

Raport nr 1 z projektu Systemy Zdarzeniowe

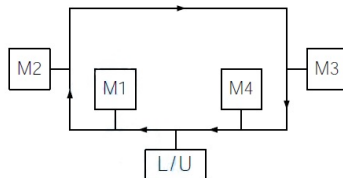
Marcin Ciopcia Daniel Gut Piotr Semberecki Hanna Sienkiewicz
Mateusz Stachowski

25 stycznia 2015

1 Problem Projektu

1.1 Opis ogólny

Pierwszym celem projektu jest symulacja zautomatyzowanej linii produkcyjnej z wieloma gniazdami wytórczymi, obsługiwanej przez wózki AGV. Linia produkcyjna ma kształt pierścienia, z ustalonym dozwolonym kierunkiem ruchu wózków AGV, poruszających się jak na rysunku 1.



Rysunek 1: Struktura ścieżek

Modelowany system posiada stację załadowczo-rozładowczą (L/U), cztery stacje maszynowe (M_1, M_2, M_3, M_4), z których każda posiada bufor wejściowy i wyjściowy oraz robota wykonującego operację przeniesienia detalu z bufora wejściowego na stanowisko robocze maszyny oraz ze stanowiska roboczego do bufora wyjściowego. Opisany system produkcyjny realizuje wspólnie do sześciu typów produktów w podanej ilości. Operacje na danym produkcie zostają wykonane na różnych maszynach w zadanej kolejności.

Następnie zostanie opracowany sterownik zdarzeniowy zarządzający logiką przepływu zadań w systemie składającą się z

- klasyfikatora zadań możliwych do realizacji,
- systemu agentowego zarządzającego kolejnością wykonywania działań,
- systemu przeciwdziałania zakleszczeniom zdarzeń w systemie.

Sterownik nadrzędny opisany zostanie za pomocą sieci Petriego, następnie zostanie oprogramowany w środowisku *Matlab*.

1.2 Zastosowanie systemów zdarzeniowych

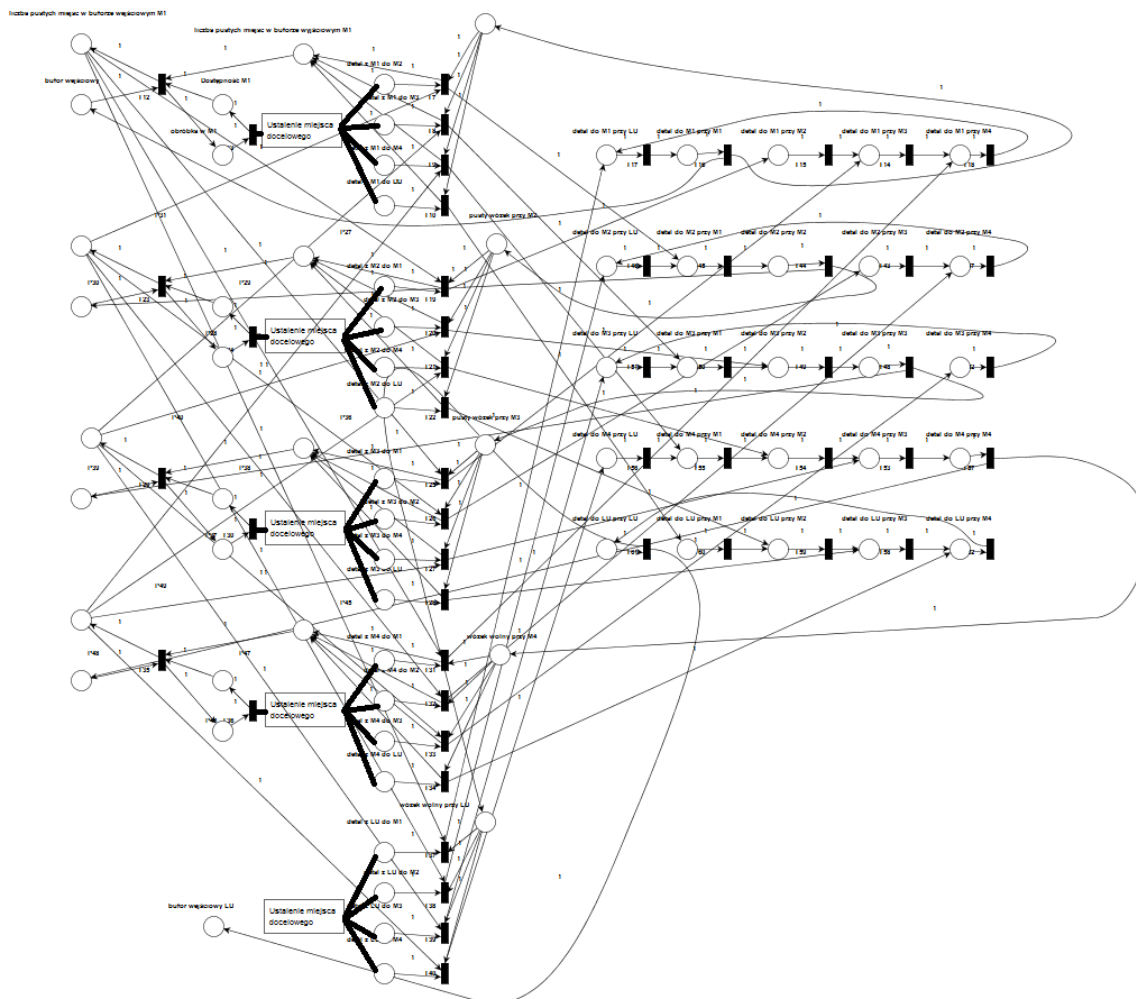
Podjęcie tego zagadnienia jest ważne ze względu na poszerzające się zastosowanie robotyzacji w wielu dziedzinach gospodarki. Za pomocą elastycznych komórek produkcyjnych (ESP) można modelować procesy produkcyjne, przepustowość sieci infrastruktury technicznej, systemy czasu rzeczywistego [1], sterowniki logiczne [2] czy sieć komunikacji miejskiej. Modele różnorodnych systemów pozwalają ocenić ich efektywność funkcjonowania [3], co może wpłynąć na zmniejszenie kosztów, niwelując błędy już przy projektowaniu danego systemu. Za pomocą systemów zdarzeniowych można modelować również poruszanie się robotów mobilnych, zapobiegając przy tym zakleszczeniom, kolizjom, optymalizując czas wykonania danego zadania czy przejazdu przez daną ścieżkę.

1.3 Upowszechnienie wyników

Wyniki projektu będą upowszechnione na serwisie www, który będzie zawierał

- archiwum z oprogramowaniem,
- dokumentację algorytmów,
- dokumentację oprogramowania dla użytkowników i deweloperów,
- przykład działania systemu,
- wyniki przeprowadzonych badań, w tym
 - zbadanie poprawności działania systemu wykorzystując dostarczone dane wejściowe,
 - wpływ ilości robotów znajdujących się na maszynach
 - wpływ pojemności buforów wejściowych oraz wyjściowych maszyn na działanie systemu wpływ ilości robotów obsługujących operacje przeniesienia detalu z bufora wejściowego do miejsca operacyjnego maszyny jak i z miejsca operacyjnego maszyny do bufora wyjściowego na działanie systemu,
 - dobór algorytmu przeciwdziałającemu blokadom,
 - wpływ pojemności wózków AGV na działanie systemu.

1.4 Sieć Petriego

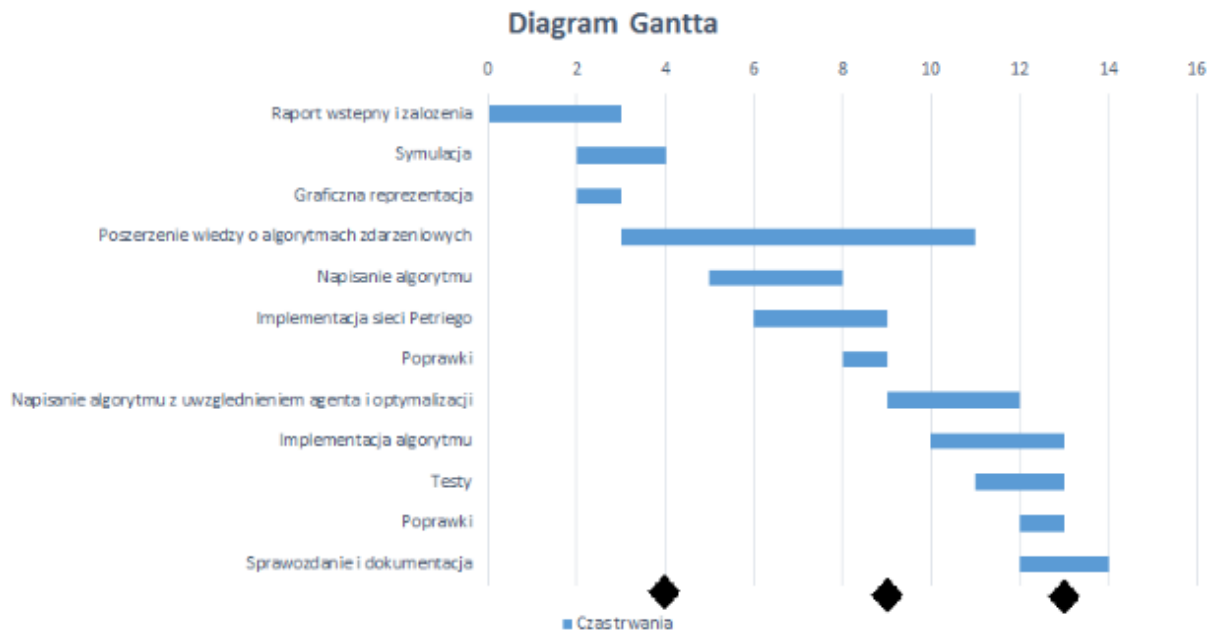


Rysunek 2: Sieć Petriego

Wykorzystaliśmy standardowe oznaczenia miejsc i tranzycji w sieciach Petriego. Wyróżniliśmy operacje maszynowe oraz realizowane przez wózki AGV, które są obserwowalne przez sterownik główny, po zakończeniu danego zadania maszyna/wózek informuje o tym kontroler główny, który może zlecić nowe zadanie dla maszyny/wózka (zdarzenie kontrolowalne).

2 Plan pracy i rozkład w czasie

Na rysunku 3 znajdują się wyselekcjonowane zadania oraz czas ich trwania jak również zaznaczone zostały kamienie milowe.



Rysunek 3: Diagram Gantta

3 Zarządzanie projektem

Spotkania zespołu odbywać się będą w terminie zajęć oraz po wcześniejszym umówieniu się członków zespołu przy użyciu *Skype'a* oraz grupy na portalu społecznościowym *Facebook*. Oficjalnym liderem grupy jest Hanna Sienkiewicz. Jest to osoba, która jest odpowiedzialna za rozliczanie z zadań członków grupy przed upływem wyznaczonych terminów. Podział zadań w grupie następuje zgodnie z umiejętnościami i oczekiwaniami poszczególnych członków grupy. Nie przewidujemy problemów z tym związanych, członkowie grupy czują się zobligowani do pomocy przy większych i trudniejszych zadaniach.

4 Zespół

W skład zespołu projektowego wchodzi:

- Hanna Sienkiewicz, lider grupy, dane kontaktowe: 184184@student.pwr.edu.pl,
- Marcin Ciopcia,
- Daniel Gut,
- Piotr Semberecki,
- Mateusz Stachowski.

5 Struktura systemu

W systemie zdarzeniowym można wyróżnić

- sterownik główny,
- sterowniki maszyn,
- sieć Petrie'go,
- symulator zdarzeń.

Symulator ma za zadanie zasymulować uruchomienie zadania na maszynie, ładowanie z bufora do maszyny oraz ładowania z maszyny na wózek.

Pierwszy model sterownika działa na zasadzie wykonywania zadań jako pierwszych, których suma czasów wykonywania poszczególnych etapów jest najmniejsza. Posiada on wiedzę o zadaniach, jakie mogą być wykonane w danym kroku czasowym oraz o maszynach, na których będą wykonywane poszczególne etapy zadania jak i o czasach wykonania poszczególnych etapów zadania.

Sieć (P/T Sieć) formalnie można zapisać za pomocą piątki $N = (P, T, F, W, M_0)$, gdzie:

- P jest skończonym zbiorem miejsc,
- T jest skończonym zbiorem przejść (tranzycji),
- $P \cap T = \{\emptyset\}$ - zbiory miejsc i przejść są rozłączne,
- $F \subseteq (P \times T) \cup (T \times P)$ jest relacją przepływu,
- $W : F \rightarrow (\mathcal{N} - \{0\})$ jest funkcją wagi łuków,
- $M_0 : P \rightarrow \mathcal{N}$ jest markowaniem początkowym.

6 Sekwencyjny system alokacji zasobów

RAS definiuje się jako piątkę $\Phi = (\mathcal{R}, C, \mathcal{P}, \mathcal{D}, T)$, gdzie:

- \mathcal{R} jest zbiorem typów zasobów,
- C jest funkcją pojemności zasobów,
- \mathcal{P} - jest zbiorem typów procesów,
- $F \subseteq (P \times T) \cup (T \times P)$ jest relacją przepływu,
- $W : F \rightarrow (\mathcal{N} - \{0\})$ jest funkcją wagi łuków.
- $\mathcal{R} = \{M_i, B_{wy_i}, B_{we_i}, L/U, AGV_j, W_j\}$, $i = 1, \dots, 4$, $j = 1, \dots, 4$, gdzie M - zasób maszynowy, B - zasób buforowy, L/U - zasób stacji, AGV - zasób wózków, W - zasób wolnego miejsca między maszynami.
- $C(M_i) = 1$, $C(B_{we_i}) = x$, $C(B_{wy_i}) = y$, $C(AGV_i) = z$, $C(L/U) = \infty$, $C(W_j) = w$; x, y, z, w - parametry,
- $\mathcal{P} = \{\Pi_i\}$, $i = 1, \dots, 6$, (6 procesów to 6 typów produktów),
- $\Pi_j = \langle \mathcal{S}_j, \mathcal{G}_j \rangle$,
- $\mathcal{S}_j = \{Om_{j1}, \dots, Om_{jn(j)}, Ow_{j1}, \dots, Ow_{jm(j)}\}$ zbiór operacji procesu Π_j , Om - operacje maszynowe, Ow - operacje transportowe,
- $\mathcal{G}_j = \{O, t\}$ jest strukturą danych wyznaczającą logikę wykonywania operacji procesu Π_j ; O to zbiór operacji, t czas operacji O
- \mathcal{D} : jest funkcją wymagań zasobowych operacji, operacje transportu wymagają dostępności wózków oraz wolnej ścieżki między danymi maszynami, operacje maszynowe wymagają miejsca w buforze wejściowym oraz wyjściowym jak i dostępności maszyny; np. $D_{11} = \{B_{we1}, M_1, B_{wy1}\}$ - wymagania zasobowe operacji maszynowej Om_{11} .

SU-RAS, w którym funkcja wymagań zasobowych jest określona przez D . Każda operacja O wymaga jednej jednostki jednego typu zasobu.

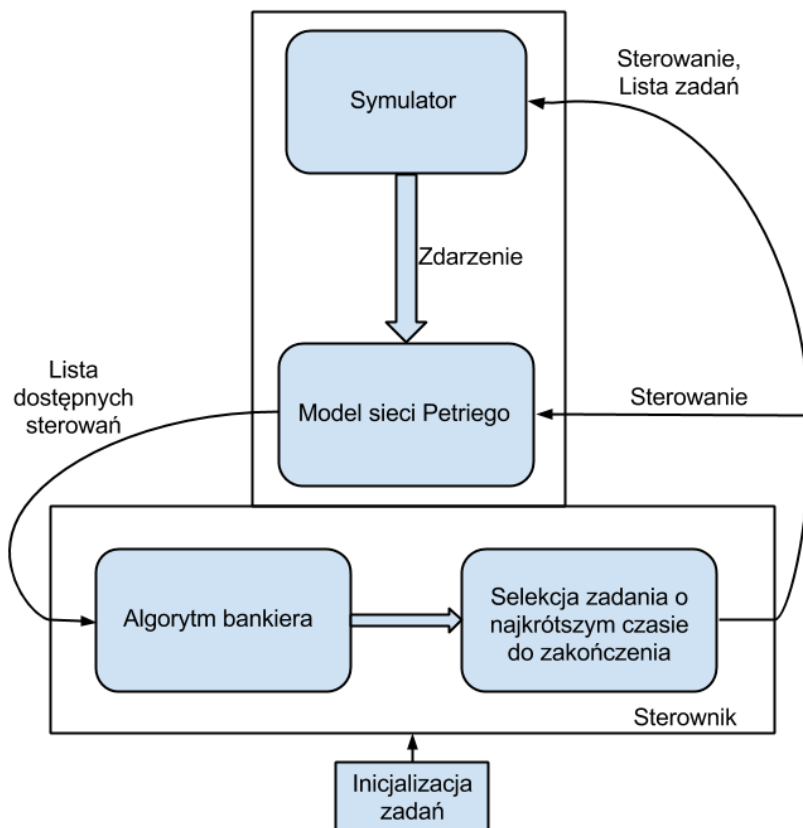
Jeżeli detal musi zostać przewieziony z bufora wyjściowego maszyny X na bufor wejściowy maszyny Y to żąda on wolnego miejsca na trasie między tymi maszynami i jednocześnie wózka, który go przewiezie, to oznacza, że jeżeli wózek i wolne miejsce na trasie to zasoby to operacja transportu wymaga dwóch jednostek zasobów różnych typów (każda operacja wymaga jednej jednostki jednego typu zasobu).

7 Unikanie blokad

W celu uniknięcia blokad stosujemy pojemność buforów wejściowych i wyjściowych maszyn większą niż 1. Zastosowaliśmy również w sterowniku główny algorytm bankiera, który służy do sprawdzenia, czy dany stan jest bezpieczny, jeżeli tak to sterownik pozwala na realizację danego zadania, które doprowadza się do tego stanu bezpiecznego. Algorytm ten polega na skonstruowaniu ciągu bezpiecznego, tzn. takiego ciągu procesów $P(1), P(2), \dots, P(n)$, w którym proces $P(i)$ żąda zasobów wolnych w danym stanie lub zajętych przez procesy począwszy od $P(1)$ do $P(i-1)$.

8 Opis sterowania

Rysunek nr 4 przedstawia działanie modelu zdarzeniowego.



Rysunek 4: Architektura sterownika

8.1 Inicjalizacja zadań

Zadanie to struktura wchodząca w skład listy zadań składająca się z

- numerów maszyn, na których są wykonywane poszczególne etapy zadań
 $Lista_zadan(i).maszyny = [M_j, \dots, M_n]$,

- czasów obróbki jednego detalu na maszynie
 $Lista_zadan(i).czasy = [t_j, \dots, t_n]$,
- liczby detali oczekujących na obróbkę na maszynie
 $Lista_zadan(i).ilosc = [k, \dots, 0]$.

8.2 Sterownik

Sterownik na podstawie listy dostępnych sterowań wygenerowanych przez model sieci Petriego, listy zadań oraz wielkości buforów wyjściowych i wejściowych maszyn generuje **Sterowanie**. **Sterowanie** to struktura zawierająca w sobie

- typ zadania (typy zadań zostały opisane w sekcji 8.3.1),
- numer maszyny - określa nr maszyny, przy której zdarzenie wystąpiło,
- numer - to numer zadania z listy zadań,
- etap - to numer etapu zadania.

Sterowanie generowane jest poprzez algorytm bankiera, następnie z pośród możliwych zadań wybierane jest to, które ma najkrótszy czas do zakończenia. Algorytm bankiera jest zaimplementowany dla maszyn z zadania. Agent przyjmuje jako parametry

- listę dostępnych sterowań wygenerowaną przez model sieci Petriego, zawierającą możliwe **Sterowania**,
- listę zadań.

8.3 Model sieci Petriego - opis stanu sieci

Stan sieci to macierz rozmiaru 11x5. Stan początkowy wygląda następująco

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ b_{we} & b_{we} & b_{we} & b_{we} & \infty \\ b_{wy} & b_{wy} & b_{wy} & b_{wy} & \infty \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (1)$$

gdzie b_{we} , b_{wy} to odpowiednio rozmiar bufora wejściowego i wyjściowego, a

1. wiersz sieci to transport do maszyny 1,
2. wiersz sieci to transport do maszyny 2,
3. wiersz sieci to transport do maszyny 3,
4. wiersz sieci to transport do maszyny 4,
5. wiersz sieci to transport do LU,
6. wiersz sieci to oznaczenie miejsca z pustym wózkiem (domyślnie jeden pusty wózek przy LU),
7. wiersz sieci to ilość detali w buforach wejściowych (domyślnie zero dla każdej maszyny),
8. wiersz sieci to liczby pustych miejsc w buforach wejściowych dla każdej z maszyn,
9. wiersz sieci to liczby pustych miejsc w buforach wyjściowych dla każdej z maszyn,
10. wiersz sieci to dostępność maszyn (domyślnie wszystkie maszyny są dostępne),
11. wiersz sieci to obróbka na maszynie.

Przykładowo transport do maszyny nr 4 z maszyny nr 3 to $S[4][3]=1$. Po wykonaniu tego transportu mamy $S[4][3]=0$ oraz $S[4][4]=1$. Gdy następuje transport z maszyny nr 4 na maszynę nr 4 to mamy $S[4][4]=0$, $S[7][4] = 1$ (detal w buforze wejściowym maszyny nr 4) oraz $S[6][4] = 1$ (pusty wózek przy maszynie nr 4).

8.3.1 Funkcje występujące w modelu sieci Petriego

Tutaj zostało opisane działanie sieci Petriego wraz z obrazującym to działanie przykładem.

Istnieją 4 typy zadań

- Typ 3 - Załaduj wózek, Zgłaszamy, że detal nie jest już w gnieździe maszyny, maszyna ta staje się dostępna. Sprawdzamy czy mamy pusty wózek przy maszynie i czy jest miejsce w buforze wejściowym maszyny docelowej. Jeśli tak to rezerwujemy wózek. Zmniejszamy liczbę dostępnych miejsc w buforze wejściowym maszyny docelowej o 1. Zwiększamy miejsce w buforze wejściowym dla maszyny, która opuszcza detal. W wierszu o numerze maszyny docelowej, zaznaczamy transport do niej począwszy od maszyny aktualnej. W takim stanie możliwy jest transport. Do **Listy dostępnych zadań** dodawane jest **Sterowanie** - transport.

$$S_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ b_{we} & b_{we} & b_{we} & b_{we} & \infty \\ b_{wy} & b_{wy} & b_{wy} & b_{wy} & \infty \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$S_1[6][1] = 1$ jest pusty wózek przy maszynie nr 1. $S_1[10][1] = 0$ maszyna nie jest dostępna. $S_1[11][1] = 0$ obróbka detalu trwa. $S_1[8][3] = 1$ jest miejsce w buforze wejściowym maszyny docelowej transportu (w tym przykładnie maszyna nr 3). $S_1[8][3] > 0$ bufor wejściowy maszyny docelowej nie jest pełny. Po zgłoszeniu zadania "Załaduj wózek" mamy

$$S_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ b_{we} & b_{we} & b_{we} - 1 & b_{we} & \infty \\ b_{wy} + 1 & b_{wy} & b_{wy} & b_{wy} & \infty \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$S_2[6][1] = 0$ rezerwujemy wózek. $S_2[10][1] = 1$ maszyna jest znów dostępna. $S_2[11][1] = 0$ obróbka zakończona. $S_2[3][1] = 1$ zaznaczamy transport z maszyny nr 1 na maszynę docelową (nr 3). $S_2[8][3] = b_{we} - 1$ zmniejszamy liczbę dostępnych miejsc w buforze wejściowym maszyny docelowej. $S_2[9][1] = b_{wy} + 1$ zwiększamy liczbę dostępnych miejsc w buforze wyjściowym maszyny nr 1. Możliwe jest **Sterowanie** - transport (dodawane do **Listy dostępnych zadań**).

- Typ 2 - Wykonaj transport

$$S_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ b_{we} & b_{we} & b_{we} & b_{we} & \infty \\ b_{wy} & b_{wy} & b_{wy} & b_{wy} & \infty \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$S_3[3][1] = 0$, $S_3[3][2] = 1$ został wykonany transport z maszyny nr 1 na maszynę nr 2, dalej wykonujemy transport z maszyny nr 2 na maszynę nr 3.

$$S_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ b_{we} & b_{we} & b_{we} & b_{we} & \infty \\ b_{wy} & b_{wy} & b_{wy} & b_{wy} & \infty \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$S_4[3][2] = 0$, $S_4[3][3] = 1$ został wykonany transport z maszyny nr 2 na maszynę nr 3. Po wystąpieniu dalej wykonujemy transport z "maszyny nr 3 na maszynę nr 3".

$$S_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ b_{we} & b_{we} & b_{we} & b_{we} & \infty \\ b_{wy} & b_{wy} & b_{wy} & b_{wy} & \infty \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$S_5[3][3] = 0$, $S_5[6][3] = 1$ mamy pusty wózek przy maszynie nr 3, $S_5[7][3] = 1$ mamy detal w buforze wejściowym maszyny nr 3.

- Typ 1 - Załaduj detal na maszynę. Jeśli jest to ostatni etap zadania i maszyna docelowa to LU, to zadanie zostaje zakończone. Zgłaszamy pusty wózek. Zwiększamy liczbę miejsc w buforze wejściowym maszyny o 1. Zgłaszamy **Sterowanie** - możliwość obróbki (do **Listy dostępnych zadań**).

$$S_6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ b_{we} & b_{we} & b_{we} + 1 & b_{we} & \infty \\ b_{wy} & b_{wy} & b_{wy} & b_{wy} & \infty \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$S_6[8][3] = b_{we} + 1$ zwiększamy liczbę miejsc w buforze wejściowym maszyny o 1. $S_6[7][3] = 0$ - przeniesienie z bufora wejściowego maszyny nr 3 do jej gniazda obróbczego.

- Typ 4 - Wykonaj obróbkę. Do odpowiedniej kolumny wiersza 11 jest wpisywana jedynka. Zostaje

zwolniony wózek.

$$S_7 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ b_{we} & b_{we} & b_{we} & b_{we} & \infty \\ b_{wy} & b_{wy} & b_{wy} & b_{wy} & \infty \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$S_7[11][3] = 1$ obróbka na maszynie nr 3. $S_7[10][3] = 0$ maszyna nr 3 nie jest wolna. Zgłaszamy możliwość **Sterowania** - załadunek wózka (do **Listy dostępnych zadań**), po wykonanej obróbce.

8.4 Symulator

Literatura

- [1] S. Samolej, B. Trybus, *Zastosowanie kolorowanych sieci Petriego w projektowaniu systemów czasu rzeczywistego*. Pomiary, Automatyka, Kontrola, 2005, tom R. 51, nr 1, 11-13.
- [2] I. Grobelna, M. Grobelny, *Projektowanie sterowników logicznych z wykorzystaniem łuków zezwalających i zakazujących sieci Petriego*. Pomiary, Automatyka, Kontrola, 2012, tom R. 58, nr 7, 605-607.
- [3] G. Bocewicz, W. Muszyński, Z. Banaszak, *Modele multimodalnych sieci i procesów transportowych*. Postępy robotyki pod redakcją K. Tchonja i C. Zielińskiego, Oficyna wydawnicza Politechniki Warszawskiej, Warszawa 2014, tom 2, s. 543-552.