Simulation-Based Optimization in the Field of Physcial Internet

Christian Haider[1], Erik Pitzer[1] and Michael Affenzeller[1]
[1] Heuristic and Evolutionary Algorithms Laboratory
School of Informatics, Communications and Media
University of Applied Sciences Upper Austria, Hagenberg Campus, Austria
{christian.haider,erik.pitzer,michael.affenzeller}@fh-hagenberg.at

Abstract: *One of the biggest challenges in the field of Physical Internet (PI) is the optimization of material flows within global transportation networks. Within such real-world logistics networks, complex problems with many restrictions have to be considered. As such problems are highly dynamic, standard formulations and algorithms of already known problem models are difficult to apply. Thus, it is necessary to develop new methods which allow a stable optimization of PI logistics networks. One solution approach is simulation-based optimization. Realistic models of real-world environments are created to be able to consider all complex restrictions. It has to be decided how simulation and optimization nodes work together and communicate with each other. These considerations have to be made due to the required coupling process between the simulation and optimization.*

This paper shows how the optimization of a PI problem is interrelated with its simulation. It is demonstrated how simulation is used to evaluate possible solution candidates of the optimization process. Furthermore, it is presented how the simulation uses the optimization algorithms to generate new feasible candidates. The developed solution approach is realized by using the frameworks HeuristicLab and Easy2Sim. HeuristicLab is used for optimization and Easy2Sim for simulation parts. Moreover, it is presented how simulation and optimization parts communicate with each other. Therefore, an interface for the exchange of data between the simulation and the optimization pars is implemented. As a result, components can be programmed in different languages and different data structures can be used.

Keywords: *simulation, optimization, real-world problems, logistics networks, physical internet*

1 Introduction

Every day a lot of orders are placed through internet shops, retailers or distributors. Due to the huge amount of just in time orders, new methods concerning logistics transportation have to be developed to be able to still handle such orders in the future with available resources (Montreuil, et al., 2012).

Nowadays, different existing logistics networks are not fully interconnected with each other and logistics providers often use their own network to ship goods (Sarraj, et al., 2013). In contrary to that, with the so-called Physical Internet, also called PI, all autonomous transportation networks of logistics providers are merged into one big global network (Montreuil, et al., 2012).

To achieve such a global network, the idea of the internet is used as a model. Mainly the concepts of the TCP/IP protocol and Open Systems Interconnection Model (OSI) are applied to PI problems. As already pointed out, all these problems and new solution approaches are difficult to handle. Therefore, new solution methods for the successful realization of such a global network have to be developed (Montreuil, et al., 2012).

Since the field of physical internet is still a young topic, there are no ready to use solutions for most of the problems which can be applied. The PI represents a very large and difficult problem which should be modeled and solved, also in terms of logistics optimization problems. Due to the novelty of this topic, there is also no suitable optimization algorithm which can be used out of the box. There are examples, which show that optimization with metaheuristic methods can achieve good results concerning logistics optimization problems. One example where metaheuristic methods are applied successfully is the Vehicle Routing Problem (Dantzig and Ramser 1959, Toth and Vigo 2014). However, all these methods face only single aspects of the real world, whereas the PI model contains a lot of aspects that need to be taken care of. These aspects are restrictions such as time constraints, capacity constraints, etc. Therefore, standard metaheuristic approaches cannot be used to model the problem. For this reason, the aspect of simulation-based optimization comes into account. With simulation-based optimization, a meta-heuristic optimization framework is used to create new feasible solution candidates and a simulation-framework is used to evaluate all these candidates (Affenzeller, et al. 2015).

In this paper, we present the simulation-based optimization, which is suitable to solve such kinds of problems. The optimization framework HeuristicLab (Wagner et al 2014), which allows to couple many different optimization algorithms with the simulation framework easy2sim[1], is used. The easy2sim framework acts as an external evaluator for the given solution candidates.

One reason why simulation-based optimization is such a suitable concept is that simulation can use optimized parameters from the optimization process to update its model. Besides, the optimization can take use of a detailed model that evaluates the new solutions candidates, which it creates (Affenzeller, et al. 2015).

The paper is organized as follows. Section 2 gives an overview of different variants of simulation-based optimization and how they might be used in the PI context. In Section 3, the coupling between the optimization part and the simulation part is covered. Section 4 gives a conclusion about simulation-based optimization in the field of PI.

2 Simulation-Based Optimization

Due to the fact that modeling of the PI has many restrictions to be aware of, standard implementations of optimization algorithms and standard problem formulations cannot be applied. This is why simulation-based optimization approaches are discussed in the following section.

Within simulation-based optimization we can distinguish between:

- control optimization
- and parametric optimization (Affenzeller, et al. 2015 and Gosavi 2003)

In the following two subsections, the control optimization and parametric optimization are applied to possible implementations concerning the PI.

[1] http://www.easy2sim.at/

2.1 Low-Level Optimization

During a so-called low-level optimization, considerations regarding specific PI scenarios are made. Within these scenarios, different operational planning problems are optimized and evaluated. Examples for such operational planning problems are the Vehicle Routing Problem, the Lot Sizing Problem (LSP) and Scheduling Problems. The VRP calculates the cost optimal route between a given amount of customers, one or more depots and a set of transportation vehicles (Toth and Vigo 2014). With Lot Sizing Problems, the problem of minimizing costs, such as production costs or storage costs, by planning the optimal amount of to be produced products, the lot size, is described (Pahl and Voß 2010). Within Scheduling Problems, so-called production jobs are assigned to time critical resources (Fink and Voß 2003). Moreover, with low-level optimization, high-level scenarios can be evaluated, as explained subsequently in 2.2

Another designation for low-level optimization is control optimization or *dynamic optimization* (Gosavi 2003). As shown in Figure 1, the simulation part is the master process and the optimization process gets called if certain events occur. If an event is triggered, the optimization process is used to make a decision and this decision is sent back to the simulation (Affenzeller et al. 2015).

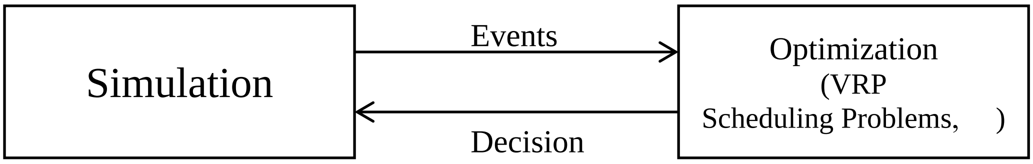

Figure 1: Control Optimization (Affenzeller et al. 2015)

For the PI, various operational planning problems have to be considered within one framework, as different organizations with different planning problems should work together within one framework. With the here described low-level and control optimization, the challenges of modeling already existing operational planning problems, such as the VRP, and also new ones, which could emerge via the further development of the PI, and at the same time validating results of necessary optimizations, should be conquered.

2.2 High-Level Optimization

While the low-level optimization covers the operational planning process, the high-level optimization is in charge of strategic and tactical planning processes. During such processes, different scenarios are considered. One example is the Plant Location Problem, which decides if new depots are opened or not under consideration of depot opening costs and delivery costs (Sridharan 1995). Another example is strategic or tactical production planning, where the allocation of t ’ r resources to be able to serve the needs of all customers, is covered (Boysen et al. 2009). On basis of these considerations (depot costs, fleet costs, construction costs, administrative expenses, etc.), the optimization potential of a specific PI scenario can be estimated and strategic (long-term) and tactical (mid-term) decisions can be made.

In Figure 2, the sch[redacted]llustration of high-level optimization is shown. Whereas simulation is the maste[redacted]ss at the low-level optimization, at the high-level optimization the optimization part is the master process. The optimization creates feasible solution candidates which are passed to the simulation. Inside the simulation process, parameters get evaluated. Moreover, a value representing the fitness of the solution candidates or even a set of values in case of multi-objective optimization, is returned (Affenzeller et al. 2015).

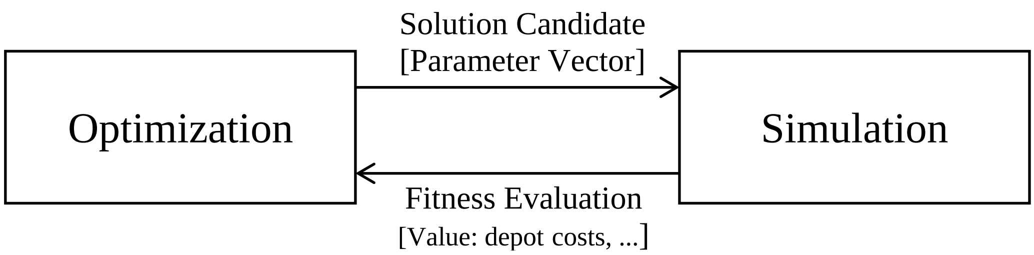

Figure 2: High-Level Optimization (Affenzeller et al. 2015)

3 Communication between Optimization and Simulation

As it comes to simulation-based optimization one challenge is to define an appropriate way to communicate between the optimization framework and the simulation environment. This is done by defining an interface that describes the way how both sides communicate with each other. As mentioned before, the HeuristicLab Framework is used on the optimization side. This offers the possibility to use a generic interface implementation (Affenzeller, et al. 2015) to communicate with external processes like the simulator easy2sim.

Whenever communication between different frameworks, programming languages or platforms is performed, the problem of interoperability has to be considered. For example, the HeuristicLab Framework is fully written in C# and the simulator framework easy2sim is Java code. To solve the problem of serializing structured data sending between the frameworks, , protocol buffer framework is used. Right now the current version of the Protocol Buffers is proto3. With this version it provides implementations for several different programming languages like Java, Python, Objective-C, Go, JavaNano, Ruby and C# and implementations for other languages are already planned. The reason why Protocol Buffers is used for the serialization of structured data is, because it performs a lot better than similar techniques like XML. The main reasons why we use protocol buffers over XML:[2]

- simplicity
- smaller size
- faster speed
- less ambiguous
- generated data access classes that are easy to use programmatically[2]

The most important reason why to use Protocol Buffers instead of XML for the communication between processes is its speed. As optimization processes are often time consuming, all possiblities for saving time have to be used. In Figure 3 a model in XML format is shown, in Figure 4 the same model is represented in the Protocol Buffers text format.[2]

```
<person>
    <name>John Doe</name>
```

[2] https://developers.google.com/protocol-buffers/

```
    <email>jdoe@example.com</email>
</person>
```

Figure 3: XML representation of a model person with two attributes name and email. This example is taken from googles developer site[3]

```
Person {
    name: "John Doe"
    email: "jdoe@example.com"
}
```

Figure 4: Protocol buffer representation of the model person with two fields name and email, taken from googles developer site[3]

In the human-readable format of the XML and Protocol Buffers format shown in Figures 3 and 4 there is no big difference in size, but if the Protocol Buffers format is translated into its binary format that is used over the wire, the size will shrink to approximately 28 bytes and will take about 100-200 nanoseconds to parse. Whereas the XML format has a size of at least 69 bytes what will take on the order of 5000-10000 nanoseconds to parse[3].

4 Conclusion

With already known problem formulations, only single aspects of the real world can be modeled. However, when speaking of the PI, modeling the real world is very complex and highly dynamic, as multiple planning problems have to be considered. This is where simulations-based optimization offers a huge benefit. It does not only allow to model a much more complex and closer model to the real world, it also allows to improve the model by taking advantage of optimized parameters from the optimization process. Two different patterns of simulation-based optimization are shown. The parametric approach where the simulation part of the simulation-based optimization, is used as an external evaluator and the optimization is the leading part, is the common way. The control based optimization pattern is another strategy for solving highly complex real-world problems with various restrictions.

The second part of the paper is focusing on the interfacing method between the optimization and simulation. It is very important that the coupling between these two parts of the simulation-based optimization is very tight. To connect both sides with each other the benefits of the optimization framework HeuristicLab are used, which offers a generic interface. For the data exchange and serialization process, the protocol buffers framework is presented. It is shown that the protocol buffer gives a lot of advantages to other formats for the serialization like XML. The biggest benefit here is timesaving and size reduction of the model representation.

5 References

Affenzeller, M., Beham, A., Vonolfen, S., Pitzer, E., Winkler, S.M., Hutterer, S., Kommenda, M., Kofler, M., Kronberger, G. and Wagner, S., 2015. Simulation-Based Optimization with HeuristicLab: Practical Guidelines and Real-World Applications. In *Applied Simulation and Optimization* (pp. 3-38). Springer International Publishing.

Boysen, N., Fliedner, M. and Scholl, A., 2009. Production planning of mixed-model assembly lines: overview and extensions. *Production Planning and Control*, *20*(5), pp.455-471.

[3] https://developers.google.com/protocol-buffers/docs/overview

Dantzig, G.B. and Ramser, J.H., 1959. The truck dispatching problem. *Management science*, 6(1), pp.80-91.

Fink, A. and Voß, S., 2003. Solving the continuous flow-shop scheduling problem by metaheuristics. *European Journal of Operational Research*, *151*(2), pp.400-414.

Gosavi, A., 2003. Simulation-based optimization. *parametric optimization techniques and reinforcement learning*.

Montreuil, B., Meller, R.D. and Ballot, E., 2013. Physical internet foundations. *In Service orientation in holonic and multi agent manufacturing and robotics* (pp. 151-166). Springer Berlin Heidelberg.

Pahl, J. and Voß, S., 2010. Discrete lot-sizing and scheduling including deterioration and perishability constraints. In *Advanced manufacturing and sustainable logistics* (pp. 345-357). Springer Berlin Heidelberg.

Sarraj, R., Ballot, E., Pan, S., Hakimi, D. and Montreuil, B., 2014. Interconnected logistic networks and protocols: simulation-based efficiency assessment. *International Journal of Production* Research, 52(11), pp.3185-3208.

Sridharan, R., 1995. The capacitated plant location problem. *European Journal of Operational Research*, 87(2), pp.203-213.

Toth, P. and Vigo, D. eds., 2014. *Vehicle routing: problems, methods, and applications*. Society for Industrial and Applied Mathematics.

Wagner S., Kronberger G. K., Beham A., Kommenda M., Scheibenpflug A., Pitzer E., Vonolfen S., Kofler M., Winkler S. M., Dorfer V., Affenzeller M., 2014. Architecture and Design of the HeuristicLab Optimization Environment. *Advanced Methods and Applications in Computational Intelligence* (Contributions to Book: Part/Chapter/Section 10), (Editors: R. Klempous, J. Nikodem, W. Jacak, Z. Chaczko) - Springer, pp. 197-261.