Supplier Optimization at Bosch with Knowledge Graphs and Answer Set Programming

Cuong Xuan Chu, Mohamed H. Gad-Elrab, Trung-Kien Tran, Marvin Schiller, Evgeny Kharlamov, and Daria Stepanova

Bosch Center for Artificial Intelligence, Renningen, Germany
{firstname.lastname}@de.bosch.com

Abstract. The automotive industry is constantly facing the challenge of optimizing their suppliers to meet customer demands while keeping costs low. Knowledge graphs have proven to be effective tools for modeling complex supply chains, but their use for optimization is limited. In this paper, we report on our experience at Bosch to use Answer Set Programming (ASP) to optimize component suppliers in the automotive industry based on knowledge graphs. Evaluation on industrial products shows both efficiency and effectiveness of our modeling framework in generating optimal solutions for supply chain management problems.

1 Introduction

Supplier Optimization. The challenge of *supplier optimization* in the automotive industry concerns identifying the best suppliers for automotive parts and components based on various criteria, *e.g.* quality, reliability, cost, and delivery time. Usually the production of a single system relies on a large number of suppliers providing various parts and components. Managing relationships with a large network of suppliers is time-consuming and resource-intensive. Therefore, supplier optimization is essential to ensure the timely delivery of high-quality components at a competitive cost.

Supply Chain Knowledge Graph. Following our semantic driven strategy [7], we rely on semantic technologies for solving the supplier optimization problem. More specifically, we represent the suppliers and their products (i.e., characteristics, such as the price, quality, lead time, delivery options, certifications, etc.) in knowledge graphs (KGs) [5]. The supplier graph is integrated with the KG representing the bill of materials (BOM), i.e., the components and materials needed for product manufacturing.

Answer Set Programming for Optimization. To compute the optimal set of suppliers, we utilize *answer set programming (ASP)* [3, 4, 6], i.e., a declarative programming paradigm which provides a simple modeling language allowing for a succinct representation of search and optimization problems. Problems are encoded in programs, i.e., finite sets of rules, whose answer sets (which are special models) computed by dedicated ASP solvers, yield the solutions of a problem.

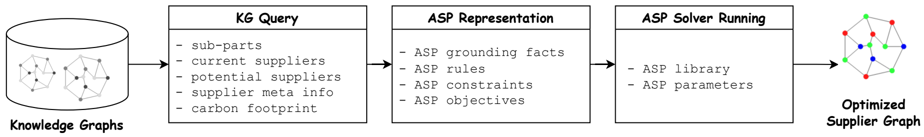

Fig. 1. Architecture overview of the supply chain optimization service.

Approach. In our solution, we utilize the flexibility of KGs and the power of ASP to optimize the suppliers for materials and components of a given product. This problem is particularly challenging due to the large size of the input KG, and directly invoking ASP solvers on the full data is not feasible. Thus, we develop strategies for extracting only relevant facts required for the optimization. Moreover, to ensure wide usage of our service, we make it accessible also for users without ASP background.

2 Supplier Optimization

Figure 1 describes the overview of the process for using ASP in optimizing suppliers for different parts of a product in the KG. We proceed with describing the required data, the details of each component of our pipeline, as well as discuss the user experience.

Required Input. To model the supplier optimization problem, the following components are required: (1) *input facts*, related to the target product. In our use-case, we are working on multiple knowledge graphs, including **supplier KGs**, containing over 4.2M *part–plant–supplier* relationships, along with information about suppliers, such as supplier types (`producer or reseller`), location and carbon footprint; **BOM KGs**, containing a hierarchical structure of the parts of a target product and their current suppliers. Usually a product contains thousands of parts and sub-parts; (2) the *objectives* of the optimization problem, *e.g.*, minimizing the number of suppliers for the product and the total carbon footprint; (3) *constraints*, depending on customers' requirements, e.g., "avoiding reseller", "restrictions on the number of suppliers per part", etc.

Optimization Process. The process starts with querying the KGs to extract the relevant facts including the product, its parts, as well as current and potential suppliers and defining the optimization objectives, rules and constraints. Then, the retrieved facts, rules, and objectives are represented in the Answer Set Programming modeling language [6].

We pass the constructed ASP encoding to the state-of-the-art ASP solver `clingo` [4] to compute possible solutions to the problem, i.e., sets of optimal suppliers for each product, along with the relevant meta information (*e.g.,* number of optimal suppliers, their total carbon footprint, etc.).

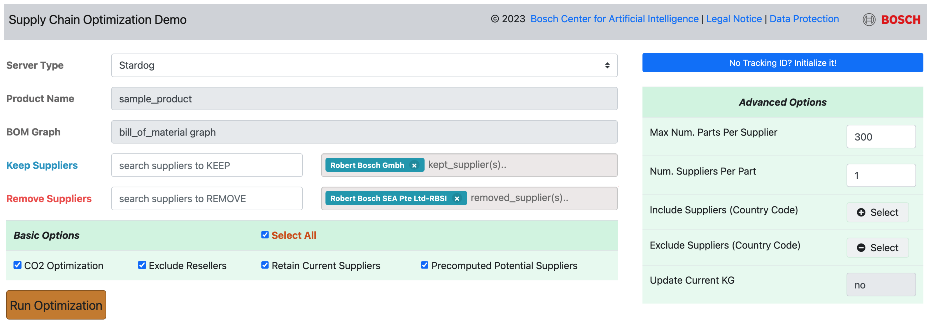

Fig. 2. Web interface for the supply chain optimization service.

Table 1. Sample statistics for the suppliers of a **single** product and the optimization results obtained by the service. ([1] two current suppliers & one new supplier)

Number of **product parts**	**362**
Number of **current** suppliers	**171**
Number of **potential** suppliers	**3302**
Number of **optimal** suppliers	**3**[1]
Total carbon footprint of **current** suppliers	**44245**
Total carbon footprint of **optimal** suppliers	**50**

User Experience. We have implemented our ASP-based optimization approach as a webservice (i.e. API) that supports asynchronous interaction, allowing the users to define new optimization problems, specify KGs used in the input, set optimization objectives as well as product-specific constraints. The customers are able to interact with the system by adding further simple constraints (e.g., exclusion or inclusion of certain potential suppliers) in an iterative fashion if the computed solution does not meet their expectations. In the output, a KG with the optimal selection of suppliers is returned to the user. Along with the API, we also provide a simple web interface to facilitate the interaction with our system for supplier optimization (see Figure 2 for illustration).

Optimization Results. We have evaluated our service for supplier optimization on real industrial data. As an example, in Table 1 we present the statistics of the input dataset and the supplier optimization results for an electronic device. One can observe that with the help of our supplier optimization service, we have managed to reduce the number of suppliers from 171 to only 3 suppliers. In Figure 3 we additionally report the detailed results of the optimization process, *i.e.,* intermediate solutions with the number of suppliers, their total carbon footprint as well as the time required for their computation. The optimal number of suppliers with the best carbon footprint has been computed in only 15 seconds, demonstrating the usability of our service in real industrial settings.

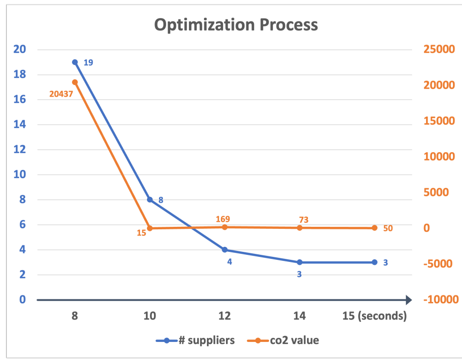

Fig. 3. Optimization results on the sample product.

3 Discussion and Future Work

Our solution for supplier optimization paves the way towards wide adoption of KGs and ASP-based technology for industrial use cases. As future work, we are going to improve the scalability of the presented service by exploiting recent extensions of answer set programming solvers with large neighborhood optimization strategies [1, 2]. We also plan to automatically learn complex ASP constraints from the user feedback in order to iteratively improve the quality of the resulting solutions.

Acknowledgements: The work was partially supported by EU projects Dome 4.0 (GA 953163), OntoCommons (GA 958371), DataCloud (GA 101016835), Graph Massiviser (GA 101093202), EnRichMyData (GA 101093202), and SMARTEDGE (GA 101092908).

References

1. T. Eiter, T. Geibinger, N. H. Ruiz, N. Musliu, J. Oetsch, and D. Stepanova. Large-neighbourhood search for optimisation in answer-set solving. In *AAAI 2022*.
2. T. Eiter, T. Geibinger, N. H. Ruiz, N. Musliu, J. Oetsch, and D. Stepanova. ALASPO: an adaptive large-neighbourhood ASP optimiser. In *KR 2022*, 2022.
3. T. Eiter, G. Ianni, and T. Krennwallner. Answer set programming: A primer. In *Reasoning Web 2009*. Springer.
4. M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub. Multi-shot ASP solving with clingo. *Theory Pract. Log. Program.*, 19(1):27–82, 2019.
5. A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, et al. Knowledge graphs. *ACM Computing Surveys*, 2021.
6. V. Lifschitz. *Answer Set Programming*. Springer, 2019.
7. B. Zhou, Z. Zheng, D. Zhou, G. Cheng, E. Jiménez-Ruiz, T.-K. Tran, D. Stepanova, M. H. Gad-Elrab, N. Nikolov, A. Soylu, et al. The data value quest: A holistic semantic approach at Bosch. In *The Semantic Web: ESWC 2022 Satellite Events*. Springer, 2022.