Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.

A SIMULATION TOOL TO PROVIDE ALTERNATIVE PRODUCTS IN OUT-OF-STOCK SITUATIONS FOR B2B COMPANIES

Sebastian Pinto Guzman

Departamento de Ingeniería Informática,
Universidad de Santiago,
Av. Las Sophoras 175, Estación Central
Santiago, Chile

Veronica Gil-Costa

Universidad Nacional de San Luis
CCT CONICET San Luis
Ejército de Los Andes 950
San Luis, D5700, ARGENTINA

Mauricio Marin

Departamento de Ingeniería Informática,
Universidad de Santiago, CeBiB
Av. Las Sophoras 175, Estación Central
Santiago, Chile

ABSTRACT

With the rise of online sales and services dedicated to advertising aimed at the profiles of online users, companies selling general products need to adapt their business rules to improve sales strategies. In this paper, we present a simulation tool to support business decision-making, and improve the efficiency of an e-commerce system. It allows testing different sales strategies within the context of Business-to-Business models which focuses on the purchase and sale of non-strategic products. The idea is to provide the client's company with different alternative products when the desired product is out-of-stock. We propose and implement in our simulation tool, three recommendation algorithms, based on sales parameters such as the price and the popularity of products and the similarity between the recommended product and the one out-of-stock. The proposed algorithms are tested with real datasets, which allows evaluating the performance of the recommendation algorithms and their effectiveness.

1 INTRODUCTION

The Business-to-Business (B2B) model focuses mainly on the purchase and sale of non-strategic products from small and medium enterprises (SMEs) or large companies. Companies based on this model generally do wholesale. Moreover, the prices for each product are determined according to the size and quantity of purchases made by the client's company, which are negotiated directly with the B2B-based company. To meet the demand for products, these B2B-based companies must be able to determine the stock for each product. In case of lack of stock, they should be able to recommend similar products to satisfy the client's demand, otherwise they can lose the clients. In this scenario, simulation is a powerful tool to analyze and evaluate the effectiveness and the efficiency of recommendation algorithms.

In this work, we present a simulation tool to analyze sale strategies for a B2B company. Our simulator design includes modules implemented as templates that connect to each other. It includes the interaction between clients, operators and the products during the sale process. Our simulation tool supports different recommendation algorithms for situations where the desired product is out-of-stock. In particular, we

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

propose three algorithms which take into account previous sells performed by different clients and also external factors such as the price. We evaluate the effectiveness of the algorithms to recommend similar products as well as their performance in terms of CPU time and memory consumption.

In particular, as case study we model the sale process for a company which is based on the B2B business model. We name this company DMB2B. The DMB2B company offers products belonging to a hierarchy of categories such as hygiene, technology, personal care, office and bookstore, etc. Each of these categories has a set of subcategories used to define a certain product. Categories are represented as a tree, where the top nodes represent the most general categories, the intermediate nodes represent more specific subcategories and the leaves represent the products. This company has a large number of clients grouped by segments depending on the size of the client's company. This information is relevant to DMB2B to negotiate product prices, as it is possible that a company of a larger segment buys a larger number of products.

Clients are also classified according to the amount of products and how often they buy those products. This information, as well as the client segment, are useful for the price negotiation process and for making product recommendations. DMB2B records each purchase made by each one of the clients. Each sale record contains information such as the product identifier (ID), the quantity and the price.

We simulate the purchase process of the DMB2B company to evaluate how the proposed recommendation algorithms affect the stock of different products. Simulation in driven by historical data of sales which is used to determine the simulation time advance for each operation as well as the tendency of the clients to accept an alternative product. The simulation results can be used by the DMB2B company to adapt its business rules and increase product sales.

The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 details our simulation model. Section 4 present the recommendation algorithms for B2B companies. Section 5 presents the simulation setup and the results. Finally, Section 6 brings the conclusions and future works.

2 RELATED WORK

The authors in (Vuksic et al. 2001) empathize that simulation is a useful tool to support the re-engineering of a company's business processes, since it can allow modeling the current operation of the company and later study at a quantitative level the effects of including new (or modifying existing) business processes. They present the study of a B2B e-commerce system as an example, which is modeled and simulated through the "IGrafx" tool. Different variables and business processes are modified to show the ability of simulation tools to predict and study changes in the system.

The work in (Tang et al. 2004) presents a supply chain simulation model aimed at optimizing, analyzing and quantifying the supply chain of different B2B-based e-commerce companies. The simulation is intended to analyze different situations and provide more information to improve different stages and processes involved in the supply chain process.

The work in (Chunyuan 2020) presents a supply chain simulation model aimed at optimizing, analyzing and quantifying the supply chain of different e-commerce companies based on the B2B model. The author in (Chunyuan 2020) aim to improve the information available for decision-making during the improvement of different stages and processes of a company's supply chain.

In (Chen et al. 2006), the authors present a process-oriented simulation using the "processmodel" tool for a B2B e-commerce system in the field of industrial waste management. The authors show the usefulness of simulation to analyze, according to different variables (profits, costs, times, etc.), the effectiveness of different business processes integrated in e-commerce systems. Authors concluded that simulation is a useful tool for decision makers or people who want to invest in this type of system.

In (Šperka and Slaninová 2012) an e-commerce system architecture based on multi agents is proposed. Where each agent represents an autonomous element that performs a specific action within the system, which can communicate with other agents to fulfill its objectives. Additionally, the authors present a simulation layer which, through agent simulation, is aimed at improving the user experience, using historical data to predict customer behavior, as well as support decision-making, such as prices, advertisements or other types of strategies.

In (Markovic et al. 2016) the authors present a simulation model of a B2B e-commerce system, based on an agent model. It is implemented with the ”NetLogo” software, with the aim of studying and monitoring the behavior of the different actors in the system (consumers, sellers, companies), allowing the evaluation of different business decisions, such as the generation of offers or the use of advertising, in order to increase sales. Finally, authors conclude that this simulation model is very useful to test different business strategies, improve market segmentation and thus increase profits.

In (Menascé et al. 2000) the authors discuss the importance of using new policies to manage server resources in e-commerce systems, which must be oriented towards the business objectives (higher revenues per second) of the company, taking into account user behavior. They propose a scheme in which more server resources are assigned to users with profiles oriented to make more purchases. They implement a discrete event based simulator in *C*, of an e-commerce system model. The simulator focuses on user navigation using a graph that defines the behavior of the user within an e-commerce system. The simulator creates different artificial flows. The results show that by prioritizing the use of resources according to user profiles, greater profits are obtained.

In (Mitrevski and Hristoski 2016) the authors present a simulation model for the “Insight Maker” web platform, aimed at studying metrics focused on the business objectives of an e-commerce system (such as the number of sales per second). Specifically, it is intended that the simulation helps to predict the number of future sales. The behavior of the users is used as a basis of the work, which is studied through a graph that represents their different actions in the system. It allows to classify the users giving as result an estimation of the purchases he/she can do.

In (Gupta et al. 2021) models and simulates a sub-process of a B2B company named order-to-cash. The simulator is developed with ARENA. The authors model different scenarios based on manpower combination of technical and non-technical skills in both existing and improved process to find out the near optimal scenario suitable for implementation by the organization. The work in (Mohsen et al. 2021) presents a simulation for a production line in a cabinet manufacturing facility carried out with the aim of better understanding and improving the production processes particularly associated with mass customization using Simphony.NET.

2.1 Recommendation algorithms for B2B

Recommendation algorithms get the most accurate prediction of users’ preferences. There are different types of recommendation systems(Adomavicius and Tuzhilin 2005) including Content-based recommendations, Collaborative recommendations and Hybrid approaches.

The work in (Dadouchi and Agard 2018) considers actual stock levels in the recommendation process in order to shift demand toward specific products for a specific user. It considers supply chain constraints and strategies. The work in (Guo et al. 2018) presents a recommendation algorithm based on the price by product categories. It proposes to use a multi-category purchase interval to model the drift of a user’s interest for different categories based on sequential pattern mining. The work in (Varsha et al. 2021) investigates the use of artificial intelligence techniques and language processing capability which helps chatbots to interact with customers and offer those customized recommendations. The authors in (Tarnowska and Ras 2021) present a knowledge-based recommender system from unstructured (text) data. The system uses an opinion mining algorithm which extracts aspect-based on sentiment score per text item, and transforms text into a structured form.

The work in (Chen et al. 2008) presents two profitability-based recommendation systems called CPPRS (Convenience plus Profitability Perspective Recommender System) and HPRS (Hybrid Perspective Recommender System). The work in (Su and Khoshgoftaar 2009) presents a classification of the recommendation methods based on collaborative filtering techniques. The work presented in (Bauer and Nanopoulos 2014) evaluates the advantages and disadvantages of recommendation systems. They propose to extend matrix factorization techniques by allowing used for client's recommendation, for different distributions.

3 SIMULATION MODEL

Our simulator tool is implemented on top of the LibCppSim library (Marzolla 2004). The code is available at https://github.com/seba322/simulador-b2b-dim. The LibCppSim library manages the creation/removal of co-routines as well as the future event list. The library ensures that the simulation kernel grants execution control to co-routines in a mode of one co-routine at a time. Co-routines are activated following the sequential occurrence of events in chronological order. Co-routines represent processes that can be blocked and unblocked at will during simulation by using the operations passivate(), hold() and activate(). When a hold(Δ_t) operation is executed, the co-routine is paused for a given amount of Δ_t units of simulation time representing the dominant cost of a task. Once the simulation time Δ_t has expired, the co-routine is activated by the simulation kernel. The costs of the simulated operations are obtained from benchmark programs and from the logs of sale records provided by DMB2B. Additionally, a co-routine executes a passivate() operation to stop itself, indicating it has paused its work. Finally, a co-routine in passivate state can be activated by another co-routine using the activate() operation.

For each client we store its ID and the top-10 most purchased products. This list of products is updated every three months. For the products we keep information about the stock and its category. Products are stored in a tree-based data structure. The upper nodes of the tree represent general categories. In the intermediate nodes we represent more specific categories and at the leaves we keep the product IDs. We also model the DMB2B employees in charge of processing the client's requests.

We use the log of sale records provided by DMB2B to set the initial stock of products. This log is used to generate synthetic sale records to test different scenarios. E.g. to create a scenario where there are peaks of products demands in different seasons of the year, or to simulate the opening of a new DMB2B sales branch in another region.

When some products of the same category run out-of-stock we execute an algorithm to compute the number of new products required to satisfy the client's demand. This is performed by the replacement module of the simulator which takes into account the initial stock and an additional percentage (R) which depends on the number of times the stock had to be replaced previously (see Eq. 1).

$$new\_stock = initial\_stock + (initial\_stock \times (R \times 5\%)) \tag{1}$$

Additionally, our simulator implements different recommendation algorithms when a product runs out-of-stock. These algorithms are detailed in the next section and are used to suggest alternative products to the clients. Figure 1 describes the components of our simulator. There is a sale generator module which reads data from a log of sales file. The file contains information about the product IDs, the prices, and the client ID. Then, the generator assigns these sales to the clients. If the client does not exist, the simulator creates it using an existing profile from the client record file. The client sends the operation "Buy" to a DMB2B operator. The operator checks if the product has enough stock to satisfy customer demand. If there is enough stock, the operator negotiates the price with the client and closes the trade. In case there is no stock, the operator recommends alternative products. The client accepts the alternative products based on its historical behavior obtained from its profile in the client records file. This file is used to build an empirical distribution for the clients about their decision on accepting/refusing alternative products. Finally, the restock module is responsible for replenishing the stock of products when a product is out-of-stock using Eq. 1.

The simulator output metrics include the records of all the sales performed during the simulation. We also show the final stock of each product and the total purchases of each client. There is an additional output file that collects the product recommendations performed by the proposed algorithms and whether they were accepted by the clients.

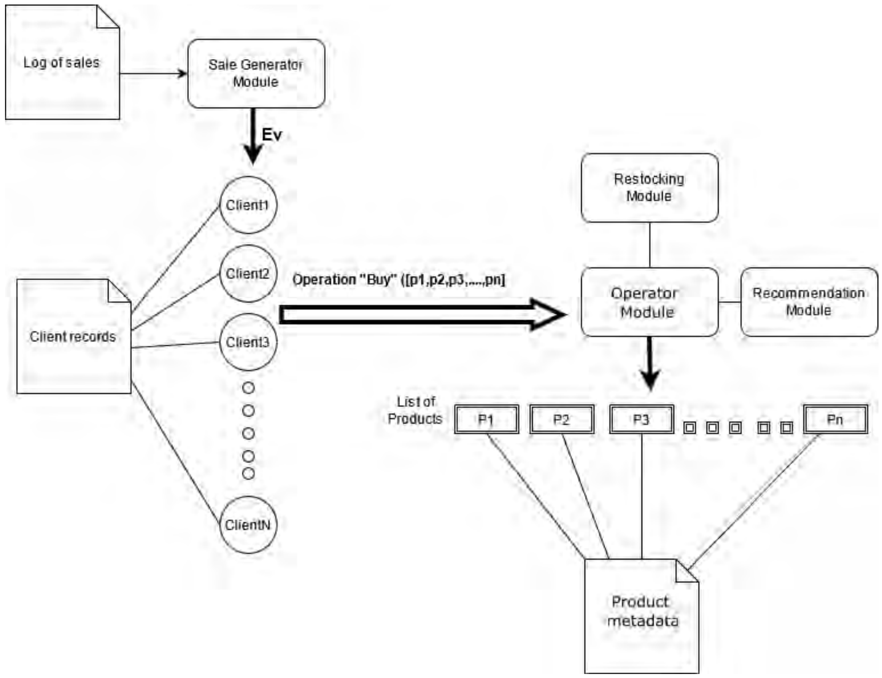


Figure 1: General scheme of our simulation tool. It is composed of a generator of tasks, the clients, the operator of the company, the restocking and recommendation modules and the products.

4 PRODUCT RECOMMENDATION ALGORITHMS

In this work we present and evaluate our proposed simulation tool with three recommendation algorithms which are executed inside the Recommendation module of the simulator. These algorithms aim to suggest alternative products to clients, when the product to be purchased does not have available stock. These methods include different parameters like the price, the popularity of the products in different client segments and the popularity obtained in different client categories. Client categories include clients who sale e.g. cars, clients offering e-commerce or finance services, among others. The client segments are defined according to the size -number of employees- of the company.

4.1 Algorithm based on the Popularity and the Price

The algorithm based on the Popularity and the Price (PP) (see Alg. 1) computes a score to each product taking into account: the popularity (ρ), the inverse of the highest price of the product ($\alpha = 1/max\_Price$) and the category similarity (σ). The inputs of the algorithm are the product identifier and the parameter's weights used to determine the relevance of ρ, σ, and α, which are used in the calculation of the total score. First the algorithm retrieves products which are similar to the product out-of-stock, using the getSimilarProducts(P) function. That it, this function retrieves the products belonging to the same category or to a category close to the product out-of-stock in the category tree. This function can be used to limit the number of levels of the category tree from where we select candidate products. Then for each product p the algorithm computes ρ, α, σ and finally the score.

The popularity ρ for each product is a value from 0 to 1 (getPopularity() function in Alg. 1). It is computed according to the number of sales counted in the DMB2B log of sale records. A product with the highest number of sales has a $\rho = 1$ and the product with the lowest number of sales has a $\rho = 0$. Then, the algorithm computes the inverse parameter $\alpha = 1/max\_Price$ of a product's price as follows. It is obtained by using the highest number for which a sale has been registered for that product. This number is obtained for each product from the sales recorded by DMB2B. The idea of using this parameter is that as the price is lower, a higher value is obtained to compute the total score of the product. Thus, products with lower prices are recommended.

Then, for each similar product the algorithm computes the σ, which is actually a value used to determine how similar is the recommended product to the product out-of-stock. As we described before, we define that two products are similar if they belong to the same category or subcategory of the category tree. That is, if they have the same parent node in the tree, then $\sigma = 1$. As we go up in the category tree to find the parent node of both products, the similarity is lower.

The algorithm returns a list of the top-10 alternative products that obtain a score value greater than a threshold value $th = 0.5$ sorted by their score.

Algorithm 1: Recomendation based on the Popularity and the Price

Input: P: product without stock, $w1, W2, W3$: parameter's weights
$selectedProducts \leftarrow productslist$
`/* get list of products similar to P */`
$similarProducts \leftarrow getSimilarProducts(P)$
foreach $p \in similarProducts$ **do**
  $\rho \leftarrow p.getPopularity()$
  $\alpha \leftarrow (1/p.getMaxPrice())$
  $\sigma \leftarrow getSimilarityDist(P, p)$
  $score \leftarrow w1 \times \rho + w2 \times \alpha + w3 \times \sigma$
  if $score > 0.5$ **then**
    $selectedProducts.push(p)$
$selectedProducts \leftarrow sortByScore(selectedProducts)$
`/* get first n product of list */`
$selectedProducts \leftarrow getFirstProducts(10)$
return $selectedProducts$

4.2 Algorithms based on the Popularity by Client Segments and by Client Category

The algorithm based on the Popularity by Client Segments (PCS) executes the same Alg. 1, but the popularity of each product is computed taking into account the segment of the clients. The segment is used to determine the size of the company, for example a micro or large company.

From the DMB2B log of sale records we compute the number of sales for each product but grouped by the client segments to create a matrix of size $m \times n$, where m is the number of client segments and n is the number of products. E.g. if client c_1 belongs to segment s_1 and buys the products p_1, p_2, p_3, p_4, client c_2 belongs to segment s_1 and buys the products p_1, p_3, p_5, client c_3 belongs to segment s_2 and buys the products p_3, p_5 and client c_4 also belongs to segment s_2 and buys the products p_1, p_3, p_6, we compute the popularity of each product as shown in Table 1.

To make a product recommendation we first identify the segment of the client. Then we select a set of products that belong to the same category of the product without stock. Then, according to the segment of the client, the top-10 product with the highest score are suggested as alternative products to the client.

Our third algorithm is based on the Client Category (PCC). It is similar to the previous algorithm based on the popularity by client segment, but it uses the client category to compute the popularity of a product.

5 EXPERIMENTS

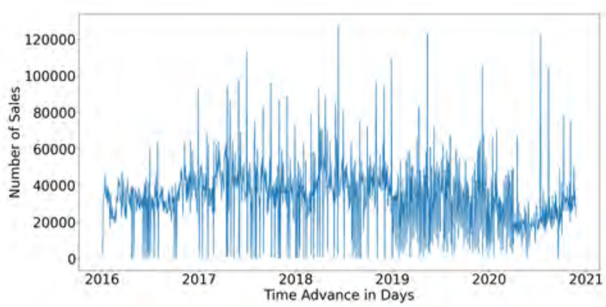
We evaluate our simulator and the proposed recommendation algorithms with a set of anonymized data provided by DMB2B. The data contains 40,000 clients with information about their corresponding segments

Table 1: Example of the Algorithm based on the Popularity by Client Segments. The columns are the products p_i and the rows represent the segments identifiers.

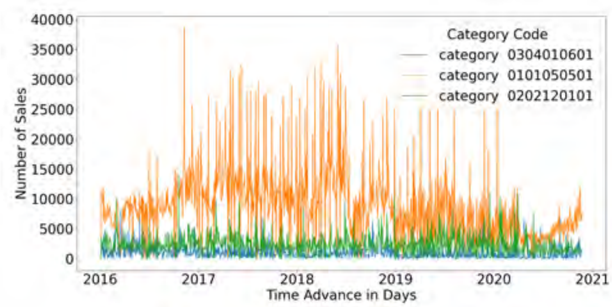
segment	p_1	p_2	p_3	p_4	p_5	p_6
s1	2	1	1	1	1	0
s2	1	0	2	0	1	1

Table 2: Execution cost of the recommendation algorithms in seconds.

Recommendation Alg.	Avg. Execution Time	Variance	Standard deviation
PP	0.00214	8.13E-06	0.002851
PCS	0.00211	8.69E-06	0,002948
PCC	0.00744	2.55E-05	0.005046



(a)



(b)

Figure 2: (a) Total number of sales for the top-100 most popular products. (b) Total number of sales for three different categories of products.

and categories. The data also includes information of 5,000 products with its corresponding category hierarchy, which is represented with a 10-digit code. This code is used to build the tree of categories. We use a log of 2,160,000 sale. This log includes the client ID, the date of the operation and the list of different products included in the sale with their respective quantities and prices.

To set the time advance of the simulation, we run benchmark programs on the recommendation algorithms. Table 2 shows the average execution time, the variance and standard deviation of the execution time in seconds reported by each recommendation algorithm. There are 7 client segments and 38 categories. The service time of the DMB2B employees was obtained from the data logs. It has a normal distribution with mean of 46 minutes and a variance of 6 minutes.

Figure 2.(a) shows the total number of sales for the top-100 most popular products reported by DMB2B between January 2016 and December 2020. Figure 2.(b) shows the total number of sales for three different product categories. This figure shows the effect of the COVID-19 in 2020, as the number of sales for these three product categories is lower than in previews years. We validate our simulator with these data obtaining a Person Correlation of 98% and a mean square error of 0.041.

5.1 Impact of the Weights on the Recommendation Algorithms

In this section we evaluate the impact of using different values of weight w on the scoring equation of the PP recommendation algorithm. We obtained similar results with the PCS and the PCC algorithms. Table 3 shows the number of recommended products $\times 1e6$. The first column shows the values of w ranging from 0.2 to 1.0. The columns named "Same Cat." show the number of recommended products reported

Table 3: Results obtained with different values of w for the parameters ρ, σ and $\alpha = 1/max\_Price$.

	ρ		σ		$1/max\_Price$	
w	Similar	Same Cat.	Similar	Same Cat.	Similar	Same Cat.
0.2	0.4	0.2	0.37	0.24	0.1	0.05
0.4	0.45	0.3	0.55	0.34	0.22	0.14
0.6	0.8	0.66	0.63	0.4	0.45	0.33
0.8	0.87	0.83	0.8	0.59	0.67	0.47
1.0	1.1	1.0	0.81	0.6	0.7	0.54

Table 4: Mean square error (ε_m), relative error (e_r) and Pearson coefficient between the results obtained without recommendation (real data) and the results with each recommendation algorithms.

	PP	PCS	PCC
ε_m	1562.33	1571.901	1502.477
e_r	0.007813	0.007858	0.007513
Pearson Coefficient	0.9972693	0.9972283	0.9974778

in the same category that the product without stock. The columns named "Similar" show the number of recommended products in a similar category going up in the category tree.

Results show that the popularity parameter ρ tends to achieve a higher number of recommended products both in the same category of the product without stock ($1.0 \times 1e6$) and in a similar category ($1.1 \times 1e6$). However, the three parameters properly combined report $1.1 \times 1e6$ products in the same category and $1.4 \times 1e6$ products in a similar category.

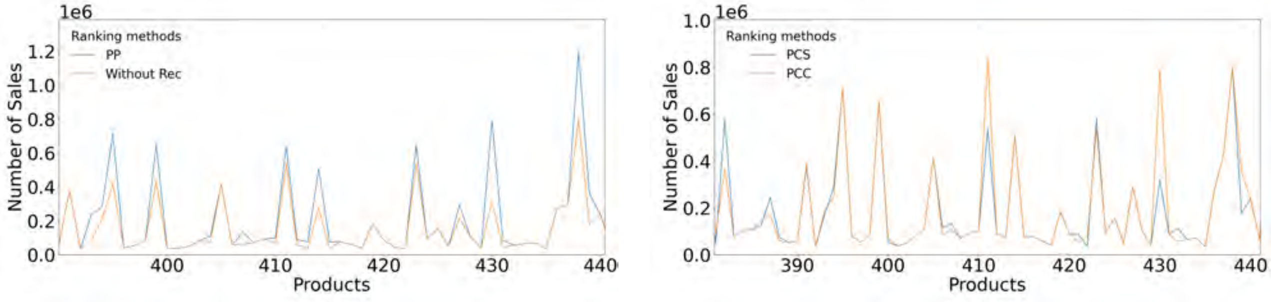
Finally, in Table 4 we show that the results presented by our simulator with the real data and the results obtained with the recommendation algorithms PP, PCS and PCC are highly correlated. In all cases the Pearson correlation is close to 1.0. We also present the mean square error of the deviation or mean square difference (ε_m). It is defined as $\varepsilon_m = \sqrt{(}\sum(x_i - \bar{x})^2/n(n-1)$. It also presents the relative error (e_r) defined as $\varepsilon_m/\bar{x}$. The relative error is the absolute error divided by the magnitude of the exact value. The ε_m shows values close to 1500 mainly because of the outliers which causes that the differences between $x_i - \bar{x}$ to be large. However, the relative error presents is very small close to 0.007.

5.2 Results

In this section, we use our simulator to analyze the recommendation algorithms with two different scenarios. These scenarios are built based on the original sale log. In the first one, scenario 1, we increase the number of sales per day up to 100 times using a uniform distribution. This allows simulating a situation in which the company expands its business areas to other regions or countries. For each sale record, we set the number of products between 1 and 100, due to we observed in the original sale record that sales tend to include 100 products at most. Finally, we set the amount of each product sold between 1 and 10.

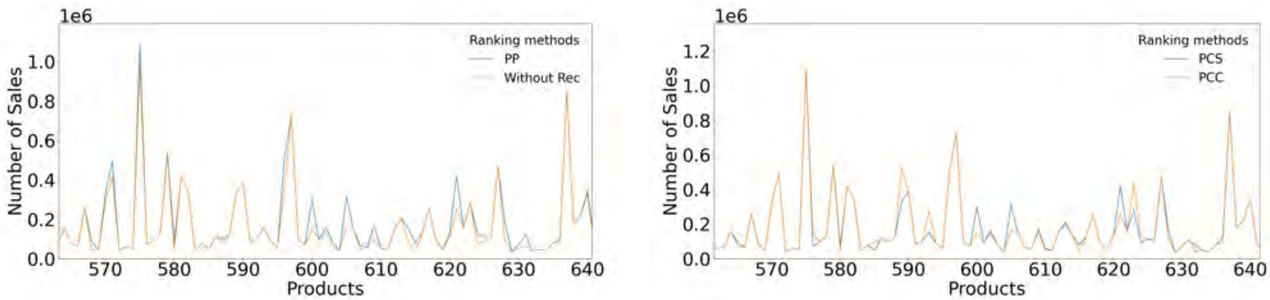
In the second scenario, scenario 2, we emulate an adverse situation, such as the one that occurs due to the COVID-19. In this case, clients purchase change dramatically. For this, we use the inverse probability of the popularity of the products to generate new sales. Thus, we give more probability to products that normally are not frequently bought.

Figure 3. (a) and Figure 3.(b) show the total number of sales reported in scenario 1 with the tree recommendation algorithms and without any recommendation algorithm. The x-axis shows the product ID and the y-axis shows the total number of sales for each product. Notice that the scale of the y-axis is 1e6. To better illustrate the results, we only show results for products IDs ranging from 380 to 440. Results show that the demand of a few products is drastically higher than the most. On the other hand, Figure



(a) (b)

Figure 3: Zoom on the total number of sales in scenario 1 for products ID ranging from 380 to 440. (a) With the PP algorithm and without recommendation. (b) the PCS and PCC algorithms.



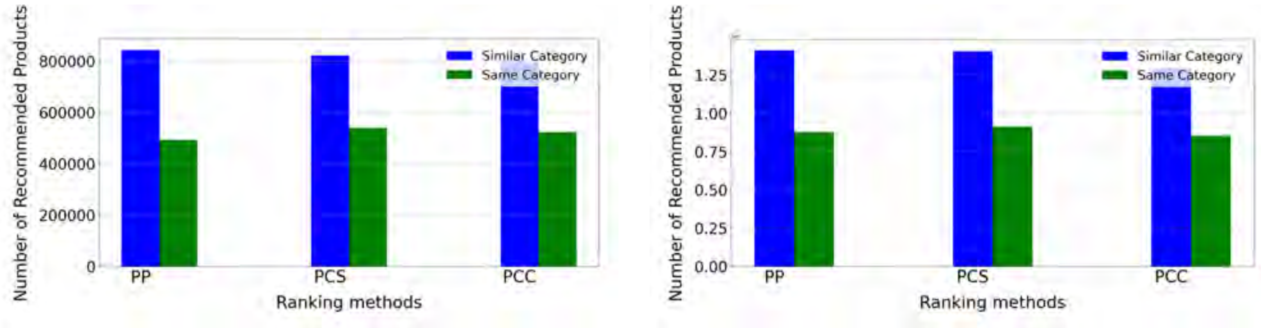
(a) (b)

Figure 4: Zoom on the total number of sales in scenario 2 for products ID ranging from 560 to 640. (a) With the PP algorithm and without recommendation. (b) The PCS and PCC algorithms.

3.(a) shows results for the PP algorithm and without recommendation. 3.(b) shows results for the PCS and the PCC algorithms. In this scenario, the results reported by all three recommendation algorithms are very similar. In average, the PP algorithm recommends 235,760.775 products, meanwhile the PCS recommends in average 235,771.053 products and the PPC recommends 236,781.891 products. The number of sales reported without any recommendation algorithm is 199,917.9. Therefore, the recommendation algorithms allow to increase by 15% the sales of the company. The average values are computed among all products IDs, e.g. the values for the PP algorithm are close to 0.25e6 in Figure 3.(b).

Figure 4.(a) and and Figure 4.(b) show the total number of sales reported in scenario 2 with the tree recommendation algorithms and without any recommendation algorithm. The *x*-axis shows IDs ranging from 560 to 640. In this case, we also have peaks of sales for a few products. Meanwhile most of the products reports sales below 2,000,000. We also plot the number of sales reported without any recommendation algorithm (Without Rec). In this case, the average number of product recommendation algorithms reported by the PP algorithm is 218,064.769, meanwhile the PCS recommends 218,571.753 products in average and the PCC recommends 218,365.851 products. The number of sales reported without any recommendation algorithm is 199,834.5. Therefore, in this case, the recommendation algorithms allow to increase by 8% the sales of the company.

Although both scenarios are different, we show that some products have a majority of sales. This is mainly due to the fact that in the records of DMB2B, those products present a large number of corporate sales and these products are of daily use and of vital importance for different companies. Therefore, their popularity tends to be extremely high.



(a) (b)

Figure 5: Number of recommendations inside the same product category or in a similar category. Results obtained for (a) scenario 1 and (b) scenario 2. In (b) the scale of the y-axis is 1e6.

Figure 5.(a) and Figure 5.(b) shows the number of recommendations performed by each proposed algorithm in scenario 1 and 2 respectively. Product recommendations can belong to the same category of the product without stock, or they can belong to a similar category. In both scenarios, the results show that the PP algorithm reports a larger number of products belonging to a similar category to the product without stock. Moreover, most of the alternative product recommendations belong to a different category than the one of the product without stock.

Regarding the recommendations of products belonging to the same category, the PCS and PCC algorithms report similar results in scenario 1. The PCS increases by 2% the number of products in the same category reported by the PCC and by 4% the number of products in the same category reported by the PP. But in scenario 2, the PCS algorithm reports a higher number of recommendations in the same category. In this case, the PCS increases by 6% the number of products in the same category reported by the PCC and by 8% the number of products reported in the same category by the PP. In other words, the PCS algorithm tends to present alternative products closed related to the one out-of-stock.

5.3 Performance Evaluation

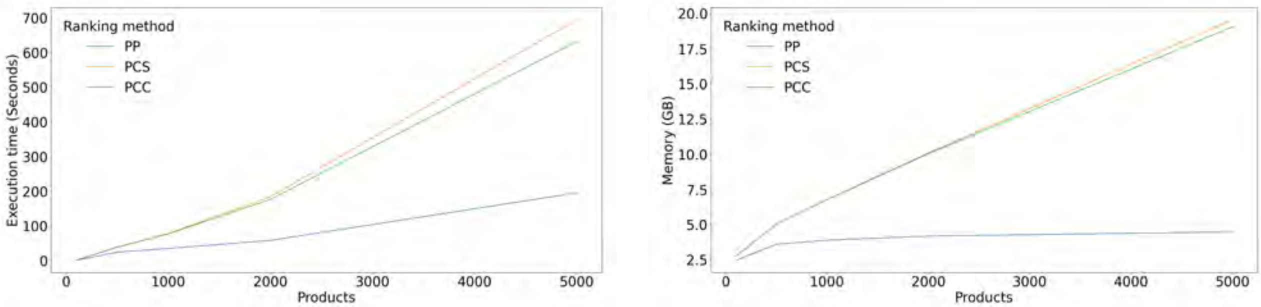
In this section we present a performance evaluation of the proposed recommendation algorithms. In particular, we evaluate the CPU time and the RAM memory consumption reported for different number of products in the database: 100, 500, 1000, 2000 and 5000.

Figure 6.(a) shows the execution time in seconds. The x-axis shows the number of products evaluated in the experiment. The PP algorithm reports lower running time than the other two algorithms. The difference between the PP and the other algorithms tends to increase up to 75% as we increase the number of products. This is because the PCS and the PCC process a matrix of size $n \times m$ to compute the product popularity ρ. The computation cost of this operation tends to increase as we increase the number of products in the database.

Similar results are presented in Figure 6.(b) for memory consumption. The y-axis shows the virtual memory consumption reported by each recommendation algorithm going up to 20 GB. Notice that the PP algorithm reports almost constant memory consumption. Meanwhile the results obtained with the remaining algorithms, PCS and PCC, the memory consumption increases almost lineally with the number of products.

6 CONCLUSION

In this paper we presented a simulation tool for a B2B-based company suitable for the recommendation of alternative products that are out-of-stock. Our simulator takes into account multiple operation profiles, recommendation algorithms and scenarios. The simulator also includes a restocking module used to estimate the number of products to satisfy futures sales. The simulator is aimed to help management teams with



(a) (b)

Figure 6: (a) Execution time in seconds reported by the PP, PCS and PCC recommendation algorithm for different number of products in the database. (b) Memory consumption reported by the PP, PCS and PCC recommendation algorithm for different number of products in the database

the assessment and forecasting business-oriented performance measures by giving information regarding which products should be prioritized when restocking the stock.

The simulator has a modular design allowing to easily add business rules in the form of new components or templates. In other words, the simulator can integrate new modules or entities which represent business processes and sale strategies involved in the flow of the stock replacement and sales. It can be used to evaluate the impact of the new strategies or processes without risking the costs of the company.

We presented three recommendation algorithms for products without stock. The algorithms take into account the price, the popularity and the similarity between the recommended product and the one without stock. In particular, we present different variants on how to compute the popularity of products within different segments or categories of clients. The simulation tool allows evaluating the effectiveness of the recommendation algorithms as well as their performance. We simulated two scenarios in which the company expands its business areas to other regions or countries. Results showed that the recommendation algorithm based on clients features increases the number of sales between 8% and 15% depending on the characteristics of the simulated scenario.

As future work we plan to extend our simulator to include more complex business rules such as the negotiation between DMB2B and the product providers and include the distribution of products to the clients. We also plan to evaluate different machine learning algorithms to predict the restocking of products.

REFERENCES

Adomavicius, G., and A. Tuzhilin. 2005. “Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions”. *IEEE Transactions on Knowledge and Data Engineering* 17 (6): 734–749.

Bauer, J., and A. Nanopoulos. 2014. “Recommender systems based on quantitative implicit customer feedback”. *Decision Support Systems* 68:77–88.

Chen, L.-S., F.-H. Hsu, M.-C. Chen, and Y.-C. Hsu. 2008. “Developing recommender systems with the consideration of product profitability for sellers”. *Information Sciences* 178 (4): 1032–1048.

Chen, Z., H. Li, S. C. Kong, J. Hong, and Q. Xu. 2006. “E-commerce system simulation for construction and demolition waste exchange”. *Automation in Construction* 15 (6): 706–718. Knowledge Enabled Information System Applications in Construction.

Chunyuan, K. 2020. “Simulation Analysis of E-commerce Supply Chain System in Business and Trade Circulation”. In *2020 5th International Conference on Smart Grid and Electrical Automation (ICSGEA)*, 524–528.

Dadouchi, C., and B. Agard. 2018. “Lowering penalties related to stock-outs by shifting demand in product recommendation systems”. *Decision Support Systems* 114:61–69.

Guo, J., Z. Gao, N. Liu, and Y. Wu. 2018. “Recommend products with consideration of multi-category inter-purchase time and price”. *Future Generation Computer Systems* 78:451–461.

Gupta, P., P. Pranjal, and A. Sachan. 2021. "Performance Analysis and Improvement of a B2B's Supplier-Side E-Tender Driven Internal Process: A Discrete Event Simulation Study". *e-journal - First Pan IIT International Management Conference – 2018,*.

Markovic, A., S. Cavoski, and A. Novovic. 2016. "Analysis of Interactions of Key Stakeholders on B2C e-Markets - Agent Based Modelling and Simulation Approach". *Organizacija* 49 (2): 138–149. Copyright - Copyright De Gruyter Open Sp. z o.o. 2016; Características del documento - ; Tables; Última actualización - 2021-01-08.

Marzolla, M. 2004. "Libcppsim: a Simula-like, portable process-oriented simulation library in C++". *ESM* - (1): 222–227.

Menascé, D., V. Almeida, R. Fonseca, and M. Mendes. 2000, 09. "Business-oriented resource management policies for e-commerce serves". *Performance Evaluation* 42:223–239.

Mitrevski, P., and I. Hristoski. 2016, 07. "Evaluation of Business-Oriented Performance Metrics in e-Commerce using Web-based Simulation".

Mohsen, O., S. Abdollahnejad, N. Sajadfar, Y. Mohamed, and S. AbouRizk. 2021. "Discrete-event simulation and data analysis for process flow improvements in a cabinet manufacturing facility". *International Journal of Simulation and Process Modelling (IJSPM)* 16 (1).

Su, X., and T. M. Khoshgoftaar. 2009. "A survey of collaborative filtering techniques". *Advances in artificial intelligence* 2009.

Tang, N., H. Benton, D. Love, P. Albores, P. Ball, J. Macbryde, N. Boughton, and P. Drake. 2004, 09. "Developing an enterprise simulator to support electronic supply-chain management for B2B electronic business". *Production Planning & Control - PRODUCTION PLANNING CONTROL* 15:572–583.

Tarnowska, K. A., and Z. Ras. 2021, Jan. "NLP-Based Customer Loyalty Improvement Recommender System (CLIRS2)". *Big Data and Cognitive Computing* 5 (1): 4.

Varsha, P. S., S. Akter, A. Kumar, S. Gochhait, and B. Patagundi. 2021. "The Impact of Artificial Intelligence on Branding: A Bibliometric Analysis (1982-2019)". *Journal of Global Information Management (JGIM)* 29 (4): 221–246.

Vuksic, V., M. Stemberger, and J. Jaklič. 2001, 12. "Simulation modelling towards e-business models development". *Transactions of The Society for Modeling and Simulation International - SIMULATION* 2.

Šperka, R., and K. Slaninová. 2012, 02. "The Usability of Agent-Based Simulation in Decision Support System of E-Commerce Architecture". *International Journal of Information Engineering and Electronic Business* 1:10–17.

AUTHOR BIOGRAPHIES

SEBASTIAN PINTO GUZMAN has a degree in Applied Engineering and he is a Master student at the University of Santiago de Chile. He works as a software developer at the CITIAPS research center. His research interests include simulation, high-performance computational and software development. His email address is sebastian.pinto.g@usach.cl.

VERONICA GIL-COSTA received her MSc (2006) and PhD (2009) in Computer Science, both from Universidad Nacional de San Luis (UNSL), Argentina. She is a former researcher at Yahoo! Labs Santiago hosted by the University of Chile. She is currently an associate professor at the University of San Luis and researcher at the National Research Council (CONICET) of Argentina. Her email address is gvcosta@unsl.edu.ar.

MAURICIO MARIN is a former researcher at Yahoo! Labs Santiago hosted by the University of Chile, and currently a full professor at University of Santiago, Chile. He holds a PhD in Computer Science from University of Oxford, UK, and a MSc from University of Chile. His research work is on parallel computing and distributed systems with applications in query processing and capacity planning for Web search engines. His email address is mauricio.marin@usach.cl.