Journal of Systems Thinking in Practice
JSTINP, *QUARTERLY*
Homepage: jstinp.um.ac.ir
Online ISSN: 2821-1669, Print ISSN: 2980-9460
Research Article
DOI: 10.22067/JSTINP.2024.83302.1064
JSTINP 2024; Vol. 3, No. 2

A Spatial Agent-Based Consumer Model: Maximizing and Satisfying Behavior within Multi-Store Market

Omid Roozmand[a], DG Webster[b], Saeed Abdolhosseini[c*]

[a] *Department of Computer Engineering, Shahreza Campus, University of Isfahan, Isfahan, Iran.*
[b] *Environmental studies program, Dartmouth College, New Hampshire, USA.*
[C] *Demaptment of Management, Faculty of Administrative Sciences and Economics, University of Isfahan, Isfahan, Iran.*

How to cite this article

Roozmand, O; Webster, D., Abdolhosseini, S., 2024. A Spatial Agent-Based Consumer Model: Maximizing and Satisficing Behavior within Multi-Store Market, *Journal of Systems Thinking in Practice*, *3*(2), pp. 36-60. doi: 10.22067/jstinp.2024.83302.1064.
URL: https://jstinp.um.ac.ir/article_45095.html

ABSTRACT

In this paper, we propose using a mixed genetic-floyd-warshall algorithm in combination with a Floyd-warshall algorithm to model the satisficing behaviour of consumers across spatially differentiated stores. Consumer agents can pick a basket of goods from different stores to either maximize their utility or to "satisfice" by selecting the first basket with a utility that is higher than their satisfaction threshold. The Floyd-warshall algorithm is used to find the shortest path between two chosen stores by considering travel cost. Factors such as price, quality of goods, the cost of travel to the store, consumers' decision-making preferences, and store locations play significant roles in the decision-making process of consumer agents. The model is tested based on mechanisms at the individual level to show how the model works and at the macro-level to reproduce foundational theories in economics.

Keywords

Economic modeling, Consumer behavior, Knowledge-based decision, Spatial agent-based modeling.

Article history

Received: 2023-07-08
Revised: 2024-03-18
Accepted: 2024-04-09
Published (Online): 2024-06-24

Number of Figures: 13 **Number of Tables: 11** **Number of Pages: 25** **Number of References: 26**

****Corresponding author***
Email: Saied.abdolhoseini@ase.ui.ac.ir

1. Introduction

Companies seek information on consumer behaviour so that they can tailor their marketing strategies to maximize sales (Bernabé-Moreno et al., 2015). Consumer choice involves at least six different stages, including need recognition, search for information, evaluation of alternatives, decision to purchase, purchase, and post-purchase learning. Within these stages, there are underlying cognitive processes that determine why consumers buy things. Psychological factors can affect consumers' purchasing rules and may lead the consumer to buy goods with near optimum utility rather than maximum utility. In this paper, we compare two of the most important purchasing rules: satisficing and maximizing. In satisfying behavior, consumers tend to buy any basket of goods with utility higher than their satisfaction threshold, considering their budget constraint (Schwartz et al., 2002). On the other hand, maximizing consumers select the basket of goods with the highest possible utility within their budget. Therefore, satisficers may buy goods more for convenience and enjoyment but consequently pay more (Schwartz et al., 2002). This is a form of bounded rationality (Simon, 1955; Schwartz, 2008).

There are many efforts to model consumer behavior. However, to the best of our knowledge, there is no agent-based model in which consumer agents choose their goods from spatially differentiated stores to maximize their utility or reach their satisfaction threshold. Selecting a basket of goods to maximize utility from different stores is a subset of the unbounded knapsack problem and is an NP-hard problem that has yet to be solved (Neapolitan and Naimipour, 2004). The NP-hard problem can be solved with pseudo-polynomial time complexity (Neapolitan and Naimipour, 2004).

To clarify the problem related to a multi-market situation, we use an example. Assume that in the bidirectional graph (Figure 1), nodes (rectangles) stand for stores and edges for the cost of moving from one node to another (including distance, gas, etc.). The first node (circle) indicates the consumer's location. In this situation, maximizers have the NP-hard problem of determining which stores they should select in order to maximize their net benefit from a purchase while minimizing search and travel costs. Satisficers do not seek to maximize, and so should have an easier time finding a satisfactory basket of goods, but even this is difficult a priori.

We solve both problems by breaking down the decision into two interrelated processes. We first use a Floyd-warshall algorithm as part of genetic algorithm to find the shortest (least expensive) path for each basket of goods and then incorporate this as part of the total utility for each basket of goods. We then incorporate this into a genetic algorithm that finds the satisfactory combination of path and purchase options for the entire system. This is similar to the way in which people learn the attributes of a spatial context and then use that information to plan their shopping trips based on what they need at any given point in time (Dellaert et al., 1998).

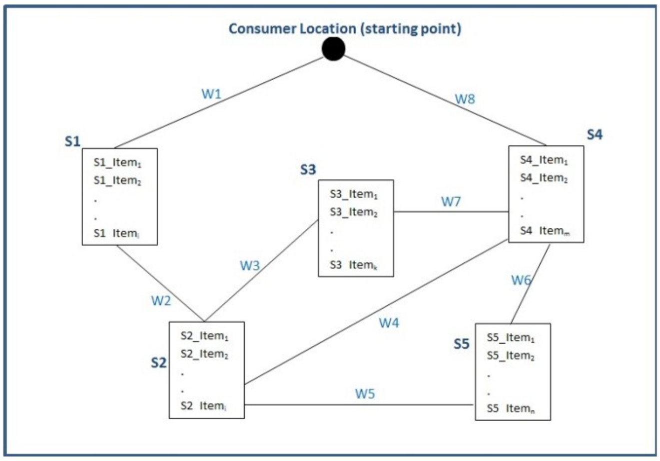

Figure 1. Example of market structure-problem definition

We use agent-based modeling (ABM) to implement this algorithm. ABM is an analytical technique for modeling complex systems (Gilbert, 2008). Agents are autonomous entities with an internal decision-making process. Taking human cognitive factors into account is important to make humans like agents, and ABMs are often used to explore the macro-level outcomes of individual-level decision processes (Fujita, Hakura and Kurematu, 2009). Most early ABMs had no real spatial component, as the focus was on developing more and more sophisticated decision processes. For instance, in their seminal paper, Jager and Janssen (1999) propose a consumer decision-process in which consumer agents are equipped with four cognitive processes: deliberation, imitation, social comparison, and repetition. Although the paper proposes a model of the agent's decision process, it mostly focuses on the internal part of the agent's mind rather than the environment. Goods are abstractly generated without physical location, and stores and agents choose from a single good category rather than a basket of goods. This simplification was useful for initial model runs but could be extended by incorporating spatial attributes into the environment and the decision process (Wang et al., 2021).

Zhang and Zhang (2007) created a model in which consumer agents choose a suitable good when encountering competing brands in a market with only a few stores. This allows them to evaluate the decoy effect, in which consumers change their preference for ordering two goods due to the presence of a third good that is asymmetrically dominant (inferior in all but one attribute). Agents make decisions based on a motivation function and are equipped with personality traits, but the model is not able to find the best basket of goods among all available stores. Also, there are no transportation costs in this model, so maximizing occurs only over the attributes of the goods sold, not the spatial location of the stores (Zhang and Zhang, 2007).

Roozmand et al. (2011) propose a model based on culture and personality for consumer agents' decision-making processes. The core of the model is human needs, where agents are motivated based on those needs which are important in their culture (Roozmand et al., 2011). It models the power distance dimension of Hofstede's model of national culture, social status and social responsibility needs, and the extroversion, agreeableness and openness of McCrae and Costa's five-factor model (McCrae and Costa, 1983, 1996, 2003; Hofstede and Hofstede, 2005; Hofstedeet al., 2010). The model is validated in eleven European countries, even though the results do not fit two other countries, Great Britain and the Netherlands. All in all, there is great detail regarding consumer decision processes in the model, but it does not consider the location of stores. Also, the consumer agent buys one good at a time and is not able to choose a basket of goods.

ABMs have also been used to explore the role of trust in market transactions. Roozmand et al. (2007) describes a market model including buyer and seller agents. Buyer agents apply reinforcement learning to model reputable sellers and seller agents use reinforcement learning to model reputable buyers. Similarly, Khosravifar et al. (2012) propose a trust model for computing agents which is evaluated with service consumer agents based on different trust models.

All of the above decision processes are more difficult when consumer agents maximize over a basket of goods rather than one good or a few different brands. Baptista et al. (2014) describe a greedy algorithm to solve the problem of finding the maximized basket of goods. The proposed algorithm does not always provide an optimal solution; however, it has linear complexity. Roozmand and Webster (2014) use a dynamic algorithm to find the best basket of goods, namely a basket of goods with the highest possible utility. They also used a genetic algorithm to find the satisficing basket of goods. Consumer multichannel choice behavior has been studied in the research conducted by Sonderegger (Sonderegger-Wakolbinger and Stummer, 2015). Heterogeneity of customers, social dynamics, and different purchasing stages have been taken into account in this research (Sonderegger-Wakolbinger and Stummer, 2015). Here again, although these models are built on sophisticated cognitive models, they do not include the spatial component.

The incorporation of spatial elements began in parallel with some of the most advanced decision-processes described above. Here, we review a few of these spatially explicit models of consumer decision-making, along with spatial economic agent-based models. Tsekeris and Vogiatzoglou (2011) propose a model based on new economic geography that simulates the complex interactions of household and firm location choices based on agent-based modeling. It considers transport costs as part of agents' decision-making processes in a system of cities. The model contains households, the production sector, firms, and central and local government agents. Household agents, as end consumers, use a utility function to choose a basket of goods that is affordable given their budget constraints. However, the model is not able to find the best basket of goods.

Also, He et al. (2014) propose an agent-based spatial model including four types of agents: the world, manufacturers, firms and consumers. In this paper, firms utilize a genetic algorithm to evaluate their location and pricing strategies. Therefore, the position of firms evolves over time to find the optimal location. The manufacturer agent provides an infinite quantity of goods for consumer agents. The world agent contains all other agents and is used to update the values of all the variables, such as price, position and other endogenous parameters, as well as provide results for later analysis. Consumer agents use a utility function to find a suitable basket of goods considering their budget constraints. The utility function contains the price of goods from each firm and the distance to that firm. Consumer agents' positions are fixed. The main problem of such consumer agent algorithm is that it does not find the optimal product basket. Consumer agents always choose one firm to purchase all of the goods that they need. However, the best basket of goods might be found in different stores, and so the consumer may improve utility by going to different stores as long as travel costs are relatively low (He et al., 2014).

According to the reviewed research, optimizing baskets of goods has not been studied in a spatially differentiated stores where stores are located at different geographical locations. In this case, not only the utility and price of each good should be considered, but also the time and cost of traveling from one store to another. In this paper, we apply a mixed genetic-floyd-warshall algorithm (Chen and Jian, 2007). The paper is organized as follows: Section 2 describes the proposed algorithm. In subsection 2-1, we use an example to show how the algorithm works. Section 3 provides the results, including verification to show the model works properly (subsection 3-1) and validation based economics theories (subsection 3-2). Finally, in section 4 we conclude the paper and propose future works.

2. Proposed algorithm

As explained in the introduction, maximizing consumer choice across a range of stores is a complex decision process and an NP-hard problem. There are a few techniques that could be used to search the problem space to find the best or satisficing basket of goods, such as brute-force which needs to search the whole solution space, dynamic programming, or local search algorithms. This problem is similar to the unbounded knapsack problem (Chen and Jian, 2007; Srisuwanna and Charnsethi, 2007) where consumer agents can choose one or more goods from different types as far as the price does not exceed the consumer's budget; therefore, dynamic programming is a solution. However, in knapsack problem, dynamic programming is still time-consuming and, moreover is usually used for maximizing utility (Neapolitan and Naimipour, 2004) rather than satisficing problems. Therefore, we apply dynamic programming for minimizing the path among stores and genetic algorithms to find the satisficing basket of goods (Roozmand and Webster, 2014). However, the proposed dynamic programming model by (Roozmand and Webster, 2014) must be modified to accommodate multiple stores with different spatial locations (and therefore transport costs).

This spatially-explicit genetic algorithm should take two features into account: the net utility from various items and the shortest path regarding transport costs. We use the Floyd-Warshall algorithm (Neapolitan and Naimipour, 2004) to find the shortest path between any two nodes in a weighted graph, which is explained in details later. That is then included in the total utility for each basket of goods, which allows us to efficiently use a genetic algorithm to find the satisfactory combination of goods and stores at each decision point.

Assume that we have n number of stores. The following vector shows the stores in Equation 1:

$$Stores = < S_1, S_2, \dots, S_n > \quad (1)$$

In which S_i indicates store i. Each store contains items as shown in Equation 2:

$$S_i = \{S_{i}\_item_1, S_{i}\_item_2, \dots, S_{i}\_item_{k_i}\} \quad (2)$$

Which $S_{i}\_item_j$ shows the jth item in store i and ki is the maximum number of items in store i. Each product has a specific quality and price (Equation 3):

$$item_{ij} =< q_{ij}, p_{ij} > \quad (3)$$

Where $q_{ij}$and p_{ij}show the quality and price of item j of store i, respectively.

Stores are connected to each other by roads. We assume the roads between stores are bidirectional and both directions can have different weights and thus are not symmetrical. The adjacency matrix represents how stores are connected to each other. This matrix is used to calculate the shortest path by the use of Floyd-Warshall algorithm to move from one store to another (one node to another). Figure 2 shows the matrix.

$$\begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & \cdots & & n \end{array} \\ \begin{array}{c} 1 \\ 2 \\ \vdots \\ n \end{array} & \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{pmatrix} \end{array}$$

Figure 2. The matrix is used for finding shortest path by the use of Floyd-warshall algorithm

Where numbers 1 to n are used as identifiers for each store (or node), and weight w_{ij}shows the cost for moving from store i to store j. The cost is the sum of all costs including distance, gas, time, etc which for the sake of simplicity we use a simple value w_{ij}.

As we mentioned before we use a genetic algorithm. The key element in genetic algorithm is choosing the structure of the chromosome. The chromosome in our model is defined as follows:

$$\text{CH} = \{<item_{11}, item_{12}, \dots, item_{1k_1}>, <item_{21}, item_{22}, \dots, item_{2k_2}>, \dots, <item_{n1}, item_{n2}, \dots, item_{nk_n}>, <sel_1, sel_2, \dots, sel_n>\} \quad (4)$$

In which, $item_{ij}$ indicates the item j from store i. $K_i$is a constant value and shows the maximum number of items in the store i. Therefore, the first set includes the items of store 1, the second set contains items of store 2, and so on. The last part of chromosome $<sel_1, sel_2, \dots, sel_n>$ shows the selected stores. Having stores in the chromosome allows the agent to calculate the costs of travel as part of total cost of each chromosome. sel_i is a binary variable. If sel_i is 0, it means that consumer does not choose the store i and does not buy anything from that store, and if 1, it means that consumer buys at least one item from store i. The total benefits of the chromosome is calculated as follows:

$$TotalBenefits = \sum_{i=1}^{n}\sum_{j=1}^{k_i} n_{ij} * q_{ij} \quad (5)$$

Where i counts the stores, and j the items in that store. Therefore, n_{ij} indicates the number of item j selected from store i, and q_{ij} is the quality of the item. Also, the total cost of the chromosome is calculated as the sum of all prices of items as well as moving costs between chosen stores:

DOI: 10.22067/JSTINP.2024.83302.1064

$$TotalCosts = \left[\sum_{i=1}^{n}\sum_{j=1}^{k_i} n_{ij} * p_{ij}\right] + MC \tag{6}$$

MC stands for minimum cost of travel and is the shortest path among all chosen stores. There might be many routes to travel through the selected stores, however, consumer agents in this model optimize their travel cost by selecting the shortest path among the stores from which they choose to buy goods. Thus, for any suite of stores that a consumer would choose to visit, the Floyd-warshall algorithm can be used to find the shortest path. The genetic algorithm then sorts chromosomes based on the total benefit minus the total cost, which includes the minimum travel costs. It is this convention that makes the solution tractable.

Although it may seem overly simple at first, anecdotally, most people do most of their shopping within an area that they know well and have a high degree of knowledge about the travel costs of different routes between stores. Even people who move to a new area will identify the quickest routes after some period of learning. It is possible that this learning process would be path dependent—that is, consumers may not explore all routes, particularly if they are satisficing and therefore their set of routes may depend on where they searched first. However, that is a question for a subsequent model that would combine some form of learning algorithm over the best routes with the genetic algorithm described here. Before adding this additional level of complexity, we solve the problem of deciding on both the basket of goods and the travel costs using the minimum travel cost as calculated by the Floyd-warshall algorithm. The utility of the chromosome is calculated as follows:

$$Utility = TotalBenefits - TotalCosts \tag{7}$$

We also apply the consumer needs as another influential factor on the consumer decision process. There are three possibilities 1) a consumer needs an exact number of each item (as when following a recipe), 2) a consumer needs a minimum number of each item (as when buying dry goods or other consumables with a long shelf-life), and 3) a consumer needs a maximum number of each item (as when buying goods that have a short shelf-life). Therefore, we have three vectors according to the strategy that we are going to choose:

$$Need_{exact} =< e_{11,}e_{12}, \dots, e_{1k1}, \dots, e_{i1,}e_{i2}, \dots, e_{iki}, \dots, e_{n1,}e_{n2}, \dots, e_{nkn} > \tag{8}$$

In which e_{iki} shows the exact number of items ki from store i that a consumer needs. Similarly, we define the minimum and maximum needs on each item as follows:

$$Need_{min} =< m_{11,}m_{12}, \dots, m_{1k1}, \dots, m_{i1,}m_{i2}, \dots, m_{iki}, \dots, m_{n1,}m_{n2}, \dots, m_{nkn} > \quad (9)$$

$$Need_{max} =< ma_{11,}ma_{12}, \dots, ma_{1k1}, \dots, ma_{i1,}ma_{i2}, \dots, ma_{iki}, \dots, ma_{n1,}ma_{n2}, \dots, ma_{nkn} > \quad (10)$$

Therefore, for strategy 1, we use the exact needs vector, for the 2nd strategy we use minimum needs vector and strategy 3 uses maximum needs vector. These vectors control the value for each item in the chromosome.

The key functions of the genetic algorithm are crossover and mutation The main idea of proposed genetic algorithm comes from (Roozmand and Webster, 2014). To do the crossover, all chromosomes are sorted based on their utilities and are paired as parents. The crossover operation is applied on each store of paired chromosomes separately. Assuming that there are k1 items in store 1, the crossover point, which is a randomly chosen value between 1 and k1-1, is selected. Then the selected items of store 1 of two parent chromosomes are substituted based on the crossover point. This operation is applied for all other stores of these two chromosomes and finally two new chromosomes are generated. The crossover function is applied on all other paired chromosomes.

The mutation function is applied on all chromosomes with probability p including parents and offspring. The mechanism for choosing mutation point is like a crossover point and is selected separately for each store. Assuming that the mutation point refers to $item_{ij}$ and the number of $item_{ij}$ at the selected point is v_{ij}. The changes of the $item_{ij}$ at the selected point is randomly chosen in the ranges $[0, v_{ij}]$ for reducing the number of $item_{ij}$ and [0, Avaibale_Budget / p_{ij}] for increasing the number of $item_{ij}$.The equation Avaibale_Budget / p_{ij} guarantees that the cost of item does not exceed the budget. A random binary variable is used to decide to reduce or increase the number of items in mutation function.

After applying the crossover and mutation functions on chromosomes, the chromosomes with highest utility are selected and substituted with initial population. This process is repeated until a satisficing chromosome is found or the agent reaches the limitation for the total number of iterations.

2.1. Example

Let’s consider an example. Assume that we have 3 stores with the following items that are connected to each other as shown in Figure 3. As in our first example, the large dot represents the agent’s start-point.

$S_1 = \{i_{11}, i_{12}, i_{13}\}$ $S_2 = \{i_{21}, i_{22}\}$ $S_3 = \{i_{31}, i_{32}\}$

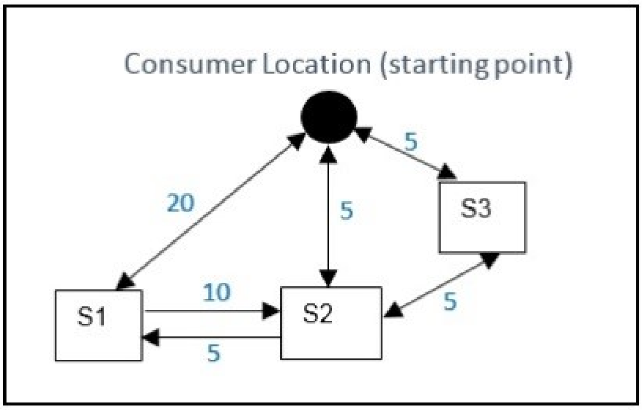

Figure 3. Market example for showing how the algorithm works

Lines with arrows represent one-way routes. Lines with two arrows show bidirectional route. The quality and price of items are as follows:

Table 1. Items used in three stores

Stores	Items	Quality	Price
S_1	i_{11}	10	5
	i_{12}	20	15
	i_{13}	30	25
S_2	i_{21}	15	15
	i_{22}	30	20
S_3	i_{31}	30	25
	i_{32}	50	40

First, let's consider what would happen without travel costs. Assume that we have the following chromosome CH:

Table 2. Structure of a chromosome

CH	1	2	3	4	5	6	7		8	9	10
Items	i11	i12	i13	i21	i22	i31	i32	Store name	S1	S2	S3
Number of item ij	5	2	4	0	0	7	1		1	0	1
	Store 1			Store 2		Store 3			Selected Stores		

This chromosome shows that, in this basket of goods there are 5 of item1, 2 of item2, and 4 of item3 from store 1, there are no items from store 2, and there are 7 of item1 and one of item2 from store 3. The last three cells show which stores would be visited if the agent selects this basket. Number 1 below the name of a store in selected store section indicates that store would be selected and 0 the store would not be selected. Here, stores 1 and 3 would be selected. Total benefits without travel costs for this chromosome is calculated as follows:

 DOI: 10.22067/JSTINP.2024.83302.1064

$$\text{Total Benefits} = [5*10+2*20+4*30] + [0*15+0*30] + [7*30+1*50] = 470 \tag{11}$$

To calculate the cost of this chromosome we need to add the distance cost to the cost of items. In this example, only stores 1 and 3 would be chosen. Therefore, there are three possible routes for the consumer. The first route is to move to store 1, then go to store 3 and lastly return to the starting location (note that this method could also accommodate an end point that is different from the starting location). Considering the place 0 (zero) for the consumer agent, it needs to follow the route [0, 1, 3, 0] to buy the items contained in the chromosome. Alternatively, the consumer can choose the route [0, 3, 1, 0]. Each of the above routes have different costs. In this example, the agent cannot move from location 1 to 3 directly (or vice versa), but rather must pass through location 2 or location 0. Therefore, we use Floyd-Warshall (Neapolitan and Naimipour, 2004) algorithm to find the shortest path between any two nodes in this weighted graph. This then gives us the value of MC for this specific chromosome.

In this example, we have two paths as follows:

$$MC_1 = Floyd[0,1]+\text{Floyd}[1,3]+\text{Floyd}[3,0] = 10+15+5 = 30 \tag{12}$$

Floyd [0,1] shows the shortest path from consumer location to store 1. There are three ways to get there including routes [0,1], [0,2,1] and [0,3,2,1]. The first route costs 20, the second 10, and the third route 15. Therefore, Floyd algorithm chooses the second route as the shortest path from the consumer location to store 1. Although, consumer passes store 2 to get store 1, it does not buy anything from store 2. The route [0, 3, 1, 0] is another possibility. Therefore, we have:

$$MC_2 = Floyd[0,3]+\text{Floyd}[3,1]+\text{Floyd}[1,0]=5+10+10 = 25 \tag{13}$$

Finally the MC is calculated as the minimum of MC1 and MC2, min(30, 25) = 25. MC should be added to the total price as part of cost of purchasing this basket of goods. In this example the cost is calculated as follows:

$$\text{Total Cost} = [[5*5+2*15+4*25] + [0*15+0*20] + [7*25+1*40]] + [MC=25] = 395 \tag{14}$$

Therefore, the total cost is 395 for this chromosome. If the Cost≤budget then this chromosome is feasible and acceptable by consumer. It means that consumer can afford this basket of items, including costs of travel among the stores.

We can then calculate the total utility of the chromosome by subtracting the costs from the benefits:

$$\text{Utility} = 470\text{-}395 = 75 \quad (15)$$

This is the value that is assessed by the genetic algorithm as it searches for the best possible combination of goods from all three stores, given travel costs.

3. Results

This section is divided into two subsections. In the subsection 3-1 we use four scenarios to show how the model works, and in subsection 3-2 we utilize the utility and indifference curve based on budget constraint to test the model based on theories in economics.

3.1. Working mechanism

The first scenario is used to test the genetic algorithm to find the optimal or satisfactory basket of goods when all items have the same quality and price, but are located in different stores. Figure 4 shows how the stores are connected to each other.

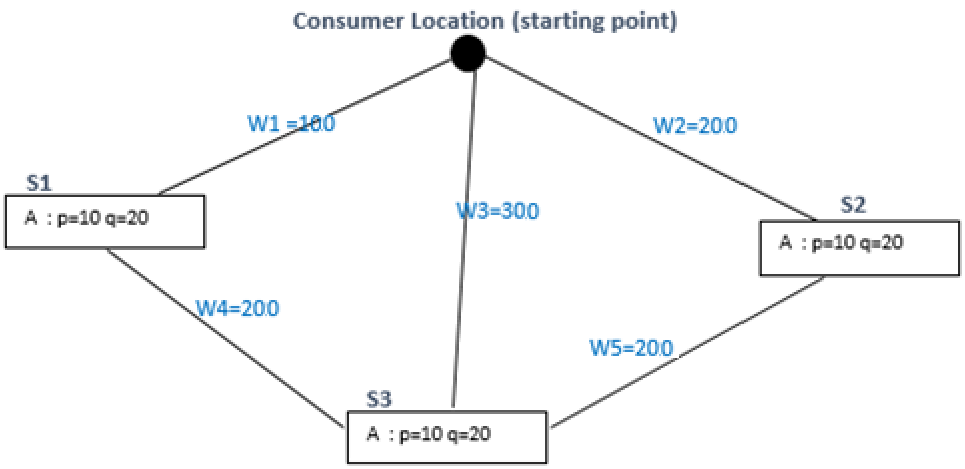

Figure 4. Market situation for scenario 1

As seen in Figure 4, all items in three stores have the same quality and price. The only difference is the distance between the consumer and the store. In fact, Figure 4 indicates that the best store for the consumer is store 1 (S1) which is the closest store to him/her. The consumer's budget is set to 3000 and the algorithm starts with 200 initial chromosomes and runs for 1000 times (Our experience shows that 1000 runs is a good number for finding optimum or near-optimum solution). We've provided the results based on three factors: fitness (utility) improvement, store selection, and route selection.

Figure 5 shows how the fitness or utility of chromosomes are improved to get the near-optimal solutions within a reasonable amount of time.

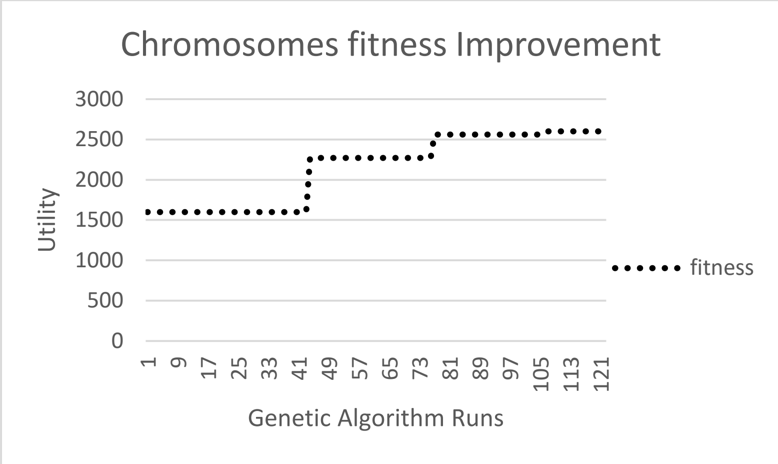

Figure 5. Fitness improvement with the increase of genetic algorithm runs

As we can see in above figure, the fitness of the selected chromosomes generally improves as the algorithm progresses. Each point shows the value of the best chromosomes' fitness. The fitness value is constant for many stages of algorithm's runs. This is because the algorithm does not produce a better solution on those runs. The best fitness value at run 1 is 1600 which belongs to a chromosome in the initial population. Finally, the algorithm ends with a fitness (Utility) of 2600 for the best chromosome, which according to this case, is the highest possible fitness. The algorithm finds the near-optimal solutions after 106 runs. Figure 6 shows how many items are selected from different stores on each run.

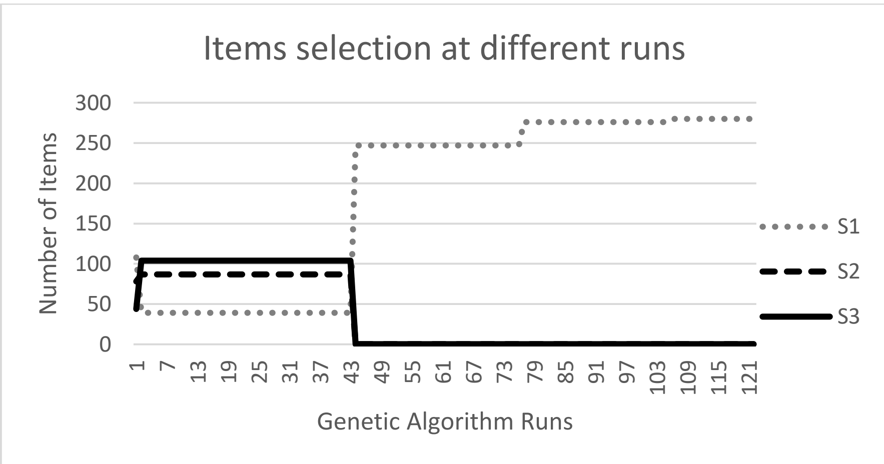

Figure 6. Item selection with the increase of genetic algorithm runs

Items are selected randomly from all three stores in the beginning. According to the effect of crossover and mutation operations on chromosomes, we see the increase in the selection of item A from store 1 (S1) which is 280 and consequently 2600 (280*20 – 280*10-200) fitness or utility (see figure 5) and 2800 for the cost of items. There is no item selection from stores 2 and 3. Considering the movement cost for the consumer, 100 for going to the store 1 and 100 for going back, we have 200 for the travel cost. At run 44, the consumer agent has 330 units of

 DOI: 10.22067/JSTINP.2024.83302.1064

budget unused. Finally, the algorithm selects 29 more units of item A from store 1 at run 77 and more 4 items at run 106. The more allow the algorithm to run, the more it turns from a satisficer strategy to a maximizer strategy. At this run, 280 items of good A have been selected and the result is 2600 for the utility and 3000 for the costs (2800 for the cost of items and 200 for the travel cost). The consumer agent has spent all of his/her available budget and reached the highest possible utility.

We should note here that the consumer appears to receive negative utility from the transaction because we assume price and utility are equal, ignoring the well-known concept of consumer surplus in economics. Consumer surplus refers to the utility that consumer receives above the amount paid for the good or service. In any market, there will be people who are willing and able to pay more than the equilibrium price. The difference between what they are willing to and able to pay what they actually pay is the consumer surplus (Turnovsky, Shalit and Schmitz, 1980). However, it does simplify the explanation somewhat and has no significant implications for the functioning of the model itself. In other words, the model would work just as well with consumer surplus included, it would just make it a bit more difficult to explain how the system operates.

We also examined the best chromosomes at runs in which there is a change in the fitness of the best chromosome (1, 2, 44, 77, and 106). Table 3 tracks the routes selected at each of these runs as well as the fitness, item costs, and route cost.

Table 3. Traveled stores, fitness, items' costs, and route cost

Run	**Route**	**S1's items**	**S2's items**	**S3's items**	**Fitness**	**Items' cost**	**Route cost**
1	0-S1-S2-S3-0	108	78	44	1600	2300	700
2	0-S1-S2-S3-0	39	87	104	1600	2300	700
44	0-S1-0	247	0	0	2270	2470	200
77	0-S1-0	276	0	0	2560	2760	200
106	0-S1-0	280	0	0	2600	2800	200

We also would like to see the behavior of the algorithm when high quality items are located at a further store. To do this, we changed the quality of item C in store 3. In other words, we substituted item C's attributes from <price = 10, quality = 20> to <price = 10, quality = 100>. The final result of the algorithm is shown in the table 4. It shows that our proposed genetic algorithm leads to store 3 for selecting item C, as it has the highest quality.

Table 4. Maximizing the basket of items by choosing only store 3

Route	**S1's items**	**S2's items**	**S3's items**	**Fitness**	**Items' cost**	**Route cost**
0-S3-0	0	0	240	21000	2400	600

Although store 3 is the furthest store from consumer location, it has the highest quality item (item C), so the consumer agent prefers to buy his/her items from this store in order to maximize its utility (fitness = 8400). In other words, the cost of traveling to the most distant store is outweighed by the improved utility from access to a higher quality item.

Scenario 3: In this scenario we altered scenario 1 by adding two more items to each store with the quality 100 and price 10. If this was the only change, the outcome would not be different. However, we also changed the consumer agent's needs. In scenario 1, there was no limitation on the number of items selected by the consumer agent, so it was able to maximize its utility based on any combination of items. In this scenario, we assume that consumer agent looks for specific amounts of item A and B and has no limitation on item C.

Needs = < A = 55, B=38, C = -1>

The above vector indicates that consumer agent needs 55 units of items A, 38 units of items B, and there is no limitation on item C. Therefore, the consumer agent should exactly buy 55 and 38 items of A and B respectively. The following graph shows that all items A, B, and C are available in all three stores with the same quality and price. Therefore, what matters in this scenario are the distance to each store and the consumer's needs.

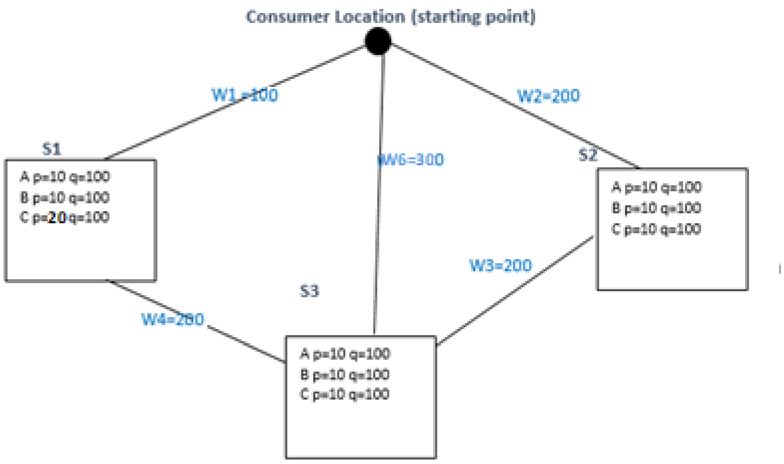

Figure 7. Market structure for scenario 3

The following Figure 8 shows how costs for route and items change over time as the algorithm progresses.

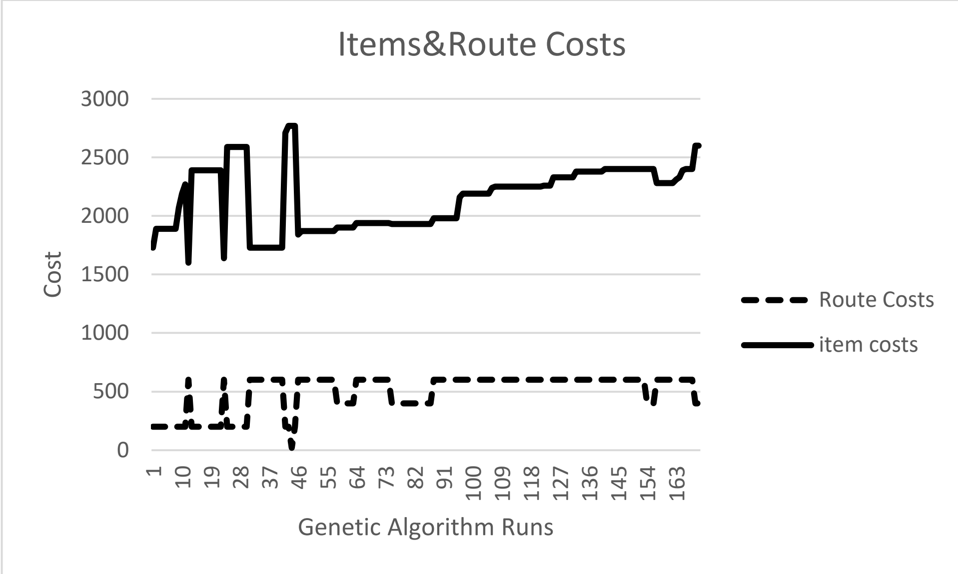

Figure 8. Items cost vs. route cost

As we see in Figure 8, the route cost decreases at the points where the algorithm finds a chromosome with higher fitness. As we described earlier, the chromosome chosen at each run is the best chromosome at that run. There might be some other chromosomes with high fitness that simply are ignored since a chromosome with better fitness exists. The following figure confirms improvement in chromosomes' fitness.

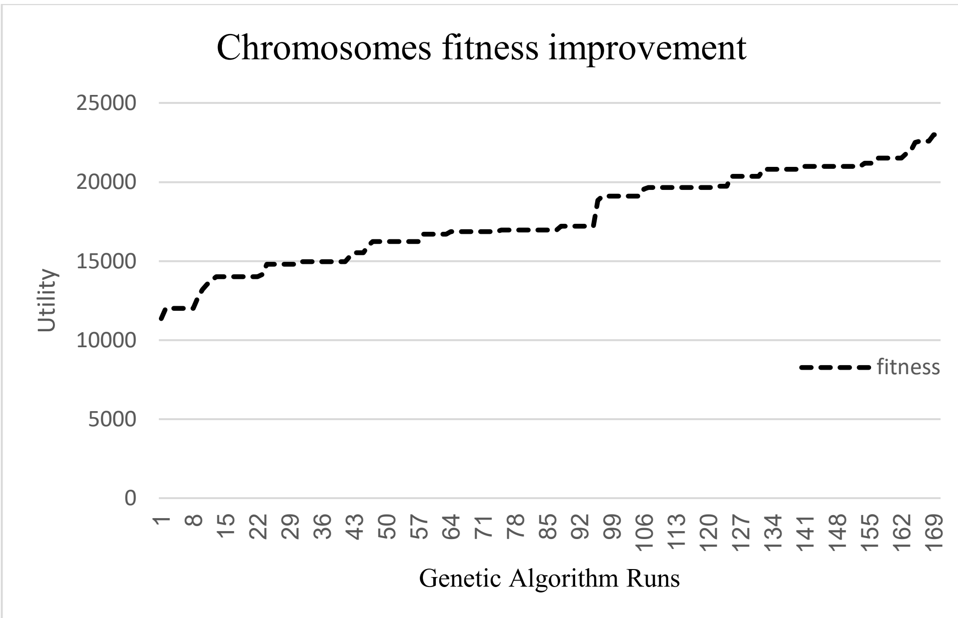

Figure 9. Fitness improvement with the increase of genetic algorithm runs

We see two improvements in the chromosomes. In this scenario, at run 169 the near-optimal solutions is found. This is 63 (169-106) runs longer than it took to find the best chromosome in scenario 1, which makes sense given the changes in scenario 3. In the first scenario, chromosomes can take any number of items, however, in this scenario many chromosomes are disqualified as they produce more than the limitations for items A and B. This consequently reduces the chance for higher fitness chromosomes. At the same time, the total number of chromosomes has increased substantially because now each store can have any combination of the three items instead of just one item. As shown in Figure 10, the result is that the agent quickly learns to purchase large amounts of C from Store 1, while fulfilling its more restricted needs for A and B from different stores, gradually learning which combination of stores provides the highest utility given travel costs.

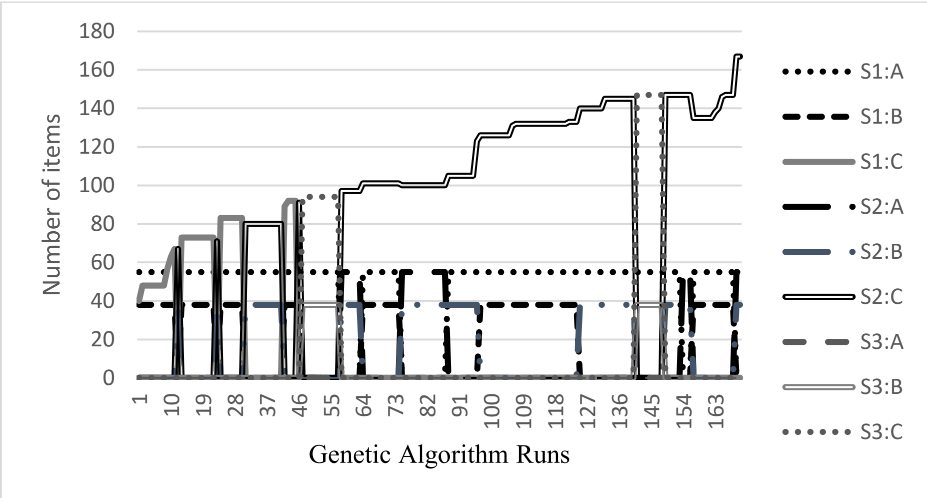

Figure 10. Items selection with the increase of genetic algorithm runs

Also, table 5 shows how the route is modified at each run where there is a change in chromosome fitness. The algorithm exactly chooses 55 units of item A, 38 units of item B, and 167 units of C from store 2 which is the closest store to the consumer. The reason for choosing store 1 and 3 is that all prices and qualities of all goods are the same in all stores. Therefore, it is quite reasonable to choose store 1 for purchasing all items as it is the closest one. Thus, this test scenario shows that algorithm is also able to consider the consumers' limitations for specific number of items.

Table 5. Results for when consumers have limitations on exact number of items

Run	Route	Store 1			Store 2			Store 3			Fitness	Items cost	Route cost
		A	B	C	A	B	C	A	B	C			
1	0-S1 -0	55	38	40	0	0	0	0	0	0	11370	1730	200
42	0-S1-0	55	38	89	0	0	0	0	0	0	15290	2710	200
141	0-S1-S3-0	55	0	0	0	0	0	0	38	147	21600	2400	600
166	0-S1-0-S2-0	55	38	0	0	0	147	0	0	0	22600	2400	600
169	0-S2-0	0	0	0	55	38	167	0	0	0	23000	2600	400

Scenario 4: In this scenario we would like to test strategies 3 and 4. Therefore, we added item D = <price = 10, quality = 150> to store 3 in the above scenario. Also, we define the consumer needs for both strategies as follows:

Strategy 3: $Needs_{min}$ = <A = -1, B = -1, C= -1, D_{min} = 3>

Strategy 4: $Needs_{max}$ = <A = -1, B = -1, C= -1, D_{max} = 3>

These two needs vectors indicate that there is no limitation on items A, B, and C. However, in strategy 3 the consumer needs at least 3 units of item D. It may choose to buy more than 3 units of item D but cannot choose to by fewer. Based on this strategy, the algorithm finds the following combination of items.

Table 6.Results for when consumers have limitations on minimum number of items

Route	**Store 1**			**Store 2**			**Store 3**				**Fitness**	**Items cost**	**Route cost**
	A	B	C	A	B	C	A	B	C	D			
0-S3-0	0	0	0	0	0	0	0	0	0	240	33000	2400	600

Although it is furthest from the consumer's start location (cost = 600), the consumer must visit store 3 since it is the only store which sells item D. It will also buy all other items from that store since there is no difference in the quality and price of A, B, and C in any of the stores. The genetic algorithm has chosen 240 units of item D. There is no contradiction with the problem definition since consumer has bought at least 3 units. In this execution of the algorithm, it has selected 0 of A, 0 of B, 0 of C, and 240 of D, but it could have selected any other combination of items with D >=3 and still achieved optimum fitness.

We see a different behavior of our genetic algorithm in the maximum strategy (#4). The consumer does not have purchase any of item D so it also does not have to pay the high travel cost to visit store 3. If consumer happens to buy item D, it cannot buy more than 3 units but given the location of the item, it is not likely that the algorithm will select a chromosome that contains item D. Table 7 shows the result with this strategy:

Table 7. Results for when consumers have limitations on maximum number of items

Route	**Store 1**			**Store 2**			**Store 3**				**Fitness**	**Items cost**	**Route cost**
	A	B	C	A	B	C	A	B	C	D			
0-S1-0	177	18	85	0	0	0	0	0	0	0	25000	2800	200

As expected, the consumer only visits store 1 and buys only items A, B, and C. As we compare results in table 7 and table6, consumer gains 8000 units of utility which is more than the 25000 units that result from strategy 3.

3.2. Aggregated results for utility and indifference curves

Since it is hard to validate the model based on real data, we instead rely on well-established economic theories. In the next two scenarios, we show how our model produces believable utility and indifference curves based on budget constraint for a set of consumers (100 agents). This includes examining differences in model behavior depending on whether or not we assume agents are maximizing or satisficing.

Before moving on to this analysis, we transition to a generalized version of the model used in the examples above. For this, we use a simple complete graph including four nodes: consumer location, store 1, store 2, and store 3. It is possible to use more complicated graphs but a simpler set up makes it easier to interpret the behavior of the agents and the model.

Figure 11 shows the stores with different distances. All nodes are connected to each other with edges d1 to d6 as shown in this figure. Each node indicates a store, and each edge shows the distance between two stores. This generalized model will also allow us to complete sensitivity analysis as described in later sections.

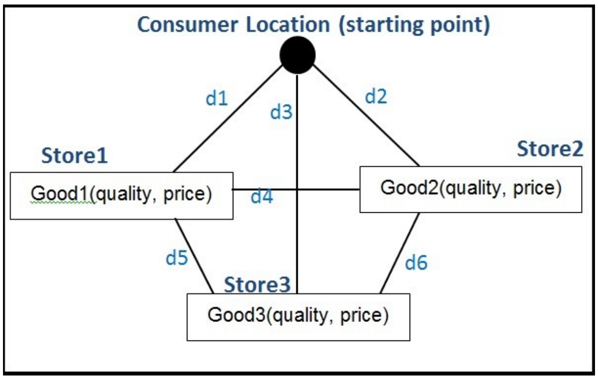

Figure 11. Market structure for validation

3.2.1. Utility

In this section, we test the effects of income (budget constraint), item quality, and price on aggregate utility. First, we consider the effects of price changes and budget constraints, holding item quality and distances constant. Our base values for item attributes and distances can be view in Table 8. We start with a skewed quality structure similar to scenario 2 above, where one store has an item of very high quality but low price and the other two stores have two lower-quality items at the same price. Note that the travel costs in this model are lower, reflecting relative proximity to stores (e.g., living in an urban or suburban environment) and allowing us to focus more on other factors in the decision process.

 DOI: 10.22067/JSTINP.2024.83302.1064

Table 8. Results for when one item's utility dominates other items and there is no substitution

Stores	Quality	Price	Distances					
			d1	d2	d3	d4	d5	d6
Store1 (Good1)	100	10	5	10	15	10	5	5
Store2 (Good2)	40	10						
Store3 (Good3)	40	10						

We generate 100 agents, each with budget of 1000 units. Because the distances are similar and the net benefit of good1 is so high, all agents decide to spend all of their budget on good1 and therefore only shop at store1. Next, we run the model multiple times, increasing the price for good1 5 units at each run; that is 15, 20, 25, and 30, respectively. Then, we run the model again with the same price variations but with a lower budget constraint of 500 units per agent. Figure 12 shows the resulting changes in aggregate utility.

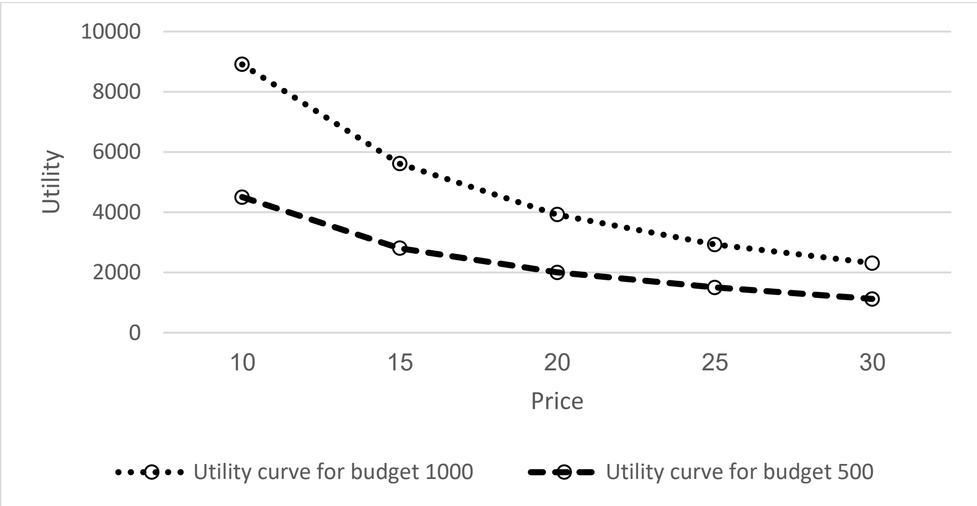

Figure 12. Obtained utility for budget 1000 and 500

The outcome is not surprising since there are no substitutes for good 1. As the price increases, agents can buy less of good1 but they do not switch to other goods because the net benefit for good1 is still higher than for goods 2 and 3. When we lower the budget constrain, the utility curve shifts down and flattens somewhat, demonstrating the expected effect of lower incomes on consumer purchasing decisions.

The results above assume that consumers are maximisers, but most consumers are satisficers. Instead of spending considerable time and effort seeking out a basket of goods with maximum utility, they select any affordable basket of items with utility larger than their satisfaction threshold (Roozmand and Webster, 2014). In this test we aim to show that how satisficing behavior can change the purchasing choices of consumers when goods vary by price, quality, and costs of travel. The following goods were generated for this test.

Table 9. Items attributes for testing satisficing behavior

Stores	Quality	Price	Distances					
			d1	d2	d3	d4	d5	d6
Store1 (Good1)	11	10	5	10	15	10	5	5
Store2 (Good2)	12	10						
Store3 (Good3)	11	15						

Given that travel costs are negligible, store 2/good 2 clearly can provide consumer agents with the highest utility when they spend their entire 500-unit budget. To investigate the effects of satisficing, we vary the satisfaction threshold by specifying it as a percentage of the maximum possible utility available. As shown in Table 10, we tested three satisfaction thresholds (82%, 46%, and 30% of maximum respectively.

Table 10. Results for testing satisficing behavior

Run time	Visited stores	Basket	Distance cost	Utility	Satisficing threshold	Budget
Run 1	0- 2 - 0	Good2=48	20	76	Maximizer	**500**
Run 2	0-1-2-0	Good1=6 Good2=41 Good3=0	25	63	82% of Max Utility	**500**
Run 3	0-1-2-3-0	Good1=12 Good2=32 Good3=2	25	35	46% of Max Utility	**500**
Run 4	0-1-2-3-0	Good1=4 Good2=34 Good3=6	25	23	30% of Max Utility	**500**

In the maximizing scenario (Run 1), the consumer selects the best basket of items including good 2 from store 2, and it gets a total utility of 76, which is the highest possible utility. Store 2 is farther away than store 1, but the consumer prefers to purchase from store 2 since the higher utility cancels out the higher travel cost. If we instead allow the consumer to satisfice at 82% of highest utility, it will select the first basket of goods that it finds which has a total utility of 63 or higher. In this case, the agent chooses both stores 1 and 2 for its purchases. It buys 6 of good 1 and 41 of good 2, reducing utility both by buying lower-utility items and by traveling more than necessary. As we decrease the satisfaction threshold to 46% and 30% in Run 3 and 4, respectively, we see that the consumer agent chooses even low quality items from other stores with lower quality. As expected, total utility decreases as the satisfaction threshold decreases, reflecting consumers' real-world preferences for convenience and minimizing transaction costs.

3.2.2. Indifference curve based on budget constraint

Consumers can get the same utility from different combinations of goods. Economists use indifference curves to represent these tradeoffs using two representative goods. In this section, we aim to check the purchasing behavior of consumer agent encountering two goods where different combinations of goods hold the same utility. To maintain tractability, we limit this analysis to only two goods: Good 1 with quality 10 and price 10, and Good 2 with quality 5 and price 5 (see Table 11). We do not test the goods in other stores, as it is difficult to interpret results with so many variables.

Table 11. Item's attributes for testing indifference curve

Stores	Quality	Price	Distances					
			d1	d2	d3	d4	d5	d6
Store1	Good1(10) Good2(5)	Good1(10) Good2(5)	5	10	15	10	5	5
Store2	-	-						
Store3	-	-						

In this test setting, purchasing two items of Good2 has the same utility as purchasing one item of Good1. Therefore, there are many combinations which have the same utility for the consumer agent. We have tested the model for 100 times with budgets 1000 and 500. The results are shown in Figure 14.

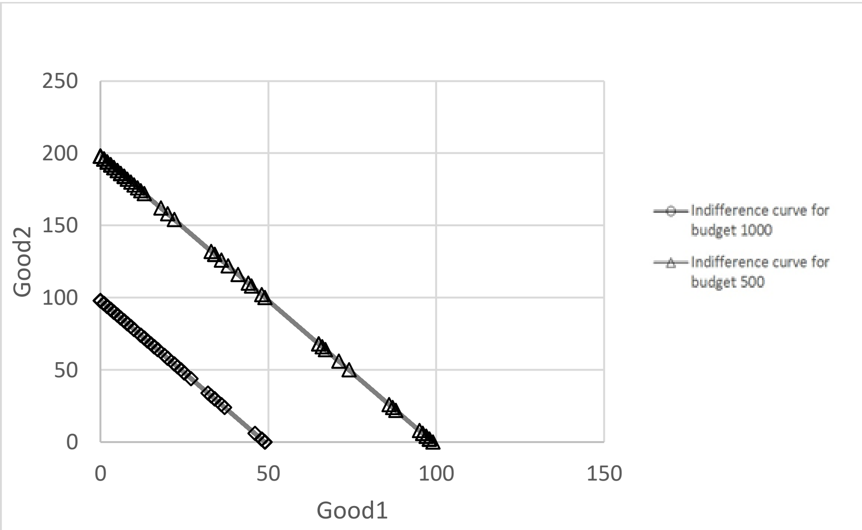

Figure 13. Indifference curve based on budget constraint for good1 and good2

With a budget constraint of 1000, and travel cost of 10 units, the consumer can use 990 units of its budget for purchasing goods. The indifference curve shows a clear 2-to-1 tradeoff between good 1 and good 2, with each combination maintaining the maximum utility at of 188 at this budget constraint. Lowering the budget constraint to 500 shifts the indifference curve to the left but does not alter the slope of the curve. Traditionally, indifference curves tend to be convex to the origin, reflecting diminishing marginal utility, but adding this dimension to our agent's behavior is a task for future work.

 DOI: 10.22067/JSTINP.2024.83302.1064

4. Conclusion and future work

In this paper, we propose a genetic algorithm equipped with the Floyd-warshall algorithm to simulate satisficing or maximizing consumer behavior. We applied a chromosome structure including items, utilities, and routes' costs to show a basket of items selected from different stores and calculate the total cost. The Floyd-Warshall algorithm helped find the shortest route among all possible routes from selected stores. The main novelty of this paper is that it proposes a genetic algorithm with a specific structure of chromosomes to solve this complex problem in economics. We simulated the model, showed how the model works in different scenarios, and presented that the model reproduces results compatible with theories in economics.

Although we believe that the model is well-designed and works reasonably well for this problem, it has two main weaknesses. First, the proposed algorithm is slow for a large and complex graph, including stores and their distances, so it is not suitable for modelling a big city with a large number of stores. Finding a more efficient approach would be a good next step in model development. Second, as the number of nodes and items increases, it becomes more and more difficult to assess the effects of changes in the model's parameters on model outcomes. More in-depth analysis is needed to fully understand how the model works, and then it should be possible to make the model more realistic by adding cultural knowledge, more diverse personal preferences, and other determinants of consumer behavior.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Baptista, M.L., Martinho, C.R., Lima, F., Santos, P.A. and Prendinger, H., 2014, March. An agent-based model of consumer behavior based on the BDI architecture and neoclassical theory. In *Developments in Business Simulation and Experiential Learning: Proceedings of the Annual ABSEL conference* (Vol. 41). Available at: https://journals.tdl.org/absel/index.php/absel/article/view/2109 (Accessed: 23 February 2021).

Bernabé-Moreno, J., Tejeda-Lorente, A., Porcel, C., Fujita, H. and Herrera-Viedma, E., 2015. CARESOME: A system to enrich marketing customers acquisition and retention campaigns using social media information. *Knowledge-Based Systems*, *80*, pp.163-179. https://doi.org/10.1016/j.knosys.2014.12.033.

Chen, R.C. and Jian, C.H., 2007. Solving unbounded knapsack problem using an adaptive genetic algorithm with elitism strategy. In *Frontiers of High Performance Computing and Networking ISPA 2007 Workshops: ISPA 2007 International Workshops SSDSN, UPWN, WISH, SGC, ParDMCom, HiPCoMB, and IST-AWSN Niagara Falls, Canada, August 28-September 1, 2007 Proceedings 5* (pp. 193-202). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-74767-3_21.

Dellaert, B.G., Arentze, T.A., Bierlaire, M., Borgers, A.W. and Timmermans, H.J., 1998. Investigating consumers' tendency to combine multiple shopping purposes and destinations. *Journal of marketing research*, *35*(2), pp.177-188. https://doi.org/10.1177/002224379803500204.

Fujita, H., Hakura, J. and Kurematu, M., 2009. Intelligent human interface based on mental cloning-based software. *Knowledge-Based Systems*, *22*(3), pp.216-234.https://doi.org/10.1016/j.knosys.2008.11.005.

Gilbert, N., 2019. *Agent-based models*. Sage Publications. Available at: http://methods.sagepub.com/book/agent-based-models (Accessed: 23 February 2021).

He, Z., Cheng, T.C.E., Dong, J. and Wang, S., 2013. Evolutionary location and pricing strategies in competitive hierarchical distribution systems: A spatial agent-based model. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *44*(7), pp.822-833. https://doi.org/10.1109/TSMC.2013.2290506.

Hofstede, G., Hofstede, G.J. and Minkov, M., 2010. *Cultures and organizations: software of the mind ; intercultural cooperation and its importance for survival*. Rev. and expanded 3. ed. New York: McGraw-Hill.

Hofstede, G.H. and Hofstede, G.J., 2005 *Cultures and organizations: software of the mind*. Rev. and expanded 2nd ed. New York: McGraw-Hill.

Jager, , M. and Jager, W., 1999. An integrated approach to simulating behavioural processes: A case study of the lock-in of consumption patterns. *Journal of Artificial Societies and Social Simulation*, *2*(2), pp.21-35. Available at: http://jasss.soc.surrey.ac.uk/2/2/2.html (Accessed: 23 February 2021).

Khosravifar, B., Bentahar, J., Gomrokchi, M. and Alam, R., 2012. CRM: An efficient trust and reputation model for agent computing. *Knowledge-Based Systems*, *30*, pp.1-16. https://doi.org/10.1016/j.knosys.2011.01.004.

McCrae, R. and Costa, P., McCrae, R.R. and Costa Jr, P.T., 1996. Towards a new generation of personality theories: theoretical contexts for the five-factor model. *The five-factor model of personality: theoretical perspectives*, pp.51-87.

McCrae, R.R. and Costa Jr, P.T., 1983. Joint factors in self-reports and ratings: Neuroticism, extraversion and openness to experience. *Personality and individual differences*, *4*(3), pp.245-255. https://doi.org/10.1016/0191-8869(83)90146-0.

McCrae, R.R. and Costa, P.T., 2003. *Personality in adulthood: a five-factor theory perspective*. 2nd ed. New York: Guilford Press.

Neapolitan, R.E. and Naimipour, K., 2004. *Foundations of algorithms using Java pseudocode*. Jones & Bartlett Learning.

Roozmand, O., Ghasem-Aghaee, N., Hofstede, G.J., Nematbakhsh, M.A., Baraani, A. and Verwaart, T., 2011. Agent-based modeling of consumer decision making process based on power distance and personality. *Knowledge-Based Systems*, *24*(7), pp.1075-1095.https://doi.org/10.1016/j.knosys.2011.05.001.

Roozmand, O., Nematbakhsh, M.A. and Baraani, A., 2007. An electronic marketplace based on reputation and learning. *Journal of theoretical and applied electronic commerce research*, *2*(1), pp.1-17. https://doi.org/10.3390/jtaer2010002.

Roozmand, O. and Webster, D.G., 2014. Consumer Choice and Aggregate Demand: An ABM Approach to Understanding the Impacts of Satisficing Behavior. *International Journal of Agent Technologies and Systems (IJATS)*, *6*(4), pp.1-18. Available at: https://doi.org/10.4018/ijats.2014100101.

Schwartz, B., Ward, A., Monterosso, J., Lyubomirsky, S., White, K. and Lehman, D.R., 2002. Maximizing versus satisficing: happiness is a matter of choice. *Journal of personality and social psychology*, *83*(5), p.1178. https://doi.org/10.1037/0022-3514.83.5.1178.

Schwartz, H., 2008. The role of aspirations and aspirations adaptation in explaining satisficing and bounded rationality. *The Journal of Socio-Economics*, *37*(3), pp.949-957. https://doi.org/10.1016/j.socec.2006.12.063.

Simon, H.A., 1955. A behavioral model of rational choice. *The quarterly journal of economics*, pp.99-118. https://doi.org/10.2307/1884852.

Sonderegger-Wakolbinger, L.M. and Stummer, C., 2015. An agent-based simulation of customer multi-channel choice behavior. *Central European Journal of Operations Research*, *23*, pp.459-477. https://doi.org/10.1007/s10100-015-0388-5.

Srisuwanna, C. and Charnsethikul, P., 2007. An exact algorithm for the unbounded knapsack problem with minimizing maximum processing time. *Journal of Computer Science*, *3*(3), pp.138-143. Available at: https://doi.org/10.3844/jcssp.2007.138.143.

Tsekeris, T. and Vogiatzoglou, K., 2011. Spatial agent-based modeling of household and firm location with endogenous transport costs. *NETNOMICS: Economic Research and Electronic Networking*, *12*, pp.77-98. https://doi.org/10.1007/s11066-011-9060-y.

Turnovsky, S.J., Shalit, H. and Schmitz, A., 1980. Consumer's surplus, price instability, and consumer welfare. *Econometrica: Journal of the Econometric Society*, pp.135-152. https://doi.org/10.2307/1912022.

Wang, S., Zoethout, K., Jager, W. and Dang, Y., 2021. The dynamical relation between individual needs and group performance: A simulation of the self-organising task allocation process. *Journal of Artificial Societies and Social Simulation*, *24*(4). https://doi.org/10.18564/jasss.4715.

Zhang, T. and Zhang, D., 2007. Agent-based simulation of consumer purchase decision-making and the decoy effect. *Journal of business research*, *60*(8), pp.912-922. https://doi.org/10.1016/j.jbusres.2007.02.006.