Business Process Modeling and the Robust PNS Problem

József Tick

Óbuda University, Bécsi út 96/b, H-1034 Budapest, Hungary
e-mail: tick@uni-obuda.hu

Csanád Imreh, Zoltán Kovács

University of Szeged, Árpád tér 2, H-6720 Szeged, Hungary
e-mail: cimreh@inf.u-szeged.hu; kovacsz@inf.u-szeged.hu

Abstract: In this paper we define and investigate a new direction of the P-graph-based Business Process Modeling which we call the robust PNS problem. We consider the model where for each operating unit two costs are given, it has a nominal cost and an extended cost, and we know that at most b operating units have the extended cost, the others will have the nominal cost. We present a branch and bound based exact solution algorithm for the general problem, and a faster, polynomial time dynamic programming algorithm for the case of the hierarchycal problems.

Keywords: Business Process Modeling; PNS problem; robust problems; P-graphs

1 Introduction

In the last decades one of the most significant changes was that informatics spread in all the fields of business processes, and has become an unavoidable component. By now business processes have become more and more complex. Their regular operations need the smooth cooperation of several systems. The solution of the cost effective and error free management of complex business processes as well as the efficiency of the solution is one of the most important issues for the profit-oriented sector, since the optimal efficiency of the complex business, production and office-automation processes is a high priority economic value.

In favor of the optimal operation modeling the business processes is necessary; furthermore, the model-based approach is necessary for the development and management of such systems.

There are more well-known approaches for the modeling of business processes. Workflow is the most widespread modeling technique in the field of industrial and office information systems. The workflow-based models can be used excellently for the analysis, modeling and optimizing of work flows in the business processes.

Several workflow representations have spread, out of which, however, the P-graph-based workflow [17] is the one that is a process-based method with correct mathematical background and gives definitely an optimal workflow network structure on a systematic way. The introduction of the p-graph in workflows has been done similarly to the modeling of process networks [18], that is why the analysis of PNS methodologies and the elaboration of further procedures are necessary.

In a manufacturing system, materials of different properties are changed by different transformation to yield desired products. Usually some raw materials and the desired products are given and our goal is to produce the desired products from the raw materials though the possible transformations. These sytems can be modelled in the P-graph framework where a bipartite graph is used. One of the sets of the vertices contains the possible materials. The other set contains the possible transformations represented as operating units defined by their input and output material sets. Some subgraphs of the graph containing all the possible transformations describe the feasible processes which produce the desired products from the raw materials and the goal is to find the cheapest such subgraph. In the combinatorial version studied in this paper each operating unit has a fixed cost and the cost of a subgraph is the sum of the costs of the operating units contained in it. In a more general, quantitive model we also consider the amounts of the used materials and the cost of the operating units depends on the amount of materials used by them. One can find the detailed backround of the PNS model in [6], [7], therefore we will recall only the main definitons in the next section.

The P-graph framework was designed to analyse and solve process network synthesis problems but later it was observed that it can be used to solve optimization problems in other areas as well. In [5] and [15] one can find how to use the P-graphs in supply chain management. Another big area where the P-graphs are useful is workflow management.

The [18], [19] and [20] present the P-graph-based modeling of workflows introduced to the analogy of process-network modeling as well as introduce the extension of the model by the time factor as special resource. It generates the relevant mathematical model from the PNS structure of the defined workflow and conducts analyses in order to define the objective function (capacity constraints, bottle necks) determined by the environment.

The [16] and [21] present the possible fuzzy extension of the P-graph-based workflow model, analyses the „real life like", uncertain and not-exactly definable situations present in the real application environment of the workflow, which for better modeling makes the fuzzy extension necessary. It introduces the fuzzy sets defined for documents and activities and presents the application of the method with an example. It introduces the parametric t-norm, which can be considered as the extension and generalization of the Zadeh and the Fodor t-norms indeed and with its application the tuning of the fuzzy regulatory systems might become more effective.

In these optimizaton problems we suppose that all costs are known exactly in advance. On the other hand, in real applications usually some uncertainty can change the data. In the PNS model these uncertainties can be handled in the case of workflow problems by fuzzy methods (see [16] for details) or at the supply chain problems by extending the P-graph with the ROA (reliability of availability) value at the materials (see [15] for details). In general, for most optimization model the problem of uncertainty is solved by stochastic optimization. On the other hand, in these cases we need some a priori information about the distribution of the data, which is usually not available in real applications. Another approach is a robust optimization, where the uncertainty is handled by deterministic worst case scenario. In these models we do not have the fixed values of the parameters we only know that they are in a given interval. There are many robust models of combinatorial problems one can find an overview in [3]. In this paper we consider the robust version where we have an a priori bound on the number of the parameters which might be changed, such models are studied in [4] and [14]. This means that we consider the model where for each operating unit two costs are given, it has a nominal cost and an extended cost, and we know that at most b operating units have the extended cost, the others will have the nominal cost. We will search for the optimal solution for this objective functions.

2 The Mathematical Backround

2.1 The Basic Definitions of the PNS Model

The structure of the PNS problem can be studied by the P-graphs defined in [7]. To define this graph let M be the finite nonempty set of the possible materials. The are two distinguished subsets of the materials, R denotes the set of raw materials, and P denotes the set of the desired products. The possible transformations are modeled as operating units, each of them is determined by two sets of materials, i.e., the set of input and output materials of the operating unit. For an operating unit u we denote the set of input materials by in(u), the set of output materials by out(u), we will use the same notation for the sets of operating units. We denote the set of all possible operating units by O. Then the process graph or P-graph in short is defined by this pair (M, O). The set of vertices of this directed graph is $M \cup O$, and the set of edges consist of the following subsets

1) the edges which go to an operating unit from its input materials

2) the edges which go from an operating unit to its output materials.

Then some subgraphs of the P-graph represent the feasible solutions which are able to produce the required materials from the set of raw materials. In [7] it is shown that a subgraph (m,o), where m and o are the subsets of M and O, represents a feasible solution if and only if it satisfies the properties listed below.

(A1) P is a subset of m,

(A2) a material from m is a raw material if and only if no edge goes into it in the P-graph (m, o),

(A3) for every operating unit of o there exists a path in the P-graph (m,o) which goes from the unit into a desired product,

(A4) all of the materials in m are either input or output materials of some operating units from set o.

For an arbitrary material m we denote by $\Delta(m)$ the set of operating units which produce the material. We extend this notation to sets as well, $\Delta(S)$ denotes the set of operating units which produces some elements from set S.

In the combinatorial optimization version of the standard PNS problem each operating unit o has a cost c(o) and the goal is to find the feasible solution where the total cost of the operating units contained in it is minimal. There are some branch and bound algorithms for the solution of this optimization problem see [10] and [11] for details. These branch and bound algorithms are based on the notion of decision mappings which are defined in [8]. We will use these decision mappings later, thus we recall this definition here.

A decision mapping assigns to each material m a subset of $\Delta(m)$ denoted by $\delta(m)$ and this shows which operating units produce the material in the feasible solution considered. On the other hand, not an arbitrary set of decision mappings can belong to a solution. If an operating unit with output set containing a material X is selected to produce in a solution a material Y, then it must be also selected to produce X as well. These contradictions are eliminated in the consistent decision mappings where it is valid for any pair of materials X, Y that $\delta(X)\cap\Delta(Y)$ is a subset of $\delta(Y)$. The consistent decision mappings are important since each feasible solution can be described by these decision mappings of the materials.

2.2 The Robust Version

In the robust model each operating unit o_i has an extended cost $c(o_i)+ e(o_i)$. We will call $c(o_i)$ the nominal cost and $e(o_i)$ the extra cost of the operating units. For any set Q of operating units we will use c(Q) to denote the sum of the nominal costs in set Q. Furthermore, we have an a priori bound b, which means that b operating units can have the extended cost and the others have the nominal cost. We are interested in the worst case, therefore, if we consider a feasible solution of the problem in the robust version its cost will be the sum of the nominal costs of the operating units plus the sum of the b largest extra costs. In the robust model we can have completely different optimal solutions than in the case of nominal costs, as the following example shows.

Example 1. Suppose we have a problem where R_1 is the raw material, P_1 is the desired product and we have one further material denoted by $X_{1.}$ There are three operating units: U_1 produces directly P_1 from R_1 and $c(u_1)=5$ $e(u_i)=2$, U_2 produces X_1 from R_1 and $c(u_2)=2$ $e(u_2)=2$, U_3 produces P_1 from X_1 and $c(u_3)=2$ $e(u_3)=2$. If we consider the standard problem then the optimal solution contains U_2 and U_3 and the optimal cost is 4. If we consider the robust version with b=1, then the optimal solution still contains U_2 and U_3 and the optimal cost is 6. But if we consider the robust version with b=2, then the optimal solution contains U_1 and and the optimal cost is 7.

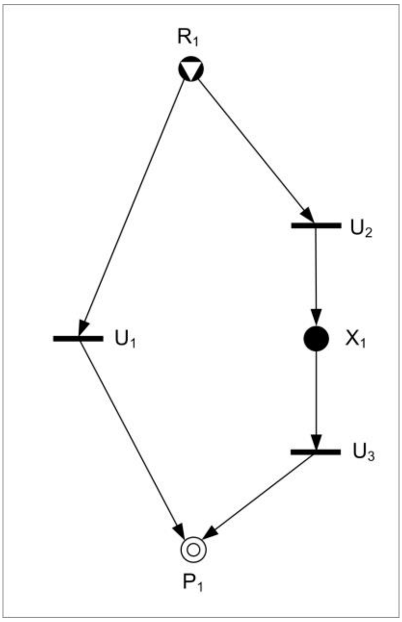

Figure 1
The P-graph of example 1

3 The Branch and Bound Algorithm

Several branch and bound-based algorithms were developed for the solution of the standard combinatorial PNS problem, one can find the details in [9] or in [11]. Here we present an extension which can be used for the the robust problem. We consider a tree where the leaves contain the feasible solutions of the problem. As it is mentioned in the previous section these solutions are identified by the decision mappings defined on the full set of materials. In the inner points of the tree we have partial solutions identified by decision mappings defined only on a subset of the materials. Since the set of the feasible solutions is the same for the standard and the robust problem our algorithm for the robust problem differs only in the bounding function, where we need an estiomation on the new cost function.

The algorithm uses a bounding function B which is defined on all partially defined decision mapping. It gives a lower bound on the costs of the feasible solutions which are the extensions of the decision mapping. The simplest function contains the total nominal cost of the selected operating units plus the sum of the b greatest extra costs among them. Some more difficult bounding functions are presented in [9] and [11], it is an interesting further question to extend them into the robust model.

Algorithm B and B

Initialization: Let the root of the tree be the empty decision mapping, calculate its bounding function and let the list L of the actual nodes contain this root with this value. Moreover, let OPT=N where N is greater than the sum of the extended cost, and OPTG be the empty graph. Continue the procedure with the iteration part.

Iteration Part (while L contains some elements):

Step 1. Choose the element of L, which has the smallest bounding function value, denote it by δ. Consider a material Y where the decision mapping δ is not defined. Extend the decision mapping of δ with all the possible consistent value of Y, denote the possible extensions by $\delta_{1,\ldots,}\ \delta_t$.

Step 2. If δ_i belongs to a feasible solution for some i, and the cost of the solution is smaller than OPT, then change OPTG with δ_i and OPT with the cost of δ_i.

Step 3. Delete δ from the list L and put all δ_i which satisfies $B(\delta_i) \geq OPT$ into L.

Solution: The decision mapping of an optimal solution is stored in OPTG and the optimal value is stored in OPT.

We note that we might have a faster algorithm if we start with a solution given by some heuristic algorithms. There are some heuristic developed for the solution of the standard PNS problem (see [2]) and since the set of feasible solutions is the same for the standard and the robust problem they give a feasible solution for the robust problem as well. On the other hand, it would be interesting to modify these algorithms or design new heuristics for the solution of the robust PNS problem.

4 The Hierarchical Robust PNS Problems

There is a polynomial time solvable class of the PNS problem, which was investigated in [1] and [12]. In this section we extend this algorithm to the more general robust PNS problem. First we recall the definitions. A PNS problem is called *hierarchical* if there exist such partition M_0=R,...,M_l=P of M and partition O_1,..., O_l of O that for each i, i=1,...,l O_i contains operating unit having input materials from M_{i-1} and output materials from M_i. A hierarchycal PNS problem is called *k-wide hierarchical* if $M_i|\leq$ k is valid for i=0,...,l and $|O_i|\leq$ k is valid for i=1,....,l. Figure 2 shows a 3-wide hierarchical problem where M_0={ R_1, R_2}, O_1={U_1, U_2,U_3}, M_1={X_1, X_2,X_3}, O_2={U_4, U_5}, M_2={R_1}.

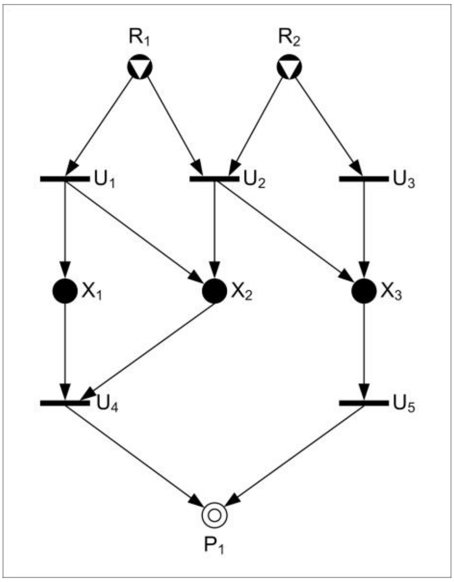

Figure 2
A 3-wide hierarchical P-graph

The assumption that a PNS problem has hierarchical structure seems to be very strong. On the other hand such problems might appear in practical applications. One example is an economic process that works by scheduled order. Then the P-graph contains no cycle and can form a hirarchical structure.

We can solve the hierarchical robust PNS problems with the following algorithm. The basic idea is to define the function F(S,j) which is the cost of the optimal solution which can produce the set S with at most i operating units having extended costs. We calculate this value by dynamic programming for each j=0,...,b and for each subsets of the sets M_i. Then F(P,b) gives the cost of the optimal solution of the problem. We will use an other, set valued function denoted by G(S,j) which notes that in the optimal solution of F(S,j) which set of operating units is used to directly produce S. Finally V(S,j) denotes the number of the operating units which have extended cost among the operating units of G(S,j) in this optimal solution.

Algorithm HSolve

Initialization Let N be a large number which is greater than the sum of the extended costs of the machines, we will use that number to observe the cases where the problem has no feasible solution. Moreover, let F(S,j)=0 for each subset S of R= M_0 and for each j.

The i-th iteration (i=1,...,l) of computing the optimal cost Execute the following steps for each subset S of M_i and for each j=0,...,b.

Step 1. Consider the subsets of Delta(S) (note that each of these sets is included in O_i) and for each such set Q examine that the union of the output materials includes S or not. Denote the sets where S is a subset of that union by $Q_{1,...,}$ Q_t. If no such set exists then F(S,j)=N, V(S,j)=N, and G(S,j) is empty for each j. Otherwise go to step 2.

Step2. For each Q_r, r=1,...t let q_r denote the number of the operating units in the set and $E(Q_r,p)$ the sum of the p largest extra costs in set Q_r for each p=0,...,q_r. Now calculate the following value for each r=1,....,t.

$$C_r = \max\{\ F(in(Q_r),j-p) + C(Q_r) + E(Q_r,p)\ \mid\ p=0,..., \min(j, q_r)\ \}$$

Let the set with the minimal value be denoted by Q, the minimal value is by C, the value of p where the minimum is obtained by p*. If there are more minimal values choose the set with the smallest index. Then let F(S,j)=C, G(S,j)=Q, N(S,j)=p*.

Determination of the optimal structure If F(P,b)≥N, then the problem has no feasible solution. Otherwise let A=P, v=b and O* be empty, and perform the following step while A is not a subset of the raw materials. Let O*= O* ∪ G(A,v) and let A=in(G(A,v)) and let v=v−V(A,j).

The optimal solution is the P-graph (M*,O*), where M*=in(O*) ∪ out(O*).

Theorem *If a PNS problem is hierarchical then algorithm Hsolve gives an optimal solution of the problem or it finds that the problem has no feasible solution.*

Proof: First we prove that if the algorithm gives a solution then the produced sets (M*,O*) yield a P-graph which is a feasible solution. Since in the determination phase we start to build O* with the set producing P, it follows immediately that P is a subset of in(O*) thus it is contained in M*. Therefore, property (A1) holds.

In a hierarchical PNS problem there is no operating unit producing raw material, thus we get that in (M*,O*) there can be no edge going into a raw material. On the other hand, consider a material X from M*. Then it is an input or an output material of some operating units of O*. If it is an output material then obviously there is an edge going into it in the solution. If it is an input material of some elements of O, then it becomes an element of A during the determination phase. If after that no further iteration comes then X is a raw meterial, otherwise we extend O* with some operating units producing X, thus there will be an edge leading into it in the P-graph (M*,O*). Therefore, we proved that (A2) is valid.

To prove (A3) we have to show that for each operating unit there exists a path from it into a desired product. One can see this statement by induction on the iteration steps of the determination phase. At the beginning we choose such operating units which produce directly desired products. Later in each step we choose operating units which produce input materials of some operating units selected earlier.

Property (A4) is obvious by the definition of M*.

Therefore we can conclude that algorithm Hsolve returns a feasible solution. Now we prove that it finds the optimal one. First we show the following lemma.

Lemma 2 *F(S,j) gives the smallest cost which can be used to produce all materials from set S with j extended costs for each set S which is a subset of M_i for i=0,...,l and for each j=0,....,b..*

Proof of the lemma: We prove this statement by induction on i. For i=0 all elements in M_0 are raw material, therefore, the P-graph which contains only S is a feasible solution with cost 0 for any j. Thus the statement is valid for i=0. Now suppose that i<l and Lemma 2 holds for all subsets of M_i. We will prove it for the subset of M_{i+1}.

Consider now an arbitrary subset S of $M_{i+1,}$ and a feasible solution producing S from the raw materials in the robust model with bound j. In this solution some operating units are selected from level $O_{i+1,}$ denote the set of these operating units by U. Then the input materials of U from level M_i are also produced. If p operating units have extended cost from set U then in(U) has to be produced with minimal cost using j-p extended costs. On the induction assumption this can be done by the cost F(in(U),j-p). Therefore, the cost of producing S will be the maximum of the values F(in(U),j-p)+C(U)+E(U,p). On the other hand, this value was considered in the minimum which calculates F(S,j) thus we obtained that F(S,j) cannot be larger than the cost of producing S. On the other hand, the determination phase produces a solution with cost F(S,j) thus the lemma is valid for the subsets of M_{i+1} as well.

By this lemma we obtain the correctness of the algorithm easily. If F(P,j)<N, then the determination phase builds a solution with cost F(S,j), therefore by Lemma 2 it is optimal. If F(P,j)≥N, then we prove that no feasible solution exists by contradiction. Suppose, we have some feasible solutions. Consider the optimal solution, and denote its cost by OPT. Then OPT cannot be greater than the sum of the extended costs of all operating units, thus OPT<N. But by Lemma 2 we have F(P,b)=OPT and this gives a contradiction.

Theorem 3. *The time complexity of algorithm HSolve is $O(4^k blk)$ which is linear if the parameters k and b are considered as constant.*

Proof. In the initialization part we assign the value of each subsets of R and for each j, thus it takes $O(2^k b)$ time. In each iteration we consider b times all subsets of M_i, therefore we perform Step 1,2 of the iteration phase all together at most $O(2^k bl)$ times. In Step 1 in the worst case we have to consider all subsets of O_i, thus we have at most 2^k sets and for each of them the maximum in Step 2 has to be determined which needs at most O(k) time. Therefore the total time needed to perform the iteration part is $O(4^k blk)$. Finally the determination phase needs time O(l).

Remark: We must note that the running time of the algorithm is exponential in the width of the problem, therefore it is effective only for small k-s.

Conclusions

In this work we defined the robust extension of the combinatorial PNS model which can be used in the analysis of process network synthesis and workflow problems. We presented the first results in this area: an exponential time branch and bound based algorithm for the solution of the problem in the general case, and a faster polyniomial time algorithm for the solution of thin hierarhical problems. We think so that there are many further interesting questions to investigate. As far as the branch and bound algorithm is concerned it would be interesting to develope further versions using other bounding functions or other material selection rules, and to compare the efficiency of the different versions. Morever, it is also an interesting question how one can extend the heuristic algorithms presented for the standard PNS problem.

On the other hand, we defined and studied only the combinatorial model in this paper, which can be used to analyse the structure of the networks. It is an interesting question how to extend this model to the quantitative PNS model where the cost of an operating unit is not a constant but depends on the amount of materials used and produced by it. Moreover, in modeling workflow problems it would be important to take into account the shifts, it is also an interesting question to extend the robust PNS model into this direction. Finally, we mention that there are some results on the PNS problem where some other objective besides the cost is also taken into account (see [13]) it would be an interesting question to extend the robust model to handle this situation as well.

Acknowledgement

This work was partially supported by the European Union and the European Social Fund through project Telemedicina (Grant no.: TÁMOP-4.2.2.A-11/1/KONV-2012-0073).

References

[1] Blázsik, Z.; Holló, Cs.; Imreh, B.; Imreh, Cs.; Kovács, Z.: On a Well Solvable Class of the PNS Problem, *Novi Sad Journal of Mathematics*, **30**, 21-30, 2000

[2] Blázsik, Z.; Holló, Cs.; Imreh, Cs.; Kovács, Z.: Heuristics for the PNS Problem, Optimization Theory, Mátrahaza 1999, *Applied Optimization* **59** eds. F. Gianessi, P. Pardalos, T. Rapcsák, Kluwer Academic Publishers, Dordrecht, Boston, London, 1-18, 2001

[3] Bertsimas, D.; Brown, D.; Caramanis, C.: Theory and Applications of Robust Optimization, *SIAM Review*, **53**, 464-501, 2011

[4] Bertsimas, D.; Sim, M.: Robust Discrete Optimization and Network Flows, *Mathematical Programming Series B*, **98**, 49-71, 2003

[5] Fan, L. T.; Kim, Y.; Yun, C.; Park, S. B.; Park, S.; Bertok, B.; Friedler, F.: Design of Optimal and Near-Optimal Enterprise-Wide Supply Networks for Multiple Products in the Process Industry, *Ind. Eng. Chem. Res*., **48**, 2003-2008, 2009

[6] Friedler, F.; Fan, L. T.; Imreh, B.: Process Network Synthesis: Problem Definition, *Networks*, **28**, 119-124, 1998

[7] Friedler, F.; Tarján, K.; Huang, Y. W.; Fan, L. T.; Graph-Theoretic Approach to Process Synthesis: Axioms and Theorems, *Chem. Eng. Sci.*, **47(8)**, 1973-1988, 1992

[8] Friedler, F.; Varga, J. B.; Fan, L. T.: Decision Mappings: a Tool for Consistent and Complete Decisions in Process Synthesis, *Chem. Eng. Sci***,** **50(11),** 1995, 1755-1768

[9] Holló, Cs.: A Look Ahead Branch-and-Bound Procedure for Solving PNS Problems, PU.M. A., **11(2)**, 2000, 265-279

[10] Imreh, B.; Friedler, F.; Fan, L. T.: An Algorithm for Improving the Bounding Procedure in Solving Process Network Synthesis by a Branch-and-Bound Method *Developments in Global Optimization*, editors: I. M. Bonze, T. Csendes, R. Horst, P. M. Pardalos, Kluwer Academic Publisher, Dordrecht, Boston, London, 1996, 301-348

[11] Imreh, B.; Magyar, G.: Empirical Analysis of Some Procedures for Solving Process Network Synthesis Problem, *Journal of Computing and Information Technology*, **6**, 1998, 372-382

[12] Imreh, Cs.: A New Well-Solvable Class of PNS Problems, *Computing*, **66**, 289-296, 2001

[13] Imreh, Cs.; Kovács, Z.: On Pollution Minimization in the Optimization Models of Process Network Synthesis, Chemical Engineering Transactions, **7(2)**, 565-570, 2005

[14] Monaci, M.; Pferschy, U.: On the Robust Knapsack Problem, *Optimization Online*, http://www.optimization-online.org/DB_FILE/2011/04/3019.pdf, 2011

[15] Süle, Z.; Bertok, B.; Friedler, F.; Fan, L. T.: Optimal Design of Supply Chains by P-Graph Framework Under Uncertainties, Chemical Engineering Transactions, **25**, 453-458, 2011

[16] Tick, J.: Fuzzy Extension to P-Graph-based Workflow Models, Proceedings of the 7th IEEE International Conference on Computational Cybernetics, ICCC 2009, 2009, pp. 109-112

[17] Tick, J.: P-Graph-based Workflow Modeling, Acta Polytechnica Hungarica (ISSN: 1785-8860) 4: (1) pp. 75-88 (2007)

[18] Tick, J.: P-gráf alapú workflow modellezés fuzzy kiterjesztéssel 95 p. 2007 (Disszertáció: PhD)

[19] Tick, J.: Workflow Modeling Based on Process Graph, Proceedings of the 5th Slovakian - Hungarian Joint Symposium on Applied Machine Intelligence and Informatics SAMI 2007, Poprad, Slovakia, January 25-26, 2007, pp. 419-426, ISBN:978-963-7154-56-0

[20] Tick, J., Kovács, Z.: P-Graph-based Workflow Synthesis, Proceedings of the 12th International Conference on Intelligent Engineering Systems, INES 2008, Miami, USA, February 25-29, 2008, pp. 249-253, ISBN: 9781424420834

[21] Tick, J.: Fuzzy Control Systems Based on Parametric T-Norm Function, Proceedings of the 4th International Symposium on Applied Computational Intelligence and Informatics SACI 2007, Timisoara, Romania, May 17-18, 2007, pp. 215-218, ISBN: 142441234X