BAU Journal - Science and Technology

Volume 2 | Issue 2
ISSN: 2706-784X

Article 5

June 2021

# OPTIMIZATION OF SUPERMARKET CHECKOUT COUNTERS USING INTEGRATED GREEDY ALGORITHM

Ramzi Fayad
*Assistant Professor, Faculty of Engineering, Beirut Arab University, Beirut, Lebanon*, r.fayad@bau.edu.lb

Hadi Abou Chakra
*Associate Professor, Faculty of Engineering, Beirut Arab University, Beirut, Lebanon*, hadi.chakra@bau.edu.lb

Follow this and additional works at: https://digitalcommons.bau.edu.lb/stjournal

Part of the Architecture Commons, Business Commons, Engineering Commons, and the Physical Sciences and Mathematics Commons

## Recommended Citation

Fayad, Ramzi and Abou Chakra, Hadi (2021) "OPTIMIZATION OF SUPERMARKET CHECKOUT COUNTERS USING INTEGRATED GREEDY ALGORITHM," *BAU Journal - Science and Technology*: Vol. 2 : Iss. 2 , Article 5.
Available at: https://digitalcommons.bau.edu.lb/stjournal/vol2/iss2/5

This Article is brought to you for free and open access by Digital Commons @ BAU. It has been accepted for inclusion in BAU Journal - Science and Technology by an authorized editor of Digital Commons @ BAU. For more information, please contact ibtihal@bau.edu.lb.

# OPTIMIZATION OF SUPERMARKET CHECKOUT COUNTERS USING INTEGRATED GREEDY ALGORITHM


## Abstract

Supermarkets worldwide are facing a service dilemma whether to increase or decrease their number of counters used at checkouts. A higher number of checkouts will undoubtedly reduce waiting time at checkout, a factor in improving customer satisfaction and service quality but this will come at a cost to the Supermarket. The work conducted in this paper will therefore focus on this tradeoff between improving the customer shopping experience versus the Supermarket Cost and profitability margins. It will do so by using an optimization algorithm that can help find the optimum number of checkouts and utilization of staff resources. The optimization algorithm uses discrete event simulation approach that applies arena integrated with Greedy algorithm, using real-life data The aim of this integration is to combine the strength of the simulation that optimize large set of feasible solutions, with the advantage of the greedy algorithm to reduce the design space of feature inputs, which would facilitate optimizing the process in the shortest time possible. The developed integrated greedy algorithm has proved successful in optimizing the staff resource efficiency as well as achieving the optimum number of checkouts.



## Keywords

Discrete Event Simulation, Greedy Algorithm, Resource Optimization, Arena Simulation, Heuristic Algorithm, System Optimization



This article is available in BAU Journal - Science and Technology: https://digitalcommons.bau.edu.lb/stjournal/vol2/iss2/5

## 1. INTRODUCTION

As the increase of customers shopping for daily groceries is beneficial to the store's profit, it may have an impact on the waiting time at the checkout counters. A rise in waiting time that if not dealt with appropriately can become an urgent problem; either increase the number of staff resources and face an increase in the staff wages and cost of running the store, or reduce the number of staff and risk customer's discontent due to overlong waiting time, not to overlook the pressure and stress on staff resource personnel during the busy period.

This study was instigated by the need of a local supermarket, where the operational manager requested to redesign the supermarket's checkout service in order to improve customer flow at checkout. Options to consider would include optimum number of staff resources as well as the possibility of introducing self-checkouts service for customers of small and regular size shopping baskets. The aim of the request was to optimize the customer visit experience by allocating staff checkout resources that reduce customer waiting times and furthermore to reduce the store's total expenses. A real challenge for supermarket operational manager is the need of powerful decision-making tools to visualize, analyze and enhance checkout process to improve service quality.

For the purpose of this study, a novel decision-making tool was developed using a discrete event simulation (Arena) integrated with Greedy algorithm to evaluate the design's performance characteristics and to look for opportunities to improve the overall efficiency and to lower the cost of the design. In order to achieve the desired goals, the model was designed to measure key aspects of the system's performance, such as system throughput capability, resource and operator utilization.

## 2. LITERATURE REVIEW

The design of a model and the choice of configuration parameters affect simulation performance and accuracy. According to Maynard and Hodson, the overall optimization techniques were developed to optimize the system, in a reasonable quantity of time that can pass through viable or unviable solution in the search space depending on the method. The "strength" of the algorithm is measured according to the ability to escape local optimums and to find good solution that can be very close to the global optimal solution. The relation of spent time and quality of solution must satisfy the expectations of the decision maker (Maynard and Hodson 2004),

There is a large number of optimization techniques used in the literature, in numerous applications, including production, Call centers, mining, etc. Several studies have considered integrating simulation to solve problems related to production and services, evaluating the effects of several design alternatives. (Tanir and Booth 1999) applied discrete event simulation to optimize call center customer service by allocating staff resources in order to reduce the customer waiting times. However, they did not consider the automated voice answering machines as substitute to operators. Others, (Rung-Chuanand et. al. 2012) used simulation optimization model integrated with genetic algorithm and data envelopment analysis to optimize resources allocation in surgical units. The performance outputs were the average patient waiting time and the system completion time. (Huggins et. al. 2014) considered an optimization model to improve resource utilization in a cancer clinic. However, the assessment of the financial aspects considered as main constraint, was not included in these studies.

(Martins et. al. 2013) aimed to optimize the number of piers and tanks used for crude oil mining. In their study, they used particle swarm optimization and compared results with genetic algorithm (GA) and OptQuest by Arena. However, OptQuest did give the optimum results despite being the slowest to converge among other methods. (Masoud et. al. 2019) considered a simulation-based optimization framework to simultaneously find the optimal facility layout design and resource allocation applicable for vegetable grafting nurseries. In their study, the optimal layout algorithms were embedded within the simulation model in order to find optimal layout design given available resources. However, authors didn't take into consideration that some of the departments were fixed and can't be changed.

(Young et. al. 2019) used Monte Carlo simulation approach to optimization of resource utilization and energy conservation in iron mines by developing a nonlinear multi-objective constrained optimization model. However, convergence was very slow even though the size of the model was relatively small.

The literature review has also revealed examples where Greedy algorithm was combined with other approaches to create a powerful optimization model. (Nakayama et. al. 2004) used generalized data envelopment analysis as a selection function in genetic algorithms to solve deterministic multi-objective optimization problems. In this study, the data envelopment analysis approach was successful to reduce the size of the population but it didn't speed up the convergence process. (Lee et. al. 2008) developed a multi-objective simulation optimization framework integrating genetic algorithms with a multi-objective computing budget allocation method to solve an aircraft spare parts allocation problem. (Cerrone et. al. 2017) applied a generalized greedy algorithm to optimize the spanning tree algorithm and the minimum vertex cover problem. (Dereventsov and Temlakov 2019) defined a class of the weak bi-orthogonal greedy algorithms to investigate the properties of convergence, rate of convergence, and numerical stability of the weak bi-orthogonal greedy algorithms. These studies proved that the greedy algorithms have managed to speed the convergence of the input parameters.

Although, the models described above present successful examples of optimization techniques, there is opportunity to consider the identified opportunities for improvement, such as the optimization convergence speed, including various parameters in the simulation as well as taking into consideration the financial aspects related to process optimization.

In this study, a new model was developed to optimize the number of various resources staff and the ideal checkout design. This model takes into consideration the financial aspects of evaluating its effect on the decision-making. It is clear from the literature review, that greedy algorithm would speed up the convergence speed and hence, it is worth integrating both greedy and discreet simulation to optimize the customer visit process by allocating staff resources reasonably in order to reduce the customer waiting times and further to reduce running expenses.

## 3. CASE STUDY

In this study, the checkouts process at a local supermarket with approximately 36,500 customers annually was considered. The customer's arrival distribution was relatively stable from one day to another during weekdays and peak during weekends following almost the same arrival distribution. At service level, the supermarket has introduced 3 checkout facilities: less than ten items, regular checkouts and quick pass self-checkouts. The normal flow of customers reaching the checkouts zone with different size trolleys and baskets can be categorized as follows:

- Customers with less than ten items (S) would either select the quick pass self-checkouts or the less than ten items checkouts.
- Customers with regular size trolleys (R) would either select the quick pass self-checkouts or the regular checkouts.
- Customers with large shopping trolleys (L) are obliged to access the regular checkouts.

## 4. The COMPUTATIONAL FRAMEWORK

Arena is a simulation and automation software, which is widely used to simulate a manufacturing or a service process in order to analyze its current performance as well as the possibility to adjust the resources allocation to observe the system behavior. The simulation model used in this study was built using the Arena visual interactive modeling. Figure 1 shows the overall layout of the built checkout systems. Entities representing customers flow through the system shown as letters depicted on the animation display as S, R, and L. The simulation model began by creating the module block needed to generate the entities following an arrival rate. The input data included realistic level of uncertainty. The data was fitted into distributions using Input Analyzer and tested using chi-square and Kolmogorov-Smirnov (K-S) goodness-of-fit tests to ensure they are a good representation of the real processes (Devore 2007).

A performance measure within the supermarket is the service level, which is defined as the percentage of customers served within some fixed time period, usually per hour or per day, for the expected number of customers reaching the checkout.

The aim of this model was to determine the optimal number of each type of checkout resources and to minimize the total daily personnel expenses. The hourly wages and the self-checkout cost were formulated in the model to assess the feasibility of the system.

Greedy algorithm integrated with Arena Simulation has the capability to reduce the design space of feature inputs in the aim to optimize the number of assigned resources. This model can be summarized as follows:

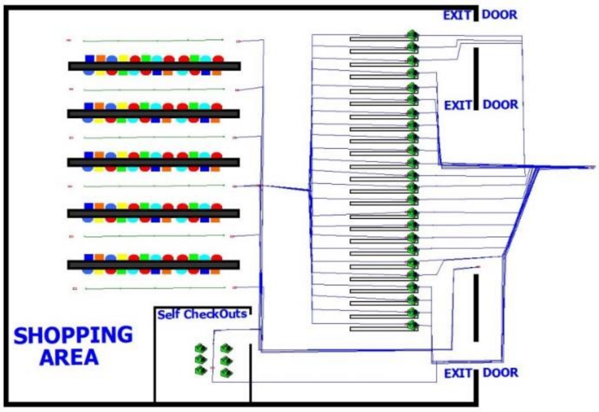


Fig.1: The layout of the checkout systems

- **Step 0**. At iteration zero, table 1 presents the greedy algorithm initiated with random starting values chosen (chosen from a range between two set values, a minimum and a maximum).

Table 1: Starting values of the greedy algorithm

| Resource Capacities | Less than Ten Items Checkouts | Regular checkouts | Quick Pass Self-Checkouts | Objective Total Cost |
|---|---|---|---|---|
| Iteration 0 | a | b | c | ObJ-0 |

- **Step 1.** Added or subtracted one value out of each resource factor presented in table 1. This resulted in 6 possibilities to consider; refer to table 2.

Table 2: The six possible resource factor values

| Resource Capacities | Less than Ten Items Checkouts | Regular checkouts | Quick Pass Self-Checkouts | Objective Total Cost |
|---|---|---|---|---|
| Possibility 1 | **a+1** | b | c | ObJ-poss 1 |
| Possibility 2 | **a-1** | b | c | ObJ-poss 2 |
| Possibility 3 | a | **b+1** | c | ObJ-poss 3 |
| Possibility 4 | a | **b-1** | c | ObJ-poss 4 |
| Possibility 5 | a | b | **c+1** | ObJ-poss 5 |
| Possibility 6 | a | b | **c-1** | ObJ-poss 6 |

- **Step 2.** Using Arena, six different objective total cost values were obtained from simulating all six combinations presented in table 2.
- **Step 3.** Comparison was carried out between the six different objective total cost values with the objective function ObJ-0 presented in table 2. If any of the six values was smallest than ObJ-0, the model considered the new corresponding resources capacities as value of the next iteration. Table 3 presents the new resource values of the greedy algorithm for the second iteration. For the new resource factor values (Iteration 1), step 1 to 3 must be repeated.

Table 3: The values of resource factor of the second iteration

| Resource Capacities | Less than Ten Items Checkouts | Regular checkouts | Quick Pass Self-Checkouts | Objective Total Cost |
|---|---|---|---|---|
| Iteration 0 | a | b | c | ObJ-0 |
| Iteration 1 | Value related to minimum ObJ | Value related to minimum ObJ | Value related to minimum ObJ | Minimum value of ObJ |

- **Step 4.** If none is smallest than the current objective total cost value, as a result, do consider the resource values as the optimized values. End Algorithm

The average measured value of the time spent queuing at the checkouts was exported to excel through visual basic under Arena. Those values were used to calculate the objective function of the scenario considered.

Eq. (1) Total Cost = Total Service Cost + Total Waiting Time Cost + Self Checkouts Cost + Regular Checkout Costs

Where:

Eq. (2) Total Service Cost = Labor Costs = Wages per hour

Eq. (3) Total Waiting Cost = ($\lambda$) ($w_q$) (Dissatisfaction Cost)

- $\lambda$ – Rate per unit time at which events (arrivals or departures) are generated.
- $w_q$ – Expected waiting time in queue

Eq.(4) Self-Checkouts Cost = Initial Cost + Maintenance Cost

Eq. (5) Total Regular Checkouts Cost = Initial Cost + Maintenance Cost

- The initial costs include the price of the checkout, shipping and installation

## 5. DATA COLLECTION

Figure 2 depicts the principle arena program related to the main logic. Entities representing customers are classified into 3 categories: L, R and S letters flow through the checkout process. According to the supermarket operational manager, around 63% of the daily customers are with small or regular shopping baskets. These customers are grouped into (S-R) categories. The remaining 37% are customers with large shopping baskets. These customers are grouped into (L) category.

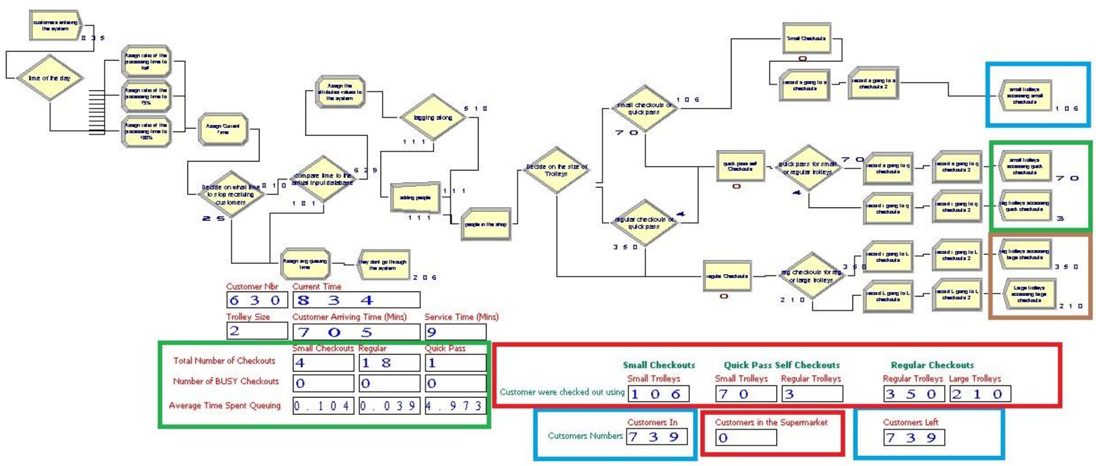


Fig.2: The principle arena program related to the main logic

The model started from a create block, in which entities were generated following the distribution obtained using the input analyzer. The input analyzer provided numerical estimation of the appropriate parameters. The histogram and the distribution for both the arrival and the service rates were provided in the table 4 and the figures 3.

Table 4: Input analyzer values for both the arrival and the service rates

| Arrival Time | Time of process start | Time of process end | Process (service) Time | Waiting Time | Inter Arrival Time | Total Time |
|---|---|---|---|---|---|---|
| 8:05 | 8:05 | 8:08 | 0:03 | 0:00 | | 0:01 |
| 8:06 | 8:06 | 8:12 | 0:06 | 0:00 | 0:01 | 0:06 |
| 8:08 | 8:08 | 8:11 | 0:03 | 0:00 | 0:02 | 0:03 |

The checkout staff resource allocation were defined in an excel sheet spreadsheet; refer to figure 4. The VBA was incorporated within the arena model to allow for an easy way to change parameters without the need to make changes to the simulation model itself. Data were gathered over a period of 5 weeks. The various cost parameters implemented in the objective function are shown in the figure 4. The input data were attained from the supermarket operation manager, and they were considered a good approximation of the accurate figures due to confidentiality agreement. The supermarket opened 14 hours every single day of the week. The labor daily wage was $20. The initial costs for a single self-checkout and regular checkout were $3,000 and $2,000 respectively.

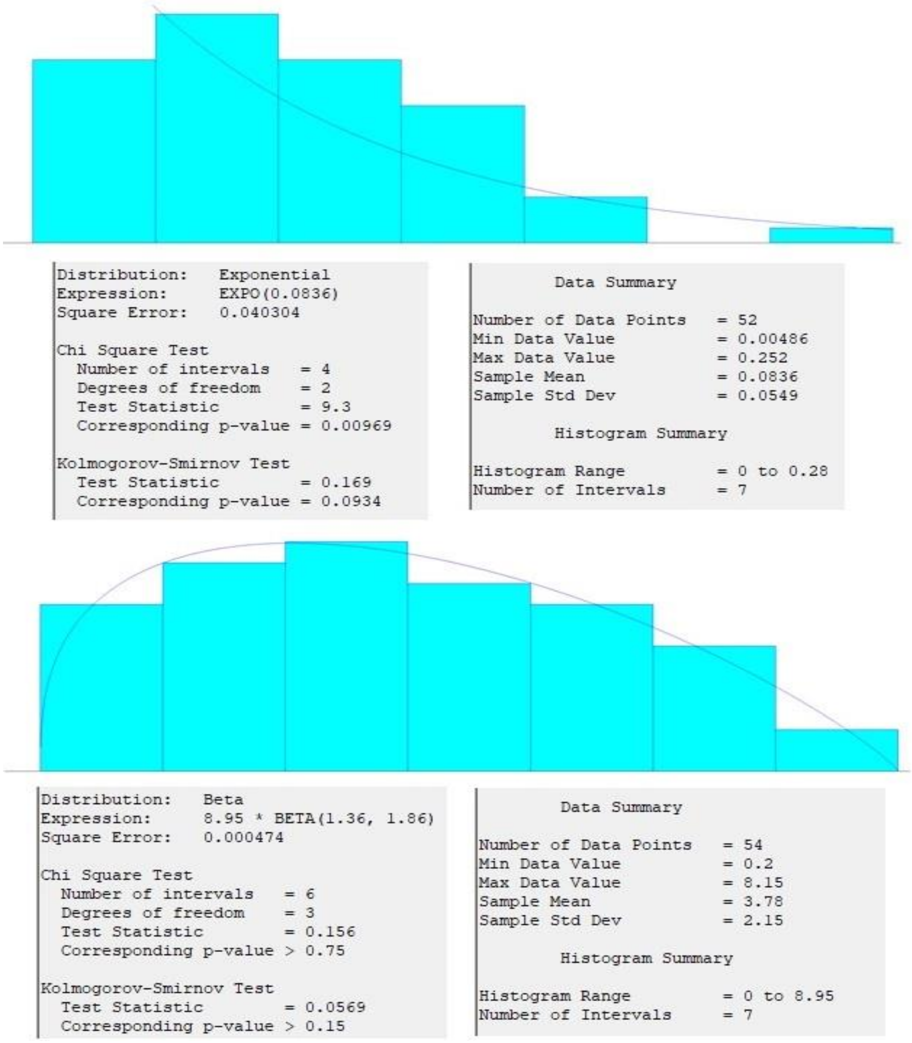


Fig.3: The histogram and the distribution of the input analyzer for both the arrival and the service rates

The initial costs were considered as uniform annual installment over a period of 10 years with an interest rate equivalent to 15 %. Hence, the uniform annual cost is equal to $597 for the self-checkout and $398 for regular checkout. The maintenance costs were equal to $250 and $100. The service life of the checkout tills is 10 years while salvage value after 10 years is zero.

| | | Cost per 1 Self-Checktout | | Cost per 1 Regular-Checktout | |
|---|---|---|---|---|---|
| Daily Labor Cost ($) | $20 | Initial Cost ($) | $3,000 | Initial Cost ($) | $2,000 |
| | | Annual Maintenance Cost | $250 | Annual Maintenance Cost | $100 |
| Interest Rate | 15% | (A/P) | 0.199 | (A/P) | 0.199 |
| Salvage Value | $0 | Uniform Annual Cost | $597 | Uniform Annual Cost | $398 |
| Number of Years | 10 years | Daily Cost + Maintenance | $2.32 | Daily Cost + Maintenance | $1.43 |
| | | | | | |
| Que. ($)/Min | $Q < 2$ | $2 \le Q < 7$ | $7 \le Q < 10$ | $10 \le Q$ | |
| | 0.0005 | 0.0006 | 0.0008 | 0.001 | |

Fig.4: Cost parameters

The capital recovery cost (A/P, i, n) converts initial cost at year 0 into a series of equivalent uniform annual year-end values, where i is the annual interest rate and n in the number of years in the service life of the equipment. The customer dissatisfaction cost per minute was quantified as $0.0005 per min if the average waiting time per customer is less than 2 minutes. For the average waiting time per customer between 2 and 7 minutes, the dissatisfaction cost per customer was $0.006 per min. For the average waiting time per customer between 7 and 10 minutes, the dissatisfaction cost per customer was $0.0008 per min. Finally, the dissatisfaction cost per customer was quantified as $0.001 per min when the average waiting time per customer is more than 10 minutes.

## 6. RESULTS AND DISCUSSION

The model was implemented using Arena 15 and was ran for 100 replications for each iteration. Table 5 presents the values used in setting the resources capacities. Bearing in mind the initial values listed in the table are arbitrary starting values and do not represent the current number of checkouts. The objective total cost was set to a large value. These values were chosen to start the integrated greedy algorithm.

Table 5: The values resource factor used in iteration 0

| Resource Capacities | Less than Ten Items Checkouts | Regular checkouts | Quick Pass Self-Checkouts | Objective Total Cost |
|---|---|---|---|---|
| Iteration 0 | 3 | 15 | 6 | 99999 |

The integrated greedy algorithm within Arena simulation was considered for its capability to reduce the design space of feature inputs and to optimize the number of assigned resources. The optimal number of each type of checkout resources and the minimum total daily personnel expenses were calculated. The output resources allocation of the integrated greedy algorithm for iteration 1 is shown in figure 5, which shows a screenshot of the excel spreadsheet highlighting various colored boxes as follows:

- The green box presents the initiated random checkout staff resource assigned values and a very large value as Obj-0
- The blue box on the right shows the average queue cost as dissatisfaction cost, labor cost and self-checkout costs
- The blue box on the left shows the 6 possibilities obtained from adding and subtracting one value out of each resource factor. The value in the right column is the sum of the values shown in the blue box on the right.
- The red box shows the resources value of the greedy algorithm for both the first and second iteration.

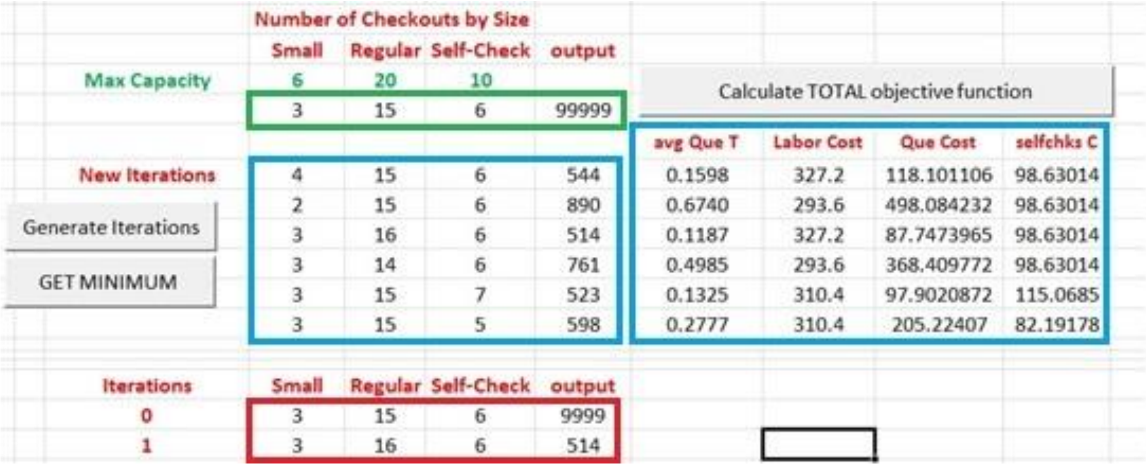


Fig.5: Resources allocation of the greedy algorithm for iteration 1

The six different scenarios of alternating the capacities of resources obtained using the integrated greedy algorithm is presented in table 6.

Table 6 shows clearly that Possibility 3 has the least objective total cost, hence, its values must be considered for the next iteration. Table 7 presents the new resource values of the integrated greedy algorithm for the second iteration.

Table 6: The six resource factor values obtained using the Greedy algorithm for Iteration 0

| **Resource Capacities** | **Less than Ten Items Checkouts** | **Regular checkouts** | **Quick Pass Self-Checkouts** | **Objective Total Cost** |
|---|---|---|---|---|
| Possibility 1 | **4** | 15 | 6 | $544 |
| Possibility 2 | **2** | 15 | 6 | $890 |
| Possibility 3 | 3 | **16** | 6 | $514 |
| Possibility 4 | 3 | **14** | 6 | $761 |
| Possibility 5 | 3 | 15 | **5** | $523 |
| Possibility 6 | 3 | 15 | **4** | $598 |

Table 7: The new resource factor values for iteration 1 (Possibility 3)

| **Resource Capacities** | **Less than Ten Items Checkouts** | **Regular checkouts** | **Quick Pass Self-Checkouts** | **Objective Total Cost** |
|---|---|---|---|---|
| Iteration 0 | 3 | 15 | 6 | $99999 |
| Iteration 1 | 3 | 16 | 6 | $514 |

The outcomes of the second iteration are presented in table 8. The optimal number of each type of checkout resources and the total cost were calculated. Table 8 presents the six different various scenarios of alternating the capacities of resources, which were obtained using the integrated greedy algorithm for iteration 1.

Table 8: The six resource factor values obtained using the Greedy algorithm for Iteration 1.

| **Resource Capacities** | **Less than Ten Items Checkouts** | **Regular checkouts** | **Quick Pass Self-Checkouts** | **Objective Total Cost** |
|---|---|---|---|---|
| Possibility 1 | **4** | 16 | 6 | $519 |
| Possibility 2 | **2** | 16 | 6 | $869 |
| Possibility 3 | 3 | **15** | 6 | $629 |
| Possibility 4 | 3 | **17** | 6 | $496 |
| Possibility 5 | 3 | 16 | **7** | $494 |
| Possibility 6 | 3 | 16 | **5** | $533 |

Table 8 shows clearly that Possibility 5 has the least objective total cost, hence, its values must be considered for the next iteration. Table 9 presents the new resource values of the integrated greedy algorithm for the second iteration.

The optimal number of each type of checkout resources and total cost for iteration 2 were calculated to optimize number of assigned resources for the next iteration. This proves that the integrated greedy algorithm has managed to optimize parameters over numerous iterations. The optimization has converged once the six objective-total cost obtained values were higher than the previous iteration. Once convergence is achieved, the greedy algorithm run would stop and the optimized number of staff resources would be the corresponding resources value of the previous iteration. Table 10 depicts the checkout resource values for 185 iterations.

Table 9: The new resource factor values for iteration 2 (Possibility 5)

| Resource Capacities | Less than Ten Items Checkouts | Regular checkouts | Quick Pass Self-Checkouts | Objective Total Cost |
|---|---|---|---|---|
| Iteration 0 | 3 | 15 | 6 | $99999 |
| Iteration 1 | 3 | 16 | 6 | $514 |
| Iteration 2 | 3 | 16 | 7 | $494 |

Table 10: The resource factor values up to iteration185

| Resource Capacities | Less than Ten Items Checkouts | Regular checkouts | Quick Pass Self-Checkouts | Objective Total Cost |
|---|---|---|---|---|
| Iteration 0 | 3 | 15 | 6 | $99999 |
| Iteration 1 | 2 | 15 | 6 | $412 |
| Iteration 2 | 2 | 14 | 6 | $392 |
| | | | | |
| | | | | |
| Iteration 184 | 1 | 9 | 5 | $279 |
| Iteration 185 | 1 | 8 | 5 | $261 |

The integrated greedy algorithm has managed to optimize the system by reaching convergence in 185 iterations. For the purpose of verification, the results obtained in iteration 185 were considered as the starting solution iteration to investigate whether the steady state has been achieved.

Table 11: Integrated greedy algorithm at Iteration 185 has reached steady state

| Resource Capacities | Less than Ten Items Checkouts | Regular checkouts | Quick Pass Self-Checkouts | Objective Total Cost |
|---|---|---|---|---|
| Iteration 185 | 1 | 8 | 5 | $261 |
| | | | | |
| Possibility 1 | | **Same as Starting Solution** | | |
| Possibility 2 | **2** | 8 | 5 | $281 |
| Possibility 3 | 1 | **7** | 5 | $267 |
| Possibility 4 | 1 | **9** | 5 | $283 |
| Possibility 5 | 1 | 8 | **4** | $268 |
| Possibility 6 | 1 | 8 | **6** | $291 |

As shown in table 11, the integrated greedy algorithm at iteration 185 has managed to optimize the model by convergence of the objective total cost into steady state. A final decision might be made, by reviewing results of the various scenarios. Iteration number 185 was most favorable because of the minimum expenses and the reduction in the number of staff resource checkouts. At the start of the study, the operational manager had to face the dilemma of hiring more staff checkout resources to reduce customers' waiting time and enhance their satisfaction. However, the integrated greedy algorithm used in this study proved that such objectives can be achieved, with no added cost of increasing number of personnel.

During the 5 weeks period of data collection, a survey was carried out to find out the percentage of customers interested in using self-checkout. The outcome of this survey revealed that out of 63% of total customers with small or regular shopping baskets around 40% shows preference to use self-checkouts, while the remaining customers do not have a preferred checkout. The Arena model was able to optimize the system where around 37% of all customers have opted to use the self-checkouts. This percentage resulted in operational cost reduction, due to the fact that less staff resources needed and more self-checkout counters have been introduced. Table 12 presents the outcomes of the Arena model, which clearly showed the number of customers willing to use the self-checkouts.

Table 12: Observed customer shopping habits and preference to use Self-Checkouts vs model results

| **Observed Customer Shopping Habits and Preference to Use Self-Checkouts** | | |
|---|---|---|
| Small and Regular Shopping Baskets | 63% | 40% Self-Checkouts |
| | | 23% any type of checkouts |
| Large Shopping Baskets | 37% | 37% any type of checkouts |
| Model Results | | |
| Small and Regular Shopping Baskets | 63% | 37% Self-Checkouts |
| | | 26% any type of checkouts |
| Large Shopping Baskets | 37% | 37% any type of checkouts |

Furthermore, the integrated greedy algorithm proved that by introducing self-checkout and optimizing the numbers of other checkouts would lead to improving system efficiency by reducing the average waiting time at checkouts counter and optimizing resource utilization. Table 13 presents the staff resource utilization and average waiting time before and after implanting the self-checkouts.

Table 13: Expected Results of Quick Pass Self-Checkouts in Arena

| Self-Checkouts Implemented | Staff Resource Utilization | Average waiting time at the checkout counters |
|---|---|---|
| Before | 95.57% | 6.25min/customer |
| After | 73.35% | 5.17 min/customer |

The integrated greedy algorithm used in this study was able to speed up the optimization process of the supermarket. The algorithm was able to assess various scenarios with respect to staff resource efficiency, total cost, self-checkouts and reducing waiting time. The outcome of the study proves the importance of implementing self-checkouts, which resulted in reduction in the waiting time that lead to customer satisfaction, and reduction in total operational cost and better utilization of staff resource which lead to operational satisfaction. The improvement of 22.22% in resources efficiency means that the Store management can redirect the extra capacity of their resources for shelf stacking or for customer services, which will further improve Customer satisfaction. The store management could also look to reduce their number of staff required to operate the shop and might consider passing on some of the savings to the customer, to invest in more automation and store improvement or to simply increase their margin of profitability. In addition, a reduction in the average waiting time has been achieved17%. And due to the advantages of the greedy integrated arena algorithm, the shop owners can see clearly the trade-off between resources efficiency and reduction of waiting times allowing them to make an educated decision based on these facts and where it is best to invest. Should they choose to use some of the savings generated through staff efficiency and invest in implementing additional self-checkouts, this algorithm has proved that there will be more benefits to be had.

## 7. CONCLUSIONS

A model of checkout service at a local supermarket has been developed to examine the service level target by using integrated greedy algorithm within Arena simulation. This is a novel approach as it is unique in the way it integrates greedy algorithm with Arena, culminating their respective unique strengths. The integrated greedy algorithm was able to reduce the design space made of all possible input parameters to locate the optimum system parameters.

In this study, Excel spreadsheet and VBA within Arena were used to facilitate adjusting the checkout staff resources allocation to observe the system behavior. The integrated greedy algorithm was able to assess the various scenarios with respect to their objective total cost values. It has found that the proposed procedure is applicable and effective to the considered application. The integrated greedy algorithm used in this study managed to implement staff resource efficiency improvement and self-checkout as an improved quick pass process.

Various models have been used to speed up the optimization processes; however, in this study an integrated greedy algorithm was used to speed up the optimization of the number of resources and self-checkout in a supermarket. The outcomes of this study show clearly that the integrated greedy algorithm was an efficient tool to speed up the optimization process of resources and self-checkout.

The novel model used in this study proved that a greedy algorithm integrated with Arena simulation is able to speed up the optimization process. The usage of the integrated greedy algorithm was a success because it was able to replicate to a very high degree how the real system would operate. The new integrated greedy algorithm is an excellent tool that the operational manager can use to reassign any system and develop new improved working settings. The integrated greedy algorithm has proven its robustness and can be applied in manufacturing or service industries to improve their efficiencies by optimizing the number of workers, machines and production flow.

## REFERENCES


- Cerrone, C., Cerulli, R., & Golden, B. (2017). Carousel greedy: a generalized greedy algorithm with applications in optimization. *Computers & Operations Research*, *85*, 97-112.
- Dereventsov A.V., and Temlakov V.N., (2019). A Unified Way of Analyzing Some Greedy Algorithms. Elsevier, *Journal of Functional Analysis*, 277.
- Devor, L., (2007) Probability and Statistics for Engineering and the Sciences, Duxbury Press.
- Huggins, A., Perez E., and Claudio D., (2014). Improving Resource Utilization in a Cancer Clinic: An Optimization Model. Conference: *Industrial and Systems Engineering Conference. Jan 2014*
- Lee, L.H., Chew E.P., Teng S., and Chen Y., (2008). Multi Objective Simulation based evolutionary algorithm for an aircraft spare parts allocation problem. *European Journal of Operations Research*, 189, 476-491
- Hamzadayi, A., & Yildiz, G. (2013). A simulated annealing algorithm based approach for balancing and sequencing of mixed-model U-lines. *Computers & Industrial Engineering*, *66*(4), 1070-1084.
- Masoud, S., Chowdhury, B. D. B., Son, Y. J., Kubota, C., & Tronstad, R. (2019). Simulation based optimization of resource allocation and facility layout for vegetable grafting operations. *Computers and Electronics in Agriculture*, *163*, 104845.
- Maynard H.B, and Hodson W.K., (2004). Maynards Industrial Engineering Handbook. McGraw-Hill.
- Nakayama, H., Yun, Y.B., Tanino T., and Arakawa M., (2004). Multiple criteria decision making with generalized DEA an aspirational method. *European Journal of Operational Research*, 158, 697-706
- Rung-Chuan L., Mustafa Y.S., and Kalyan S.P., (2013). Multi-Objective simulation optimization using data envelopment analysis and genetic algorithm: specific application to determining optimal resource levels in surgical services. *Elsevier, Omega*, 41, 881-892
- Tanir O., and Booth R.J., (1999). Call center simulation in Bell Canada. In Proceedings of the 1999 *Winter Simulation Conference*, ed. P. A. Farrington, H. B.
- Yong, H, Liao, N, Rao, J, Fu, F, (2019) The optimization of investment strategy for resource utilization and energy conservation in iron mines based on Monte Carlo and Intelligent Computation. Elsevier, *Journal of Cleaner Production*, 232, 672-69.