Comparing Genetic Algortihms and Matrix Factorization for Learning Heuristics in Constraint Solving

Seda Polat Erdeniz and **Andrei Popescu** [1]

Abstract.
Knowledge-based recommender systems assist users in the active configuration of complex products. These systems rely on solving Constraint Satisfaction Problems (CSP). In constraint solving, variable and value ordering heuristics help to increase efficiency. Applying such heuristics can increase the performance of CSP solvers. On the other hand, if we apply specific heuristics to similar CSPs, CSP solver performance could be further improved. In previous work, we have proposed novel approaches to learn such heuristics, however, an evaluation in terms of consistency and prediction quality is still lacking. In this paper, we evaluate an compare two proposed approaches to learn heuristics, one relying on Genetic Algorithms and Clustering, and one on Matrix Factorization, on the same problem. Our results provide valuable insights for future research in this domain.

1 Introduction

The configuration of complex items, such as financial services, software artefacts, and cars, is a cumbersome task (from a user point of view) in the scope of the mass customization business model [5]. In this context, a user often requires (or would benefit) from intelligent support during the configuration task, to overcome sub-optimal scenarios induced by time constraints, information overload, and product suitability issues. A widespread intelligent support system in this scenarios is provided by recommender systems.

A recommender system can be defined as any system that guides a user in a personalized way to interesting or useful objects in a large space of possible options or that produces such objects as output [4]. Knowledge-based recommender systems, in specific constraint-based recommender systems [2], have been widely adopted in scenarios involving the recommendation of complex tasks. These systems generate recommendations by solving the corresponding Constraint Satisfaction Problem (CSP). Since the search space can quickly become challenging, heuristics [6] are of fundamental importance in constraint-based recommender systems.

Erdeniz et al. [10] have proposed a constraint-based recommendation approach that provides accurate heuristics based on historical configuration transactions. By integrating matrix factorization into the computation of search heuristics for the feature model configuration task, Erdeniz et al. [10] guarantee the consistency of determined recommendations.

The prediction of the preference of a user for a specific item was based on elementary matrix multiplication operation, so-called *matrix factorization* [7], as often are model-based collaborative filtering approaches. Matrix factorization is based on the idea of parameterizing two low-dimensional matrices U and V in such a way that U x V = R'. The matrix R is an approximation of the original user x item preference matrix R. Consequently, missing values in R can be estimated by the multiplication of the two low-dimensional matrices U and V.

There exist a couple of research contributions focusing on the integration of feature model configurators with recommender systems. Rodas-Silva et al. [11] introduce a content-based and collaborative filtering approach to the recommendation of components that should be selected for the implementation of a given configuration. Pereira et al. [8, 1] integrate different collaborative filtering approaches into feature model configuration processes to proactively support users in the navigation through complex feature spaces. In this context, the authors also apply matrix factorization with the goal to further improve the prediction quality of the recommender system. Finally, Falkner et al. [3] provide an overview on different scenarios that can benefit from the integration of recommender systems with knowledge-based configurators. In another work, Erdeniz and Felfernig [9] use *k-means clustering* and a *genetic algorithm* (GA) in order to compute search heuristics to tackle the graph colouring problem.

Compared to the approach of Erdeniz et al. [10], all of the mentioned approaches do not propose a solution focusing on the integration of recommendations into the search heuristics of a configurator and thus not being able to guarantee the consistency of determined recommendations and runtime performance at the same time.

In this paper, we provide an evaluation and comparison between the approach developed by Erdeniz et al. [10] and the work of Erdeniz and Felfernig [9], in terms of prediction quality and achieved consistency in the recommendation task. The contributions of our paper are three-fold: (1) we apply the approach of Erdeniz and Felfernig [10], and Erdeniz et al. [9] on a configuration dataset, allowing a direct comparison in terms of consistency and prediction performance; (2) we evaluate and compare the two approaches in terms of achieved consistency and prediction quality; (3) we point out future research directions to be pursued towards machine learning-based search heuristics.

In the remainder of the paper, we first describe the approach followed by Erdeniz et al. [10] in Section 2, and explain the work of Erdeniz and Felfernig [9] in Section 3. In Section 4 we cover in more details our evaluation approach, the results of which we discuss in Section 5. We conclude in Section 6.

[1] Graz University of Technology, Austria, email: andrei.popescu@ist.tugraz.at

2 Matrix Factorization-based Heuristics

Erdeniz et al. [10] have applied a constraint-based recommendation approach to compute value ordering heuristics, in order to support the recommendation of missing configuration parameters in the scope of an online personalized bike shop. The proposed approach exploits historical transactions in order to recommend missing configuration parameters to the currently active user configuring his/her product, by means of the *matrix factorization*. Matrix factorization is first applied on the sparse matrix composed of historical transactions (complete or incomplete) concluded by past users. This step produces a *dense matrix* R' providing estimated configuration parameter values for the whole matrix. For users that completed their configuration, the recommendation of parameters is straightforward, as it is sufficient to provide the complete historical transaction (HT1), whereas for users who have provided an incomplete configuration, the corresponding dense transaction is used to obtain search heuristics for the constraint-based recommender.

Concerning the transaction currently active (AT), in the scope of which the user needs support, value ordering for missing variable assignments are obtained based on the *k-nearest neighbours* (Euclidean Distance) applied with respect to the dense matrix. The obtained search heuristic is then used by a CSP solver to provide an accurate recommendation.

3 Genetic Algorithms with Clustering Heuristics

Another approach to computing optimal variable and value ordering heuristics for Constraint Satisfaction Problems is to employ *genetic algorithms*.

Genetic algorithms are a subclass of Evolutionary Algorithms inspired by the idea of natural selection. The algorithm starts with a random *population* of random individuals (the population can be seen as the solution to the problem, for example a set of variable assignments), which is iteratively improved *generation* by *generation*. An improvement of the *population* in the current generation is achieved by selecting the most appropriate individuals through a *fitness function* (in our case based on the satisfaction of problem's constraints), and by modifying the genome of such individuals in order to compose the next generation.

Erdeniz and Felfernig [9] have proposed *Cluster and Learn*, an approach that first uses *k-means clustering* and then applies a genetic algorithm to learn heuristics for solving the graph colouring problem. In the first phase, *Cluster and Learn* performs a clustering operation based on the *euclidean distance* between two user requirements. A genetic algorithm is then applied to the obtained clusters in order to minimize the runtime of CSP solving, returning the optimal variable and value ordering heuristics.

4 Methodology

In this paper, we compare the performance of both the work of Erdeniz et al. [10] and Erdeniz and Felfernig [9] in terms of overall achieved consistency and prediction quality. While Erdeniz et al. [9] have evaluated their approach in terms of runtime, in the scenario of active configuration, consistency and prediction quality are of fundamental relevance to assure a high user satisfaction, and thus ensure a positive interaction with the configurator. In this work, we provide a more detailed evaluation, that compares these two approaches with respect to the consistency and prediction quality of recommended configurations.

To allow direct comparison between the two studies, we implement both approaches and evaluate their performance on the basis of a CSP comprised of 10 variables having a domain of size 5, and two constraints, one domain constraint and one user constraint. The following is the definition of the used CSP.

```
Variables: V0, V1, V2, V3, V4, V5, V6, V7, V8, v9, V10
Domains: dom(V1) = dom(V2) .... = dom(v10) = {1, 2, 3, 4, 5}
Domain Constraints:
c1: x > 2 -> y >= 4
User Requirements:
c2: x = req.val where req.val is a value imposed by the user
```

Pre-computed solutions of the above introduced CSP (without the user requirement) were used to build the training and test set. The test set in particular, consists of solutions to the problem, with an increasing number of missing variable assignments (up to #variables-1). We use matrix factorization (MP), matrix factorization with CSP solving (MF with CSP solving), and genetic algorithms with CSP solving (GA with CSP solving) to recommend values for the missing assignments. The genetic algorithm was hyper-tuned with the following parameters:

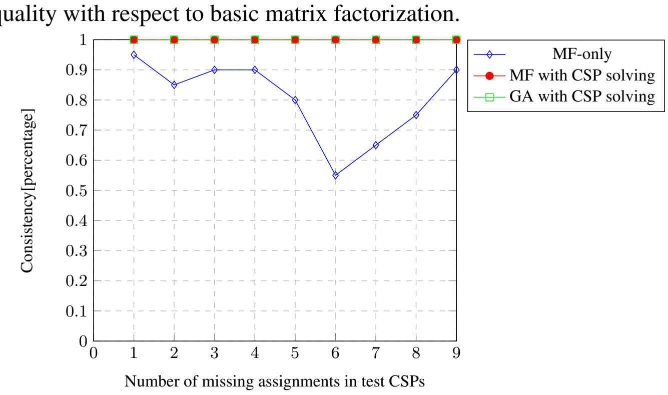
Generations	10
Cross-over Rate	0.9
Mutation Rate	0.05 / sizeOfGenes

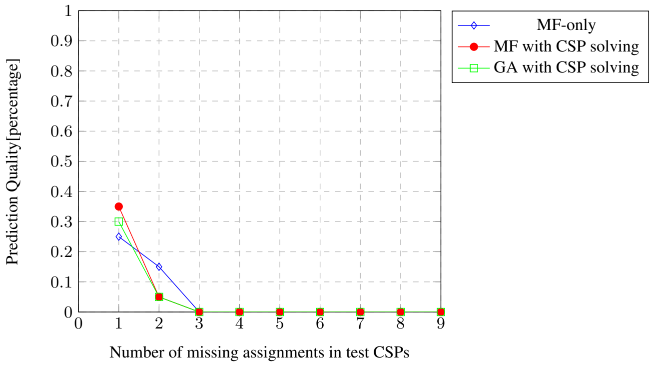
Table 1. GA hyper-parameters

The number of clusters for the Cluster and Learn approach [9] was set to 4 for simplicity reasons. For each test case, the average performance metrics (consistency and prediction quality) are calculated and discussed in Section 5.

5 Evaluation Results

According to the results depicted in the following figures, the approach of Erdeniz et al. [10] and Erdeniz and Felfernig [9] perform similarly. It can also be observed that two approaches perform better in terms of consistency, and only slightly better in terms of prediction quality with respect to basic matrix factorization.





Consistency

In terms of consistency, we can observe stable and high consistency of recommended variable assignments. In specific, as shown in the first figure, by using the heuristics obtained with Cluster and Learn and with the approach integrating Matrix Factorization with Constraint solving, we are able to recommend variable assignments which are always consistent. The improvement is significant with respect to an approach that would simply rely on matrix factorization, especially when the number of missing variable assignments in the test set increases. In other words, consistency is not guaranteed when using matrix factorization only.

Prediction Quality

With respect to prediction quality, all approaches perform similarly, showing an abrupt decay in quality as the missing variable assignments increase.

6 Conclusion and Future Work

We have evaluated and compared two previously proposed value ordering heuristics, [9] and [10], that improve both the runtime and quality of achieved recommendations in the context of active product configuration. We observed compelling growth in terms of the consistency of the two approaches, when compared with an approach relying on matrix factorization alone.

We foresee potential in machine learning-based heuristics for constraint solving to be applied in the scope of active configuration, multi-configuration, and reconfiguration. Consequently, we call for further research in this direction, for example, using a different set of learning models, for instance artificial neural networks.

REFERENCES

[1] Juliana Alves Pereira, Pawel Matuszyk, Sebastian Krieter, Myra Spiliopoulou, and Gunter Saake, 'Personalized recommender systems for product-line configuration processes', *Computer Languages, Systems Structures*, **54**, (02 2018).

[2] Robin Burke, 'Hybrid recommender systems: Survey and experiments', *User Modeling and User-Adapted Interaction*, **12**, (11 2002).

[3] Andreas Falkner, Alexander Felfernig, and Albert Haag, 'Recommendation technologies for configurable products', *AI Magazine*, **32**(3), 99–108, (Oct. 2011).

[4] Alexander Felfernig and Robin Burke, 'Constraint-based recommender systems: technologies and research issues.', p. 3, (08 2008).

[5] Alexander Felfernig, Lothar Hotz, Claire ONeill, and Juha Tiihonen, *Knowledge-Based Configuration: From Research to Business Cases*, 05 2014.

[6] Chris Groër, Bruce Golden, and Edward Wasil, 'A library of local search heuristics for the vehicle routing problem', *Mathematical Programming Computation*, **2**, 79–101, (06 2010).

[7] Yehuda Koren, Robert Bell, and Chris Volinsky, 'Matrix factorization techniques for recommender systems', *Computer*, **42**(8), 30–37, (2009).

[8] Juliana Alves Pereira, Pawel Matuszyk, Sebastian Krieter, Myra Spiliopoulou, and Gunter Saake, 'A feature-based personalized recommender system for product-line configuration', in *Proceedings of the 2016 ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences*, GPCE 2016, p. 120–131, New York, NY, USA, (2016). Association for Computing Machinery.

[9] Seda Polat Erdeniz and Alexander Felfernig, 'Cluster and learn: Cluster-specific heuristics for graph coloring', in *12th International Conference on the Practice and Theory of Automated Timetabling (PATAT'18)*, pp. 401–404, (2018).

[10] Seda Polat Erdeniz, Alexander Felfernig, Müslüm Atas, and Ralph Samer, 'Matrix factorization based heuristics for constraint-based recommenders', in *SAC '19, Proceedings Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 1655–1662, United States, (2019). Association of Computing Machinery.

[11] Jorge Rodas-Silva, José A. Galindo, Jorge García-Gutiérrez, and David Benavides, 'Selection of software product line implementation components using recommender systems: An application to wordpress', *IEEE Access*, **7**, 69226–69245, (2019).