możliwych stanów dopuszczalnych z okresem czasu wymaganym dla przeprowadzenia tej analizy. Poszukiwana odpowiedź, każdorazowo wymagająca uwzględnienia specyfiki problemu, uwarunkowana jest zdolnością do kreowania różnych warunków wystarczających, warunków ograniczających przestrzeń analizowanych stanów (w tym również rozwiązań dopuszczalnych). Naturalnym narzędziem dla programowania takich warunków, programowania deklaratywnego są techniki programowania z ograniczeniami [2], [3], [4].

W przedstawionym kontekście, celem pracy jest ukazanie możliwości wykorzystania technik programowania z ograniczeniami do rozstrzygania konfliktów zasobowych. Zaproponowana metodyka programowania procedur sterowania współbieżnego zilustrowana została na przykładzie gniazda obróbkowego, w którym ze względu na ograniczoną pojemność bufora obrabiarki mogą pojawić się stany blokady systemu. Załączony przykład obliczeniowy zrealizowano w oparciu o system wspomagania decyzji Mozart [4], [5].

Przedstawione w pracy techniki programowania z ograniczeniami pozwalają na poszukiwanie rozwiązań wolnych od blokad w dziedzinie czasu bądź to w dziedzinie zdarzeń. Poszukiwanie rozwiązania w dziedzinie czasu charakteryzuje się tym, że zmienne decyzyjne określają termin realizacji poszczególnych operacji. Oznacza to, że otrzymany wynik stanowi w praktyce gotowy harmonogram pracy zasobów, uwzględniający wielkości partii produkcyjnych i czasy obróbki. Jednak w przypadku zmian dotyczących parametrów czasowych zlecenia, czy też wielkości poszczególnych partii, konieczne jest ponowne poszukiwanie rozwiązania.

Poszukiwanie rozwiązań w dziedzinie zdarzeń polega na rozstrzygnięciu kolejności wykonywania operacji na zasobach współdzielonych (nazywanych zasobami krytycznymi), przez co najmniej dwa procesy lub dwie operacje w ramach jednego procesu. Zmienne decyzyjne są binarne. Upływ czasu jest rozpatrywany od zdarzenia do zdarzenia. Zmienne decyzyjne określają, jaka operacja (lub grupa operacji) stanowi kolejne zdarzenie w systemie (kolejny stan dopuszczalny). Podejście takie pozwala, wyszukiwać rozwiązania ogólne. W efekcie zmiana wielkości zlecenia nie prowadzi do konieczności ponownego poszukiwania rozwiązania.

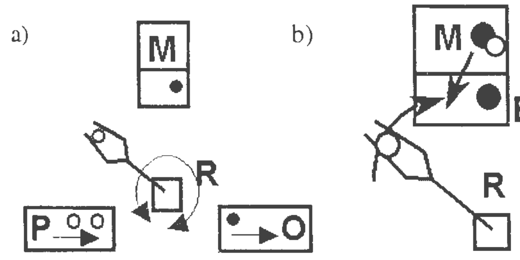
Poszukiwanie w dziedzinie zdarzeń jest podejściem ogólniejszym. Uzyskiwane rozwiązania są wolne od blokad, czyli spełnione są ograniczenia jakościowe. Rozwiązania te nie dają jednak gwarancji, że po uwzględnieniu czasu trwania operacji zostaną spełnione ograniczenia ilościowe, np. termin realizacji zleceń. W przypadku poszukiwania rozwiązań w dziedzinie czasu, otrzymane rozwiązanie spełnia obie (jakościowe i ilościowe) kategorie ograniczeń.

W pracy pokazano sposób wykorzystania metody programowania logicznego z ograniczeniami (CLP) do poszukiwania rozwiązań w dziedzinie czasu i zdarzeń. Procedury poszukiwania rozwiązań omówiono na przykładzie zadania dotyczącego planowania operacji w obrębie zrobotyzowanego gniazda obróbkowego.

2. SFORMUŁOWANIE PROBLEMU

W systemach produkcyjnych konflikty zasobowe występują na zasobach wykorzystywanych przez co najmniej dwa procesy. Na rys. 1 pokazano zrobotyzowane gniazdo obróbkowe, w którym zasobami krytycznymi są: bufor obrabiarki o ograniczonej pojemności oraz robot transportowy zdolny do przenoszenia jednego detalu. W systemie tym realizowane są potokowe procesy produkcyjne związane z obróbką detali P_k, gdzie k=1,...,5. Każdy proces realizuje kolejno 5 operacji (*OP1*,*OP2*,*OP3*,*OP4*,*OP5*) na zasobach określonych przez marszrutę (*R*, *B*, *M*, *B*, *R*). W rozważanym przypadku są to: przenoszenie detalu z podajnika do bufora (*OP1*), z bufora na obrabiarkę (*OP2*), obróbka na obrabiarce (*OP3*), następnie przenoszenia obrobionego detalu z maszyny do bufora (*OP4*) i transport obrobionego detalu z bufora na odbierak (*OP5*).

W praktyce przed wykonaniem operacji transportowej *OP1* robot *R* może znajdować się w jednym z trzech stanów: bezpośrednio przy podajniku *P*, przy buforze obrabiarki *M*, lub przy odbieraku *O*. Przyjmuje się, że w przypadku dwóch ostatnich stanów w ramach operacji *OP1* jest realizowany powrót do podajnika *P*. Czasy powrotu będą uwzględniane w ramach ujednoliconego czasu wykonywania operacji *OP1*. Podejście takie pozwala uniknąć konieczności modyfikacji czasów operacji w zależności od stanu wykorzystywanego zasobu. Analogicznie, przed wykonaniem operacji *OP5* robot *R* może znajdować się przy buforze obrabiarki *M* lub przy odbieraku *O*. Również i w tym przypadku przyjmuje się, że czas wykonania operacji *OP5* będzie ten sam niezależnie od stanu początkowego zasobu.



Legenda:
P - podajnik, O - odbierak
M - obrabiarka, B - bufor
Detale (nieobrobiony i obrobiony): ○ ●

Rys. 1. Stanowisko obróbkowe: a) schemat, b) stan blokady.

W systemie przedstawionym na rys. 1 może dojść do blokady, gdy organizacja procesów w obrębie stanowiska produkcyjnego nie będzie uwzględniała stanów przyszłych. W rozpatrywanym przykładzie w wyniku zajęcia bufora przez detal nieobrobiony, przenoszony przez robota, lub detal obrobiony pochodzący z obrabiarki, nastąpi sytuacja, w której nie będzie możliwości „powrotu" obrobionego detalu do bufora, jak również możliwości zwolnienia miejsca w buforze (czy to przez odebranie obrobionego detalu, bo takiego nie będzie w buforze, czy to przez przeniesienie kolejnego nieobrobionego detalu na obrabiarkę). Wynika stąd, że rozwiązywanie konfliktów zasobowych, nie sprowadza się tylko do zaplanowania kolejności operacji na krytycznym zasobie. Konieczne jest również takie określenie ich następstwa, aby w systemie nie pojawił się stan blokady.

Na przykładzie konfliktów przedstawionych na rys. 1, omówione zostaną metody wariantowania rozwiązań wolnych od blokad oparte o techniki programowania w logice ograniczeń (CLP) [2], [5]. Metody te pozwalają wybrać z całego zbioru rozwiązań tylko te, które spełniają odpowiednie ograniczenia jakościowe (np. związane z wzajemnym wykluczaniem procesów w dostępie do zasobów, nie występowaniem blokad) i ograniczenia ilościowe (np. związane z czasami dostępności zasobów, terminem realizacji zlecenia). Otrzymywane warianty organizacji produkcji nie wymagają dodatkowej weryfikacji (np. na drodze symulacji komputerowej lub w sposób analityczny). Efektywność tych metod jest rezultatem wykorzystania procedur propagacji, które pozwalają na wzajemne ograniczanie się dziedzin zmiennych decyzyjnych. Inaczej mówiąc, jeśli zostaje podstawiona wartość do zmiennej opisującej rozpoczęcie operacji na danym zasobie lub zmiennej opisującej stan zajętości zasobu, to procedury propagacji usuwają z dziedzin pozostałych zmiennych, odnoszących się do tego samego zasobu, wartości mogące doprowadzić do blokady.

Harmonogramy wolne od blokad będą poszukiwane w dziedzinie czasu lub w dziedzinie zdarzeń [4], [5]. Poszukiwanie rozwiązania w dziedzinie czasu charakteryzuje się tym, że zmienne decyzyjne problemu CP określają terminy rozpoczęcia poszczególnych operacji z udziałem odpowiednich zasobów gniazda obróbkowego. W przypadku poszukiwania rozwiązania w dziedzinie zdarzeń zmienne decyzyjne określają liczbę procesów (detali) realizujących równocześnie daną operację (liczba detali znajdujących się na danym zasobie). Rozwiązaniem jest wówczas sekwencja stanów (wektorów) reprezentujących dopuszczalne alokacje zasobowe w systemie. Podejście zdarzeniowe może być rozpatrywane jako podejście czasowe, w którym wszystkie czasy operacji są równe 1, tj. upływ czasu jest liczony od zdarzenia do zdarzenia.

Rozpatrywane zadanie zostało sformułowane poniżej. Dany jest zbiór zasobów produkcyjnych *Z* oraz zbiór procesów *P*. Część zasobów jest wspólna dla operacji realizowanych w ramach różnych procesów. Znane są parametry operacji realizowanych na zasobach. Czy możliwe jest takie zaplanowanie operacji żeby nie nastąpiło zablokowanie systemu, a wszystkie zadania (procesy) zostały zrealizowane w zakładanym horyzoncie planowania? Rozwiązanie tak sformułowanego problemu wymaga rozstrzygania konfliktów zasobowych, tj. określania kolejności operacji konkurujących o dostęp do tych samych zasobów (zasoby krytyczne). Otrzymane wyniki powinny spełniać ograniczenia jakościowe (związane z rozstrzyganiem konfliktów na zasobach krytycznych) oraz ilościowe (związane z oceną uzyskanego rozwiązania wg kryteriów czasowych, kosztowych, itp.)

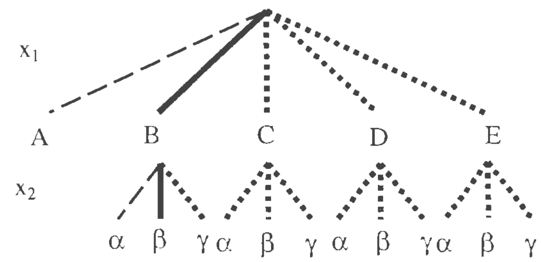
3. PROGRAMOWANIE Z OGRANICZENIAMI

Programowanie z ograniczeniami opiera się na podstawowych mechanizmach: propagacji ograniczeń i dystrybucji zmiennych. Propagacja ograniczeń jest efektywnym mechanizmem wnioskowania opartego na równoległym działaniu propagatorów wymieniających i gromadzących informacje w tzw. zbiorze wymiany ograniczeń. Dystrybucja zmiennych dzieli problem na uzupełniające się wzajemnie elementy według założonej strategii przeszukiwania [2], [4], [5].

Rozwiązanie problemu, jeśli istnieje, jest wyznaczane w kolejnych krokach propagacji i dystrybucji. Dystrybucja zmiennych prowadzi, na ogół, do eksplozji kombinatorycznej liczby rozpatrywanych podproblemów. Jej rozmiary warunkują odpowiednio dobierane heurystyki propagacji i dystrybucji [2], [5].

Poszukiwanie rozwiązania metodami programowania z ograniczeniami (CP) przedstawiono na rys. 2. Rozpatrywany problem opisano za pomocą dwóch zmiennych decyzyjnych x_1, x_2. Dane są dziedziny zmiennych $x_1 \in \{A, B, C, D, E\}$, $x_2 \in \{\alpha, \beta, \gamma\}$.

Pierwszym krokiem poszukiwania rozwiązań jest wstępna propagacja - dla znanych dziedzin zmiennych decyzyjnych analizowany jest zbiór ograniczeń. W dziedzinach zmiennych decyzyjnych pozostawiane są wartości spełniające przynajmniej jedno ograniczenie. Na rys. 2 ilustruje to odcięcie gałęzi A (usunięcie wartości A z dziedziny zmiennej x_1). Następnym krokiem jest podstawianie wartości (distribution). Dla pierwszej ze zmiennych nadawana jest wartość z dziedziny. W rozważanym przypadku x_1=B. Podstawienie prowadzi do rozwinięcia zbioru ograniczeń, dlatego po każdej dystrybucji wartości do zmiennych przeprowadza się propagację ograniczeń. W efekcie w kolejnych krokach dziedziny zmiennych, których wartości nie są podstawione, są zawężane. Jeśli w wyniku podstawienia wartości jednej zmiennej dziedzina innej zmiennej staje się zbiorem pustym, to nie istnieje rozwiązanie spełniające zadane ograniczenia. Wówczas, następuje podstawienie innej wartości dla ostatnio podstawianej zmiennej decyzyjnej, tj. realizowany jest nawrót w drzewie rozwiązań. Dla drzewa pokazanego na rysunku 2 następstwem podstawienia x_1=B i związanej z tym propagacji jest usuniecie wartości α z dziedziny zmiennej decyzyjnej x_2. Kolejnym podstawianiem jest $x_2=\beta$.



Legenda:
—— Gałęzie drzewa rozwiązań odrzucone w procedurze propagacji.
– – Gałęzie wyznaczające rozwiązanie dopuszczalne.
.... Nieprzeszukane gałęzie drzewa rozwiązań.

Rys. 2. Drzewo rozwiązań problemu CP.

Użycie metod programowania z ograniczeniami do rozwiązania przedstawionego problemu wymaga sformułowania trzech rodzajów ograniczeń:
- ograniczenia związane z wymaganym terminem realizacji zleceń,
- ograniczenia związane z kolejnością operacji,
- ograniczenia związane z wykluczaniem się operacji realizowanych na zasobach krytycznych.

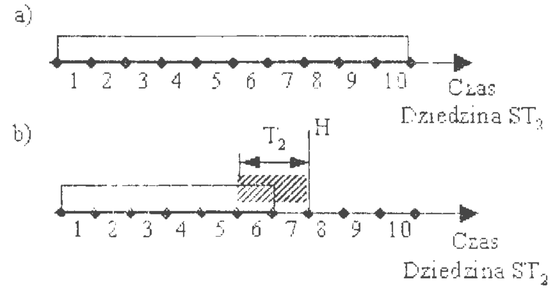
Zmiennymi decyzyjnymi są terminy rozpoczęcia operacji. W dalszych rozważaniach zmienne te oznaczane będą jako ST_n, gdzie n jest indeksem operacji. Wielkością bezpośrednio powiązaną ze zmienną ST_n jest czas trwania danej operacji, oznaczany jako T_n.

Propagację ograniczeń należących do wcześniej wymienionych grup można przedstawić graficznie lub analitycznie. Uwzględniając ograniczenia związane z terminem realizacji całego procesu przyjmuje się, że ostatnia operacja powinna zakończyć się przed zadanym terminem H. Jeśli założy się, że ostatnia operacja trwa T_2=2 umowne jednostki czasu (u.j.c.), wartość H=7, natomiast dziedzina $ST_2 \in \{1,...,10\}$, to propagacja ograniczeń wygląda tak jak na rys. 3.

Formalnie, dla zmiennej ST_n, można takie ograniczenie zapisać następująco:

$$ST_n + T_n - 1 \leq H \tag{1}$$

W szczególności, dla n=2 jest $ST_2 + T_2 - 1 \leq H$, a stąd $ST_2 \leq 6$, tj. $ST_2 \in \{1,...,6\}$; rys. 3b.



Legenda:
- Dziedzina zmiennej ST_2.
- Czas trwania operacji T_2.
H - Termin realizacji zlecenia.

Rys. 3. Propagacja ograniczenia dotyczącego terminu realizacji: a) początkowa dziedzina zmiennej ST_2, b) dziedzina po propagacji.

Analogicznie, uwzględniane są ograniczenia dotyczące kolejności operacji. Przyjmując, że termin rozpoczęcia operacji związanej z czasem ST_2 jest następstwem operacji związanej z czasem ST_1 i trwającej $T_1=3$, formalny zapis ograniczeń wygląda następująco:

$$ST_1 + T_1 - 1 \leq ST_2 \qquad (2)$$

Zakładając dziedzinę $ST_1 \in \{1,...,10\}$ można przeprowadzić następującą propagację ograniczeń. Uwzględniając, że $1 \leq ST_1 < ST_2$ oraz $ST_2 \leq 6$, jest $1 \leq ST_1 < 6$, tj. $ST_1 \in \{1,...,5\}$. Przyjmując $ST_1 = 1$ oraz $T_1=3$, z warunku (2) wynika, że $3 < ST_2$ oraz $ST_2 \leq 6$, a więc $ST_2 \in \{4,5,6\}$.

W kolejnym kroku uwzględniane są ograniczenia dotyczące wykluczania się operacji realizowanych na tym samym zasobie. Jeśli operacja związana z czasem ST_2 jest realizowana na tym samym zasobie, co operacja związana z czasem ST_3 ($T_3=3$), to propagacja ograniczeń rozstrzygających konflikt zasobowy przebiega w oparciu o następujące ograniczenia: $ST_2 \notin \{ST_3,..., ST_3 + T_3 - 1\}$ oraz $ST_3 \notin \{ST_2, ...,ST_2 + T_2 - 1\}$. W ogólnym przypadku formułuje się ograniczenia gwarantujące wzajemne wykluczanie operacji niezależnie od kolejności podstawiania wartości zmiennych decyzyjnych.

$$ST_i \notin \{ST_k, \ldots, ST_k + T_k - 1\} \qquad (3)$$

$$ST_k \notin \{ST_i, \ldots, ST_i + T_i - 1\} \qquad (4)$$

Tak sformułowane ograniczenia pozwalają rozstrzygać w szybki sposób konflikty zasobowe. Dodatkową zaletą przedstawionego podejścia jest przyjęcie czasu jako dziedziny zmiennych decyzyjnych. Oprócz kolejności realizacji operacji na zasobach, otrzymuje się gotowy harmonogram realizacji zadań (procesów)

4. ROZSTRZYGANIE KONFLIKTÓW W DZIEDZINIE CZASU

W celu znalezienia bezblokadowego harmonogramu operacji, spełniającego przyjęte reguły realizacji zadań, przyjęto następujący model predykatowy systemu pokazanego na rys. 1. Zmienna decyzyjna SP_{KX} oznacza termin rozpoczęcia operacji produkcyjnej na zasobie X z udziałem detalu K, natomiast zmienna TP_{KX} oznacza czas operacji produkcyjnej. Zmienna decyzyjna ST_{KX} oznacza termin rozpoczęcia operacji transportowej na zasobie X z udziałem detalu K, a zmienna TT_{KX} oznacza czas operacji transportowej. Dla uproszczenia rozpatrywana jest obróbka L=3 detali, stąd K=1,...,3. Zasoby wykorzystywane przez kolejne operacje są oznaczane jako: *R1*, *B1*, *M*, *B2*, *R2*. Przyjęto, że operacje przenoszenia detali z podajnika na bufor i z bufora na odbierak trwają jedną jednostkę czasu, obróbka detalu trwa cztery jednostki czasu, a czas przeniesienia detalu z bufora na obrabiarkę i z obrabiarki do bufora jest pomijalny. Szczegółowe zestawienie zmiennych i czasów operacji zawarto w tabeli 1.

Tabela 1. Zmienne decyzyjne i czasy operacji

Detal	Operacja	Zasób	Zmienna początku operacji	Zmienna czasu operacji
P_k	*OP1*	*R*	ST_{KR1}	$TT_{KR1}=1$
	OP2	*B*	ST_{KB1}	$TT_{KB1}=0$
	OP3	*M*	SP_{KM}	$TP_{KM}=4$
	OP4	*B*	ST_{KB2}	$TT_{KB2}=0$
	OP5	*R*	ST_{KR2}	$TT_{KR2}=1$

Ponieważ czasy $TT_{KB1} = TT_{KB2} = 0$ spełnione są zależności $(ST_{KB1} = SP_{KM}) \wedge (ST_{KB2} = ST_{KR2})$. Wynika stąd, że wystarczy rozpatrywać tylko ograniczenia dla zmiennych decyzyjnych ST_{KR1}, SP_{KM}, ST_{KR2}. Przy założeniu, że bufor B ma pojemność 1 predykaty ograniczeń uwzględniają:

- kolejność obróbki detali

$ST_{1R1} < ST_{2R1} < ST_{3R1}$, $SP_{1M} < SP_{2M} < SP_{3M}$,
$ST_{1R2} < ST_{2R2} < ST_{3R2}$,

- kolejność operacji podczas obróbki detali

$SP_{1M} \geq ST_{1R1} + TT_{1R1}$, $ST_{1R2} \geq SP_{1M} + TP_{1M}$,
$SP_{2M} \geq ST_{2R1} + TT_{2R1}$, $ST_{2R2} \geq SP_{2M} + TP_{2M}$,
$SP_{3M} \geq ST_{3R1} + TT_{3R1}$, $ST_{3R2} \geq SP_{3M} + TP_{3M}$,

- następstwo operacji transportowej po załadowaniu detalu do bufora lub odłożeniu na odbierak

$ST_{2R1} > SP_{1M}$, $ST_{2R1} > ST_{1R2} + TT_{1R2}$,
$ST_{3R1} > SP_{2M}$, $ST_{3R1} > ST_{2R2} + TT_{2R2}$.

- realizacja zleceń przed końcem terminu H

$ST_{3R2} + TT_{3R2} - 1 \leq H$,

- wykluczanie się operacji

Jest to predykat stosowany dla każdej pary operacji realizowanych na tym samym zasobie przez różne procesy P_K, P_L. Dotyczy on w tej samej marszrucie zasobów *R* i *B*.

- zasób R:

$ST_{KR1} \notin \{ST_{LR1},...,ST_{LR1} + TT_{LR1} - 1\}$,
$ST_{LR1} \notin \{ST_{KR1},...,ST_{KR1} + TT_{KR1} - 1\}$,
$ST_{KR2} \notin \{ST_{LR2},...,ST_{LR2} + TT_{LR2} - 1\}$,
$ST_{LR2} \notin \{ST_{KR2},...,ST_{KR2} + TT_{KR2} - 1\}$ - różne detale nie mogą realizować tej samej operacji;

$ST_{KR1} \notin \{ST_{LR2},...,ST_{LR2} + TT_{LR2} - 1\}$,
$ST_{LR2} \notin \{ST_{KR1},...,ST_{KR1} + TT_{KR1} - 1\}$ - różne detale nie mogą realizować różnych operacji z udziałem tego samego zasobu;
- bufor B (to samo co dla R przy założeniu $ST_{XB1}=SP_{XM} \wedge ST_{XB2}=ST_{XR2}$):
$SP_{KM} \notin \{SP_{LM},...,SP_{LM} + TP_{LM} - 1\}$,
$SP_{LM} \notin \{SP_{KM},...,SP_{KM} + TP_{KM} - 1\}$,
$ST_{KR2} \notin \{ST_{LR2},...,ST_{LR2} + TT_{LR2} - 1\}$,
$ST_{LR2} \notin \{ST_{KR2},...,ST_{KR2} + TT_{KR2} - 1\}$
oraz
$SP_{KM} \notin \{ST_{LR2},...,ST_{LR2} + TT_{LR2} - 1\}$,
$ST_{LR2} \notin \{SP_{KM},...,SP_{KM} + TP_{KM} - 1\}$.
Do wyznaczenia dopuszczalnych wartości zmiennych decyzyjnych przedstawionego problemu wykorzystano język programowania w logice z ograniczeniami Oz [3], [4], [5]. Przyjęto horyzont planowania H=20 ujednoliconych jednostek czasu (u.j.c.).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
R	P					O	P					O	P					O		
M		1	1	1	1			2	2	2	2			3	3	3	3			
B																				

Rys. 4. Harmonogram bezblokadowy dla B=1, L=3.

W efekcie działania procedur propagacji ograniczeń i podstawiania wartości zmiennych otrzymano terminy rozpoczęć operacji na zasobach definiujące bezblokadowy harmonogram realizacji procesów produkcyjnych w zadanym horyzoncie planowania (rys. 4).

5. ROZSTRZYGANIE KONFLIKTÓW W DZIEDZINIE ZDARZEŃ

W celu znalezienia w systemie (rys. 1) bezblokadowego harmonogramu operacji w dziedzinie zdarzeń założono, że każda operacja trwa jedną umowną jednostkę czasu. Zatem można przyjąć, że upływ czasu jest liczony od zdarzenia do zdarzenia. Przyjęto, że każdej operacji odpowiada wektor zmiennych decyzyjnych $X_{i,j}$, gdzie i=1,...,5 oznacza realizowaną operację, j – kolejne zdarzenie w systemie zmiennej decyzyjnej, j=1,...,H, gdzie H – horyzont planowania (liczba zdarzeń w systemie; maksymalna wartość H jest równa liczbie operacji pomnożonej przez liczbę detali). Wartości zmiennych określają liczbę procesów (detali) zajmujących dany zasób (dla jednostkowych pojemności zasobów wartości zmiennych są binarne). Zatem 1 oznacza, że kolejny detal został umieszczony w określonym miejscu (odpowiednio dla operacji *OP1* – jest to chwytak robota, dla operacji *OP2* bufor obrabiarki, *OP3* – obrabiarka, *OP4* – bufor, *OP5* – chwytak robota). Stan systemu dla zdarzenia o numerze j jest reprezentowany za pomocą wektora $X(j) = (X_{1,j}, X_{2,j}, X_{3,j}, X_{4,j}, X_{5,j})$.

Stan blokady wystąpi, gdy z zapełnionego nieobrobionymi detalami bufora zostanie pobrany element na obrabiarkę, a na zwolnione w ten sposób miejsce zostanie dostarczony nowy detal. Nastąpi sytuacja, w której nie będzie możliwości „powrotu" obrobionego detalu do bufora. W celu wyeliminowania sytuacji tego typu sformułowane zostały następujące predykaty.

Wykluczanie się operacji robota (może on albo pobrać detal z podajnika, albo odłożyć obrobiony)

$$\underset{j=1..H}{\forall} X_{1,j} + X_{5,j} \leq 1 \tag{5}$$

Różnica między liczbą elementów dostarczonych z podajnika do bufora i odłożonych po obróbce bufora na odbierak nie powinna przekraczać pojemności bufora.

$$\underset{j=1..H}{\forall} \sum_{k=1}^{j} X_{1,k} - \sum_{k=1}^{j} X_{5,k} \leq B \tag{6}$$

Ale powinna być większa lub równa zeru

$$\underset{j=1..H}{\forall} \sum_{k=1}^{j} X_{1,k} - \sum_{k=1}^{j} X_{5,k} \geq 0 \tag{7}$$

Ponadto suma detali odebranych do chwili t z danej lokalizacji nie powinna być większa niż suma detali dostarczonych z operacji poprzedzającej do chwili t-1.

$$\underset{j=2..H,\, o=2..5}{\forall} \sum_{k=1}^{j-1} X_{o-1,k} - \sum_{k=1}^{j} X_{o,k} \geq 0 \tag{8}$$

Detal jest albo przenoszony na obrabiarkę albo z niej odbierany

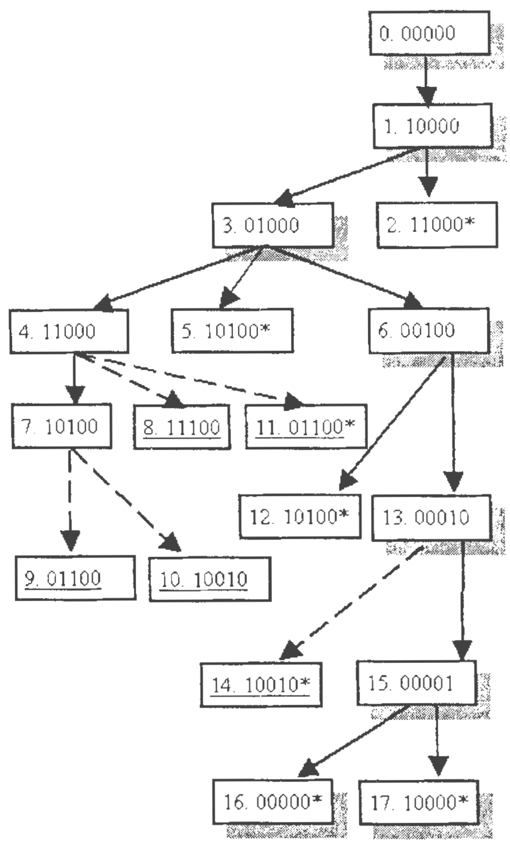
$$\underset{j=1..H}{\forall} X_{3,j} + X_{4,j} \leq 1 \tag{9}$$

Suma detali, które „przejdą" przez daną operację jest równa liczebności realizowanej partii L.

$$\underset{o=1..5}{\forall} \sum_{j=1}^{H} X_{o,j} = L \tag{10}$$

Przedstawione ograniczenia zostały zaimplementowane w języku Oz, który umożliwia korzystanie z procedur propagacji i podstawiania [4], [5].

Drzewo stanów dla systemu z buforem o pojemności 1 i chwytakiem o pojemności 1 pokazano na rys. 5. Każdy stan jest zapisany w notacji xx.xxxxx, gdzie dwie pierwsze cyfry oznaczają numer stanu, następne 5 cyfr (po kropce rozdzielającej) podaje ile detali znajduje się w poszczególnych lokalizacjach rozpatrywanego gniazda obróbkowego. Przyjęto, że zmiana stanu wiąże się ze zmianą wartości na dowolnej pozycji wektora stanu lub na kilku pozycjach równocześnie (np. zmiana stanu z 01000 na 10100 jest związana z zajęciem obrabiarki M i zwolnieniem bufora B oraz z rozpoczęciem operacji przenoszenia detalu z podajnika P do bufora B przez robota R). Stany oznaczone cieniem oznaczają sekwencję dopuszczalną, gwarantującą bezblokadową realizację zadań (rys. 5).



Legenda:
xx . xxxxx - stan akceptowany
xx . xxxxx* - stan już wystąpił
xx . xxxxx - stan blokady

Rys. 5. Drzewo stanów dla B=1. Cieniem zaznaczono stany wyszukiwane przez system Mozart.

6. PODSUMOWANIE

Przedstawiona metoda programowania z ograniczeniami (CP) umożliwia znalezienie harmonogramów dopuszczalnych, spełniających zadane ograniczenia jakościowe oraz ilościowe w systemie produkcyjnym zagrożonym blokadą.

Poszukiwanie rozwiązań zrealizowano w dziedzinie czasu i w dziedzinie zdarzeń. Podejście oparte o wykorzystanie dziedziny czasu umożliwia wyznaczenie gotowych harmonogramów operacji uwzględniających wielkości partii produkcyjnych i czasy operacji. Poszukiwanie rozwiązań w dziedzinie zdarzeń jest podejściem ogólniejszym, gdyż pozwala wyznaczyć wszystkie dopuszczalne sekwencje stanów prowadzące do stanu zakończenia realizacji zadań. Struktura przestrzeni stanów nie zależy od liczby realizowanych zadań, a jedynie od parametrów zasobów i realizowanych marszrut produkcyjnych. W efekcie zmiana liczby zleceń nie wymaga ponownego generowania przestrzeni stanów. Jednak w przypadku uwzględniania czasów operacji zmiana ich parametrów nie gwarantuje, że wyznaczona sekwencja alokacji zasobowych nadal będzie spełniała zadane ograniczenia ilościowe.

W przypadku stosowania metod CP nie jest konieczne weryfikowanie rozwiązania po jego znalezieniu, gdyż z definicji spełnia ono wszystkie ograniczenia. Jeśli sformułowane predykaty ograniczeń stanowią warunki konieczne i wystarczające dla unikania blokad możliwe jest mówienie o poszukiwaniu rozwiązań optymalnych w sensie poszukiwania ekstremum globalnego, gdyż metoda pozwala wyznaczyć wszystkie rozwiązania (harmonogramy) dopuszczalne.

SOLVING RESOURCE CONFLICT RESOLUTION TASKS USING CONSTRAINT PROGRAMMING TECHNIQUES

Abstract: This work presents constraint logic programming method (CLP) applied to determine material flows in concurrent production processes. The rules of resource conflict resolution have been expressed by means of a set of predicates, which define constraints for the problem decisive variables. Predicate model has been defined in time and event domains and implemented in procedures of propagating and distributing the variable values. These procedures allow for fast exploration of admissible solutions within the CP problem search tree. The predicates defined represent sufficient conditions to determine deadlock-free schedules of operations, which meet the prescribed qualitative and quantitative constraints.

Literatura

[1] Banaszak Z., Krogh B. (1990) Deadlock Avoidance in Flexible Manufacturing Systems with Concurrently Competing Process Flows. *IEEE Trans. On Robotics and Automation*, Vol. 6, No. 6, 724-734.

[2] Barták R. (2003) Constraint-based scheduling: An introduction for newcomers. *Preprints of the 7th IFAC Workshop on Intelligent Manufacturing Systems*, Budapest, Hungary, 75-80.

[3] Benhamou F. (1995) Interval constraint logic programming. In Andreas Podelski, editor, Constraints: Basics and Trends. *Lecture Notes in Computer Science*, Vol. 910, Springer Verlag, Berlin, 1-21.

[4] Bzdyra K., Sitek P., Banaszak Z. (2004) CLP-based approach to planning of production flow. *Konferencja Naukowo – Techniczna "Automatyka – Nowości i perspektywy AUTOMATION 2004"*, Warszawa, 344-352.

[5] Schulte Ch., Smolka G., Wurts J. (1998) *Finite Domain Constraint Programming in Oz*. DFKI OZ documentation series, German Research Center for Artificial Int., Saarbrucken, Germany.

[6] Wójcik R. (1999) Wyznaczanie przestrzeni stanów bezpiecznych w systemach procesów współbieżnych. *Zeszyty Naukowe Wydz. Mech. nr 26, Politechnika Koszalińska, Materiały XVII Ogólnopolskiej Konferencji Polioptymalizacja i Komputerowe Wspomaganie Projektowania*, Mielno '99, Koszalin-Mielno, 325-332.

Instytut Badań Systemowych
Polskiej Akademii Nauk

ISBN 83-89475-01-4