

Hybrid Planning by Combining SMT and Simulated Annealing^{*}

Extended Abstract

Jaroslav Skaruz¹, Artur Niewiadomski¹, and Wojciech Penczek²

¹ Institute of Computer Science, Siedlce University, Poland

{jaroslav.skaruz}, {artur.niewiadomski}@uph.edu.pl

² Institute of Computer Science, PAN Warsaw and Siedlce University, Poland
wpenczek@gmail.com

1 Introduction

We present a new approach to the concrete planning (CP) shown to be a NP-hard problem [7]. This is the third stage of Web service composition (WSC) in the Planics framework [1]. The first two phases, namely an abstract planning and offer collecting, basing on an ontology, a user query, and a service registry provide data necessary for CP. A new hybrid algorithm (HSA), which combines Simulated Annealing (SA) [6] with Satisfiability Modulo Theories (SMT) [3], has been designed and implemented. The main idea of our hybrid solution relies upon generating an initial individual by an SMT-based procedure. Then, in the subsequent iterations of SA, the individual is improved. The experimental results show that HSA is superior to the other methods we have applied to the CP problem, including these based on Genetic Algorithm (GA) [4], SMT used separately [10], and SMT combined into the hybrid algorithms RH and SRH [9], as well as the IPH algorithm [8].

Our direct motivation to develop hybrid algorithms is based on the observation that every method applied separately to WSC yields fair results, but suffers from some disadvantages. While the SMT-based algorithm is able to find always the optimal solution, its main problem is a long execution time and large memory consumption. On the other hand the evolutionary methods are quick and demand less resources, but at the price of the quality and a lower probability of finding solutions. We are aiming at combining the algorithms in order to get a trade-off between speed and quality.

Recently, we have developed three planning methods based on joining GA and SMT: RH (Random Hybrid), SRH (Semi-Random Hybrid), and IPH (Initial Population Hybrid). The first two algorithms run alternately several iterations of GA and the SMT-based procedure which is aimed at improving the best individuals of a GA population. The IPH algorithm makes use of an SMT-based procedure in order to generate (a part of) the initial population meeting the given constraints, and then the individuals are improved by GA. The experiments have shown that the latter method is superior in most cases, thus we have chosen this scheme to be used in further investigations.

^{*} This work has been supported by the National Science Centre under the grant No. 2011/01/B/ST6/01477.

After replacing GA by SA (and introducing several necessary technical modifications) we have obtained the HSA algorithm which seems to be the best one so far.

The problems similar to CP have been intensively studied in the last decade. In [2] the authors use two algorithms to deal with WSC, namely GA and SA. Experiments show better quality of the solutions found by SA comparing to GA, but at the price of a higher computation time and lower scalability. On the other hand, hybrid algorithms for WSC are also known in the literature, see e.g., [5], where a modified version of the Particle Swarm Optimization algorithm is used.

Next, we present our solution followed by experimental results and conclusions.

2 Solution

First, we briefly recall the formulation of CP as a constrained optimization problem. We focus here on the intuitions only, but all the formal definitions can be found in [10].

The input for the concrete planning consists of n *offer sets* and a set of *constraints*. Each offer from a particular set is a tuple of values corresponding to attributes of objects being processed by services. Overall, a solution of CP consists in selecting one offer from each offer set, such that all the constraints are satisfied and the value of the quality function Q (given as a part of a user query) is maximized.

Algorithm. SA is an optimisation technique based on an observation of the physical process of annealing in metallurgy, i.e., a technique of heating and slow cooling of a material in order to improve its properties. SA implements the cooling process as a slow decrease in the probability of accepting worse solutions while the algorithm explores the search space. In SA, our “processed material” is a potential solution of a problem, called an individual. The space exploration is realised by applying a neighbourhood operator to the individual, which results in obtaining a new potential solution. In our case, the individual is a sequence of natural values representing offer indices, while the neighbourhood operator changes one value randomly.

It is quite hard to force SA to search for better solutions with the constraints satisfied, because employing standard mechanisms like penalty functions known from GA is problematic. Thus, our algorithm applies an SMT-based procedure (similar to the one described in [9]) to generate an initial individual which satisfies all constraints, and the next steps of our hybrid algorithm are similar to those of a standard SA. The main difference is that new individuals may be accepted provided that all constraints are still satisfied. Since an initial individual is already a solution (usually of a low quality), the HSA algorithm always returns a result, and the objective is to increase the value of the quality function.

Experiments. We have evaluated the efficiency of HSA using the same six benchmarks as in the papers [10, 9]. All the instances represent plans of length 15. Each offer set of Instance 1, 3, and 5 contains 256 offers, which makes the number of the potential solutions equal to $256^{15} = 2^{120}$. In the case of Instance 2, 4, and 6, each offer set consists of 512 offers, which results in the search space size as large as $512^{15} = 2^{135}$.

In Table 1 we compare the performance of HSA with our previous results. The first column contains the instance number, while the next columns present computation

times and the quality function values of the solutions found with our different methods: HSA, IPH (in two variants - with 1 and 500 individuals generated by SMT), and non-hybrid ones: SMT and GA. The HSA algorithm is the fastest one and finds solutions of reasonable quality. It has to be mentioned that all the presented methods return solutions with 100% probability, except for GA, where the probability is much lower: from 7% to 12%.

Table 1. Experimental results of the IPH and HSA algorithms.

Instance	HSA		IPH ₁		IPH ₅₀₀		SMT		GA	
	t[s]	Q	t[s]	Q	t[s]	Q	t[s]	Q	t[s]	Q
1	4.0	1420	5.6	1229.5	13.9	1317.9	266.0	1443.0	5.0	1218.5
2	5.1	1402	6.4	1248.8	20.1	1382.4	388.0	1467.0	5.6	1319.9
3	5.4	2153	6.4	1706.8	14.6	2090.6	500.0	2266.0	6.0	2085.4
4	6.5	2194	7.5	1788.0	20.9	2280.3	500.0	2409.0	6.6	2001.9
5	4.7	354	6.4	329.1	27.8	634.5	500.0	781.0	5.1	436.0
6	6.3	541	8.8	458.2	55.2	524.7	500.0	755.0	5.9	537.8

3 Conclusions

We have presented a new approach to concrete planning. An SMT-based procedure is employed for generating an initial individual of the SA algorithm. Such an individual is already a solution because it satisfies all the constraints. During the next steps of our algorithm the individual is improved in order to find a concrete plan having a greater value of the quality function. The experiments confirm that HSA is efficient, since the results are better than those obtained by all our former algorithms, including the hybrid ones.

References

1. et al., D.D.: PlanICS - a web service composition toolset. *Fundam. Inform.* 112(1), 47–71 (2011)
2. Arockiam, L., Sasikaladevi, N.: Simulated annealing versus genetic based service selection algorithms. *International Journal of u- and e- Service, Science and Technology* 5, 35–49 (2012)
3. De Moura, L., Bjørner, N.: Satisfiability modulo theories: Introduction and applications. *Commun. ACM* 54(9), 69–77 (Sep 2011), <http://doi.acm.org/10.1145/1995376.1995394>
4. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edn. (1989)
5. Hu, C., Chen, X., Liang, X.: Dynamic services selection algorithm in web services composition supporting cross-enterprises collaboration. *Journal of Central South University of Technology* 16, 269–274 (2009)

6. Kirkpatrick, S., Jr, C.G., Vecchi, M.: Optimization by simulated annealing. *Sci.* 220, 671–680 (1983)
7. Niewiadomski, A., Penczek, W., Skaruz, J.: SMT vs genetic algorithms: Concrete planning in PlanICS framework. In: CS&P. pp. 309–321 (2013)
8. Niewiadomski, A., Penczek, W., Skaruz, J.: Combining Genetic Algorithm and SMT into Hybrid Approaches to Web Service Composition Problem. *Int. Journal On Advances in Software* 7(3, 4), 675–685 (2014)
9. Niewiadomski, A., Penczek, W., Skaruz, J.: A hybrid approach to web service composition problem in the PlanICS framework. In: Awan, I., Younas, M., Franch, X., Quer, C. (eds.) *Mobile Web Information Systems, Lecture Notes in Computer Science*, vol. 8640, pp. 17–28. Springer International Publishing (2014), http://dx.doi.org/10.1007/978-3-319-10359-4_2
10. Niewiadomski, A., Penczek, W., Skaruz, J., Szreter, M., Jarocki, M.: SMT versus Genetic and OpenOpt Algorithms: Concrete Planning in the PlanICS Framework. *Fundamenta Informaticae* 135(4), 451–466 (2014)