Solving Train Load Planning Problems with Boolean Optimization

Lorenzo Foti, Marco Maratea, Simona Sacone, and Silvia Siri

DIST, University of Genova
lorenzo.foti@studenti.unige.it,
{marco.maratea,simona.sacone,silvia.siri}@unige.it

Abstract. The train load planning problem is very important in the context of transportation and logistics in seaport container terminals. It allows to optimally assign containers to wagons of a train in order to satisfy capacity constraints while optimizing re-handling operations and train utilization. In this paper, we first present a basic mathematical model of the problem, inspired from a real case of an Italian port. Then, we extend the basic model in order to more realistically represent the re-handling operations in the storage area. We finally present the results obtained by running several Boolean optimization, i.e. LP, PB, and SMT solvers, on a set of benchmarks coming from the two models. A preliminary experimental analysis shows that (i) challenging problems can be generated even with a relatively low number of containers and wagons, that in our models correspond to relatively small formulas; (ii) CPLEX shows the best results, and (iii) the only other solver evaluated that performs well in this domain is SCIP.

1 Problem description and formulation

The need for optimization in container terminals has become more and more important in recent years; among the different planning problems arising in a terminal, an important decision aspect is the train load planning problem, especially for those terminals characterized by high rail traffic volumes [17, 16]. This is a peculiar problem, basically different from other loading problems (for instance the ship stowage problem that presents completely different characteristics), because the weight constraints and the loading operations are specific for rail wagons. In the literature, there are some papers dealing with the train load planning problem in which mathematical models are proposed for the optimal assignment of containers to wagons but without taking into account the real weight restrictions [2, 4, 15] (often simply a maximum weight constraint for each wagon is considered in the related problems). In this paper, instead, the proposed model considers explicitly the real weight constraints for wagons, as also done in [3]. In comparison with [3], in the present paper different cost terms and container types are considered and, first of all, re-handling operations in the storage areas are explicitly modeled; this is an important aspect to be considered in real applications since the

Proceedings of the 19^{th} RCRA workshop on *Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion* (RCRA 2012).
In conjunction with AI*IA 2012, Rome, Italy, June 14–16, 2012.

minimization of re-handling operations is a crucial issue for the optimal management of storage areas.

The problem under investigation regards the definition of the loading plan for one train at a time, given a set of containers stocked in the storage area characterized by different weights, lengths and commercial values. Each wagon of the train has a specified length and some weight constraints coming from real cases. A model is defined in order to find the optimal assignment of containers to each slot of the train wagons in order to satisfy length and weight constraints while maximizing the train utilization and minimizing the re-handling operations in the storage area. Moreover, in this work the train is supposed to be loaded sequentially by cranes. Given that in our setting traffic volumes by rail are not so high and there is one crane loading a train at a time, it is not possible that more trains are loaded at the same time, and thus it is reasonable to generate a different plan for each train. Let us introduce the following notation:

- $\mathcal{C}$ is the set of containers in the storage area and $\mathcal{W}$ is the set of wagons;
- ω_i is the weight of container $i \in \mathcal{C}$ and π_i is the penalty paid if container i is not loaded, taking into account the urgency and commercial value of the container (it is important to consider different penalties for the containers in order to take into account their different urgency and commercial priorities);
- $\gamma_{i,j}$, $i, j \in \mathcal{C}$, $i \neq j$, indicates the relative position between container i and j in the storage area; in particular, $\gamma_{i,j} = 1$ means that container i is located below j, $\gamma_{i,j} = 0$ otherwise;
- $\overline{\Omega}_w$ and $\overline{\Omega}$ are, respectively, the weight capacity of wagon $w \in \mathcal{W}$ and of the train;
- $\mathcal{S}$ is the set of slots; $\mathcal{S}_w$ is the set of possible slots for wagon w and, analogously, w_s indicates the wagon including slot s;
- $\mathcal{B}_w$ is the set of weight configurations for wagon w and $\delta_{b,s}$ is the maximum weight for slot s in the weight configuration b;
- α is the unitary re-handling cost in the storage area and T is the maximum number of tiers.

In the notation it is assumed that the indexes associated with wagons describe their position along the train; since the train is supposed to be loaded sequentially, wagons with a smaller index are loaded before those with a higher index. The same applies for the indexes of the slots which indicate their position in the train.

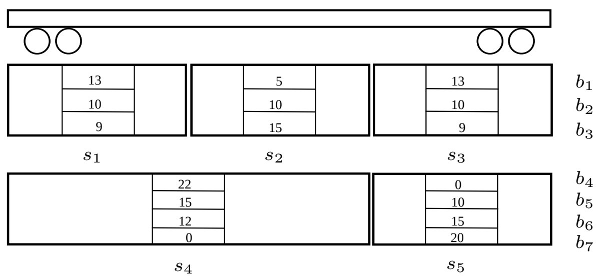

Fig. 1. Sketch of wagon weight restriction.

In order to better clarify the meaning of the real weight constraints, consider the example sketch provided in Figure 1: there are two possible load configurations, i.e. it can be loaded either with three 20’ containers or with one 20’ container and one 40’ container. Considering for example the first load configuration, there are 3 slots, i.e. s_1, s_2 and s_3, and three different weight configurations, b_1, b_2 and b_3 (for instance if the first weight configuration is chosen, the three slots can be loaded with maximum weights $\delta_{b_1 s_1} = 13$, $\delta_{b_1 s_2} = 5$ and $\delta_{b_1 s_3} = 13$, respectively).

The decision variables are the following:

- $x_{i,s} \in \{0,1\}$, equal to 1 if container i is assigned to slot s;
- $t_{w,b} \in \{0,1\}$, equal to 1 if weight configuration b is chosen for wagon w;
- $y_{i,w} \in \{0,1\}$, equal to 1 if container i is re-handled (i.e. it is moved but not assigned) when wagon w is loaded.

It is worth noting that the $x_{i,s}$ variables are defined only if the assignment of containers to slots is feasible as regards the fitting of the container length with the slot length; if for instance the considered container is 40’ long and the slot is 20’, the corresponding variable is not defined at all. Herewith we present a first model for the train loading problem.

$$\min \ \alpha \cdot \sum_{i \in \mathcal{C}} \sum_{w \in \mathcal{W}} y_{i,w} + \sum_{i \in \mathcal{C}} \pi_i \cdot \left(1 - \sum_{s \in \mathcal{S}} x_{i,s} \right) \tag{1}$$

s.t.

$$\sum_{s \in \mathcal{S}} x_{i,s} \leq 1 \qquad \forall i \in \mathcal{C} \tag{2}$$

$$\sum_{i \in \mathcal{C}} x_{i,s} \leq 1 \qquad \forall s \in \mathcal{S} \tag{3}$$

$$\sum_{b \in \mathcal{B}_w} t_{w,b} = 1 \qquad \forall w \in \mathcal{W} \tag{4}$$

$$\sum_{i \in \mathcal{C}} \omega_i \cdot x_{i,s} \leq \sum_{b \in \mathcal{B}_{w_s}} \delta_{b,s} \cdot t_{w,b} \qquad \forall s \in \mathcal{S} \tag{5}$$

$$\sum_{i \in \mathcal{C}} \sum_{s \in \mathcal{S}_w} \omega_i \cdot x_{i,s} \leq \overline{\Omega}_w \qquad \forall w \in \mathcal{W} \tag{6}$$

$$\sum_{i \in \mathcal{C}} \sum_{s \in \mathcal{S}} \omega_i \cdot x_{i,s} \leq \overline{\Omega} \tag{7}$$

$$\sum_{j \in \mathcal{C}: \gamma_{j,i} = 1} \sum_{s \in \mathcal{S}_w} x_{j,s} \leq (T-1) \cdot \left(y_{i,w} + \sum_{r \in \mathcal{S}_h : h < w} x_{i,r} \right) \qquad \forall i \in \mathcal{C} \quad \forall w \in \mathcal{W} \tag{8}$$

$$x_{i,s} \in \{0,1\} \qquad \forall i \in \mathcal{C} \quad \forall s \in \mathcal{S} \tag{9}$$

$$t_{w,b} \in \{0,1\} \qquad \forall w \in \mathcal{W} \qquad \forall b \in \mathcal{B}_w \tag{10}$$

$$y_{i,w} \in \{0,1\} \qquad \forall i \in \mathcal{C} \quad \forall w \in \mathcal{W} \tag{11}$$

The cost function (1) to be minimized includes a term for re-handling operations and a term for the penalty of not loading containers (this term allows to maximize the train load). It is worth noting that the costs $\pi_i,\ i \in \mathcal{C}$, are always much higher than the cost α, in order to assure that the train is loaded as much as possible. Constraints (2) and (3) ensure that each container is assigned at most to one slot and that in each slot no more than one container is loaded. Constraints (4) impose that for each wagon only one weight configuration is chosen, whereas (5), (6) and (7) impose that the maximum weight conditions are respected for slots, wagons and the train, respectively. Constraints (8) guarantee that re-handling operations are correctly defined: container i is re-handled (then $y_{i,w}$ must be equal to 1) if container j (stocked below i) is loaded on a wagon w and container i is not loaded on any wagon h with $h < w$.

In this model, the definition of the $y_{i,w}$ variables is associated with wagons but this is a limitation since it could be better, instead, to associate these variables to slots. In other words, the use of $y_{i,w}$ variables is generally suitable for the train load planning but it is not completely precise. In fact, it does not correctly evaluate the re-handling operations in case container j (stocked below i) is loaded on the same wagon in which container i is loaded but in a slot with a smaller index (i.e. a slot that is loaded before): in such a case, the re-handling operation takes place in the reality but it is not correctly modeled in the optimization problem. Hence, we have defined an extended version of the model presented above in order to represent more precisely the re-handling movements, by adopting a different set of decision variables $z_{i,s},\ s \in \mathcal{S}$, instead of $y_{i,w},\ w \in \mathcal{W}$. In the new model, cost function (1) and constraints (8) are substituted, respectively, by the following:

$$\min\ \alpha \cdot \sum_{i\in\mathcal{C}} \sum_{s\in\mathcal{S}} z_{i,s} + \sum_{i\in\mathcal{C}} \pi_i \cdot \left(1 - \sum_{s\in\mathcal{S}} x_{i,s}\right) \tag{12}$$

$$\sum_{j\in\mathcal{C}:\gamma_{j,i}=1} x_{j,s} \leq z_{i,s} + \sum_{r\in\mathcal{S}:r<s} x_{i,r} \qquad \forall i \in \mathcal{C} \quad \forall s \in \mathcal{S} \tag{13}$$

In this way, container i is re-handled (then $z_{i,s}$ must be equal to 1) if container j (stocked below i) is loaded on a slot s and container i is not loaded on any slot r with $r < s$. Obviously, this extended model captures the real-life situation more precisely, but with an increase in the size of the problem, in terms of number of variables and constraints.

2 Implementation, Benchmarks and Experiments

We generated instances in 0-1 Linear Programming (LP), Pseudo-Boolean (PB) and Satisfiability Modulo Theories (SMT) ver. 1 formats[1] of the two models, and organized

[1] See, e.g. `http://cgm.cs.mcgill.ca/~avis/courses/567/cplex/reffileformatscplex.pdf`, `http://www.cril.univ-artois.fr/PB11/format.pdf` and `http://goedel.cs.uiowa.edu/smtlib/docs.html` for 0-1 LP, PB and SMT formats, respectively.

the benchmarks into four groups A-D of 10 instances each: the number of containers and wagons can be found in the second and third rows of Tables 1, respectively. We considered real data (referring to a container terminal in the North of Italy) about the characteristics of wagons, the weight constraints, the storage area, whereas we randomly generated the characteristics of containers (e.g. length, weight, percentage of 20' and 40' containers) and the train composition. The fourth and fifth (resp. sixth and seventh) columns of Table 1 report the mean number of variables and constraints for the LP instances of the first (resp. extended) model.

Table 1. Characteristics of the groups of instances.

Global setting			First model		Extended model	
Instance group	#containers	#wagons	#variables	#constraints	#variables	#constraints
A	20	10	941	345	1909	1337
B	30	10	1352	458	2884	2050
C	30	15	1977	670	4605	3151
D	40	15	2593	829	5855	4098

We evaluated a number of Boolean optimization solvers on the resulting instances, featuring not only different formats but, more importantly, different techniques and heuristics for solving the resulting optimization problems. The solvers that we considered are CPLEX ver. 12[2], the PB solvers WBO [13] ver. 1.6, MINISAT+ [6] ver. 1.14, PBCLASP (based on the Answer Set Programming system CLASP [8] ver. 1.3.6), BSOLO ver. 3.0.17 [12], GLPPB ver. 0.2 (by the same authors of PUEBLO [14]) and SCIP ver. 2.0.2 [1], and the SMT solvers YICES [5] ver. 1.0 and HYSAT ver. 0.8.6 [7] via the QF_LIA logic of SMT-LIB[3], augmented with optimization features.[4] The timeout has been set to 1200s and the memory limit to 500MB on a Linux box equipped with a Pentium IV 3.2GHz processor and 512MB of RAM.

The first observation is that no solver other than CPLEX or SCIP was able to solve any of the generated instances in the allotted time. This is quite surprising given that the instances contain a relatively low number of variables and constraints, as showed in Table 1. The results of CPLEX and SCIP are shown in Table 2, where the first column is the instance group, the second and the third columns report the results for the first and extended models, respectively, and are further divided into two columns, one for CPLEX and one for SCIP. Results of each solver are presented in the form $x(y)$ where x is the mean solving time of the y solved instances out of 10, as customary, e.g. in Max-SAT

[2] Through the IBM Academic Initiative at `https://www.ibm.com/developerworks/university/academicinitiative/`.

[3] `http:www.smtlib.org`.

[4] Solvers have been downloaded from `http://www.minisat.se/MiniSat+.html`, `http://www.eecs.umich.edu/~hsheini/pueblo`, `http://www.csi.ucd.ie/staff/jpms/soft/soft.php`, `http://scip.zib.de/`, `http://yices.csl.sri.com/download.shtml`, `http://hysat.informatik.uni-oldenburg.de/26273.html`, or made available by authors.

evaluations[5]. We can see that CPLEX is the best solver on these problems: it solves the highest number of instances in all instance groups, and in the shortest time. CPLEX is in fact the most used system in the field of transportation and logistics to solve linear programming problems. Despite this, we can nonetheless note that SCIP performs quite well, having results sometimes close to CPLEX, thus confirming its good results in recent PB Competitions. The poor performances of SMT solvers can be explained by the fact that they are not tuned for optimization problems and can solve much more expressive problems than the ones considered in our work. Finally note that the instances of group B (resp. D) are in general easier the the ones of group A (resp. C), even if the instances are bigger: this is because, having the number of wagons fixed, more containers allow for more options for loading each wagon.

Table 2. First and extended models: mean CPU time of solved instances and number of solved instances (in parenthesis) for CPLEX and SCIP.

Instance group	First model		Extended model	
	CPLEX	SCIP	CPLEX	SCIP
A	106.82(10)	116.86(8)	50.23(9)	164.17(9)
B	2.5(10)	75.9(10)	3.96 (10)	72.4(10)
C	68.47(8)	165.51(6)	230.42(3)	440.98(1)
D	15.37(9)	424.31(6)	285.23(6)	294.53(3)

3 Conclusions and Future Work

We have presented two mathematical models for the train load planning problems, inspired by a real-world scenario. Then, we have expressed these models as Boolean optimization problems, and we have run several solvers, relying on different formats, optimization techniques and heuristics on these problems. Our analysis shows that (i) relatively small but difficult benchmarks can be generated, and (ii) only CPLEX and SCIP can effectively handle the generated instances: while CPLEX is the best, SCIP can have good performance as well.

About future work, on the modeling side we plan to further extend the model by considering, on the one hand, a more detailed treatment of the movements and localizations in the storage area and, on the other hand, more complex train loading policies. More specifically, an important evolution of the model regards the loading plan of more trains with different destinations (for which it is necessary to know the destinations of containers and, in some cases, compulsory assignments of some containers to some trains, due to specific requests of freight forwarders). Another interesting direction in extending the proposed model regards the definition of loading plans in case trains are loaded by more than one crane at the same time or, in some other cases, one crane loads two trains contemporaneously. On the solving side, we first would like to evaluate other Boolean optimization solvers, possibly taken from the up-coming Evaluations and

[5] See, e.g., `http://maxsat.ia.udl.cat` for the last.

Competitions of interest, by testing all available solvers also in the new models that we have outlined above. Then, given that our ultimate goal is to strengthen as much as possible the efficient solution of these problems, if also the new benchmarks confirm the superior performance of CPLEX, we plan to investigate the use of automatic algorithm configuration framework for tuning the performance of CPLEX on our domain: results on SAT and Mixed Integer Linear Programming problems on other domains, e.g. [11, 10, 9], are very promising.

The instances generator and the scripts to the various formats, together with other materials, can be found at:

`http://www.star.dist.unige.it/~marco/TLPP/.`

References

1. T. Achterberg, T. Berthold, T. Koch, and K. Wolter. Constraint integer programming: A new approach to integrate CP and MIP. In L. Perron and M. A. Trick, editors, *Proc. of the 5th International Conference Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2008)*, volume 5015 of *Lecture Notes in Computer Science*, pages 6–20. Springer, 2008.
2. N. Bostel and P. Dejax. Models and algorithms for the container allocation problem on trains in a rapid transshipment yard. *Transportation Science*, 32:370379, 1998.
3. F. Bruns and S. Knust. Optimized load planning of trains in intermodal transportation. *Accepted to OR Spectrum*, 2010.
4. P. Corry and E. Kozan. Optimised loading patterns for intermodal trains. *OR Spectrum*, 30:721750, 2008.
5. B. Dutertre and L. De Moura. A fast linear-arithmetic solver for DPLL (T). In T. Ball and R. B. Jones, editors, *Proc. of the 18th International Conference on Computer Aided Verification (CAV)*, volume 4144 of *Lecture Notes in Computer Science*, pages 81–94. Springer, 2006.
6. N. Eén and N. Sörensson. Translating pseudo-Boolean constraints into SAT. *Journal on Satisfiability, Boolean Modeling and Computation*, 2:1–26, 2006.
7. M. Franzle, C. Herde, T. Teige, S. Ratschan, and T. Schubert. Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure. *Journal on Satisfiability, Boolean Modeling and Computation*, 1:209–236, 2007.
8. M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. Conflict-driven answer set solving. In M. M. Veloso, editor, *In Proc. of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 386–392. Morgan Kaufmann Publishers, 2007.
9. H. H. Hoos. Programming by optimization. *Communications of the ACM*, 55(2):70–80, 2012.
10. F. Hutter, H. H. Hoos, and K. Leyton-Brown. Automated configuration of mixed integer programming solvers. In A. Lodi, M. Milano, and P. Toth, editors, *Proc. of the 7th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2010)*, volume 6140 of *Lecture Notes in Computer Science*, pages 186–202. Springer, 2010.
11. F. Hutter, H. H. Hoos, K. Leyton-Brown, and T. Stützle. ParamILS: An automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36:267–306, 2009.
12. V. M. Manquinho and J. P. Marques-Silva. On using cutting planes in pseudo-Boolean optimization. *Journal on Satisfiability, Boolean Modeling and Computation*, 2:209–219, 2006.

13. V. M. Manquinho, J. P. M. Silva, and J. Planes. Algorithms for weighted Boolean optimization. In O. Kullmann, editor, *Proc. of the 12th International Conference on Theory and Applications of Satisfiability Testing (SAT 2009)*, volume 5584 of *Lecture Notes in Computer Science*, pages 495–508. Springer, 2009.
14. H. M. Sheini and K. A. Sakallah. Pueblo: A modern pseudo-boolean sat solver. In *Proc. of the Design, Automation and Test in Europe Conference and Exposition (DATE)*, pages 684–685. IEEE Computer Society, 2005.
15. W. Souffriau, P. Vansteenwegen, G. Berghe, and D. V. Oudheusden. Variable neighbourhood descent for planning crane operations in a train terminal. In *Metaheuristics in the Service Industry*, volume 624 of *Lecture Notes in Economics and Mathematical Systems*, pages 83–98. Springer Berlin Heidelberg, 2009.
16. R. Stahlbock and S. Voss. Operations research at container terminals: a literature update. *OR Spectrum*, 30:1–52, 2008.
17. D. Steenken, S. Voß, and R. Stahlbock. Container terminal operation and operations research - a classification and literature review. *OR Spectrum*, 26:3–49, 2004.