Expert Systems with Applications 42 (2015) 7292–7302

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Train routing in shunting yards using Answer Set Programming ☆

Jan-Alexander Adlbrecht *, Benno Hüttler, Nikola Ilo, Manfred Gronalt

University of Natural Resources and Life Sciences, Institute of Production and Logistics, Feistmantelstraße 4, A-1180 Vienna, Austria

ARTICLE INFO

Article history:
Available online 31 May 2015

Keywords:
Train routing
Shunting yard
Answer Set Programming
Optimisation

ABSTRACT

We use Answer Set Programming (ASP), a modern method for knowledge representation and reasoning, to build a fully automated decision support system for routing of trains in shunting yards. The system leverages the knowledge of experienced shunting yard operators and yields optimal, consistent and transparent routing decisions. In addition, the system remains easily adaptable to new expert knowledge that may become available in the future. We embedded this routing system into a simulation environment and conducted a study in order to investigate and confirm the validity and limits of this new approach. The study is based on the track layout and legal regularities of an actual shunting yard and therefore ensures the applicability to real world problem instances. The results confirm that ASP can be used to solve complex routing problems, but cannot yet match the solving speed of proprietary and custom fit algorithms. Therefore the suitability of ASP to solve complex routing problems is subject to the trade-off between transparency, adaptability and flexibility vs. speed.

1. Introduction

Shunting yards serve the purpose of reassembling cargo trains at critical nodes of the international railway track network. They are highly complex and dynamic systems that offer a huge variety of planning and optimisation problems, which are presented by Boysen, Fliedner, Jaehn, and Pesch (2012). Our research group developed a simulation testbed for shunting yards called SimShunt that is able to simulate all processes that are required to operate a shunting yard (Hüttler & Gronalt, 2015). The simulation engine relies on a number of elaborate decision support systems. The system presented in this paper serves as such a system.

A classical shunting yard consists of four major areas. They are usually passed by a rail car in the following order: receiving yard (RY), hump (H), classification bowl (CB) and the departure yard (DY). The receiving yard is used for pre-processing and disassembling incoming cargo trains. Then the rail cars are pushed over the hump by a shunting engine. The hump enables the rail cars to travel into the classification bowl without external propulsion by following a system of automated switches. The departure yard is used for delaying and post-processing of reassembled trains. The sorting bowl (SB) is an optional, but not uncommon extension. It is used to build trains with multiple restrictions that cannot be efficiently met by the default process.

Almost all processes require movements of trains or shunting engines on the track network. Finding these routes on the track network of a shunting yard is a non-trivial task that is usually semi-automated but ultimately subject to manual decision making. Valid routes have to respect a large number of heterogonous constraints and are therefore hard to find. Operators have to rely on their extensive experience in order to make good routing decisions. Due to the subjective nature of this approach, the reasons that lead to a particular routing decision are non-transparent and often irreproducible. The quality of routing decisions has a large impact on the overall yard efficiency, therefore a routing system that can propose high quality decisions is useful and important. In addition to being able to handle all problem instances in a reasonable amount of time, such a system has to be flexible and easily maintainable. This shifts the focus from a purely functional black-box to a transparent and descriptive system design, preferably in a standardised way that allows professionals to share and discuss their respective sets of rules. Therefore such a system enables professionals to evolve their routing system over time and adapt to events that impact routing decisions such as malfunctioning switches without any knowledge about optimisation algorithms and solving strategies, which makes it much more useful in practice.

Finding optimal routes of shunting trains is a shortest path problem with multiple domain-specific constraints. Most of them

☆ Funding was provided by the Austrian Federal Ministry for Transport, Innovation and Technology by means of The Austrian Research Promotion Agency (FFG) Grant No. 835777.

* Corresponding author.

E-mail addresses: j.adlbrecht@boku.ac.at (J.-A. Adlbrecht), benno.huettler@boku.ac.at (B. Hüttler), nikola.ilo@boku.ac.at (N. Ilo), manfred.gronalt@boku.ac.at (M. Gronalt).

http://dx.doi.org/10.1016/j.eswa.2015.05.044

are clearly structured and easy to implement in any modelling language or routing algorithm. An example for such a constraint would be the very common flow conservation constraint for network flows in linear programmes. Unstructured constraints on the other hand are much more complicated and deal with exceptions and personal experience of decision makers. Modelling these constraints is often verbose and almost always results in very large models and computer codes that quickly become error prone and hard to maintain, hence undesirable. In addition, some of the constraints restrict certain properties of a route which turn the relatively simple task of finding a shortest path on a network graph into a NP-complete search problem. Therefore the modelling system has to offer both ease and clarity of modelling as well as the availability of powerful solvers. The validity of this argument is supported by the comparison to the fairly recent contribution of Riezebos and van Wezel (2009) which relies on a very complex and hard to understand Mixed Integer Programme to solve a similar problem.

Based on these presented problem aspects and requirements to the decision support system, we decided to investigate the feasibility and suitability of modern methods in the domain of knowledge representation and reasoning. More specifically, we decided to try the newly emerging declarative modelling system Answer Set Programming (ASP). ASP features a rich set of language elements and modelling assumptions that are useful to describe the routing problem. It also features a complete separation of model description and model solving which meets the requirements for transparency and adaptability.

This paper is structured in the following fashion: The next section is dedicated to embedding our work in the context of related publications in the domains of shunting yards and knowledge representation and reasoning. We especially emphasise the foundations of ASP and its applications in recent years to build decision support systems. Then we give a detailed description of our routing problem and discus all aspects and constraints. The next section provides an introduction to the ASP modelling language and syntax, but focuses on language elements that we used in our model. Then we present the first part of the ASP routing model. This first part is a generalised model capable of finding shortest paths on a directed weighted graph. In the next section this generalised model is elaborated and we introduce all constraints specific to routing in shunting yard that are presented in the problem description section. Then we present our numerical study that investigates the validity of the approach as well as the performance of a modern ASP solver and the impact of certain rules thereon. We compose 6 models based on different amounts of expert knowledge of the presented rules and solve a number of randomly generated problem instances. The final part of this paper is dedicated to presenting our findings and discussing future research topics.

2. Related work

Routing problems are a very well-studied field with a great number of related publications. Algorithms for vehicle routing are often applied to make transportation processes more efficient. In recent years the motivation for utilising optimisation techniques has started to shift from purely economic to sustainable or "green" as reported by Lin, Choy, Chung, and Lam (2014). They also present an extensive survey of recent publications on the subject. In the field of sustainable transportation systems the role of railway based transportation is very important. Piu and Speranza (2013) present a survey of recent works dealing with routing and scheduling problems in the railway context. They also propose a classification scheme for railway routing applications for better orientation in this huge field. One of these classes is called "Yard switching and in-plant railroad LAP" (Locomotive Assignment Problem). Our routing problem can be assigned to this class.

Shunting yards offer a large number of planning and optimisation problems. Boysen et al. (2012) provide a well-structured overview and presentation of these problems. Jaehn, Rieder, and Wiehl (2014) study one of those planning problems and also explain how these planning problems relate to the primary shunting process on the tactical level. The problem of routing trains within a shunting yard is an operational problem and is presented by Hüttler and Gronalt (2015). The model and concepts presented in Hüttler and Gronalt (2015) are the basis for this paper. We use their network graph and shunting rules to build an ASP-based routing system.

One of the most similar contributions to our planning problem is provided by Riezebos and van Wezel (2009). They propose a special network graph design and a complex mixed integer programme to tackle the problem of routing trains in shunting yards and finding k-shortest paths. In comparison to this paper it becomes quite clear how fundamentally different our declarative approach is. They use a special routing graph in combination with a complex Mixed Integer Programme, whereas our approach relies on a list of rules that capture expert knowledge in a formalised yet readable language.

Logic programmes are traditionally an excellent tool to solve computationally hard planning problems, as confirmed by Dovier, Formisano, and Pontelli (2007). They provide planning, scheduling and optimisation problems as examples. Train routing in shunting yards is a combination thereof. The planning part covers the determination of a valid route under multiple constraints and the optimisation part deals with the selection of possible routes in order to find the most desirable one. Another feature of ASP is its strict declarative approach. This enables a complete separation of the model and the solving process. Therefore it is possible to focus on the problem itself without considering the impact of modelling choices on the ability of a solver to find a solution.

ASP was first introduced by Lifschitz in 2002. From Lifschitz (2002) can be learned that ASP is based on the principles of non-monotonic reasoning and features "negation as failure" (NAF) under the stable model semantics. Non-monotonic reasoning is designed to capture aspects of human common-sense reasoning, according to Dimopoulos, Nebel, and Koehler (1997). The main aspect is the withdrawal of previous conclusions if new information becomes available. This feature is realised by a special form of negation called "negation as failure". In its most basic form an answer set programme consists of facts and rules that produce new facts. There may be cases where certain facts can neither be confirmed nor denied by the rules and facts provided. In this case it is assumed, that the fact in question is denied. If the programme is extended by some new rule or fact, this fact may become deductible and may lead subsequently to different conclusions. In the context of ASP, stable models are called answer sets. In general an answer set consists of all positive facts that jointly satisfy all constraints. There are as many answer sets as possible combinations thereof. Dovier et al. (2007) state very concisely that "in ASP, each problem is modelled as a collection of rules, in such way that the solution to the problem corresponds one-to-one to the answer set of the program".

Brewka, Eiter, and Truszczynski (2011) provide an excellent introductory paper to ASP that is particular helpful for readers that are not yet familiar with this method. They present many language features and develop simple example programmes to present their respective application. As an indicator for the increasing popularity and maturity of ASP, the ASP-syntax has recently been standardised by the ASP Standardization Working Group in Calimeri et al. (2012). We use this standard throughout our publication. The latest version of the popular ASP solver clingo that we use for our numerical study already adheres to most of the standardised