Competitive Analysis of on-line Stack-Up Algorithms

J. Rethmann and E. Wanke

University of Düsseldorf, Department of Computer Science, D-40225 Düsseldorf, Germany

Abstract. Let $q = (b_1, \ldots, b_n)$ be a sequence of bins. Each bin is destined for some pallet t. For two given integers s and p, the stack-up problem is to move step by step all bins from q onto their pallets such that the position of the bin moved from q is always not greater than s and after each step there are at most p pallets for which the first bin is already stacked up but the last bin is still missing. If a bin b is moved from q then all bins to the right of b are shifted one position to the left. We determine the performance of four simple on-line algorithms called First-In, First-Done, Most-Frequently, and Greedy with respect to an optimal off-line solution for the stack-up problem.

1 Introduction

In this paper, we consider the problem of stacking up bins from a conveyer onto pallets by so-called stack-up systems that are usually located at the end of pick-to-belt order-picking systems (see [dK94, LLKS93]). Each order consists of several bins. The bins arrive the stack-up system on a conveyer. At the end of the conveyer the bins are moved by stacker cranes onto pallets, where they are stacked up. The pallets are build on certain stack-up places. Vehicles take full pallets, put them onto trucks, and bring new empty pallets to the stack-up places.

All bins belonging to one order have to be placed onto the same pallet and bins for different pallets have to be placed onto different pallets. Unfortunately, the bins arrive the stack-up system not in a succession such that they can be placed one after the other onto pallets. Furthermore, not all bins of the sequence are known in advance. So the stack-up problem is to compute a step by step placement of bins onto pallets such that the position of the placed bin is always not greater than s (an upper bound on the storage capacity) and after each step there are at most p (an upper bound on the number of stack-up places) pallets for which the first bin is already stacked up but the last bin is still missing.

The stack-up problem is up to now not intensively investigated by other authors, although it has important practical applications. However, the following results are already known about the stack-up problem. The stack-up decision problem is in general NP-complete [GJ79] but can be solved very efficiently and even in parallel if the storage capacity s or the number of stack-up places p is bounded, see [RW97b]. In [RW97a], a polynomial time off-line approximation algorithm for the processing of sequences q with a storage capacity of $\lceil s_{\min}(q,p) \cdot \log_2(p+1) \rceil$ and $p+1$ stack-up places is introduced, where $s_{\min}(q,p)$ is the minimal storage capacity to solve the stack-up problem with p stack-up places. Off-line algorithms assume that the complete sequence of bins q is given to the input and thus known in advance.

In this paper, we analyze the worst-case behavior of simple on-line algorithms by competitive analysis [MMS88]. That means, we determine the performance of our on-line algorithms with respect to an optimal off-line solution. In an off-line processing the complete sequence is known in advance, whereas in an on-line processing in each step only the first s bins for q are known. On-line algorithms are very interesting from a practical point of view. The algorithms introduced and analyzed in this paper are called First-In (FI), First-Done (FD), Most-Frequently (MF), and Greedy. Let $s_A(q,p)$ ($p_A(q,s)$) be the minimum storage capacity (minimum number of stack-up places, respectively,) such that algorithm A processes sequence q with p stack-up places (with a storage capacity of s bins, respectively). Let q be a sequence of bins that can be processed with a storage capacity of s bins and p stack-up places. Our results can be summarized as follows.

A	FI	FD	MF & Greedy
$s_A(q,p) \leq$	$s-(p-1)+p(B_q-1)$	$s+p(B_q-1)$	$(p+1)s-p$
$p_A(q,s) \leq$	$s-1+p$	$s-1+p$	$\min\{s-1+p,\ p(\log_2(s)+2)\}$

We give also general upper and lower bounds on $s_A(q,p)$ and $p_A(q,s)$ for arbitrary stack-up algorithms A. The results introduced in this paper show that even simple on-line algorithms like Most-Frequently and Greedy have provable good worst-case performance.

References

[dK94] R. de Koster. Performance approximation of pick-to-belt orderpicking systems. *European Journal of Operational Research*, 92:558–573, 1994.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability.* W.H. Freeman and Company, San Francisco, 1979.

[LLKS93] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys. Sequencing and Scheduling: Algorithms and Complexity. In S.C. Graves, A.H.G. Rinnooy Kan, and P.H. Zipkin, editors, *Handbooks in Operations Research and Management Science, vol. 4, Logistics of Production and Inventory*, pages 445–522. North-Holland, Amsterdam, 1993.

[MMS88] M.S. Manasse, L.A. McGeoch, and D.D. Sleator. Competitive algorithms for on-line problems. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, pages 322–333. ACM, 1988.

[RW97a] J. Rethmann and E. Wanke. An approximation algorithm for stacking up bins from a conveyer onto pallets. In *Proceedings of the Annual Workshop on Algorithms and Data Structures*, volume 1272 of *Lecture Notes in Computer Science*, pages 440–449. Springer-Verlag, 1997.

[RW97b] J. Rethmann and E. Wanke. Storage Controlled Pile-Up Systems. *European Journal of Operational Research*, 103(3):515–530, 1997.