ACTA TECHNICA CORVINIENSIS – BULLETIN OF ENGINEERING
TOME XVI [2023] | FASCICULE 2 [APRIL – JUNE]

[1.]Miklós SZABÓ, [1.]Zoltán András NAGY, [1.]Tamás HARTVÁNYI

RANDOM ALLOCATION EFFECT ON STORAGE PERFORMANCE

[1.] Széchenyi István University, Győr, HUNGARY

Abstract: Todays not the products race with products, but supply–chains with supply–chains, also every chain is so strong as the weakest part of it. It is essential to show high–performance on market, not just in production, but in logistic process too. A compact rack–system has one of the best area utilization, but it is afraid of less dynamic capacity. The authors started find a solution, how to utilize area for logistic process next to fast material handling. In this paper we publish a simulation that shows out the effect of random allocation on storage performance in a compact rack–system that opens new ways for research.
Keywords: Storage Location Assaignment Problem (SLAP), simulation, compact rack–system

INTRODUCTION

The inventories have two main criteria: static– and dynamic capacity. Static capacity defines the amount of materials could be hold at the same time, dynamic capacity defines the amount of materials could be handled in time–period. Most cases companies use conventional pallet rack system that seems to be the most effective in dynamic capacity, because every pallet is available without moving other pallets, but many times inventories has limited area to use and high amount of materials to handle. In that reason companies have to use other rack systems that are more effective in area utility.

According to Pareto thesis most of the handled materials came from a few Stock Keeping Units (SKUs), while the other small part of materials is many. Because of that the compact rack–system is more logic choice than the conventional one in aspect of SKUs.

We think that compact rack–system can be dynamic too. With compact rack systems less travel–distance has to be done. The question is how to minimize the material–handling. In this paper we make a case study simulation to determine how to set up a warehouse in this situation and what can be reached by that way.

We have two options, how to influence the material–handling performance, first we can say how to allocate materials in a warehouse. By other words: the required time for a list of tasks is depends on, how the warehouse was look like, when the work was started. The second method is to say, what to do, the incoming materials where should be placed and which one should be given out first.

In this paper we present simulations of the warehouse behavior for random allocations compared to the scientific ABC organized solution. The second method will be covered in other time.

LITERATURE REVIEW

Before the simulations we have to pay attention for state of art, because the Storage Location Assignment Problem (SLAP) is an NP–hard problem and researched by many others. Juan José Rojas Reyes, Elyn Lizeth Solano–Charris and Jairo Rafael Montoya–Torres collected 71 representative papers published in the theme between 2005 and 2017 [6].

The problem is often inspected with Genetic Algorithm (GA), for example it is discussed end enveloped by Changkyu Park & Junyong Seo in [11] and [12] or Jing Xie, Yi Mei, Andreas T. Ernst, Xiaodong Li & Andy Song in [7] and [8]. GA makes generally many computations and last long time. In our simulation it is solved much faster aware of could be less effective. Our research could be a good base for GA computations too.

Other approaches collected by Behnam Bahrami, Hemen Piri & El–Houssaine Aghezzaf in [3]. Problem could be solved by classifying the stored materials that is presented by Ren–Qian Zhang, Meng Wang & XingPan in [5] or R.Micale, C. M. La Fata, G. La Scalia in [4]. In our research ABC analysis is compared to total random allocations.

Our results could be utilized not just in the modelled warehouse, but in many other field where compacted storage systems are preferred, for example in works of Sacramento Quintanilla, Ángeles Pérez, Francisco Ballestín & Pilar Lino [9] or in maritime terminals as shown in works of Xiaoyuan Hu, Chengji Lianga, Daofang Chang & Yue Zhang [2] or Lu Chen & Zhiqiang Lu [10].

SIMULATIONS

— System Description

To solve SLAP we have planned many simulations, they help choose between the solutions. In this paper we represent the first simulations, in what we created with random arrangements for a real situation's reduced model.

Our case study based on data of a factory's raw material inventory. In our model all the materials stored on pallets and use the same size of store location. The FIFO is not a requirement that could be anywhere, if the materials counted in bound of big series and a new serial means a new material number.

The entrance of the warehouse is in opposite of the exit, so the materials' flow has straight line shape, every material moves across the warehouse to production and none of them comes back, so there is no rest material in our model. In real life the materials have buffing area in production, it is unnecessary to move them back, they can be hold there for later production, if it is more than the actual serial required. The finished goods stored in other inventory.

We kept the shape of the rack system that was a drive–in system with 6 pallets deep width and 3 levels height. Materials moved by a forklift and locations are available if there are no other materials in the lane closer to the corridor, but it does not matter on which height it stands. There are two block of racks on the left and right side of the corridor. In each block there are 18 lanes so the maximal static capacity of the model is 648 pallets.

The model inherited the volume ratio of materials in the original inventory. The amount of SKUs is reduced to 100 by selecting every 19th material for simulation, but their volume ratio is almost equal to volume of original inventory's SKU percent volume ratio. That can be seen in Figure 1, how well the Pareto thesis is represented in simulations.

There are 557 materials to starts with. We have to say how to range these items in the warehouse to influence the performance. We don't know what will be, we know only chance what will be a task, so the system is stochastic as randomness of reality, not a deterministic model.

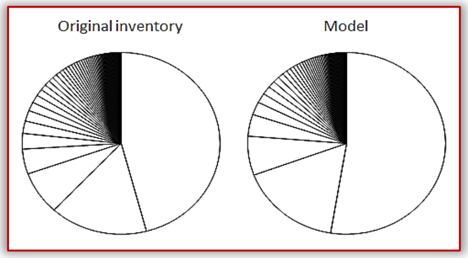

Figure 1. Volume ratio of SKUs

To count the dynamic capacity, we created a table where it is recorded, how many time is needed to move in and out of the rack to or from that exact location. If a needed material is blocked by other materials in the lane, then the others have to be moved on the corridor, then move out the searched one and then move back the others without change the order, just shift a bit deeper. It is not allowed to leave an empty location blocked in any lane. The time needed to make a location free is calculated in table too, depending on the location coordinates and the blocking locations before that.

To evaluate the efficiency of storage location assignment we made simulations with 100 task–lists, each list contains 2000 tasks. Every task could be either get in or a give out. The task lists were build up in aspect of the past and it tries to get the inventory fulfilment about 80% and never ask a material that is not stored in yet. When the task–lists were built, there was higher chance to store in the lower fulfilment and low chance to get something out, and there was lower chance to store in something in higher fulfilment and higher chance to get something out.

We kept the circulate habit of the materials, rate of materials and amount of items should store in and given out, so for example if a material used to come in on 20 pallets, then it generated 20 tasks of the same material to store in followed by each other, before another task were generated.

Each task–list is independent to others, but each were generated by the same chances. The task–lists were recorded to keep it in every simulation, in that case they are comparable.

The exact location is chosen by greedy algorithm: when a unit has to be placed in, the system chooses the shortest way in time to deploy if that is a valid location, and when a unit has to be given out, the system chooses the fastest available unit to give out. There is no restriction between materials and locations every material could be placed on any location.

The value of solutions was calculated with the following formula:

$$v = \frac{\sqrt{\frac{\sum_{i=1}^{n}\left(\sum_{j=1}^{m} t_{ij}\right)^2}{n}}}{m} \qquad (1)$$

where v means the value of a simulation, m is the amount of tasks in a task–list, n is the amount of task–lists and t_{ij} is the time required to do the j task in i task–list. With this formula a weighted average is given for time required to do a task. Behind the weighting stands the same theory a behind the average distance to mean value and deviation. If a solution solved the task–lists with the same average time required in another solution, but it has less deviation, than it get a better value. Of course the aim of the simulations is to minimize v value.

— First run

We generated total random arranges to see, how it can impact the performance of warehouse. The first 1000 solutions were ranged from 90.7 to 101.3 as it is shown in Figure 2.

The ABC analytic solution was 94.4, so the 26% of random arranges were better than this. We made hundred tusk–lists to avoid getting solutions around one exact situation – that could be easily defined by the given order, we wanted to get an approximately good solution for any stochastic–possible situations. For a similar reason it is important to have many tasks in every task–lists, and that helps find solutions for long time.

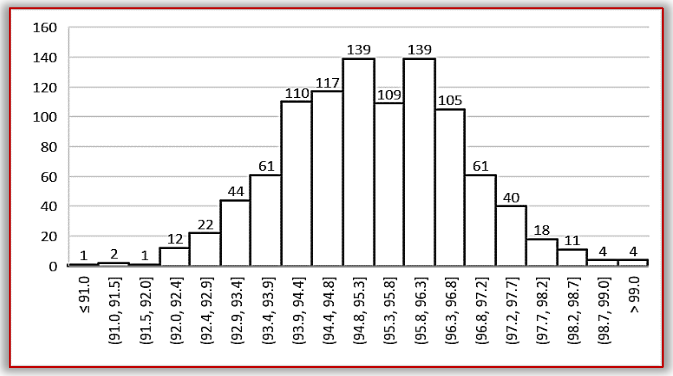

Figure 2. Histogram of simulations by the first 1000 arrangements

The required time for a task could be less than 30 seconds if it could be done near to the corridor, but if a material is deeply covered and 15 positions have to be empty before it and later move the materials from there back, then it could take almost a half–hour. The question is how many times they will occur.

The average operation–time is changing during the list – it could be seen in Figure 3, where we display it for the best, the worst and the median solution. The ABC analytic solution is indicated with red line.

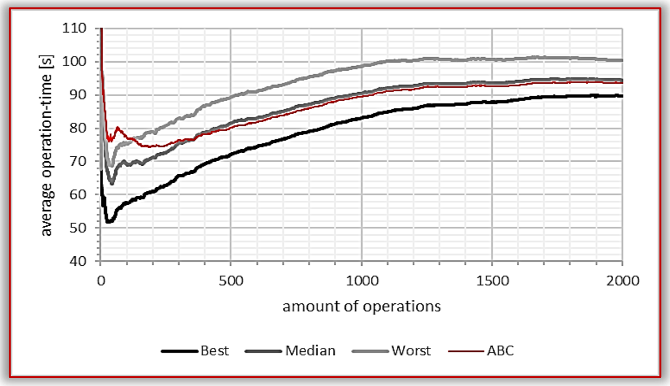

Figure 3. Average operation–time in simulation depending on amount of operations

The first 300–400 tasks' average operation–time is varying a lot, from here to 1100–1600 it is increasing constantly and in the final part it is converging to one value. In the first part they can be judged well, but in the followings every line is going parallel to each other and lastly no big changes are expected, it seems unnecessary to have longer task–lists.

— Validation

To prove the simulations are good to test the effectiveness of arranges after the first thousand simulations we created a new hundred task–lists with 2000 tasks in each as it was written earlier. The simulations' *v* value in the original and the newly generated validating data have to be near to each other. If the result is the same, then the arrange optimization would be independent to task lists, but if the differences are high, then the result is task–list specific.

The new values of the solutions were luckily only 0.32% different from the first run. The biggest difference 1.34% was at the Case 334, as it can be seen between some of the two values are sampled in Figure 4. The vertical axis is for the *v* values and the horizontal axis shows the identification number of cases. The values of the whole 1000 members range are on average 0.20% higher than the original ones.

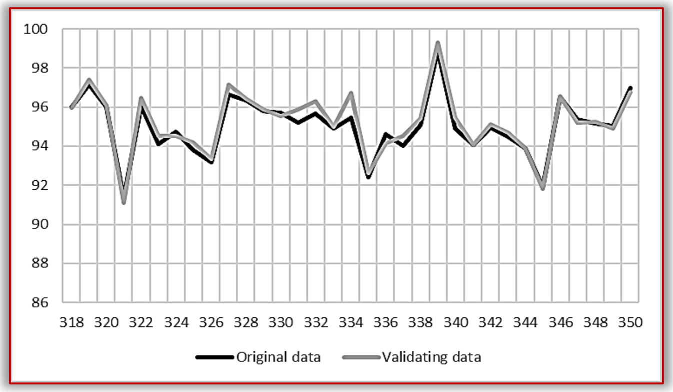

Figure 4. The value of original and validating data around the biggest differences (Case 334)

With this small differences, the method seems to be right, the simulations could be continued for a bigger research with the original task–list.

— Result of ten–thousand simulations

After the validation of the method, we continued the simulations to extend the case–numbers to ten–thousand. The mean value changed from 95.32 to 95.30, the best value was reduced from 90.70 to 90.54, but there was no worse than in the first thousand case, so the worst value didn't change from 101.29.

As it was shown in Fig. 3 for the first thousand simulations, we present the average operation–time changing during the task–lists by the best, the worst and the median solution in Fig. 5 for the extended range.

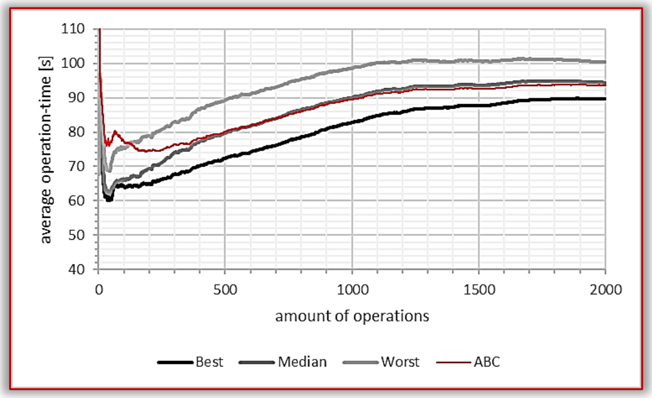

Figure 5. Average operation–time in simulation depending on amount of operations

The three parts are on the same period, the behavior of the lines are also the same, but the best solution comes from much higher value, so it started from a worse position, but the final value becomes better.

The median line goes in the opposite way, it started from a much better first period, but become the same as in case of the first runs, they different only after the 5^{th} digit: the *v* value changed from 95.28212 to 95.28211. These changes make us sure, we had to make all the 2000 tasks in all task–lists.

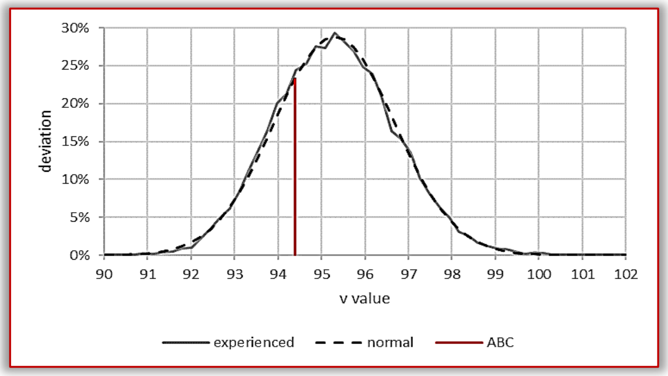

Figure 6. Experienced deviation of simulations' *v* value and normal distribution

The result of the simulations is well described by a normal distribution. According to Kolmogorov–Smirnov test, it has 61% confidence, because the biggest difference between normal and experienced distribution is 0.0090 around 44%. The experienced deviation is indicated with it in Fig. 6. A red line shows, where the ABC is analytic solution and all the solutions on left to it are better. The best experienced solution makes it about 4% better.

CONCLUSION AND FUTURE WORK

We presented the Pareto thesis' impact to utility of rack–systems efficiency in warehouses, collected methods and to SLAP, developed a simulation system for an exact problem, to show out, how important is to pay attention on materials arrangements, validated the method and made 10 000 simulations.

We don't think, that our best random arrangement for the problem is good enough, but this experience proved that it is worth to looking for better solutions. The presented simulations could be a good base to start GA population or could be used for a neural net building.

There are many questions that we would like to answer in the future:

- We would like to build a general model, to make simulations with other rack–systems, other shapes and constructions to optimize the area–utility. What shapes is ideal for a situation and what influence it?
- How many racks should be compact and many conventional ones should be used to optimize area–utility? How deep and how high they should be?
- How would impact the result if the temporary storage on the corridor is prohibited?
- What can we reach if the selection in model would be changed from greedy algorithm to something else?
- What we have to do if we can see further for example ten or twenty tasks and not just one?
- How to arrange the inventory if we would have better solutions in short terms?
- When is it worth to make the calculations for a new arrangement and rearrange the warehouse? Could it be done by new dynamic process?

When we answer these questions, then the supply chains could be better served by the warehouses.

References

[1] Trindade, M. A. M., Sousa, P. S. A. & Moreira, M. R. A., "Ramping up a heuristic procedure for storage location assignment problem with precedence constraints." in Flex Serv Manuf J, 2021
[2] Xiaoyuan Hu, Chengji Lianga, Daofang Chang & Yue Zhang, "Container storage space assignment problem in two terminals with the consideration of yard sharing", in Advanced Engineering Informatics Volume 47, 2021
[3] Behnam Bahrami, Hemen Piri & El–Houssaine Aghezzaf, "Class–based Storage Location Assignment: An Overview of the Literature", Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics – Volume 1: ICINCO, pp. 390–397, 2019
[4] R. Micale, C. M. La Fata, G. La Scalia, "A combined interval–valued ELECTRE TRI and TOPSIS approach for solving the storage location assignment problem" in Computers & Industrial Engineering Volume 135, pp. 199–210, 2019
[5] Ren–Qian Zhang, Meng Wang & XingPan, "New model of the storage location assignment problem considering demand correlation pattern" in Computers & Industrial Engineering Volume 129 pp. 210–219, 2019
[6] Juan José Rojas Reyes, Elyn Lizeth Solano–Charris & Jairo Rafael Montoya–Torres, "The storage location assignment problem: A literature review" in International Journal of Industrial Engineering Computations 10 pp. 199–224, 2019
[7] Jing Xie, Yi Mei, Andreas T. Ernst, Xiaodong Li & Andy Song, "Scaling Up Solutions to Storage Location Assignment Problems by Genetic Programming" in Asia–Pacific Conference on Simulated Evolution and Learning SEAL 2014: Simulated Evolution and Learning pp. 691–702, 2014
[8] Jing Xie, Yi Mei, Andreas T. Ernst, Xiaodong Li & Andy Song "A Genetic Programming–based Hyper–heuristic Approach for Storage Location Assignment Problem" in 2014 IEEE Congress on Evolutionary Computation (CEC) July 6–11, 2014, Beijing, China
[9] Sacramento Quintanilla, Ángeles Pérez, Francisco Ballestín & Pilar Lino, "Heuristic algorithms for a storage location assignment problem in a chaotic warehouse" in Engeneering Optimization Volume 47 – Issue 10 pp. 1405–1422, 2014
[10] Lu Chen & Zhiqiang Lu, "The storage location assignment problem for outbound containers in a maritime terminal" in International Journal of Production Economics Volume 135, Issue 1, pp. 73–80, 2012
[11] Changkyu Park & Junyong Seo, "Comparing heuristic algorithms of the planar storage location assignment problem" in Transportation Research Part E: Logistics and Transportation Review Volume 46, Issue 1 pp. 171–185, 2010
[12] Changkyu Park & Junyong Seo, "Mathematical modeling and solving procedure of the planar storage location assignment problem" in Computers & Industrial Engineering Volume 57, Issue 3 pp.1062–1071, October 2009

ISSN: 2067–3809