DE GRUYTER OPEN

Logistics & Sustainable Transport
Vol. 9, No. 2, October 2018, 37-45
doi: 10.2478/jlst-2018-0008

Comparison of Lowest-Slot and Nearest-Stack Heuristics for Storage Assignment of Steel Bar Sets

Jakob MAROLT*[1] and Tone LERHER[1]

[1] University of Maribor/Faculty of Logistics, Celje, Slovenia

[Corresponding Author indicated by an asterisk *]

***Abstract*—Our research objective is to lower intralogistics costs by minimizing the number of shuffling operations in a steel plant company commercial warehouse. The process of dispatching products consists of retrieving set of steel bar (SSB) from a floor stored stack or a special stacking frame by an overhead crane. To retrieve a targeted merchandise all SSB above targeted must be reshuffled. Proper assignment of storage locations is a key logistics problem for efficient order picking. We are comparing two heuristics, that do not require information of dispatching sequence of any stored products. We simulated the problem at hand with both methods. Our objective is to count the number of reshuffles using each heuristic on randomly generated examples and decide which is better in the long run. Our problem has similarities with storage assignment of steel plates or steel coils for minimization of reshuffling operations. The problem is also comparable to storage assignment of containers in a container yard. In our case we are dealing with a special stacking configuration of products, that demands different approach. We want to demonstrate which heuristic should be used in companies that lack necessary storage information infrastructure.**

***Index Terms*—order picking, shuffling operations, sets of steel bars, storage assignment.**

I. INTRODUCTION

All shuffling problems have in common rearrangement of items in a given configuration for possible access. Top items in a stack are reachable, while items below are not. There are two possible strategies to solve shuffling problem: pre-marshalling (a posteriori) and reshuffling (a priori). Placing all objects in exact sequence for later dispatchment, without the need for further reshuffling is called pre-marshalling. Rearranging only items above targeted item is called reshuffling. The advantage of pre-marshalling is that items can be quickly retrieved from a stack, but all items must be moved at least twice. In this paper pre-marshalling in not considered.

Order picking operations in a steel plant company consists of retrieving sets of steel bars (SSB) from a stack. SSB are moved by an overhead crane. To retrieve a targeted SSB inside a stack all sets above targeted must be reshuffled to other stacks. Shuffling operations do not add value but are necessary to order pick a specific product. Their contribution is estimated to over a half of total warehouse expenditure [5]. The key to the problem is to determine the stack to which reshuffled SSB should be reshuffled to, without the information of dispatch priority arrangement among SSB.

We could not find any previous references for this exact problem, but the work has been done on similar problems.

Kim et.al [4] presented paper on minimizing the number of relocations during pickup operation for block stacking. Two methods were presented. Firstly, Branch and bound algorithm and secondly, a decision rule heuristic based on probability with estimator for an expected number of additional relocations.

In steel plant industry Tang et. al. [6] studied item shuffling (IS) problem. The authors described two study cases, plate shuffling problem and coil shuffling problem. Linear program was formulated, and some special cases were studied, where they obtained algorithms with polynomial time. They created a greedy heuristic and later improved it with tabu search.

Wan et. al. [7] studied storage assignment of containers in a container yard. The authors formulated integer problem and later derived few heuristics. They dealt with both static problem to empty a given stack, without any new container arrival and dynamic, where they simulated arrival of new containers.

Logistics & Sustainable Transport
Vol. 9, No. 2, October 2018, 37-45
doi: 10.2478/jlst-2018-0008

König et. al. [3] studied pre-marshalling problem of steel slab stacks. The items had to be moved in a specific time window. The goal was to rebuild slab stacks with minimal number of shuffles, so no further reshuffling was needed.

We are dealing with a comparable problem that has two different storing configurations and no information of the complete order picking sequence is demanded. Order picker must know only, where the next item is to be dispatched.

This paper is structured as follows. In section two the problem is described in detail. In section three both heuristics are defined. In section four, results are presented, followed by conclusions in section five.

II. PROBLEM DESCRIPTION

Steel bars are of various longitudinal sizes ranging from 3 to 6 meters. In cross section they can be either circular or rectangular shape. At the end of production line, bars with same attributes are wrapped together by wraps into a set. Set of steel bars (SSB) are floor stored in a special configuration or are stacked into a special stacking frame as shown in Figure 1.

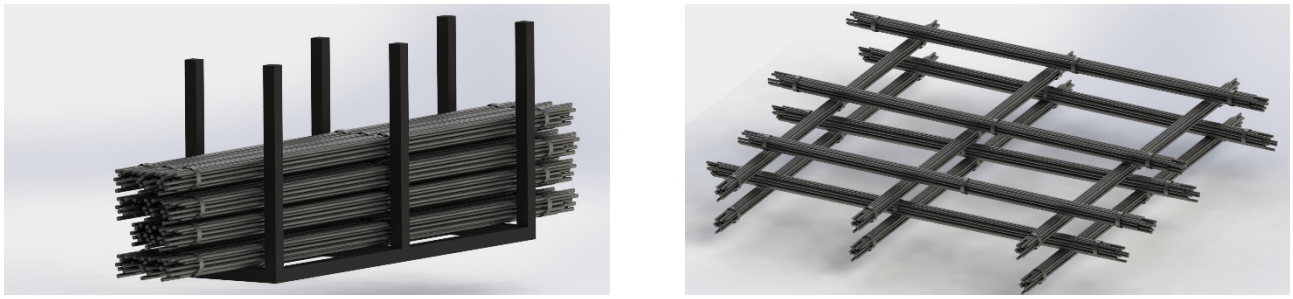

Figure 1: SSB stored in a special stacking frame (left), SSB floored stored in a perpendicular configuration (right)

Floor stored SSB in perpendicular configuration are placed down parallel to one another approximately 1-2m away from each other. Every next tier has the same configuration as first but is rotated perpendicularly to previous. In that manner we can achieve a configuration as shown in Figure 1 (right). Maximum height of a stack is six tiers. Anything above six tiers would be dangerous for workers inside the steel plant and can cause damage to the shape of steel bars.

Other possibility is a special stacking frame, where SSB are all longitudinal orientated in the same way. There is no required space between SSB for stable configuration as in floor stored configuration. Each tier can store three SSB. Tiers are placed parallelly on one another. Maximum height of a stack is theoretically the size of stacking frames that give support, but in practice the height of a stack is usually the same as in floor stored configuration. Special stacking frame has also better space utilization in a warehouse area as floor stored configuration but requires additional infrastructure.

Mathematically it does not matter which storing configuration is used, because SSB are retrieved from both configurations in the same method. In order to retrieve a targeted SSB all SSB above targeted must be reshuffled. The algorithm for reshuffling is graphically presented in Figure 2. It contains six logical steps, as follows:

Logistics & Sustainable Transport
Vol. 9, No. 2, October 2018, 37-45
doi: 10.2478/jlst-2018-0008

1.) Try to retrieve a SSB with highest priority
2.) Return all reachable SSB
3.) Select a SSB
4.) Return all genuine empty slots
5.) Select an empty slot
6.) Move selected SSB to an empty slot

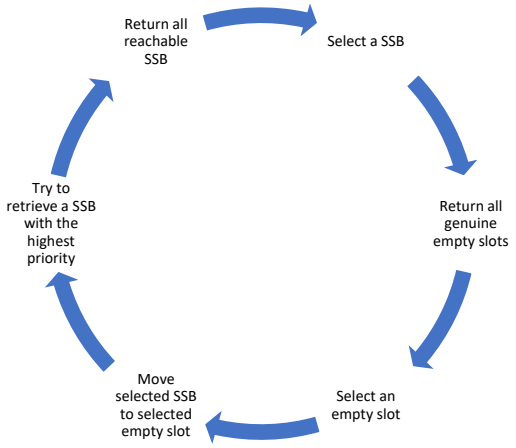

Figure 2: Logical steps of an algorithm

In the first step algorithm tries to retrieve SSB from a matrix. This can be done, if no SSB with lower priority is blocking SSB with the highest priority. In the second step algorithm returns positions of all possible SSB that can move without further shuffling. These are SSB that are stored at the top of each stack. Note that SSB can be moved only if the whole row above its position is empty. In the third step algorithm selects a SSB that will be moved. Both heuristics work in the same manner. SSB to be moved is always top SSB (from left to right in a row) in a stack that contains SSB with lowest total priority. In the fourth step algorithm return all genuine empty slots. Firstly, it looks for all empty slots inside the matrix, that can store SSB (no levitation is allowed). Next it discards empty slots that are in the same stack as selected SSB to be moved. In the fifth step an empty slot is selected. At this point the protocols of both heuristics diverges. LS heuristics chooses empty position that is in a stack which has most empty slots. NS heuristics chooses a stack that is nearest to the stack of selected SSB to be moved. Because of many possible layouts of stacks, random function is used for selection of nearest stack. In the last-six step selected SSB is reshuffled to the empty slot preferred by each heuristic.

Storing configuration has three attributes that describe its size. Number of tiers in a stack, number of SSB in a tier and number of stacks stored in a warehouse. These attributes are defined by the size of an array. Rows in an array represent tiers in a stack, columns represent number of SSB in a tier and aisles of an array represent number of stacks stored in a warehouse. Positive numbers in an array represent priority amongst SSB for dispatch, zeros or negative numbers represent empty space. Example of an optimal reshuffling operation for simplified case is presented in Figure 3. As of now, no exact solution is known to authors on how to solve reshuffling operation optimally without examination of all possible outcomes of every single reshuffle. As number of SSB increases method for examination of all possible outcomes quickly becomes unfeasible, yet no proof of NP completeness of this exact problem was found. Avriel et. al. [1] proved that similar problem that deals with a stowage plan for container ship is NP-complete. They also displayed relationship between the stowage problem and colouring of circle graphs problem.

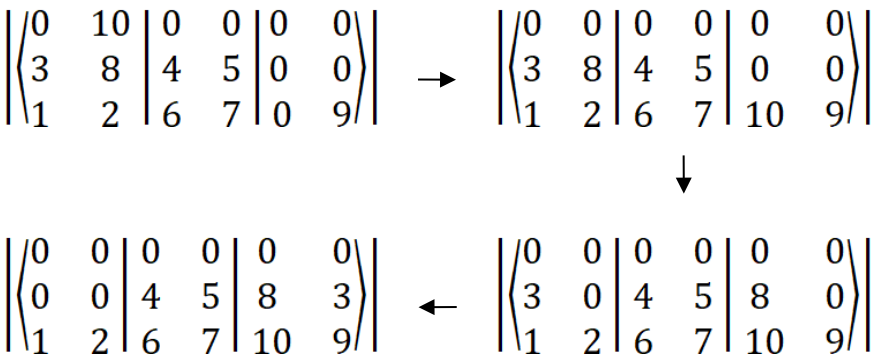

Figure 3: Optimal solution of simple example

Logistics & Sustainable Transport
Vol. 9, No. 2, October 2018, 37-45
doi: 10.2478/jlst-2018-0008

In this article we compared two heuristics for reshuffling SSB problem. Both compared heuristics are solvable in real time. This paper is mostly meant for companies that lack sufficient infrastructure or IT support to determine an order by which SSB will be dispatched from warehouse. Because both heuristics do not require information about order picking schedule. To fully describe working environment of the scheme following assumptions are specified.

Assumptions

1. The problem at hand is assumed to be a static. Meaning, no new set of steel rods is being produced from production line while unloading is at play.
2. Slab stacks consists of 6 x 3 (tiers x number of SSB in tier) configuration because of safety considerations.
3. All stored SSB have assigned priority number that defines an order picking schedule of retrieval. Priority numbers are natural numbers from 1 to N. SSB with lowest natural number, has highest priority. These will be the first item to be dispatch from warehouse.
4. Priority numbers are unknown to the shuffling operation worker/system. SSB with lower priority than 1, can also change priority while reshuffling is in the process.
5. To solve the problem all SSB inside a warehouse must be dispatched.

III. PROPOSED HEURISTICS FOR STORAGE ASSIGNMENT

The whole logical cycle of reshuffling operations takes six steps as shown in Figure 2. Our observed two heuristics deal with storage assignment, the fourth step of an algorithm. Heuristics choose one empty slot that should be used among all genuine empty slots.

A. Lowest-Slot heuristic

Lowest-Slot (LS) heuristic is used in some container yards without advanced information infrastructure [7]. It selects an empty slot from a stack that has the most empty slots. In this manner all SSB should be evenly spread across the stacks in the warehouse. It can occur, that more than one stack has the same number of empty slots. When this happens in real world the nearest stack should be selected. When this phenomenon occurs in our simulation, nearest stack is replicated by a random selection of a stack among the group with the same number of empty slots. As mentioned before shuffling operations of containers in a container yard is just a special case of shuffling operations of SSB. It was shown by Zhang [2] that the LS heuristic performs optimally if a container yard uses FIFO rule in each column separately.

To understand LS heuristic, Figure 4 displays the whole process of reshuffling SSB on a simple example. First, we must reshuffle a stack on the far left of a matrix, to reach a SSB with priority 1. Top most SSB has priority 10 which is the initial move. It is moved to the third stack, because it has 3 empty slots which is greater than second stack, which has 2 empty slots. Next SSB to be moved has a priority 8. Using LS heuristic, it can be moved either to second or third stack, because they have the same number of empty slots. Both scenarios are equally likely with 50% probability. With the same logic we can continue until we reach a state, where all SSB can be dispatched without further need of reshuffles. In the scenario 1 total number of reshuffles is 4. In scenario 2 total number of reshuffles is 3.

Logistics & Sustainable Transport
Vol. 9, No. 2, October 2018, 37-45
doi: 10.2478/jlst-2018-0008

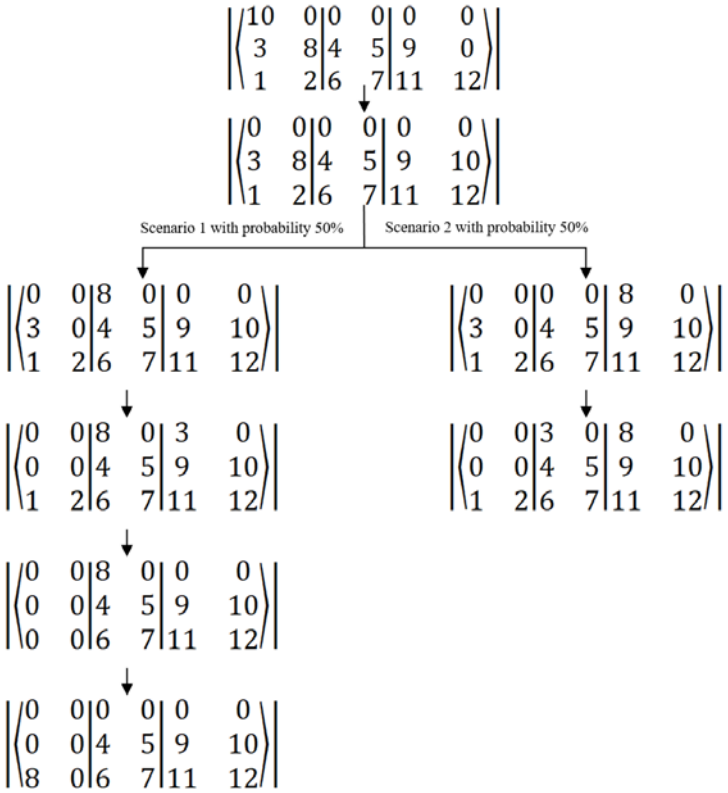

Figure 4: Process of reshuffling three stacks with LS heuristics

B. Nearest-Stack heuristic

Nearest-Stack (NS) heuristic tries to imitate the selection process of an empty slot by crane workers in a warehouse. They usually choose a stack that is nearest, in such manner all traveling distances of a crane are short. We simulated this decision-making process by a random selection of a stack. In the third step, algorithm returns all genuine empty slots. Empty slots, that have the same stack coordinates as selected SSB, are discarded. Further, when random generator selects a stack in a fourth step, coordinates of selected stack stay fixed until the stack runs out of empty slots or SSB with highest priority can be dispatched. When this occurs, new random stack is selected from a group. This selection process continues until all SSB are dispatched from a given configuration. Randomness in decision-making process ensures that there will be both good and bad decisions.

To understand NS heuristic, Figure 5 displays the whole process of reshuffling SSB on a simple example. Firstly, we start by selecting SSB with priority 10, because it is the first SSB to be moved, that is above SSB with highest priority. It can be moved either to second or third stack. Random generator selects one of the possible stacks. There is 50% chance for either of scenarios 1 or 2, to occur. For scenario 2 there is only one possible way, how reshuffles will resolve. For scenario 1 there is one more fork after SSB with priority 1, 2 and 3 are dispatched. Therefor there are in total three end scenarios (1.1, 1.2 and 2) how NS heuristic resolves given configuration.

Logistics & Sustainable Transport
Vol. 9, No. 2, October 2018, 37-45
doi: 10.2478/jlst-2018-0008

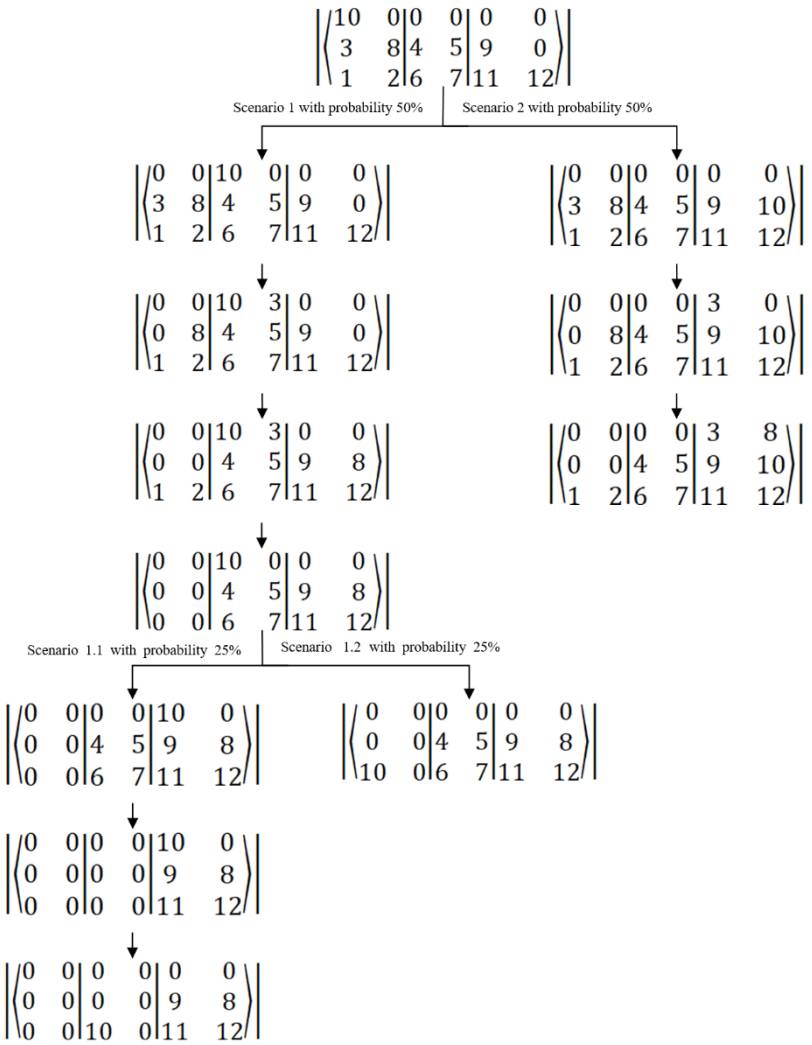

Figure 5: Process of reshuffling three stacks with NS heuristics

IV. RESULTS

We ran many numerical simulations of both LS and NS heuristics. Each iteration we generated 30 random arrays and specified the number of stacks and the number of empty slots. In this manner we compared the performance on four different occupancy levels, by changing the number of empty slots and at each occupancy level we simulated different number of stacks ranging from 2 to 20. The maximum number of tiers in a stack was 6, as is standardised in a steel plant company. In each row there were 3 openings for either empty slots or SSB.

For each repetition we calculated the average number of reshuffles and one standard deviation from the average. The results from numerical simulations are represented in Table 1.

Logistics & Sustainable Transport
Vol. 9, No. 2, October 2018, 37-45
doi: 10.2478/jlst-2018-0008

Table 1: Results of numerical simulation for four different occupancy levels

33% OCCUPANCY	NS HEURISTICS	NS HEURISTICS	LS HEURISTICS	LS HEURISTICS
NUMBER OF STACKS	Average	Standard deviation	Average	Standard deviation
2	10.6	4.2	10.5	3.7
3	15.8	5.1	11.1	3.9
4	19.3	7.8	13.9	4.5
5	20.6	6.3	17.6	4.6
6	24.8	7.7	20.0	5.5
7	33.2	11.1	21.7	4.8
8	40.8	12.8	24.6	5.8
9	47.6	15.9	25.4	6.1
10	50.3	14.1	29.6	6.1
11	50.5	11.6	30.3	7.1
12	56.7	14.4	36.3	5.8
13	61.5	17.0	35.4	6.8
14	62.1	14.0	41.5	6.4
15	67.7	18.7	41.9	4.7
16	73.0	16.4	46.3	6.0
17	83.4	19.5	46.4	7.4
18	84.5	18.2	50.5	7.8
19	91.7	20.2	52.9	8.5
20	96.2	18.8	55.2	7.5

50% OCCUPANCY	NS HEURISTICS	NS HEURISTICS	LS HEURISTICS	LS HEURISTICS
NUMBER OF STACKS	Average	Standard deviation	Average	Standard deviation
2	24.3	8.7	25.5	8.0
3	44.4	12.4	33.1	9.3
4	58.0	17.7	44.2	8.3
5	72.5	20.1	54.0	9.9
6	88.1	16.6	62.1	10.7
7	112.9	19.7	71.7	10.6
8	122.0	25.3	84.1	11.3
9	138.8	23.8	91.0	12.8
10	162.1	25.2	100.9	10.4
11	172.0	21.5	110.9	13.6
12	171.8	27.7	114.6	17.2
13	199.4	26.4	126.0	15.0
14	216.5	32.5	138.0	14.1
15	216.6	29.2	148.1	15.1
16	249.6	35.1	155.3	15.2
17	254.0	33.7	166.0	18.1
18	274.9	44.9	176.3	15.1
19	293.9	32.0	181.9	17.6
20	312.9	36.5	193.0	15.2

67% OCCUPANCY	NS HEURISTICS	NS HEURISTICS	LS HEURISTICS	LS HEURISTICS
NUMBER OF STACKS	Average	Standard deviation	Average	Standard deviation
2	31.4	24.9	23.1	21.8
3	86.6	17.1	74.8	14.1
4	110.0	21.3	95.9	14.4
5	142.1	24.6	114.5	17.6
6	177.6	27.1	133.7	15.6
7	196.0	32.8	152.0	18.1
8	231.3	40.2	174.3	24.6
9	265.8	37.8	194.8	23.7
10	282.0	28.9	224.0	25.4
11	320.6	39.0	235.6	23.4
12	350.1	43.3	256.7	25.2
13	376.9	38.3	283.5	24.7
14	411.3	44.4	290.9	22.3
15	452.7	42.7	318.5	25.9
16	479.4	54.3	341.0	31.7
17	498.1	46.3	362.2	25.7
18	534.7	32.6	387.9	33.7
19	545.7	52.2	404.9	32.5
20	585.1	55.3	415.9	29.5

83% OCCUPANCY	NS HEURISTICS	NS HEURISTICS	LS HEURISTICS	LS HEURISTICS
NUMBER OF STACKS	Average	Standard deviation	Average	Standard deviation
2	5.1	2.5	6.0	3.8
3	15.7	28.4	30.1	40.7
4	86.0	79.3	73.8	72.9
5	234.0	31.2	197.6	21.8
6	285.2	34.4	232.0	25.5
7	328.4	33.0	268.4	31.5
8	375.2	36.4	304.6	30.2
9	421.2	35.5	348.8	26.8
10	475.5	43.1	382.8	28.4
11	518.1	47.5	419.7	32.1
12	560.6	43.9	457.9	36.6
13	608.5	54.0	501.8	35.9
14	654.4	46.7	525.6	38.2
15	701.5	51.6	566.8	39.7
16	770.4	55.6	601.8	37.0
17	805.8	62.4	628.7	43.6
18	845.2	57.0	673.8	45.9
19	898.0	66.5	723.8	35.4
20	921.5	69.9	743.5	35.4

From the results it is clear that LS heuristic outperforms NS heuristic. For occupancy level of 33% the relative average reduction of number of reshuffles is 56%. At 50% occupancy level reduction of number of reshuffles is 48%. At 67% occupancy level there is 33% reduction and at 83% occupancy level there is on average 17% reduction. The higher the occupancy of the warehouse the lower is the reduction of reshuffles. The results are displayed in Table 2.

Table 2: Outperformance of LS heuristic against NS heuristic

NUMBER OF STACKS	OCCUPANCY LEVEL 33%	OCCUPANCY LEVEL 50%	OCCUPANCY LEVEL 67%	OCCUPANCY LEVEL 83%
2	1%	-5%	36%	-14%
3	43%	34%	16%	-48%
4	39%	31%	15%	16%
5	17%	34%	24%	18%
6	24%	42%	33%	23%
7	53%	57%	29%	22%
8	66%	45%	33%	23%
9	87%	53%	36%	21%
10	70%	61%	26%	24%
11	67%	55%	36%	23%
12	56%	50%	36%	22%
13	74%	58%	33%	21%
14	50%	57%	41%	25%
15	62%	46%	42%	24%
16	58%	61%	41%	28%
17	80%	53%	38%	28%
18	67%	56%	38%	25%
19	73%	62%	35%	24%
20	74%	62%	41%	24%
ON AVERAGE	56%	48%	33%	17%

The results from Table 1 are graphically represented in Figure 5. The green line with diamond pattern represents NS heuristic, the blue line with square patterns represent LS heuristic. Both heuristics have linear trend line of complexity growth.

Figure 6: Graphical representation of results for four different occupancy levels

V. CONCLUSIONS

This study addressed the problem of reshuffling SSB in a steel pant company that does not have storing information system. Two approaches were described on assigning storage locations for reshuffled SSB. The performance of both LS and NS heuristics were compared on many numerical experiments. Simulation was done on four different occupancy levels of a warehouse. For each occupancy level the number of stacks ranged from 2 to 20. The comparisons showed that LS heuristics outperforms NS heuristics by 56% on average for occupancy level of 33%, 48% on average for occupancy level of 50%, 33% on average for occupancy level of 67% and 17% on average for occupancy level of 83%.

The problem dealt with was static. Meaning that no new SSB came in to a warehouse from a production line. Further research topic should include dynamic problem, also other possible configurations of items should be studied. All these decision-making problems for storage location assignment of SSB will be challenging topics for future studies.

REFERENCES

1. Avriel, M., Penn, M., & Shpirer, N. (2000). Container ship stowage problem: complexity and connection to the coloring of circle graphs. Discrete Applied Mathematics, 103(1), 271–279.
2. C.Q. Zhang, J.Y. Liu, Y.-w. Wan, K.G. Murty, and R.J. Linn, Storage space allocation in container terminals, Transport Res Part B: Methodological 37 (2003), 883-903
3. F.G. König, M. Lübbecke, R. Möhring, G. Schäfer and I. Spenke, "Solutions to real-world instances of PSPACE-complete stacking" in: Proceedings 15th European Symposium on Algorithms, Eilat, Israel, 2007, pp. 729-740
4. Kim, K. H., & Hong, G.-P. (2006). A heuristic rule for relocationg blocks. Computers & Operations Research, 33(4), 940–954.
5. R. De Koster, T. Le-Duc and K.J. Roodbergen, Design and control of warehouse order picking: A literature review, Eur J Oper Res 182 (2007), 481-501
6. Tang, L., Zhao, R., & Liu, J. (2012). Models and algorithms for shuffling problems in steel plants. Naval Research Logistics, 59(7), 502–524
7. Wan, Y., Liu, J., & Tsai, P.-C. (2009). The assignment of storage locations to containers for a container stack. Naval Research Logistics, 56(8), 699–713

AUTHORS

A. Jakob Marolt is an assistant at the Faculty of logistics, University of Maribor, Celje, Slovenia (e-mail: jakob.marolt@um.si)

B. Tone Lerher is a full professor at the Faculty of logistics, University of Maribor, Celje, Slovenia and an associate professor at the Faculty of mechanical engineering, University of Maribor, Maribor, Slovenia (e-mail: tone.lerher@um.si)