

Bartłomiej Gudowski*, Jarosław Wąs*

Wybrane algorytmy szeregowania na przykładzie obsługi pętli tramwajowej

1. Wstęp

W ostatnich latach rozpowszechniły się technologie związane z monitorowaniem i zarządzaniem flotą pojazdów oparte m.in. na technologii GPS, np. [3, 5]. W szczególności rozwiązania te znajdują coraz szersze zastosowania w miejskich przedsiębiorstwach komunikacyjnych do monitorowania i zarządzania pojazdami.

Dzięki zastosowaniu wspomnianych wyżej technologii możliwe staje się, przykładowo optymalne rozmieszczenie wjeżdżających na pętlę tramwajów, przez miejskiego dyspozytora, w sytuacji zaistniałych zakłóceń ruchu. Uwzględniając pewną liczbę dostępnych na pętli tramwajowej peronów (torów), a jednocześnie zazwyczaj większą liczbę linii tramwajowych kończących bieg na pętli od liczby dostępnych peronów, otrzymujemy ciekawe zadanie do rozwiązania (tzn. rozmieszczenie przybywających tramwajów różnych linii na peronach, pod kątem punktualnych odjazdów). Oczywiście w przypadku „statycznym”, tzn. deterministycznych, zgodnych z rozkładem jazdy przyjazdów pociągów tramwajowych, zadanie to jest niezbyt skomplikowane. Znacznie trudniejszym przypadkiem jest rozpatrzenie „dynamicznego” przypadku, czyli optymalne rozmieszczenie tramwajów przybywających niezgodnie z rozkładem jazdy.

W artykule zaproponowano i omówiono przykładowe algorytmy rozmieszczenia tramwajów przybywających na pętlę tramwajową. Opisywany problem nawiązuje zarówno do zagadnień szeregowania zadań [2], jak i systemów kolejkowych [4]. Omawiane w pracy przypadki wykazują, przez analogię, pewne podobieństwa do następujących systemów kolejkowych: $D/D/./\infty/FIFO$ oraz $M/D/./\infty/FIFO$. W pewnych specyficznych warunkach możemy również mówić o modelu $M/M/./\infty/FIFO$, w sytuacji gdy opóźnienie tramwaju jest większe niż bufor czasu przewidziany na jego oczekiwanie na pętli.

2. Wybrane algorytmy szeregowania

W rozdziale tym zaproponowano kilka algorytmów szeregowania pociągów tramwajowych przybywających na pętlę, w sytuacji kiedy ich przybycie nie jest deterministyczne. To znaczy możliwe są opóźnienia i tzw. nadspieszenia w stosunku do rozkładu jazdy.

* Katedra Automatyki, Akademia Górniczo-Hutnicza w Krakowie

W tabeli 1 zgrupowano wszystkie oznaczenia związane z algorytmami szeregowania pociągów tramwajowych.

Tabela 1
Spis oznaczeń stosowanych w artykule

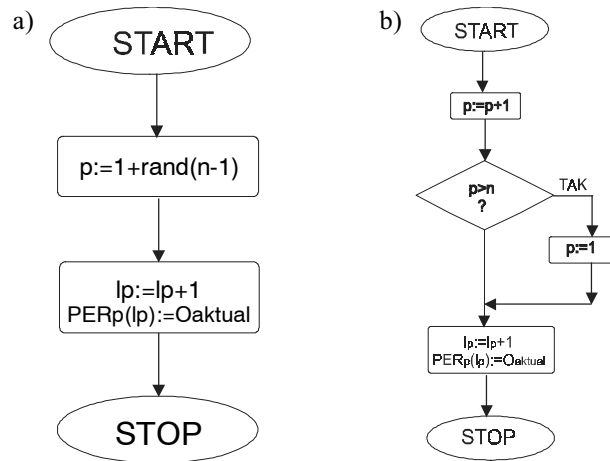
Oznaczenie	Opis
P_i	czas przyjazdu i -tego pociągu, $i = 1, \dots, m$
m	liczba przyjazdów/odjazdów pociągów na pętlę w ciągu dnia
O_i	czas odjazdu i -tego pociągu, $i = 1, \dots, m$
n	liczba peronów
PER_k	ciąg reprezentujący k -ty peron, zaś O_i jego elementy dla $i = 1, \dots, m$
l_k	liczba pociągów oczekujących na k -tym peronie
p	nr peronu wybranego dla aktualnie rozpatrywanego pociągu
$T1$	zbiór indeksów peronów, z których aktualny pociąg odjedzie bez opóźnień
$T2$	zbiór indeksów peronów, z których aktualny pociąg odjedzie z opóźnieniem
O_{akt}	zaplanowany odjazd aktualnego pociągu
$\text{rand}(x)$	funkcja generująca liczbę losową, całkowitą z przedziału $[0, x]$

2.1. Najprostsze algorytmy: losowy i równomiernego przydziału

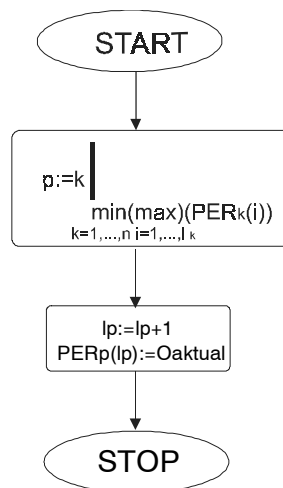
Jednym z najprostszych przypadków jest losowe przydzielanie przybywających tramwajów pomiędzy perony. Algorytm ten przedstawiono na rysunku 1a. Kolejnym prostym przypadkiem jest kolejne przydzielanie tramwajów pomiędzy perony, co przedstawiono na rysunku 1b.

2.2. Algorytm najwcześniejszego odjazdu

Algorytm najwcześniejszego odjazdu przedstawiony jest na rysunku 2. Jego zasada działania jest następująca: program sprawdza zaplanowane czasy odjazdów pociągów tramwajowych oczekujących na odjazd na peronach i dla każdego peronu wyznacza termin najpóźniejszego zaplanowanego odjazdu. Dla peronu, na którym nie oczekuje żaden skład, ten czas wynosi 0. Następnie wybierany jest ten peron, dla którego zaplanowany najpóźniejszy odjazd jest najwcześniejszy. Oczywiście oznacza to preferencję do wybierania pustych torów pętli (kolejność pustych peronów jest nieistotna).



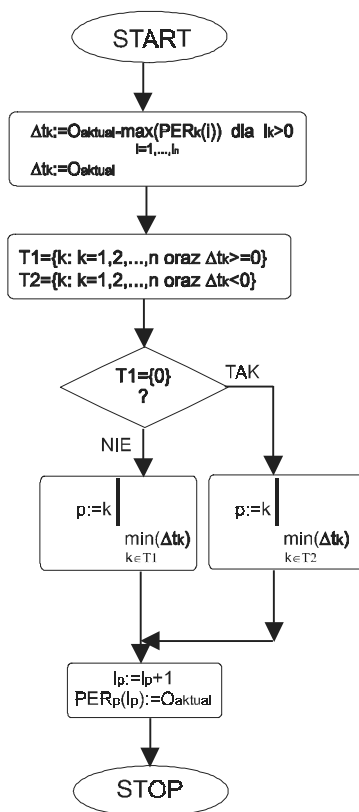
Rys. 1. Algorytmy: a) losowy; b) kolejnego przydzielania



Rys. 2. Algorytm najwcześniejszego odjazdu

2.3. Algorytm quasi-optymalny

Algorytm quasi-optymalny przedstawiony jest na rysunku 3. Podobnie jak w algorytmie najwcześniejszego odjazdu ustalane są najpóźniejsze zaplanowane czasy odjazdu dla każdego z peronów. Następnie program sprawdza, czy różnica między zaplanowanym odjazdem pociągu, który właśnie oczekuje na wjazd na pętlę, a ostatnim odjazdem z danego peronu jest nieujemna – oznacza to, że jeśli skład tramwajowy zostanie wpuszczony na taki peron, będzie mógł odjechać o czasie. Spośród wszystkich peronów, które spełniają powyższy warunek, wybierany jest ten, dla którego wspomniana różnica czasów jest najmniejsza.



Rys. 3. Algorytm quasi-optymalny

W praktyce oznacza to, że przyjeżdżające pociągi wjeżdżają na puste tory tylko w ostateczności. Może się jednak zdarzyć, że wszystkie perony są zajęte, zaś nowo przybyły skład przyjechał mocno opóźniony i na który peron by nie wjechał i tak odjedzie opóźniony w stosunku do rozkładu jazdy. W takim przypadku różnica między planowanym czasem odjazdu nowo przybyłego pociągu a czasami ostatnich odjazdów z poszczególnych peronów będzie ujemna. Jeśli zajdzie taka okoliczność, wybierany jest peron, dla którego wartość bezwzględna omawianej różnicy będzie najmniejsza, co zagwarantuje nam najmniejsze opóźnienie pociągu.

3. Implementacja

Zaproponowane algorytmy zaimplementowano w języku C i przetestowano za pomocą metody symulacji. Program symulacyjny w całości został napisany w ANSI C z wykorzystaniem klasycznego podejścia zdarzeniowego opisanego wyczerpująco w pracy [6]. Każdy przyjazd i odjazd pociągu tramwajowego reprezentowany jest przez pojedyncze zdarzenie. Zdarzenia przechowywane są na chronologicznie uporządkowanej liście zdarzeń

zarządzanej przez moduł tzw. **koordynatora** (I poziom trójpoziomowej struktury modelu symulacyjnego). W każdym kroku czasowym koordynator przesuwa zegar symulacji do czasu wystąpienia pierwszego zdarzenia na liście. Następnie realizowane są wszystkie zdarzenia zaszeregowane na daną chwilę czasową. Obsługa zdarzenia przyjazdu powoduje zaszeregowanie zdarzenia odjazdu (obsługa zdarzeń tworzy II poziom modelu). Dane wejściowe oraz wyniki symulacji zapisywane są w zwykłych plikach tekstowych.

4. Wyniki symulacji

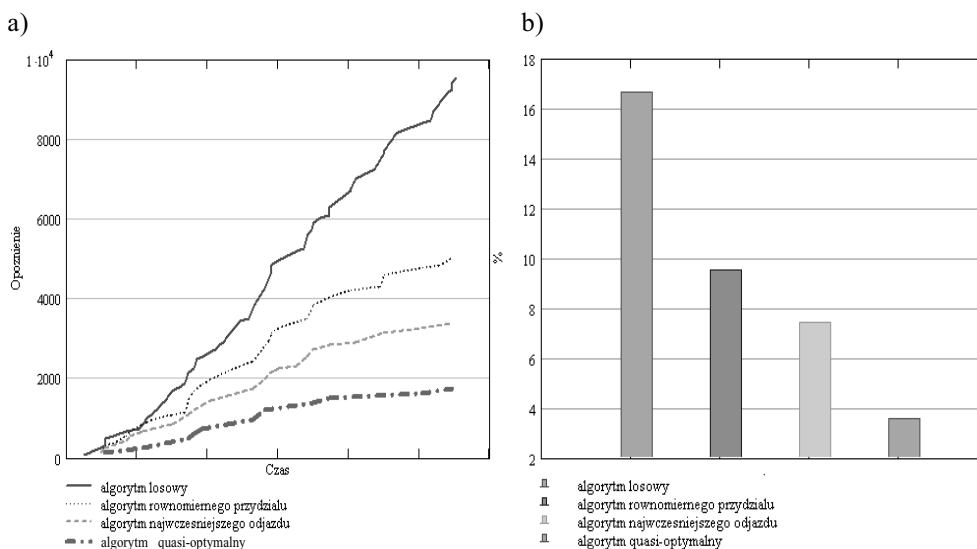
Rozpatrzmy najpierw wyniki symulacji przeprowadzonej dla pętli tramwajowej z dwoma peronami, która obsługuje cztery linie tramwajowe.

W tabeli 2 przedstawiono zestawienie danych wejściowych dla pierwszego przypadku.

Tabela 2

Linia	Częstotliwość	Pierwszy odjazd	Ostatni odjazd
9	10 min	6:04	21:54
14	10 min	5:01	23:01
16	10 min	6:03	21:13
20	20 min	7:17	19:57

Wykresy na rysunku 4 przedstawiają wyniki symulacji dla pierwszego przykładowego zestawu danych wejściowych.



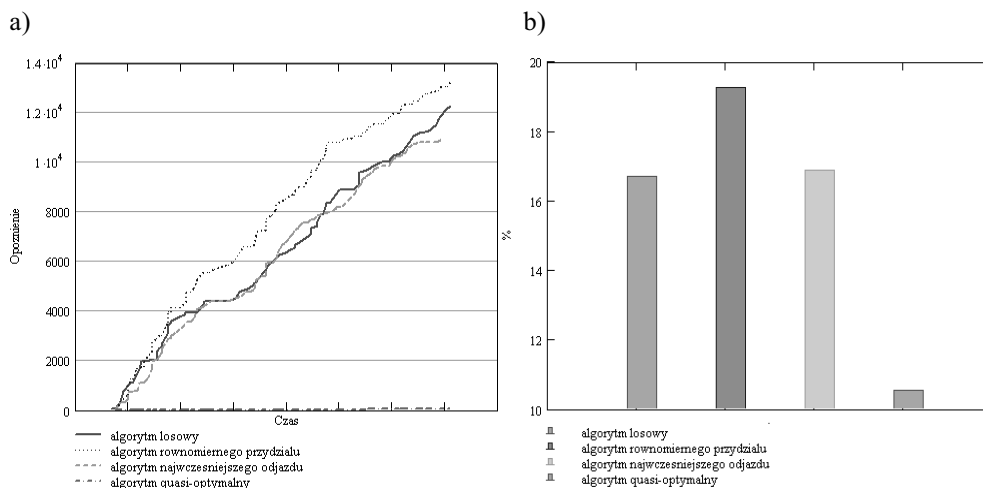
Rys. 4. Sumaryczne czasy opóźnień tramwajów, przy zastosowanych różnych algorytmach szeregowania (a); procentowa liczba opóźnionych pociągów (w %) w ciągu dnia (b)

Kolejną rozpatrywaną sytuacją to obsługa pętli tramwajowej z trzema peronami, która obsługuje pięć linii tramwajowych. W tabeli 3 przedstawiono zestawienie danych wejściowych dla tego przypadku.

Na rysunku 5 przedstawiono wyniki symulacji dla kolejnego przykładowego zestawu danych wejściowych.

Tabela 3

Linia	Częstotliwość	Pierwszy odjazd	Ostatni odjazd
1	7 min	4:30	22:42
2	15 min	5:03	21:48
3	10 min	4:38	22:58
4	10 min	4:11	23:21
5	20 min	5:09	21:49



Rys. 5. Sumaryczne czasy opóźnień tramwajów, przy zastosowanych różnych algorytmach szeregowania (a); procentowa liczba opóźnionych pociągów w ciągu dnia (b)

5. Podsumowanie

W artykule zaproponowano i porównano cztery algorytmy szeregowania pociągów tramwajowych na pętli. Z analizy uzyskiwanych rozwiązań wynika, że zastosowanie najprostszych algorytmów, tzn. losowego i równomiernego przydziału, daje prawie zawsze rozwiązania słabej jakości. Z kolei algorytm najwcześniejszego odjazdu dawał zazwyczaj lepsze rezultaty niż dwa poprzednie. Natomiast zdecydowanie najlepsze wyniki uzyskane

zostały przy zastosowaniu algorytmu quasi-optimalnego. Gwarancję uzyskania rozwiązania optymalnego można uzyskać przez wykonanie przeglądu pełnego wszystkich możliwych rozwiązań. Należy zwrócić jednak uwagę, że uzyskanie takiego rozwiązania, poprzez konieczność rozważenia wszystkich możliwych kombinacji rozwiązań, nawet po usunięciu rozwiązań niedopuszczalnych, będzie bardzo czasochłonne ze względu na bardzo liczny zbiór rozwiązań dopuszczalnych.

W artykule rozważono kilka możliwych algorytmów, których atutem jest prostota i niski nakład obliczeniowy, pozwalający realizować je wielokrotnie w ciągu dnia i dający wyniki w bardzo krótkim czasie, co umożliwiłoby zastosowanie go np. w półautomatycznym sterowniku zwrotnicy. Należy jednak zwrócić uwagę, że przy opracowywaniu algorytmów przyjęto pewne uproszczenia. W praktyce na pętlach komunikacji tramwajowej oprócz „regularnych” peronów, stosuje się tzw. żeberka postojowe, które doraźnie powiększają liczbę dostępnych peronów. Poza tym w sytuacjach opóźnień czy awarii włączane są składy rezerwowe. Kompletnie rozwiązanie zagadnienia sterowania ruchem tramwajowym w dużym mieście jest niezwykle złożonym problemem i wymaga rozważenia dużej liczby możliwych scenariuszy.

Literatura

- [1] Janiak A.: *Wybrane problemy i algorytmy szeregowania zadań i rozdzielania zasobów*. Warszawa, Akademiczna Oficyna Wydawnicza PLJ 1999
- [2] Dudek-Dyduch Ewa: *Formalizacja i analiza dyskretnych systemów produkcyjnych*. Zeszyty Naukowe AGH, Kraków 1991
- [3] <http://elte.f.pl/?menu=technologia&id=OFMZAI>, 2004
- [4] Filipowicz B.: *Modelowanie i optymalizacja sieci kolejkowych. Część I: Systemy markowskie*. Kraków, Wydawnictwa AGH 1997
- [5] <http://www.xtrack.pl/>, 2005
- [6] Pidd M.: *Computer Simulation in Management Science*. Wiley 1994