

Wojciech Bożejko*, Mieczysław Wodecki**

Problem rozsyłania informacji w sieciach komputerowych

1. Wstęp

Istotnym czynnikiem decydującym o efektywności współczesnych systemów informatycznych jest zapewnienie sprawnej komunikacji pomiędzy użytkownikami. Jako protokół komunikacji wybiera się zazwyczaj e-mail, który pomimo wielu wad jest obecnie powszechnie stosowany. Komunikacja przy użyciu tego protokołu jest realizowana przez liczne witryny internetowe o charakterze społecznościowym, platformy aukcyjno-handlowe, serwisy udostępniające skrzynki poczty elektronicznej, itd. Dla wielu firm jest to dzisiaj jedyny sposób komunikowania się z klientami. Poprzez sieć rozsyłane są: zamówienia, powiadomienia, potwierdzenia wykonanych czynności, reklamy, itp. Są wśród nich informacje pilne (np. aktywacyjne) oraz takie, które mogą być wysłane z pewnym opóźnieniem (np. reklamy). Informacje są do sieci wysyłane poprzez kanały o różnej przepustowości. W każdym przypadku, a szczególnie przy dużej liczbie generowanych automatycznie informacji, problemem staje się obsługa (terminowa wysyłka), a więc wyznaczenia informacjom odpowiednich kanałów oraz ustalenie kolejności ich wysyłania.

W rozpatrywanym modelu informacje są generowane przez informatyczne systemy źródłowe i przesyłane do modułu zarządzającego wysyłką. Proces ten odbywa się w czasie rzeczywistym, bez czekania oraz ograniczeń przepustowości łączy. Z każdą informacją jest związany: rozmiar, priorytet, dopuszczalny termin wysłania oraz kara za niedotrzymanie tego terminu. Napływające informacje są zapisywane w lokalnej bazie danych i przechowywane aż do momentu ich wysłania. Co pewien ustalony okres czasu (takt), następuje przydział informacji do kanałów oraz ustalana jest kolejność ich wysyłki. Aby zminimalizować ewentualne kary za przekroczenie terminu wysyłki, przy ustalaniu kolejności uwzględnia się atrybuty informacji. Wysyłka informacji do poszczególnych odbiorców odbywa się równoległe z wykorzystaniem wielu kanałów wyjściowych. Każdy kanał ma określoną przepustowością, tj. wielkość danych wysłanych w jednostce czasu.

* Politechnika Wroclawska, Instytut Informatyki, Automatyki i Robotyki

** Uniwersytet Wroclawski, Instytut Informatyki

W pracy przedstawiamy problemy optymalizacji w systemach zarządzających wysyłką informacji w sieciach. Do ich rozwiązywania stosujemy algorytmy oparte na metodzie przeszukiwania z tabu. Przeprowadzono także eksperymenty obliczeniowe.

2. Definicje i oznaczenia

Rozpatrywany w pracy problemu rozsyłania informacji w sieciach komputerowych można sformułować następująco:

PROBLEM: Dany jest zbiór informacji $I = \{1, 2, \dots, n\}$ oraz zbiór kanałów wyjściowych $K = \{1, 2, \dots, m\}$, przez które informacje będą rozesyłane w sieci. Każda informacja ma ustalony rozmiar i termin wysyłki, a każdy kanał – maksymalną przepustowość. Problem polega na przydzieleniu informacji do kanałów oraz na wyznaczeniu kolejności ich wysyłki, aby zoptymalizować pewne kryterium. Muszą być przy tym spełnione następujące ograniczenia:

- (i) każda informacja może być wysłana tylko przez jeden kanał,
- (ii) dowolny kanał można wysłać jednocześnie co najwyżej jedną informację,
- (iii) nie może być przekroczona przepustowość kanału,
- (iv) wysyłanie informacji nie może być przerwane.

Oznaczmy przez b_k przepustowość kanału $k \in K$, tj. maksymalny sumaryczny rozmiar informacji, jakie można wysłać w ciągu jednej jednostki czasu. Dalej, niech r_i będzie *rozmiarem* wiadomości $i \in I$, d_i – *najpóźniejszym żądanym terminem* rozpoczęcia wysyłki, a w_i – *współczynnikiem kary* za spóźnienie. Jeżeli S_i jest *terminem rozpoczęcia* wysyłki informacji $i \in I$, to $T_i = \max\{0, S_i - d_i\}$ nazywamy *spóźnieniem*, a $w_i T_i$ *karą* za spóźnienie wysyłki. Wówczas, $\sum_{i=1}^n w_i T_i$ jest karą za nieterminowe rozsyłanie wszystkich informacji ze zbioru I .

Problem rozsyłania informacji w sieci (w skrócie problem **RIwS**) polega na przydzieleniu informacji do kanałów oraz ustaleniu terminów ich wysyłki tak, *aby kara za nieterminowe rozsyłanie informacji* $\sum_{i \in I} w_i T_i$ była minimalna. Muszą być przy tym spełnione ograniczenia (i)–(iv).

Niech I^k będzie zbiorem informacji do wysłania przez k -ty kanał. Ciąg zbiorów

$$Q = [I^1, I^2, \dots, I^m]$$

takich, że

$$\bigcup_{k=1}^m I^k = I \text{ oraz } I^k \cap I^l = \emptyset, k \neq l, k, l = 1, 2, \dots, m,$$

nazywamy *przydziałem informacji ze zbioru I do kanałów ze zbioru K* (w skrócie **przydziałem informacji do kanałów**). W szczególności I^k ($k \in K$) może być zbiorem pustym. Oznaczmy przez P zbiór wszystkich takich możliwych przydziałów.

Jeżeli dokonano przydziału informacji do kanałów, wówczas wyznaczenie optymalnych terminów wysyłki (w tym i kolejności wysyłki) sprowadza się do rozwiązania pewnego silnie NP-trudnego problemu szeregowania zadań, tj. jednomaszynowego problemu szeregowania z minimalizacją sumy kosztów spóźnień. W literaturze jest on oznaczany przez $1 \| \sum w_i T_i$. Algorytm optymalny jego rozwiązywania, oparty na metodzie podziału i ograniczeń, przedstawiono w pracy [5], a przybliżony oparty na metodzie przeszukiwania z tabu – w pracy [2]. Obszerny przegląd metod i algorytmów rozwiązywania wielu problemów szeregowania jest zamieszczony w monografii [6].

Dla dowolnego przydziału informacji do kanałów $Q = [I^1, I^2, \dots, I^m]$ ($Q \in P$) przez $\Pi^k(Q)$ oznaczamy zbiór wszystkich permutacji elementów zbioru I^k , a przez

$$\pi(Q) = (\pi_1(Q), \pi_2(Q), \dots, \pi_m(Q)),$$

konkatenację (złożenie) m permutacji, gdzie $\pi_k(Q) \in \Pi^k(Q)$. Niech

$$\Pi(Q) = \Pi^1(Q) \times \Pi^2(Q) \times \dots \times \Pi^m(Q),$$

będzie zbiorem wszystkich takich konkatenacji.

Wobec tego, jeżeli $Q = [I^1, I^2, \dots, I^m]$ jest pewnym przydziałem informacji do kanałów, to zbiór $\Pi(Q)$ zawiera wszystkie permutacje (możliwe kolejności) wysyłania wiadomości ze zbioru I .

Dla ustalenia uwagi, niech $\pi^k(Q) = (\pi^k(1), \pi^k(2), \dots, \pi^k(v))$ będzie pewną permutacją (kolejnością wysyłania) informacji ze zbioru $I^k \in Q$ ($Q \in P$) przez k -ty kanał. Jeżeli wysyłanie informacji rozpoczyna się w chwili 0, to najwcześniejszy możliwy termin wysyłania i -tej informacji (tj. informacji $\pi^k(i)$)

$$S_i = \sum_{j=1}^{i-1} b_k / r_j \quad (1)$$

Przez

$$\Omega = \{(Q, \pi(Q)) : Q \in P \wedge \pi(Q) \in \Pi(Q)\},$$

oznaczamy zbiór par, których pierwszy element jest przydział do poszczególnych kanałów elementów zbioru I , a drugi – konkatenacją permutacji tych podzbiorów. Dowolne rozwiązanie dopuszczalne problemu **RIwS** może być reprezentowane przez parę $\Theta = (Q, \pi(Q)) \in \Omega$, przy czym terminy rozpoczęcia wysyłania poszczególnych informacji są wyznaczone z (1). Łatwo sprawdzić, że takie rozwiązanie spełnia ograniczenia (i)–(iv). Wówczas, koszt rozsyłania

$$F(Q, \pi(Q)) = \sum_{i=1}^m \sum_{j=1}^{n_i} w_j T_j \quad (2)$$

gdzie $n_k = |I^k|$ jest liczbą informacji wysyłanych przez k -ty kanał.

Wobec tego rozpatrywany w pracy problem rozsyłania informacji w sieci sprowadza się do wyznaczeniu optymalnej pary $(Q^*, \pi(Q^*))$ takiej, że

$$F(Q^*, \pi(Q^*)) = \min\{F(Q, \pi(Q)) : (Q, \pi(Q)) \in \Omega\}.$$

3. Metoda rozwiązania

Ze względu na moc zbioru informacji I , istnieje wykładnicza liczba możliwych przydziałów informacji do kanałów. Po dokonaniu przydziału, dla każdego kanału należy wyznaczyć optymalną kolejność wysyłania informacji (sprowadza się to do rozwiązania silnie NP-trudnego problemu $1 \parallel \sum w_i T_i$). Wobec tego, problem **RIWS** wymaga rozwiązania wykładniczej liczby silnie NP-trudnych problemów. Dlatego też do jego rozwiązania będziemy stosowali algorytm przybliżony sprowadzający się do iteracyjnego wykonywania dwóch następujących kroków:

Krok 1: Wyznaczenie przydziału wysyłanych informacji do kanałów.

Krok 2: Wyznaczenie kolejności wysyłania informacji przez każdy z kanałów (rozwiązanie problemu $1 \parallel \sum w_i T_i$).

Zarówno w Kroku 1, jak i Kroku 2 będą stosowane algorytmy oparte na metodzie przeszukiwania z tabu.

3.1. Problem przydziału informacji do kanałów

Problem rozbicia (partycji) zbioru informacji I na m podzbiorów I^1, I^2, \dots, I^m z kryterium (2) jest NP-trudny (bowiem szczególnym jego przypadkiem jest problem podziału zbioru). Dlatego też do jego rozwiązania stosujemy algorytm przybliżony oparty na metodzie przeszukiwania z tabu.

Generalnie, metoda ta polega na iteracyjnym polepszaniu bieżącego rozwiązania poprzez lokalne przeszukiwanie. Rozpoczyna się od pewnego rozwiązania początkowego (startowego) x^0 . W każdej iteracji, dla bieżącego rozwiązania x^i , wyznacza się jego otoczenie $N(x^i)$ – podzbiór zbioru rozwiązań dopuszczalnych. Otoczenie jest generowane przez ruchy, tj. ustalone przekształcenia rozwiązania x^i . Następnie, z otoczenia jest wyznaczany najlepszy element x^{i+1} , który przyjmuje się za bieżące rozwiązanie w następnej iteracji. Dopuszcza się przy tym możliwość zwiększania wartości funkcji celu (przy wyznaczaniu nowego bieżącego rozwiązania), aby w ten sposób zwiększyć szansę na osiągnięcie minimum globalnego. Takie ruchy „w górę” należy jednak w pewien sposób kontrolować, ponieważ w przeciwnym razie po osiągnięciu minimum lokalnego nastąpił by szybki do niego powrót. Aby zapobiec generowaniu, w nowych iteracjach rozwiązań niedawno rozpatrywanych (powstawaniu cykli), zapamiętuje się je (ich atrybuty) na liście rozwiązań zakazanych, tzw. liście tabu.

Poniżej opisujemy w skrócie najważniejsze elementy takiego algorytmu rozwiązywania rozpatrywanego w pracy problemu **RIWS**.

Otoczenie

Niech $\Theta = (Q, \pi(Q))$ będzie pewnym rozwiązaniem, gdzie $Q = [I^1, I^2, \dots, I^m]$ jest przydziałem wiadomości do kanałów. Otoczenie jest generowane przez ruchy, polegające na przenoszeniu pojedynczych informacji przydzielonych do poszczególnych kanałów.

Ruch $v = (I^k, I^l, \alpha)$, gdzie informacja $\alpha \in I^k$ polega na przeniesieniu α ze zbioru I^k do I^l (tj. wykonanie operacji $I^k := I^k \setminus \{\alpha\}$, $I^l := I^l \cup \{\alpha\}$). Generuje on nowe rozbitcie $v(Q)$ zbioru informacji I oraz nowe rozwiązanie dopuszczalne Θ' problemu **RIWS**. Wyznaczając otoczenie rozwiązania $(Q, \pi(Q))$, stosujemy więc ruchy postaci $v = (I^k, I^l, \alpha)$, gdzie $\alpha \in I^k$ jest informacją o **największej** karze za spóźnienie wysyłaną przez k -ty kanał. Jeżeli wszystkie informacje są wysyłane terminowo, to przy generowaniu otoczenia kanał taki pomijamy. Tak wyznaczone otoczenie ma co najwyżej $m(m-1)/2$ elementów.

Wykonanie ruchu $v = (I^k, I^l, \alpha)$ generuje z $\Theta = (Q, \pi(Q))$ nowe rozwiązanie Θ' . Ponieważ zmieniają się jedynie zbiory I^k oraz I^l rozbitcia, więc zmianie ulegają jedynie dwa składniki funkcji kary (pierwszej sumy w (2)), tj. dla $i = k$ oraz $i = l$. Do wyznaczenia kosztów wysyłania informacji przez kanał k oraz l stosujemy algorytm konstrukcyjny COVERT (Potts i Van Wassenhove [4], a także [3]) rozwiązywania jednomaszynowego problemu szeregowania zadań oznaczany przez $1 \parallel \sum w_i T_i$. Jeżeli w miejsce składnika k i l sumy (2) wstawimy wyznaczone przez ten algorytm koszty, wówczas otrzymamy górne ograniczenie wartości funkcji celu dla rozwiązania Θ' . Jest ono stosowane w algorytmie przeszukiwania z tabu do wyznaczenia elementu z otoczenia. Procedura generowania i wyznaczania tego elementu ma złożoność obliczeniową $O(n^2 m^2)$. Ostatecznie, z otoczenia wybieramy element o najmniejszej wartości górnego ograniczenia.

Lista tabu

Jeżeli nowe rozwiązanie wygenerowano, wykonując ruch $v = (I^k, I^l, \alpha)$, wówczas na liście tabu jest pamiętana trójka (k, l, α) . W algorytmie zastosowano listę (kolejkę FIFO) o długości siedem. Jeżeli na liście tabu znajduje się trójka (i, j, β) , to generując otoczenie pomijamy ruchy $v = (I^i, I^j, \beta)$ oraz $v' = (I^j, I^i, \beta)$. Algorytm przeszukiwania z tabu kończy (Kryterium_Stopu) działanie po wykonaniu n iteracji.

3.2. Wyznaczanie terminów wysyłania informacji

Wyznaczenie optymalnej kolejności wysyłania wiadomości przez pojedynczy kanał jest równoważne rozwiązaniu problemu szeregowania zadań na jednej maszynie z kryterium $\sum w_i T_i$. W monografii [6] problem ten został dokładnie opisany. Algorytmy optymalne umożliwiają rozwiązywanie w rozsądnym czasie przykładów z liczbą zadań nieprzekraczającą 50 (80 w środowisku wieloprocesorowym). Dlatego do jego rozwiązywania stosujemy algorytm przybliżony oparty na metodzie przeszukiwania z tabu TS+M,

zamieszczony w pracy Bożejki, Grabowskiego i Wodeckiego [2]. Algorytm kończy działanie po wykonaniu n_k (liczba informacji wysyłanych przez k -ty kanał) iteracji.

4. Eksperymenty obliczeniowe

Dla rozpatrywanego w pracy problemu rozsyłania informacji w sieci brak w literaturze zarówno algorytmów rozwiązywania, jak i przykładów testowych. Stąd trudności w ocenie jakości wyznaczanych rozwiązań przez zamieszczony w pracy algorytm.

Eksperymenty obliczeniowe przeprowadzono na przykładach z liczbą kanałów $m = 2$ oraz liczbą informacji $n = 20, 30, 40, 50, 60$ i 70 . Rozmiary informacji r_i generowano losowo (z rozkładem jednostajnym na przedziale $[1,10]$) w taki sposób, aby $\sum_{i=1}^n r_i = 2b$. Przyjęto ponadto $d_i = b$, $w_i = 1$ ($i = 1, 2, \dots, n$) oraz nieograniczoną przepustowość każdego kanału. Dla każdego n generowano 25 przykładów. Można wykazać, że optymalne rozwiązanie takiego przykładu może być wyznaczone przez algorytm rozwiązywania problemu podziału zbioru (na dwa podzbiory). Do jego rozwiązywania zastosowano algorytm oparty na metodzie programowania dynamicznego (Bellman, Dreyfus [1]). Obliczenia algorytmu rozwiązywania problemu **RIWS** wykonano na komputerze z procesorem 2.8 GHz. Liczba iteracji algorytmu (wykonań Kroku 1 i Kroku 2) wynosi $\ln n$. Średni błąd względny wartości funkcji celu dla rozwiązań wyznaczonych przez algorytm przeszukiwania z tabu, w stosunku do wartości optymalnych, przedstawiono w tabeli 1.

Tabela 1
Średni błąd względny oraz średni czas obliczeń

Liczba informacji n	Średni błąd względny (%)	Średni czas obliczeń (s)
20	0,8	0,2
30	1,7	0,3
40	1,6	0,5
50	2,3	0,9
60	4,9	2,2
70	4,6	3,5
średnio	2,7	1,3

Zarówno średni błąd względny jak i czas obliczeń są niewielkie. Wykonano także obliczenia dla znacznie większych przykładów. Dla liczby informacji $n = 10^6$ i liczby kanałów $m = 100$ czas obliczeń algorytmu wynosi kilkadziesiąt sekund. Jest on zapewne w pełni akceptowalny z punktu widzenia zastosowań.

5. Podsumowanie

W pracy przedstawiono problem rozsyłania dużej liczby informacji w sieciach komputerowych z wykorzystaniem wielu kanałów o różnej przepustowości. Poszczególne informacje należy wysłać przed upływem ustalonego terminu, którego przekroczenie powoduje naliczenie kary. Jako kryterium optymalizacyjne przyjęto sumę kar za przekroczenie terminów wysyłki. W rozpatrywanym problemie należy więc przydzielić informacje do kanałów oraz ustalić kolejność ich wysyłania, aby zminimalizować sumę kar. Do jego rozwiązania zastosowano algorytm, w którym iteracyjnie rozwiązywane są dwa problemy: przydziału informacji do kanałów oraz ustalenie kolejności ich wysyłki. Oba należą do klasy problemów NP-trudnych, dlatego do ich rozwiązywania zastosowano algorytmy przybliżone oparte na metodzie przeszukiwania z tabu.

Praca finansowana z projektu badawczego MNiSW Nr N N514 232237.

Literatura

- [1] Bellman R.E., Dreyfus S.E., *Programowanie dynamiczne*. Państwowe Wydawnictwo Ekonomiczne, Warszawa 1967.
- [2] Bożejko W., Grabowski J., Wodecki M., *Block approach-tabu search algorithm for single machine total weighted tardiness problem*. Computers&Industrial Engineering, 50, 2006, 1–14.
- [3] Crauwels H.A.J., Potts C.N., Van Wassenhove L.N., *Local search heuristics for the single machine total weighted tardiness scheduling problem*. INFORMS Journal on Computing, vol. 10, No. 3, 1998, 341–350.
- [4] Potts C.N., Van Wassenhove L.N., *Single machine tardiness Sequencing Heuristics*. IIE Transactions, 23, 1991, 346–354.
- [5] Wodecki M., *A block approach to earliness-tardiness scheduling problems*. International Journal on Advanced Manufacturing Technology, 40, 2009, 797–807.
- [6] Wodecki M., *Metody agregacji w problemach optymalizacji dyskretniej*. Wydawnictwo PWr, Wrocław 2009.