ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2010, Vol. 39, No. 1

SIMULATION AND TESTING OF FIFO CLEARING ALGORITHMS

Donatas Bakšys, Leonidas Sakalauskas

Institute of Mathematics and Informatics
Akademijos Str. 4, LT-08663 Vilnius Lithuania
e-mail:sakal@ktl.mii.lt

Abstract. The clearing systems of interbank settlements guarantee the movement and repartition of assets in the market. The increase in demand of real-time money movement induces new requirements to settlement systems. However, the interbank payment and settlement sector is very sensitive to changes in the market. This calls a demand to foresee adaptation of the payment and settlement system in a dynamic environment. Since FIFO algorithms are often applied by the operators of settlement systems to meet the requirements on settlement transfering, the subject of study of FIFO algorithms is topical in interbank systems. The objective of this paper is to investigate and survey the clearing algorithms in settlement processes, analyzing the most popular FIFO settlement algorithms. The results of simulation study of FIFO settlement algorithms are given, based on the guidelines of settlement system TARGET2.

Keywords: Interbank payments, settlements, modelling of interbank settlements, settlement algorithms, FIFO algorithms.

1. Introduction

The sector of settlement is among institutions requiring of simulation environment. This sector, due to its sensitivity to changes, is not fit for empirical researches of the real environment. The efficiency of clearing systems applied in this sector strongly depends on the choice of its structure, processing algorithms and system parameters. A variety of the latter factors and a possibility to use their different combinations make this subject of investigation topical both in theory and in practice. The simulation environment enables us to study the clearing algorithms (CAs), which are of topical importance in the settlement processes.

The transfers in the accounts of settlement system participants are booked on the basis of the CA. The efficiency of a settlement process depends on the architecture of the settlement system and usage of CA. Typically, the central bank (CB) fulfils the function of the Clearing house (CH) and accomplishes the monetary policy of the country economy [1]. Taking this into account, CBs are interested in the efficiency of the settlement process. For the mentioned purposes, the CBs are active developers of simulation environments and testers of the CA on the basis of simulation environments. The Bank of Finland (BoF-PSS1, BoF-PSS2 – The Bank of Finland Payment and Settlement Simulator), The Bank of France (PNS – Paris Net Settlement large-value payment system), the Bank of Sweden (RIX system), the Bank of Austria (ARTIS – the Austrian Real-Time Interbank Settlement system) and the Bank of England intensively work in this area. Settlements simulators are investigated and compared by Leinonen and Soramaki [13], Mazars and Woelfel [14], Schmitz and Puhr [15], Shafransky and Doudkin [16], Bakšys and Sakalauskas [2].

The objective of the article is to investigate and survey CAs in the settlement processes analyzing, by simulation environment, the most popular settlement algorithm FIFO (First In First Out) implemented according to the guidelines of settlement system TARGET2. The study is performed on the base of the data taken from the payment and settlement system of the Bank of Lithuania.

2. The group of clearing algorithms

The operators of a settlement system use different settlement algorithms to ensure an efficient money transfer process. The algorithms realized at different stages of a settlement process are divided into the following basic groups [18]:

- submission algorithms (SUB);
- entry algorithms (ENT);
- settlement algorithms (SET);
- end-of-day algorithms (END).

The task of the submission algorithm is to determine which transaction is next to being processed from all the pending transactions in all systems. It can be understood as a process in which the bank distributes the received transactions for the system submission process. The other algorithms are specified at the system level. The entry algorithms are used to perform the initial processing of each transaction [3]. ENT algorithms are divided into injection algorithms (INJ) and splitting algorithms (SPL). The SPL algorithms split a large transaction into sub-transactions according to specific rules. The INJ algorithms transfer liquidity between the subsidiary and basic systems. The Queue release (QUE) algorithms, SPL, INJ, Bilateral off-setting (BOS) algorithms, partial netting algorithms (PNS) and multilateral netting algorithms (MNS) are used with SET algorithms. The QUE algorithms check and fetch in the transactions from the waiting queue in the given order once an account or participant has received more liquidity attempts to settle all the queued transactions in one netting event. The BOS algorithms check and fetch the transactions from the waiting queues that can be bilaterally off-set. The PNS algorithms seek to settle a part of the queued transactions. The MNS algorithm attempts to settle all the queued transactions in one netting event. The END algorithms process the final steps during a day or settlement cycle.

3. The simulation environment for testing the clearing algorithms

The procedures of modelling and simulation of settlement systems are realized to study the processing of a real settlement system. The simulation environment developed consists of the following parts [13]:

- The subsystem of statistical analysis of settlement data;
- The subsystem of simulation of a settlement process.

During the statistical analysis, the data of a real system are read out and analyzed estimating statistical characteristics of the settlement flow. The estimated statistical characteristics are used to calculate the parameters of a settlement model.

The settlement simulation subsystem itself consists of the following basic parts [13]:

- The procedure for generation of transactions flow;
- The procedure of simulation of a day settlement process;
- The procedure of calculation of liquidity positions;
- The procedure of statistical simulation of a settlement system.

Transaction flows are realized generating the transaction values and the time moments of transaction submission. The procedures of evaluation of liquidity positions are calculation of the statistical characteristics of the system and participants, and analysis of the various strategies of management of the settlement process as well.

An intensive development of the settlement system modelling and simulation started in the 1990s [13]. The first researches were oriented at the study of new settlement procedures [5]. The systems of settlement simulation were developed on the basis of the Bank of Finland Payment and Settlement system, and the Bank of England Clearing House Automated Banking System CHAPS [10].

Real, statistical, and adapted data of settlement systems are used in the settlement simulation systems. The real data are used for simulating the "what if" types of scenario, and parameters of the real settlement system are predicted. The statistical data are used for simulating processes of a system with undisclosed data or on purpose to compare the real data and statistical. The BoF-PSS2 simulator is a tool for making a variety of analyses of the payment system [13]. The simulator is not a deterministic econometric optimization model, but rather a heuristic tool for analyzing systems that are too complex for deterministic models. In the simulation process, only a partial processing of the settlement system is simulated and a few selected criteria are analyzed (i.e. liquidity, queuing et al.) because the full simulation is complicated. The basic principle is that the given payment flows are processed in a given model of the existing or contemplated payment and settlement system structure. The simulator supports the real-time gross settlement (RTGS), continuous net settlement (CNS), and deferred net settlement (DNS) systems. The processing options for these systems are defined by selecting appropriate algorithms.

4. The FIFO algorithms

FIFO is the most institutionalized processing order in payment systems [7]. FIFO is applied to release payments from queues when the participants of the settlement system lack the liquidity. Since transactions have different levels of urgency the more important transactions bypass the FIFO order. Other queuing orders are also possible. The payments queue can be released starting from small transactions [17]. In some systems, participants may also reorder the transaction queues according to their priority (e.g., the CHAPS system) [10].

The FIFO principle is easy to realize. A variation of the FIFO rule is the 'Bypass FIFO' rule [19]. Taking into account liquidity constraints, FIFO works reasonably well when smaller payments are submitted generally earlier in the day. In this case, an earlier entry into the queue has a priority over later entries except that, if the paying bank has a lack of liquidity, an attempt is made to settle the next payment [6].

Instead of applying the FIFO rule, an algorithm can be used that maximizes either the number or value of payments that get settled with the available liquidity [8]. This problem can be compared to the well-known 'knapsack' problem, i.e. the situation where items, each having a cost and a value, are included in a collection so that the total value is maximized subject to the total cost being less than a specified amount. If such an algorithm is applied in settlement queueing, the liquidity usage is optimized by selecting payments so as to maximize the total value of payments settled subject to the requirement that the total settled value is less than the amount of liquidity available to the paying bank. A disadvantage of this type of queueing arrangement is that some payments may remain unsettled in the queue for a long time while the FIFO rule is not in effect [9].

Splitting of transactions allow more efficient usage of the available liquidity [12]. This can be done using two main conventions: by defining a maximum transaction size according to which larger transactions are split or by using the available liquidity in full to create a part of the current transaction that could be settled.

Settlement systems also exhibit hoarding behaviour. Participants may delay transactions to reduce their own liquidity needs, which in turn can cause congestion at the end of the day, if other participants also delay their transactions [11]. To control fair reciprocity, multilateral or bilateral sending limits can be used. If bilateral limits are used, a participant will only release new payments to counterparties that have released the anticipated flow of transactions [4].

5. FIFO simulation algorithms

In this section, simulation algorithms are described. The simulation is executed using the system of modelling, simulation, and optimization of settlements described in [2]. According to the transaction model used, the system generates flows of moments of bilateral payments by the Poisson distribution and the corresponding flow of payment volumes according to the lognormal distribution. The parameters of the Poisson-lognormal model are estimated according to the real data.

Denote the number of banks that are participants of the settlement system by J. For $i, j = 1,...,J$, let z_{ij} be the number of payments from bank i to bank j. Clearly, $i \neq j$ in all the cases where we are considering a pair of banks i and j, so further we will not mention it. Denote by p_{ij}^k the sum of the k th payment from bank i to bank j, $k \in \{1,...,z_{ij}\}$. Let us introduce the variable $C_{ij}^k \in \{0,1\}$, $i, j = 1,...,n$, $k = 1,...,z_{ij}$. Hence, $C_{ij}^k = 1$ denotes that the k th payment from bank i to bank j is included into the set of settled payments. Respectively, $C_{ij}^k = 0$ means that the payment is not included into the set.

The settlement balance δ_i is computed as follows:

$$\delta_i = \sum_{j=1}^{J} \left(\sum_{k=1}^{z_{ij}} C_{ij}^k \cdot p_{ij}^k - \sum_{k=1}^{z_{ij}} C_{ij}^k \cdot p_{ji}^k \right), \qquad (1)$$

The simulation procedure has been developed, which realizes the following algorithms, used in the settlement system TARGET2 [13]:

- the transfers generation algorithm,
- the basic FIFO settlements algorithm,
- the bypass FIFO settlement algorithms,
- the day balance computation algorithm.

***Algorithm 1*: Transfers generation algorithm**

Purpose: to generate transfer flows.

Preconditions: mean and standard deviation of transactions logarithms: μ, σ, P=$[P_{ij}]_1^J$ is the matrix of intensities of settlement flows from sender i to receiver j, (the number of applications / minutes), $P_i = \Sigma P_{ij}$, J is the number of agents, n is the number of transfers to be settled, T is the number of intervals of transactions generation.

Post-conditions: W is the vector of transaction values, v is the number of transactions to be sent, r is the number of transactions to be received, i is the sender of transaction, j is the receiver of transaction, s is the time of transaction, C is the vector of indicators of transfers.

The flow of transactions is generated:

```
s=0;
i=0;
j=0;
n=0;
while s<=T
       tmp= U(0,1);
       for l=0 to J do
       if Pl>tmp i=l;
       vn=i;
       s= s -ln(U(0,1)) · λ;
       ttn = s;
       Cn = 1;
       j=i;
done
while j == i
       tmp= U(0,1);
       for l=0 to J do
       if Pij >tmp j=l;
       rn=j;
       Wn=exp(Gaus · σvn+ μvn);
       n = n+1;
done
```

Algorithm 2: Basic FIFO settlement algorithm

Purpose: to settle transfers according to FIFO algorithm.

Preconditions: *J* is the number of agents, *v* is the number of transactions, *K* is the vector of position of the correspondent account of participants before the session, *W* is vector of transaction values, *i* is sender of transaction, *j* is receiver of transaction.

Postconditions: *K* is the vector of position of the correspondent account of participants after the session, *C* is the vector of indicators of transfers (1 if the transfer is fulfilled, 0 if the transfer is delayed), *B* is the vector of indicators of queued transactions (1 if the transfer is queued), *w* is the vector of values of queued transactions, *e* is the volume of fulfilled transactions, *L* is the limit of total volume of transactions.

The transfer settlement procedure is as follows:

```
while e>0
        for i=0 to J do
        B_i=0;
        w_i=0;
        L_i=100;
        e=0;
        for i=0 to n do
        if C_i==1 and B_v^i==0
        if K_v^i > W_i and w_v^i < L_v^i
        K_v^i = K_v^i – W_i;
        w_v^i = w_v^i + W;_i
        K_r^i = K_r^i + W_i;
        C_i = 0;
        e=e+1;
        else B_v^i =1;
done
```

Algorithm 3: Bypass FIFO settlement algorithm

Purpose: to settle transfers according to the first bypass FIFO algorithm (to delay the largest value of transactions).

Preconditions: *J* is the number of agents, *v* is the number of transactions, *K* is the vector of positions of participants correspondent account before the session, *W* is vector of transaction values, *i* is the sender of transaction, *j* is the receiver of transaction.

Postconditions: *K* is the vector of positions of correspondent account of participants after the session, K^* is the vector of positions of participants correspondent temporary account after the session, *C* is the vector of indicators of transfers (1 if the transfer is fulfilled, 0 if the transfer is delayed), *W* is vector of transaction values, *e* is the volume of fulfilled transactions, *u* is the value of unfulfilled transactions, *o* is the volume of unfulfilled transactions.

The transfer settlement procedure is as follows:

```
for j=0 to J do
        C_j=1;
        z1 = 0;
        o=0;
        while z1==0
        for j=0 to J do
        K_j^* =0
        for i=0 to n do
        K^*_v^i=K^*_v^i +W_i ·C_i;
        K^*_r^i=K^*_r^i-W_i · C_i;
        done
        z = 0;
        i=-1;
        for j=0 to J do
        if K_j^* -K_j >z
        z=K_j^* - K_j
        i=j
        done
        x=0;
        y = 0;
        if z>0
        for j=0 to n do
        if v_j == i and W_j > y and C_j ==1
        y =W_j
        x =j;
        C_x =0
        else  z1=1
        done
        for i=0 to n do
        K_v^i =K_v^i –C_i · W_i
        K_r^i =K_r^i +C_i · W_i
        o=o+1-C_i
        done
        for i=0 to n do
        u=u+C_i · W_i
        e=e+C_i
done
```

Algorithm 4: Bypass FIFO settlement algorithm

Purpose: to settle transfers according to the second bypass FIFO algorithm (to delay the last transaction of the participants providing largest loss of liquidity account).

Preconditions: *J* is the number of agents, *v* is the number of transactions, *K* is the vector of position of correspondent account of participants before the session, *W* is vector of transaction values, *i* is sender of transaction, *j* is receiver of transaction.

Postconditions: *K* is the vector of positions of participants correspondent account after the session, K^* is the vector of positions of participants correspondent temporary account after the session, *C* is the vector of indicators of transfers (1 – if the transfer is fulfilled, 0 – if the transfer is delayed), *e* is the volume of fulfilled transactions.

The transfer settlement procedure is as follows:

```
for j=0 to J do
        Cj=1;
        z1 = 0;
        o=0;
        while z1==0
        for j=0 to J do
        Kj* =0
        for i=0 to n do
        K*v^i=K*v^i +Wi ·Ci;
        K*r^i=K*r^i-Wi · Ci;
        done
        z = 0;
        i=-1;
        for j=0 to J do
        if Kj* -Kj >z
        z=Kj* - Kj
        i=j
        done

        x=0;
        y = 0;
        if z>0
        for j=n to 0 do
        if vj == i and Cj ==1 and x==-1) {
        y =Wj
        x =j;
        Cx =0
        else  z1=1

        for i=0 to n do
        Kv^i =Kv^i –Ci · Wi
        Kr^i =Kr^i +Ci · Wi
        o=o+1-Ci
        done

        for i=0 to n do
        u=u+Ci · Wi
        e=e+Ci
done
```

***Algorithm 5*: The day balance statistics computation algorithm**

Purpose: to calculate the daily statistics of settlements.

Preconditions: J is the number of agents, v is the number of transactions, K is the vector of position of participants correspondent account before the session, C is the vector of indicators of transfers (1 if the transfer is fulfilled, 0 if the transfer is delayed), A is the vector of deposited values of participants, P is the vector of participant cost, R is the interbank credit interest rates, F is the transaction fulfilling fee, R^* is the short term interest rates.

Postconditions: h is the vector of participant debts, SS is the total value of transactions to be sent, SG is the total value of supply transactions, SR is the total value of receive transactions.

```
for i=0 to J do
        Ki =Ai
        Pi =Ki ·R/360;
done

for i=0 to J do
        skoli[j]=0;
        for i=0 to n do
        hv^i=hv^i+Ci ·Wi
        for i=0 to n do
        Pv^i= Pv^i +F ·Ci
        for j=0 to n do
        SSv^i = SSv^i+(1-Ci) ·Wi
        SG v^i= SGv^i+ Wi
        SRr^i = SRr^i+ Wi
        done
        for i=0 to J do
        if hi>0
        Pi = Pi +(hi –Ki) · R*/360
done
```

The algorithms have been realized with real parameters of the settlement system participants. The algorithms are actualized as the Java class library and were tested by the Java/J2EE modular, standards-based, integrated development environment NetBeans IDE 6.0.1.

6. The results of simulation

During the simulation, the influence of the value deposited in the correspondent account on the coefficient of settlements as well as on dynamics of correspondent account has been explored. The fulfillment coefficient indicates the level of performed transactions.

The algorithms were tested using the following real data of the settlement system: $J = 11$, $\mu = 7.813$, $\sigma = 2.189$, $T = 100$.

In Figures 1 – 3, the dependences of the settlement costs and the fulfillment coefficient on the value of the correspondent account are presented using different FIFO algorithms. In these figures, the impact of deposited value on the correspondent account is observed on the transaction fulfillment. The figures show, that an increase in the deposited value in the correspondent account influences the fulfillment of transactions as well as that an increase of this value conditions the increase of liquidity. Dynamics of the fulfillment coefficient shows the existence of the maximal volume of the value deposited on the correspondent account, which guarantees the full liquidity of the participant.

The testing results show that the Bypass FIFO settlement algorithm (to delay the last transaction of the participants with the largest loss of liquidity account) is most effective in comparison to the other algorithms tested.

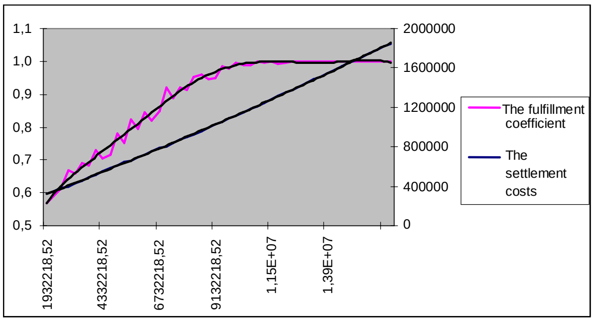

Figure 1. Dependence of the settlement costs and the fulfillment coefficient on the value deposited on the bank correspondent account using the Basic FIFO algorithm

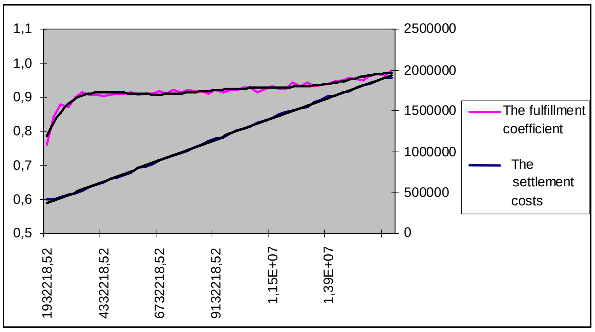

Figure 2. Dependence of the settlement costs and the fulfillment coefficient on the value deposited on the bank correspondent account using the Bypass FIFO algorithm (to delay the largest value of transaction)

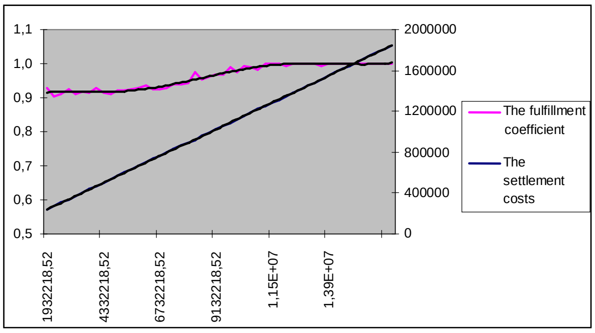

Figure 3. Dependence of the settlement costs and the fulfillment coefficient on the value deposited on the bank correspondent account using the Bypass FIFO algorithm (to delay the last transaction of the participants providing largest loss of liquidity account)

7. Conclusion

The exploration of the settlement algorithms by using the system, developed for modelling and simulation of settlements, allows us to estimate the efficiency of algorithms and to propose the recommendations for executing of settlement procedures. The most popular settlement algorithm FIFO has been investigated by simulation environment developed according to the guidelines of settlement system TARGET2. The study has been performed on the base of the data of the payment and settlement system of the Bank of Lithuania.

Using three FIFO algorithms, realized according to the guidelines of settlement system TARGET2, the increase of liquidity has been established by the increase on the value deposited on the bank correspondent account. The dynamic of the fulfillment coefficient shows the existence of a maximal volume of the value deposited in correspondent account, which guarantees the liquidity of a participant with an admissible fulfillment coefficient.

The test results have shown that the Bypass FIFO settlement algorithm (to delay the last transaction of participants providing a largest loss of liquidity account) is most effective in comparison with the other algorithms tested.

The simulation basically depends on the balance of the payment intensity matrix. If at least the averages of income and outcome payments flow are different for one participant, $\sum_{j=1}^{J}\mu_{ij} \neq \sum_{j=1}^{J}\mu_{ji}$, then the matrix is unbalanced. In the system with an unbalanced matrix, the participants have different liquidity positions. Thus, the positions of participants in the bank correspondent account become positive in one cluster, and the participants in the next cluster require of liquidity necessarily at the end of the settlement day. Therefore additional requirements should be applied to the participants having scarcity of liquidity. The value of requirement will be chosen in view of the liquidity position at the end of the settlement period. The long-term negative liquidity position shows that the participant has outside incoming assets or is in the pre-bankrupt situation. The results of simulation have shown that there exists an optimal value of the correspondent account, which ensures the admissible fulfillment of all transactions.

Acknowledgement

The research is partially supported by the Lithuanian State Science and Studies Foundation Project No T-33/08 "Development and research of system for simulation, modelling and optimization of settlement system".

References

[1] **P. Angelini.** An Analysis of Competitive Externalities in Gross Settlement Systems. *Journal of Banking and Finance, Vol.* 22, 1998, 1–18.

[2] **D. Bakšys, L. Sakalauskas**. Modelling, simulation and optimisation of Interbank Settlements. *Information Technology and Control, Vol.* 36, *No.* 1, 2007, 43-52.

[3] **M.L. Bech, R. Garratt.** The Intraday Liquidity Management Game. *Journal of Economic Theory, Vol.* 109, 2003, 198–219.

[4] **A.N. Berger, D. Hancock, J.C. Marquardt.** A Framework for Analyzing Efficiency, Risks, Costs, and Innovations in the Payment System. *Journal of Money, Credit, and Banking, Vol.* 28, *No.* 4, 1996, 696-732.

[5] **M.J. Flannery.** Financial Crisis, Payment System Problems, and Discount Window Lending. *Journal of Money, Credit, and Banking*, *Vol*. 28, *No*. 4, 1996, 804-824.

[6] **X. Freixas, B. Parigi.** Contagion and Efficiency in Gross and Net Interbank Payment Systems. *Journal of Financial Intermediation*, *Vol*. 7, *No*. 1, 1998, 3–31.

[7] **M.M. Güntzer, D. Jungnickel, M. Leclerc.** Efficient Algorithms for the Clearing of Interbank Payments. *European Journal of Operational Research*, *Vol*. 106, *No*. 1, 1998, 212–219.

[8] **D. Humphrey.** Comment on Intraday Bank Reserve Management: The Effects of Caps and Fees on Daylight Overdrafts. *Journal of Money, Credit and Banking*, *Vol*. 28, *No*. 4, 1996, 909–913.

[9] **J.M. Lacker.** Clearing, settlement, and monetary policy. *Research Department Federal Reserve Bank of Richmond*, *Richmond*, 1997.

[10] **K. James, M. Willison.** Collateral posting decisions in CHAPS Sterling. *Bank of England. Financial Stability Review*, *December* 2004.

[11] **C.M. Kahn, W. Roberds.** Payment System Settlement and Bank Incentives. *Review of Financial Studies*, *Vol*. 11, *No*. 4, 1998, 845–870.

[12] **R. Koponen, K. Soramaki.** Intraday Liquidity Needs in a Modern Interbank Payment System. *Bank of Finland, Studies* E:14, 1998.

[13] **H. Leinonen, K. Soramaki.** Simulating interbank payment and securities settlement mechanisms with the BoF-PSS2 simulator. *Bank of Finland, Research Discussion Paper* 23, 2003.

[14] **E. Mazars, G. Woelfel.** Analysis, by simulation, of the impact of a technical default of a payment system participant. *Banque de France. Financial Review Stability No*. 6, *June* 2005.

[15] **W.S. Schmitz, C. Puhr.** Liquidity, Risk Concentration and Network Structure in the Austrian Large Value Payment System. *Oesterreichische Nationalbank* (OeNB), *Viena*, 2006.

[16] **M.Y. Shafransky, A.A. Doudkin.** An optimization algorithm for the clearing of interbank payments. *European Journal of Operational Research, Vol*. 171, *No*. 3, 2006, 743–749.

[17] **D. Schoenmaker.** A Comparison of Alternative Interbank Settlement Systems. *Discussion Paper* 204, *Financial Markets Group, London School of Economics*, 1995.

[18] **K. Soramaki, M.L. Bech, J. Arnold, R.J. Glass, W. Beyeler.** The Topology of Interbank Payment Flows. *Federal Reserve Bank of New York, Staff Reports*, *No*. 243, *March* 2006.

[19] **C. Vital.** An Appraisal of the Swiss Interbank Clearing System SIC. *Presentation at the IBC Conference on International Payment Systems, London*, 1994.

Received April 2009.

The research is partially supported by the Lithuanian State Science and Studies Foundation Project No T-33/08 "Development and research of system for simulation, modelling and optimization of settlement system".