



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

A Simulator to Prevent Premature Convergence in YM using GA

P.K.Suri^{#1}, Rakesh Kumar^{#2}, Pardeep Kumar Mittal^{#3}

^{#1}Dean (R&D), Chairman & Professor (CSE/IT/MCA), H.C.T.M., Kaithal (Haryana), India

^{#2}Professor, Deptt. of Computer Science & Applications, K.U., Kurukshetra (Haryana), India

^{#3}Assistant Professor, Deptt. of Computer Science & Applications, K.U., Kurukshetra (Haryana), India

Abstract- The use of Elitism within Genetic Algorithm is well augmented allowing the best solution from any generation to be carried across the new population allowing it to sustain. Social Disaster Techniques (SDT) are generally used when premature convergence occurs and by creating random offspring the problem of premature convergence may be avoided by inserting diversity in the population. This paper attempts to use the both concepts of Elitism and Social Disaster techniques spanning across various generations. A previous solution is chosen and it has been looked upon how Elitism and Social Disaster techniques fares towards the same problem. The problem that has been taken into consideration is that of Yield Management.

Keywords- Elitism, Social Disaster, Multiple Time-Period, Premature Convergence, Yield Management

I. Introduction

Since its origin GA has been improving day by day. The concept of Elitism has now been used for quite a long time. Elitism is basically a replacement policy, which is also sometimes known as replace worst policy. Another concept applied in this paper is that of Social Disaster Techniques. These techniques are basically designed to avoid premature convergence by introducing diversity in the population. The SDT comprises of two techniques named as Judgement Day and Packing. The purpose of both remains same. The work presented in this paper is basically using elitism and SDT on an earlier solution found by Suri et.al. [1]. The problem identified in that solution was that of premature convergence in Roulette Wheel Selection with one-point crossover during second time-slice. The assumptions remain same as that in the previous paper.

In the present work authors intend to add elitism and SDT to see their impact on convergence, if any. The work that has been done in this paper is implementing simulator for genetic algorithm using elitism and using SDT. The paper has been divided into eight sections including present one, the section 2 provides related literature, while in section 3 problems has been defined and formulated. Section 4 discuss the concept of Elitism and SDT and section 5 implements the problem using Modified GA with Elitism, Modified GA with Judgement Day Technique and Modified GA with Packing Technique. In section 6 results are presented, while section 7 interprets the results. In the last section conclusion has been drawn.

II. Literature Review

In the literature elitism has been used in a number of applications and it has been observed that elitism prove to be very good replacement strategy for the purpose of improved convergence. Some of the applications where elitism has been used successfully are described briefly:

Leung and Liang [2] introduced a new technique called “adaptive elitist population search method”, which allowed extension of unimodal function optimization methods to locate all optima of multimodal problems in an efficient manner. In [3], a replacement strategy for steady-state genetic algorithms considering two features of the candidate chromosome to be included into the population: a measure of the contribution of diversity to the population and the fitness function. In [4], the relationship between the elitist genetic algorithm performance and algorithm termination criteria was examined. A genetic algorithm with elitism has been proposed by [5]. It has been established in that article that convergence has been achieved in a number of numerical examples by using elitism. In a paper by [6], an elitism-based immigrant scheme for genetic algorithms has been proposed. The immigrants have been found to be more adapting using elitism. Another proposal in the same paper has also been provided which is hybrid immigrants scheme using elitism based method and traditional random approaches. Results in both the cases have been found to be better than traditional approaches. This paper [7] studies the influence of elitism in Evolutionary Multiobjective Optimization. The paper presents some algorithms which use elitism and some algorithms which not use elitism and realizes a comparison of some algorithms for each category. The SDT has been mostly used in TSP so far. In other cases SDT has not been much tried. Therefore, this attempt to implement SDT on Yield Management is more or less first of its kind. Some of the literature related to SDT is given below:

In a paper by [8], some new features in genetic solution of the TSP have been suggested. One of those features in applying SDT very effectively. SDT also have been successfully used in TSP by [9] using DARO.

In another paper for preventing premature convergence to local maxima in GA via random offspring by [10], some of the basic techniques were compared with an innovative one, the Random offspring Generation, was evaluated in its merits. Again TSP was taken as a problem.

III. Problem Definition and Formulation

The problem considered in this paper is basically Yield Management problem with multiple time-periods. 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 is divided more than one time slices. Another assumption is to fix the fare of each class during each time slice and also assumed as known.

The objective function can be defined as: $\text{Max. } \sum_{\beta} \sum_{\alpha} N_{\alpha, \beta} F_{\beta}$

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 α and β ,

$N_{\alpha, \beta} \geq 0$, Where C_t = Total capacity of a flight

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

F_{β} = Fare for class β .

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

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

The solution of the above stated problem has already been identified by an earlier work [1] and is shown in table 1. But the problem with that solution was that in case of Roulette Wheel selection and one-point crossover the convergence stuck to the local maxima in almost all the cases. So the problem of premature convergence was found. Therefore the GA was modified in two different ways i.e. using elitism and SDT in an attempt to get better results.

IV. Elitism and SDT

Elitism: In elitism, complete population of genome is replaced except for the best member of each generation which is carried over to next generation without modification [11]. When creating a new population by crossover and mutation, there are chances that the best chromosome is lost. Elitism is the name of the method that first copies the best chromosome (or few best chromosomes) to the new population [12]. The rest of the population is constructed according to GA. Elitism can rapidly increase the performance of GA, because it prevents a loss of the best found solution.

SDT: The Social Disasters Technique (SDT) was introduced by Kureichick and colleagues [13] in order to avoid the premature convergence to local optima, when the GA is applied to the TSP. The general idea is to diagnose the situations of loss of genetic diversity of the population, and in such a case to apply a catastrophic operator to it. These operators were defined with the purpose to return the population to an acceptable degree of genetic diversity, by replacing a number of selected individuals, by others, generated at random. Two different operators were considered.

Packing. Of all the individuals having the same fitness value, only one remains unchanged; all the others are fully randomized.

Judgment Day. Only the individual with the best fitness value remains unchanged; all the others are fully randomized.

V. Implementation using Genetic Algorithm

For solving the above formulated problem, first a modified genetic algorithm with elitism has been proposed which is implemented using Matlab and is stated below:

Genetic Algorithm with Elitism for Yield Management

1. Init_pop = Randomly Generated population.
2. curr_pop = Init_pop.
3. While (!termination_criterion)
4. Evaluate Fitness of curr_pop using fitness function.
5. Select mating pool according to Roulette-wheel Selection.
6. Apply One-point Crossover on mating pool with probability 0.80.
7. Apply Mutation on mating pool with probability 0.03.
8. Replace the worst chromosome (individual) in the population with the best one and add to curr_pop.
// Elitism
9. Replace remaining generation with $(\lambda + \mu)$ -update and add to curr_pop .
10. End While
11. End

In this algorithm the elitism has been applied in 8th step which is replacing the worst fit individual with the best fit individual. The other parameters of the GA remain same as in a normal GA.

Now another variation of GA with Judgement Day will be proposed and is shown below:

Genetic Algorithm with Judgement Day as SDT

1. Init_pop = Randomly Generated population of size N.
2. curr_pop = Init_pop.
3. While (!termination_criterion)
4. Evaluate Fitness of curr_pop using fitness function.
5. Select mating pool according to Roulette-wheel Selection.
6. Apply One-point Crossover on mating pool with probability 0.80.
7. Apply Mutation on mating pool with probability 0.03.
8. If standard_Deviation of fitness == 0 // Plateau
9. F_B = Best Fit Individual // For Judgement Day SDT
10. Temp_pop = Randomly generated population of size N-1.
11. New_pop = Temp_pop + F_B
12. Curr_pop = New_pop
13. Else
14. Replace remaining generation with (λ + μ)-update and add to curr_pop .
15. End if
16. End While
17. End

In this algorithm Judgement Day technique has been applied to avoid premature convergence. As shown in step 8, the standard deviation for the last 20 iteration is calculated (not shown in algorithm) and if it is found to be zero meant that a plateau has reached. Therefore Judgement Day technique is applied i.e. the best fit individual is sustained while the other individuals are randomly created in next iteration. The Judgement Day is applied three times as and when plateau has reached. In other iterations normal GA is implemented with standard method. The standard deviation is calculated using the following formula.

$$S = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$$

where N is the number of elements in the sample and

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

In the end, third variation of GA with packing SDT will be presented:

Genetic Algorithm with Packing as SDT

1. Init_pop = Randomly Generated population.
2. curr_pop = Init_pop.
3. While (!termination_criterion)
4. Evaluate Fitness of curr_pop using fitness function.
5. Select mating pool according to Roulette-wheel Selection.
6. Apply One-point Crossover on mating pool with probability 0.80.
7. Apply Mutation on mating pool with probability 0.03.
8. If standard_Deviation of fitness == 0 // Plateau
9. F_p = Select one individual (chromosome) each from the groups having same fitness values.
// For Packing in SDT
10. Temp_pop = Random Generated of size N- sizeof(F_p)
11. Curr_pop = Temp_pop + F_p
12. Else
13. Replace generation with (λ + μ)-update and add to curr_pop.
14. End if
15. End While
16. End

In the above algorithm Packing technique has been applied with the same purpose of avoiding premature convergence. As shown again in step 8, the standard deviation for the last 20 iteration is calculated (not shown in algorithm) and if it is found to be zero meant that a plateau has reached. When a Plateau has reached, the Packing is applied i.e. one individual is selected from each group having same fitness is selected while the other individuals in each group are randomly created in next iteration. The Packing is also applied three times as and when plateau has reached, otherwise normal GA is applied in all other iterations.

All the three algorithms were implemented using MATLAB and the results are shown and explained through graphs and tables in the next section.

VI.Results

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 = 200; Cross-over probability = 0.80

Mutation probability = 0.03; Number of simulations = 30

Using the above parameters and various combinations one can get graphs as shown in fig. 1- fig.12. Also a table i.e. table 2 is drawn in the end which shows the comparison of maximum yield obtained in each case.

Results obtained for each combination of operators are shown below:

- Using combination of Roulette Wheel selection, Roulette Wheel selection with Elitism, and one-point crossover in both cases, the following results were obtained during first time period in a Multi Time-Period Environment:

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

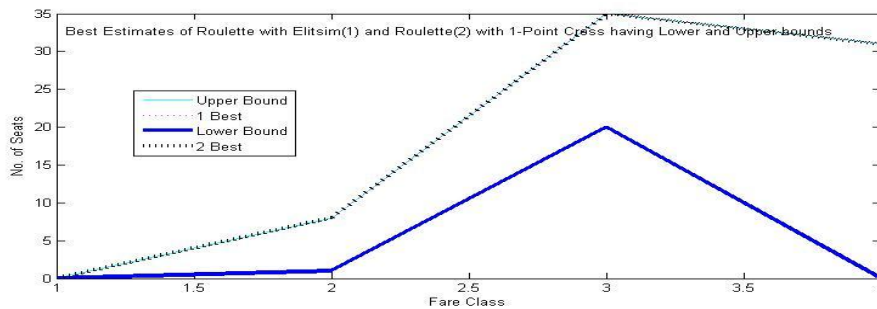
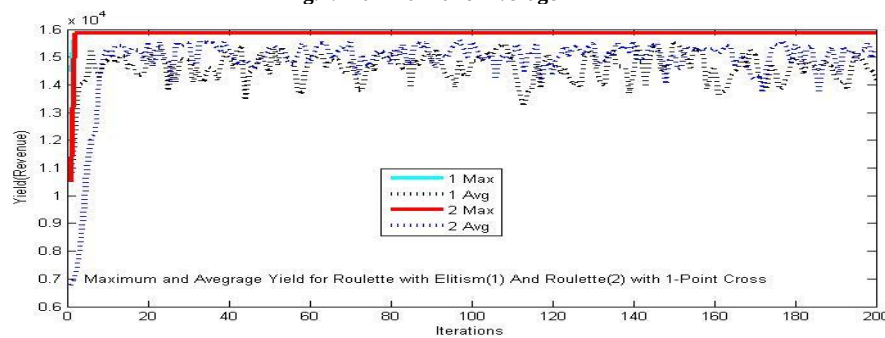


Fig.2: Maximum and Average



Fitness

- Using combination of Roulette Wheel selection, Roulette Wheel selection with Elitism, and one-point crossover in both cases, the following results were obtained during second time period in a Multi Time-Period Environment:

Fig.3: Lower Bound, Upper Bound, and Estimated Fitness

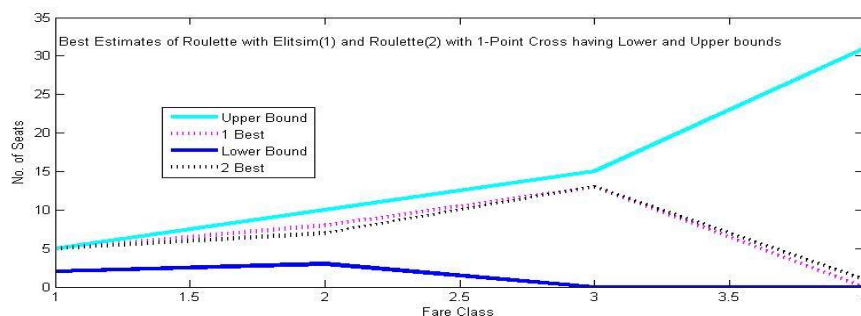
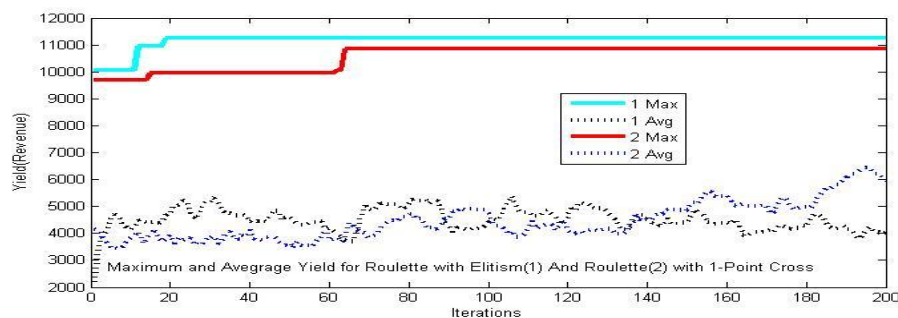


Fig.4: Maximum and Average Fitness



- Using combination of Roulette Wheel selection, Roulette Wheel selection with Judgement Day Technique, and one-point crossover in both cases, the following results were obtained during first time period in a Multi Time-Period Environment:

Fig. 5: Lower Bound, Upper Bound, and Estimated Fitness

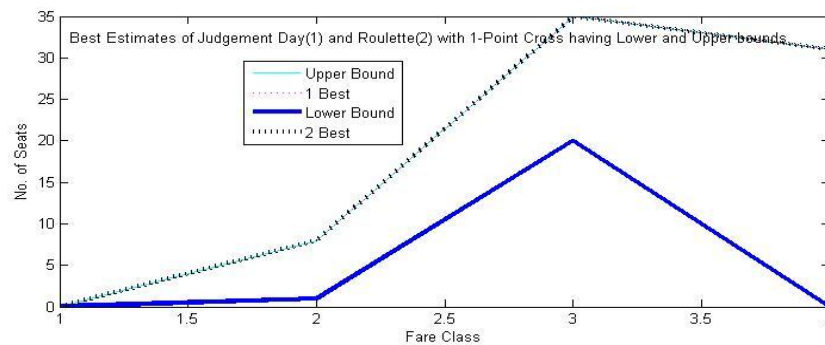
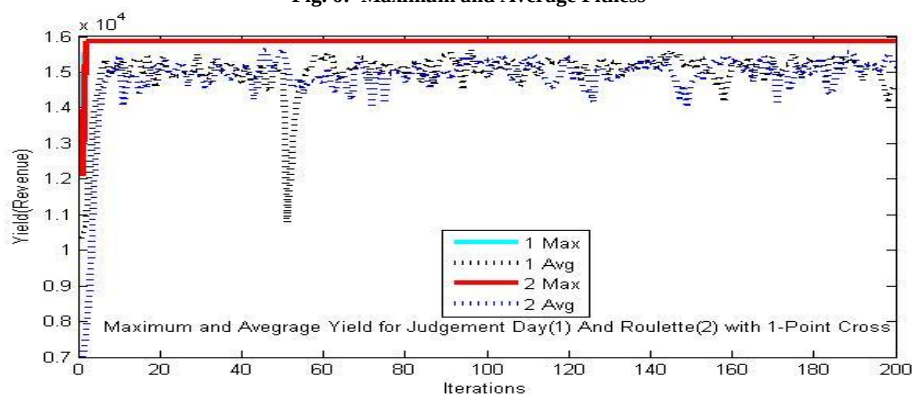


Fig. 6: Maximum and Average Fitness



- Using combination of Roulette Wheel selection, Roulette Wheel selection with Judgement Day Technique, and one-point crossover in both cases, the following results were obtained during second time period in a Multi Time-Period Environment:

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

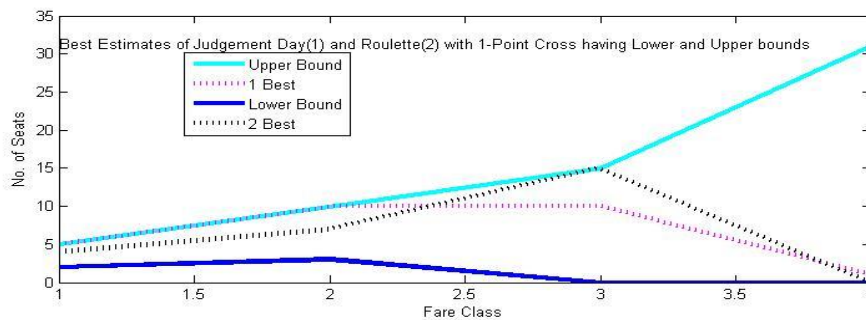
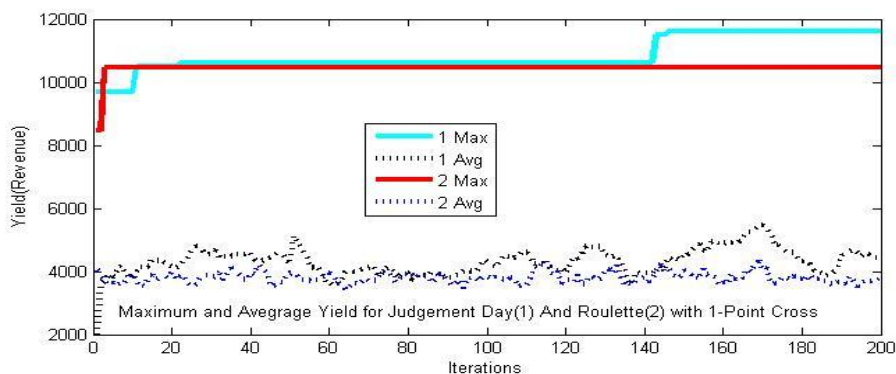


Fig. 8: Maximum and Average Fitness



- Using combination of Roulette Wheel selection, Roulette Wheel selection with Packing Technique, and one-point crossover in both cases, the following results were obtained during first time period in a Multi Time-Period Environment:

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

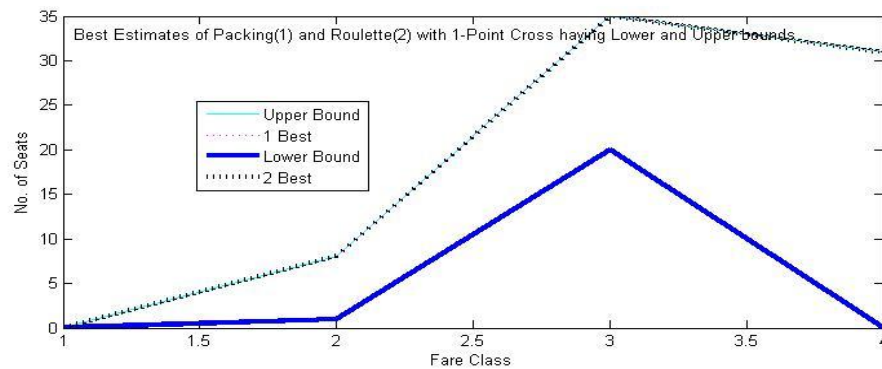
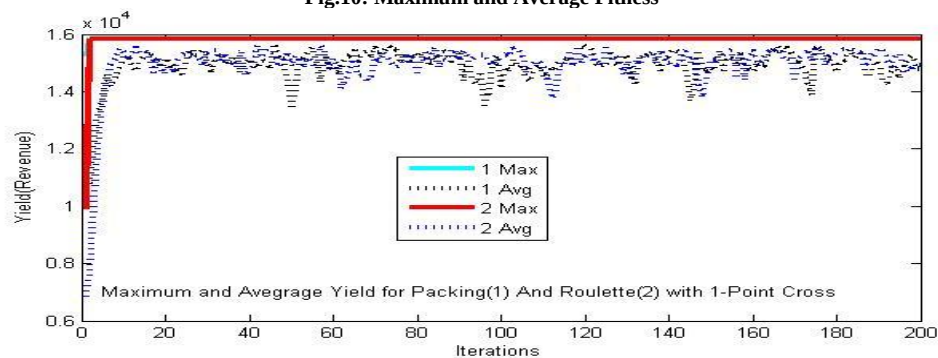


Fig.10: Maximum and Average Fitness



- Using combination of Roulette Wheel selection, Roulette Wheel selection with Packing Technique, and one-point crossover in both cases, the following results were obtained during first time period in a Multi Time-Period Environment:

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

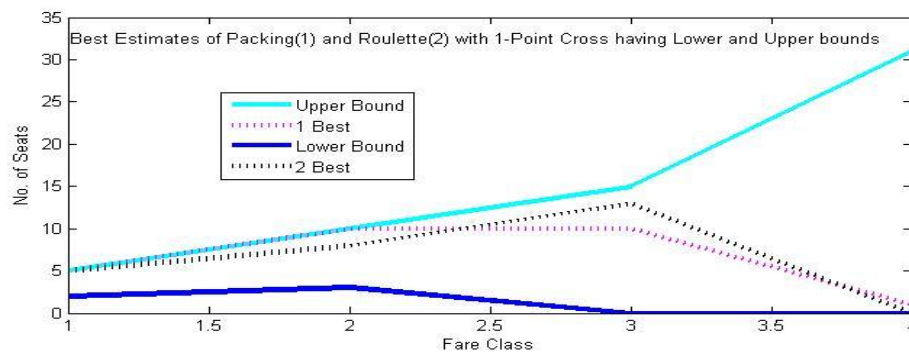
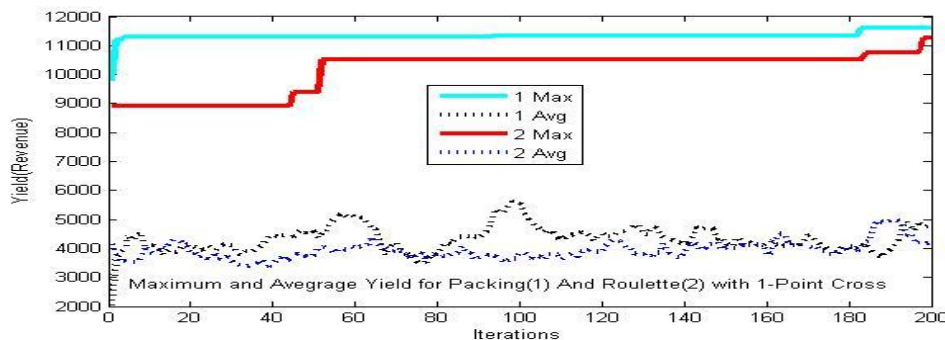


Fig.12: Maximum and Average Fitness



- The table 1 shows the best estimated number of seats allocated for each time-period as obtained from [1].

Table1: Lower, Upper and Best estimated demands in each assumed fare class in multiple time-slices

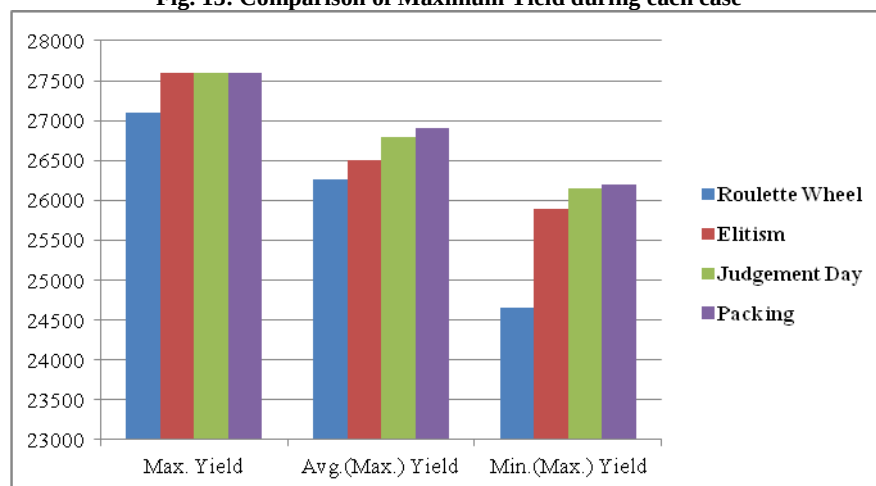
Fare Class	Fare	Time-Slice I			Time-Slice II		
		Demand			Demand		
		Lower Limit	Upper Limit	Best Estimates	Lower Limit	Upper Limit	Best Estimates
1	100	0	31	31	0	31	0
2	250	20	35	35	0	15	11
3	500	1	8	8	3	10	10
4	800	0	0	0	2	5	5

- Finally table 2 shows the comparison of total of maximum yield obtained during all the time-periods in each case i.e. Roulette Wheel selection, Roulette Wheel selection with Packing Technique, Roulette Wheel selection with Judgement Day Technique and Roulette Wheel selection with Elitism along with one-point crossover. The comparison can also be shown graphically through fig.13.

Table 2: Comparison of Maximum Yield obtained during each case. The Average and Minimum within Maximum yield obtained is also considered

	Roulette Wheel	Elitism	Judgement Day	Packing
Max. Yield	27100	27600	27600	27600
Avg.(Max.) Yield	26270	26500	26800	26900
Min.(Max.) Yield	24650	25900	26150	26200

Fig. 13: Comparison of Maximum Yield during each case



VII. Interpretation

Upon comparing the results of various techniques implemented in this paper, it has been observed that the results obtained by all the methods are same as shown in table 1 during time-period 1. But the results found in time-period 2 varies from the best estimated. In case of roulette wheel the optimum solution is never obtained in the specified 200 iterations. While in other cases, it is visible that in some cases optimum results are obtained. The findings of the paper can be summarized as follows:

- By looking at fig. 1, fig.5 and fig.9, it is clear that global optimum is achieved in each case, which means if the global optimum is already obtained then the elitism, and SDT does not change it.
- Looking upon fig.2, fig.6 and fig.10, it can be observed that the average fitness of all types of populations is also very good during first time-period. Although it can also be observed that in fig.6 and fig.10, there are slight dips in the average population which are due to SDT.
- Looking at fig. 4, fig. 8 and fig. 12, it is clearly visible that elitism, and both the SDT performs better than roulette wheel technique and the pre-mature convergence is avoided in all the cases.
- Observing table 2 and graph 13, it is found that elitism and the SDT outperform the simple roulette wheel selection mechanism in most of the cases. However, it is visible that SDT provides better results than elitism. In can also be concluded that premature convergence can be eliminated by using SDT and elitism.

VIII. Conclusion

In this paper, the problem of premature convergence in yield management with the help of elitism and SDT in GA has been considered. It has been observed that convergence has been improved with the help of elitism and SDT in almost all the cases. The premature convergence problem can be avoided up to some extent through elitism and SDT. Till date SDT has only been applied to TSP. With the help of this paper, it can be observed that it can also be useful in Yield Management. In future SDT can also be applied to some other problems that are solved through GA.

IX. References

- [1] Suri P.K., Kumar Rakesh, Mittal Pardeep Kumar, "Simulating the Multiple Time-Period Arrival in Yield Management", *International Journal of Computer Trends & Technology (IJCTT)*, ISSN 2231-2803, Vol.4. Issue 10, October, 2013
- [2] Leung, K.-S. and Liang, Y., "Adaptive Elitist-Population Based Genetic Algorithm for Multimodal Function Optimization", *Springer-Verlag Berlin Heidelberg, GECCO (2003)*, 1160-1171
- [3] Lozano, M., Herrera, F., Cano, J.R., "Replacement Strategies to Preserve Useful Diversity in Steady-State Genetic Algorithms", *Information Sciences*, 178 (2008), 4421–4433
- [4] Kim, J.-L., Examining the Relationship Between Algorithm Stopping Criteria and Performance Using Elitist Genetic Algorithm, *Proceedings of the 2010 Winter Simulation Conference*, (2010), 3220-3227
- [5] Biman Chakraborty and Probal Chaudhuri, "On The Use of Genetic Algorithm with Elitism in Robust and Nonparametric Multivariate Analysis", *Austrian Journal of Statistics* Volume 32 (2003), Number 1&2, 13–27
- [6] Shengxiang Yang, "Genetic Algorithms with Elitism-Based Immigrants for Changing Optimization Problems", *EvoWorkshops 2007, LNCS 4448*, pp. 627–636, 2007, Springer.
- [7] Crina Groşan, Mihai Oltean., Mihaela Oltean, "The role of elitism in Multiobjective Optimization with Evolutionary Algorithms", <http://www.cs.ubbcluj.ro/~cgrosan/grosanZUAL.pdf>
- [8] V. M. Kureichick, A. N. Melikhov, V. V. Miagkikh, O. V. Savelev and A. P. Topchy., "Some New Features in Genetic Solution of The Traveling Salesman Problem", *Proceedings of ACEDC'96 PEDC, University of Plymouth, UK*.
- [9] Nitin S. Choubey, "Solving TSP using DARO", *International Journal of Computer Applications (0975 – 8887) Volume 58–No.9, November 2012*
- [10] Miguel Rocha and Jose Neves, "Preventing Premature Convergence to Local Optima in Genetic Algorithms via Random Offspring Generation".
- [11] Affenzeller, M., Winkler S. and Wagner S., (2009) "Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications", *Chapman-Hall CRC*, ISBN 1584886293
- [12] De Jong, K.A., (1975) "An Analysis of the behaviour of a class of genetic adaptive systems". (Doctoral dissertation, University of Michigan), 36(10), 5140B (University Microfilms No. 76-9381).
- [13] V.Kureichick, A.N.Melikhov, V.V.Miaghick, O.V.Savelev and A.P.Topchy, "Some New Features in the Genetic Solution of the Traveling Salesman Problem". In Ian Parmee and M.J.Denham eds. *Adaptive Computing in Engineering Design and Control 96(ACEDC'96)*, 2nd International Conference of the Integration of Genetic Algorithms and Neural Network Computing and Related Adaptive Computing with Current Engineering Practice, Plymouth, UK, March 1996.