REASSIGNING STORAGE LOCATIONS IN A WAREHOUSE TO OPTIMIZE THE ORDER PICKING PROCESS

Monika Kofler[(a)], Andreas Beham[(b)], Stefan Wagner[(c)] , Michael Affenzeller[(d)], Clemens Reitinger[(e)]

[(a – d)] Upper Austria University of Applied Sciences
School for Informatics, Communications, and Media
Heuristic and Evolutionary Algorithms Laboratory
Softwarepark 11, 4232 Hagenberg, Austria

[(e)] ROSENBAUER INTERNATIONAL AG
Paschinger Strasse 90
4060 Leonding, Austria

[(a)]monika.kofler@fh-hagenberg.at, [(b)]andreas.beham@fh-hagenberg.at, [(c)]stefan.wagner@fh-hagenberg.at,
[(d)]michael.affenzeller@fh-hagenberg.at, [(e)]clemens.reitinger@rosenbauer.com

ABSTRACT

Warehouses are an essential component of the supply chain, used for buffering the material flow, stock consolidation and value-added-processing. Operation managers typically do a good job of filling their warehouse but initially assigned storage locations might become sub-optimal over time, due to seasonal fluctuations in demand, short product life cycles or high inventory levels. Fragmented storage is a particular issue in order picking environments, where the optimal storage location of a product is not only dependent on its turn-over rate but also on the storage locations of items that frequently occur in the same picking job. In practice, operations managers are forced to periodically reorganize the warehouse to keep it operating efficiently. This process is generally done manually without any decision-support tool. In this paper we introduce an optimization approach that automatically reorganized item locations. Results are evaluated using a simulation model that simulates the picking and transport processes in the warehouse.

Keywords: re-warehousing, storage location problem, warehouse simulation

1. THE STORAGE ASSIGNMENT PROBLEM AND PERFORMANCE INDICATORS

The storage assignment problem involves the placement of a set of items or pallets in a warehouse in such a way that one or more performance measures are optimal. In typical distribution centres travel time to retrieve an order has been found to be the largest component of labour, amounting to 50% or more of total order picking time according to (Tompkins et al. 1996). By contrast, only 10% of the total order picking time is invested in the actual retrieval of products from their storage locations. Under the assumption that picking times are not correlated with particular storage locations we may treat them as fixed costs and omit them in the evaluation of warehouse assignments, thereby focusing on travel time.

Directly optimizing the picker travel times is complex, requiring a detailed warehouse layout and resource model (pickers, possible routes, routing strategy, collision detection and avoidance). Most approaches do therefore use an alternative, albeit related, objective measure. Early attempts to reduce travel time were for example based on the idea that fast-moving items should be located in easily accessible forward pick areas. Heskett (1964) extended this simple policy and proposed the cube order index (COI) rule, which ensures that heavy or fast-moving products are stored in more desirable locations close to ground level. Modifications of the COI rule have since been published, which also consider inventory costs or zoning constraints (Malmborg 1996). In general, these turn-over based policies work well if the order sizes are small and pickers return to the shipping deck after each pick.

In order picking environments a picker usually retrieves multiple items per order. Items that are frequently ordered together are said to be *correlated* or *affine* (Garfinkel 2005). Storing affine items close to each other may reduce the total travel time of the order pickers, although this is not guaranteed and depends on the picker routing. As noted by Waescher (2004) the fact that two items appear in the same order does not necessarily mean that a picker will directly proceed from one to the other on his route. In addition, structural conditions, such as narrow aisles that do not allow reverse back out, or large orders might require a full traversal of the warehouse anyway. In this case storing by affinity does not significantly reduce travel time but could on the contrary lead to congestion in certain aisles since it does not enforce balanced storage of fast-moving items.

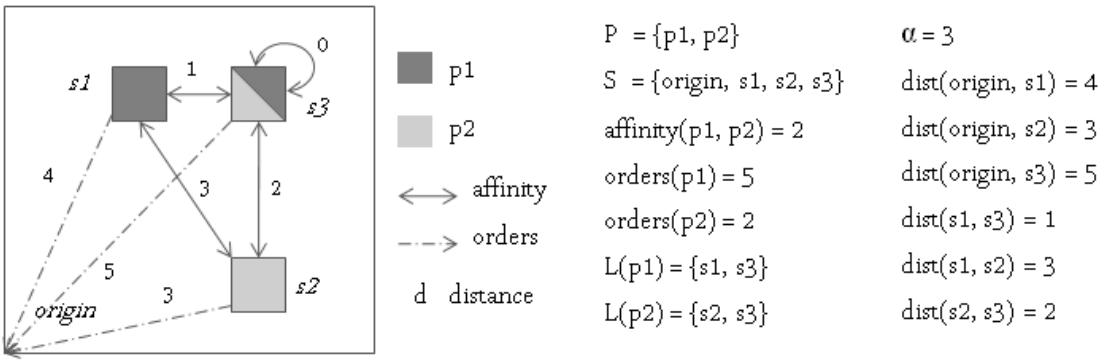

Figure 1: Example for the calculation of the quality of a simple assignment, consisting of a rack where two *products* p_1 and p_2 are stored in three *locations* s_1, s_2 and s_3. In addition, the location *origin* denotes the lower left corner of the rack. The stored *quantity* per location and product is one. All other storage locations are empty. The two quality measures, as defined in Equation 8 and 9, amount to *totalPickFrequencyScore* = 30.5 and *totalPartAffinityScore* = 6.

2. WAREHOUSE ASSIGNMENT EVALUATION

We combine two objectives to evaluate warehouse assignments generated by our optimization approach. Products with strong affinity should be placed together and fast-moving objects should be placed close to the shipping docks.

Prior to defining the objective function, we need to introduce a couple of variables. First of all, the warehouse definition consists of the set of storage locations **S** and a distance matrix denoting the travel distances (or efforts) between pairs of storage locations.

S = set of all storage locations s_k ; $0 < k <= m$ (1)

where $m = |S|$

$dist(s_k, s_l)$ = distance between storage location s_k and s_l (2)

In addition to the storage locations that can hold products, at least one special *origin* locations must be defined that denotes the shipping dock. In our case we employ a return routing strategy, were aisles are always entered and left from the front (cf. *Section 4: Warehouse Simulation Model*). Therefore we have defined an *origin* location in the lower left corner of each warehouse rack, ensuring that fast-moving objects are placed in more favorable locations close to the ground and near the front of an aisle.

The distance is a scalar value that can be assigned in different ways. For a rough estimate one might pick the linear distance between storage locations. For a more realistic approximation one could sum up the total travel distance, taking transport paths and perhaps even height differences into account. For our scenario we used a travel time estimate. First we split the travel distances into sub-movements such as aisle switch, forward/backward movement of the truck and fork up-down/left-right movement. Then we weighted the travel distances with empirically determined average travel velocities for the different movement types and summed up the resulting values to obtain an estimate for the required travel time between locations.

Usually, the storage locations and the distance matrix need to be determined only once for a given warehouse and can later be re-used for different problem instances.

Conversely, the following parameters are likely to change over time and need to be retrieved from the enterprise resource planning or warehouse management system. Most importantly, the set **P** lists all products that are present in a particular assignment.

P = set of all products p_i; $0 < i <= n$ where $n = |P|$ (3)

For each product p_i we need to know the total number of picking orders **orders(p_i)** in which the product occurs. Similarly, the affinity matrix stores how often two products are ordered together. Finally, the current warehouse assignment defines how many products p_i are stored at location $\mathbf{s_k}$. The set of locations **L(p_i)** stores all locations of a particular product.

$orders(p_i)$ = number of orders in which p_i occurs (4)

$affinity(p_i, p_j)$ = number of orders in which products p_i and p_j occur together (5)

$quantity(p_i, s_k)$ = number of packing units of product p_i stored at location s_k (6)

$L(p_i)$ = set of all $s \in S$ where $quantity(p_i, s) > 0$ (7)

The entities defined in 3-7 can be calculated from order picking histories and the current warehouse assignment. We can now define the objective functions in equation 8 and 9.

totalPickFrequencyScore = (8)

$$\sum_{i=1}^{n} \frac{orders(p_i)}{|L(p_i)|} * \sum_{s \in L(p_i)} dist(s, origin)$$

The *totalPickFrequencyScore*, as defined in Equation 8, ensures that frequently picked products are placed in more favorable storage locations near the ground and the aisle entries. For each product it detects all current

storage locations $L(p_i)$, calculates their distance to the origin and weighs each distance with the expected number of picks given the number of previous orders(p_i). The picks are uniformly distributed on all storage locations, independent of the actual stored quantities in the different locations.

totalPartAffinityScore = (9)

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\text{affinity}(p_i,p_j)}{|L(p_i)|*|L(p_j)|} * \sum_{s_k\in L(p_i)}\sum_{s_l\in L(p_j)}\text{dist}(s_k,s_l)$$

The *totalPartAffinityScore* takes all pairs of products p_i and p_k, and retrieves all respective storage locations $L(p_i)$ and $L(p_j)$ from the current assignment. The distance between each resulting storage location pair is calculated and weighted with the part affinity divided by the number of location pairs $|L(p_i)|* |L(p_k)|$. The term reduces to zero for products with no part affinity, therefore the calculation can be sped up by only looking at products p_k that have an affinity greater than zero with a given product p_i.

The resulting multi-objective evaluation function for assignments is computed as weighted sum of the two objective functions given in Equation 8 and 9 such that

quality = α * totalPickFrequencyScore (10)
+ β * totalPartAffinityScore.

Figure 1 demonstrates how to calculate the quality of a small sample assignment with the given objective function.

As already mentioned the assessment of storage configurations via the proposed objective function alone might - in some cases - lead to solutions that are far from optimal in practice. We believe that a realistic evaluation of routing and storage strategies needs to incorporate dynamic aspects such as fluctuating travel and picking times, floor space utilization in the docking area, resource constraints (e.g. a limited number of pallets) or potential congestion situations when forklifts wish to access the same aisles simultaneously. We therefore employ a complementary simulation model for the evaluation of storage configurations as described in Section 4.

3. HEURISTICLAB

HeuristicLab (http://dev.heuristiclab.com) is a framework for heuristic and evolutionary optimization which is based on the Microsoft .NET framework. One core design goal of HeuristicLab was to shift the application of optimization strategies from an implementation point of view to a modeling point of view. In HeuristicLab algorithms are modeled by combining several generic parts using a graphical user interface. Similarly, problems are abstracted such that they make use of a certain representation and provide a fitness function as well as import parsers and graphical representations. The underlying representation, also called the encoding of a solution, provides manipulation operators such as crossover or mutation. All optimization runs were conducted with HeuristicLab and solutions were evaluated via the objective function given in Section 2. The best solutions were subsequently validated via simulation, to get a more realistic assessment of the quality of the generated assignments.

4. WAREHOUSE SIMULATION MODEL

We developed a simulation model in AnyLogic™ 6 that was based on a real-world high rack warehouse. Our project partner kindly provided enterprise data such as layout information, order picking histories and daily warehouse storage assignments. The warehouse floor plan was built to scale, with two fork-lift trucks for picking. Travel and pick time distributions were obtained empirically and used to parameterize the fork-lifts. Storage assignments can be loaded into the model as well as a set of picking jobs that ought to be simulated. The model employs a return routing strategy for incoming orders, were aisles are always entered and left from the front (de Koster and Roodbergen 2007), and calculates and performs an optimal picking sequence for each order.

The simulation model allows decision makers to parameterize and evaluate different warehouse configurations according to the various performance indicators, such as

- **Total travel distance:** The total distance travelled by all pickers.
- **Fork lift blocking time**: The model puts certain access restraints on the pickers. For instance, only one fork lift trunk can access the aisle per time. The total blocking time sums up the time spent waiting for an aisle to become free again.
- **Average order picking time**: The average time required to complete an order, including travel, waiting and picking times

All generated warehouse assignments were evaluated according to objective function introduced in Section 2 and the three indicators given above.

5. STORAGE LOCATION REASSIGNMENT

The literature about re-warehousing activity is limited. As stated in (Garfinkel 2005) one known approach was introduced by (Sadiq 1993), who periodically revises the assignments in accordance to the variation of item pick frequencies over a longer time period. Similarly, Housseman et al. (2009) used a simulation model to estimate the impacts of re-warehousing in cryo-conservation centers. In this paper we employ

- **first improvement local search** and
- **simulated annealing** (see Kirkpatrick 1983)

to optimize a given initial warehouse configuration. The base of both improvement methods is a set of moves that relocate a given quantity of items to a new location. In particular, we implemented different *move* operators, which swap the whole content of two randomly selected locations.

- **Random Swap 2**: Random swap of two pallets
- **Random Swap 3:** Cyclic swap of three pallets
- **Attraction Move:** Movement of pallets towards a more *attractive* position, meaning closer to affine products or – in case of a high turnover rate – the shipping dock

Table 1: Parameters for SA

Parameter	Value
Iterations	500,000
Temperature	80000
Annealing Factor	0.998
Annealing Scheme	Multiplicative
Inner Iterations	20

6. EXPERIMENTS AND RESULTS

We conducted test runs on data from a high-rack warehouse with more than 7,000 storage locations. As already mentioned, our project partner provided historical order pick data, consisting of information about more than 10,000 products and roughly 300,000 individual picking operations. The affinity matrix was calculated using this historical data set. Moreover, daily snapshots of the current warehouse assignment and planned picking orders for the day were exported from the warehouse management system. We used five such snapshots, optimized each assignment with respect to the objective function and subsequently evaluated the resulting assignments by simulating the scheduled picking orders. All optimization runs were conducted in a high performance computing environment on an 8-core machine with 2x Intel Xeon CPU, 2.5 Ghz and 32GB memory.

To investigate the trade-offs between placement by part affinity and placement by retrieval frequency, we had to find a good setting for the parameters α and β in the objective function (cf. Section 2, Equation 10). We first sampled the Pareto-optimal set with first improving neighborhood search and simulating annealing. We fixed $\beta = 1$ and conducted tests for $\alpha \in \{1, 2, 3, \ldots 20, 30, 40, 50, 60\ 70, 80, 90, 100\}$. We found that setting $\alpha = 2$ achieved a good trade-off between the two objectives for the given warehouse. The parameterization must of course be adapted for other warehouses, but in this paper all test runs were conducted with these settings. The result tables list the individual quality values separately, to better compare the assignments.

We initially employed simulated annealing with fairly long algorithm execution times of thirty minutes to five hours to get a rough estimate of the optimization potential. With the algorithm settings from Table 1 and by stochastically selecting from the three move operators for each move creation with equal probability we were able to generate assignments that improved the total quality, as defined by our objective function, on test set 1 by 32%. The optimized assignment improved the part affinity score by 11% and the pick frequency score by 40%. To illustrate the effects of the optimization, we implemented two visualizations for the inspection of assignments. The front view (cf. Figure 2) depicts the perspective of a worker, standing within an aisle and looking at one set of racks. The top view (cf. Figure 3) shows a bird's eye perspective on the warehouse, clearly showing the different aisles. On the one hand the views can show the quality of the assignment in a heat map like display. On the other hand it is possible to select a particular part in the warehouse and only display all affine parts with their locations and qualities. In this case, the relative lightness or darkness of the locations depicts the weighted part affinity score. Darker locations store parts with higher scores than lighter locations.

As can be seen in Figure 2, the initially scattered affine parts from test set 1 are tightly packed within the rack after optimization. Parts with higher scores (and therefore a probably high individual pick frequency) are positioned in more favorable locations towards the lower left edge of the row. Similarly, Figure 3 shows that affine parts are mostly concentrated within one aisle after the optimization.

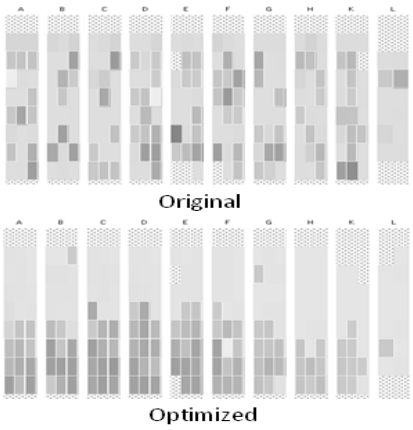

Figure 2: Set of affine parts before and after the optimization within one rack (so-called front view).

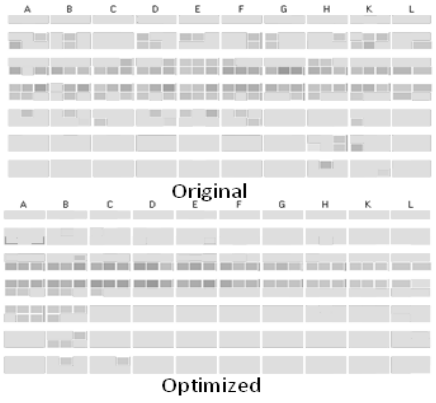

Figure 3: Set of affine parts before and after the optimization visualized in top view.

It should be noted that this is only an example, illustrating the effects of the optimization on the storage locations of one particular part and its affine parts. Also, the displayed result took 3 hours to generate with SA and led to a complete overhaul of the warehouse with more than 95% of the parts changing locations compared to the initial assignment.

For practical purposes, shorter algorithm runtimes would be preferable, in particular if re-assignments should be carried out on demand, when resources are available in the warehouse. We therefore also conducted tests with first-improvement local search and tight optimization time windows of 1-3 minutes. Once again, all three move operators were used with equal probability.

Table 2: Best results for tests with local search and a very tight optimization time window of 1-3 minutes for five test sets

Test Set	Affinity Score	Frequency Score	Computing time
1	-1%	-11%	1-3 min
2	-1%	-13%	
3	-1%	-8%	
4	-3%	-19%	
5	-3%	-18%	

As can be seen in Table 2, improving the placement of parts according to their pick frequency is much easier than grouping affine parts close together. This is not surprising, since the latter requires more moves and products that are picked with a wide variety of other products may experience conflicting "pulls" (e.g. via the attraction move) towards multiple areas in the warehouse.

Finally, to assess the impact on picker travel distance and order picking time, we conducted simulation runs for the test sets to compare the initial and generated assignments. In particular, we wished to investigate if assignments with better qualities would also lead to improved picker travel times and if the ratios were similar. We simulated five replications per assignment to account for stochastic variability. Due to the employed deterministic picker routing algorithm the travelled distance per assignment is the same across replications.

We exemplarily list and discuss the simulation results for test set 1 and the best SA test run in Table 3 and 4. While the generated assignment achieves better results on all three performance indicators, the improvement is not as great as one might hope for, given a 40% improvement on the objective function. The total travel distance of the two forklifts could be reduced by 1.85 km or about 7%. Blocking time, which we knew to be an issue beforehand, could also be improved by 18% and the average order picking time dropped by 8.5 minutes or roughly 35%. It should be noted, though, that results fluctuate a lot between replications, since the underlying distribution from which we estimate the picking times has a large variance. We are currently discussing these results with the warehouse operator and evaluating possible improvements to the simulation model. However, even our initial and very basic tests show that the employment of simulation is crucial for a more realistic estimate of the impacts of different assignments on warehouse logistics.

Table 3: Simulation results for the original warehouse assignment from test set 1.

Replication number	Travel distance [km]	Blocking time [min]	Average order picking time [min]
1	18.91	255.17	24.96
2		239.42	28.92
3		232.08	23.76
4		252.5	24.84
5		210.7	18.3

Table 4: Simulation results for the warehouse assignment from test set 1 after optimization with SA.

Replication number	Travel distance [km]	Blocking time [min]	Average order picking time [min]
1	17.06	208.08	18.54
2		209.08	15.36
3		194	15.66
4		146.83	12.78
5		217.17	15.84

7. CONCLUSIONS AND FUTURE WORK

The optimization approach presented in this paper is still a work in progress. The main innovations of our approach lie in the custom objective function and the employment of simulation for a realistic evaluation of the generated assignments. So far, we have acquired and pre-processed the required data, specified a generic model for the warehouse assignment problem, created and parameterized a simulation model for the evaluation of results, implemented solution manipulation operators and generated preliminary results with two standard algorithms. Our future research will focus on the following aspects:

First of all more exhaustive tests need to be conducted, also employing a more diverse set of heuristic optimization techniques, such as tabu search, evolution strategy and force-driven algorithms. This will be a major research focus in the second stage of our project.

Secondly, even short optimization runs such as those conducted with local search perform a large number of moves and thus re-arrange a multitude of parts. Such extensive rearrangements, an approach that we call *re-warehousing*, can block the fork lift truck for a couple of hours at least, is costly and might therefore

not be possible too frequently. Conversely, it should be easier to conduct a small number of cleanup tasks in idle slots between order picking or at the end of shifts. The idea behind this *healing* approach is that iteratively improving the placement of parts will lead to a good total warehouse assignment. We plan to conduct a study on the relative merits and efforts involved in *re-warehousing* vs. *healing* and derive recommendations for different warehouse types.

Finally, the company data used in this study is copyrighted and proprietary. We do however plan to publish a properly pre-processed and masked data set in the near future to allow other researchers to reproduce our results. In addition, we wish to apply our approach to different warehouses to investigate scaling, applicability and variance.

ACKNOWLEDGEMENTS

The work described in this paper was done within the Josef Ressel Centre for Heuristic Optimization *Heureka!* and sponsored by the Austrian Research Promotion Agency (FFG). For more information about Heureka! please visit http://heureka.heuristiclab.com.

REFERENCES

Garfinkel, M., 2005. *Minimizing Multi-zone Orders in the Correlated Storage Assignment Problem*, PhD thesis, GA Tech.

Heskett, J.L., 1964. Putting the cube-per-order index to work in warehouse layout, *Transportation and Distribution Management*

Housseman, S., Absi, N. Feillet, D. and Dauzère-Pérès, S., 2009. Impacts of Radio-identification on cyro-conservation centers through simulation, *Proceedings of the 2009 Winter Simulation Conference*, 2065–2077.

Kirkpatrick, S., Gelatt,, C. D. and Vecchi, M. P., 1983. Optimization by simulated annealing. *Science* 220, 671–680.

de Koster, R., Le-Duc, T. and Roodbergen, K.J., 2007. Design and Control of Warehouse Order Picking: a literature review, *European Journal of Operational Research*, 182 (2), 481-50.

Malmborg, C.J., 1996. Storage Assignment Policy Tradeoffs, *International Journal or Production Research*, 33, 989–1002

Sadiq, M., 1993. *A hybrid clustering algorithm for reconfiguration of dynamic order picking systems*, Ph.D. Dissertation, University of Arkansas.

Waescher, G., 2004. *Order Picking: A Survey of Planning Problems and Methods*, In: Dyckhoff, H., Lackes, R. and Reese, J., *Supply chain management and reverse logistics*, Springer, 323-347, Heidelberg, Berlin.

Tompkins, J.A., White, J.A., Bozer, Y.A., Frazelle, E.H., Tanchoco, J.M.A. and Trevino, J., 1996. *Facilities planning*. Wiley, New York.

AUTHORS BIOGRAPHY

MONIKA KOFLER studied Medical Software Engineering at the Upper Austrian University of Applied Sciences, Campus Hagenberg, Austria, from which she graduated in 2006. She is currently employed as a research associate at the Research Center Hagenberg and pursues her PhD in engineering sciences at the Johannes Kepler University Linz, Austria.

ANDREAS BEHAM received his MSc in computer science in 2007 from Johannes Kepler University (JKU) Linz, Austria. His research interests include heuristic optimization methods and simulation-based as well as combinatorial optimization. Currently he is a research associate at the Research Center Hagenberg of the Upper Austria University of Applied Sciences (Campus Hagenberg).

STEFAN WAGNER received his MSc in computer science in 2004 and his PhD in engineering sciences in 2009, both from Johannes Kepler University (JKU) Linz, Austria; he is professor at the Upper Austrian University of Applied Sciences (Campus Hagenberg). Dr. Wagner's research interests include evolutionary computation and heuristic optimization, theory and application of genetic algorithms, machine learning and software development.

MICHAEL AFFENZELLER has published several papers, journal articles and books dealing with theoretical and practical aspects of evolutionary computation, genetic algorithms, and meta-heuristics in general. In 2001 he received his PhD in engineering sciences and in 2004 he received his habilitation in applied systems engineering, both from the Johannes Kepler University of Linz, Austria. Michael Affenzeller is professor at the Upper Austria University of Applied Sciences, Campus Hagenberg, and head of the Josef Ressel Center *Heureka!* at Hagenberg.

CLEMENS REITINGER studied Mechanical Engineering - Economics (Industrial Engineering) at Vienna University of Technology, Austria, from which he graduated in 2003. He is currently employed as Head of Logistics at Rosenbauer International AG and also involved in the *Heureka!* project.