

Adam Piórkowski*

Heurystyczne algorytmy szeregowania wiadomości z terminami dostarczeń

1. Wstęp

Komunikacja asynchroniczna staje się najpopularniejszym rozwiązaniem wymiany danych w systemach rozproszonych. Jest to związane z jej licznymi zaletami, przede wszystkim nieblokowaniem nadawcy podczas wysyłania wiadomości oraz z możliwością pełnego wykorzystania przydzielonego pasma transmisji, łączącego komunikujące się węzły. Realizację zadań obsługi transmisji sieciowej powierza się najczęściej systemom wiadomości kolejkowanych *MQ* (*message queuing*), stanowiących oprogramowanie warstwy pośredniej (*middleware*). W systemie takim przesyłane dane są formowane w wiadomości o określonej długości i priorytecie. Nadmiar niewysłanych wiadomości jest kolejkowany. Od systemu wiadomości kolejkowanych wymaga się odpowiedniego wyznaczenia kolejności ich wysyłania, co może stanowić obszar optymalizacji.

W niniejszym artykule rozważana jest optymalizacja kosztu całkowitego systemu przy uwzględnieniu zagadnienia transportu sieciowego, jakim jest podział strumienia danych na pakiety oraz przy uwzględnieniu ograniczeń czasu rzeczywistego dla przesyłanych wiadomości.

2. Model systemu i założenia

Do rozważań analitycznych systemu konieczne jest wprowadzenie formalnego modelu komunikacji asynchronicznej.

Wiadomość jest pewnym spójnym ciągiem danych, przesyłanym pomiędzy nadawcą a odbiorcą, cechowany przez parę: długość i wagę (priorytet).

Formalny zapis wiadomości

$$m = [l, w],$$

* Zakład Informatyki w Naukach o Ziemi, WGGiOŚ, Akademia Górniczo-Hutnicza w Krakowie; pioro@agh.edu.pl

gdzie:

- m – wiadomość (*message*),
- l – długość wiadomości (*length*),
- w – waga (priorytet) wiadomości (*weight*).

Długość wiadomości jest liczbą elementarnych jednostek danych (w zdecydowanej większości przypadków mierzona w bajtach [B]). Waga zaś jest liczbą naturalną z pewnego ustalonego przedziału i może być rozpatrywana w dwóch kategoriach jako:

- 1) waga – wartość danej wiadomości, np. odwzorowanie konkretnego odniesienia finansowego na przedział priorytetów;
- 2) priorytet – ważność wiadomości w odniesieniu do innych wiadomości, związana z pierwszeństwem przesłania; priorytet ma zastosowanie przede wszystkim w systemach czasu rzeczywistego.

Wagę i priorytet wiadomości powinno się rozpatrywać osobno, niestety, w rzeczywistych rozwiązaniach na ogół nie spotyka się takiego podwójnego cechowania, dlatego też w niniejszej pracy atrybutów tych używa się zamiennie.

Nadawane wiadomości tworzą **kolejkę** nadawczą. Ich kolejność (uszeregowanie) jest najczęściej związana z czasem nadejścia. Wysyłanie wiadomości odbywa się według nowej kolejności, spełniającej pewne wymogi i realizującej cele optymalizacji.

Formalny zapis kolejki ma postać:

kolejka nadawcza: $Q : (m_1, \dots, m_n)$, n – liczba wiadomości;

nowe uszeregowanie: QN – jest pewną permutacją Q , $QN = f(Q)$.

Optymalne wykorzystanie łącza komunikacyjnego w komunikacji asynchronicznej narzuca wysyłanie wiadomości połączonych w strumień, który następnie jest dzielony na pakiety sieciowe o stałym rozmiarze, charakterystycznym dla danej sieci. Jako założenie przyjmuje się, że pakiety te są wysyłane w równych odstępach czasowych. Za wartościową wiadomość uznaje się tylko otrzymaną w całości (poprawnie). W takim systemie czas oczekiwania na wiadomość m_j wyznacza się jako czas przesłania jej i wszystkich wiadomości poprzedzających m_j w kolejce, aż do momentu otrzymania ostatniego pakietu zawierającego dane m_j . Jest to więc funkcja nieliniowa. Wartość czasu oczekiwania dla wiadomości m_j z kolejki wiadomości QN przygotowanej do wysłania, przy uwzględnieniu ewentualnego początkowego zapełnienia bufora q , przedstawia wzór

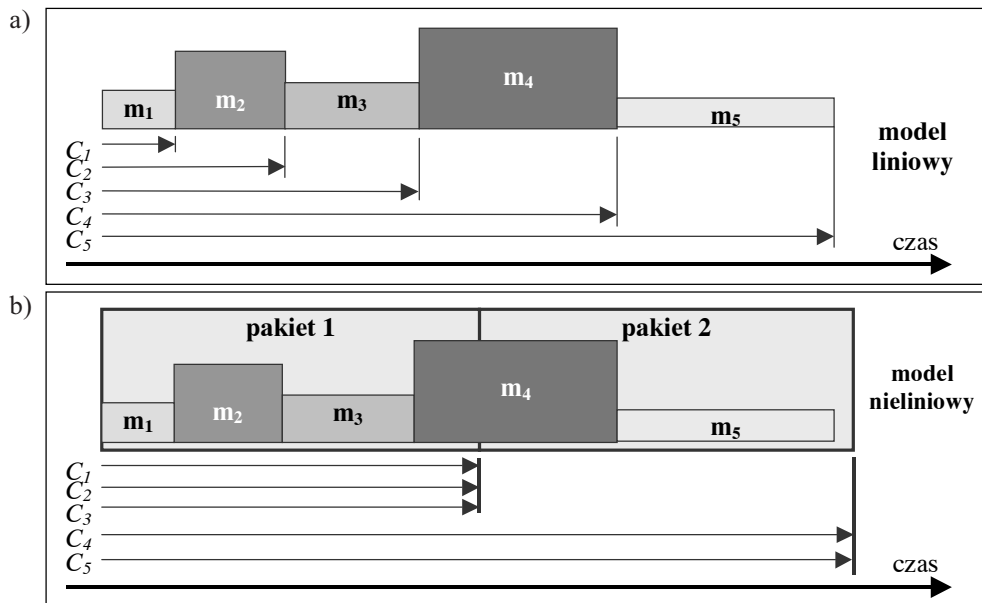
$$C_j = \left[\frac{q + \sum_{i=1}^j l_i}{RP} \right] \cdot CTP \quad (1)$$

gdzie:

- C_j – czas oczekiwania na wiadomość j ,
- q – początkowe zapełnienie bufora sieciowego,
- l_i – długość wiadomości i ,
- RP – rozmiar pakietu sieciowego,
- CTP – czas transmisji pojedynczego pakietu.

Należałoby dodać, że nieliniowość rozpatrywanego problemu dotyczy głównie wiadomości, których długość jest porównywalna z rozmiarem pakietu sieciowego. Model, który zakłada liniową zależność czasu transmisji od długości danych, będzie nazywany w niniejszej pracy modelem liniowym transmisji.

Porównanie czasu oczekiwania na wiadomość C_j dla liniowej oraz nieliniowej zależności czasu transmisji od długości danych przedstawia rysunek 1.



Rys. 1. Porównanie czasu oczekiwania dla liniowego (a) i nieliniowego modelu transmisji (b)

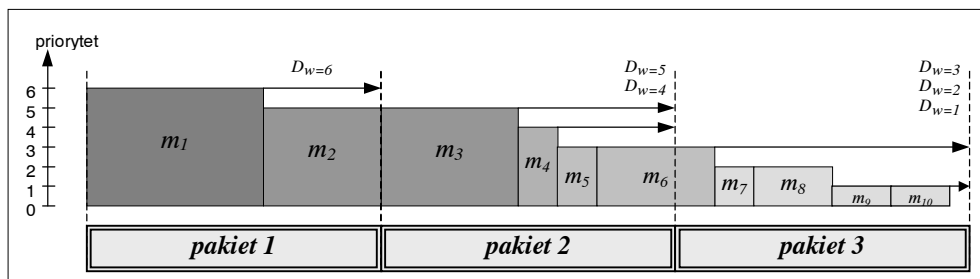
3. Ograniczenia czasu rzeczywistego

W systemach czasu rzeczywistego stosuje się priorytety wiadomości do oznaczania danych, których dostarczenie jest pilne. Na podstawie priorytetów można wyznaczać kolejność wysyłanych wiadomości z kolejki. W większości rozwiązań przyjmuje się, że wiadomość o wyższym priorytecie powinna być wysłana wcześniej niż wiadomość o niższym priorytecie, nawet, jeśli ta mniej ważna oczekuje dłużej w kolejce. Wiadomości o tym samym priorytecie wysyłane są w kolejności nadejścia. Ów algorytm, spełniający warunek (2), w dalszej części pracy będzie określany akronimem FIFO RT

$$QN = (m_1, m_2, \dots, m_n) : w_1 \geq w_2 \geq \dots \geq w_n \quad (2)$$

Moment dostarczenia wiadomości przy uwzględnieniu nieliniowego modelu transmisji to moment, w którym nadszedł ostatni pakiet, zawierający dane wiadomości. W kolejce może znajdować się większa liczba wiadomości o danym priorytecie. Termin dostarczenia

D_i wiadomości m_i o priorytecie w_i wyznacza się, określając czas nadejścia ostatniego pakietu, zawierającego dane ostatniej w kolejce wiadomości o priorytecie w_i . Termin dostarczenia wiadomości o priorytecie x oznaczony będzie jako $D_{w=x}$. Zilustrowane jest to na rysunku 2.



Rys. 2. Wyznaczanie terminów dostarczenia wiadomości

W niniejszej pracy zakłada się, że zestaw wiadomości podczas wysyłania nie zmienia się. Cykliczne lub losowe nadchodzenie nowych wiadomości w trakcie wysyłania danych z kolejki omówiono w [4].

4. Kryteria szeregowania wiadomości

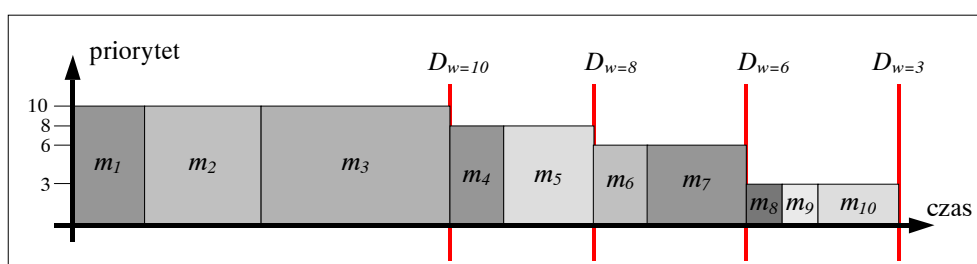
Zagadnienie szeregowania wiadomości jest procesem analogicznym do szeregowania zadań przy założeniu liniowej zależności czasu transmisji danych od ich długości. Podobnie jak przy szeregowaniu niezależnych zadań, szeregowanie wiadomości bez terminów dostarczenia można ocenić kryteriami sumy czasów oczekiwania ($\sum C_j$) i ważonej sumy czasów oczekiwania ($\sum w_j C_j$).

Kryterium sumy czasów oczekiwania wiąże się z oceną sprawności przesyłania danych. Pożądana więc jest jego minimalizacja. W przypadku systemów o liniowej zależności czasu oczekiwania od długości wiadomości do jego optymalizacji można użyć regułę SPT [1]. Optymalizacja kryterium sumy czasów oczekiwania z uwzględnieniem problemu podziału strumienia na pakiety sieciowe, także dla przypadku systemów czasu rzeczywistego, została przedstawiona w [2].

Kryterium sumy ważonych czasów oczekiwania wnosi, w porównaniu do kryterium sumy czasów oczekiwania, różnicowanie udziału wiadomości w końcowej sumie. Dokonuje się tego przez wymnożenie czasu oczekiwania wiadomości przez jej wagę. Szacowanie udziału wiadomości przez jej priorytet pozwala na określenie kosztu oczekiwania wiadomości (albo strat), czyli dla wszystkich wiadomości w systemie można określić **koszt całkowity**. Wskaźnik ważonej sumy czasów oczekiwania przedstawia wzór

$$\sum w_j C_j = w_1 C_1 + w_2 C_2 + \dots + w_n C_n \quad (3)$$

Minimalizację wskaźnika $\sum w_j C_j$ dla systemu o liniowej zależności czasu oczekiwania na wiadomość od jej długości można przeprowadzić stosując regułę WSPT [1]. Dla modelu rozszerzonego o terminy dostarczeń zaproponowano w [4] algorytm RT STD, który polega na ustawieniu wiadomości w kolejce według nierosnącego priorytetu, a w obrębie grupy wiadomości o tym samym priorytecie – według niemalejącej długości. Szeregowanie takie zachowuje terminy dostarczeń, jednocześnie minimalizując w jedyny możliwy sposób wskaźniki $\sum C_j$ i $\sum w_j C_j$. Zilustrowano to na rysunku 3.



Rys. 3. Schemat przykładowego szeregowania algorytmem RT STD

W przypadku nieliniowego modelu transmisji danych ani reguła WSPT, ani algorytm RT STD nie zawsze dają rozwiązanie minimalne (w odpowiednich kryteriach: bez i z terminami dostarczeń). W pracy [3] udowodniono, iż problem minimalizacji wskaźnika $\sum w_j C_j$ dla przypadku bez ograniczeń czasowych jest zagadnieniem *NP*-trudnym. Można pokazać, że wprowadzenie opisanych w rozdziale 3 ograniczeń czasowych nie zmienia trudności badanej optymalizacji. Dowód jest analogiczny do dowodu przedstawionego w pracy [3]. Korzystając z przykładu w postaci rysunku 2, można wskazać grupę wiadomości o priorytetach $\{3,2,1\}$ o jednakowym terminie dostarczenia, związanym z nadejściem trzeciego pakietu. Z owej grupy tylko część można wybrać do przesłania pakietem drugim. Wybór ten jest transformowalny do problemu plecakowego – transformację przedstawiono w pracy [3].

5. Heurystyczne algorytmy optymalizacji kosztu całkowitego z terminami dostarczeń

W związku z nieistnieniem wielomianowych algorytmów dokładnych dla omawianego zagadnienia należy stworzyć algorytmy heurystyczne.

Algorytmy takie składają się z dwóch części:

- 1) algorytmu konstrukcyjnego – generującego dopuszczalne rozwiązanie początkowe,
- 2) algorytmu poprawy – wykonywanego w celu ulepszenia rozwiązania.

Jako algorytm konstrukcyjny wybrano algorytm RT STD – użycie RT FIFO we wstępnych testach wykazało gorsze wyniki optymalizacji. Po wykonaniu algorytmu RT STD

wyznaczane i przypisywane są terminy dostarczeń wiadomościom w kolejce. Algorytmy poprawy stworzono na bazie algorytmu bąbelkowego. Dzięki temu mają niską złożoność (przeważnie rzędu $O(n^2)$), możliwe jest przerwanie obliczeń w dowolnym momencie. Wszystkie algorytmy poprawy dokonują zamiany sąsiednich wiadomości tylko wówczas, gdy nie powoduje to przekroczenia terminu dostarczenia którejs z nich.

5.1. Algorytmy poprawy

Algorytm I jest oparty na algorytmie sortowania bąbelkowego – wykonywane jest n pętli, w których szacuje się wpływ zamiany kolejności każdych dwóch sąsiednich wiadomości m_i i m_{i+1} w kolejce (i przebiega w przedziale $\langle 1, n-1 \rangle$). Zamiana jest dokonywana wtedy, kiedy zmniejszy wartość wskaźnika $\sum w_j C_j$.

Algorytm II jest oparty na algorytmie bąbelkowym z następującą regułą zamiany wiadomości: dwie wiadomości są zamieniane, jeśli zamiana dokona minimalizacji wskaźnika $\sum w_j C_j$, albo wówczas, gdy zamiana nie zmieni wskaźnika, a wiadomość krótsza jest dalej od początku kolejki niż wiadomość dłuższa.

Algorytm III jest oparty na algorytmie bąbelkowym z następującą regułą zamiany wiadomości: dana wiadomość jest przesuwana bliżej początku kolejki, jeśli ma rozmiar mniejszy od rozmiaru pakietu i oszacowanie korzyści wypada dla niej lepiej.

Algorytm IV jest oparty na algorytmie bąbelkowym z następującą regułą zamiany wiadomości: dana wiadomość jest przesuwana bliżej początku kolejki, jeśli ma rozmiar mniejszy od rozmiaru pakietu i ważone oszacowanie jej dopuszczalnej trasy wypada dla niej lepiej.

Oszacowanie korzyści przesunięcia wiadomości polega na wyznaczeniu drogi, jaką może ona przebyć od obecnego miejsca w kolejce w stronę początku tak, aby nie zmieniać czasów oczekiwania mijanych wiadomości. W efekcie tego wyznaczana jest liczba pakietów, o którą zmniejsza się czas oczekiwania.

Oszacowanie dopuszczalnej trasy przesunięcia wiadomości polega na wyznaczeniu drogi, jaką może ona przebyć od obecnego miejsca w kolejce w stronę początku tak, aby nie zmieniać czasów oczekiwania mijanych wiadomości. Wynikiem jest droga, czyli liczba wiadomości, które są mijane.

5.2. Modyfikacja algorytmów

Dodatkową modyfikacją może być zmiana kierunków przesuwania się bąbelków – od końca do początku kolejki.

5.3. Algorytmy heurystyczne

Przedstawione algorytmy konstrukcyjne i poprawy w odpowiednio dobranych zestawach tworzą poszczególne algorytmy heurystyczne. Do testów wydzielonych zostało czter-

naście zestawów. Ich skład, kolejność, złożoność obliczeniową oraz oznaczenia zamieszczono w tabeli 1.

Przyjęto następujące oznaczenia:

AHx – algorytm heurystyczny nr x,

apX – algorytm poprawy nr X,

t – przeciwny kierunek przesuwania bąbelków (od końca do początku kolejki).

Tabela 1
Oznaczenia algorytmów heurystycznych

Oznaczenie	Algorytmy poprawy (w kolejności wykonywania)	Zł. obl.
AHRT1	aprtI	$O(n^2)$
AHRT1t	aprtIt	$O(n^2)$
AHRT2	aprtII	$O(n^2)$
AHRT2t	aprtIIIt	$O(n^2)$
AHRT3	aprtIII, aprtI	$O(n^3)$
AHRT3t	aprtIIIIt, aprtIt	$O(n^3)$
AHRT4	aprtIV, aprtI	$O(n^3)$
AHRT4t	aprtIVt, aprtIt	$O(n^3)$
AHRT5 mix	aprtIII, aprtIV, aprtII, aprtIIIIt, aprtIVt, aprtIIIt	$O(n^3)$
AHRT5t mixt	aprtIIIIt, aprtIVt, aprtIIIt, aprtIII, aprtIV, aprtII	$O(n^3)$

6. Wyniki eksperymentalne

W celu przetestowania algorytmów została napisana aplikacja testująca. Generuje ona losowe zestawy wiadomości przy zadanych parametrach:

- liczbie wiadomości (N),
- minimalnej i maksymalnej ich długości (L_{min} , L_{max}),
- rozmiarze pakietu sieciowego (RP),
- maksymalnym poziomie priorytetów (P_{max}).

Losowanie jest oparte na rozkładzie jednostajnym.

Przy ocenie działania algorytmu będą brane pod uwagę:

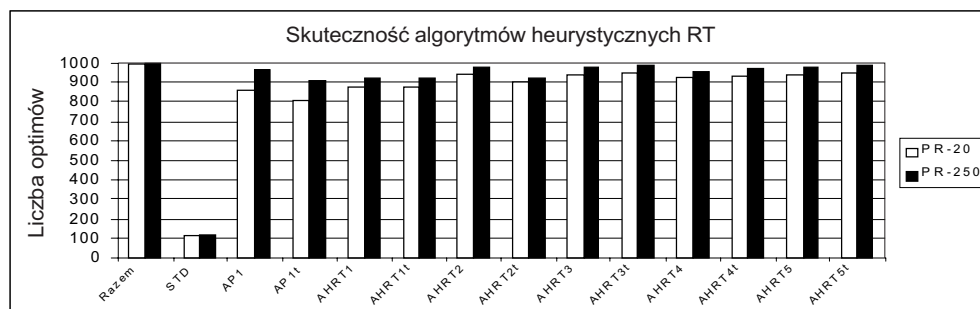
- **skuteczność:** liczba osiągniętych minimów (w porównaniu z przeglądem pełnym lub innymi algorytmami, jeśli przegląd pełny nie jest możliwy);
- **wydajność:** oznacza ogólną efektywność algorytmu i ma zastosowanie w jego porównaniu z innymi algorytmami (dla tych samych danych oczywiście); sumy ważone algorytmów są zawsze niemniejsze od wyniku przeglądu pełnego, zatem suma sum ważonych dla tych samych parametrów przedstawia ogólną sprawność algorytmu.

Wykonano dwie serie symulacji dla parametrów zamieszczonych w tabeli 2.

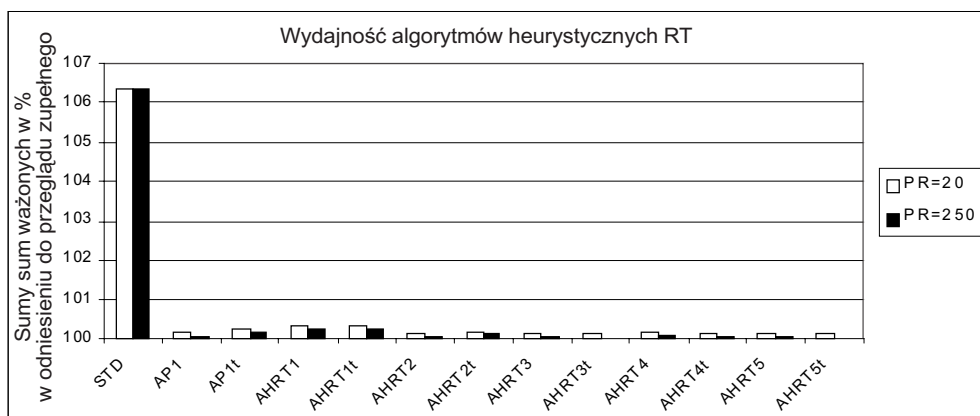
Tabela 2
Parametry serii 1 i 2

Seria	N	Pmax	Lmin	Lmax	RP
1	12	20	4	100	50
2	12	250	4	100	50

Otrzymane dla obu serii wyniki skuteczności przedstawione są na rysunku 4, natomiast wydajność algorytmów ukazano w odniesieniu procentowym do wyników przeglądu zupełnego na rysunku 5.



Rys. 4. Skuteczność algorytmów heurystycznych w seriach 1 i 2



Rys. 5. Wydajność algorytmów heurystycznych czasu rzeczywistego w seriach 1 i 2

6.1. Wnioski

Na podstawie otrzymanych wyników można wyciągnąć następujące wnioski:

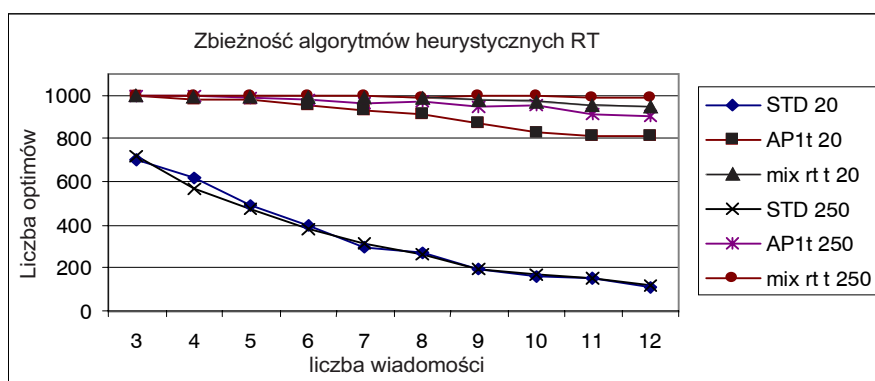
- Największą wydajność osiągnął algorytm AHRT5t, który w obu seriach był najlepszy, przy czym w serii drugiej algorytm AHRT3t miał identyczne wyniki.
- Względne wyniki wydajności i skuteczności algorytmu standardowego w porównaniu z przeglądem zupełnym różnią się nieznacznie dla różnych zakresów priorytetów.
- Algorytm standardowy okazał się być mało wydajny (ponad 6% błędów w obu seriach), a także mało skuteczny – liczba osiągniętych przez niego minimów nieznacznie przekraczała 100 symulacji na 1000 przeprowadzonych.
- Łączne wyniki dla wszystkich algorytmów są zadowalające – 99% symulacji dla 12 wiadomości miało optymalne wyniki znajdowane przez algorytmy heurystyczne.
- Algorytmy przesuwające bąbelki w kierunku od końca kolejki do jej początku dawały na ogół lepsze rezultaty.

6.2. Zbieżność algorytmów heurystycznych

Skuteczność algorytmów heurystycznych wydaje się wysoka dla 12 wiadomości, dlatego eksperymentalnie prześledzono jak zachowują się one wraz ze wzrostem rozmiaru problemu. W tabeli 3 zamieszczono parametry kolejnych serii, natomiast wyniki naniesiono na wykres (rys. 6).

Tabela 3
Parametry serii 3 i 4

Seria	N	Pmax	Lmin	Lmax	RP
3	3, ..., 12	20	4	100	50
4	3, ..., 12	250	4	100	50



Rys. 6. Zbieżność algorytmów heurystycznych czasu rzeczywistego

Analizując otrzymane wyniki można stwierdzić, iż zaproponowane zestawy algorytmów (mix rt t) osiągają bardzo dobrą zbieżność do rozwiązania optymalnego na przedziale liczby wiadomości 3–12, podczas gdy algorytm standardowy już dla 10 wiadomości nie osiąga 20% optimum. Uzasadnia to stosowanie zaproponowanych algorytmów.

Literatura

- [1] Sysło M.M., Narsingh D., Kowalik J.S.: *Algorytmy optymalizacji dyskretnej z programami w języku Pascal*. Warszawa, PWN 1993, rozdział 4.2.5
- [2] Piórkowski A.: *Szeregowanie wiadomości z uwzględnieniem podziału na pakiety sieciowe w systemach wiadomości kolejgowanych*. W: Materiały Konferencyjne IX Konferencji Systemów Czasu Rzeczywistego, Ustroń 2002
- [3] Piórkowski A.: *Heurystyczne algorytmy optymalizacji kosztu całkowitego w systemach wiadomości kolejgowanych*. W: Półrocznik AGH Automatyka, t. 7, 2003
- [4] Piórkowski A.: *Optymalizacja komunikacji asynchronicznej w systemach rozproszonych czasu rzeczywistego*. Kraków, EAIiE AGH 2004 (Praca doktorska)