Warehouse Stock Prediction Using Krill Herd Algorithm

Somula Ramasubbareddy, Aditya Sai Srinivas T, Govinda K, Manivannan .S.S, Swetha .E

Abstract: ***E-commerce and online shopping have seen a surge in demand that surpasses offline shopping. To keep up with this demand and stay relevant in the market, companies need to provide best service in the minimum time. The conventional trend is to get the product to the nearest warehouse after it is ordered or stock the warehouses to their full capacity to cater to customer demands. Those methods are ineffective because if the product is called after it is ordered it will lead to a waiting time of a couple of days, also special transportation needs to be arranged to get a less number of parcels to the destination which raises costs and pollution. Stocking up of warehouses is also ineffective as you waste space that could have been used for other products but also the products that no one needs have to be sent back which doubles the transportation cost. What we propose is a method to estimate the demand and strategically get and order products on trend analysis to save time, money and the environment. The algorithm we try to reach that end is Krill Herd Algorithm and Particle Swarm Optimisation which are a part of the genetic algorithms.***

Keywords: Genetic Algorithms, Krill Herd Algorithm, Particle Swarm, E-commerce, Online Shopping, Warehouse Optimisation, Optimisation, Inventory Management.

I. INTRODUCTION

Optimising resources has been a challenge over decades to every business. It becomes incredibly important for the online-segment to do this as the volume of products are massive and in order to be more competitive they need to be reducing costs elsewhere. The size of the operations also makes them vulnerable to huge overhead costs if they misallocate resources or deprioritise inventory management.

Krill herd algorithm is based on the herding nature of krill individuals that are found in nature. The two main properties of this algorithm is using mixture of random search or exploration and local search or exploitation. The balancing of the two factors is absolutely essential to get best results from the algorithm.

The Particle Swarm Algorithm is used as we do not need to update the particle's positon.

The self-updating algorithm helps in providing the best results over a huge number of iterations and hence provide fine-tuned results.

The local warehouse for any e-commerce organization is taken into consideration as the inventory stock in those warehouses are the key to finding the difference between the supply and sales of a particular product.

II. LITERATURE REVIEW:

M. Ali Ilgin and SemraTunali (2006) in their paper on Joint optimization of spare parts inventory and maintenance policies using genetic algorithms [1] created a simulation model of the manufacturing system and integrated genetic algorithms to optimise the model. They concluded that there is often more than one objective while attempting to optimise a maintenance management system. Their ground breaking result concluded that the company could save 53% of the annual cost while also increasing the throughput when it used the genetic algorithm approach to the control variable method.

Ata Allah Taleizadeha ,SeyedTaghiAkhavanNiaki b, Mir-BahadorAryanezhada in their paper on A hybrid method of Pareto, TOPSIS and genetic algorithm to optimize multi-product multi-constraint inventory control systems with random fuzzy replenishments [2] use a random fuzzy replenishment multi product inventory model. They use two mathematical models for two cases fuzzy and deterministic programming to solve multi-objective, integer-nonlinear problems. They give us a test case to show the validity and efficiency of the proposed model.

Masao Yokoyama in his paper on Integrated Optimization of inventory-distribution systems by random local search and genetic algorithm [3] provides an optimised model based on order-up-to-R policy and transportation problem. They try and determine the target inventory levels and transportation quantities in order to minimise the total cost. The model first uses linear programming and simulation to find the initial costs then moves to local random search to determine the optimum target inventory levels.

P.Radhakrishnanet. al. in their paper Integrated Optimization of inventory-distribution systems by random local search and genetic algorithm [4] used MATLAB 7.4 to integrate genetic algorithms to find out the most probable excess stock level and shortage level needed for inventory optimisation

Revised Manuscript Received on March 26, 2019

Somula Ramasubbareddy, Information TechnologyVNRVJIET, JNTUH, Hyderabad, India

Aditya Sai Srinivas T, School of Computer Science and engineering, VIT University, Vellore, Tamilnadu, India.

Govinda K, School of Computer Science and engineering,VIT University, Vellore, Tamilnadu, India.

Manivannan .S.S, School of Information Technology,VIT University, Vellore, Tamilnadu, India.

Swetha .E, Computer Science and engineering,VIT University, Vellore, Tamilnadu, India.

in the supply chain to result in minimum cost. Their research concluded with reducing the costs.

Scott et al. proposed a new technique to gain benefits of using supply chain management for the purchase of direct goods. [5]. RFQs have been constructed on basis of the projections for future prices and demand and the quotes that optimize the level of inventory each day besides minimizing the cost have been accepted. The problem was represented as a Markov decision process (MDP) that allows for the calculation of the utility of actions to be based on the utilities of substantial future states. The optimal quote requests and accepts at each state in the MDP were determined with the aid of Dynamic programming.

Multi-matrix real-coded Generic Algorithm (MRGA) is used by Pupong et al. [6],that is an optimisation tool which aids in reduction of total costs associated with in supply chain transport. Procedures such as the chromosome initialization procedure, crossover and mutation operations that ensure feasible solutions have been incorporated. Evaluation of the algorithm was done with the help of three sizes of datasets that are mostly faced by large manufacturing companies.

Sherbrooke [7] is the first application of mathematical programming in spare parts inventory management problem. Various approaches have been used to solve the spare part problem such as mathematical programming and simulation. Mathematical programming concerns the development of mathematical models based on linear programming, dynamic programming, goal programming, etc. Multi-echelon technique for recoverable item control model.

Sarmiento et. al. in Stability analysis of the supply chain by using neural networks and genetic algorithms [8] propose a methodology that will aid in detecting changes in the environment and will eliminate possible oscillatory behaviours. A supply chain configuration supplied by an input vector is taken and entered into a Behaviour Monitor Module which uses the Neural Network and pattern recognition capabilities which will help improve the forecasts and give more accurate predictions.

A useful industry case applying Genetic Algorithms has been put forward by Wang, K. and Y. Wang. [9] The case has made use of Genetic Algorithms for the optimization of the total cost of a multiple sourcing supply chain system. The system has been explained in detail by a multiple sourcing model with stochastic demand. A mathematical model has been implemented to portray the stochastic inventory with the many to many demand and transportation parameters as well as price uncertainty factors.

Pardoe, D. and P. Stone in their paper “An Autonomous Agent for Supply Chain Management “used TacTex-06 that comprises of optimizing, predictive and adaptive components. The job of the TacTex-06 is to use a multitude of algorithms including Genetic Algorithms to make predictions about various things such as the economy. It was also used to determine the most beneficial strategy to use in the future to maximise profit in the coming years.

III. PROPOSED METHODOLOGY:

We are following the Langragian model Krill Herding. The two criteria to be reached are maximum density (max range of products) and minimum distance from food (distance from customers). The following parameters will be taken into consideration when the algorithm performs its functions.
Results are based on the following metric:
Movement induced by other krill individuals. (Sales and stock of other products).
Foraging activity- Movement induced by food. (Sales and production change induced by the customer demand).
Random Diffusion (Random change in sales and production).
The parameters on which the results will be reached are:
Number of Runs, Number of Iterations, Number of Krills (No of Products), Number of Dimensions, Maximum Induced speed, Foraging Speed (Changes shown by customers), Maximum Diffusion Speed (Entropy), Random variable, Small positive number.
Particle Swarm Optimisation uses similar techniques, generating random population and putting them through fitness function. The objective is to find the best position and least error.
But unlike Krill Herd algorithm the particles update themselves through vectors, specifically velocity vectors.
The dataset that we will use will have the annual sales of various products and will also indicate the monthly same of those products. The algorithm will run over multiple iterations to predict costs of the following month.

IV. INITIALISATION:

The dataset is passed through the following operations to normalize

Demand average = (Demand + Demand (area))/2

Discount Factor=Demand average *((1+ (0.01*Discount (That day))))

Rating Factor = (Rating* Discount Factor)/2.5

Supply Difference = Supply-sales

Ratio = (Rating Factor +Supply Difference)/2

Define the population on which we want to apply our genetic algorithm (a dataset). To ensure the data set is in synchronisation with the algorithm we need to define a few parameters such as,

ND: The number of entries that the dataset contains (No of Days)

NK: Average Sales to supply ratio, this we will get by importing the relevant values from the file.

MI: The maximum number of iterations that we will perform, the more the iteration the closer we can approximate the results but we have to keep this a reasonable number for the program to be feasible on a system.

It is also at this stage where we decide the parameters that will influence our results, which we have chosen to be as follows:

No. of Runs

No. of Iterations

No. of Krills (No of Products)
No. of Dimensions
Maximum Induced speed
Foraging Speed (Changes shown by customers)
Maximum Diffusion Speed (Entropy)
Random variable
Small positive number

3.2Cost Function: Certain test functions, in the realm of mathematics better known as artificial landscapes are used to evaluate characteristics of optimization algorithms.

To calculate the convergence rate we have chosen Ackley Function.

$$f(x) = -a\exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1)$$

Recommended variable values are: a = 20(Population), b = 0.2(Small Number) and c = 2π.

Fitness Function

3.3.1 Krill Herd Algorithm

The next step is to define the fitness function, each iteration of the resulting program will lead to several values, and we need to ensure that they cross a particular threshold to be considered as legitimate results. In layman's terms fitness function can be defined as how close the intermediate result is to the aim we set to achieve.

Our fitness function is defined as follows:
For i=1:No. of Products
Xvector(i) = (random number between the upper and lower bound)
Kvector(i)=eval(Xvector(i))
Kib=K; (for best fitness value)
Xib=X; (for best customer demand reach)
Kgbest=min (k)
Xgbest= (return the position of Kgbest)

Which can be summed up as:

$$dXi/dt = Ni + Fi + \mathbf{Xi}$$

Where,
Ni: Motion induced on ith krill individual due to the other krill individuals.
Fi: Foraging motion.
Xi: Relative position of the krills

3.3.2 Particle Swarm Optimization:

The fitness function of this algorithm does not depend upon updating via any equation but rather by comparison of values and minimizing the distance.

3.4 Selection: This section will include several calculations including those of Foraging Motion, Motion Induced, Random Diffusion and Updating.

3.4.1 Foraging Motion: Foraging is defined as the searching for wild food resources and in our case it would go onto signify the motion that needs to be carried out to reach the food. We calculate this by these formulae:

Foraging Motion = Small Number*βi + Inertia Weight*Foraging Motion (old)

βi = Effect due to the presence of food + Effect due to the current krill's best fitness value recorded

3.4.2 Motion Induced: This parameter takes care of the total motion over all iterations that is induced in the Krill population.

$Ninew = Nmax\alpha i + \omega nNiold$

αi = ∑j=1-(Total number of neighbours)$\hat{K}ij\hat{X}ij$

$\hat{X}ij = Xj - Xi/\|Xj - Xi\|$ + Small positive number

$\hat{K}ij$ =Fitness value of the ith krill individual – Fitness value of the jth neighbour /$Kworst - Kbest$

Where,
Ki: Fitness value of the ith krill individual
Kj: Fitness value of the jth neighbour (j = 1, 2, …,NN)
X: Relative position of the krills

3.4.3 Random Diffusion: This parameter is included to take into account the changes that in the market that we cannot see, such as bonus sales, competitor's product launch etc. We define all such uncertainties by the random diffusion parameter and set a diffusion speed as factor of maximum diffusion that can take place.

We calculate random diffusion by the formula:
Random diffusion = Maximum diffusion speed $(1 - I/Imax)$ Random directional vector

3.4.4 Updating:After all the calculations we need to recalibrate the dataset for the new values that we calculated, the updating is based on the following formula:

dXi=delta_t*(N(i)+F(i)+D(i));
X(i)=X(i)+dXi

We repeat the entire thing till the final criteria is reached.

3.5 Crossover and Mutation: Iterating over the values will have crossover of certain of certain genes and some mutation as well. We need to ensure that the mutation are not so much that they provide us with completely unrelated results, hence we will check them with the fitness function and if any value(s) seem astray they will be discarded.

3.6 Termination: There can be three situations where such an algorithm terminates, which are:

There is no improvement in the population for over x iterations.
We have already predefined an absolute number of generation for our algorithm.
When our fitness function has reached a predefined value.

In our situation we have chosen to go with a predefined number of iterations bearing in mind the computational limitations of the system.

V. RESULTS AND DISCUSSIONS:

Fig.1 with change on the Y-axis and X-axis containing the number of iterations, the region(s) of interest are the ones where the two lines intersect. If there is a single point of intersection then we can conclude that 10^(change value) is the delta change that we need to take into account when ordering the product next time. If there are multiple points of intersection then we shall take the average value of 10^(change value) and then round that off to the nearest integer that will be the delta change value.

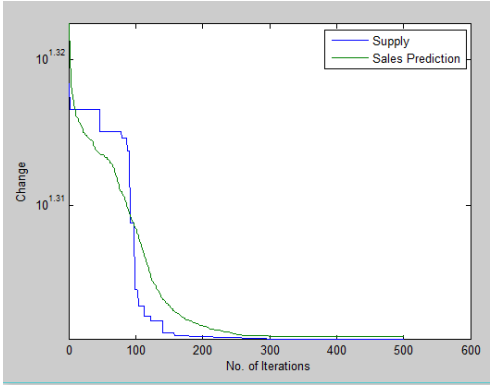

Fig.1. Sales Prediction based on no.of Iteration

VI. CONCLUSION

The change in the supply are calculated and thus can be applied. Both the algorithms give extremely close and comparable values which could be verified by comparison. Both the algorithms give the change that should be brought in the supply of the particular commodity.

REFERENCES:

1. M. Ali Ilgin ,SemraTunali , "Joint optimization of spare parts inventory and maintenance policies using genetic algorithms" , Int J AdvManufTechnol (2007) 34: 594–604; DOI 10.1007/s00170-006-0618-z.
2. Ata Allah Taleizadeha , SeyedTaghiAkhavanNiaki b, Mir-BahadorAryanezhada "A hybrid method of Pareto, TOPSIS and genetic algorithm to optimize multi-product multi-constraint inventory control systems with random fuzzy replenishments" Mathematical and Computer Modelling, Volume 49, Issues 5–6, March 2009, Pages 1044-1057.
3. Masao Yokoyama, " Integrated Optimization of inventory-distribution systems by random local search and genetic algorithm". Computers & Industrial Engineering, Volume 42, Issues 2–4, 11 April 2002, Pages 175-188.
4. P.Radhakrishnanet. al. "Inventory Optimization in Supply Chain Management using Genetic Algorithm", IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.1, January 2009
5. S. Buffett, N. Scott, "An Algorithm for Procurement in Supply Chain Management", AAMAS-04 Workshop on Trading Agent Design and Analysis, New York, 2004.
6. Pongcharoen, P., Khadwilard, A. and Klakankhai, A., "Multi-matrix real-coded Genetic Algorithm for minimizing total costs in logistics chain network", Proceedings of World Academy of Science, Engineering and Technology, vol. 26, pp .458-463, December 14th-16th, 2007.
7. Sherbrooke CC (1968) METRIC: a multi-echelon technique for recoverable item control. Oper Res 16:122–141
8. Sarmiento, A. Rabelo, L. Lakkoju, R. Moraga, R., "Stability analysis of the supply chain by using neural networks and genetic algorithms", Proceedings of the winter Simulation Conference, pp: 1968-1976, 2007.
9. Wang, K. and Y. Wang, 2008. Applying genetic algorithms to optimize the cost of multiple sourcing supply chain systems-An industry case study. Stud. Comput.Intell., 92: 355-372. DOI: 10.1007/978-3-540-76286-7_16
10. Pardoe, D. and P. Stone, 2007. An Autonomous Agent for Supply Chain Management. In: Handbooks in Information Systems Series: Business Computing, Adomavicius, G. and A. Gupta (Eds.). Elsevier. Amsterdam. http://www.cs.utexas.edu/~pstone/Papers/bib2html/b2hd-TacTex-Book07.html

AUTHORS PROFILE

Somula Ramasubbareddy, Information TechnologyVNRVJIET, JNTUH, Hyderabad, India.

Aditya Sai Srinivas T, School of Computer Science and engineering, VIT University, Vellore, Tamilnadu, India.

Govinda K, School of Computer Science and engineering,VIT University, Vellore, Tamilnadu, India.

Manivannan .S.S, School of Information Technology,VIT University, Vellore, Tamilnadu, India.

Swetha .E, Computer Science and engineering,VIT University, Vellore, Tamilnadu, India.