Blending Roulette Wheel Selection & Rank Selection in Genetic Algorithms

Rakesh Kumar, *Senior Member, IACSIT* and Jyotishree, *Member, IACSIT*

***Abstract*—Both exploration and exploitation are the techniques employed normally by all the optimization techniques. In genetic algorithms, the roulette wheel selection operator has essence of exploitation while rank selection is influenced by exploration. In this paper, a blend of these two selection operators is proposed that is a perfect mix of both i.e. exploration and exploitation. The blended selection operator is more exploratory in nature in initial iterations and with the passage of time, it gradually shifts towards exploitation. The proposed solution is implemented in MATLAB using travelling salesman problem and the results were compared with roulette wheel selection and rank selection with different problem sizes.**

***Index Terms*—Genetic algorithm; rank selection; roulette wheel; selection.**

I. Introduction

Genetic algorithms are adaptive algorithms proposed by John Holland in 1975 [1] and were described as adaptive heuristic search algorithms [2] based on the evolutionary ideas of natural selection and natural genetics by David Goldberg. They mimic the genetic processes of biological organisms. Genetic algorithm works with a population of individuals represented by chromosomes. Each chromosome is evaluated by its fitness value as computed by the objective function of the problem. The population undergoes transformation using three primary genetic operators – selection, crossover and mutation which form new generation of population. This process continues to achieve the optimal solution. Basic flowchart of genetic algorithm is illustrated in Fig. 1.

Generally all the optimization techniques are influenced by two important issues - exploration and exploitation. Exploration is used to investigate new and unknown areas in the search space and generate new knowledge. Exploitation makes use of the generated knowledge and propagation of the adaptations. Both techniques have their own merits and demerits. Both the terms are contradictory to each other and need to be balanced. In common view, exploration of search space is done by search operators in evolutionary algorithms and exploitation is done by selection. It has been observed in previous researches that any one technique is not enough to obtain best optimal solution, especially with large TSPs [3,4]. So, many researches are being carried out to combine two or more algorithms in order to improve performance and obtain better results.

Manuscript received May 27, 2012; revised June 26, 2012.

Rakesh Kumar is with Department of Computer Science & Applications, Kurukshetra University, Kurukshetra, Haryana, India (e-mail: rsgawal@gmail.com).

Jyotishree is with Department of Computer Science, Guru Nanak Girls College, Yamunanagar, Haryana, India (e-mail: jyotishreer@gmail.com).

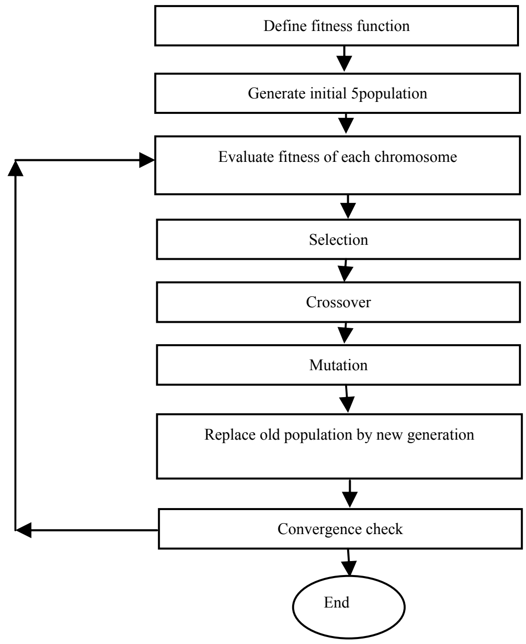

Fig. 1. Basic flowchart of genetic algorithm

In this paper, focus is to blend the two selection operators and generate a new selection operator to obtain perfect mix of exploration and exploitation. The blended selection operator shows exploratory nature initially and shifts to exploitation later. The paper is organized in the following sections. In section II, literature review is given on different researches using combination techniques related to this field. Different notations used throughout the paper are given in section III. Selection methods and their computation formulae are described in section IV. Algorithms related to selection methods under study in this paper are presented in section V. Implementation procedure and computational results are provided in section VI and concluding remarks are given in section VII.

II. Related Work

Holland showed that both exploration and exploitation are used optimally by genetic algorithm at the same time using k-armed bandit analogy [1]. This work is also described by David Goldberg [2]. It has been observed that due certain parameters, stochastic errors occur in genetic algorithms and this may lead to genetic drift [5,6]. In certain cases, selection operation gets biased towards highly fit individuals. This can be avoided by use of Rank Selection technique. Rank scaling ranks the individuals according to their raw objective value [2]. Another problem that arises with genetic algorithms is premature convergence which occurs when the population reaches a state where genetic operators can no longer produce offspring that outperforms their parents [7]. This would likely trap the search process in a region containing a non-global optimum and would further lead to loss of diversity.

Genetic algorithms and simulated annealing are widely used in search and optimization problems. Dan Adler proposed a method of hybridizing genetic algorithms with simulated annealing and replaced standard mutation and recombination operator by their simulated annealing variants – SAM and SAR [8]. The hybrid algorithm improved accuracy of genetic algorithm and exhibited more consistency. Modified operators improved convergence and speed of simulated annealing. Eiben and Schippers [9] surveyed different operators and reviewed different existing viewpoints on exploration and exploitation. They distinguished three levels at which exploration and exploitation occur. Selection was found to be source of exploitation and mutation and crossover were adjudged as source of exploration. Van Dijck etal proposed 2-stage GA based feature subset selection algorithm in which the correlation structure of the features was exploited [10]. Simulations on a real-case data set with correlated features showed that the 2-stage GA found better solutions in fewer generations compared to a standard GA.

Al jaddan et al. compared the roulette wheel selection GA (RWS) and ranked based roulette wheel selection GA (RRWS), by applying them on eight test functions from the GA literature [11]. They concluded that RRWS outperformed the conventional RWS in convergence, time, reliability, certainty, and more robustness. Tsenov proposed a combined way of using simulated annealing and genetic algorithm on telecommunications concentrator networks and obtained good performance [12]. Eiben et al. suggested that the selection pressure is an aggregated parameter determined collectively by the individuals in the population. They implemented their viewpoint in two different ways – Self adaptation and hybrid self adaptation [13]. They compared three genetic algorithms in their study - Simple GA (SGA) as benchmark, the GA with hybrid self adaptive tournament size (GAHSAT), and the GA with self adaptive tournament size (GASAT). They concluded that GAHSAT was very competitive and lead to 30-40% performance increase in terms of speed.

Wang et.al proposed a new hybrid of genetic algorithm and simulated annealing, referred to as GSA and then evaluated its performance against a standard set of benchmark functions [14]. Notably, there was remarkable improvement in performance of Multi-niche crowding PGSA and normal PGSA over conventional parallel genetic algorithm. Liu et.al. proposed a new heuristic algorithm for classical symmetric TSP and tested its performance against benchmark TSP problems [15]. They presented overlapped neighbourhood based local search algorithm to solve TSP and concluded that the proposed algorithm is superior in terms of average deviation and smallest deviation from optimal solutions.

In order to improve the balance between the exploration and exploitation in differential evolution algorithm, Sa Angela et al. proposed a modification of the selection that was successful in avoiding entrapment in local optima and could be helpful in many real world optimization problems [16]. R.Thamilselvan and P.Balasubramanie presented a Genetic Tabu search Algorithm (GTA) for TSP and compared with Tabu search [17]. They concluded that GTA is better than GA and TS. Elhaddad and Sallabi proposed a new Hybrid Genetic and Simulated Annealing Algorithm (HGSAA) to solve the TSP [18]. The proposed hybrid algorithm combined both the SA and GAs, in order to help each other overcome their problems to obtain the best results in the shortest time. HGSAA improved the convergence rate of the algorithm with better solutions to TSP compared with other algorithms.

III. Notations

Some of the symbols used in these algorithms are listed below:

- ngen → total number of generations
- nogen → current number of generation
- N → total population size
- PS → Problem Size, in terms of number of cities
- RWS → Roulette Wheel Selection
- RS → Rank Selection
- PBS → Proposed Blended Selection
- $FRW_{i,j}$ → fitness of j^{th} individual in i^{th} generation for roulette wheel selection
- $r_{i,j}$ → rank of j^{th} individual in i^{th} generation for rank selection
- $rsum_i$ → sum of ranks in i^{th} generation
- $FX_{i,j}$ → fitness of j^{th} individual in i^{th} generation for proposed blended selection
- mpool → number of chromosomes in mating pool
- CN → Generation number after which there was no change in population
- $\overline{FX_i}$ → Average Fitness of the population in i^{th} generation in Proposed Blended Selection
- $\overline{FRW_i}$ → Average Fitness of the population in i^{th} generation in Roulette Wheel Selection
- $\overline{FR_i}$ → Average Fitness of the population in i^{th} generation in Rank Selection
- F_{best} → Best Fitness value i.e. minimum distance of the route computed in all generations
- $\overline{FR}$ → Average Fitness of the population in all generations in Rank Selection
- $\overline{FRW}$ → Average Fitness of the population in all generations in Roulette Wheel Selection
- $\overline{FX}$ → Average Fitness of the population in all generations in Proposed Selection

IV. Selection

Selection is the first genetic operation in the reproductive phase of genetic algorithm. Its purpose is to choose the fitter individuals in the population that will create offsprings for next generation, commonly known as mating pool. The mating pool thus selected takes part in further genetic operations, advancing the population to the next generation and hopefully close to the optimal solution. Selection of individuals in the population is fitness dependent and is done using different algorithms [19]. Selection chooses more fit individuals in analogy to Darwin's theory of evolution – survival of fittest [20]. Too strong selection would lead to sub-optimal highly fit individuals and too weak selection may result in too slow evolution [21]. There are many methods in selecting the best chromosomes. Some are roulette wheel selection, rank selection, steady state selection and many more. The paper would focus on first two approaches and compare them with proposed selection approach.

A. Roulette Wheel Selection

Roulette wheel is the simplest selection approach. In this method all the chromosomes (individuals) in the population are placed on the roulette wheel according to their fitness value [2,19,22]. Each individual is assigned a segment of roulette wheel. The size of each segment in the roulette wheel is proportional to the value of the fitness of the individual - the bigger the value is, the larger the segment is. Then, the virtual roulette wheel is spinned. The individual corresponding to the segment on which roulette wheel stops are then selected. The process is repeated until the desired number of individuals is selected. Individuals with higher fitness have more probability of selection. This may lead to biased selection towards high fitness individuals. It can also possibly miss the best individuals of a population. There is no guarantee that good individuals will find their way into next generation. Roulette wheel selection uses exploitation technique in its approach.

The average fitness of the population for ith generation in roulette wheel selection is calculated as

$$\overline{FRW_{i,j}} = \frac{\sum_{j=1}^{N} FRW_j}{N} \tag{1}$$

where i varies from 1 to ngen and j varies from 1 to N. Therefore, the probability for selecting the j^{th} string is

$$PRW_j = \frac{FRW_j}{\sum_{j=1}^{N} FRW_j} \tag{2}$$

where N is the population size and FRW_j is the fitness of individual j.

B. Rank Selection

Rank Selection sorts the population first according to fitness value and ranks them. Then every chromosome is allocated selection probability with respect to its rank [23]. Individuals are selected as per their selection probability. Rank selection is an explorative technique of selection. Rank selection prevents too quick convergence and differs from roulette wheel selection in terms of selection pressure. Rank selection overcomes the scaling problems like stagnation or premature convergence. Ranking controls selective pressure by uniform method of scaling across the population. Rank selection behaves in a more robust manner than other methods [24,25].

In Rank Selection, sum of ranks is computed and then selection probability of each individual is computed as under:

$$rsum_i = \sum_{i=1}^{N} r_{i,j} \tag{3}$$

where i varies from 1 to ngen and j varies from 1 to N.

$$PRANK_i = \frac{r_{i,j}}{rsum_i} \tag{4}$$

C. Proposed Annealed Selection

The proposed selection approach is to move the selection criteria from exploration to exploitation so as to obtain the perfect blend of the two techniques. In this method, fitness value of each individual is computed. Depending upon the current generation number of genetic algorithm, selection pressure is changed and new fitness contribution, $X_{i,j}$ of each individual is computed. Selection probability of each individual is computed on the basis of $X_{i,j}$. As the generation of population changes, fitness contribution changes and selection probability of each individual also changes.

The proposed blended selection operator computes fitness of individual depending on the current number of generation as under:

$$FX_i = \frac{FRW_i}{(ngen+1) - nogen} \tag{5}$$

The probability for selecting the i^{th} string is

$$PX_i = \frac{FX_i}{\sum_{i=1}^{N} FX_i} \tag{6}$$

V. Algorithms

Algorithms of three methods of selection to be compared in the paper are given below. Here, c_i is variable storing cumulative fitness and r is random number generated between given interval.

A. Roulette wheel selection

1. Set l=1, j=1, i=nogen
2. While l <= mpool
   Begin
   a) While j<=N
      Begin
         Compute $FRW_{i,j}$
      End
   b) Set j=1, S=0
   c) While j<=N
      Begin
         Compute $S=S+FRW_{i,j}$
      End
   d) Generate random number **r** from interval (0,S)
   e) Set j=1, S=0
   f) While j<=N
      Begin
         Calculate $c_j=c_{j-1}+FRW_{i,j}$
         If $r<=c_j$, Select the individual j
      End
   g) l=l+1
   End

B. Rank Selection

1. Set l=1, j=1, i=nogen
2. While l <= mpool

Begin

a) While j<=N
   Begin
      Compute $rsum_{,i}$
   End
b) Set j=1
c) While j<=N
   Begin
      Compute $PRANK_j$
   End
d) Generate random number **r** from interval (0,rsum)
e) Set j=1, S=0
f) While j<=N
   Begin
      Calculate $c_j = c_{j-1} + PRANK_j$
      If r<=cj, Select the individual j
   End
g) l=l+1

End

C. Proposed Annealed selection

1. Set l=1, j=1, i=nogen
2. While l <= mpool

Begin

a) While j<=N
   Begin
      Compute $FX_{i,j}$
   End
b) Set j=1, S=0
c) While j<=N
   Begin
      Compute $S=S+FX_{i,j}$
   End
d) Generate random number **r** from interval (0,S)
e) Set j=1, S=0
f) While j<=N
   Begin
      Calculate $c_j = c_{j-1} + FX_{i,j}$
      If r<=cj, Select the individual j
   End
g) l=l+1

End

VI. Implementation and Observation

In this paper, MATLAB code has been developed to assess the performance of genetic algorithm by using three different selection techniques on the same population. for its implementation using the same initial population. Except selection criteria, all other factors affecting the performance of genetic algorithm are kept constant. The code considers the Travelling Salesman Problem (TSP) which is a classical combinatorial optimization problem. The problem is to find the shortest tour or Hamiltonian path through a set of N vertices so that each vertex is visited exactly once [26].

The problem is solved under following assumptions:

- Each city is connected to every other city.
- Each city has to be visited exactly once,
- The salesman's tour starts and ends at the same city.

The TSP problem have been considered for four different population sizes – 10 cities, 20 cities, 50 cities and 100 cities. The solution was run for 100 generations in each case. Firstly, the rank selection is applied and then the roulette wheel selection followed by the implementation of proposed blended selection operator on the same population. Convergence point of population was noted when no further changes occurred in the generation.

Average and minimum fitness in each generation is computed over 100 generations and plotted to compare the performance of three approaches. Fig. 2 depicts the comparison of $\overline{FRW_i}, \overline{FR_i}, \overline{FX_i}$ and Fig. 3 depicts the comparison of F_{best} in three different selection methods. Table I lists the detailed data for four different problem sizes and various parameters to analyze performance of the three methods. Comparison of $\overline{FRW_i}, \overline{FR_i}, \overline{FX_i}$ is presented in Fig. 4 and comparison of F_{best} in Fig. 5.

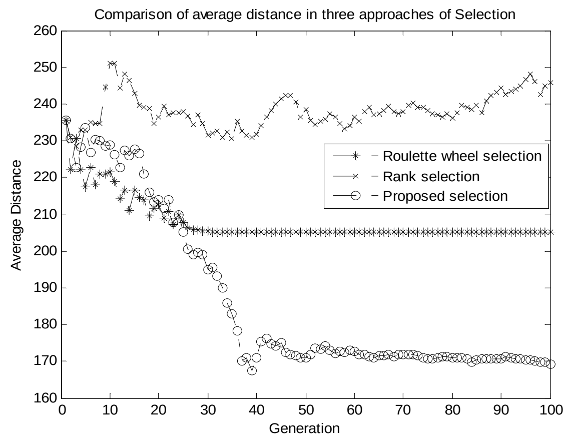

Fig. 2. Comparison of average distance of TSP tour

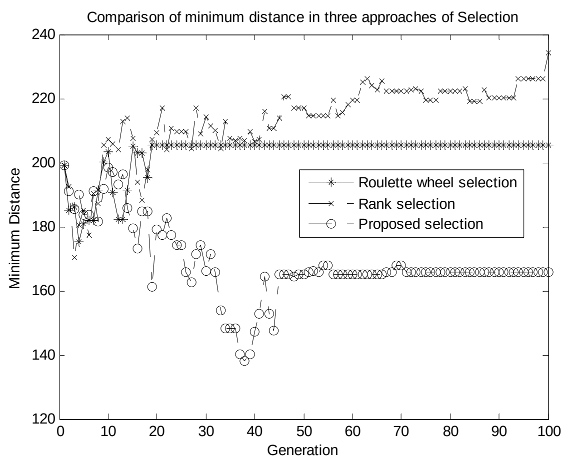

Fig. 3. Comparison of minimum distance of TSP tour

TABLE I: COMPARISON OF THREE SELECTION APPROACHES

PS	RS			RWS			PBS		
	F_{best}	CN	$\overline{FR}$	F_{best}	CN	$\overline{FRW}$	F_{best}	CN	$\overline{FX}$
100	424.78	97	525.24	424.78	15	520.21	417.19	96	457.14
50	181.94	98	253.15	176.70	30	220.42	163.43	94	209.99
20	61.62	57	93.45	62.29	28	85.31	56.71	41	71.07
10	25.83	14	43.61	22.92	9	26.23	15.37	14	17.69

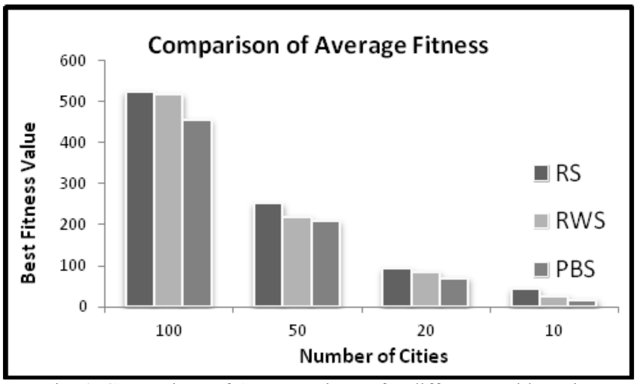

Fig. 4. Comparison of Average Fitness for different problem sizes

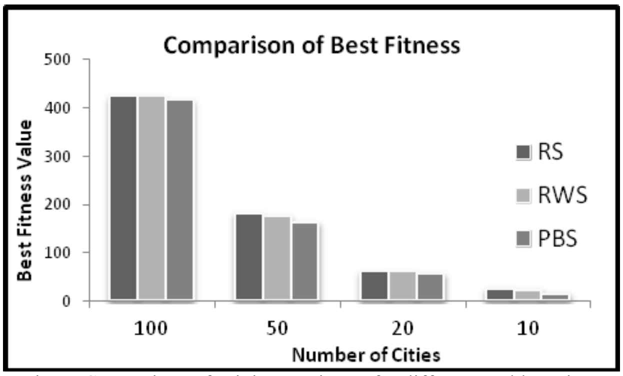

Fig. 5. Comparison of Minimum Fitness for different problem sizes

It was observed that the RWS had more of exploitation approach and found better chromosomes in early runs of generation and converged earlier than RS. On the contrary, RS had more of exploratory nature and kept on exploring new solutions. In case of PBS, $\overline{FX_i}$ and PX_i reduced gradually with increasing number of generation. In early runs of generation, the method depicted exploration and with increasing generation, it had exploiting nature and converged to find the better solution. It is clear from the Fig. 1 and Fig. 2 that PBS performs better than the other two selection methods. Further, figure 4 and figure 5 show the comparison of results for different number of cities. It has been found that with increasing problem size, problem did not converge prematurely and PBS gave better performance in each case.

VII. CONCLUSION

In this paper, a blended selection operator - PBS is proposed having balanced tradeoff between exploration and exploitation. The performance of PBS selection operator is compared with RS and RWS technique on standard TSP problem. RWS performed like nature selecting the most fit individuals. RS did more of exploration and maintained diversity in population. PBS had both the features and outperformed the other two techniques. Its performance was dependent on the current number of generation. In early generations, there is less pressure on selection, so it had exploratory nature. As the number of generation increased, selection pressure also increased and exploratory nature gradually turned into exploiting nature. It is evident from above results that performance of PBS is superior over than that of RS and RWS. Further research in this area is intended to incorporate factors influencing performance of genetic algorithms and knowledge based operations.

REFERENCES

[1] J. Holland, *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, 1975.

[2] D. E. Goldberg, *Genetic algorithms in search, optimisation, and machine learning,* Addison Wesley Longman, Inc., ISBN 0-201-15767-5, 1989.

[3] P. Merz and B. Freisleben, “Genetic Local Search for the TSP: New results”, Proceedings of IEEE International Conference on Evolutionary Computation, IEEE Press, pp 159-164, 1977.

[4] S. Ray, S. Bandyopadhyay and S.K. Pal, “Genetic operators for combinatorial optimization in TSP and microarray gene ordering”, SpringerScience + Business Media, LLC, 2007.

[5] D. E. Goldberg and P. Segrest, “Finite Markov chain analysis of genetic algorithms”, Proceedings of the the Second International Conference on Genetic Algorithms, Lawrence Erlbaum Associates, pp 1-8, 1987.

[6] L. Booker, *Improving search in genetic algorithms*, Genetic Algorithms and Simulated Annealing, Pitman, chapter 5, pp 61-73, 1987.

[7] D. Fogel , “An introduction to simulated evolutionary optimization”, IEEE Trans. Neural Networks 5 (1), pp 3-14, 1994.

[8] Dan Adler, “Genetic Algorithms and Simulated Annealing: A Marriage Proposal”, IEEE International Conference on Neural Networks 1993, San Francisco (CA), 1993.

[9] A. E. Eiben and,C.A.Schippers, *On evolutionary exploration and exploitation*, Fundamenta Informaticae, 35 IOS Press, pp 1-16, 1998.

[10] G. Van Dijck, M. M. Van Hulle and M.Wevers, “Genetic Algorithm for Feature Subset Selection with Exploitation of Feature Correlations from Continuous Wavelet Transform: a real-case Application”, International Journal of Information and Mathematical Sciences 1:4 2005 pp 233-237, 2005.

[11] O. Al *jaddan*, L. Rajamani and C. R. Rao, “*Improved Selection Operator for GA”,* Journal of Theoretical and Applied Information Technology, pp 269–277, 2005.

[12] A. Tsenov, “Simulated Annealing and Genetic Algorithm in Telecommunications Network Planning”, International Journal of Information and Mathematical Sciences 2:4, pp 240-245, 2006.

[13] A. E. Eiben, M. E. Schut and A. R. de Wilde, “Boosting Genetic Algorithms with Self-Adaptive Selection”, IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, pp 1584-1589, 2006.

[14] Z. G. Wang, M. Rahman, Y. S. Wong and K.S.Neo, “Development of Heterogeneous Parallel Genetic Simulated Annealing Using Multi-Niche Crowding”, International Journal of Information and Mathematical Sciences 3:1, pp 55-62, 2007.

[15] S. B. Liu, K. M. Ng and H. L. Ong, “A New Heuristic Algorithm for the Classical Symmetric Travelling Salesman Problem”, International Journal of Computational and Mathematical Sciences, Volume 1, Number 4, pp 234-238, 2007.

[16] A. R. Sa Angela, A. O. Andrade, A. B. Soares and S.J. Nasuto, “Exploration vs. Exploitation in Differential Evolution”, Volume 11: Proceedings of the AISB 2008 Symposium on Swarm Intelligence Algorithms and Applications, ISBN 1 902956 70 2, pp 57-63, 2008.

[17] R. Thamilselvan and P.Balasubramanie, “A Genetic Algorithm with a Tabu Search (GTA) for Travelling Salesman Problem”, International Journal of Recent Trends in Engineering, Issue I, Vol I, pp 607 – 610, 2009.

[18] Y. Elhaddad and O. Sallabi, “A New Hybrid Genetic and Simulated Annealing Algorithm to Solve the Traveling Salesman Problem, Proceedings of the World Congress on Engineering 2010, Vol. I, ISBN: 978-988-17012-9-9, ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online), pp 11-14, 2010.

[19] D. E. Golberg and K. Deb, “A comparative analysis of selection schemes used in genetic algorithms”, Foundations of Genetic Algorithms, San Mateo, CA, Morgan Kaufmann, pp 69-93, 1991.

[20] D. Fogel, *Evolutionary Computation*, IEEE Press, 1995.

[21] Melanie Mitchell, *An Introduction to genetic algorithm,*. Prentice Hall of India, New Delhi, ISBN-81-203-1358-5, 1996.

[22] K. A. De Jong, *An Analysis of the behavior of a class of genetic adaptive systems* (Doctoral dissertation, University of Michigan) Dissertation Abstracts International 36(10), 5140B University Microfilms No. 76/9381, 1975.

[23] J. E. Baker, "Adaptive selection methods for genetic algorithms", Proceedings of an International Conference on Genetic Algorithms and their applications, pp 101-111, 1985.

[24] D. Whitley, "The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best", Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann, pp 116-121, 1989.

[25] T. Back and F.Hoffmeister, "Extended Selection Mechanisms in Genetic Algorithms",ICGA4, pp. 92-99, 1991.

[26] Perling,M.and GeneTS , A Relational-Functional Genetic Algorithm for the Travelling Salesman Problem", Technical Report, Universitat Kaiserslautern, ISSN 09460071, 1997.

Dr. Rakesh Kumar obtained his B.Sc. Degree, Master's degree – Gold Medalist (Master of Computer Applications) and PhD (Computer Science & Applications) from Kurukshetra University, Kurukshetra. Currently, he is Associate Professor in the Department of Computer Science and Applications, Kurukshetra University, Kurukshetra, Haryana, India. His research interests are in Genetic Algorithm, Software Testing, Artificial Intelligence, Networking. He is a Senior member of International Association of Computer Science and Information Technology (IACSIT).

Mrs. Jyotishree is working as Assistant Professor in the Department of Computer Science and Applications, Guru Nanak Girls College, Santpura, Yamunanagar. She is M.Sc – Gold Medalist from Guru Jambheshwar University, Hisar and is pursuing PhD from Kurukshetra University. She is member of International Association of Computer Science and Information Technology (IACSIT). She has presented papers in 3 International and 5 National Conferences. Her research interests are Genetic Algorithms, Soft Computing methods and Networking.