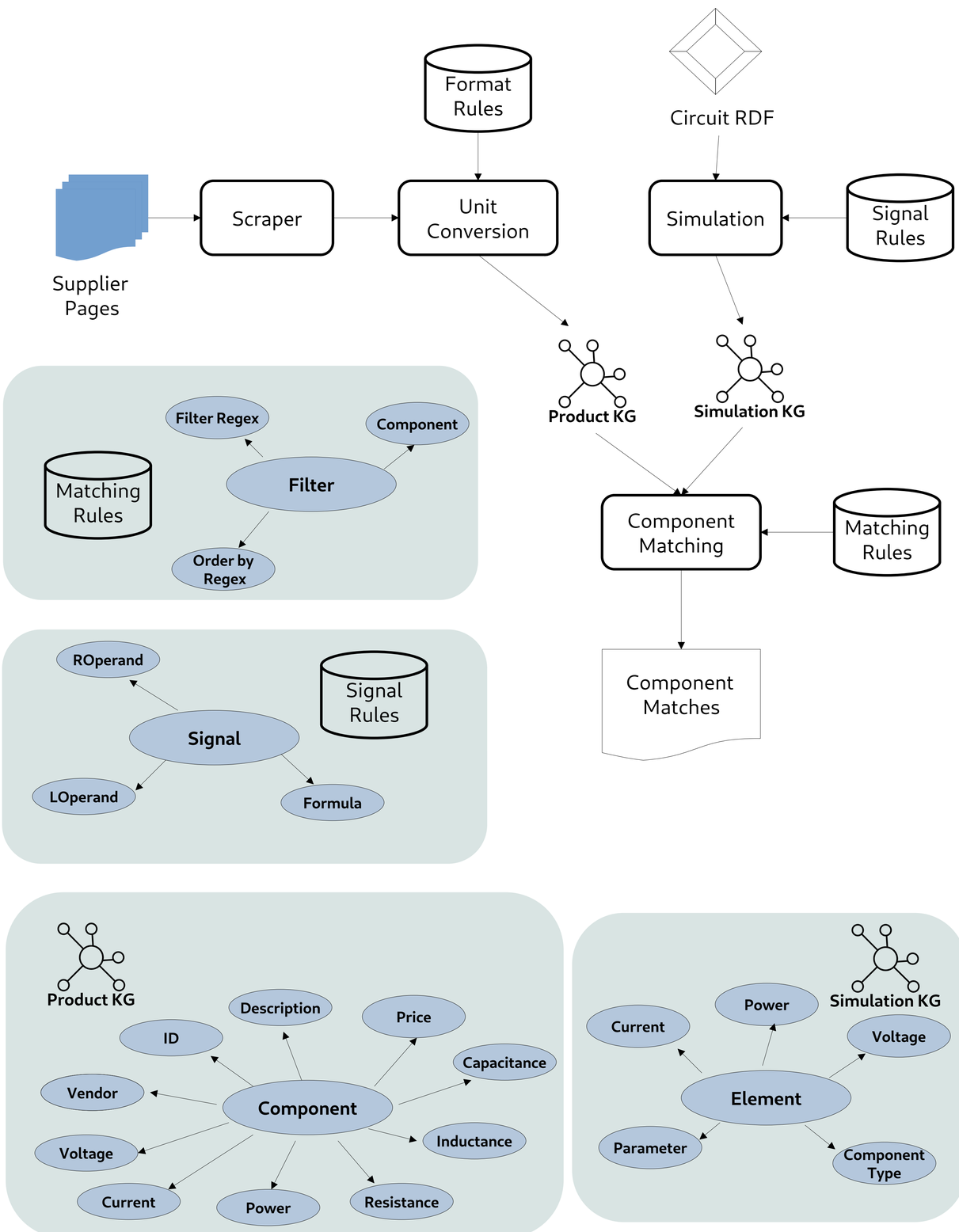
Knowledge Graph based Electrical Circuit Simulation and Component Selection

Mizanur Rahman Syed, Johannes Bayer
{mizanur_rahman.syed, bayer.johannes}@dfki.de



Electrical circuits can be interpreted as Graph structures with components represented as nodes and edges between components represented as edges between components. Node properties and design constraints can be viewed as node properties.

The simulation of electrical circuits requires domain knowledge (Ohm's law etc.), system knowledge (circuit) and product knowledge (information of different products available), similar to approach proposed in [2]. The proposed flow encapsulates much of this information in the form of knowledge graphs (KGs) and RDF triplets, which enables the addition of additional functionality and data simple and without the need for additional code.

Product KG: A knowledge graph for different products available in online stores is constructed with the help of purpose built scrapers, which read online product catalogues and save this information in a KG using predefined property relations.

Online data sources can use different notations and conventions to describe electrical properties, because of which, standardization of the notations and units is necessary. This is handled by specifying RegEx based format rules to standardize product properties stored in the Product KG.

Simulation KG: The system knowledge or input circuit to the system is also described using RDF. With the help of scripts, the circuit is simulated with PySpice[1] and node voltage outputs from PySpice are saved.

The system then uses signal rules (also specified using RDF) to identify formulae that can be applied using known parameters to calculate new signal values (e.g. power consumption of a resistor can be calculated if the potential difference and resistance are known). The enriched circuit graph is saved in a Simulation KG for the circuit.

Component Selection: Information in the Simulation KG can be used to shortlist products from the Product KG using matching rules in SPARQL [3] [4], which specify how parameters for circuit components can be used to shortlist products.

For ex: the power consumption for a resistance can be used to filter out products with a lower max power rating in the Product KG. Finally, component level ordering rules can be used to find the best match for the component.

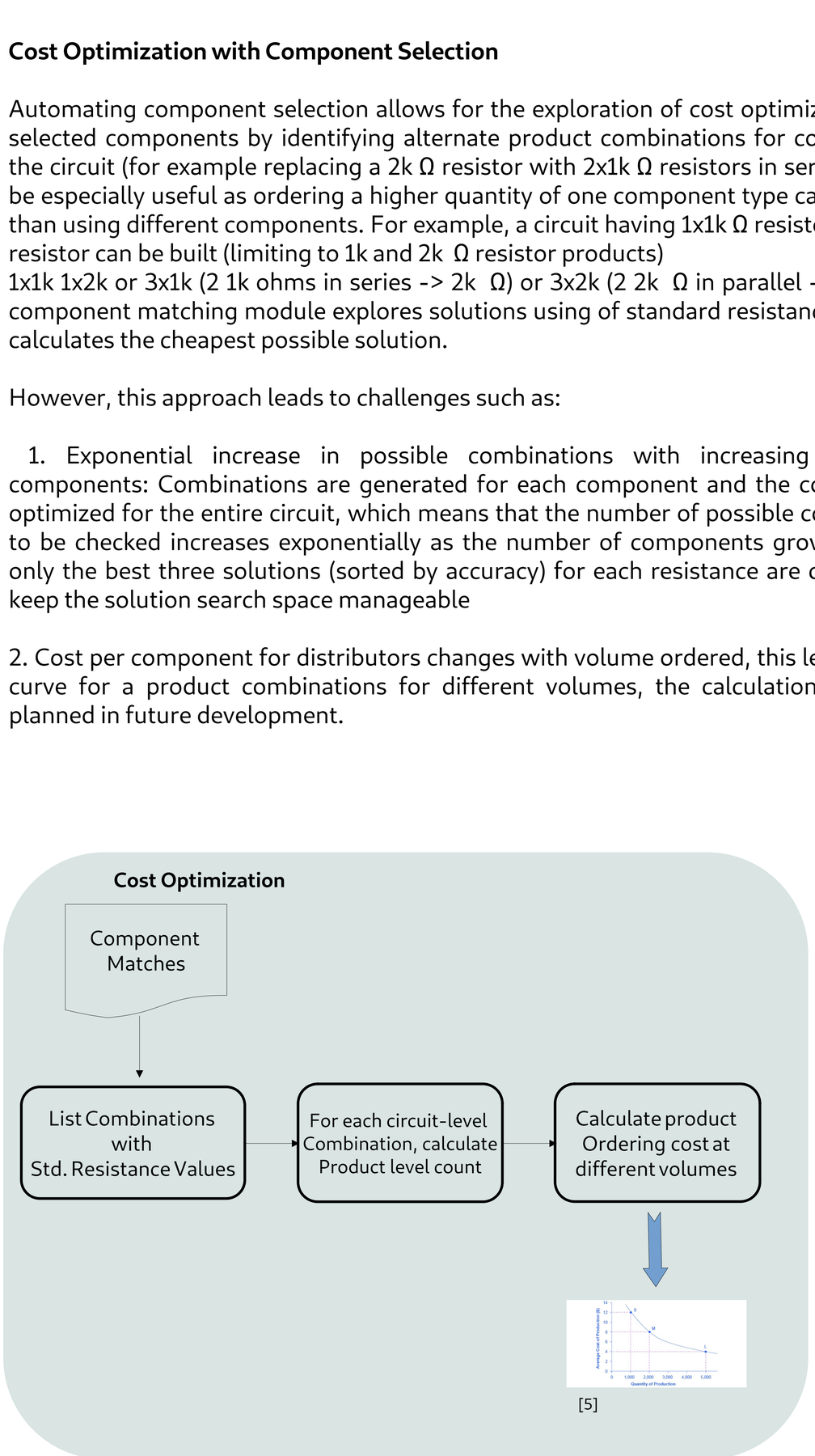
Cost Optimization with Component Selection

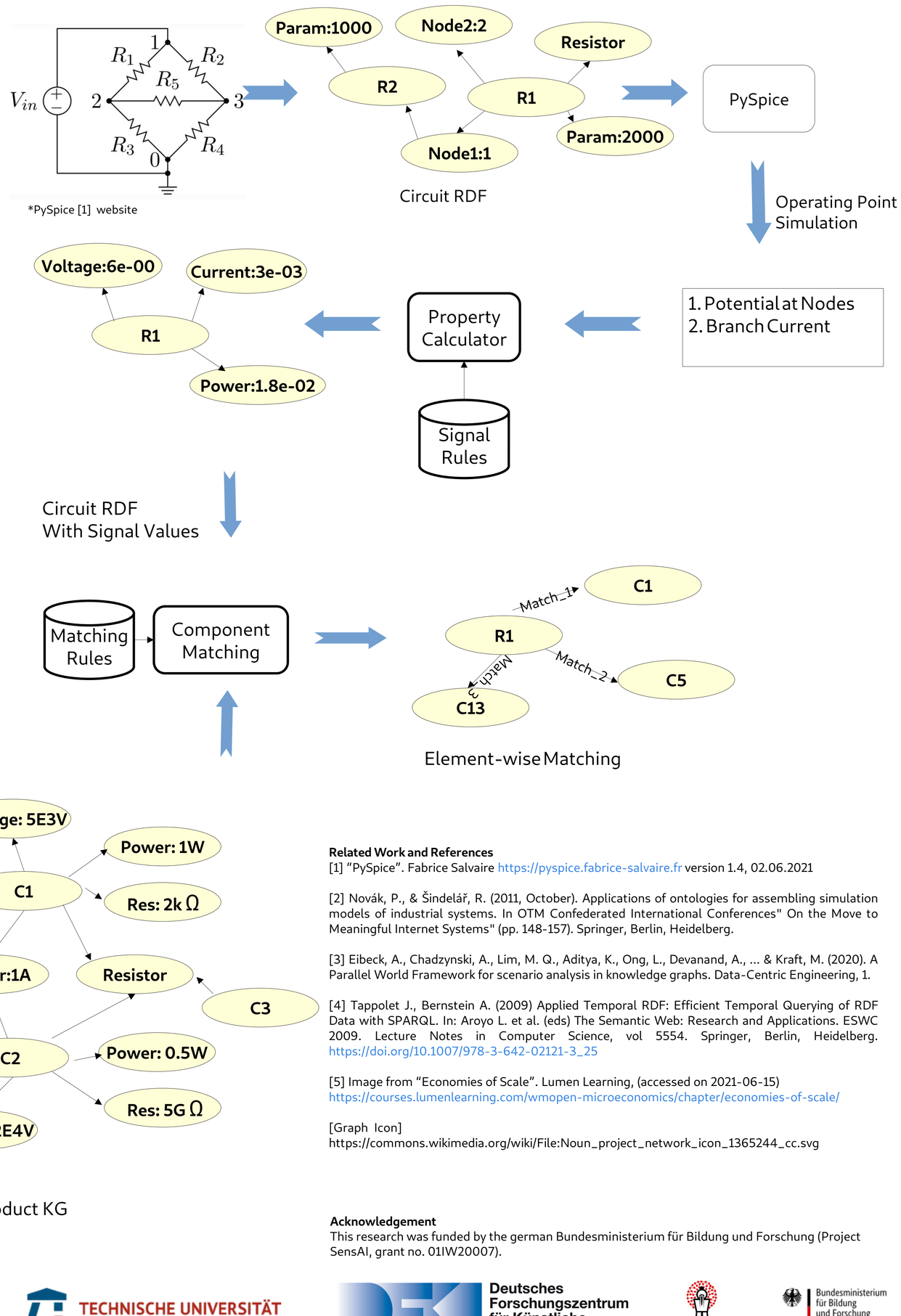
Automating component selection allows for the exploration of cost optimization for the selected components by identifying alternate product combinations for components in the circuit (for example replacing a 2k Ω resistor with 2x1k Ω resistors in series). This can be especially useful as ordering a higher quantity of one component type can be cheaper than using different components. For example, a circuit having 1x1k Ω resistor and 1x2k Ω resistor can be built (limiting to 1k and 2k Ω resistor products)
1x1k 1x2k or 3x1k (2 1k ohms in series -> 2k Ω) or 3x2k (2 2k Ω in parallel -> 1k Ω). The component matching module explores solutions using of standard resistance values and calculates the cheapest possible solution.

However, this approach leads to challenges such as:

1. Exponential increase in possible combinations with increasing number of components: Combinations are generated for each component and the cost has to be optimized for the entire circuit, which means that the number of possible configurations to be checked increases exponentially as the number of components grows. Currently only the best three solutions (sorted by accuracy) for each resistance are considered to keep the solution search space manageable

2. Cost per component for distributors changes with volume ordered, this leads to a cost curve for a product combinations for different volumes, the calculation of which is planned in future development.





Product KG

Related Work and References
[1] "PySpice". Fabrice Salvaire https://pyspice.fabrice-salvaire.fr version 1.4, 02.06.2021

[2] Novák, P., & Šindelář, R. (2011, October). Applications of ontologies for assembling simulation models of industrial systems. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 148-157). Springer, Berlin, Heidelberg.

[3] Eibeck, A., Chadzynski, A., Lim, M. Q., Aditya, K., Ong, L., Devanand, A., ... & Kraft, M. (2020). A Parallel World Framework for scenario analysis in knowledge graphs. Data-Centric Engineering, 1.

[4] Tappolet J., Bernstein A. (2009) Applied Temporal RDF: Efficient Temporal Querying of RDF Data with SPARQL. In: Aroyo L. et al. (eds) The Semantic Web: Research and Applications. ESWC 2009. Lecture Notes in Computer Science, vol 5554. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-02121-3_25

[5] Image from "Economies of Scale". Lumen Learning, (accessed on 2021-06-15) https://courses.lumenlearning.com/wmopen-microeconomics/chapter/economies-of-scale/

[Graph Icon]
https://commons.wikimedia.org/wiki/File:Noun_project_network_icon_1365244_cc.svg

Acknowledgement
This research was funded by the german Bundesministerium für Bildung und Forschung (Project SensAI, grant no. 01IW20007).

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN
Deutsches Forschungszentrum für Künstliche Intelligenz GmbH
SENSAI
Bundesministerium für Bildung und Forschung