Order Picking Operation and Warehouse Layout Optimization in a Textile Industry by Using Genetic Algorithm

Maxsuel Vieira Orti and Raimundo C. G. Teive
Laboratory of Applied Intelligence
University of Vale do Itajaí (UNIVALI)
Itajaí, SC, Brazil
maxsuel@edu.univali.br, rteive@univali.br

Abstract

The optimization of Warehouse Layout and Order Picking are essential to warehouse management in companies trying to follow the trend of one-day shipping. This paper studies a Textile Industry with this objective, proposing a change in warehouse layout and applying a Genetic Algorithm (GA) with the objective of optimizing the order picking operation. The proposed GA uses sales data and data mining to minimize the distance between the company`s bestselling items. The results have demonstrated the feasibility of this approach, supplying two optimal warehouse layout configurations with minimal distances between the most sold items.

Keywords
Genetic Algorithm, Order Picking, Textile Industry, Warehouse Layout Optimization

1. Introduction

Warehouse operations are a critical part of logistics and its optimization can lead to lower costs and better productivity. Warehouse operations have a strong influence on both investment and direct operational costs (Chen et al. 2015, Pan et al. 2018). Same-day shipping is one of the objectives for business that are trying to follow a trend set by companies like Amazon (Giannikas et al. 2017). Considering the fast development of logistics, the use of conventional storage method is insufficient to meet the demands of production and transportation (Liu et al. 2014). The optimization of both inventory and storage allocation is an important issue for warehouse management (Gibson et al. 2016). Thus, it is known that to meet customers demands and apply a same-day shipping policy, the optimization of warehouse operations is paramount in a company. One of the warehouse operations that can be optimized is the Order Picking (OP)

Order Picking is a critical warehouse operation. This activity handles the retrieving of items from their storage location to satisfy customers orders (Henn, 2012). This operation is repeated for each order and may be influenced by sectoral requirements, customer requests, order structures, transportation routes and warehouse layout (Yener, 2019). OP usually involves at least four actions: pre-action, searching, picking, transport and other (Chen, 2014). OP is one of the most labor-intensive operations in a warehouse (Koster, 2007) and it is estimated that OP can be up to 70% of the costs of a warehouse (Chen, 2014). Thus, the optimization of OP can reduce the operational costs and decrease the average time of shipping. One of the approaches to optimize OP is the utilization of a Genetic Algorithm (GA).

Metaheuristics, mostly Genetic Algorithm (GA), has been used to solve optimization problems related to order picking and warehouse management. Ardjmand et al. (2019) applied a GA to optimize order picking in a put wall-based picking system. Pan et al. (2018) uses GA and a simulation model to improve layout planning in a warehouse. Beroule et al. (2017) implemented a GA to improve routing and items allocation in a warehouse of a hospital pharmacy. Chen et al. (2014) developed a hybrid solution using GA and Ant Colonization Optimization to solve an integrated order batching, sequencing, and routing problem. As far as we know, there are no publications of GA applied to OP considering the limitations of a textile industry.

The proposed study differs from the related works in three aspects: case scenario, problem modelling and solution method. On Ardjmand et al. (2019) and Beroule et al. (2017), the major challenger is related to the order batching and picker route, both part of order picking operation, focused in the specific industry. Pan et al. (2018) propose a mathematical approach to minimize carrying costs and efficiency of out-put and in-put of warehouse using GA while Chen et al. (2015) work on order batching, order sequencing and picker route. Hence, we will use a datamining-oriented approach to apply a GA in the textile industry to optimize the OP operation by minimizing the distance between the bestselling items. This minimization will directly impact the time needed to pick an order and will allow for faster shipping times.

2. The Context of the Study

The studied company is a market leader industry in Brazil. It has a production capacity of 10000 rolls of knitting fabric per month and is specialized in products made of Viscose and Polyester yarn. The warehouse has 11250m² and should have one month of production stored. The bestselling products are made of Viscose and sold in 50 different colors (variants). The product made of Polyester is sold in about 30 different colors. The company has been growing throughout the years and the same-day shipping policy became an increasing request by customers, but the current OP and Warehouse Layout has been inefficient in attending this demand.

The products are stored in pallets that can have up to 100 rolls and should be organized by color. The problem is that this usually does not happen, because the people who are responsible for storing the goods, do it in a random way, without any policy or guideline. In fact, each employee stores the items following their own logic and most of the time not communicating with each other. This lack of management presents numerous challenges that impacts the average time of order picking.

Thus, a new Warehouse Layout will be presented followed by an Order Picking policy based on allocating the bestselling items together. We expect that this strategy will lead to a minimization of the average OP time and make it possible to apply the same-day shipping policy.

2. Modeling

To optimize the Order Picking we proposed to the company: an identification system; storage allocation policy; minimum stock per product and color. With that in place, we used a dataset based on Association Rules to find which color should be close to each other. The last step was to work on the GA by designing the chromosome, the selection operator, fitness function, crossover operator, and mutation operator.

2.1 Identification System and Storage Policy

Items should be stored with an identification of pallets, pallets rows, and aisles. The pallet identification will be based on its location, following a sequential number of aisles, rows, and pallets. The identification system can be abbreviated for: aisle: A; rows of pallets: R; pallets: P; color: C. For example, a pallet can be identified as A:01 R:01 P: 01 C: black, meaning that this pallet is in the aisle 01, row 01, pallet 01 and has color equals to black. Figure 1 presents an example of the identification system.

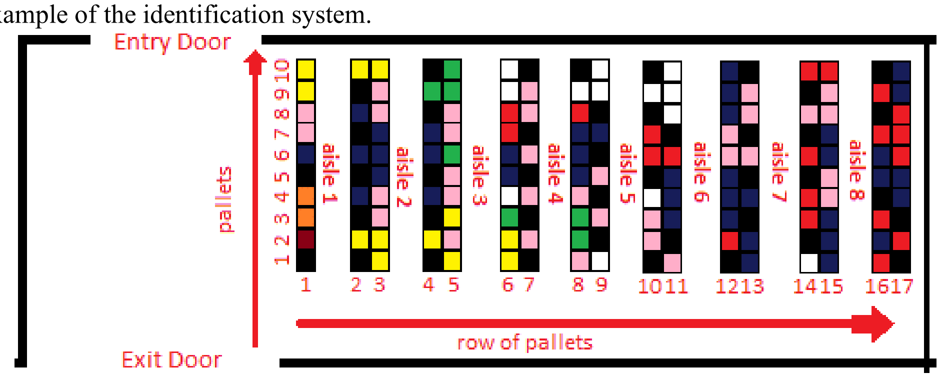

Figure 1. Identification Example

This identification system allows a fitness function to calculate the distance between two pallets using rows, aisles, and pallets numbers.

2.2 Color Selection and Minimum Stock Level

For the purpose of the study, we established that 10000 rolls of knitting fabric should be stored, and we organize it using ten pallets per row and ten rows of pallets. Additionally, following management guidelines, we are studying the products made of Viscose and the colors who sell more than 100 rolls per month. The number of pallets per color was fixed based on sales proportion, i.e, a color who is responsible for 8.5% of the sales will have 9 pallets. To make sure that most of the colors would have a minimum stock, we decreased the number of pallets from the bestselling item. The full distribution is shown at Table 1.

Table 1. Sales and Pallets Stored per Color

Color	Sales	Pallets	RGB	Color	Sales	Pallets	RGB
Preto	0.345	31		Campos	0.015	2	
Marinho Black	0.085	9		Gold	0.014	1	
Picadilly	0.075	8		Crystal	0.012	1	
Branco	0.070	7		Empire	0.009	1	
Pink	0.043	4		Marinho	0.008	1	
Jade	0.040	4		Brownie	0.006	1	
Beterraba	0.035	4		Roxo	0.006	1	
Marfim	0.032	3		Risk Red	0.006	1	
Azul Chic	0.027	3		Oxford Blue	0.006	1	
Festival	0.026	3		Flan	0.006	1	
Rosa Barroco	0.021	3		Macaron	0.006	1	
Perola	0.020	2		Fraser	0.006	1	
Coral	0.019	2		Viera	0.004	0	
Ink Blue	0.017	2		Atlantis	0.004	0	
Paris	0.016	2		Astana	0.004	0	

The color name follows a company naming convention and it may not have any meaning for an outsider. In Table 1 is presented the color name, the sales proportion, the number of pallets and the RGB identification of that color name. Additionally, we decided to not have a standard deviation greater than 4% in relationship between pallets and sales proportion, and due to the experiment size, 3 out of 30 colors will not have any pallets in the warehouse.

2.3 Knowledge Discovery in Databases (KDD)

The studied company has applied Knowledge Discovery in Databases (KDD) and the topic has been discussed at length in Orti and Martins (2018). This study replicates what is presented at Orti and Martins (2018) with some minor adjustments to the dataset. We have changed the prior studied and the attributes selected in the database. For the sales period, we are using the sales data from 01/01/2017 to 12/22/2018. The attributes selected were invoice id, product id, color, and weight.

We are using the RapidMiner as tool for applying the FP-GROWTH algorithm and using the CREATE ASSOCIATION RULES operator to present which bestselling items are sold together. The association rules are configured to consider only rules with support above 7% and confidence above 75%. This configuration found 30 rules that are presented at Table 2.

Table 2. Association Rules

Rule	Premise 1	Premise 2	Premise 3	Conclusion	Support	Confidence
1	Branco			Preto	0.29	0.79
2	Marinho Black	Branco		Preto	0.17	0.86
3	Marinho Black	Picadilly		Preto	0.16	0.80
4	Branco	Picadilly		Preto	0.15	0.85

5	Marinho Black	Pink		Preto	0.12	0.80
6	Picadilly	Pink		Preto	0.12	0.80
7	Picadilly	Jade		Preto	0.12	0.80
8	Branco	Pink		Preto	0.11	0.84
9	Marinho Black	Jade		Preto	0.11	0.81
10	Branco	Jade		Preto	0.10	0.83
11	Pink	Jade		Preto	0.10	0.78
12	Marinho Black	Branco	Picadilly	Preto	0.10	0.87
13	Marinho Black	Beterraba		Preto	0.10	0.77
14	Marinho Black	Azul Chic		Preto	0.09	0.79
15	Picadilly	Azul Chic		Preto	0.09	0.77
16	Marinho Black	Picadilly	Pink	Preto	0.08	0.83
17	Marinho Black	Picadilly	Jade	Preto	0.08	0.84
18	Preto	Marinho Black	Jade	Picadilly	0.08	0.77
19	Marinho Black	Festival		Preto	0.08	0.82
20	Branco	Azul Chic		Preto	0.08	0.85
21	Branco	Marfim		Preto	0.08	0.84
22	Marinho Black	Marfim		Preto	0.08	0.82
23	Picadilly	Pink	Jade	Preto	0.08	0.82
24	Branco	Beterraba		Preto	0.08	0.79
25	Picadilly	Beterraba		Marinho Black	0.08	0.77
26	Picadilly	Beterraba		Preto	0.08	0.76
27	Marinho Black	Branco	Pink	Preto	0.08	0.87
28	Branco	Picadilly	Pink	Preto	0.08	0.86
29	Branco	Picadilly	Jade	Preto	0.08	0.87
30	Branco	Festival		Preto	0.07	0.83

2.4 Genetic Algorithm

The GA was made using Python 3.6 and it was created a data structure that made all GA operators available without having to relay on conversion to a binary table. We created a class Population, which attribute is the individuals who are member or a population. Them, we have the Individual class, which is the solution of the problem and has a list of pallets and a set of association rules. The pallet class has all the attributes related to the storage position of a product and its color. The color class has the sales data and the number of pallets per color. The DM class has the structure for the association rules. Figure 2 present the data structure.

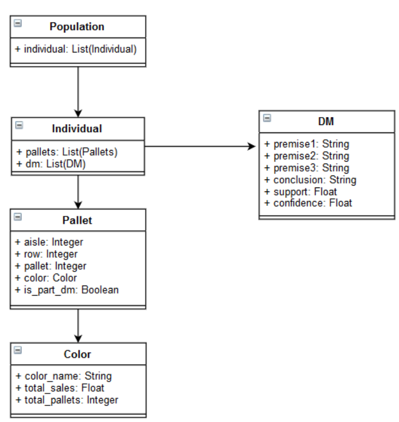

Figure 2. Data Structure

This data structure allows for the code to iterate over every individual in the population that can access every pallet, the colors and the association rules. All the GA operator can go over the list and do the changes needed without any correlation table or the utilization of binary numbers to represent data.

2.4.1 Fitness Function

A Fitness Function was created with the objective of minimizing the distance between the colors that are sold together. This minimization can be achieved by calculating the distance between all the premises of the Association Rule against the conclusion. In an example, given a premisse1 that has color white and a conclusion that has color black, the function will look for every pallet that has the color white and then looks for every pallet with the color black. Afterward, every pallet that has the white color will have its distance calculated against every pallet that has the black color. Table 3 shows a Python code of the distance between two pallets was calculated.

Table 3. Calculating the Distance between two Pallets

```
def distance_pallets(self, pallet1, pallet2):
    distance_aisle = abs(pallet1.aisle - pallet2.aisle)
    distance_rows = abs(pallet1.row - pallet2.row)
    distance_pallets = 0
    if distance_aisle == 0:
        distance_pallets = 1 + abs(pallet1.pallet -
pallet2.pallet)
    elif distance_aisle > 0 and distance_rows > 0:
        distance_pallets = (1 + abs(pallet1.pallet -
pallet2.pallet)) * distance_rows
    return distance_pallets
```

It is important to note that there is a penalization for pallets that are in different aisles. This penalization happens because of the layout proposed in Figure 1, where an item on rows that face each other are closer than pallets that are on distant rows. The result is not in a metric system, is a numeric representation of distance where the closer ones will have a smaller total distance.

2.4.2 Selection Method

The selection method is responsible for choosing the individuals in the population that will create the next generation, having an emphasis in selecting the fitter ones with the hope that their offspring will have an even higher fitness score (Mitchell 1998).

For this study, a comparative test was performed between the Fitness Proportionate Selection, also known as Roulette Wheel, and the Tournament Selection method. In the Fitness Proportionated Selection, every individual is assigned a slice of a circular "roulette wheel", the size of the slice is proportional to the individual's fitness (Mitchell 1998). For the Tournament Selection method, *k* individuals are randomly selected from the population to participate in a tourney where the one with the best fitness score is selected (Linden 2012).

Following the assignment of the roulette wheel, a random number is generated, and the algorithm iterates over the roulette looking for the winning portion of the wheel. Based on experiments demonstrated by Linder (2012), the Tournament Selection method was configured to have 3 participants, who were randomly selected from the population and add to a list of tournament participants. The winner was the one with the best fitness.

2.4.2 Crossover operator

After selecting two individuals, it is necessary to generate an offspring based on them. Thus, it was needed to access the pallet list from the individuals and pick *n* pallets from each individual. This was done by generating a random number between 0 and 99 (total pallets that an individual has in this experiment) and then using this number as a cut point in the list. Figure 3 presents a small example where the random number is equal to 4, which means that the first five pallets of the offspring come from individual 1 while the others are from individual 2.

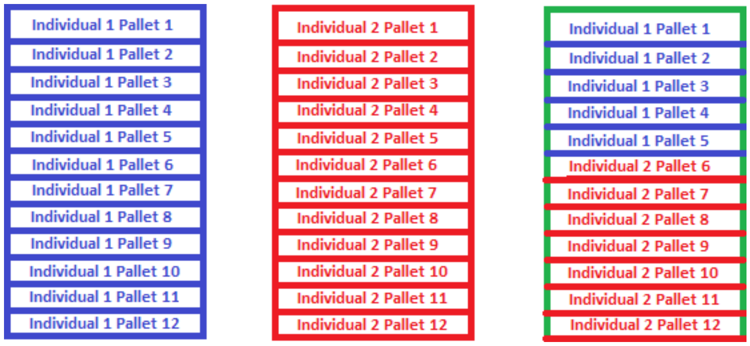

Figure 3. Example of crossover

This method allows for a wide range of possibilities within a population and even if the same individuals are selected, it is unlikely that the offspring will be the same, and if it happens, the algorithm will ask for a new selection to be made.

2.4.2 Mutation operator

Mutation operator changes a solution by disturbing them using random changes (Kramer 2017). It is important to note that finding a balance is a difficult task when working with the mutation operator, because if the GA has a high mutation rate it can lose efficiency by changing individuals who are fit to the solution, and if the GA has a low mutation rate, it may converge too early to a local solution (Linden 2012).
Based on observations during the development process of the GA, it was decided that for every crossover made there is a 1% possibility of the offspring will suffer a mutation. Moreover, mutation can only change ¼ of the new individual, to guarantee that it will not completely lose the parents genes.
Given the fitness function and how the data structures work, it is not needed to add a new pallet to have a mutation. For a mutation to happen is only needed to rearrange the pallets positions in the offspring. Thus, a random number between 0 and ¼ of the size of the pallet list is generated. Afterward, the result of this random number is used to generate another random number that will be the positions that need to be changed. Figure 4 demonstrates a situation where the random number for a change is 2, and the positions selected for the mutation are 0-9 and 3-11.

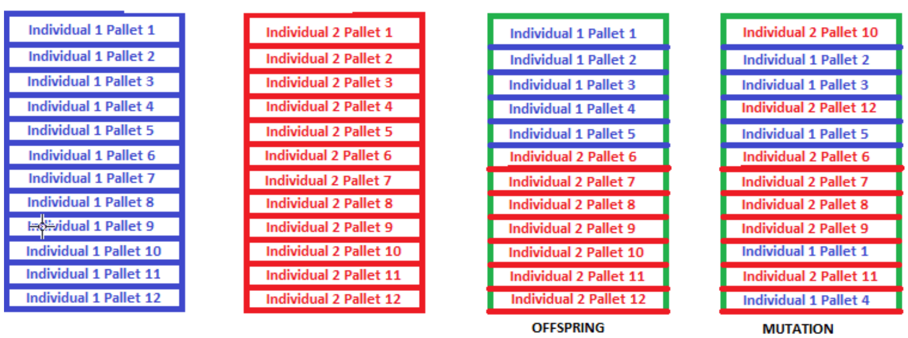

Figure 4. Example of mutation

2. Results

The study used two different approaches to identify the best solution for the problem. One of the approaches used the Fitness Proportionate Selection method and the other used the Tournament Selection method. Both methods used the same GA configuration. It was used 100 generations, 100 individuals as population size, 1% of mutation rate and 10 rows of pallets each with 10 pallets, making a total of 100 pallets. It is important to note that the code can be adapted for any size of generations, individuals, mutations or numbers of pallets in stock.
Scientific observations (Mitchel 1998, Linden 2012) shows that in some datasets, the Tournament Selection can have a better performance than the Fitness Proportionate Selection. In this work, it was possible to perceive some difference between the results of the two methods, but the overall best individual was very close in fitness score.

Figure 5 presents the average fitness per generation for the Fitness Proportionate Selection method where the first generation had a very high fitness score and as expected, the average went down generation by generation. There are up and downs that can be attributed to bad mutations happening in any given generation.

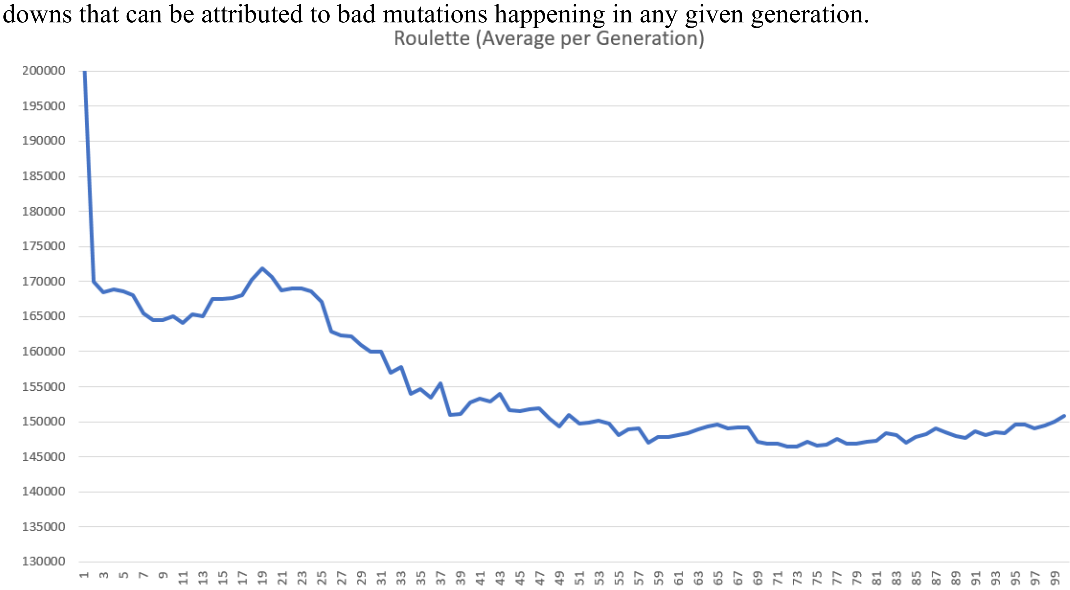

Figure 5. Fitness Proportionate Selection (Average per Generation)

Figure 6 shows the average per generation for the Tournament Selection method where the evolution of generation happens faster and with better fitness than the Roulette Selection.

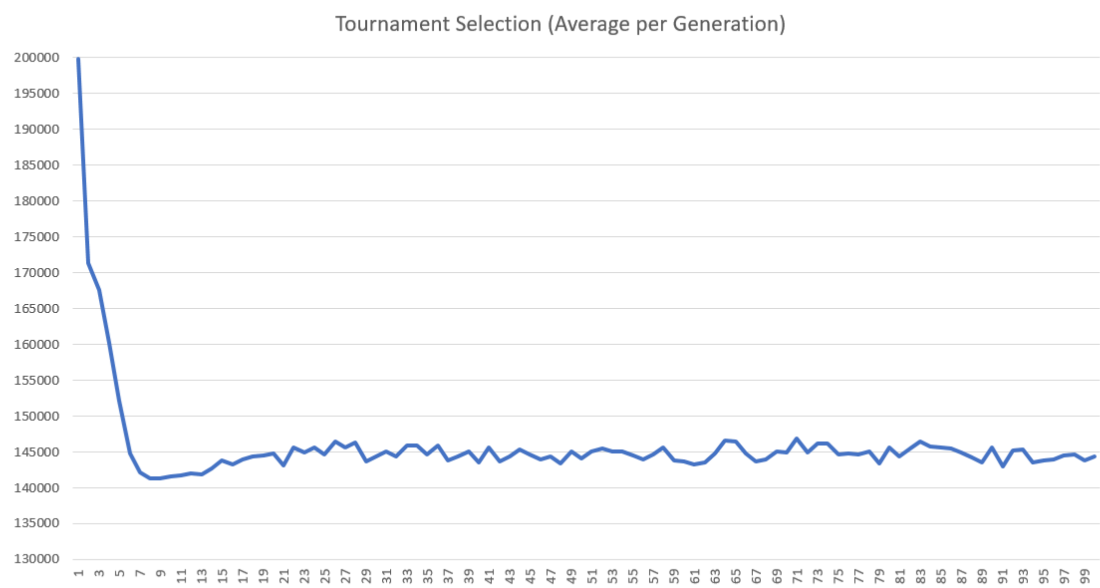

Figure 6. Tournament Selection (Average per Generation)

The best individual per generation using the Fitness Proportionate method is presented in Figure 7. There is an initial variation, but the fitness score stabilizes after the 63rd generation. It ends in a higher score due to an above average mutation in the last generation and some non-optimal selections on the roulette wheel.

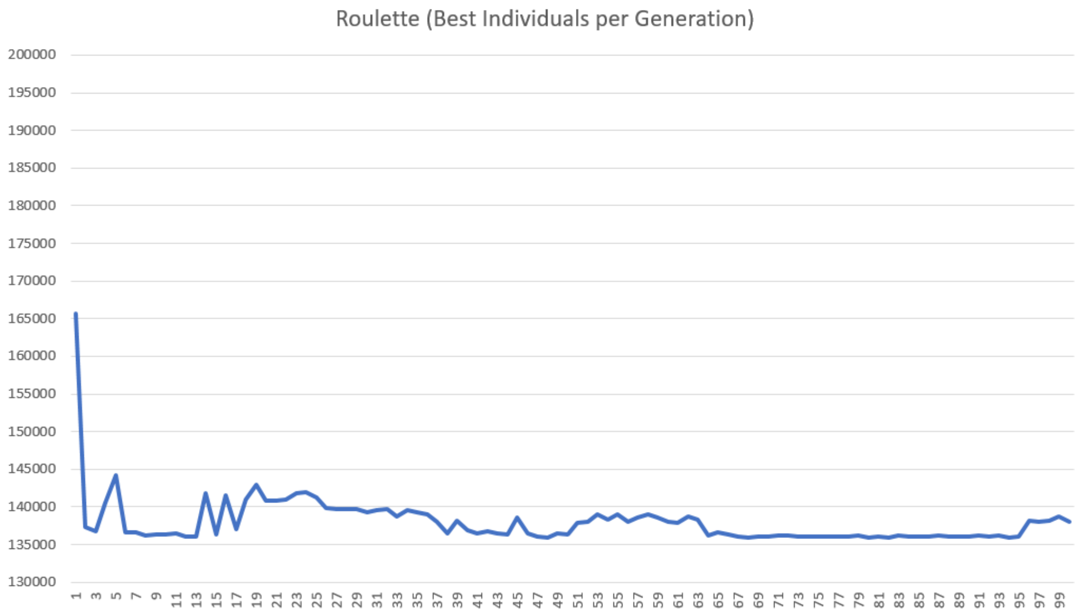

Figure 7. Fitness Proportionate (Best individual per Generation)

The results of the Tournament Selection for Best Individuals per Generation are shown in Figure 8. The optimal solution is found in an early generation than the Fitness Proportionate method and it has less variation of best individuals. It is important to note that both methods have the best individual with close fitness score, 134885 for the Tournament Selection and 135891 for the Roulette Wheel selection.

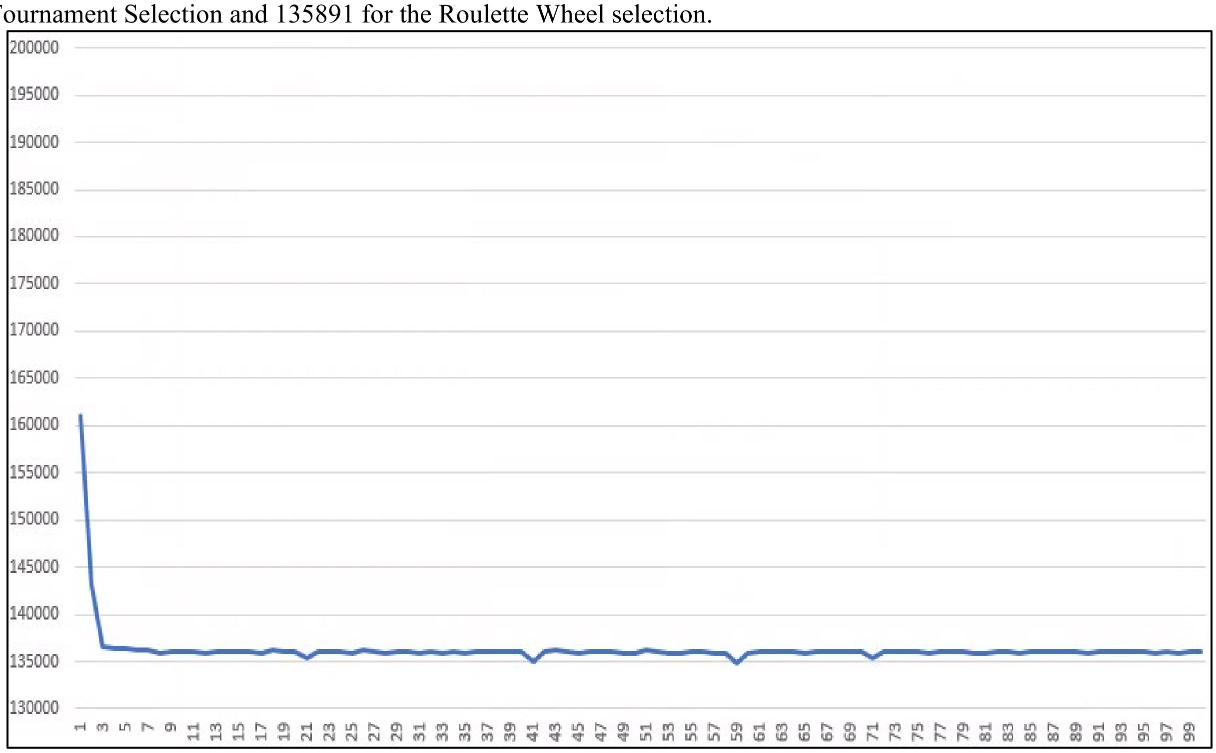

Figure 8. Tournament Selection (Best Individuals per Generation)

The best individual for the Fitness Proportionate method is achieved at the 49th generation while the best for the Tournament Selection is obtained at the 59th generation. The result is two individuals with different pallets organization but with close fitness score.

Figure 9 shows the warehouse layout configuration following the best individual created by the Tournament Selection method at the 59th generation.

Figure 9. Warehouse layout for the Best Individual (Tournament Selection)

There is an agglomeration of the top 6 colors on aisle 0 and aisle 1, where is possible to observe 5 data mining rules (#1, #2, #3, #4 and #12 from Table 2) on aisle 0 and one rule (#9) on aisle 1. Together, aisle 0 and 1, presents 5 rules (#7, #10, #17, #18, #28) and concentrate 60% of the company's sold colors.
Aisle 2 presents two new rules (#5 and #8) and combined with aisle 1 the rules (#11 and #27) are observed. Aisle 3 and 4 does not present any new associations and it concentrates colors who sell less on the top 30.

3. Conclusion

This paper presented an implementation of GA to support changes in a warehouse layout and to provide a better order picking time using a real case scenario, with data extracted directly from the company's database. Besides, it was possible to test and compare two selection methods used in the study.
Both selection methods presented feasible solutions and it was possible to identify some differences between them. Considering the GA designed in this paper, the Tournament Selection method showed more stability between generations and was able to reach an optimal region earlier. The Fitness Proportionate Selection method had more variations through generations but was able to reach the best individual earlier than the Tournament Selection.

Besides configuring the layout by top selling colors, the use of association rules enabled the fitness function of the GA to put the items that are sold together in a minimal distance, which leads to a minimization of the order picking time.

The results demonstrate that the GA was able to minimize the distance between the best-selling items, having 15 out of 30 association rules within one aisle of distance while concentrating 60% of the sales on aisle 0 and 1. By optimizing the warehouse layout using association rules and best-selling items, it is expected to reduce the order picking time if compared with the current method used.

This study is being analyzed by the company's management and it is expected to be implemented until the end of 2021. The proposed GA will be used to evaluate warehouse layout changes following the sales data throughout the years.

References

Ardjmand, E., Barjgiran, O. S., Youseef, E., Using list-based simulated annealing and genetic algorithm for order batching and picker routing in a put wall-based picking system, *Applied Soft Computing,* vol. 75, pp. 106-119, 2019.

Beroule, B., Grunder, O., Bakarat, O., Aujoulat, O., Order Picking Problem in a Warehouse Hospital Pharmacy, *International Federation of Automatic Control,* vol. 50, no. 1, pp. 5017-5022, 2017.

Boysen, N., Koster, R., Weidinger, F., Warehousing in the e-commerce era: a survey, *Euroupean Journal of Operational Research,* vol. 277, no. 2, pp. 396-411, 2019.

Chen, T. -L., Cheng, C. -Y., Chen, Y. -T., Chan, L. -K., An efficient hybrid algorithm for integrated order batching, sequencing and routing problem, *International Journal of Production Economics,* vol. 159, pp. 158-167, 2015.

Giannikas, V., Lu, W., Robertson, B., McFarlane, D., An interventionist strategy for warehouse order picking: evidence from two case studies, *International Journal of Production Economics,* vol. 189, pp. 63-76, 2017.

Gibson, B. J., Defee, C. C., Ishfaq, R., The state of the retail supply chain: essential findings of the sixth annual study, Available: https://www.researchgate.net/publication/304630415_2016_Annual_State_of_Retail_Supply_Chain_Report, June, 2016

Gils, T. V., Ramaekers, K., Caris, A., Koster, R. B. M., Designing efficient order picking systems by combining planning problems: state-of-the-art classification and review, *European Journal of Operational Research,* vol. 267, no. 1, pp. 1-15, 2018.

Henn, S., Algorithms for on-line order batching in an order picking warehouse, *Computers & Operations Research,* vol. 39, no. 11, pp. 2549-2563, 2012.

Koster, R., Le-Duc, T., Roodbergen, K. J., Design and control of warehouse order picking: a literature review, *European Journal of Operational Research,* vol. 182, no. 2, pp. 481-501, 2007.

Kramer, O., Genetic Algorithms Essentials, Vol 679 of *Studies in Computational Intelligence,* Springer International Publishing, Germany, 2017.

Linden, R., Algoritmos Genéticos [Genetic Algorithms], Ciência Moderna, Brazil, 2012.

Liu, S. -A., Sun, J., Wang, Q., Optimization of storage performance for generic tiered warehouse by Genetic Algorithm, *Proceedings of 26th Chinese Control and Decision Conference (2014 CCDC),* Changsha, China, May 31 – 2 June, 2014.

Mitchell, M., An Introduction to Genetic Algorithms, MIT Press, London, 1998.

Orti, M. V., Martins, R. B., Aplicação de técnicas de mineração de dados para identificação de fatores que influenciam o desempenho comercial e industrial em uma indústria de beneficiamento de malha, *Proceedings of IX Computer on the Beach,* Florianópolis, Brazil, March 22- 24, 2018.

Pan, C., Yu, S., Du, X., Optimization of warehouse layout based on genetic algorithm and simulation technique, *Proceedings of 2018 Chinese Control and Decision Conference (2018 CCDC),* Shenyang, China, June 9-11, 2018

Yener, F., Yazgan, H. R., Optimal Warehouse design: literature review and case study application, *Computers & Industrial Engineering,* vol. 129, pp. 1-13, 2019.

Biography

Raimundo C. Ghizoni Teive is a professor at UNIVALI – Brazil and he is also researcher for SEEnergia Company. Raimundo Teive has worked in the application of Artificial Intelligence techniques into engineering problems, with special interest in Electrical Power Systems, for more than 20 years. He has been involved in several research projects related to Knowledge-Based Systems, Genetic Algorithm, Neural Networks, Bayesian Networks and Data Mining. He is an IEEE member.

Maxsuel Vieira Orti, is a student in the Master's Program in Applied Computing at UNIVALI – Brazil and an IT manager for Incotextil Industrial Ltda and Refisa Indústria e Comércio Ltda. Maxsuel Orti has worked with the application of Artificial Intelligence techniques applied to operational research in the Textile field. He has been involved in projects related to Knowledge-Based Systems, Data Mining and Metaheuristics.