

Heuristic algorithms for the optimization of total weighted completion time for asynchronous transmission in a packet data transmission system*

by

Adam Piórkowski¹, Jan Werewka² and Marzena Bielecka¹

¹Department of Geoinformatics and Applied Computer Science
AGH University of Science and Technology
al. Mickiewicza 30, 30-059 Cracow, Poland
<http://www.geoinf.agh.edu.pl>

²Department of Applied Computer Science
AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Cracow, Poland

Abstract: In this paper, the minimization of total weighted completion time (total cost) for asynchronous transmission in distributed systems is discussed. Special attention has been paid to the problem of message scheduling on the sender side. Messages to be sent form a queue, therefore the order in which they are to be sent has to be set. Scheduling algorithms can be chosen to optimize scheduling criteria such as total completion time or total weighted completion time. The message scheduling problem becomes complicated considerably when the transmitted data stream between the sender and the receiver is formed into packets.

The WSPT (Weighted Shortest Processing Time) scheduling rule, which orders messages according to non-decreasing length and weight ratios has been proven to be non-optimal. It has been demonstrated that the problem of minimizing the total weighted completion time is NP-hard. Here, we propose heuristic algorithms for scheduling messages and experimentally evaluate the performance of these scheduling algorithms.

Keywords: total weighted completion time, total cost factor, optimization, heuristic algorithms, packet transmission

1. Introduction

Asynchronous communication is the most popular way of exchanging data in distributed systems. This popularity stems from the numerous advantages of this type of communication, which include the fact that the sender is not blocked

*Submitted: November 2008; Accepted: December 2014

during the sending of messages, and the available transmission bandwidth between the communicating nodes can be fully used. Network transmission servicing is frequently managed by middleware message queuing systems (MQS). The data transmitted in such systems is formatted so as to form a message of specific length and priority. Messages, which at any given moment cannot be sent, are buffered in queues, where they wait to be sent later, when the transmission channel is free.

The scheduling of messages to be sent (on the sender side) has a significant influence on the data transmission quality. Message scheduling methods are usually based on task scheduling solutions. There are numerous analogies between multitask production systems and asynchronous communication. Data transmission systems also have peculiarities, which make them different from other systems. One such property is stream communication, which occurs in a continuous data transmission between nodes (stream). Streams are divided into packets of uniform size, meaning that some messages can be sent in one packet, while messages that are larger than the packet size are fragmented and sent in many packets. The scheduling of these message types is the focus of this paper.

Scheduling algorithms in real systems should have low computational complexity, as so to ensure that delays in sending data caused by the scheduling of messages are minimized. Moreover, if the transmission system is ready to send another packet and the data to be sent is available, then the packet should be sent whether the scheduling algorithm has finished its operation or not. Hence, an algorithm should provide a solution, which is at least acceptable for every computation step.

The authors assumed that network communication is the bottleneck of asynchronous communication. Therefore, the issue of data transmission optimization was analyzed only in terms of the sender. The presented solutions are directly applicable on the assumption of an "impatient" receiver, waiting to receive and process each successive message immediately.

2. Asynchronous communication

In distributed systems, which are oriented at sending messages, synchronous and asynchronous types of communication can be distinguished (Dijkstra, 2002).

In the case of synchronous communication, after a message is sent, the sender is in a state of passive waiting for the receiver to receive and process the message. Such a communication scheme has two basic drawbacks:

- The sender is blocked until the moment the receiver acknowledges the receipt of the message.
- Full transmission bandwidth is not utilized.

Asynchronous communication is an alternative to synchronous communication and has the advantage that the sender is not blocked during the sending of messages. This means that any number of messages can be sent without waiting for receipt notifications, but this is not a simple task. Asynchronous communication enables better use of the allocated transmission bandwidth, but

the messages, which cannot be sent at a given moment, have to be queued.

A complex asynchronous communication solution is offered by Message Queuing (MQ) systems. These systems sit between the senders and the receivers of messages. The software for such systems is called Message Oriented Middleware (MOM).

The main task of MQ systems is to deliver messages from the sender to the receiver. The message sending process is undertaken by the MOM brokers and covers the following stages:

- Reception of data to be sent by the sender as a message to the receiver.
- Queuing data messages to be sent.
- Transmission of queued messages.
- Reception of data messages by the receiver and queuing of messages.
- Acknowledgement of the receipt and waiting for a query for new data to be received.

Attention should be paid to the fact that messages are queued by a broker on the side of the sender. There are situations, in which the quantity of scheduled data exceeds transmission capacity, meaning that some messages are withheld by the broker until the time when the data can be transmitted. The broker puts together the messages into one stream of data to fully use the allotted transmission bandwidth, and then the stream is divided by the transport layer into network packets and sent to the broker on the side of the receiver.

3. Research concerning asynchronous packet communication

One of the basic questions, discussed in terms of asynchronous communication, is the scheduling of messages to be sent. Scheduling has a significant influence on the basic quality of the system's operation. Message scheduling problems (Gajer, 2010) are in a sense similar to task scheduling problems, therefore selected solutions developed for task scheduling are commonly employed for solving data transmission issues. Some interesting examples of scheduling algorithms proposed for message scheduling and their computational complexity are included below.

Ramanathan and Rupnick (1991) proposed certain message scheduling algorithms. The proposed Minimum Cost Scheduling (MCS) algorithm is based on penalties for late deliveries, which are determined for a given node with each arrival of a new message. Upon each new arrival a new message is queued so as to minimize the respective factor value. The computational complexity of the algorithm is low and equals $O(n)$. The principle and conditions of the algorithm (a system with known delivery deadlines) resemble EDF (Earliest Deadline First) and ELF (Earliest Laxity First) algorithms. The presented experimental methods revealed better results (lower cost) for the MCS algorithm when compared to EDF and FIFO.

Scheduling of messages sent in a network with a line topology is presented in Adler et al. (1998). The optimization problem was analyzed for messages with

a known generation time and due time of delivery. It was additionally assumed that exactly one packet was used for sending one message. Optimum message scheduling, with or without the sender buffer, was proved to be an NP-hard task. Similar considerations for a graph (tree, network) topology are presented in Adler et al. (1999).

A solution to the problem of scheduling messages with due times of delivery for distributed systems with complex structure is proposed in Tsai and Shin (1996). The authors also present the experimental results of the proposed algorithm as compared to other well-known scheduling algorithms, e.g.: Longest First (LF), Shortest First (SF), Farthest First (FF), Nearest First (NF), Largest RBR First (LBF) and Smallest RBR First (SBF). The latter two algorithms, LBF and SBF, account for the Remaining Bandwidth Requirement (RBR).

An interesting optimization method, which groups messages for CAN bus and reduces load on the data transmission system was proposed by Dobrin and Fohler (2001). The optimization of the network buffer emptying procedure is presented in Harchol-Balter, Bansal and Schroeder (2000), and Bansal and Harchol-Balter (2001). An experiment was conducted, in which the server control of websites was modified. With this approach, the static query was serviced through the Shortest Remaining Processing Time (SRPT). Such scheduling was proved to considerably reduce the average response time and variance of the response time of a server. The idea of dividing a data packet was forwarded by Zhu, Yu and Doyle (2001). A packet is a document divided into parts by a WWW server. Depending on the requirements, the documents are sent in parts, e.g. by pages.

Nowadays, communication optimization is an important issue, especially in wireless networks. A power control method for minimizing total completion time of user stream packets is presented in Ng, Médard and Ozdaglar (2009). A similar problem is described by Yang and Ulukus (2010).

The optimization of asynchronous communication can be also applied to agent systems (Yang, Liu and Yang, 2002) or to computational environments (Kielmann et al., 1999).

The above considerations reveal that investigations devoted to the optimization of asynchronous communication do not follow one pathway as they are oriented at solving various specific problems.

4. Model of an asynchronous communication

The data transmission model, presented in this paper, accounts for the message queue on the sides of both the sender and the receiver. This model will be used for further analyses for developing message scheduling methods and algorithms on the side of the sender. A discussion concerning the detailed asynchronous communication MOM systems is presented in Huang (2007).

A model of message queuing systems is also shown in the studies reported in Flieder (2005) and in Gawlick (2002), with particular focus on the organization of data, types of messages, and the basics of the transmission of messages in

MOM systems, with further special emphasis on JMS (Java Message Service).

Most of the considerations devoted to message queuing assume complex message distribution models, see, e.g., Wozniak et al. (2014), or use advanced mathematical methods for describing queues (e.g. Atencia, 2014). In the current article the authors are concerned only with transmission between two points (sender and receiver) and are focused on the scheduling problem of messages in a queue. The asynchronous communication and its model are the topic of articles by Shafer and Ahuja (1992), and Ramesh and Perros (2001), which also take FIFO order into account.

For the sake of precision, let us specify that the following general assumptions on data transmission have been made:

- Guaranteed transmission bandwidth, expressed by minimum (or constant) transmission velocity (volume of data sent in a unit of time).
- Most effective use of the medium lies in sending data in an uninterrupted stream of successive scheduled messages.
- Only one stream of messages can be transmitted at a time.
- The transmitted stream is divided into packets, which can only be processed by the receiver when the entire packet is received.
- Data is formatted into packets of uniform length and sent in constant units of time, regardless of the size of the transmitted data.
- Message scheduling time is negligible compared to the transmission time of messages.
- Processing time of received packets (identifying the message and servicing it) is negligible compared to transmission time.
- The receiver waits for new messages all the time ("impatient receiver"). In other words, there are no delays caused by data reception and processing on the side of the receiver.

4.1. Model of a message

A message is a coherent sequence of data, which has definite value to the receiver. No interpretation of the content of sent messages is made in the analyzed system. It was assumed that a message m is a pair:

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

where

l – length of a message,

w – weight of a message.

The length of a message is a number of elementary data units. In this paper, the size of a message (length) is defined in bytes. The weight is a natural number from a given interval and can be considered in terms of two categories, as follows:

- weight – **value** of a given message, considered in this article,

- priority – **urgency** of a message compared to other messages and connected with the priority of sending; priority is mainly applicable in real-time systems.

4.2. Message scheduling model (sender)

The incoming and unsent messages create a queue Q in the system:

$$Q = (m_1, \dots, m_n) . \quad (1)$$

Index i in a message m_i denotes its order in Q . The number of messages n changes over time and depends on the state of the system. After a new set of data arrives, a new order in the sending queue Q is established. Messages are sent according to the new order. Determining the queue's new order is called scheduling. New scheduling of a queue Q_n is a permutation of a queue with n elements, and is denoted as NU (2):

$$NU = f(Q_n) \quad (2)$$

where:

- n – number of messages in the queue,
- f – scheduling algorithm.

4.3. Linear transmission model

The linear dependence of data transmission, which can be adapted to a transmission system with a message queue, is presented in Coulouris, Dollimore and Kindberg (2005). The basic value, describing the quality of service, is completion time C_j of a queued message m_j . This factor is the sum of the time of this message transmission and of the times of transmission of the remaining preceding messages. It can be described by the following formula:

$$C_j = d + \frac{\sum_{i=1}^j l_i}{V} \quad (3)$$

where:

- C_j – completion time of a message m_j ,
- l_j – length of a message m_j ,
- $\sum_{i=1}^j l_i$ – sum of the lengths of the messages preceding m_j and the message m_j ,
- d – delay, initial data to be sent, e.g. an overhead of connection establishment,
- V – velocity of data transmission.

4.4. Non-linear transmission model

Data is not transmitted continuously, when it is divided into network packets, therefore transmission time cannot be considered linear due to this division.

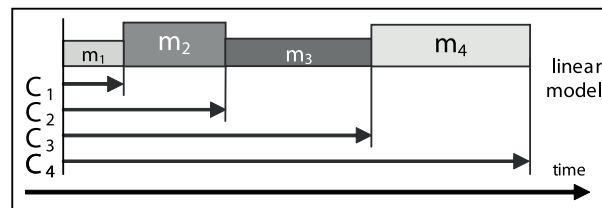


Figure 1. The linear model of network transmission

This phenomenon mainly manifests itself on low-capacity lines (e.g. dial-up connections or GSM), over which successive packets are received in time intervals of hundreds of milliseconds and the processing of the received data is tens to hundreds of times faster than transmission.

For queued and scheduled messages, the time in which they will be delivered to the receiver should be determined. The non-linear completion time $\sum C'_j$ of the message m_j , from the moment it is queued until the moment the receiver receives the message, includes the time required to send the filling of the buffer q and other messages, preceding the message m_j . The value of $\sum C'_j$ is obtained by determining the number of packets into which the messages (m_1, m_2, \dots, m_j) will be divided, and then multiplying it by the transmission time of a single packet. The completion time of delivery for a message m_j in this case can be described by the formula (4).

$$C'_j = \left\lceil \frac{q + \sum_{i=1}^j l_i}{PS} \right\rceil \cdot TTP \quad (4)$$

where:

C'_j – completion time of a message m_j (non-linear model),

l_j – length of a message m_j ,

$\sum_{i=1}^j l_i$ – sum of the lengths of the messages preceding m_j and the message m_j ,

q – initial filling of the buffer (in bytes),

PS – size of the packet,

TTP – transmission time of a single network packet.

A comparison of completion times for linear and non-linear (packet) transmission models is presented in Figs. 1 and 2.

4.5. Non-linear transmission model with priorities

Another model may also be proposed. Messages can be described not only by their weight, but also by priority, indicating the urgency, meaning that the higher-priority messages must be sent sooner than the lower-priority messages. The importance and possibility of optimizing the completion time in non-linear

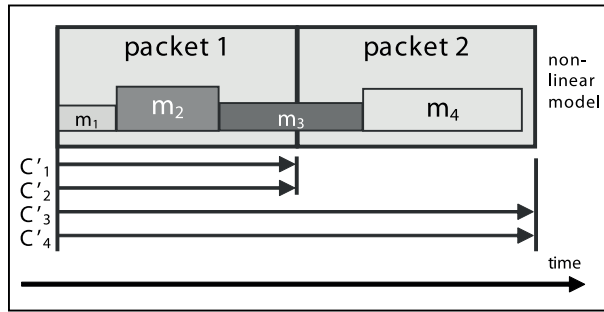


Figure 2. The non-linear model of network transmission

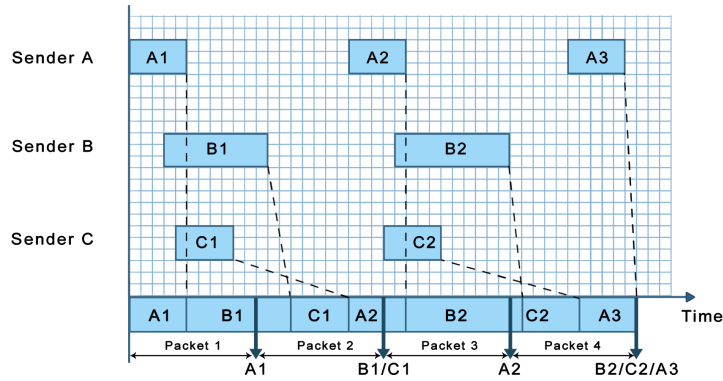


Figure 3. An example of scheduling in the non-linear model with priorities

(packet) transmission can be shown using an example in which there are different data streams (e.g. A, B and C) on the sender's side. It is assumed that the waiting data must be sent instantly when the network is free for transmission, and that current data transmission cannot be stopped or pre-empted. In Figs. 3-5 three possible ways of scheduling data to be sent are illustrated: in Fig.3 data with higher priority (stream A, then B, and then C) are sent first, in Fig.4 the data is sent according to a FIFO rule, for which the data received first is sent first, and in Fig.5 the data is sent so as to minimize the time of data delivery. The data is considered as delivered when the whole packet with the data arrives. The quality factor in the example is the sum of arrival times for all data.

Table 1 and Figs. 3-5 illustrate the results in terms of completion times defined as the sum of arrival times. In the presented paper the more general problem of finding optimal ordering is considered, and a weight factor is given for each data to be sent.

The optimization of the model with priorities goes beyond the considerations of this paper and could be the subject of future work.

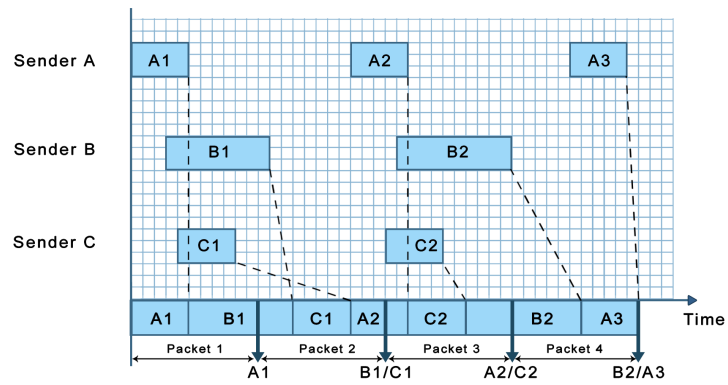


Figure 4. An example of scheduling in the non-linear model according to the FIFO rule

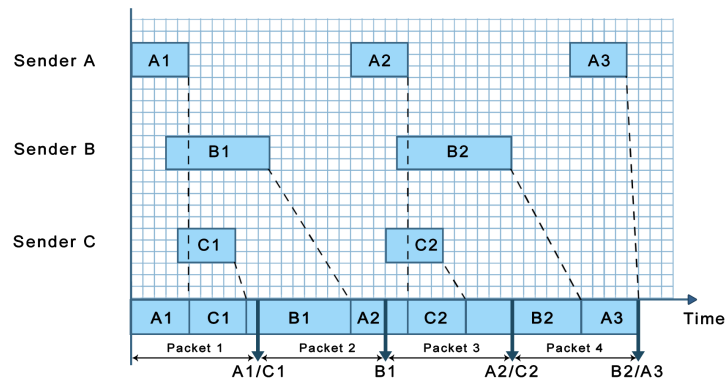


Figure 5. An example of optimal scheduling in the non-linear model

Data	Data priority ordering	FIFO ordering	Optimal ordering
A1	11	11	11
B1	22	22	11
C1	22	22	22
A2	33	33	33
C2	44	33	33
B2	44	44	44
A3	44	44	44
Sum of arrival times	220	209	198

Table 1. Data arrival times in nonlinear model of transmission

5. Message scheduling

For packet communication, the optimization criteria, which are significant for defining the quality of services of such systems, should be determined. The total weighted completion time criterion is selected for the analyses in this paper.

Total weighted completion time (total cost) optimization criterion

Total weighted completion time ($\sum w_j C_j$) is a weighted sum of the receiver's waiting time for the reception of all messages C_j queued on the side of the sender. The criterion value is given by the formula:

$$\sum w_j C_j = w_1 C_1 + w_2 C_2 + \dots + w_j C_j \quad (5)$$

5.1. Minimizing the total weighted completion time

The cost of data transmission expressed by the formula (5) was defined for the continuous transmission. This should be performed on the basis of the analogy between the problem of message scheduling and task scheduling in a multitask operating system for a defined case. For minimizing the criterion $\sum w_j C_j$ the Weighted Shortest Processing Time (SPT) rule can be applied, meaning that messages are sequenced in the non-decreasing order of l_j/w_j ratios. A theorem is presented below, based on this rule, adapted for the needs of asynchronous communication.

THEOREM 1 *WSPT (Smith, 1956): Scheduling according to the non-decreasing l_j/w_j ratio for messages minimizes the sum of the total weighted completion time.*

In case of the packet transmission, analogous operations have to be performed in order to transmit packets, taking into account the non-linear relation

of data transmission time expressed with formula (4). Hence, the question arises as to whether or not the WSPT rule is also valid for the non-linear transmission model (see Piórkowski, 2003). Let us analyze an example to verify this.

Example 1

Let us assume the following data:

$$\begin{aligned} PS &= x, \quad TTP = 1, \\ m_1 &= \langle l_1, w_1 \rangle, \quad l_1 = 0.5x, \quad w_1 = 1; \quad l_1/w_1 = 0.5 \\ m_2 &= \langle l_2, w_2 \rangle, \quad l_2 = 1.5x, \quad w_2 = 4; \quad l_2/w_2 = 0.375. \end{aligned}$$

According to the WSPT principle, the optimization criterion of minimum of $\sum w_j C_j$ is satisfied when messages are sent in the order of non-decreasing ratio l_i/w_i . Therefore, message m_2 should be sent first, then m_1 . In this case the criterion value is obtained from the formula (6):

$$\begin{aligned} \sum w_j C_j' &= w_2 * \left\lceil \frac{l_2}{PS} \right\rceil * TTP + w_1 * \left\lceil \frac{l_2 + l_1}{PS} \right\rceil * TTP = \\ 4 * \left\lceil \frac{1.5x}{x} \right\rceil * 1 + 1 * \left\lceil \frac{1.5x + 0.5x}{x} \right\rceil * 1 &= 10. \end{aligned} \quad (6)$$

Sending of the messages in the reverse order gives, however, a lower result (7):

$$\begin{aligned} \sum w_j C_j' &= w_1 * \left\lceil \frac{l_1}{PS} \right\rceil * TTP + w_2 * \left\lceil \frac{l_1 + l_2}{PS} \right\rceil * TTP = \\ 1 * \left\lceil \frac{0.5x}{x} \right\rceil * 1 + 4 * \left\lceil \frac{0.5x + 1.5x}{x} \right\rceil * 1 &= 9. \end{aligned} \quad (7)$$

Therefore, the WSPT theorem is not true when packet division of data streams is involved. In this case, the issue of optimization for the total weighted completion time criterion can be expected to be a non-linear problem: NP-complete or even NP-hard.

THEOREM 2 *WSPT for non-linear model: The problem of minimizing the total weighted completion time for message scheduling systems when the data stream is divided into packets is NP-hard.*

Proof

Let us analyze a model situation:

Assume a set of n messages, the total length of which does not exceed the size of two packets (in other cases - u packets). Depending on the packet, to which they are allotted, the scheduling times will be given by the formula (8):

$$C_j' = TTP * (1 + x_j) \quad (8)$$

where: x_j - is the decision variable, equal:

$$x_j = \begin{cases} 0 & \text{for } m_j \text{ being in the first packet} \\ 1 & \text{for } m_j \text{ in the second packet} \\ u - 1 & \text{for } m_j \text{ in the } u\text{-th packet.} \end{cases}$$

Allotting messages to the first packet is connected with the size of the packet, namely:

$$\sum l_i \leq PS, \quad \text{where } i : x_i = 0. \quad (9)$$

Then the optimized optimization criterion may assume the following form:

$$\sum w_j C'_j = TTP * \sum w_j (1 + x_j) = TTP * \sum w_j + TTP * \sum w_j x_j. \quad (10)$$

Since $TTP * \sum w_j$ is a constant value, therefore the search for an optimum sum $\sum w_j C'_j$ is reduced to solving the problem for $\sum w_j x_j$ with the constraints (9).

In order to prove that NWSPT (the nonlinear WSPT), which is the optimization problem, is NP-hard, it is necessary to show that its decision version is NP-complete. The corresponding decision problem for the NWSPT has the following form.

Let $Q = \{m_1, \dots, m_n\}$ and $PS > 0$, where, as it has been defined above, Q denotes the set of messages and PS denotes the size of a packet. Let a constant $k > 0$ be given.

Does there exist such a subset of Q , for which the following inequalities are satisfied:

$$\begin{aligned} \sum_{t=1}^{\alpha} l_{j_t} &\leq PS, \\ \forall_{s \in \{\alpha+1, \dots, n\}} (\sum_{t=1}^{\alpha} l_{j_t}) + l_{j_s} &> PS, \\ \sum_{j=1}^n x_j w_j &\leq k, \quad \text{where } x_j \in \{0, 1\} \end{aligned} \quad (11)$$

(meaning that we limit the considerations to the case of two packets).

The NWSPT problem is NP-complete, if:

1. *Its decision version is NP.*

The decision version (11) of the NWSPT problem is NP. Thus, we need to show that a given solution can be verified as a solution to the NWSPT in polynomial time by a non-deterministic Turing machine (NTM). Since, in this case, two sums have to be computed and two comparisons performed, the complexity of verification of a given solution is polynomial.

2. *There exists a polynomial transformation of any known NP-complete problem to the NWSPT problem.*

It is a known result that the binary knapsack problem (KP) is NP-complete (see Cormen et al., 2001, and Júdice, Faustino and Ribeiro, 2002). Thus, the transformation of the KP into the NWSPT problem is presented. The classical KP problem can be described as follows. We are given a set of n items, each of them with a positive profit p_j and a positive weight v_j and knapsack capacity a . The problem consists in finding a subset of items such that total weight does not exceed a and its profit is maximal. It can be formulated in the decision version as follows.

Let $H = \{h_1, \dots, h_n\}$ and $a, b > 0$ be constants. Does there exist such a subset of items for which the following inequalities are satisfied:

$$\begin{aligned} \sum_{j=1}^n v_j y_j &\leq a, \\ \sum_{j=1}^n p_j y_j &\geq b, \text{ where } y_j \in \{0, 1\}. \end{aligned} \quad (12)$$

A polynomial transformation of the KP problem to the NWPST problem is a function $f : D_{KP} \rightarrow D_{NWPST}$ such that, for every instance $I \in D_{KP}$ the answer to the above question is affirmative if and only if for the instance $f(I) \in D_{NWPST}$ the answer is also affirmative and the time of computation of the function f by DMT for every instance $I \in D_{KP}$ is bounded by a polynomial.

The proof of $KP \propto NWPST$:

(\Rightarrow)

Let for $I \in D_{KP}$ the answer be affirmative. So, there exists such a subset of items that $\sum_{j=1}^n v_j y_j \leq a$ and $\sum_{j=1}^n p_j y_j \geq b$, where $y_j \in \{0, 1\}$. Based on the transformation $y_j = x_j - 1$, where $x_j \cdot y_j = 0$, it is easy to show that the KP problem is equivalent to the NWPST problem. Namely, if we put $a = PS$, $p_j = w_j$, $v_j = l_j$ then, the maximization of the following sum,

$$b = \sum_{j=1}^n p_j y_j = \sum_{j=1}^n w_j (1 - x_j) = \sum_{j=1}^n w_j - \sum_{j=1}^n w_j x_j,$$

is equivalent to minimizing the sum

$$\sum_{j=1}^n w_j x_j = \sum_{j=1}^n w_j - b = k.$$

Thus, every instance, which is a solution of the binary KP problem is also a solution of the NWPST problem.

(\Leftarrow)

Let for $I \in D_{NWPST}$ the answer be affirmative. Then there exists such a subset of messages that $\sum_{j=1}^n l_j x_j \leq PS$ and $\sum_{j=1}^n w_j x_j \leq k$, where $x_j \in \{0, 1\}$. Based on the transformation $x_j = y_j - 1$, where $x_j \cdot y_j = 0$, it is easy to show that the NWPST problem is equivalent to the KP problem. Namely, if we put $PS = a$, $w_j = p_j$, $l_j = v_j$ then minimizing the following sum,

$$k = \sum_{j=1}^n w_j x_j = \sum_{j=1}^n p_j (1 - y_j) = \sum_{j=1}^n p_j - \sum_{j=1}^n p_j y_j,$$

is equivalent to maximizing the sum

$$\sum_{j=1}^n p_j y_j = \sum_{j=1}^n p_j - k = b.$$

Thus, every instance, which is a solution of the NWPST problem is also the solution of the KP problem. The time of computation of the function f by DMT is bounded by polynomial $p(N(I_{KP}))$ because, in order to construct data $f(I_{KP}) = I_{NWPST}$, DMT has to rewrite $3n$ values, to sum n numbers and to execute $n + 1$ subtractions. Thus, the decision version of NWPST problem is NP-complete. As the result, the NWPST, (11), is an NP-hard problem. ■

The minimization of the $\sum w_j C_j$ criterion for larger problems requires heuristic algorithms (see Nawrocki et al., 2009), therefore it is productive to analyze how much packet division influences the WSTP algorithm results. This should be also analyzed in the context of message length and packet size. These problems were investigated experimentally.

6. Heuristic algorithms of total weighted completion time optimization for packet transmission

Heuristic algorithms had to be worked out due to the lack of accurate polynomial algorithms for minimizing the optimization criterion of $\sum w_j C_j$. These algorithms consist of two parts:

- Construction algorithm - for generating an admissible initial solution,
- Improvement algorithm - for improving the solution.

Construction algorithms

Preliminary experimental analyses revealed that of the proposed construction algorithms (FIFO, SPT, WSPT), the best initial solution was generated by WSPT. Therefore, only WSPT will be used as a construction algorithm for all the proposed heuristic algorithms. The computational complexity of all the mentioned algorithms does not exceed $O(n^2)$ (average: $O(n \log(n))$).

Improvement algorithms

Four improvement algorithms based on the bubble sort algorithm were proposed for the analyzed case. The bubble sort algorithm guarantees that if computation is interrupted at an arbitrary moment, the obtained solution will not be worse than that of the preceding step, including the construction algorithm.

Algorithm **apI** is based on the bubble sort algorithm; n loops are performed, in which the influence of the changed order of two neighboring messages, m_i and

m_{i+1} , in a queue is analyzed (i belongs to $[1, n - 1]$). A swap is made when the value of $\sum w_j C_j$ decreases.

Algorithm **apII** is based on the bubble sort algorithm with the following rule for message shift: two messages are shifted when this operation minimizes the value of $\sum w_j C_j$, or when the value does not change and the shorter message is closer to the start of the queue than the longer messages.

Algorithm **apIII** is based on the bubble sort algorithm with the following principle of message shift: a given message is shifted closer to the beginning of the queue if its size is less than that of the packet and the profit evaluation is more advantageous for the message. Evaluation of message shift profit consists in finding a path which the message can follow from its present place towards the beginning of the queue without changing the scheduled times of messages which it moves ahead of. As a result, the number of packets implying the shortening of the scheduling time is determined. The calculation of message shift profit evaluation has the additional complexity $O(n)$.

Algorithm **apIV** is based on the bubble sort algorithm with the following principle of message shift: a given message is shifted closer to the beginning of the queue if its size is smaller than that of the packet and the weighted evaluation of its admissible pathway is more favorable. Evaluation of the admissible message shift pathway consists in determining a path, which the message can follow from its present place in the queue towards the beginning of the queue, without changing the scheduled times of messages which it moves ahead of. As a result the "distance" is obtained, i.e. the number of passed messages.

Modification of the improvement algorithms

In the proposed improvement algorithms the modified bubble sort method checks or shifts messages from the beginning to the end of the queue. An additional improvement algorithm modification is proposed which shifts bubbles from the end to the beginning of the queue. In further studies, an opposite direction of bubble shifting (from the end to the beginning of the queue) will be determined in the improvement algorithm (marked by adding the letter "t" to the name of the algorithm, e.g. apIt).

Basing on the selected construction algorithm and the proposed improvement algorithms, heuristic algorithms were created. Their names and content are presented in Table 2.

Moreover, heuristic algorithms, which are the combinations of the improvement algorithms are created. For instance, the heuristic algorithm AH3 consists of two improvement algorithms apIII and apI. Performing successive improvement algorithms assumes that the previous algorithm on the list has been finished.

7. Experimental analyses of heuristic algorithms

The aim of the conducted experiments was to check the hit ratio and average deviation of the heuristic algorithms. The highest number of messages $N = 10$

Name	Improvement algorithms (in the order as they were performed)	Computational complexity
AH1	apI	$O(n^2)$
AH1t	apIt	$O(n^2)$
AH2	apII	$O(n^2)$
AH2t	apIIIt	$O(n^2)$
AH3	apIII, apI	$O(n^3)$
AH3t	apIIIIt, apIt	$O(n^3)$
AH4	apIII, apII	$O(n^3)$
AH4t	apIIIIt, apIIIt	$O(n^3)$
AH5	apIV, apI	$O(n^3)$
AH5t	apIVt, apIt	$O(n^3)$
AH6	apIV, apII	$O(n^3)$
AH6t	apIVt, apIIIt	$O(n^3)$
AH7	apIII, apIV, apII, apIIIIt, apIVt, apIIIt	$O(n^3)$
AH7t	apIIIIt, apIVt, apIIIt, apIII, apIV, apII	$O(n^3)$

Table 2. Proposed heuristic algorithms

was selected for the experiment. For this number, the full enumeration algorithm could be performed in a reasonable amount of time using a tree traversal algorithm (depth-first search). Looking through all permutations of these ten messages using the author's software on a single processor (Pentium 3GHz) computer takes around 10 seconds. The processing time of the heuristic algorithms was negligible, taking only about 20 ms.

The following criteria were taken into account when assessing the operation of the algorithms:

- **hit ratio:** ratio (in %) of the obtained true optimum solution values among all the obtained solutions,
- **average deviation from optimum:** this measure is calculated as the average difference between the tested algorithm value and the optimal solution, related to the optimum value. This measure can be used for comparing the proposed algorithms. The closer its value to zero, the more efficient the optimization:

$$\rho = \frac{1}{N} \sum_{i=1}^N \frac{OPT_i - HEUR_i}{OPT_i} \quad (13)$$

where:

N – number of tests,

$HEUR_i$ – the $\sum w_j C_j$ criterion value achieved by used algorithm for the test (set) i ,

OPT_i – the $\sum w_j C_j$ criterion for the optimal solution for the test (set) i .

An experiment, which simulated message scheduling with the proposed algorithms, was conducted. A series of 1000 simulations was carried out for each set of parameters. The message sets had the following properties:

- the number of messages in a set - N - because of the computational complexity of the full enumeration algorithm and the available computing power was $N = 10$ in all sets,
- packet size - $PS=100$ (units), thanks to which scaling to real applications can be easily performed (e.g., 1460B for TCP/IP),
- weight range - a random number for each message (constant distribution) from $[1, W_{max}]$,
- range of message size - a random number for each message (constant distribution) from $[L_{min}, L_{max}]$.

Preliminary simulations revealed that in the majority of cases the heuristic algorithm AH7 turned out to be the best for minimizing the value of $\sum w_j C_j$, therefore the remaining ones were ignored in the analysis of the results.

The experiment

Experiments, which scheduled a series of 1000 sets of 10 messages for each set of parameters were performed. In each set the length of the messages l was randomly generated with constant distribution from $[4, L_{max}]$; analogously, the message weight values were generated from $[1, W_{max}]$. The assumed size of the packet was $PS = 100$.

The results obtained from the experiments for hit ratios of the heuristic algorithms WSPT and AH7 are listed in Table 3, and the calculated average deviation values from optimum for these algorithms (due to formula 13) are collected in Table 4.

The results of the analyses of the hit ratio of a selected heuristic algorithm as compared to the algorithm WSPT are shown in the plot presented in Fig. 6. The maximum lengths of messages for a given set are presented on the OX axis. The percent of optima congruent with the optimal solution are on the OY axis. The obtained values are introduced on the plot and the data series of a given algorithm and of the same weight ranges ($W_{max} = 10, 20, 100$, respectively) are linked with lines.

The results collected in Table 4 are presented on the plot (Fig. 7) which illustrates the average deviation from the optimum of algorithms AH7 and WSPT as compared to the optimal solution (OS) due to formula (13).

The following conclusions can be drawn on the basis of the conducted experiments:

- for large messages the WSPT solution is close to optimum as the influence of non-linearity intuitively decreases,
- for larger weight ranges, the hit ratio of the heuristic algorithms decreases and the average deviation increases,
- there is a so-called "saddle" of the WSPT accuracy, meaning that the least numbers of optimum solutions are observed with the medium size messages, comparable with the packet size (maximum message size: 150-

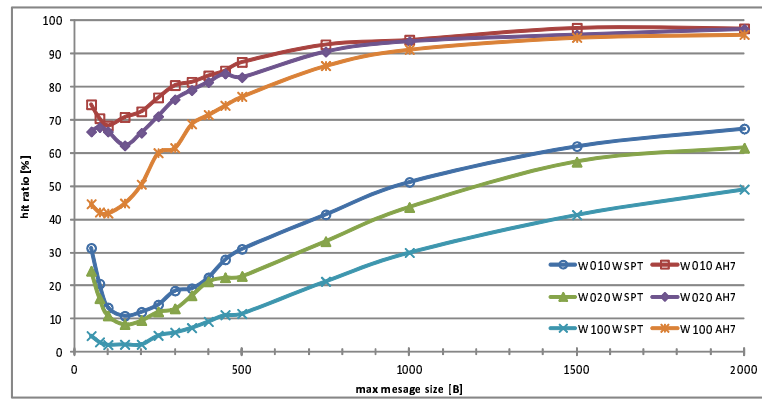


Figure 6. Hit ratios for AH7 and WSPT

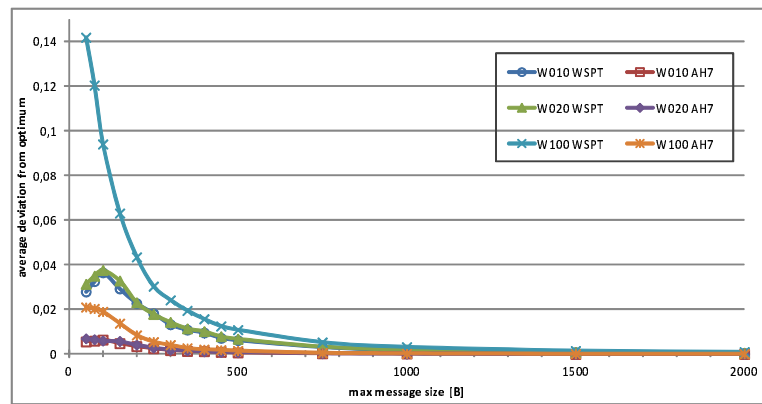


Figure 7. Average deviation from the optimum for algorithms WSPT and AH7

L_{max}	$W_{max} = 10$		$W_{max} = 20$		$W_{max} = 100$	
	WSPT	AH7	WSPT	AH7	WSPT	AH7
50	31.4	74.7	24.5	66.5	4.9	44.7
75	20.6	70.5	16.2	67.8	3.0	42.2
100	13.4	68.3	11.0	66.5	2.2	41.9
150	10.9	70.9	8.4	62.3	2.3	44.9
200	12.1	72.6	9.6	66.1	2.3	50.6
250	14.3	76.8	12.2	71.1	5.0	60.1
300	18.5	80.5	13.1	76.2	5.9	61.6
350	19.2	81.5	17.1	79.0	7.3	68.8
400	22.5	83.4	21.3	81.4	9.2	71.6
450	27.9	84.9	22.5	84.0	11.2	74.4
500	31.1	87.5	22.9	82.9	11.6	77.1
750	41.5	92.8	33.4	90.6	21.3	86.4
1000	51.3	94.2	43.7	93.7	30.0	91.3
1500	62.1	97.8	57.4	95.7	41.4	94.9
2000	67.4	97.6	61.6	97.4	49.1	95.8

Table 3. Results of hit ratio for the experiments for the heuristic algorithms AH7 and WSPT

- 200-250, average message size as compared with the packet length),
- under the influence of network packet division, the WSPT solutions can be as much as 10% less efficient than the optimum solutions (the precise value depends on the weight range),
 - the hit ratio of the proposed algorithm AH7 turned out to be as much as 7 times higher for messages comparable with the packet size (4-100/150/200) than the hit ratio of WSPT in all weight range cases,
 - the algorithm AH7 displayed a much better (smaller) average deviation from optimum than WSPT for messages comparable with the packet size; for instance, for the same data the average deviation from the optimal solution for WSPT was 5-6 times higher than for AH7,
 - for medium size messages, which were larger than the packets, the differences in WSPT and AH7 efficiency were not that distinct,
 - as far as hit ratio and average deviation are concerned, the algorithm AH7 showed lower sensitivity to changes in the weight range than the algorithm WSPT.

To conclude, the use of the AH7 algorithm instead of WSPT makes it possible to substantially reduce losses in total weighted completion time optimization, resulting from the NP-hard complexity of the problem. This is true of messages of size comparable to the packet size. In the case of messages which are bigger than the packet, the number of optima for AH7 is still higher than the number of optima for WSPT. However, the difference of the average deviation from the optimum for these algorithms is not very high. The results determine the

L_{max}	$W_{max} = 10$		$W_{max} = 20$		$W_{max} = 100$	
	WSPT	AH7	WSPT	AH7	WSPT	AH7
50	0.02798	0.00558	0.03143	0.00698	0.14193	0.02103
75	0.03264	0.00598	0.03508	0.00661	0.12050	0.02037
100	0.03655	0.00648	0.03761	0.00602	0.09409	0.01905
150	0.02934	0.00481	0.03290	0.00581	0.06314	0.01381
200	0.02278	0.00347	0.02315	0.00419	0.04350	0.00855
250	0.01821	0.00254	0.01791	0.00286	0.03042	0.00551
300	0.01331	0.00172	0.01433	0.00194	0.02423	0.00412
350	0.01089	0.00137	0.01145	0.00161	0.01949	0.00270
400	0.00961	0.00121	0.01020	0.00122	0.01575	0.00214
450	0.00720	0.00091	0.00791	0.00088	0.01255	0.00186
500	0.00622	0.00070	0.00681	0.00091	0.01088	0.00152
750	0.00312	0.00028	0.00334	0.00033	0.00534	0.00054
1000	0.00179	0.00018	0.00204	0.00013	0.00320	0.00027
1500	0.00088	0.00004	0.00090	0.00007	0.00154	0.00010
2000	0.00054	0.00003	0.00057	0.00003	0.00100	0.00007

Table 4. Values of the average deviation for the experiments involving heuristic algorithm AH7 and WSPT

optimization area, i.e.: asynchronous communication in systems, in which the average message size is comparable with packet size.

8. The real implementations

Minimization of the total weighted completion time is needed in systems for which costs or profits depend on time. For example, a stock exchange system is designed to deliver orders (messages) in such a schedule as to achieve maximum profit. The users of the system use various methods of connection, e.g. modems, dialups and GSM.

This consideration was supported by the experiments, in which there were two computers directly connected by a serial cable (RS 232). Special software, which transmitted a stream of messages over TCP/IP, emulated the considered system. Details about the system and information about the tests are provided in Piórkowski and Werewka (2010). The experiment has proven that the main issue of the article is important for practical reasons.

9. Conclusions and future work

The minimization of total weighted completion time is important for optimizing communication in distributed financial systems, making them more efficient and more fault-tolerant. It was proven that for asynchronous transmission with packetization, no low complexity optimal standard algorithm can be applied,

which is why heuristic algorithms were developed. The proposed algorithms for packet transmission produce 14% efficiency increase compared to the standard algorithm (WSPT).

Future work involves developing optimal and heuristic algorithms, which consider the total weighted completion time ($\sum w_j C_j$) for systems with deadlines.

References

- ADLER, M., KHANNA, S., RAJARAMAN, R., ROSEN, A. (1999) Time-constrained scheduling of weighted packets on trees and meshes. In: *Proceedings of the eleventh annual ACM symposium on Parallel algorithms and architectures*. ACM, 1–12.
- ADLER, M., SITARAMAN, R.K., ROSENBERG, A.L., UNGER, W. (1998) Scheduling time-constrained communication in linear networks. In: *Proceedings of the tenth annual ACM symposium on Parallel algorithms and architectures*. ACM, 269–278.
- ATENCIA, I. (2014) A discrete-time system with service control and repairs. *International Journal of Applied Mathematics and Computer Science* **24**(3), 471–484.
- BANSAL, N., HARCHOL-BALTER, M. (2001) Analysis of srpt scheduling: Investigating unfairness. *SIGMETRICS Perform. Eval. Rev.* **29** (1), 279–290, <http://doi.acm.org/10.1145/384268.378792>
- CORMEN, T.H., LEISERSON, C.E., RIVEST, R.L., STEIN, C., et al. (2001) *Introduction to Algorithms*, 2. MIT Press, Cambridge.
- COULOURIS, G.F., DOLLIMORE, J., KINDBERG, T. (2005) *Distributed Systems: Concepts and Design*. Pearson Education.
- DIJKSTRA, E.W. (2002) *Cooperating Sequential Processes*. Springer
- DOBRIN, R., FOHLER, G. (2001) Implementing off-line message scheduling on controller area network (can). In: *Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference on*. IEEE, 241–245.
- FLIEDER, K. (2005) Testing and visualizing a message queuing infrastructure. In: N. Callaos, W. Lesso. eds., *Proc. of WMSCI '05*, Orlando, Florida, 51–57.
- GAJER, M. (2010) Task scheduling in real-time computer systems with the use of an evolutionary computations technique. *Przełqd Elektrotechniczny* **86**(10), 293–298
- GAWLICK, D. (2002) Message queuing for business integration. *eAi Journal* (October 2002)
- HARCHOL-BALTER, M., BANSAL, N., SCHROEDER, B. (2000) Implementation of srpt scheduling in web servers. Technical Report cmu-cs-00-170. Carnegie Mellon School of Computer Science
- HUANG, Y. (2007) *Scalable Web Service-based XML Message Brokering Across Organizations*. ProQuest

- JÚDICE, J.J., FAUSTINO, A.M., RIBEIRO, I.M. (2002) On the solution of np-hard linear complementarity problems. *Top* **10**(1), 125–145
- KIELMANN, T., HOFMAN, R.F., BAL, H.E., PLAAT, A., BHOEDJANG, R.A. (1999) MPI's reduction operations in clustered wide area systems. *ACM Sigplan Notices* **34** (8), 173–182.
- NAWROCKI, J., COMPLAK, W., BŁAŻEWICZ, J., KOPCZYŃSKA, S., MAĆKOWIAK, M. (2009) The knapsack-lightening problem and its application to scheduling hrt tasks. *Bulletin of the Polish Academy of Sciences: Technical Sciences* **57**(1), 71–77
- NG, C.T., MÉDARD, M., OZDAGLAR, A. (2009) Completion time minimization and robust power control in wireless packet networks. In: *Communications, 2009. ICC09. IEEE International Conference on*. IEEE, 1–6.
- PIÓRKOWSKI, A. (2003) Heurystyczne algorytmy optymalizacji kosztu całkowitego w systemach wiadomości kolejkowanych (Heuristic algorithms to optimize total cost in the systems of queued messages; in Polish). *Automatyka*. Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie 7, 221–227
- PIÓRKOWSKI, A., WEREWKA, J. (2010) Minimization of the total completion time for asynchronous transmission in a packet data-transmission system. *International Journal of Applied Mathematics and Computer Science* **20**(2), 391–400
- RAMANATHAN, P., RUPNICK, G.M. (1991) Deadline constrained message scheduling in point-to-point interconnection. In: *Proc System Design Synthesis Technology Workshop*, Silver Spring, MD, Naval Surface Warfare Center, 183–192.
- RAMESH, S., PERROS, H.G. (2001) A multi-layer client-server queueing network model with non-hierarchical synchronous and asynchronous messages. *Performance Evaluation* **45**(4), 223–256
- SHAFER, K., AHUJA, M. (1992) Process-channel agent-process model of asynchronous distributed communication. In: *Distributed Computing Systems, 1992., Proceedings of the 12th International Conference on*. IEEE, 4–11.
- SMITH, W.E. (1956) Various optimizers for single-stage production. *Naval Research Logistics Quarterly* **3**(1-2), 59–66
- TSAI, B.R., SHIN, K.G. (1996) Combined routing and scheduling of concurrent communication traffic in hypercube multicomputers. In: *Distributed Computing Systems, 1996, Proceedings of the 16th International Conference on*. IEEE, 150–157.
- WOZNIAK, M., KEMPA, W.M., GABRYEL, M., NOWICKI, R.K. (2014) A finite-buffer queue with single vacation policy – analytical study with evolutionary positioning. *International Journal of Applied Mathematics and Computer Science* **24**(4), 887–900
- YANG, B., LIU, D.Y., YANG, K. (2002) Communication performance optimization for mobile agent system. In: *Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on*. IEEE 1, 327–335.
- YANG, J., ULUKUS, S. (2010) Transmission completion time minimization

in an energy harvesting system. In: *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*. IEEE, 1–6.

ZHU, X., YU, J., DOYLE, J.(2001) Heavy tails, generalized coding, and optimal web layout. In: *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*. IEEE. 3, 1617–1626.