Modeling a Lot-Aware Slab Stack Shuffling Problem

Judith Fechter[1,2], Andreas Beham[1,2], Stefan Wagner[1], and Michael Affenzeller[1,2]

[1]Heuristic and Evolutionary Algorithms Laboratory, School of Informatics, Communications and Media, University of Applied Sciences Upper Austria, Softwarepark 11, 4232 Hagenberg, AUSTRIA, {judith.fechter,andreas.beham,stefan.wagner,, michael.affenzeller}@fh-hagenberg.at

[2]Institute for Formal Models and Verification, Johannes Kepler University Linz, Altenberger Straße 69, 4040 Linz, Austria

February 13, 2018

Technical Report[1]

Abstract

Stacking and shuffling problems are key logistics problems in various areas such as container shipping or steel industry. The aim of this paper is motivated by a real world instance arised in steel production. Slabs, continously but randomly casted, need to be arranged for transport while having a certain number of buffer stacks available. The optimization problem arising is assigning transport lotnumbers, regarding properties of slabs, as well as minimizing shuffling movements while arranging the slabs, regarding the implicitly given transport order. For that purpose, a combined optimization problem, being composed of two sub-problems is developed. Further computational studies are conducted in order to investigate the complexity of the problem. A combined solution approach is developed solving the problem in a sequential way using customized algorithms in order to make advantage of specialised algorithms.

1 Introduction

The problem of stacking and restacking plays an important role in production and logistics, arising in various areas such as container terminals and steel production. The former addresses the shuffling of containers within a yard where

[1]This technical report represents material published in *Computer Aided Systems Theory – EUROCAST 2015. Lecture Notes in Computer Science, vol 9520.* The original source of the publication is available at `https://link.springer.com/chapter/10.1007/978-3-319-27340-2_42`. The final publication is available at `link.springer.com`.

containers are stacked temporarily. [5] developed a simulated annealing (SA) algorithm for finding stacking strategies. Results show that the number of shuffling movements is reduced compared to a strategy grouping same-weight-containers. In an earlier work [2] considered two stacking strategies. One aims to keep all stacks at an equal height. The other one locates containers according to their arrival time. The results show that the measures of a bay most influence the expected number of shuffling movements. In [9] a comprehensive literature review of operations, space allocation, yard layout and stacking logistics is provided.
Another approach for optimal stacking is the pre-marshalling problem. It considers the optimal shuffling of blocks that are assumed to be of the same size. The delivery order is assumed to be known. [6] propose a method for finding an optimal strategy for shuffling containers by decomposing the problem into two stages and three subproblems. They use dynamic programming and the transportation problem technique, which overall take a considerable computational time. In [8] a mathematical model is proposed for the shuffling while considering a given loading sequence as well as a given yard layout. They present an integer programming model based on a multi commodity flow problem. A simple heuristic is proposed in order to solve problems close to real-world size.
Considering steel industry, most research work has been done on the Slab Stack Shuffling (SSS) problem which aims to choose appropriate slabs for the rolling schedule out of ponderous stacks of slabs while minimizing the shuffling movements needed. [11] propose an integer programming model for the SSS problem and developed a two-phase heuristic algorithm. In [12] a genetic algorithm using specially designed operators is presented. For the SSS problem considered in [12], [10] propose an improved parallel genetic algorithm. Two operators are developed, namely a modified crossover operator and a kin selection operator. [11], [12] and [10] consider that a lifted slab is moved back immediately after the required slab is taken out. [13] takes into account that lifted slabs can be placed on other stacks. [3] propose a linearization of the SSS problem solved by using a linear binary integer programming (BIP) model.

1.1 Motivation

Considering the hot storage area, slabs are continously casted and then lifted to buffer stacks to allow for an arranging process, while the number of stacking locations is limited. Subsequently the slabs are lifted to a delivery point to be picked up by transportation facilities. As there is only a limited number of vehicles, it is essential that slabs are carried to the delivery point in assorted lots. The slabs should be arranged according to measurements and properties.
[7] propose an exact method for solving a stacking problem allowing moves from and to a buffer stack and to a target stack with the aim of minimizing shuffling movements. They use ideas from dynamic programming and discrete optimization and obtain a solution quality of 25% off optimum. Closely related to this problem is the blocks relocation problem (BRP). [1] provide a formal analysis of the problem and developed a heuristic based upon a set of relocation rules.
In contrast to the above mentioned research work, we consider the delivery order not as predefined but as flexible. We developed a combined optimization problem composed of two subproblems: a lot-building problem which provides

the transport lots and implicitly the delivery order and a stacking problem which provides the optimal stacking in order to minimize shuffling movements. Further experimental studies are performed and a combined solution approach is proposed. In section 2 a description and formulation of both problems is provided. In section 3 results on complexity and performnace and a sequential solution approach are presented.

2 Problem Description

The lot-aware slab stack shuffling problem is composed of a lot-building problem and a stacking problem. This section provides a formulation of both problems as well as of the interrelation on each other. The lot-building problem shall provide the transport lot number as input for the stacking problem (see equation 8).

2.1 Lot-Building Problem

Input A set of slabs with properties (weight, measures, type, production time).
Objective The aim is to minimize the number of lots used.
Constraints A lot may only has a maximum weight and minimum number of slabs. Further, one lot may only contains one type of slab and only slabs with certain measurements limits.

$\Omega = \{j\}_{j=1}^{N}$	Set of cast slabs
$\mathcal{L} = \{i\}_{i=1}^{L}$	Set of lots
w_j, t_j, b_j	Weight, production time, measures (width) of slab j
W, T, B	Maximum values of weight, difference of production time and measures per lot

$$\min \sum_{i=1}^{L} y_i \tag{1}$$

$$s.t. \quad x_{ij} \leq y_i, \qquad \forall i \in L, j \in N \tag{2}$$

$$\sum_{i=1}^{L} x_{ij} = 1, \qquad \forall j \in N \tag{3}$$

$$\sum_{j=1}^{N} x_{ij} \geq 2y_i, \quad \sum_{j=1}^{N} x_{ij} w_j \leq W y_i \qquad \forall i \in L \tag{4}$$

$$(t_j - t_l) x_{ij} x_{il} \leq T, \quad (b_j - b_l) x_{ij} x_{il} \leq B \qquad \forall i \in L, j, l \in N \tag{5}$$

$$x_{ij} \in \{0,1\}, y_i \in \{0,1\} \qquad \forall i \in L, j \in N \tag{6}$$

Decision variable x_{ij} takes 1 if slab j is assigned to lot i, otherwise 0. Decision variable y_i takes 1 if lot i is used, otherwise 0. Cosntraint (3) and (2) ensure that each slab is assigned to a lot, but only if the lot exists. (4) state the minimum number of slabs and a maximum weight per lot. (5) ensure that a maximum difference of production time and measures are not exceeded.
The fitness function of the heuristic approach is the sum of the number of lots used and a penalty for each constraint violation.

2.2 Stacking Problem

Input A set of slabs is produced that arrive on parallel inputs over time. Each slab is assigned to an article depending on its characteristics, as well to a lot. A lot contains only one type of slab.
Objective The aim is to arrange all slabs on buffer stacks in order to minimize the number of shuffling movements when lifting slabs to the delivery point every time a lot is completely produced.
Stacking Constraints Due to reasons of stability, slabs must be arranged according to their measures. Predefined stacking constraints determine which slab type is allowed to be placed on another one.
Stacks and Moves We have a set of buffer stacks and delivery stacks. A buffer stack has a maximum stack height. The initial buffer stacks are assumed to be empty. The delivery stack is assumed to be of infinite capacity. Only moves from the input to a buffer stack are considered as each slab must go to a target stack.

$\Omega = \{\omega_j\}_{j=1}^{N}$	Set of cast slabs
$\mathcal{A} = \{a_i\}_{i=1}^{A}$	Set of articles
$\mathcal{L} = \{l_i\}_{i=1}^{L}$	Set of built lots
$\mathcal{K} = \{B_k\}_{k=1}^{K}$	Set of buffer stacks

The input may contain the following mappings. Equation (7) assigns one type of article to each slab. Equation (8) assigns a lot to each slab. Equation (9) denotes the last cast slab of one lot. $e(\omega_j)$ takes 1, if ω_j is the last slab of a lot. Based on the input, α (see (10)) takes 1 if two slabs are in the same lot.

$$f : \Omega \to \mathcal{A} : f(\omega_j) = a_i \qquad \forall j \in N, i \in A \tag{7}$$

$$g : \Omega \to \mathcal{L} : g(\omega_j) = l_i \qquad \forall j \in N, i \in L \tag{8}$$

$$e : \Omega \to \{0, 1\} \tag{9}$$

$$\alpha : \Omega \times \Omega \to \{0, 1\} \tag{10}$$

Equation (11) and (12) depend on the decision variable z. β denotes the index of the first cast slab assigned to lot l_i and put on stack B_k. γ denotes whether any of the slabs, belonging to the same lot as ω_j, is put on stack B_k.

$$\beta : \Omega \times \mathcal{K} \to \Omega : \beta_{ik} = \left\{ \begin{array}{l} \min\{j | \alpha_{ij} z_{kj} = 1\} \\ i, \quad \text{otherwise} \end{array} \right\} \qquad \forall i, j \in N, k \in K \tag{11}$$

$$\gamma : \Omega \times \mathcal{K} \to \{0, 1\} \qquad \forall j \in N, k \in K \tag{12}$$

2.2.1 Objective Function

Every time a lot l_i is completely produced, given by mapping (9), the shuffling movements are evaluated. The number of shuffling movements has to be minimized and is given in equation (13).

$$\min \sum_{j=1}^{N} \sum_{k=1}^{K} \Big((h_{\omega_j}(B_k) + \gamma_{jk} - h'_{\omega_j}(B_k) - n_{\omega_j}(B_k)) * e(\omega_j) \Big) \tag{13}$$

The objective function is composed of the following parts.

$$h_{\omega_j}(B_k) = \sum_{l=1}^{\omega_j} z_{kl} - \sum_{i=1}^{\omega_j - 1} \left(e(\omega_i) \sum_{n=1}^{i} \alpha_{in} z_{kn} \right) \tag{14}$$

$$h'_{\omega_j}(B_k) = h_{\omega_{\beta_{jk}}}(B_k) - \sum_{i=\beta_{jk}}^{\omega_j - 1} \left(e(\omega_i) \sum_{n=1}^{\beta_{jk}} \alpha_{in} z_{kn} \right) \tag{15}$$

$$n_{\omega_j}(B_k) = \sum_{l=\beta_{jk}}^{j} \alpha_{jl} z_{kl} \tag{16}$$

Equation (14) denotes the height of buffer stack B_k when slab ω_j is produced. That is the sum of slabs placed on B_k reduced by the sum of slabs moved away. Equation (15) denotes the position of the slab in buffer stack B_k with index β at the time that slab ω_j is cast. Equation (16) denotes the number of slabs of the same lot in buffer stack B_k. Decision variable z determines whether slab ω_j is placed on buffer stack B_k.

$$s.t. \quad \sum_{k=1}^{K} z_{kj} = 1 \qquad \forall j \in N \tag{17}$$

$$h_{\omega j}(B_k) \leq H \qquad \forall j \in N, k \in K \tag{18}$$

$$z_{kj} \in \{0, 1\} \qquad \forall j \in N, k \in K \tag{19}$$

Stacking constraints (20) state whether an article, characterized by its measurements, is allowed to be placed on another one. The constraints are based on a matrix of dimension $C : A \times A : (a_i, a_j) \mapsto \{0, 1\}$, being 1 if a_i is allowed to be placed on a_j. Summed up over all stacks and all slabs, equation (20) ensures that a constraint is only violated by slabs which are already moved away.

$$\sum_{l=1}^{j} \left((z_{kl})(z_{kj})(1 - C_{a_j a_l}) \right) = \sum_{i=1}^{j-1} \left(e(\omega_i) \sum_{n=1}^{i} \left(\alpha_{in} (z_{kn})(z_{kj})(1 - C_{a_j a_n}) \right) \right) \tag{20}$$

2.3 Interdependency

Due to the fact that slabs are moved to a target as soon as their lot is completely produced, the delivery order is implicitly determined by the assigned lotnumber. Hence, it is reasonable that the assignment of lots has a significant impact on the efficiency of the stacking problem. Thus, an exchange of the solution qualities is worthwhile in order to achieve a best possible combined solution. In empirical studies we focused on the complexity of the problem as well as on a combined solution approach.

	Runtime (sec)		**Best Solution**		**Feasible (%)**
Instance	CPLEX	VNS	CPLEX	VNS	VNS
30-3	26.73	41.72	**9**	9	79
50-6	213.97	**108.80**	32	**31**	76
60-6	16313.41	**258.12**	41	**40**	79
68-7	7706.96	**339.54**	31	**31**	50
100-8	x	**244.39**	x	**74**	83

Table 1: Results on complexity

3 Empirical Studies

Empircial studies are conducted on problem instances differing in the number of slabs and buffer stacks, indicated by the naming (X-Y stands for X slabs, Y buffer stacks). Instances are solved by using CPLEX [4] and variable neighbourhood search (VNS) in order to show the complexity of the problem by investigating the runtime, solution quality and number of feasible solutions. Instances of higher dimensions are solved by using a multi-objetive algorithm namely NSGA II and by using a combined method using VNS. A neighbour is defined as a swap of two assigned stacks/lots as well as the assignment of a random stack/lot to a slab. Studies using heuristic methods are performed by HeuristicLab [14] [15]. All tests were calculated on a laptop with a Intel(R) Core(TM) i7-4600U CPU 2.10GHz 2.70GHz processor.

3.0.1 Complexity

Results of runtime and number of feasible solutions show that the lot-aware slab stack shuffling problem is of high complexity. CPLEX could solve the problem up to 20 slabs to optimality. Only instances up to 68 slabs could be solved before running out of memory. The runtime of CPLEX shows exponential behaviour. The VNS performs with a rather constant runtime. Table 1 allows a comparison of CPLEX and VNS regarding the best found solutions and runtime. For this study we dissociated the lot-building problem from the stacking problem. Presented results are of the stacking problem only.

We can derive that even for heuristic methods the problem is hard to solve. For solving real world instances of more than 200 slabs only heuristic approaches work. Due to the number of feasible solutions the algorithm may be adapted by developing specialized operators. Worth to mention is that the best found qualities of VNS are at least as good as obtained by CPLEX.

3.0.2 Sequential Solution Approach

In order to exploit the property of interrelation we consider a sequential solution approach. Using VNS, the developed method solves the lot-building several times and gives each solution as input to the stacking problem. The best found combined solution is returned. An advantage of this approach is that also sub-optimal solutions of the lot-building problem are considered in order to find the best possible solution for the combined problem. Results are compared to a multi-objective algorithm, NSGA II.

	Quality	
Instance	NSGA II	Sequ.Method
20-5	17	17
50-6	44	**33**
100-8	114	**72**

Table 2: Results of combined solving

Table 2 show that the sequential approach achieves good solutions compared to a standard NSGA II. A sequential approach is reasonable in case specialised algorithms for the sub-problems exist since a separate application of both algorithms is allowed.

4 Conclusion and Future Work

In this research work an optimization problem composed of two autonomous but related problems has been developed. Experimental results show that the modelled problem is very hard to solve. Further work will consider customized heuristic algorithms in order to allow the solving of real world instances in reasonable time. Research work will also be done on linking autonomous but related optimization problems. A method for solving several optimization problems in an interrelated way shall be developed. With that purpose an approach for exchanging solution qualities and deduced parameters shall be researched in order to reach a combined sequential solution approach.

Acknowledgments

The work described in this paper was done within the COMET Project Heuristic Optimization in Production and Logistics (HOPL), #843532 funded by the Austrian Research Promotion Agency (FFG).

References

[1] M. Caserta, S. Schwarze, S. Voß: A mathematical formulation and complexity considerations for the blocks relocation problem. European Journal of Operational Research **219**(1) (2012) 96 – 104

[2] De Castillo, B., & Daganzo, C. F.: Handling strategies for import containers at marine terminals. Transportation Research Part B: Methodological, **27**(2) (1993) 151-166

[3] E.F.A. Fernandes, L. Freire, A.C. Passos, A. Street: Solving the Non-linear Slab Stack Shuffling Problem Using Linear Binary Integer Programming. EngOpt 2012 - 3rd International Conference on Engineering Optimization

[4] IBM, C. IBM Software. Retrieved from CPLEX Optimizer: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/ (30.4.2015)

[5] Kang, J., Ryu, K.R., Kim, K.H.: Deriving stacking strategies for export containers with uncertain weight information. Journal of Intelligent Manufacturing **17** (2006) 399–410

[6] K.H. Kim, J.W. Bae: Re-marshalling export containers in port container terminals. Computers and Industry Engineering **35** (1998) 655–658

[7] F.G. König, M.E. Lübbecke, R.H. Möhring, G. Schäfer, I. Spenke: Solutions to Real-World Instances of PSPACE-Complete Stacking. Lecture Notes in Computer Science Volume **4698** (2007) 729-740

[8] Y. Lee, N.Y. Hsu: An optimization model for the container pre-marshalling problem. Computers & Operations Research **34** (2007) 3295–3313

[9] J. Luo, Y. Wu, A. Halldorsson, X. Song: Storage and stacking logistics problems in container terminals. OR Insight **24** (2011) 256-275

[10] Singh, K. A., Srinivas, Tiwari, M.K.: Modelling the slab stack shuffling problem in developing steel rolling schedules and its solution using improved Parallel Genetic Algorithms. International Journal of Production Economics **91** (2004) 135-147

[11] L.X. Tang, J.Y. Liu, A.Y. Rong, Z.H. Yang: An effective heuristic algorithm to minimise stack shuffle in selecting steel slabs from the slab yard for heating and rolling. Journal of Operational Research Society **52** (2001) 1091–1097

[12] L.X. Tang, J.Y. Liu, A.Y. Rong, Z.H. Yang: Modelling and a genetic algorithm solution for the slab stack shuffling problem when implementing steel rolling schedules. International Journal of Production Research **40**(7) (2002) 1083–1095

[13] Tang, L., Ren, H.: Modelling and a segmented dynamic programming-based heuristic approach for the slab stack shuffling problem. Computers&Operations Research **37** (2010) 368-375

[14] S. Wagner, G. Kronberger, A. Beham, M. Kommenda, A. Scheibenpflug, E. Pitzer, S. Vonolfen, M. Kofler, S. Winkler, V. Dorfer and M. Affenzeller: Architecture and Design of the HeuristicLab Optimization Environment. Topics in Intelligent Engineering and Informatics **6** (2014) 197–261

[15] S. Wagner: Heuristic Optimization Software Systems - Modeling of Heuristic Optimization Algorithms in the HeuristicLab Software Environment. PhD Thesis (2009) Johannes Kepler University Linz