

Projekt symulacji oraz optymalizacji linii produkcyjnej, za pomocą sieci BCMP

Nazwa przedmiotu:	Systemy i sieci kolejkowe
Kierunek:	AGH, Automatyka i robotyka 2016/2017
Osoby wykonujące:	<ul style="list-style-type: none">• Świderek Piotr,• Noga Jakub,• Dubański Jakub,• Nerushkin Kostya,• Kasprowicz Marcin,• Turbak Karol,• Trąd Antoni.
Data oddania:	21.01.2016r
Ocena:	

Spis treści:

1	Tematyka projektu:	3
2	Opis matematyczny sieci i wyznaczanie wartości charakterystycznych	4
2.1	System otwarty.....	4
2.1.1	Średni współczynnik wizyt zadania w systemie.....	4
2.1.2	Współczynnik przychodzenia zgłoszeń - metoda dokładna	5
2.1.3	Średnia ilość zgłoszeń w systemie:.....	5
2.2	System zamknięty.....	6
2.2.1	Średni współczynnik wizyt zadania w systemie.....	6
2.2.2	Współczynnik przychodzenia zgłoszeń - metoda SUM.....	7
2.2.3	Średnia ilość zgłoszeń w systemie:.....	8
2.3	Wspólne parametry charakterystyczne	9
3	Opis implementacji systemu	10
3.1	Budowa rozwiązania:.....	10
3.1.1	Klasa sieci: Network_nClass:	10
3.1.2	Skrypt konfiguracji.....	10
3.1.3	Skrypt symulacji.....	11
3.2	Instrukcja obsługi:	11
3.3	Przykładowy zapis konfiguracji sieci:.....	11
4	Przykład - sformułowanie problemu BCMP	13
4.1	Opis słowny:	13
4.2	Założenia i ograniczenia:	13
4.3	Podział na systemy kolejkowe:.....	13
4.3.1	Stacje:	13
4.3.2	Współczynniki obsługi poszczególnych klas w stacjach:	14
4.4	Podział na klasy wewnątrz systemu i współczynniki przychodzenia zgłoszeń:	14
4.5	Diagram:	14
4.6	Macierze przejścia do kolejnych stacji	14
5	Problem optymalizacji parametrów sieci	16
5.1	Opis algorytmu karalucha:.....	16
5.2	Funkcja oceny rozwiązania ogólnie:	17
5.2.1	Funkcja:.....	17
5.3	Wyniki optymalizacji:.....	17
6	Podsumowanie i wnioski	20

1 Tematyka projektu:

Przedmiotem projektu było stworzenie platformy przeznaczonej do symulacji rozbudowanych sieci kolejkowych. Program miał dostarczyć użytkownikowi możliwość łatwej konfiguracji, obliczania parametrów charakterystycznych oraz optymalizacji pod względem ilości stanowisk w każdym z systemów mając na uwadze wcześniej zdefiniowaną funkcję oceny sieci.

Wymagania, stawiane wobec programu były następujące:

- Obsługa dwóch typów sieci: zamkniętej i otwartej,
- Konfiguracja obsługująca dowolną ilość systemów jednego z czterech typów,
- Definiowalna ilość stanowisk w każdym z systemów,
- Obsługa wielu klas zgłoszeń w sieci,
- Dowolna infrastruktura połączeń systemów,
- Dowolna ilość zgłoszeń obecnych/przychodzących do sieci,
- Łatwa w określeniu funkcja oceny sieci stanowiącą podstawę dla metody optymalizacyjnej
- Optymalizacja za pomocą algorytmu karalucha.

W celu spełnienia tych oczekiwań wykorzystano wzory i algorytmy wyprowadzone na wykładach oraz dokładniej dowodzone w książce: „Queueing Networks and Markov chains- modeling and performance evaluation with computer science”, G.Bolch, S.Greiner, H.Meer, K.Trivedi 2006.

Podstawowe wzory i zależności zostały opisane w rozdziale 2. Wskazano tam wzory na obliczenie podstawowych parametrów sieci za pomocą metod dokładnych (sieci otwarte) oraz metody przybliżonej SUM (dla sieci zamkniętej).

Rozdział 3 stanowi swoistą dokumentację programu oraz instrukcje, w jaki sposób należy wprowadzić potrzebne parametry wejściowe do symulacji.

W rozdziale 4 zaprezentowano rzeczywisty przykład sieci otwartej przedstawiający linię produkcyjną w fabryce o niewielkim rozmiarze.

W końcowym rozdziale 5 opisano sposób na rozwiązanie problemu optymalizacji wielu zmiennych przy pomocy algorytmu karalucha.

Ostatni rozdział stanowi podsumowanie całego projektu oraz powiązane z nim wnioski.

2 Opis matematyczny sieci i wyznaczanie wartości charakterystycznych

Sieć BCMP – nazwa pochodzi od nazwisk współtwórców tego podejścia (Baskett, Chandy, Muntz, Palacios). Stanowi ona rozszerzenie definicji sieci kolejkowych, które pozwala zdefiniować wiele systemów wchodzących w jej skład.

Elementami bazowymi sieci stanowią 4 typy systemów, których można użyć w trakcie jej budowy. Są nimi:

1. Typ 1: -/M/m/FIFO
 - System ze skończoną kolejką o długości m,
 - Obsługa zgłoszeń opisywalną za pomocą rozkładu wykładniczego.
2. Typ 2: -/G/1/PS
 - System z podziałem pracy procesora (harmonogramowanie round-robin),
 - Obsługa zgłoszeń z dowolnym rozkładem,
 - Procesory CPU, ALU's,
3. Typ 3: -/G/inf
 - System na zasadzie terminalu,
 - Obsługa zgłoszeń z dowolnym rozkładem,
4. Typ 4: -/G/1/LIFO PR
 - System oparty na przerwaniach,
 - Obsługa zgłoszeń z dowolnym rozkładem.

Zastosowanie metod przeznaczonych dla sieci BCMP (w tym metody przybliżonej SUM) wymagają spełnienia kilku założeń. Są to między innymi:

- Współczynnik μ jest stały i niezależny od obsługi,
- Współczynnik obsługi μ jest taki sam dla jednej stacji, niezależnie od klasy zgłoszenia,
- Zgłoszenia nie mogą zmieniać przynależności do klas w czasie przebywania w sieci,
- Liczba zgłoszeń każdej z klas w każdym systemie jest nieujemna:
 - $K_{ir} \geq 0$ dla $1 \leq r \leq R, 1 \leq i \leq N$
 - Gdzie R – ilość klas, N ilość systemów w sieci,
- Jeżeli istnieje droga od wejścia do i-tego systemu dla r-tej klasy to:
 - $K_{ir} > 0$
- Dla zamkniętej sieci ilość zgłoszeń w systemie jest stała.

2.1 System otwarty

Zakładamy, że każdy system jest traktowany niezależnie. Początkowo jest wyznaczana przepustowość w danym systemie, a następnie używane są wzory, jak w pojedynczym systemie.

2.1.1 Średni współczynnik wizyt zadania w systemie

Metoda korzysta ze współczynnika wizyt oznaczanego poprzez:

$$e_i = \frac{\lambda_i}{\lambda}$$

Gdzie λ – współczynnik przepływu przez cały system

Do jego wyznaczenia potrzebujemy zdefiniowanego prawdopodobieństw przejść pomiędzy systemami dla każdej z klas.

$$\lambda_{0i} = \lambda p_{0i}$$

$$e_{ir} = p_{0,ir} + \sum_{j=1}^N \sum_{s=1}^R e_{js} p_{js,ir} \quad \text{dla } i = 1, \dots, N; r = 1, \dots, R$$

Gdzie $p_{js,ir}$ – prawdopodobieństwo przejścia z stacji j-tej do stacji i-tej przy jednoczesnej zmianie przynależności do klasy z s do r.

Przy założeniu, że zgłoszenie nie może zmieniać przynależności do klas w trakcie przebywania w sieci można pominąć wewnętrzną sumę dla różnych klas zgłoszeń ($p_{js,ir} = 0$, gdy klasa s jest inna od klasy r).

Dodatkowo zapiszemy tutaj działania tylko dla jednej klasy. Jednak operacja ta zostaje powtórzona dla każdej z klas występujących w sieci.

$$e_i = p_{0i} + \sum_{j=1}^N e_j p_{ji}$$

Gdzie p_{ji} – prawdopodobieństwo przejścia z stacji j-tej do stacji i-tej.

Przechodząc do zapisu macierzowego otrzymujemy następujące równanie:

$$\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix} = \begin{bmatrix} p_{01} \\ p_{02} \\ \vdots \\ p_{0N} \end{bmatrix} + \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & & p_{2N} \\ \vdots & & \ddots & \\ p_{N1} & p_{N2} & & p_{NN} \end{bmatrix}^T \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}$$

$$E = P_0 + P^T E$$

$$E = (I - P^T)^{-1} P_0$$

Gdzie E to finalny wektor współczynników wizyt w danym systemie dla danej klasy.

2.1.2 Współczynnik przychodzenia zgłoszeń - metoda dokładna

Przemnażając tak wyliczoną wartość otrzymujemy wektor współczynników przychodzenia zgłoszeń do każdego z systemów występujących w sieci..

$$\lambda_i = E_i \lambda_{0i}$$

Gdzie i – rozpatrywana klasa.

2.1.3 Średnia ilość zgłoszeń w systemie:

Dla Typ 1,2,4, (m=1)

$$K_{ir} = \frac{\frac{\lambda_{ir}}{\mu_{ir}}}{1 - \rho_i}$$

Dla Typ 1, (m>1)

$$K_{ir} = m_i \rho_i + \frac{\rho_i}{1 - \rho_i} P_{mi}$$

$$P_{mi} = \frac{(m_i \rho_i)^{m_i}}{m_i! (1 - \rho_i)} \cdot \frac{1}{\sum_{k_i=0}^{m_i-1} \frac{(m_i \rho_i)^{k_i}}{k_i!} + \frac{(m_i \rho_i)^{m_i}}{m_i!} \cdot \frac{1}{1 - \rho_i}}$$

Dla Typ 3

$$K_{ir} = \frac{\lambda_{ir}}{\mu_{ir}}$$

2.2 System zamknięty

Stosujemy metodę przybliżoną SUM. Opiera się ona na założeniu, że dla każdego systemu w sieci średnia liczba zgłoszeń \bar{K} w systemie jest funkcją przepustowości tego systemu λ .

Metoda ta ma poniższe założenia, które są wymagane do spełnienia, przed jej zastosowaniem:

- System zamknięty,
- Stała wartość współczynnika obsługi dla jednej stacji.

2.2.1 Średni współczynnik wizyt zadania w systemie

Ogólnie używane jest poniższe równanie:

$$e_{ir} = \sum_{j=1}^N \sum_{s=1}^R e_{js} p_{js,ir} \quad \text{dla } i = 1, \dots, N; r = 1, \dots, R$$

Podobnie jak dla sieci otwartej poniższy wzór możemy uprościć. Następnie na podstawie poniższego równania otrzymujemy tylko (N-1) liniowo niezależnych równań.

$$e_i = \sum_{j=1}^N e_j p_{ji}$$

$$\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & & p_{2N} \\ \vdots & & \ddots & \\ p_{N1} & p_{N2} & & p_{NN} \end{bmatrix}^T \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}$$

W celu znalezienia rozwiązania musimy założyć jedną z wartości e_i np. $e_1 = 1$. Podstawiając do powyższego wzoru:

$$\begin{bmatrix} 1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & & p_{2N} \\ \vdots & & \ddots & \\ p_{N1} & p_{N2} & & p_{NN} \end{bmatrix}^T \begin{bmatrix} 1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}$$

$$\left(\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & & 1 \end{bmatrix} - \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & & p_{2N} \\ \vdots & & \ddots & \\ p_{N1} & p_{N2} & & p_{NN} \end{bmatrix}^T \right) \begin{bmatrix} 1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Rozbijając ten wzór na dwie sumy (pierwszy wiersz można pominąć, ponieważ nie wprowadza nowej informacji, natomiast pierwszą kolumnę rozdzielimy na dodatkowy czynnik:

$$e_1 \begin{bmatrix} p_{21} \\ \vdots \\ p_{N1} \end{bmatrix} + \begin{bmatrix} 1-p_{22} & & -p_{2N} \\ & \ddots & \\ -p_{N1} & & -p_{NN} \end{bmatrix}^T \begin{bmatrix} e_2 \\ \vdots \\ e_N \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

Finalnie otrzymujemy:

$$\begin{bmatrix} e_2 \\ \vdots \\ e_N \end{bmatrix} = - \left(\begin{bmatrix} 1-p_{22} & & -p_{2N} \\ & \ddots & \\ -p_{N1} & & -p_{NN} \end{bmatrix}^T \right)^{-1} \begin{bmatrix} p_{21} \\ \vdots \\ p_{N1} \end{bmatrix}$$

2.2.2 Współczynnik przychodzenia zgłoszeń - metoda SUM

Opierając się na metodzie iteracyjnej możemy obliczyć aproksymację tego współczynnika dla każdej z klas.

Algorytm składa się z poniższych kroków:

1. Inicjalizacja: $\lambda_1 = \lambda_2 = \dots = \lambda_R = 0.00001$, $\varepsilon = 0.00001$

2. Wyliczamy λ_r dla $r = 1, \dots, R$ według wzoru:

$$\lambda_r = \frac{K_r}{\sum_{i=1}^N fix_{ir}(\lambda_1, \dots, \lambda_R)}$$

przy czym:

$$fix_{ir} = \begin{cases} \frac{\frac{e_{ir}}{\mu_{ir}}}{1 - \frac{K-1}{K} \rho_i}, & Typ 1, 2, 4 (m_i = 1) \\ \frac{e_{ir}}{\mu_{ir}} + \frac{\frac{e_{ir}}{m_i \mu_{ir}}}{1 - \frac{K-m_i-1}{K-m_i} \rho_i} \cdot P_{m_i}(\rho_i), & Typ 1 (m_i > 1) \\ \frac{e_{ir}}{\mu_{ir}}, & Typ 3 \end{cases}$$

3. Wyliczamy błąd:

$$e = \sqrt{\sum_{r=1}^R (\lambda_{r_n} - \lambda_{r_{n+1}})^2}$$

Jeśli $e > \varepsilon$ wracamy do punktu 2.

Gdy warunek stopu zostanie osiągnięty i pętla powyższego algorytmu zostanie zakończona przechodzimy do dalszej części – wyliczania parametrów charakterystycznych sieci.

2.2.3 Średnia ilość zgłoszeń w systemie:

Wykorzystywana jest tutaj metoda przybliżona, która opiera się ona na założeniu, że średnia ilość zgłoszeń w systemie wchodzącym w skład sieci można przedstawić, jako funkcję zależną od współczynnika przychodzenia zgłoszeń.

$$f_{ir}(\lambda_{ir}) = \bar{K}_{ir} = \begin{cases} \frac{\rho_{ir}}{1 - \frac{K-1}{K}\rho_i}, & \text{Typ 1, 2, 4 } (m_i = 1) \\ m_i \rho_{ir} + \frac{\rho_{ir}}{1 - \frac{K-m_i-1}{K-m_i}\rho_i} \cdot P_{m_i}, & \text{Typ 1 } (m_i > 1) \\ \frac{\lambda_{ir}}{\mu_{ir}}, & \text{Typ 3} \end{cases}$$

gdzie:

$$\rho_i = \sum_{r=1}^R \rho_{ir}; \quad K = \sum_{r=1}^R K_r$$

oraz:

$$P_{mi} = \frac{(m_i \rho_i)^{m_i}}{m_i! (1 - \rho_i)} \cdot \frac{1}{\sum_{k_i=0}^{m_i-1} \frac{(m_i \rho_i)^{k_i}}{k_i!} + \frac{(m_i \rho_i)^{m_i}}{m_i!} \cdot \frac{1}{1 - \rho_i}}$$

Oraz

$$\lambda_{ir} = \lambda_r e_{ir}$$

Gdzie λ_i jest brana z metody iteracyjnej, a współczynnik wizyt został wyprowadzony w pierwszym podpunkcie.

2.3 Wspólne parametry charakterystyczne

Średnia ilość zgłoszeń w systemie jest wyliczana w sposób inny dla sieci otwartej oraz dla sieci zamkniętej. Wyprowadzone wzory zamieszczone zostały w poprzednich podrozdziałach.

W celu obliczenia pozostałych parametrów stosowane są już wspólne wzory, które zostały poniżej zamieszczone. Są one takie same dla obydwu typów sieci.

Przy założeniu, że współczynniki obsługi zgłoszeń są niezależne od ilości zgłoszeń, wtedy:

$$\rho_i = \frac{\lambda_i}{m_i \mu_i}$$

Średnia długość kolejki wyliczona na podstawie reguły Littl'a

$$\bar{Q}_{ir} = \lambda_{ir} \bar{W}_{ir}$$

Średni czas przebywania zgłoszenia klasy r w systemie i:

$$\bar{T}_{ir} = \frac{\bar{K}_{ir}}{\lambda_{ir}}$$

Średni czas oczekiwania zgłoszeń klasy r w systemie i, przy założeniu, że współczynniki obsługi są niezależne od liczby zgłoszeń:

$$\bar{W}_{ir} = \bar{T}_{ir} - \frac{1}{\mu_{ir}}$$

3 Opis implementacji systemu

3.1 Budowa rozwiązania:

Symulacja została oparta o środowisko Matlab. Zostało ono wybrane z powodu możliwości szybkiego podglądania wartości symulowanych oraz łatwość używania wszelkich operacji macierzowych bez konieczności importowania dodatkowych bibliotek. Wykorzystano czysty język skryptowy Matlab, który umożliwia definiowanie własnych klas, metod i funkcji. Ten element pozwala na oddzielenie od siebie fragmentów implementacji a tym samym rozdzielenie modułów.

Całość rozwiązania możemy podzielić na trzy elementy:

1. Klasa sieci,
2. Klasa optymalizatora,
3. Skrypt konfiguracji i skrypt symulacji.

3.1.1 Klasa sieci: `Network_nClass`:

Składa się z poniższych sekcji:

- Definicje pól (`%% Fields`)
 - N – ilość stacji,
 - R – ilość klas,
 - Stations_lambda – obliczona wartość λ dla każdej z klas i każdego z systemów,
 - Stations_visitRatio – wartość współczynnika odwiedzin,
 - Stations_types, Stations_mi, Stations_m – typ stacji, współczynnik obsługi zgłoszeń w danej stacji dla danej klasy, ilość stanowisk,
 - P – macierz incydencji,
 - P_in – wektor z jakim prawdopodobieństwem zgłoszenia z zewnątrz będą wchodziły do konkretnych stacji,
 - P_out – jak wyżej, tylko dotyczące wyjścia,
 - Lambda_in – wartości λ dla każdej z klas,
 - K_initial – ilość zgłoszeń danej klasy w systemie w sieci zamkniętej
- Konstruktor klasy (`%% Constructor`)
 - Inicjalizuje typ sieci, ilość klas i stacji.
- Zestaw funkcji wspólnych i pomocniczych (`%% Common functions`)
 - Obliczanie λ ,
 - Obliczanie współczynnika wizyt,
 - Zwracanie obiektu jednej stacji
- Funkcje obliczające podstawowe parametry (`%% Podstawowe parametry`)
 - Obliczanie dla każdej klasy i stacji parametrów:
 - P, K, Q, m0, rho, lambda, W, T.
- Funkcje obsługujące sieć zamkniętą (`%% Closed network`)
 - Obsługa algorytmu fix,
 - Obliczanie przybliżonej ilości λ za pomocą metody iteracyjnej,
 - Obliczanie średniej ilości zgłoszeń w systemie.
- Funkcje obsługujące sieć otwartą (`%% Open network`)
 - Obliczanie przybliżonej ilości λ za pomocą metody iteracyjnej,
 - Obliczanie średniej ilości zgłoszeń w systemie.

3.1.2 Skrypt konfiguracji

Elementy konfigurowalne przez użytkownika są możliwe do ustawienia z poziomu pliku konfiguracyjnego. W skrypcie zawarte są sekcje:

- Konfiguracja – opis sieci, definiowanie typu sieci i ilość klas/systemów

- Opis Stacji – definiowanie typów i parametrów poszczególnych stacji,
- Opis przejść pomiędzy stacjami – definiowanie macierzy incydencji, macierzy wyjścia/wejścia, ilości zgłoszeń, lambdy wejściowej.

3.1.3 Skrypt symulacji

Zwiera sekcje:

- Konfiguracja – uruchomienie skryptu konfiguracji,
- Oblicz parametry – obliczenie lambd w systemie,
- Pokaz wyniki – zaprezentowanie wyników i charakterystycznych parametrów.

3.2 Instrukcja obsługi:

1. Zdefiniowanie skryptu konfiguracji
2. Wypisanie w skrypcie symulacji, które parametry chcemy obliczyć
3. Uruchomienie skryptu symulacji
4. Odczytanie z konsoli wyników.

Kroki opcjonalne:

5. Wykonanie kroku optymalizacji na wcześniej zdefiniowanej sieci,
6. Wykonanie części skryptu wyświetlające wyniki,
7. Odczytanie z konsoli wyników.

Przykładowy zestaw plików wymaganych do uruchomienia to:

- /03. Siec BCMP/SymulacjaSieciZamknietej_przyklad.m – główny plik „**skrypt symulacji**”,
- /03. Siec BCMP/confSiecZamknietej_przyklad.m – „**skrypt konfiguracji**”.

3.3 Przykładowy zapis konfiguracji sieci:

```
%% Konfiguracja - Opis sieci:
%(Ilosc stacji, Ilosc klas, Typ sieci (open/close))
siec = Network_nClass(6, 3, 'close');

%% Opis Stacji:
%Typy stacji (1/2/3/4)
siec.stations_types = [1; 1; 1; 1; 1; 3];
%Ilosc kanalow obslugi
siec.stations_m = [10; 38; 40; 40; 20; 1];
%Wspolczynnik obslugi zgloszen w stacjach dla kazdej z klas
siec.stations_Mi{1} = [20; 5; 8; 10; 5; 100];
siec.stations_Mi{2} = [20; 5; 8; 10; 5; 100];
siec.stations_Mi{3} = [20; 5; 8; 10; 5; 100];

%% Opis przejasc pomiedzy stacjami:
%Macierz prawdopodobienstw przejasc z jednej stacji do drugiej
%Przejscie z Pij -> prawdopodobienstwo przejścia z i-tej stacji do j-tej.
siec.P{1} = [0, 1, 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, 1;
            1, 0, 0, 0, 0, 0];

siec.P{2} = [0, 0, 1, 0, 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, 1;
        1, 0, 0, 0, 0, 0];

siec.P{3} = [0, 0, 0, 1, 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, 1;
             1, 0, 0, 0, 0, 0];

%% Ilosc zgloszen danej klasy w systemie
siec.K_initial{1} = 20;
siec.K_initial{2} = 10;
siec.K_initial{3} = 20;

```

4 Przykład - sformułowanie problemu BCMP

4.1 Opis słowny:

W zakładzie produkowane są różnego rodzaju części, między innymi ramy samochodowe, zbiorniki paliwa, zderzaki. Częstotliwość i czas produkcji uzależniony jest od rodzaju produkowanych części oraz wielkości zlecenia.

W skład modelu linii produkcyjnej części samochodowych (rys.) wchodzi następujące systemy:

S_1 - prasy tłocznie nadające kształt blachą (10 maszyn, $\mu_1 = 20$ szt./min),

S_2 - spawarki do ram (38 maszyn, $\mu_2 = 5$ szt./min),

S_3 - spawarki do zbiorników paliwa (40 maszyn, $\mu_3 = 8$ szt./min),

S_4 - zgrzewarki do zderzaków (40 maszyn $\mu_4 = 10$ szt./min),

S_5 - szlifierki (20 maszyn, $\mu_5 = 5$ szt./min),

S_6 - magazyn (może przechować 50 tys. sztuk, system $M|M|inf$).

Głównym ograniczeniem jest liczba szlifierek dlatego przyjęto, że są systemami bez priorytetów.

Wprowadzono do systemu następujące klasy produktów :

klasa 1 - 20 ram samochodowych,

klasa 2 - 10 zbiorników paliwa,

klasa 3 - 20 zderzaków.

4.2 Założenia i ograniczenia:

- System zamknięty (otwarty),
- Bez priorytetów,
- Zgłoszenia nie mogą zmieniać klasy w trakcie przebywania w systemie,
- Sieć składa się z dwóch typów systemów:
 - $M/M/\infty$
 - $M/M/n/FIFO/\infty$
- Jedno źródło wejściowe zgłoszeń – generująca różne klasy z stałymi współczynnikami

4.3 Podział na systemy kolejkowe:

4.3.1 Stacje:

Numer	Nazwa stacji	Opis	Rodzaj systemu
1	Prasy/ tłocznie	Kształtowanie produktu.	$M/M/10/FIFO/\infty$
2	Spawarki do ram	Łączenie produktów w całość.	$M/M/38/FIFO/\infty$
3	Spawarki do zbiorników	Łączenie produktów w całość.	$M/M/40/FIFO/\infty$
4	Zgrzewarki	Łączenie produktów w całość.	$M/M/40/FIFO/\infty$
5	Szlifierki	Końcowe wygładzanie produktów.	$M/M/20/FIFO/\infty$
6	Magazyn	Przechowalnia produktów i surowców.	$M/M/\infty$

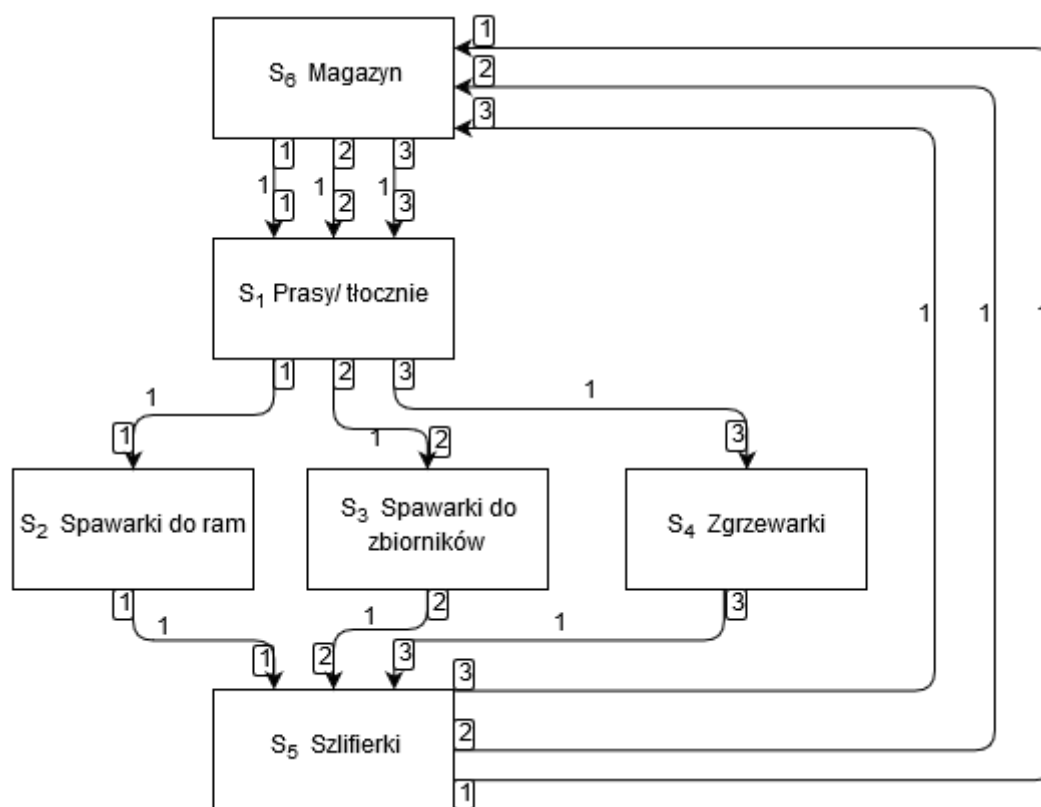
4.3.2 Współczynniki obsługi poszczególnych klas w stacjach:

Numer	Nazwa stacji	Współczynniki obsługi [jednostka/minutę]
1	Prasy/ tłocznie	$\mu_1 = \mu_{11} = \mu_{12} = \mu_{13} = \mu_{14} = \mu_{15} = \mu_{16} = 20 \text{ szt./min}$
2	Spawarki do ram	$\mu_2 = \mu_{21} = \mu_{22} = \mu_{23} = \mu_{24} = \mu_{25} = \mu_{26} = 5 \text{ szt./min}$
3	Spawarki do zbiorników	$\mu_3 = \mu_{31} = \mu_{32} = \mu_{33} = \mu_{34} = \mu_{35} = \mu_{36} = 8 \text{ szt./min}$
4	Zgrzewarki	$\mu_4 = \mu_{41} = \mu_{42} = \mu_{43} = \mu_{44} = \mu_{45} = \mu_{46} = 10 \text{ szt./min}$
5	Szlifierki	$\mu_5 = \mu_{51} = \mu_{52} = \mu_{53} = \mu_{54} = \mu_{55} = \mu_{56} = 5 \text{ szt./min}$
6	Magazyn	$\mu_0 = \mu_{01} = \mu_{02} = \mu_{03} = \mu_{04} = \mu_{05} = \mu_{06} = 100 \text{ szt./min}$

4.4 Podział na klasy wewnątrz systemu i współczynniki przychodzenia zgłoszeń:

Klasa	Nazwa klasy	Opis	Współczynnik pojawiania się zgłoszeń
1	Ramy samochodowe		$\lambda_1 = 20 \text{ szt./min}$
2	Zbiornik paliwa		$\lambda_2 = 10 \text{ szt./min}$
3	Zderzaki		$\lambda_3 = 20 \text{ szt./min}$

4.5 Diagram:



Rysunek 1 Model sieci zamkniętej BCMP przedstawiającej proces produkcyjny części samochodowych

4.6 Macierze przejścia do kolejnych stacji

	1 – Ramy samochodowe					
System:	1	2	3	4	5	6
1	-	1	-	-	-	-
2	-	-	-	-	1	-
3	-	-	-	-	-	-
4	-	-	-	-	-	-
5	-	-	-	-	-	1
6	1	-	-	-	-	-

	1 – Zbiorniki paliwa					
System:	1	2	3	4	5	6
1	-	-	1	-	-	-
2	-	-	-	-	-	-
3	-	-	-	-	1	-
4	-	-	-	-	-	-
5	-	-	-	-	-	1
6	1	-	-	-	-	-

	1 – Zderzaki					
System:	1	2	3	4	5	6
1	-	-	-	1	-	-
2	-	-	-	-	-	-
3	-	-	-	-	-	-
4	-	-	-	-	1	-
5	-	-	-	-	-	1
6	1	-	-	-	-	-

5 Problem optymalizacji parametrów sieci

W oparciu o algorytm karalucha możliwość optymalizacji wartości kilku parametrów charakterystycznych dla wybranych stacji na podstawie określonej funkcji celu. Funkcja oceny rozwiązania bazuje na wartościach opisujących sieć BCMP w stanie ustalonym – np. średni czas przebywania w systemie, średni czas oczekiwania w kolejce, średnia ilość wolnych kanałów obsługi itd.

Poszukiwana jest liczba kanałów obsługi w poszczególnych systemach, dla których funkcja oceny sieci jest najlepsza. Pozostałe parametry sieci pozostają niezmiennie.

Funkcja oceny rozwiązania bazuje na długości kolejki oraz średnim użyciu kanałów w każdym z systemów. Możliwe jest zdefiniowanie własnej funkcji, która na podstawie parametrów charakterystycznych sieci oblicza jej ocenę.

Problem ten jest nietrywialny, ponieważ obiekt badań jest opisywany nieanalitycznymi funkcjami. W celu znalezienia rozwiązania posłużono się algorytmem bazującym na procesach biologicznych – algorytm karalucha.

5.1 Opis algorytmu karalucha:

Algorytm CSO (Cockroach Swarm Optimization) jest algorytmem stadnym, który bazuje na zachowaniu roju karaluchów. Opiera się on o trzy podstawowe zachowania grupy:

- Pogoń za stadem
- Rozproszenie
- Zachowanie „ruthless” (ang. bezwzględne)

Rój karaluchów modelowany jest przez przestrzeń R_D gdzie D - rozmiar problemu, zatem i -ty karaluch to wektor $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$. W roju wyróżniamy dodatkowo karalucha P_g będącego globalnym optimum. oraz karaluchy P_i , które są optimumami lokalnymi - najlepsze rozwiązania w zasięgu wzroku innych karaluchów - $P_i = \text{Opt}_j\{X_j: ||X_i - X_j|| < \text{zasięg wzroku}\}$.

Pogoń za stadem dana jest następującym wzorem:

$$X'_i = \begin{cases} X_i + \text{krok} * \text{rand} * (P_i - X_i) & \text{dla } X_i \neq P_i \\ X_i + \text{krok} * \text{rand} * (P_g - X_i) & \text{w p.p.} \end{cases}$$

Rozproszenie może być stosowane w wybranych iteracjach algorytmu. Dane jest wzorem:

$$X'_i = X_i + \text{rand}$$

Zachowanie bezwzględne polega na zastąpieniu losowego karalucha przez P_g . Dane jest wzorem:

$$X(k) = P_g$$

Podsumowując, algorytm składa się z następujących kroków:

- Inicjalizacja populacji
- Wyznaczenie P_i oraz P_g
- Pogoń za stadem i odświeżenie P_g
- Rozproszenie i odświeżenie P_g

- Zachowanie bezwzględne
- Sprawdzenie kryterium stopu i ewentualny powrót do pkt. 2

5.2 Funkcja oceny rozwiązania ogólnie:

5.2.1 Funkcja:

$$f = \sum_{i=1}^N \left[\sum_{j=1}^R C_{1ij} \bar{Q}_{ij} + C_{2i} m_{nzi} \right]$$

Gdzie:

- Q_{ij} – średnia ilość zleceń j-tej klasy czekających w kolejce w i-tym systemie
- m_{nzi} – średnia ilość nie zajętych kanałów
- C_{1ij} – współczynniki, z jakimi zostaną uwzględnione wartości Q_{ij} . Koszt oczekiwania zgłoszenia,
- C_{2i} – współczynniki, z jakimi zostaną uwzględnione wartości m_{nzi} . Koszt obsługi zgłoszenia w stacji,

5.3 Wyniki optymalizacji:

Ilość zgłoszeń danej klasy w systemie:

- Klasa 1 - 20szt.
- Klasa 2 - 10 szt.
- Klasa3 - 20 szt.

Liczba stanowisk						
Nr stacji	S1	S2	S3	S4	S5	S6
Początkowo	10	38	40	40	20	1
Po optymalizacji	59	15	17	27	83	1

Lambda per stacja						
Nr stacji	S1	S2	S3	S4	S5	S6
Klasa 1						
Początkowo	56,64	56,64	0	0	56,64	56,64
Po optymalizacji	43,37	43,37	0	0	43,37	43,37
Klasa 2						
Początkowo	35,96	0	35,96	0	35,96	35,96
Po optymalizacji	25,97	0	25,97	0	25,97	25,97
Klasa 3						
Początkowo	79,01	0	0	79,01	79,01	79,01
Po optymalizacji	55,56	0	0	55,56	55,56	55,56

Rho per stacja						
Nr stacji	S1	S2	S3	S4	S5	S6
Klasa 1						
Początkowo	0,28	0,30	0	0	0,57	0,57
Po optymalizacji	0,04	0,58	0	0	0,1	0,02
Klasa 2						
Początkowo	0,18	0	0,11	0	0,36	0,36

Po optymalizacji	0,02	0	0,19	0	0,06	0,01
Klasa 3						
Początkowo	0,4	0	0	0,2	0,79	0,79
Po optymalizacji	0,05	0	0	0,21	0,13	0,02

K per stacja						
Nr stacji	S1	S2	S3	S4	S5	S6
Klasa 1						
Początkowo	3,79	11,33	0	0	4,3	0,57
Po optymalizacji	2,17	8,72	0	0	8,67	0,43
Klasa 2						
Początkowo	2,4	0	4,49	0	2,74	0,36
Po optymalizacji	1,3	0	3,25	0	5,19	0,26
Klasa 3						
Początkowo	5,28	0	0	7,9	6,02	0,79
Po optymalizacji	2,78	0	0	5,56	11,11	0,56

Q per stacja						
Nr stacji	S1	S2	S3	S4	S5	S6
Klasa 1						
Początkowo	0,95	0	0	0	0	0
Po optymalizacji	0	0,05	0	0	0	0
Klasa 2						
Początkowo	0,61	0	0	0	0	0
Po optymalizacji	0	0	0	0	0	0
Klasa 3						
Początkowo	1,33	0	0	0	0	0
Po optymalizacji	0	0	0	0	0	0

m0 per stacja						
Nr stacji	S1	S2	S3	S4	S5	S6
Klasa 1						
Początkowo	2,83	11,33	0	0	4,31	0,57
Po optymalizacji	2,17	8,67	0	0	8,67	0,43
Klasa 2						
Początkowo	1,8	0	4,49	0	2,74	0,36
Po optymalizacji	1,3	0	3,25	0	5,19	0,26
Klasa 3						
Początkowo	3,95	0	0	7,9	6,02	0,79
Po optymalizacji	2,78	0	0	5,56	11,11	0,56

Porównując poszczególne parametry przed i po optymalizacji możemy zauważyć poprawność działania algorytmu karalucha. Zgodnie z założeniem tworzonej sieci, największe kolejki pojawiają się na stanowisku 5. Dla klasy 3 parametr ρ osiąga 0.79 (im większe ρ tym dłuższa generuje się kolejka). Natomiast po wykonaniu optymalizacji liczba stanowisk(kanałów) na stacji 5 znacznie się zwiększyła z 20 do 83. Pozwoliło to jednak dwukrotnie zwiększyć ilość klientów.

Jednak ilość kanałów w niektórych stacjach została zredukowana. Oznacza to, że dla wprowadzonych danych wejściowych część kanałów jest nieobsługiwana. Algorytm karalucha pozwala zarówno do zwiększanie ilości dostępnych kanałów jak i ich redukcję w celu obniżenia kosztów użytku.

Początkowo suma wszystkich kanałów obsługi z wyłączeniem stacji 6 (system $M/M/\infty$) wynosiła 148, natomiast po optymalizacji aż 201. Jest to znaczna różnica jednak suma długości kolejek w systemie spadła do bliska zera.

Opracowany algorytm przyniósł oczekiwane rezultaty. Pozwolił na zniwelowanie jakichkolwiek kolejek w całym systemie oraz wyrównał szybkość obsługi na poszczególnych stacjach. Jedną z wad tego algorytmu jest brak możliwości prostego wprowadzania ograniczeń dotyczących ilości stanowisk na danej stacji, co po optymalizacji przyczynia się do dużych różnic liczby kanałów na poszczególnych systemach. Uwzględniając tę wadę rzeczywistego wykorzystania tej implementacji powinniśmy dodać ograniczenia regulujące maksymalną możliwość zmian wykonanych w poszczególnych systemach.

6 Podsumowanie i wnioski

Napisana symulacja dostarczyła nam do dyspozycji narzędzie, dzięki któremu można przewidywać wydajność danej kluczowych obszarów firmy, uczelni, czy dowolnego miejsca gdzie możemy wyróżnić przepływ pewnego typu „zgłoszeń” danej klasy, oraz systemy/stanowiska, przez które muszą przebyć.

Założenia sieci BCMP nakładają wiele ograniczeń, które zmuszają nas do uproszczenia rzeczywistego modelu w celu symulacji. Widzimy jednak, że można wiele z procesów przedstawić w tej formie bez straty dla ogólności i poprawności.

Tego typu oprogramowanie uważamy, że może stanowić dobry punkt wyjścia do dalszej optymalizacji i przyspieszania produkcji. Poprawne zdefiniowanie obliczanie funkcji celu pozwala, przy wykorzystaniu optymalizacji wzorowanej na zjawiskach przyrodniczych, na wstępne wyznaczenie parametrów potencjalnie dających lepsze wyniki.

W projekcie widać kilka miejsc, które wymagałyby jeszcze poprawy, jeżeli miałby on być wykorzystywany w większych sieciach. Są to między innymi ustalanie przedziałów zmiennych (ilości kanałów obsługi) w każdym z systemów w obrębie których będziemy szukać rozwiązania sub-optimalnego, dodatkowo należało by wzbogacić obszar prezentacji wyników o przystępniejszą reprezentację w formie wykresów i porównań parametrów sieci przed i po optymalizacji.

Samo prowadzenie projektu też pokazało znaczenie podziału na zadania, obszary i osoby odpowiedzialne. Stanowi to dobre doświadczenie, które zdecydowanie przyda się w dalszym życiu.