

## A Genetic Simulator for Airline Yield Management

Pardeep Kumar Mittal  
Assistant Professor  
Dept. of Computer Science  
& Applications  
K.U.Kurukshetra

Dr. Rakesh Kumar  
Professor  
Dept. of Computer Science  
& Applications  
K.U.Kurukshetra

Dr. P.K.Suri  
Dean (R&D)  
Chairman & Professor  
(CSE/IT/MCA)  
H.C.T.M., Kaithal

### Abstract

*Airline yield management now-a-days has become a very important research area. As more and more companies are coming in the field of airlines, it has become very difficult to sustain for a company without incorporating yield management. Therefore most of the companies are trying to incorporate the yield management in their system. The basic purpose of the yield management is to maximize revenue within the specified constraints. This paper is an attempt to design a basic decision-support tool for airline yield management using genetic algorithm. The genetic simulator has been designed using Matlab.*

### 1. Introduction

Yield management is the concept of identifying various strategies for optimizing yield (revenue) in various capacity-constrained services. The core concept of yield management is to provide the right service to the right customer at the right time for the right price.

A popular definition of yield management is given by Netessine as "Yield management is the process of understanding, anticipating and reacting to consumer behaviour in order to maximize revenue or profits." [1]

The big question is when yield management should be applied. Are there any conditions under which it can be applied? The answer lies in the following three basic conditions:

- There is a fixed amount of resources available for sale.
- The resources sold are perishable.
- Different customers are willing to pay a different price for using the same amount of resources.

Yield management can be applied in a number of industries, although it is more popular in airlines industry and hotel industry.

Although there are no specific guidelines for a yield management process, it may contain data collection,

segmentation, forecasting, optimization, and dynamic re-evaluation

In mathematics, computer science, or management science, optimization is the selection of a best element (with regard to some criteria) from some set of available alternatives. In the simplest case, an optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function. Generally, optimization includes finding "best available" values of some objective function given a defined domain, including a variety of different types of objective functions and different types of domains.

To solve various optimization problems, one may use algorithms that terminate in a finite number of steps, or iterative methods that converge to a solution (on some specified class of problems), or heuristics that may provide approximate solutions to some problems.

### 1. Literature Review

From a historical perspective, the interest in Yield (Revenue) Management practices started with the pioneering research of Rothstein [2] and Littlewood [3] on airline yield management. However, it was probably after the work of Belobaba ([4], [5] and [6]) and the American Airlines success [7] that the field really took off. The airline industry provided researchers with a concrete example of the tremendous impact that Revenue Management tools can have on the operations of a company (e.g. [7]). At this stage, however, much of the work was done on capacity management and overbooking with little discussion of dynamic pricing policies. In essence, prices (fares) in these original models were assumed to be fixed and managers were in charge of opening and closing different fare classes as demand evolved. During the 90's, the increasing interest in Revenue Management became evident in the different applications that were considered. Models became industry

specific (e.g. airlines, hotels, or retail stores) with a higher degree of complexity (e.g. multi-class and multiperiod stochastic formulations). Furthermore, it was in the last decade that pricing policies really became an active component of the Revenue Management literature (e.g.[8], [9], [10] and [11]).

Several authors report “success stories” on how effective Revenue Management has been done in practice ([12] and [13]). An example of successful Revenue Management system can be found in British Airways. Among the reported in 1992 reasons of company success were cost cutting and sophisticated Yield Management. The company attributed its success to being “a low-cost and high-revenue carrier” [12]. The next successful Revenue Management implementation example can be found within the Austrian Airlines company that has been one of the most consistently profitable airlines in Europe in the past decades. During the Gulf War that has caused profits of the most airlines to decline, the Austrian Airlines has experienced its “twenty-first consecutive year of profits” [13]. This has proven the company capability to perform effectively during extremely cyclical periods. The company owes its success to smart investment choice into a Revenue Management and decision-support computer system. This system has helped to monitor all historical data on the company flights as well as has made it possible to perform flights forecasts up to a one year period with high precision. Moreover, the new Revenue Management system has allowed the company management to look for future business opportunities by supplying it with decision-making tools to forecast future flights demand. In addition, the implementation of this system has enabled the company to keep her prices at stable level while other companies were trying to survive by lowering their flight prices [13]. Bitran and Caldentay in their paper examined the research and results of dynamic pricing policies and their relation to Revenue Management[14]. The survey is based on a generic Revenue Management problem in which a perishable and non-renewable set of resources satisfy stochastic price-sensitive demand processes over a finite period of time.

#### **Genetic algorithm in yield management**

In a system proposed by Aloysius George et.al. , in order to maximize the revenue of airline, an optimized flight booking and transportation terminal open/close decision system has been presented using Genetic Algorithm[15]. In this system, the particular booking terminal’s historical booking data is observed. Consequently, its frequency is generated with linguistic variable and deviation of booking is interpreted. Using the observed data and genetic algorithm, the terminal open/close decision system is optimized.

In an article presented by Srinivas and Shashi [16], the focus is on developing a decision-support tool to estimate the number of seats to each fare class. Genetic algorithm is

used as a technique to solve this problem. The decision support tool considers the effect of time-dependent demand, ticket cancellations and overbooking policies.

## **2. Problem Definition and Formulation**

In this problem an assumption regarding a flight operating between a specified origin and destination has been made. The reservation for the flight starts from the first date of expected reservation up to the date of departure. The period of reservation may be divided in a single or more than one time slices. Another assumption is to fix the fare of each class during each time slice and also assumed as known. Following notations has been assumed for this problem:

$C_t$  = Total capacity of a flight

$N_{\alpha, \beta}$  = Number of customers belonging to class  $\beta$  during time slice  $\alpha$ .

$F_{\beta}$  = Fare for class  $\beta$ .

$U_{\alpha, \beta}$  = Upper limit of demand for class  $\beta$  during time slice  $\alpha$ .

$L_{\alpha, \beta}$  = Lower limit of demand for class  $\beta$  during time slice  $\alpha$ .

For finding the objective function, the purpose of which obviously is to maximize the revenue, one have to assume some constraints, which can be:

First assumption is that there will be no cancellations and no-shows. The total number of passenger travelling should be less than or equal to the capacity of the flight.

The number of customers travelling in each class should be greater than or equal to lower bound and less than or equal to the upper bound.

On the basis of above assumptions, the objective function can be written as:

Max.  $\sum_{\beta} \sum_{\alpha} N_{\alpha, \beta} F_{\beta}$  ..... (1)

Subject to the constraints

$\sum_{\beta} \sum_{\alpha} N_{\alpha, \beta} \leq C_t$  &  $L_{\alpha, \beta} \leq N_{\alpha, \beta} \leq U_{\alpha, \beta}$  for all  $\alpha$  and  $\beta$ ,

$N_{\alpha, \beta} \geq 0$ , which indicates that number of customers in each time slice can be positive only

## **3. Implementation using Genetic Algorithm**

The formulation specified in above section is basically a Linear Programming Problem with the variables being assumed as integers. Therefore the problem actually becomes an integer programming problem. This problem can easily be solved using any standard method to solve LPP. But the traditional LPP methods becomes complex when there is a presence of discrete integer variables. Also there is large amount of input information is required such as constraints, therefore necessitating complex modelling and simulation. Therefore instead of using the basic techniques, an attempt has been made to solve the problem using genetic algorithm.

The basic advantage of using GA lies in the fact that it starts searching for optimal solution from multiple points, while most of the other methods start from a single point in search space. Thus there is less chance of sticking at local maxima when GA is being used. Further, if someone

wishes to modify the model for the whole airlines, it can be easily implemented using GA. Additionally GA does not require gradient and derivative information and thus can be used to solve even complex real world problems with discontinuous functions. [17].

For solving the above formulated problem, a genetic simulator has been implemented using Matlab and is explained below:

### **Representation**

To solve any problem using Genetic Algorithm, the very first issue is the selection of encoding scheme to represent the chromosomes. There are a number of encoding schemes available in literature such as binary, octal, hexadecimal, permutation and tree etc. Encoding scheme selected will further dictate what will be the crossover and mutation operators. There are two basic principles to select an encoding scheme: (i) Principle of minimal alphabet and (ii) Principle of meaningful building blocks [17]. It has been observed that in the said problem, binary encoding scheme is the base option that also satisfies both criteria. While this technique is very commonly used, in present case the first thing is to decide the length of the binary string. The number of genes in the binary string in present case is based on number of classes and number of time slices that are considered in an airplane. The number of genes will be evaluated as

$$\text{Numberofgenes} = \text{numberoftimeslices} * \text{numberofclasses} * \text{numberofbits} \dots\dots\dots (2)$$

For example, if it is assumed that number of time slices are 1, number of classes are 4 and number of bits are 8, then the number of genes will be 32 and these genes will be represented with a binary sequence generated randomly.

### **Evaluation Function**

Closely related to the representation, the issue of the evaluation function arises. This function associates a value to every individual in the population, and corresponds to the quality of that individual. Thus, different representations of the same problem may have different evaluation functions, since this is typically calculated from the values of the genes of each individual and through the genotype-to-phenotype mapping. The evaluation function is often referred to as fitness function in the Evolutionary Computation field. In proposed simulator, an evaluation function in Matlab has been designed to evaluate the chromosomes. The fitness value of each string (Chromosomes) is calculated by decoding the binary strings which represents the information about the number of passengers belonging to each fare class. However, the fitness value is bounded by the expected demand range for each fare class. The final parameter value  $\lambda$  for the number of customers belonging to a particular class can be evaluated as:

$$\lambda = L_b + ((U_b - L_b) * d) / (2^l - 1),$$

where  $d$  is the decoded value and  $l$  is the length of the chromosome.

### **Initial Population**

Once the representation is fixed, the first issue in developing the algorithm is that of the initial population. This is typically performed by randomly generating individuals so that the population can cover wider areas of the search space. Nonetheless, there are other more specialized methods. A very common approach is to generate the individuals in a greedy manner, which means that every individual is constructed in a way such that at every time, the next gene is given the value that optimizes the evaluation function for that individual. Occasionally, the solutions may be somehow seeded in areas where solutions are likely to be found. In proposed simulator the basic method of randomly generating individuals has been used.

GA parameters such as total number of generations, crossover probability, mutation probability, population size, total number of iterations are fed as input in this step. Here assumption is that the typical range for population is 50-500. The random numbers are generated assuming population is normally distributed.

### **Parent Selection**

Selection is the method through which certain elements in the population are chosen to be combined. This selection mechanism tries, in general, to choose parents that are likely to produce a high-quality descendant. Typically, two individuals are chosen to reproduce and yield descendants. Different kinds of selection mechanism are Roulette-Wheel, Ranking, Tournament, Multiparent, etc. In proposed simulator, researchers have used two selection mechanisms: (i) Roulette-Wheel and (ii) Tournament selection and are explained below:

**Roulette-Wheel Selection:** In this method, a certain probability is to be chosen for every individual in the population. This probability depends directly on the absolute fitness of the individual [18]. The main drawback of this mechanism is that the best candidates are very likely to take over the whole population very quickly which may result in the loss of diversity sometimes. Diversity is one of the main causes for the genetic algorithm being used so widely. To maintain diversity a combination of roulette-wheel selection with some other selection mechanism can be used.

**Tournament Selection:** This is one of the simplest mechanisms, and also the least time-consuming. It consists of choosing  $k$  individuals completely at random, and then selecting the two individuals with highest fitness function. Obviously, the complexity of this method depends on the value of  $k$ . The probability for tournament selection is fixed as 0.75 in proposed simulator.

### **Reproduction**

For the purpose of reproduction, the parents(individuals) are combined in such a way that a high quality individual (descendant) will be obtained. This mechanism is also known as crossover. In some GAs, this operator is able to generate more than one descendant (usually two), but one

will assume from now on that only one descendant is going to be generated. Thus, different crossover operators are one point, multiple point, uniform, arithmetic, etc. In present case, one point, two point and uniform crossover has been used, and are explained as follows:

**One point crossover:** This is the most popular method. It consists of choosing a point randomly, and copying the genes of a parent, from the beginning until this point, to the descendant, and the genes of the other parent from that point till the end.

As an example, imagine two parents of the form:

First = ( 0 1 0 0 1 1 1 1 0 ), Second = ( 0 0 0 1 1 0 1 0 0 )

and  $k = 5$  is the crossover point, the descendant would either be

( 0 1 0 0 1 / 0 1 0 0 ) or ( 0 0 0 1 1 / 1 1 1 0 )

**Two point crossover:** It is similar to the previous crossover, and the only difference is that instead of 1 point, two points are chosen randomly. Then, to generate a descendant it would copy the genes of each parent in turns after each crossover point.

**Uniform crossover:** It is slightly different than the previous one. It treats each gene independently and decides from which parent it is going to be inherited (typically with the same probability).

#### **Mutation**

This operator is the source of great diversity. It is based in the biological fact that some genes can mutate for different reasons, and thus, the descendant can acquire genes that are from neither of its parents. The most common ones are random bit modification, swap mutation, insert mutation, scramble mutation, etc. In this particular case, the random bit modification has been used which is explained below:

**Random bit modification:** Consists on changing the value of some bits with a given probability. The operator changes the value of every bit in the sequence with a certain probability. If the representation is binary as in our case, the effect is that of flipping a bit, either from 0 to 1 or from 1 to 0.

#### **Selection of the New Generation**

This is the mechanism that replaces the last population by a new one. In order to do so, some algorithms completely replace the previous population for the new set of descendants or offspring. However, this is usually not a very effective technique, and GAs normally implements mechanism to generate the new population from both, the previous one and the offspring. Some of these mechanisms are fitness based, generation based, replace worst, etc. Derived forms of generational update are used like  $(\lambda + \mu)$ -update and  $(\lambda, \mu)$ -update. This time from a parent population of size  $\mu$ , a little of children is produced of size  $\lambda \geq \mu$ . Then the  $\mu$  best individuals from either the offspring population or the combined parent and offspring populations (for  $(\lambda, \mu)$ - and  $(\lambda + \mu)$ -update respectively), form the next generation[19]. In proposed simulator the fitness based replacement has been used i.e. selection focuses on keeping the individuals with higher fitness for

the next generation. A technique associated with this operator (independent of the mechanism type) is to always maintain the highest quality individual in the population. This technique is usually referred to as elitism.

#### **Termination**

The termination condition indicates when it is time for the algorithm to stop. At this point, the algorithm will usually return the best individual according to its fitness function.

One can distinguish two kinds of termination condition:

**Objective reached:** when a GA is implemented to reach a certain goal (i.e., a solution of a certain quality), reaching that goal should be the indication for the algorithm to stop.

**External conditions:** However, the previous case is very rarely achieved, due to the stochastic nature of these algorithms. Therefore, different criteria must be used. Different conditions include fixed number of generations reached, maximum time allowed reached, fitness improvement does not occur for a certain period of time/generations, manual inspection, a combination of the above. In proposed simulator, researchers have maintained a fixed number of iterations.

#### **Genetic Simulator: Single-Leg Flight Yield Management**

1. Init\_pop = Randomly Generated population.  
//Initialize Population with size n
2. curr\_pop = Init\_pop.
3. While ( !stop\_criterion)  
// Condition to stop with maximum iteration being fixed as max
4. Evaluate Fitness of curr\_pop.
5. Create mating pool according to Roulette-wheel Selection OR Tournament Selection.  
// Selection
6. Apply Crossover like One-point, Two-point & Uniform Crossovers on mating pool with probability 0.80.
7. Apply Mutation on mating pool with probability 0.03.  
// Mutation
8. Replace generation with  $(\lambda + \mu)$ -update as curr\_pop.  
// Replacement
9. End While
10. End

#### **4. Results**

Genetic algorithm is used as a solution technique for the single-leg airlines problem using various combinations of different operators.

In this case, a single flight is considered to operate between given origin and destination. The capacity of the flight is assumed to be 100. The following GA parameters are taken into considerations:

Population size = 75

Maximum number of iterations = 50



Cross-over probability = 0.80

Mutation probability = 0.03

Tournament Selection parameter = 0.75

Number of simulations = 30

Using the above parameters and various combinations one can get the tables 1 and 2, which are shown at the end of the paper.

Results obtained for each combination of operators are shown below:

- Using combination of Roulette Wheel selection and the three types of cross-overs i.e. one-point, two-point and uniform, the following results were obtained:

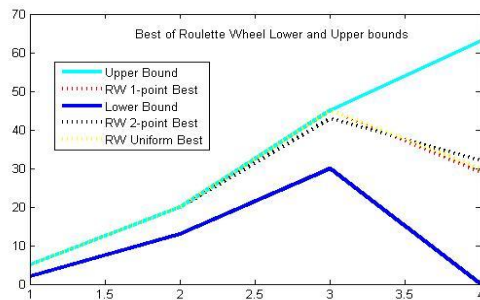


Fig.1 : Lower Bound, Upper Bound, and Estimated Fitness

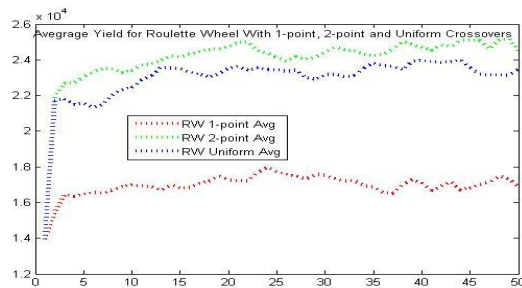


Fig.2: Average Fitness

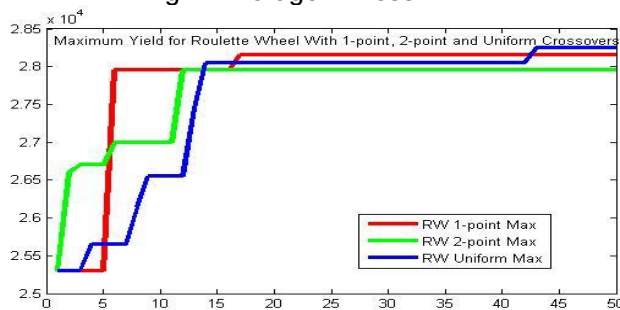


Fig.3: Maximum Fitness

- Using combination of Tournament selection and the three types of cross-overs i.e. one-point, two-point and uniform, following results have been observed:

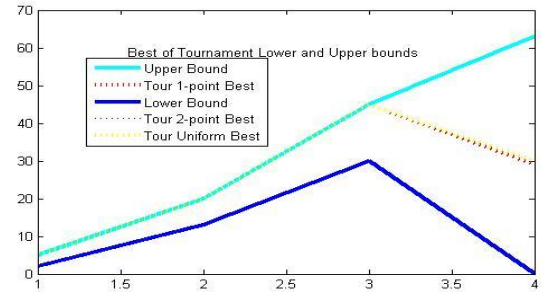


Fig.4 : Lower Bound, Upper Bound, and Estimated Fitness

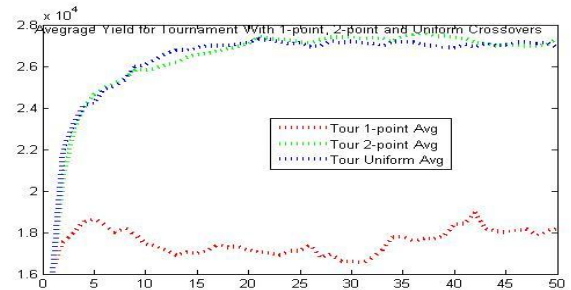


Fig.5: Average Fitness

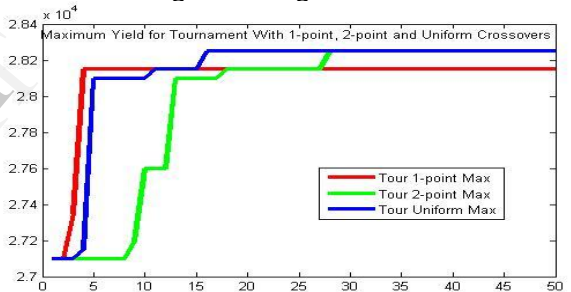


Fig.6: Maximum Fitness

- Using combination of Roulette Wheel and Tournament selection along with one-point crossover, following results have been obtained:

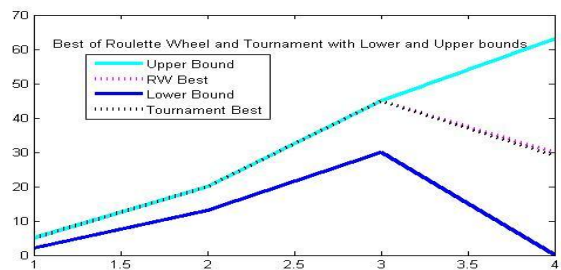


Fig.7 : Lower Bound, Upper Bound, and Estimated Fitness

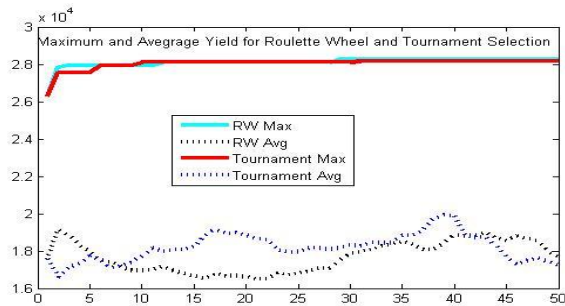


Fig.8 : Maximum and Average Fitness

- Using combination of Roulette Wheel and Tournament selection along with two-point crossover, following results have been observed:

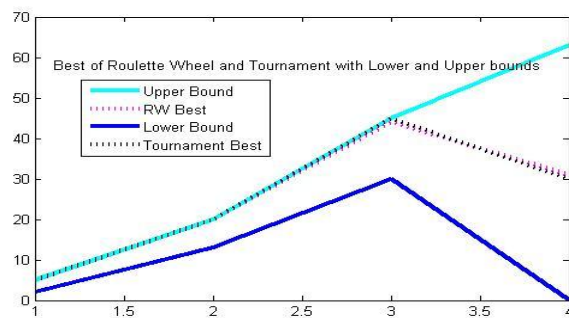


Fig.9 : Lower Bound, Upper Bound, and Estimated Fitness

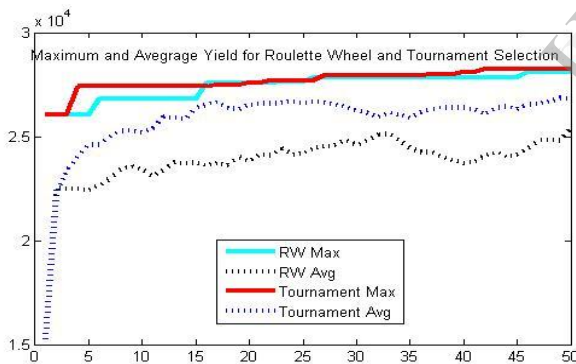


Fig.10 : Maximum and Average Fitness

- Using combination of Roulette Wheel and Tournament selection along with uniform crossover, following results have been observed:

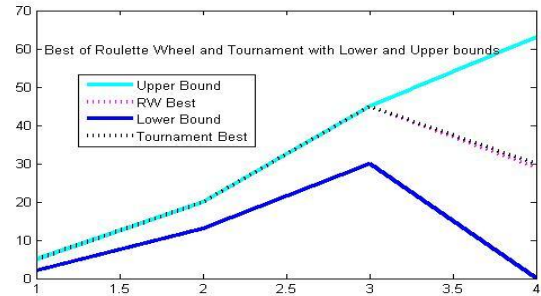


Fig.11 : Lower Bound, Upper Bound, and Estimated Fitness

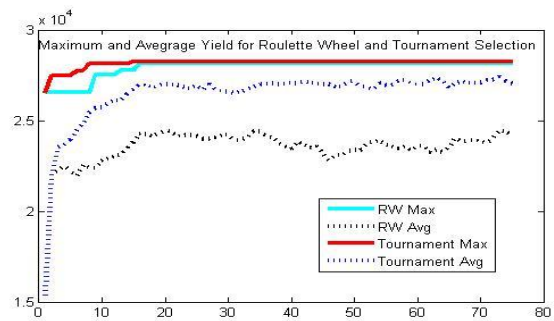


Fig.12 : Maximum and Average Fitness

## 5. Interpretation

Upon comparing the above results with [16], it has been observed that the solution found using GA is optimal and the results obtained are at par with [16]. Looking at the tables and graphs obtained by different combinations of various selection and cross-over methods the following findings has also been observed:

1. The best estimation is obtained in most of the cases when tournament selection is used irrespective of the cross-over operators, while when roulette wheel selection is used the best estimation is generally obtained in case of one-point cross-over.
2. There are three different combinations which can be considered as better combinations as compared to others. These are tournament selection along with uniform cross-over or one-point cross-over and roulette wheel selection with one-point cross-over.
3. The combinations which should be discarded completely are roulette wheel selection and two-point cross-over, roulette wheel selection and uniform cross-over as the results produced by these combinations are not found to be very useful.
4. Looking at various graphs for the average fitness of the population, it is clearly visible that one-point cross-over always yields not a very good average fitness, while when two-point or uniform cross-over is used the average fitness is quite good and almost same in each case.

## 6. Conclusion

In this paper, an attempt has been made to solve the problem of yield management using Genetic Algorithm, GA has been used in the said experiment with a number of variants in selection and crossover. It is inferred that GA can be successfully used in the problem of yield management. Yield management is a problem of optimization and GA has been used in past in a number of other problems of optimization. GA being an evolutionary algorithm is a good option for such type of problems where the result should evolve with passage of time. The particular case study which has been picked is airlines yield management. The various combinations of different operators were tried in an attempt to find the best possible combination for maximizing the profit for airlines. The combinations which proved to be best were tournament selection along with either uniform cross-over or one-point cross-over. However, if one takes into account the improvement in the total population, then the one-point cross-over fails. In that case the only combination found to be very good is tournament selection along with uniform cross-over.

## 7. Future Scope

In this paper a basic case of yield management in airlines has been taken into consideration with only a single decision period. The results obtained although can prove to be useful for airlines industry; still there are a number of things that can be considered for practical implementation. Some of them can be consideration of more than one decision period, overbooking and cancellation, arrival pattern of customers, etc. The concept of genetic algorithm can also be applied to other industry such as hotel industry, sea-cargo industry, etc.

## 8. References

- [1] Netessine, S. and R. Shumsky (2002), *Introduction to the Theory and Practice of Yield Management*, INFORMS Transactions on Education, Vol. 3, No. 1
- [2] Rothstein, M. 1971. *An Airline Overbooking Model*. Trans. Sci. 5, 180-192.
- [3] Littlewood, K. 1972. *Forecasting and Control of Passenger Bookings*. AGIFORS 12th Annual Symposium Proceedings, 95-128, Nathanya, Israel
- [4] Belobaba, P. (1987). *Air Travel Demand and Airline Seat Inventory Management*. Ph.D. Dissertation, MIT.
- [5] Belobaba, P. 1989. *Application of a Probabilistic Decision Model to Airline Seat Inventory Control*. Ops. Res. 37, 183-197.
- [6] Belobaba, P. (1987), *Airline Yield Management: An Overview of Seat Inventory Control*, Transportation Science, 21, 63-73.
- [7] Smith, B., J. Leimkuhler, R. Darrow, J. Samuels. 1992. *Yield Management at American Airlines*. Interfaces, 22, 8-31.
- [8] Gallego, G., G. van Ryzin G. 1994. *Optimal Dynamic Pricing of inventories with Stochastic Demand over Finite Horizons*. Mgmt. Sci. 40, 999-1020.
- [9] Bitran, G., S. Monschein. 1997. *Periodic Pricing of Seasonal Product in Retailing*. Mgmt. Sci. 43, 427-443.
- [10] Feng, Y., G. Gallego. 1995. *Optimal Starting Times for End-of-Season Sales and Optimal Stopping Times for Promotional Fares*. Mgmt. Sci. 41, 1371-1391.
- [11] Feng, Y., G. Gallego. 2000. *Perishable Asset Revenue Management with Markovian Time Dependent Demand Intensities*. Mgmt. Sci. 46, 941-956.
- [12] Unternehmen und Markte (1992) *Unternehmen und Markte in Handelsblatt*, 12 April 1992, p. 13.
- [13] Robert G Cross, *Revenue Management: Hard-Core Tactics for Market Domination*, 1997, Broadway Books, New York
- [14] Gabriel Bitran and Rene Caldentey, *An Overview of Pricing Models for Revenue Management*, December, 2002
- [15] Aloysius George, B.R.Rajakumar, D. Binu, *Genetic Algorithm Based Airlines Booking Terminal Open/Close Decision System*, ICACCI'12, August 3-5, 2012, Chennai, T Nadu, India.
- [16] Srinivas S. Pulugurtha & Shashi S. Nambisan, *A Decision-Support Tool for Airline Yield Management Using Genetic Algorithm*, Computer Aided Civil and Infrastructure Engineering, 2003, 214-233.
- [17] Goldberg, D.E.(1989), *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley, New York.
- [18] J.H. Holland, *Adaption in Natural and Artificial Systems*, MIT Press, 1975.
- [19] S.N.Sivanandam, S.N.Deepa, *Introduction to Genetic Algorithm*, Springer-Verlag Berlin Heidelberg 2008

TABLE1: LOWER, UPPER AND BEST ESTIMATED DEMANDS IN EACH ASSUMED FARE CLASS

Fare Class	Fare	Demand		Best Estimation
		Lower Limit	Upper Limit	
1	100	0	63	30
2	250	30	45	45
3	500	13	20	20
4	800	2	5	5

TABLE 2: NUMBER OF ITERATIONS FOR THE BEST ESTIMATES IN VARIOUS COMBINATIONS OF SELECTION AND CROSS-OVER PARAMETERS

Parameters		Iterations		No. of simulation when max. is not achieved in specified iterations
Selection	Cross-over	Min. in all simulations	Avg. of all simulation except when max is not achieved	
Roulette Wheel	One-Point	4	19	9
Roulette Wheel	Two-Point	23	27	25
Roulette Wheel	Uniform	17	32	24
Tournament	One-Point	2	19	6
Tournament	Two-Point	9	23	6
Tournament	Uniform	5	17	2