Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 6 (2023)

Genetic Algorithm and Evolutionary Architecture for Optimizing Stochastic Processes – A Supply Chain Use Case

Chakraborty A.[1], Ganguly S.[2], Das S.[3], Shaw A. K.[4], Keshri R. K.[5]

[1, 2, 3, 4, 5]*Computer Science and Engineering, JISASR, Kolkata, West Bengal, India*

Abstract: -Product placement is one of the most complicated process in supply chain and logistic management for eCommerce industries. It requires prioritizing which product to be placed and in which channel to optimize product delivery time and maximize profit. There can be no perfect decision in this matter but can be chosen from a wide array of possible solutions which in turn represents only a subset of a big and complex solution space. This important decision has impact downstream on product delivery, product return, customer satisfaction and profitability of the eCommerce platform. This can also have impact on churn and other business drivers. We have taken up this problem and tried to apply Genetic Algorithm (GA) to come up with a best fit solution. In this heuristic study we will also propose a reference public cloud-based deployment strategy to deploy the solution proposed in a public cloud environment. Its no new news that machine learning applications are now widespread and often an integral part of the primary software that's supporting one or more business processes. With the advent of new machine learning and deep learning paradigms, developing and applying models to solve business problems have become comparatively easier. This fueled with cloud-based platform support for big data has created an ecosystem of increasingly complex predictive solutions. Such solutions when deployed at a large scale have intrinsic problems associated with dynamic software systems, viz., complication in architecture at scale and all-around observability. Deployment of a system that has predictive algorithms at the end of numerous data pipelines coupled with end point management for consumption by downstream applications and visualization platforms suffers from problems such as scalability and security. My paper goal is to explore these problems in a growing ecosystem of predictive analytics project and propose microservices architecture for predictive models as using containerization and container orchestration.

Keywords*: Genetic Algorithm, Heuristics, Optimal Solution, Product Placement, Supply Chain, Logistics, Deployment, Public Cloud.*

1. Problem Description

eCommerce is one of the prime revenue generating industries at present. One of the perennial problems faced by this industry is ostensibly the problem of logistics. The prioritization strategy to transport goods from one place to another presents a multi-faceted problem. One dimension of this problem is to make a informed choice regarding the priority of which goods to transport so as to optimize the transportation cost and thereby the profit. It is to be noted that we are not considering the prioritization wherein other factors like selective membership to the eCommerce site, offers in delivery etc. are coming into play. In this paper we would like to present an implementation approach to solve this problem using Genetic Algorithm. We would begin by highlighting implementation details focusing on the technical and architectural details and the results of the study.

2. Approach of Study

As mentioned above in this paper we are going to approach to use Genetic Algorithm (GA) to help optimize the problem of goods logistic for a hypothetical eCommerce company. Our approach will consist of first building our solution based on GA algorithm step. This will include addressing how evolutionary algorithm and specially

GA will serve our propose, define the fitness function, define crossover strategy, study mutation combinations, identify population fit to be propagated, select individuals further down, selecting the best individuals. Once this is implemented, we will study the result and deliberate on the effectiveness of this method and hence identify the future scope of this work. To obtain the corresponding parameters, a random population of chromosomes is produced and decoded. The system model is then updated with these parameter values. Using a performance metric based on a cost function like that used in simulated annealing, a simulation is run, and results are obtained for each set of parameters within the population. [6] When all the cost values have been found, they are sorted into ascending order, along with the chromosomes that they correlate to. As before, the lowest cost values are picked as the best, and they are then exposed to reproduction, crossover, and mutation processes.

During the reproduction process, the best chromosomes are kept for the next population. New chromosomes are created by crossover and mutation processes to replace the other chromosomes. Only the elite chromosomes can advance to the next iteration of the reproduction process, which is known as 'rank-based selection.' As a result, this approach is a top-tier genetic algorithm. Crossover is a process in which some genes from one chromosome are swapped with genes from the same places on the other chromosome in the current generation (parent chromosomes). This technique produces two new chromosomes (offspring), and it is continued until there are enough offspring to replace the remaining 1% of the current population with the lowest cost values. Mutation is the process of selecting a particular number of genes in the current population at random and then making random changes to their values. This adds a random aspect to the GA search process, allowing it to consider more of the search space. As with the previous generation, once the chromosomes have been modified to generate the new population, it must be evaluated. To arrive at a final answer, the technique is repeated for a predetermined number of iterations (generations). It is worth noting that, in addition to optimization of nonlinear physically based models, evolutionary algorithms and simulated annealing can be used to model systems directly from empirical data and to linearize models. They are robust and provide optimization over huge amounts of space state. [7]

3. Business Problem

Supply chain management system is one of the principal ways of improving the financial plan of building products and adding benefits. It is an area where advancements and computerized interruption could profoundly change the conventional approach to carrying on with work. Then again, it is very complicated to change and needs tight ability.

Simultaneously, an extraordinary job in the stockpile chains is played by logistics - the administration of physical, educational, and human streams to optimize them and stay away from misuse of assets.

Beneath we will discuss the basic need of logistics workflow in supply chain system that incorporates numerous providers, transit areas, as well as source and destination. A more profound comprehension of supply chain management system can reveal insight into how to robotize the standard cycles and what goals we ought to put prior developing the legitimate software.

Assuming we organize all areas of logistics that should be created for the objective administration of production assets, we can single out the accompanying capabilities:

- *Defining a streamed supply chain process*

Congruity of work process is a fast step towards both lessening the costs and expanding by and large consumer loyalty. Generally, this is accomplished through legitimate preparation and the development of a shortcoming lenient plan of collaboration between the singular connections in the supply chain network.[6]

- *Reduction in labor resources*

On the opposite side of adaptation to non-critical failure is the end of repetitive components (delegates), the investment of which involves extra expenses. Subsequently, it is vital to find an equilibrium in which the reduction in labor assets does not involve constrained margin time during the execution of supply chains.

- *Building a network to find new target audience*

Expanding consumer loyalty is a fast move toward promoting your image through informal channels. An extra benefit is a reality that such publicizing (which is likewise perhaps of the best technique) comes totally free for you.

- *Cost Reduction*

By dispensing with numerous go-betweens joins in the supply chain network, you will decrease the net expense of an item or administration and thus there is an increment in its accessibility to the end-client.

Let us conclude the importance of logistic workflow in a supply chain management system. As we can see that logistic workflows and supply chain systems are two indivisible ideas that assist with decreasing the organization's general expenses for creating labor and products and work on the general impression of the degree of administration for your interest group. The decision of the right programming likewise assumes a huge part in streamlining strategic optimized logistics workflow. All things considered; automation today is at the top of the advancement for modern industrial enterprises.[6]

In this problem we have derived a dataset to represent the real world as much as possible. We have taken the most common items that are purchased by consumers through e-commerce platforms and measured the volume taken by these items to determine a matrix representing the product, product price and space taken by the same while in transit:

Table 1. Product Matrix

Product Price, Per Unit Cost and Volume Occupied		
Product	***Per Unit Cost ($)***	***Volume Occupied (m^3)***
Refrigerator Single Door	999.90	0.75
Mobile Phone	2199.12	0.00008
TV 55’	4346.99	0.399
TV 50’	3999.90	0.290
TV 42’	2999.90	0.200
Laptop A	2499.90	0.00349
Table/Stand Fan	199.90	0.495
Microwave Solo	308.66	0.042
Microwave Convection	429.90	0.0424
Microwave Grill	299.39	0.0543
Refrigerator De-frost	849.00	0.0312
Refrigerator Double door	1199.89	0.634
Assembled Desktop	1999.90	0.869
Notebook	3999.00	0.497

Let us assume that the maximum capacity of the delivery unit is $3m^3$. In our paper we will call this “Actual Capacity” (AC). On adding the volumes of these individual items, we get a desired capacity for transportation as $4.30m^3$. Hence the need for optimization based on which items should be prioritized to maximize profit.

Below are some facets of the implementation details:

Product Class – The product class serves as a template to list the individual items, their price and the volume occupied. The class constructors will consist of "name", "volume" and "price" of the products. Each of the products can be appended as a list and the list is assigned to the class.

Individual Class – Individual class represents the solutions to the problem. Each of these individuals will represent a unique attempt to solve the problem. A set of considered individuals make up a population of the solution space. Individuals contains chromosomes that represents the choices made by the individual. Below is an example of an individual containing 14 different chromosomes. "0" means that the particular product is not included in the logistic attempt and "1" means that the product is included. The attributes of this class would be the list containing the product prices, spaces, space limit (which in our case is $3m^3$) and generation. In the first iteration we have considered there to be only one generation. We have randomized to initialize the instances of the individual class to randomize the chromosome generation. The individual class has been constructed so that it can be generically used to contain the individual gene implementation for other problems as well.

Given below is a pseudocode for the individual class built:

I. Initialize the class parameters, in this case the parameters be space, price, space_limit, score_evaluation, used_space, generation

II. Initialize the chromosome with a blank array

III. For the length of spaces, randomize the chromosome and add to the array defined with '0' or '1' as per the random number enumeration

IV. Fitness Function – This will include quality measurements to determine the effectiveness of the solutions, firstly the randomly generated ones and secondly the ones that will be generated after iterations. It will decide whether the solution is an acceptable one and can be used for evolution. In this particular use case, it will be formulated as the total dollar value of the arrangement of the chromosomes and the volume occupied by them. For example, as per the above table (Table 1) if one of the chromosomes is "01001101001010" then the total dollar value is $10856.48, and the total volume is $1.76m^{3.}$ This may not be a good solution because we are only using $1.76m^3$ of the delivery space. In short, the solutions that have the highest prices and don't exceed the space limit can for this particular problem be taken as optimal solutions. Two variables are defined to contain the "Volume Score" and the "Value Score" for each of the chromosomes. An example iteration for chromosome "10101111010011" the "Value Score" is 17652 and "Volume Score" is 2.94. [7]

The pseudocode for the same in as follows:

I. Initialize the score and sum_spaces to 0

II. If the corresponding chromosome == '1', score and sum_spaces are iteratively added to themselves

III. If sum_spaces > self_space_limit, score =1 && Self_used_space = sum_space

Crossover – We have considered one-point crossover to focus on the process and implementation than on a in depth study of different crossover functions. It combines the genes of both the parents to form new set of chromosomes that can again try to solve the optimization problem further. The cutoff variable is randomized as a product of a random number and the length of the chromosome.

Mutation – Mutation introduces diversity in the population by randomizing changes in the genes of the chromosomes. We have applied it much less frequently than our crossover. The changes are made to these genes based on a probabilistic value. If the random value is greater than the length of the chromosome the chromosome is marked as '0' and on the other hand if its more than the initialized rate the chromosome is assigned a value of '1'. [8]

Initializing the population, measuring them against the fitness function, selecting the best individual is done. The best individual is selected via evaluating the individuals against the fitness function. Before we begin selecting the individuals from the next generation, we should select the best parents from the previous generation. Some individuals were discarded based on the measurements against the fitness functions.

The following are a sample set of results from a few iterations of the algorithm:

Iteration 1: As per the product listing above following is the placement array generated from the algorithm: ['1', '0', '0', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '1']

Iteration 2: ['0', '1', '1', '0', '0', '1', '0', '1', '1', '0', '1', '1', '0', '1']

Iteration 3 (with score evaluation): ['0', '1', '1', '0', '0', '1', '0', '1', '0', '1', '1', '0', '1', '1'] – 16502.97

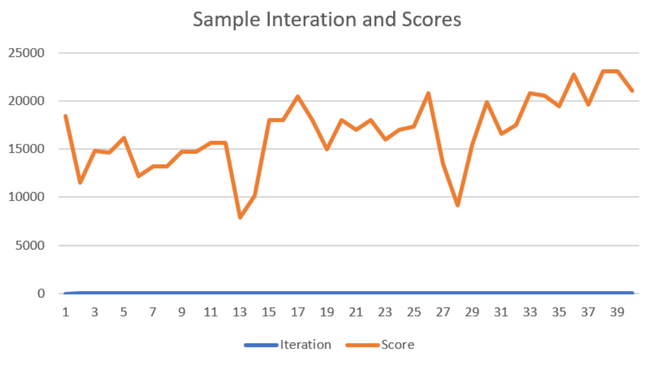
Iteration 4: With mutation = 0.01: Before: ['1', '0', '0', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '1']

After: ['1', '0', '0', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '1']

The program iteratively shows the various individuals processing the constraint based optimization problem:

Individuals	Score
1	18453.50
2	18403.25
3	15301.58
4	15272.85
5	14746.10
6	13931.50
7	13005.27

The above table is just an indication of how the score improves along with the iterations.



4. Explainability

Layer Wise Relevance Propagation

Heat map of pixel areas for an image under consideration for a classification problem gives a contour map of pixel importance, which is helpful in obtaining an explanation for the classification basis. This explanation in turn contributes to the overall black box explainability of the neural network used for the said classification problem. The class that is predicted with the highest probability displays a uniform trend of highlighting the areas of interest in the heat map in a “warmer” color than the surrounding “cooler” or insignificant areas. The same can also be used in relation to textual corpus while classifying a text by showing the words in the text that are primarily important for the basis of the classification. [8]

Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 6 (2023)

5. Working Principal of Layer wise Relevance Propagation

Layer wise Relevance Propagation or LRP is a pixel wise explanation technique for non-linear classifiers, that is classifiers which use neural networks as their basis for classification. In Bach, Sebastian, et al. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation." PloS one 10.7 (2015): e0130140, the technique was first proposed. It is a model specific model and designed mainly for neural networks. It presumes that classifier can be decomposed into several layers of computation. LRP applied on neural network gives us the relevance for an input for a deep neural network (like an image illustrated in the use case) that helps to explain the part of the image or area of the image that is responsible for our classification in a said category. From this we can conclude that LRP can be commonly used image classification tasks through the preparation of a heat map of the inputimage in deciding about the final class. The CNN model trained in images for a classification problem look at important features of an input image. The action of LRP starts after getting the result, that is it starts after getting the class of the image and then propagates in the reverse order along the layers visited in the forward pass and calculates the relevance of each of the neurons in each of the neural network layers. It provides positive and negative relevance score. A positive relevance score means that the pixel contributes to the classification while a negative score indicates that the pixel is against the final classification. This leads to a heat map. This method is also known as pixel device decomposition. [8]

Layer Wise Relevance Modelling

In a CNN the input image is converted to pixel values of 0s and 1s. On applying various filters the matrix representation of the image is transformed into multiple feature maps which forms the convolutional layers. To reduce the size of the convolutional layers we use the pooling layer which in turn is fed to the flattening layer. The "flattened" image representation is provided as an input to the fully connected ANN wherein it is classified. LRP starts from the predicted class and is propagated backward to obtain the relevance value of each neuron of each of the layer. Lets say that in predicting "Class A" the probability was 90%, that is, 1R4 = f(x) = 0.9 where, R is the relevance score for the 4th layer of the CNN for neuron 1. R<l,i> = Σz<i,j>z/(Σz<i^',z>) R<l+1,j>,

The relevance score for neuron i for Layer l is given by the above equation. Z<i,j> is the product of the input of the neuron i with the weight that goes from the neuron i to the next neuron j in the next layer. We can interpret this score as a relative activation of a specific neuron compared to all activations in that same layer. The final step is to multiply it with the relevance score of the neuron in the next layer.

6. Deployment

With cloud providers such as Azure, AWS, GCP coming into the picture there has been a revolutionary and significant change in the architecture of the applications and infrastructure design. The applications are designed keeping the advantages and flexibility of serverless in mind.

One key point is that despite the term Serverless, servers still runs the code but the infrastructure provisioning and management of the servers are abstracted from the developers. This helps them to concentrate on the business logic and deliver more value. Events can be described as a lightweight notification published by publishers which notifies about some state change or some condition. Azure functions are an advanced event driven based serverless design. Model being used by Azure Functions is of event driven where functions are being invoked by a trigger. Developers can write code in the language of their choice. Events driven architecture can be used when we want nearly real time processing of the events being generated and further can be processed by multiple sub system.

For instance, say when in an e-commerce site where order are placed/cancelled/updated the following flow is being followed:

New order/ Order Cancelled / Update Order events are being generated.

- These are being published to Service Bus Queue

- Azure Service bus triggered Function comes into the picture and processes them one by one pulling from the Service Bus Queue.
- After processing it Insert/Update/Delete data in the Cosmos Db.
- Whenever there is any Insert/Update/Delete in the Cosmos Db document(s), a change feed triggered Function gets triggered based on the document being modified.

So here we are leveraging the benefits of serverless design along with real time processing of the data, without being concerned about the infrastructure and the Scaling up or scaling out of the application based on the traffic over the application.[8]

7. Conclusion and Future Work

The above experiment shows clearly how Genetic Algorithm can be utilized for product placement in stochastic constrained supply chains. The individuals get better and attain a optimum score whose improvement precedes the cost of computation over time and thus converging to a optimized solution. Data centers are an integral part of modern computing infrastructure. They host racks which in turn hosts the servers that form the basis of virtualization. A data center design process involves the designing of infrastructure resources, architectural layouts, and physical system design. Cooling is an integral part of the design which keeps a data center functioning. The energy consumption of data center is potentially maximum for maintaining the cooling operations and optimization of the same will decrease energy requirements and move towards an economically more viable operation. We tend to explore this problem of energy consumption optimization via machine learning. Reinforcement learning is a special type of machine learning where the learning agent learns based on the outcomes of its actions on an environment of interest. It receives a positive or negative reward based on the state it is in and the action it takes. The goal of RL is to find a set of suitable actions that would maximize the total cumulative reward of the agent. The action is non-deterministic and the stochastic nature of it exposes the agent to temporal learning, the goal being reducing the temporal differences and obtain a convergence so that the agent learns to take desired actions. Q learning and its deep learning counterpart are popular in use cases where there is a lack labelled data sets and/or the strategy of optimization depends on the past, current, and future states of the agent. Here in this paper, we intend to study and implement Deep Q learning to train an agent that would help a server to stay in an optimal temperature by controlling the cooling and heating of the same. Once server temperature control is made optimal by the agent, we will quantify the energy savings to inspect the efficiency of the agent. We will experiment in a parameterized environment so that the environment itself can be made configurable. An edge deployment architecture will also be proposed to deploy the neural network in edge devices to be used in data centers. We also intend to include future scope of work to scale this neural network deployment to be used in a cloud-based platform to maximize availability.

References

[1] Genetic algorithms - https://link.springer.com/article/10.1007/s11042-020-10139-6
[2] A Study on Genetic Algorithm and its Applications - https://www.researchgate.net/publication/309770246_A_Study_on_Genetic_Algorithm_and_its_Applications
[3] Genetic algorithms: concepts and applications [in engineering design] - https://ieeexplore.ieee.org/document/538609
[4] Metaheuristics -- the metaphor exposed - https://www.researchgate.net/publication/237009138_Metaheuristics_--_the_metaphor_exposed
[5] Application of Genetic Algorithms in Machine learning - https://www.researchgate.net/publication/267369803_Application_of_Genetic_Algorithms_in_Machine_learning
[6] Chakraborty, Amit, Ankit Kumar Shaw, and Sucharita Samanta. "On a Reference Architecture to Build Deep-Q Learning-Based Intelligent IoT Edge Solutions." Convergence of Deep Learning In Cyber-IoT Systems and Security (2022): 123-146.

[7] Shaw, Ankit Kumar, et al. "Scalable IoT Solution using Cloud Services–An Automobile Industry Use Case." 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC). IEEE, 2020.

[8] Mohapatra, Debaniranjan, Amit Chakraborty, and Ankit Kumar Shaw. "Exploring novel techniques to detect aberration from metal surfaces in automobile industries." Proceedings of International Conference on Communication, Circuits, and Systems: IC3S 2020. Springer Singapore, 2021.