# Towards an Automated Optimization-as-a-Service Concept

Sascha Bosse[a], Abdulrahman Nahhas[b], Matthias Pohl[c] and Klaus Turowski
*Very Large Business Applications Lab, Faculty of Computer Science Otto-von-Guericke University Magdeburg, Germany*

Keywords: Cloud Computing, Optimization, as-a-Service, Business Analytics.

Abstract: Many organizations try to apply analytics in order to improve their business processes. More and more cloud services are offered to support these efforts. However, the support of prescriptive analytics is weak. While concepts for such an optimization-as-a-service exist, these require much expert knowledge in solution methods. In this paper, a workflow for optimization-as-a-service is proposed that utilizes an optimization knowledge base in which machine learning techniques are applied to automatically select and parametrize suitable solution algorithms. This would allow consumers to use the service without expert knowledge while reducing operational costs for providers.

## 1 INTRODUCTION

Data-intensive IT systems nowadays play an important role in information systems, demonstrated by the fact that organizations are in desperate need for data scientists, big data analysts or comparable experts (King and Magoulas 2015). In order to take advantages of the opportunities of data-based analysis, a process with three phases has been identified (Evans and Lindner 2012):

- Descriptive analytics, which means characterizing and understanding the past;
- Predictive analytics, in which it is desired to estimate the likely future; and
- Prescriptive analytics, i.e. making decisions to improve the future.

Recent developments in AI technology drive analytics efforts today. While the former two phases are strongly connected to the field of machine learning, optimization is a crucial aspect especially in the latter phase.

An organization applying analytics can build its own capabilities or obtain them from external service partners, while both approaches are cost-intensive (Pohl et al. 2018). In this context, cloud computing has revolutionized the way IT services are obtained in the past years. While there are many cloud offerings for descriptive and predictive analytics, there are almost no IT service providers offering optimization-as-a-service (OaaS).

Especially meta-heuristics such as genetic algorithms are interesting in this context. These are general, computing-intensive procedures based on function comparisons that are applicable for a wide range of real-world problems, for instance, virtual machine placement (Müller et al. 2016), redundancy allocation (Coit and Smith 1996), order sequencing (Nahhas et al. 2017), or design planning (Lanza et al. 2015). Furthermore, especially population-based meta-heuristics are by nature well suited for a high degree of parallelization. Thus, the availability of massive, distributed computing power makes the cloud “the ideal environment for executing metaheuristic optimization experiments” (Pimminger et al. 2013).

For consumers, the concept of cloud computing eliminates the problem of oversized systems for computing-intensive tasks (Foster et al. 2008; Pimminger et al. 2013) by the use of elastic, pay-per-use self-services over the internet (Mell and Grance 2011). Thus, costs can be reduced for obtaining optimization services for consumers, but also for providers which can utilize economies of scale (Marston et al. 2011).

Although concepts for such a service have been introduced (e.g. in (Kurschl et al. 2014)), it remains unclear how it can be effectively used by consumers

[a] https://orcid.org/0000-0002-2490-363X
[b] https://orcid.org/0000-0002-1019-3569
[c] https://orcid.org/0000-0002-6241-7675



Bosse, S., Nahhas, A., Pohl, M. and Turowski, K.
Towards an Automated Optimization-as-a-Service Concept.
DOI: 10.5220/0007746303390343
In *Proceedings of the 4th International Conference on Internet of Things, Big Data and Security (IoTBDS 2019)*, pages 339-343
ISBN: 978-989-758-369-8

with limited knowledge of problem formulations and solution algorithms. On the other hand, providers are bound to identify potential solutions to a problem efficiently in order to reduce costs for both provider and consumer. With a sufficient knowledge base of optimization problems and the use of machine learning techniques, the process of selecting a suitable solution algorithm could be automatized. Therefore, this short paper aims at presenting a standard workflow for solving optimization problems in the context of OaaS in order to discuss automation potentials and future research opportunities.

## 2 RELATED WORK

Although the idea of combining meta-heuristics and cloud computing is promising, only a few scientific works deal with this topic. Most existing analytics cloud service concepts lack of a prescriptive component (cf. e.g. (Pohl et al. 2018)) and most commercial offerings such as SAP Leonardo[d] do not contain optimization services. Other such as IBM Bluemix [e] or Google Optimization Tools [f] only support mathematical optimization methods. In contrary to meta-heuristics, these methods are often not scalable to real-world (NP-hard) problems or restrict the search space artificially (Coit and Smith 1996; Soltani 2014). Despite the lack of cloud services, several frameworks for parallel, distributed meta-heuristic optimization have been implemented such as HeuristicLab[g] which could be utilized in an OaaS context.

In (Pimminger et al. 2013), the authors investigate the suitability of performing meta-heuristic optimization in cloud scenarios. For that reason, large scale experiments are conducted with different deployment strategies. The authors conclude that utilizing cloud resources for optimization massively reduces costs for users.

Kurschl et al. follow this idea and provide a requirements catalogue as well as a reference architecture for OaaS (Kurschl et al. 2014). Although technical questions such as multi-tenancy and scalability are discussed, workflow-related topics are not analyzed in detail. Consumers can access a cloud platform to select and parametrize solution methods for defined problems. Accounting is proposed to be done on basis of the obtained computing power, concepts for automation of the problem solution workflow are not discussed.

## 3 A WORKFLOW FOR OaaS

In order to address the idea of solving optimizations automatically without expert knowledge, a workflow in context of the OaaS concept is presented in the following.

When a potential user of an OaaS offering encounters an optimization problem, this problem has to be formalized by providing (Gill et al. 1993):

- A set of decision variables $x_1, \dots, x_n$ with the respective domain,
- A set of $k$ constraints $c_i: (x_1, \dots, x_n) \mapsto \{0; 1\}$ with $1 \leq i \leq k$, and
- An objective function $f: (x_1, \dots, x_n) \mapsto (y_1, \dots, y_m)$ mapping decision variables to $m$ objective values.

The workflow of problem formulation and solution in context of the OaaS concept is modeled as a diagram in the Business Process Model and Notation (BPMN) language and presented in Figure 1.

In order to simplify the process of problem formulation, basic problem classes should be provided (Kurschl et al. 2014). However, a user should also have the opportunity to formulate novel optimization problems. Anyway, decision variables should be characterized by defining a domain for each variable, e.g. the set of real numbers or defined discrete (or even binary) values. Based on the decision variables, constraints can be defined by entering symbolic functions such as simple bounds on decision variables or linear combinations.

The first critical point of the workflow is the definition of the objective function. Available optimization frameworks often require a symbolic objective function which simplifies the optimization process, but may not always be available or applicable. Therefore, implicit definitions of objective functions should be supported additionally. These can either be defined with a white-box or a black-box approach.

In the white-box approach, the objective function is modeled analytically, e.g. in a state-space model such as Markov chains. For that reason, the OaaSshould make use of a modeling engine which

[d] https://www.sap.com/germany/products/leonardo.html
[e] https://www.ibm.com/us-en/marketplace/decision-optimization-cloud
[f] https://developers.google.com/optimization/
[g] https://dev.heuristiclab.com/trac.fcgi/

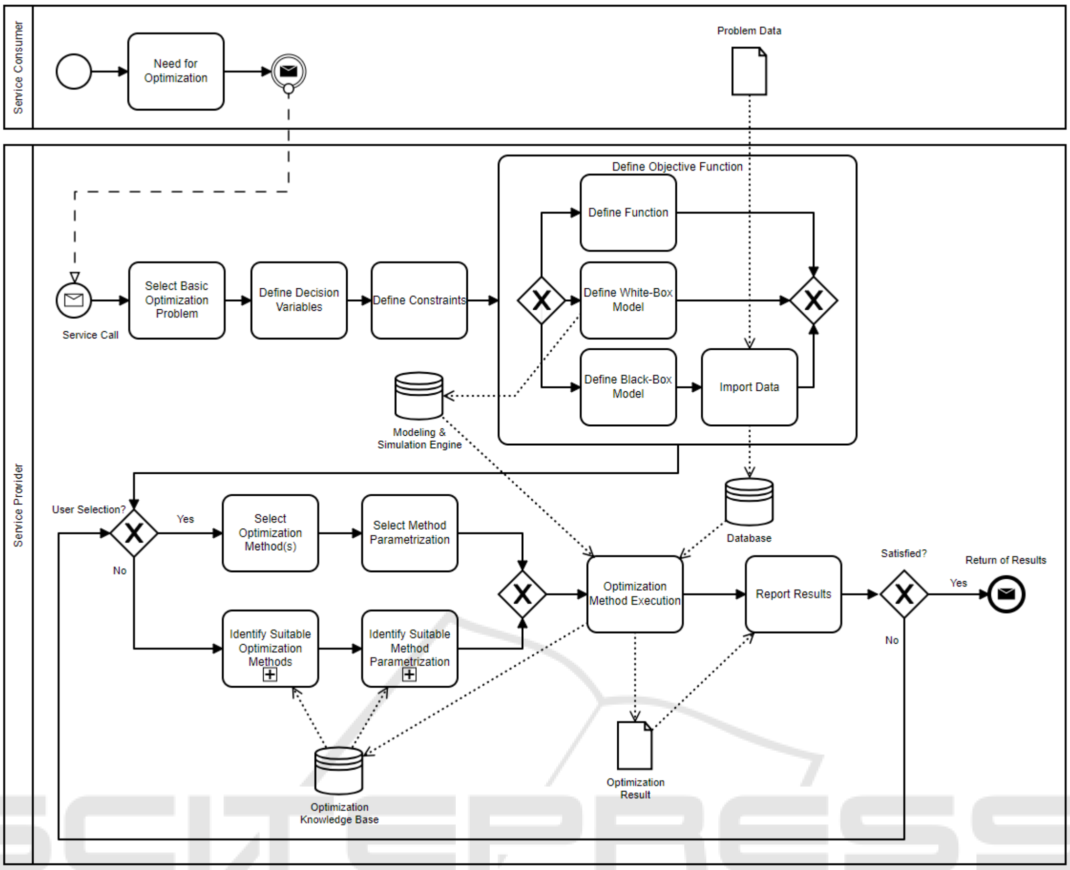


Figure 1: Workflow of an optimization service request.



allows users to define a variety of models for the objective function. These models can be evaluated numerically or by simulation. In the black-box approach, the mapping from decision variables to objective values is done by providing a high amount of data which allows a supervised machine learning algorithm to model the objective function.

After the problem has been formulated, (feasible) solutions should be identified. In related work, it is assumed that the user makes their own selection of algorithms and parametrization. If this knowledge is available to the user, this will be an efficient approach for both consumer and provider. However, if the problem is novel or there is a lack of expertise which algorithm will provide best results, there will be two alternatives: first, the user executes an own search for the best algorithm by trying different algorithms and parametrizations. However, this will lead to high costs due to the amount of needed computing resources which may not be tolerated by the consumer.

As a second alternative, the provider could take care of the process to identify suitable solution algorithms and parametrizations. In order to make a good price to the consumer as well as minimizing operational costs, this process should be very efficient by utilizing economies of scale. Therefore, an optimization knowledge base is proposed in the workflow. This knowledge base includes (meta-)data about different optimization problems as well as about the effectivity and efficiency of solution algorithms and their parametrization. While the knowledge base is sparse, a high number of experiments should be done by the provider in order to generate this data. In order to keep response times low, efficient parallelization should be applied. Although this will induce high operational costs in the beginning, these can be dramatically reduced when the knowledge base is enriched.

While this idea is novel to the field of meta-heuristic optimization, such concepts have been developed for the field of machine learning in which the question which algorithm to use with which hyper parameters is also a difficult one (meta-learning). These include landmarking (Pfahringer et al. 2000) or meta-regression (Charles et al. 2000; García-Saiz and Zorilla 2017).

When a method has been selected and parametrized, the experiment can be executed by applying the optimization method on the decision variables and the defined objective function to generate results. If the user is not satisfied with these results, they can select another one or with the gained knowledge, the optimization method can be chosen more carefully. Otherwise, results are returned to the consumer.

# 4 ILLUSTRATION: THE REDUNDANCY ALLOCATION PROBLEM

In order to illustrate the generic workflow description given above, the redundancy allocation problem (RAP) is used. In this problem, the allocation of parallel-redundant components in a series system is to be optimized. The following, simple problem definition is reported to be NP-hard (Chern 1992): A system consists of $n$ required subsystems. In each subsystem, a number of components can be operated in active redundancy. These components are characterized by a reliability $r_i$ and cost $c_i$. The reliability of the system is to be maximized subject to a cost constraint. With respect to the generic problem definition given above, this would lead to:

- Decision variables $x_1, \dots, x_n$ with $x_i \in [1,2,\dots,n_{max}]$ indicating the number of components to be allocated in each subsystem (limited by an upper bound),
- As a constraint, system cost should not exceed a certain value: $\sum_{i=1}^{n} x_i c_i \leq C$, and,
- The objective function under the assumption of independent component failures, which is a non-linear function

$$\max \prod_{i=1}^{n} (1 - (1 - r_i)^{x_i}).$$

However, this formulation has been adapted and extended in the last decades in order to approach reality of complex systems. For instance, heterogeneous or passive redundancy have been introduced in decision variables (Coit and Smith 1996; Sadjadi and Soltani 2015). This led to more complex objective functions that would be evaluated by white-box, e.g. in (Bosse et al. 2016; Chi and Kuo 1990; Lins and Droguett 2009), or black-box approaches, e.g. in (Hoffmann et al. 2004; Silic et al. 2014). Therefore, a possible user of an RAP module in an OaaS context would require to freely define the problem class to be solved.

Several meta-heuristic algorithms have been developed in recent years to solve different RAP. These include, for instance, genetic algorithms, simulated annealing, tabu, harmony and cuckoo search, as well as ant colony, immune-based, swarm, and bee colony optimization (Soltani 2014). Although several experiments have been conducted and presented in the literature, the question which algorithm is to be preferred under which circumstances remains unanswered in a general scale (Kuo and Prasad 2000). Additionally, algorithm selection and parametrization can depend on the exact problem formulation.

As an example, consider the boundary for the number of components $n_{max}$. This boundary has been intended to limit the problem space in order to increase efficiency of solution algorithms. However, it has also an effect to parametrization as illustrated by the genetic algorithm presented in (Coit and Smith 1996): In this paper, a solution is encoded as an integer string of length $n \cdot n_{max}$, in which every integer indicating the index of the component used or, if the integer is the successor of the last index, that no component is used in this slot. If $n_{max}$ is a large number or is even undefined, this encoding scheme cannot be applied effectively in a genetic algorithm, so that other encoding schemes should be utilized.

In order to efficiently offer an OaaS, the provider would require to run many experiments to serve requests effectively. In these experiments, different algorithms and parametrizations are to be applied to a specific RAP. By relating information about problem formulation (e.g. number of decision variables, upper bound, ratio of cost constraint to mean component cost etc.) and class (e.g. type of objective function) to quality of solution algorithms (e.g. solution feasible, (penalized) objective value etc.), the RAP knowledge base can be filled and analyzed. On this basis, the provider can select and parametrize solution methods more efficiently for future requests.

# 5 CONCLUSION AND FUTURE WORK

Combining meta-heuristics and cloud computing would allow a high number of organizations to leverage the opportunities of prescriptive analytics without obtaining dedicated resources or expert knowledge. While some concepts for an optimization-as-a-service exist, these are not discussing the workflow challenges of such a self-service. In order to achieve a high degree of automation, especially the smart selection and parametrization of methods should be the focus of future research. This would allow a provider to minimize operational costs and guaranteeing low services prices even for organizations without optimization capabilities.

## REFERENCES


Bosse, S., Splieth, M. and Turowski, K., 2016. Multi-Objective Optimization of IT Service Availability and Costs. *Reliability Engineering & System Safety*, 147, pp.142–155. Available at: http://www.sciencedirect.com/science/article/pii/S0951832015003312.

Charles, C.K., Taylor, C. and Keller, J., 2000. Meta-analysis: From data characterisation for meta-learning to meta-regression. In *Proceedings of the PKDD-00 workshop on data mining, decision support, meta-learning and ILP*.

Chern, M.-S., 1992. On the computational complexity of reliability redundancy allocation in a series system. *Operations Research Letters*, 11, pp.309–315.

Chi, D.-H. and Kuo, W., 1990. Optimal Design for Software Reliability and Development Cost. *IEEE Journal on Selected Areas in Communications*, 8(2), pp.276–282.

Coit, D.W. and Smith, A.E., 1996. Reliability Optimization of Series-Parallel Systems Using a Genetic Algorithm. *IEEE Transactions on Reliability*, 45, pp.254–266.

Evans, J.R. and Lindner, C.H., 2012. Business analytics: the next frontier for decision sciences. *Decision Line*, 43(2), pp.4–6.

Foster, I. et al., 2008. Cloud Computing and Grid Computing 360-Degree Compared. *2008 Grid Computing Environments Workshop*, abs/0901.0(5), pp.1–10.

García-Saiz, D. and Zorilla, M., 2017. A meta-learning based framework for building algorithm recommenders: An application for educational area. *Journal of Intelligent and Fuzzy Systems*, 32, pp.1449–1459.

Gill, P.E., Murray, W. and Wright, M.H., 1993. *Practical Optimization*, Academic Press.

Hoffmann, G.A., Salfner, F. and Malek, M., 2004. *Advanced Failure Prediction in Complex Software Systems*, Informatik-Bericht 172 der Humboldt-Universität zu Berlin.

King, J. and Magoulas, R., 2015. *2015 Data Science Salary Survey*, O'Reilly Media. Available at: https://duu86o6n09pv.cloudfront.net/reports/2015-data-science-salary-survey.pdf.

Kuo, W. and Prasad, V.R., 2000. An Annotated Overview of System-Reliability Optimization. *IEEE Transactions on Reliability*, 49(2), pp.176–187.

Kurschl, W. et al., 2014. Concepts and Requirements for a Cloud-based Optimization Service. In *Asia-Pacific Conference on Computer Aided System Engineering (APCASE)*.

Lanza, G., Haefner, B. and Kraemer, A., 2015. Optimization of selective assembly and adaptive manufacturing by means of cyber-physical system based matching. *CIRP Annals*, 64(1), pp.399–402.

Lins, I.D. and Droguett, E.L., 2009. Multiobjective optimization of availability and cost in repairable systems design via genetic algorithms and discrete event simulation. *Pesquisa Operacional*, 29, pp.43–66.

Marston, S. et al., 2011. Cloud Computing - The Business Perspective. In *44th Hawaii International Conference on System Sciences (HICSS)*.

Mell, P. and Grance, T., 2011. The NIST Definition of Cloud Computing. *National Institute of Standards and Technology - Special Publication*, 800-145, pp.1–3.

Müller, H., Bosse, S. and Turowski, K., 2016. Optimizing server consolidation for enterprise application service providers. In *Proceedings of the 2016 Pacific Asia Conference on Information Systems*.

Nahhas, A. et al., 2017. Metaheuristic and hybrid simulation-based optimization for solving scheduling problems with major and minor setup times. In A. G. Bruzzone et al., eds. *16th International Conference on Modeling and Applied Simulation (MAS)*. Rende, Italy.

Pfahringer, B., Bensusan, H. and Giraud-Carrier, C.G., 2000. Meta-Learning by Landmarking Various Learning Algorithms. In *17th International Conference on Machine Learning (ICML)*. Stanford, CA, USA, pp. 743–750.

Pimminger, S. et al., 2013. Optimization as a Service: On the Use of Cloud Computing for Metaheuristic Optimization. In R. Moreno-Diaz, F. Pichler, & A. Q. Arencibia, eds. *14th International Conference on Computer-Aided Systems Theory (EUROCAST)*. Lecture Notes in Computer Science. Las Palmas De Gran Canaria, Spain, pp. 348–355.

Pohl, M., Bosse, S. and Turowski, K., 2018. A Data-Science-as-a-Service Model. In *8th International Conference on Cloud Computing and Service Science (CLOSER)*.

Sadjadi, S.J. and Soltani, R., 2015. Minimum–Maximum regret redundancy allocation with the choice of redundancy strategy and multiple choice of component type under uncertainty. *Computers & Industrial Engineering*.

Silic, M. et al., 2014. Scalable and Accurate Prediction of Availability of Atomic Web Services. *IEEE Transactions on Service Computing*, 7(2), pp.252–264.

Soltani, R., 2014. Reliability optimization of binary state non-repairable systems: A state of the art survey. *International Journal of Industrial Engineering Computations*, 5, pp.339–364.