# Optimization in Graph Transformation Systems Using Petri Net Based Techniques *

Szilvia Varró-Gyapay and Dániel Varró

Budapest University of Technology and Economics
Department of Measurement and Information Systems
H-1117 Budapest, Magyar tudósok körútja 2.
{gyapay,varro}@mit.bme.hu

**Introduction.** The design of business or production systems frequently necessitate to build a system that is *simultaneously correct and optimal*, i.e., the system has to simultaneously fulfill logical and numerical conditions. For instance, in workflows, each activity can be constrained by certain budget restrictions, and a typical requirement is to find the cost-optimal solution trajectory respecting all budget and temporal correctness constraints over this trajectory that leads to a desirable (target) system configuration.

Thus, we need to combine reachability analysis with optimization. A primary basis of a combined technique can be based upon [4] where optimal scheduling problems are solved by embedding Branch-and-Bound techniques into the model checker Spin [3]. The same problem has been tackled on a Petri net basis using Process Network Synthesis algorithms for optimization in [1]. In [2], we aimed at solving optimization problems in graph transformation systems with time using the CheckVML tool [5] and SPIN. However, in all these cases, the dynamic creation and deletion of system objects is a priori bounded, which is a major restriction.

In this paper, we assume that the evolution of the system is captured by graph transformation systems (GTS) where a cost parameter is attached to each graph transformation rule denoting the cost of firing that rule.

The joint optimization and reachability analysis can be carried out by exploring the state space induced by the graph transformation system (also called as a graph transition system) as in case of model checking. However, we apply numerical conditions to guide the exploration strategy by temporally pruning the search tree as in case of branch-and-bound techniques.

**A Petri net abstraction of GTS.** These cuts are based on a Petri net abstraction of graph transformation systems introduced in [6]. The essence of this technique is to derive a cardinality P/T net which simulates the original GTS by abstracting from the structure of instance models (graphs) and only counting the number of elements (nodes or edges) of a certain type by placing tokens to a corresponding place. These tokens are circulated by transitions derived from

* This work was partially supported by the Sensoria European IP (IST-3-016004) and the János Bolyai scholarship.

each graph transformation (GT) rule which simulate the effect of the rule on the number of elements of certain types by adding and removing tokens from corresponding places. This technique is extended by simply copying the cost factors of a GT rule to the corresponding transition.

**A search strategy for optimization.** By formulating the target state configuration as submarking of the P/T net, we can solve the linear integer programming problem of the derived P/T net using the incidence matrix as described in [1] to determine a transition occurrence vector which delivers a cost-optimal solution. Note that solving the optimization problem on the P/T level is of much lower complexity than solving the problem directly on the GTS-level.

Obviously, due to the abstractions, this transition occurrence vector might not be fireable, i.e. there might not exist a corresponding trajectory in the graph transition system. However, this vector provides excellent guidance for the state exploration strategy. Those branches which are not compliant with the optimal transition occurrence vector are temporally cut, i.e. compliant steps are explored first when traversing the state space.

This means that if this minimal cost is exceeded along a path or a GT rule is applied more times than it is prescribed in the transition occurrence vector, then the exploration of the branch is postponed.

If no solution is found on the GTS-level, then the next P/T-level candidate for the optimal solution is derived, and the exploration of the GTS continues towards this direction.

## References


1. S. Gyapay and A. Pataricza. Optimal trajectory generation for Petri nets. *Acta Cybernetica*, vol. 17:pp. 225–245, 2005.
2. S. Gyapay, Á. Schmidt, and D. Varró. Joint optimization and reachability analysis in graph transformation systems with time. In *Proc. GT-VMT 2004, International Workshop on Graph Transformation and Visual Modelling Techniques*, vol. 109 of *ENTCS*, pp. 137–147. Elsevier, 2004.
3. G. Holzmann. The model checker SPIN. *IEEE Transactions on Software Engineering*, vol. 23(5):pp. 279–295, 1997.
4. T. C. Ruys. Optimal scheduling using branch and bound with SPIN 4.0. In *Proc. 10th International SPIN Workshop*, vol. 2648 of *LNCS*, pp. 1–17. Springer, Portland, Oregon, USA, 2003.
5. Á. Schmidt and D. Varró. CheckVML: A tool for model checking visual modeling languages. In P. Stevens, J. Whittle, and G. Booch (eds.), *Proc. UML 2003: 6th International Conference on the Unified Modeling Language*, vol. 2863 of *LNCS*, pp. 92–95. Springer, San Francisco, CA, USA, 2003.
6. D. Varró, S. Varró-Gyapay, H. Ehrig, U. Prange, and G. Taentzer. Termination analysis of model transformations by Petri nets. In *Proc. Third International Conference on Graph Transformation (ICGT 2006)*, LNCS. Springer, 2006. In press.