Chapter 6: SELECTION

6.1 Introduction

Selection is the first genetic operation in the reproductive phase of genetic algorithm. The objective of selection is to choose the fitter individuals in the population that will create offsprings for the 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. In other words, Selection is the process of choosing breeding stock or parents from a population. As the generations pass, the members of population should get fitter and fitter. Individuals from the mating pool are used to generate new offsprings, with the resulting offspring forming the basis of next generation. So it is desirable that the mating pool should have good individuals. Selection operator works at the level of chromosomes. The key idea of selection operator is to give preference to better individuals by allowing them to pass on their genes to the next generation and prohibit the entrance of worst fit individuals into the next generation. The goodness of each individual depends on its fitness. Fitness value is determined by an objective function. Selection of individuals in the population is fitness dependent and is done using different algorithms [Goldberg *et al.* 1991]. Some are roulette wheel selection, rank selection, steady state selection and many more.

Selection acts as driving force in a genetic algorithm by directing the genetic search towards promising regions in the search space. Selection operator emulates phenomena and processes in nature. Selection chooses more fit individuals in analogy to Darwin's theory of evolution – survival of fittest [Fogel 1995]. All the individuals have a chance of being selected into the mating pool, but there are chances that an individual in the population can be selected more than once depending upon its fitness. Selection schemes are characterised by selection pressure, selection variance and loss of diversity. They primarily determine the convergence characteristics of genetic algorithms. Selection has to be balanced. Too strong selection means suboptimal highly fit individuals will take over the population reducing the diversity and too weak selection will result in too slow evolution [Mitchell 1996].

Broadly, Selection schemes can be classified into two broad categories – namely, proportionate selection and ordinal-based selection. Proportionate selection picks out individuals based upon their fitness values relative to the fitness of the other individuals in the population. It is generally more sensitive to selection pressure. Hence, a scaling function is employed for redistributing the fitness range of the population in order to adapt to the selection pressure. Roulette wheel selection, stochastic remainder selection, and stochastic universal selection are some selection schemes grouped under this category. Ordinal-based selection scheme selects individuals not upon their raw fitness, but upon their rank within the population. In ordinal selection, the selection pressure should be independent of the fitness distribution of the population and solely based upon the relative ordering i.e. ranking of the population. Tournament selection, selection, truncation selection, and linear ranking selection schemes are included in ordinal selection.

Goldberg and Deb grouped selection methods in to four categories: Proportionate, Ranking, Tournament, Steady state selection. Other methods were grouped under these categories. Detailed classification of selection schemes is shown in the following figure 6.1:

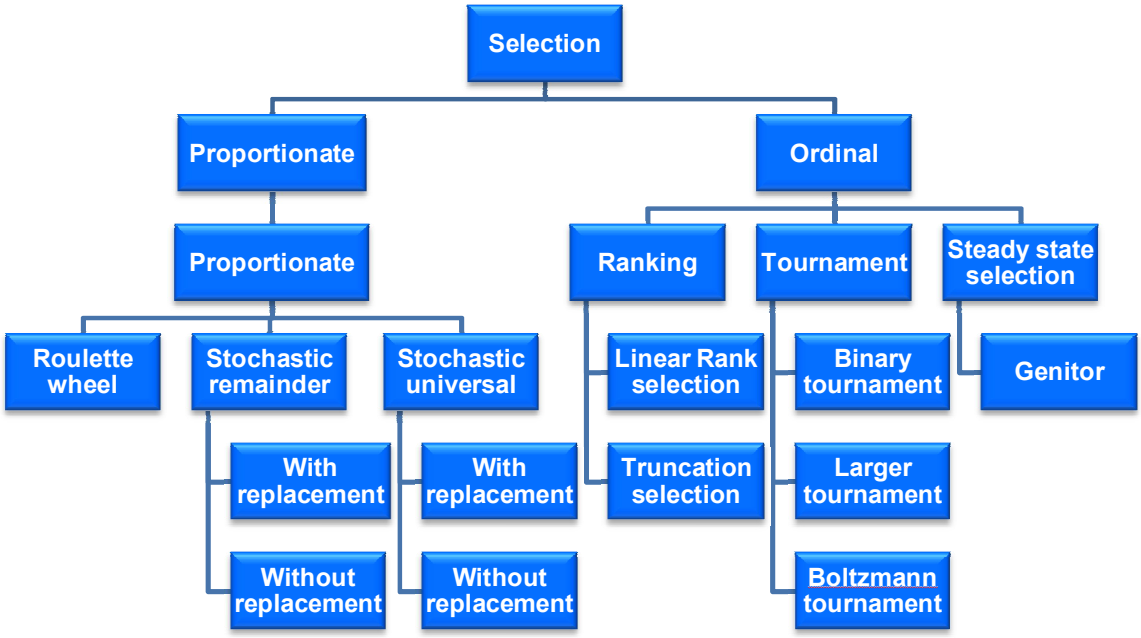


Figure 6.1 Classification of Selection Schemes

6.2 Exploration vs. Exploitation

Generally all the optimisation techniques are influenced by two important issues - *exploration* and *exploitation*. Exploitation refers to the tendency of the algorithm to steer its search direction by previously obtained information [Thierens *et al.* 1994]. Exploitation makes use of the generated knowledge and propagation of the adaptations. It means that during the solution searching, algorithm uses information obtained in the past in order to determine smaller regions that are promising for further search. Exploitation operations often incorporate small changes into already tested individuals leading to new, very similar solution candidates or try to merge building blocks of different promising individuals. Exploration is a metaphor for the procedure which allows search operations to find novel and maybe better solution structures. Exploration investigates new and unknown areas in the search space and generates new knowledge. Exploration is a procedure that obtains new information by visiting new regions in the search space in order to find promising points or sub regions. In contrast to exploitation, exploration includes jumps into the unknown. 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. The trade-off between exploitation and exploration is mainly determined by the selective pressure created by the selection operator. Pure random search is good at exploration, but does no exploitation whereas hill climbing is good at exploitation and does no exploration [Beasley 1993a]. It has been observed in previous researches that any one technique is not enough to obtain best optimal solution, especially with large TSPs [Merz *et al.* 1977, Ray *et al.* 2007]. So, many researches are being carried out to combine two or more algorithms in order to improve performance and obtain better results. Optimisation algorithms that favour exploitation over exploration have higher convergence speed but run the risk of not finding the optimal solution and may get stuck at a local optimum. Generally, optimisation algorithms should employ at least one search operation of explorative character and at least on which is able to exploit good solutions further. Holland showed that a genetic algorithm combines both exploration and exploitation at the same time in an optimal way [Holland 1975].

6.3 Parameters for Selection

- Selection pressure: It is the degree to which the individuals in the mate pool needed to be better. It can be defined as the probability of best individual being selected compared to the average probability of selection of all individuals. Low selection pressure leads to slow convergence rate and long time will be needed to find the optimal solution. The selection pressure is the degree to which the better individuals are favoured. The higher the selection pressure, the more the better individuals are favoured. This selection pressure drives the genetic algorithm to improve the population fitness over the successive generations. The convergence rate of genetic algorithm is largely determined by the magnitude of the selection pressure, with higher selection pressures resulting in higher convergence rates.
- Selection Variance: It is the expected variance of the fitness distribution of the population after applying the selection method to the normalised Gaussian distribution.
- Selection Intensity: It is the expected average fitness value of the population after applying a selection method on population whose fitness is distributed according to unit normal distribution [Zhang *et al.* 1995]. It measures the magnitude of selection pressure provided by selection scheme.
- Loss of Diversity: It is the proportion of individuals of a population that is not selected during the selection phase.
- Bias: It is the absolute difference between an individual's normalised fitness and its expected probability of reproduction.
- Spread: It is the range of possible values for the number of offsprings of an individual.

6.4 Description of Selection Schemes

6.4.1 **Proportionate Selection**: It selects the individual from the population according to the fitness of an individual. Proportionate selection is carried out in many ways - Roulette Wheel Selection, Deterministic Selection, Stochastic Remainder Selection and Stochastic Universal Selection.

6.4.1.1 Roulette Wheel Selection

Roulette wheel is the simplest and traditional stochastic selection approach proposed by Holland. It is categorised under proportionate selection as it selects the individuals based on a probability proportional to the fitness. The principle of roulette selection is a linear search through a roulette wheel with the slots in the wheel weighted in proportion to the individual's fitness values. All the chromosomes (individuals) in the population are placed on the roulette wheel according to their fitness value [De Jong 1975, Goldberg 1989, Goldberg *et al.* 1991]. 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, is 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.

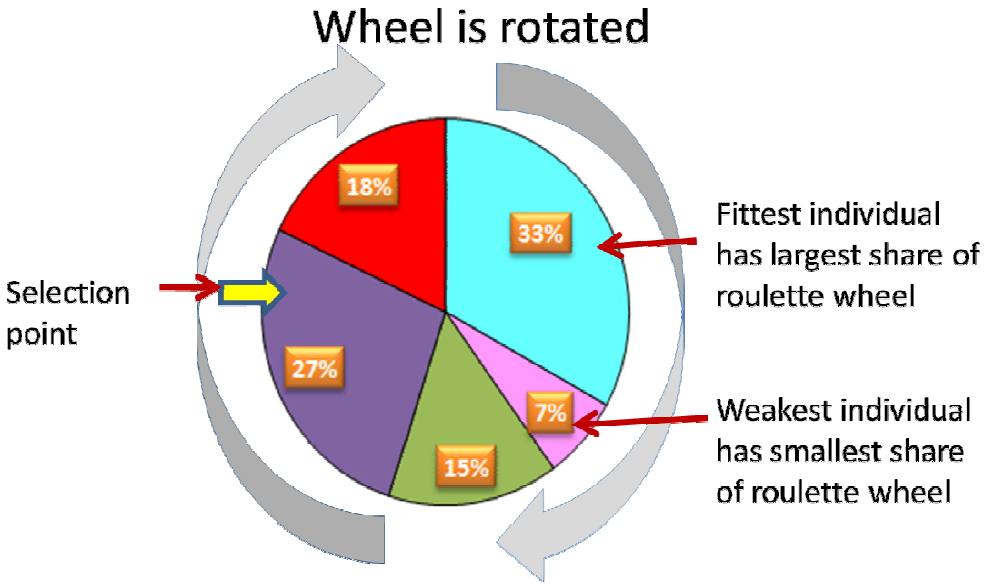


Figure 6.2 Fitness Distribution on Roulette Wheel

Roulette wheel selection is easier to implement but is noisy. The rate of evolution depends on the variance of fitness's in the population. The average fitness of the population for ith generation in roulette wheel selection is calculated as

$$\overline{\mathrm{FRW}_{\mathrm{i}}} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{FRW}_{\mathrm{i}}}{\mathrm{N}} \qquad \text{(Eq.ix)}$$

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

Therefore, the probability for selecting the j^{th} string is

$$\mathrm{PRW}_{\mathrm{j}} = \frac{\mathrm{FRW}_{\mathrm{j}}}{\sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{FRW}_{\mathrm{j}}} \qquad \text{(Eq.x)}$$

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

Selection Intensity is calculated as:

$$I_{RW} = \frac{\bar{\sigma}}{N} \qquad \text{(Eq.xi)}$$

where σ is mean variance of fitness values of population before selection.

In roulette wheel selection, if the search for the location of the chosen slot is performed using linear search from beginning of the list, each selection requires O(n) steps, because on an average half the list will be searched. Overall time complexity of roulette wheel selection is $O(n^2)$ steps because it requires n spins to fill the population in a generation. Roulette wheel selection can also be carried out using binary search, but it requires additional memory locations. In this case, overall time complexity reduces to $O(n \log n)$ because binary search requires $O(\log n)$ steps per spin and n spins.

Algorithm of Roulette Wheel Selection

```
Set l=1, j=1, i=nogen
While l <= mpool
Begin
      While j<=N
       Begin
            Compute FRW_{i,j}
       End
      Set j=1, S=0
      While j<=N
       Begin
```

```
        Compute S=S+FRW_{i,j}
      End
  Generate random number r from interval (0,S)
  Set j=1, S=0
  While j<=N
      Begin
        Calculate c_j=c_{j-1}+FRW_{i,j}
        If r<=cj, Select the individual j
  End
  l=l+1
End
```

6.4.1.2 Stochastic Remainder Selection

Stochastic Remainder Selection operates on the concept that proportionate fitness of each chromosome should be reflected in that chromosome's incidence in mating pool [Grefenstette *et al.* 1989]. Each chromosome in the population has a probability of selection based on its relative fitness. In Stochastic Remainder Selection Mechanism, relative fitness of a chromosome is calculated as the fitness associated with an individual divided by average fitness, say $m_i = f_i/f_{avg}$. Integer part of m_i is used to select one parent deterministically and then uses roulette selection on the remaining fractional part to fill up the free places in the mating pool. For example, if the value of m_i of an individual is 2.3, that individual is listed twice as a parent because the integer part is 2. After parents have been assigned according to the integer parts of the scaled values, the rest of the parents are chosen stochastically. The probability that a parent is chosen in this step is proportional to the fractional part of its scaled value. This technique provides greatest probability of selection to most fit members of the population.

$$Pselect_i = f_i/f_{avg} \qquad \text{(Eq.xii)}$$

Expected occurrence in mating pool is calculated as:

$$e_i = pselect_i \times n \qquad \text{(Eq.xiii)}$$

This method has two variations: with replacement and without replacement

i) Stochastic Remainder Selection with Replacement

If stochastic remainder selection is done with replacement, then the remainders are used to size the slots of a roulette wheel selection process. The fractional parts of the

expected occurrence value are used to calculate weights in roulette wheel selection and then used to fill the remaining population slots in the mating pool. The algorithm takes on the complexity of the roulette wheel i.e. $O(n^2)$ in this case because O (n) of the individuals is likely to have fractional parts to their m values.

ii) Stochastic Remainder Selection without Replacement

If stochastic remainder selection is done without replacement, the fractional parts of the expected occurrence value are treated as probabilities. Remaining slots are filled on the flip of a coin that determines whether the remainder part receives another copy or not. Time complexity for the algorithm without replacement is O(n) because the deterministic assignment requires only a single pass. It is an improved method of roulette wheel selection.

6.4.1.3 Stochastic Universal Selection

Stochastic universal selection was developed by Baker [Baker 1987]. It is a single phase selection algorithm with minimum spread and zero bias. It is performed by sizing the slots of a weighted roulette wheel, placing equally spaced markers along the outside of the wheel, and spinning the wheel once; the number of copies an individual receives is then calculated by counting the number of markers that fall in its slot. The algorithm is O (n), because only a single pass is needed through the list after the sum of the function values is calculated.

It exhibits less variance than repeated calls to roulette wheel selection. The individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness exactly as in roulette-wheel selection. Here equally spaced pointers are placed over the line, as many as there are individuals to be selected. Stochastic universal selection ensures a selection of offspring, which is closer to what is deserved than roulette wheel selection. Stochastic Universal Selection can be used to make any number of selections. It is preferred in situations where more than one sample is to be drawn from the distribution.

Algorithm of Stochastic Universal Selection

```
Set l=1, j=1
μ=F/N
While l <=  mpool
```

```
begin
        generate random number r between interval (0, μ)
        While  r<= fitness[i]
        begin
             select[l]=parent[i]
             r=r+1/μ
        end
        l=l+1
 end
```

6.4.2 Rank Selection

Rank Selection sorts the population first according to fitness value and ranks them. Rank N is assigned to the best individual and rank 1 to the worst individual. Then every chromosome is allocated selection probability with respect to its rank [Baker 1985]. 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 [Whitley 1989, Back *et al.* 1991].

Algorithm of Rank Selection

```
Set l=1, j=1, i=nogen
While l <= mpool
Begin
   While j<=N
    Begin
          Compute rsum,i
     End
      Set j=1
      While j<=N
      Begin
          Compute PRANKj
     End
     Generate random number r from interval (0,rsum)
     Set j=1, S=0
     While j<=N
     Begin
          Calculate cj=cj-1+PRANKj
          If r<=cj, Select the individual j
      End
      l=l+1
End
```

6.4.2.1 Linear Rank Selection

Linear Rank Selection was proposed by Baker. He stated that each individual in the population is ranked in increasing order of fitness, from 1 to n. the user chooses the expected value of 'max' of an individual with rank n with max >=0 . Linear ranking assigns a selection probability to each individual that is proportional to the individual's rank. Linear ranking can be implemented by specifying a single parameter Prank which is the expected number of offsprings to be allocated to the best individual during each generation.

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

$$rsum_i = \sum_{i=1}^{N} r_{i,j} \qquad \text{(Eq.xiv)}$$

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

$$\text{PRANK}_i = r_{i,j} / \text{rsum}_i \qquad \text{(Eq.xv)}$$

Expected value of each individual x_i in the population at time t is given by

$$ExpVal(x_i, t) = Min - (Max - Min)\frac{Rank(x_i,t)-1}{n-1} \qquad \text{(Eq.xvi)}$$

where Min is the expected value of the individuals with rank 1. Given constraints Max ≥ 0 and $\sum_i ExpVal(x_i, t) = n$. It is required that 1 ≤ Max ≤ 2 and Min = 2- Max [mit98].

Probability of an individual to be selected=

$$p_i = \frac{1}{N}\left(\eta^- + (\eta^+ - \eta^-)\frac{i-1}{N-1}\right) \qquad \text{(Eq.xvii)}$$

where η/N is probability of worst individual to be selected.

Selection intensity is calculated as:

$$I_R(\eta^-) = (1 - \eta^-)\frac{1}{\sqrt{\pi}} \qquad \text{(Eq.xviii)}$$

Rank Selection is a two step process. First the list is sorted and then individuals are ranked. So time complexity calculations consider both as separate steps. Assuming the best sorting technique is adopted, rank selection has time complexity O(*n log n*).

6.4.2.2 Non Linear Ranking

It uses non linear distribution [Pohlheim 1995]. Non-linear ranking permits higher selective pressures than the linear ranking method [Weblink17]. Non linear ranking assigns selection probabilities that are based on each individual's rank but are not proportional to the rank. Eg: selection probability may be proportional to square of the rank.

6.4.2.3 Truncation Selection

In truncation selection the candidate solutions are ordered by fitness, and some proportion, p, (e.g. p=1/2, 1/3, etc.), of the fittest individuals are selected and reproduced 1/p times. Truncation selection is less sophisticated than many other selection methods, and is not often used in practice. Only the individuals above the threshold T are selected as parents. T indicates the proportion of population to be selected as parents and takes values ranging from 50% - 10%. Individuals below threshold do not produce offsprings [Weblink17].

Its overall time complexity is O(n log n) as sorting is involved in its implementation. It is used in Muhlenbein's breeder genetic algorithm[Muhlenbein *et al.* 1993].

Selection Intensity = $I_{Tr}(T) = \frac{1}{T}.\frac{1}{\sqrt{2\pi}}\ e^{-\frac{{f_c}^2}{2}}$ (Eq.xix)

Selection Variance= $V_{Tr}(T) = 1 - I_{Tr}(T).(I_{Tr)} - f_c)$ (Eq.xx)

Algorithm of Truncation Selection

Input : Population P, Truncation Threshold T □ 0,1

```
Truncation(P, T, N)
        Sort(P) according to fitness with worst individual at first position
        For i=1 to N do
        Begin
                R= random((1-T).N,N)
                Select=P( r)
        End
Return
```

6.4.3 Tournament Selection

Tournament selection was stated by Goldberg, Korb and Deb [Goldberg *et al.* 1989a]. In tournament selection, number of chromosomes, say t, is randomly selected from the population. Here, t is tournament size that refers to number of randomly selected individuals. Tournament selection selects the best individual from this group (tournament) as parent. Tournament selection strategy provides selective pressure by holding a tournament competition among t individuals. The best individual from the tournament is the one with the highest fitness, among s individuals. Winner of tournament competition is then inserted into the mating pool. The tournament competition is repeated until the mating pool for generating new offspring is filled [Blickle *et al.* 1995]. The mating pool comprising of the tournament winner has higher average population fitness. The fitness difference provides the selection pressure, which drives Genetic algorithm to improve the fitness of the succeeding generations. Tournament selection is an efficient method and is more amenable to parallel implementation. Increased selection pressure can be provided by simply increasing the tournament size ,t, as the winner from a larger tournament will on an average have a higher fitness than the winner of a smaller tournament. This method is not very useful when we use large population because we will need a lot of N time to search every time on new element from selected group randomly.

Time complexity of tournament selection is O(n) because each competition requires selection of constant number of individuals from the population and there are n such competitions required to fill a generation. In this method, no sorting of population is required.

Selection intensity=

$$I_{To}(t) \approx \sqrt{2(\ln t) - \ln(\sqrt{4.14 \ln t})} \qquad \text{(Eq.xxi)}$$

Selection variance

$$V_{To}(t) \approx \sqrt{\frac{2.05+t}{3.14t^{3/2}}} \qquad \text{(Eq.xxii)}$$

Algorithm of Tournament Selection:

P is population, t is tournament size

For cur=1 to mpoolno

Begin

Pick t individuals randomly from P

Select best of t individuals depending on their fitness value and store in sl

```
        mpool(cur)=sl
        cur=cur+1
End
```

6.4.3.1 Binary Tournament Selection

Binary tournament selection is variant of tournament selection. In this method, two individuals are chosen at random and the better of the two individuals is selected with fixed probability p, $0.5 < p \sim 1$.

6.4.3.2 Large Tournament Selection

In this case, size of each tournament is very large. This would increase selection pressure and the individual selected in each tournament competition would be fitter than other individuals.

6.4.3.2 Boltzmann Tournament Selection

Boltzmann Tournament Selection was motivated by the process of simulated annealing [Kirkpatrick *et al.* 1983]. Simulation annealing is an optimisation method that simulates the process of slow cooling of molten metal to achieve the minimum function value in a minimization problem. Concept of Boltzmann probability distribution was introduced to simulate the cooling phenomenon by controlling the temperature- continuously varying parameter. Boltzmann tournament selection was developed to carry over Boltzmann distributions and cooling schedules using genetic algorithms. Boltzmann Tournament Selection maintains diversity in a population by sampling in such a way that the samples, over time, become Boltzmann distributed. The factor of genetic drift works against this selection by limiting the amount of diversity it can maintain.

Population of individuals is created using Boltzmann Tournament selection in following manner:

A population contains *n* individuals. In each generation, a three-way competition or "tournament" is held for each population slot.

1. First individual is chosen uniformly at random
2. Second individual is chosen with an objective function value different from the first by a threshold amount x.

3. There are two choices to choose the third individual.
    a) *Strict choice* states that the third individual is chosen with an objective function value different from the first and second individuals by the threshold amount.
    b) *Relaxed choice* states that the third individual is chosen with an objective function value different by the threshold amount from the first individual alone.

This algorithm is successful creates a steady proportion of individuals according to the Boltzmann distribution. If a uniform random sample is drawn from a Boltzmann -distributed population, it will be biased toward better individuals.

6.4.4 Steady State Selection - Genitor

Genitor works individual by individual, choosing an offspring for birth according to linear ranking and choosing the currently worst individual for replacement [Whitley 1989]. In steady−state selection, only a few individuals are replaced in each generation: usually a small number of the least fit individuals are replaced by offspring resulting from crossover and mutation of the fittest individuals. In this scheme, the worst λ individuals of the population are replaced

Elitism

Elitism copies the first best chromosome or the few best chromosomes to the new population. The rest of the population is created by normal method [De Jong 1975]. Elitism accumulates best individuals and prevents them from being lost or destroyed. This significantly improves the genetic algorithm's performance. Elitism is carried out in conjunction with age-based and stochastic fitness based replacement schemes.

6.5 Other Selection Methods

Sigma Scaling:

Sigma scaling is also called as "sigma truncation" [Forrest 1985]. Sigma Scaling keeps the selection pressure relatively constant over the course of the run rather than depending on the fitness variances in the population.

Under sigma scaling, an individual's expected value is a function of its fitness, the population mean, and the population standard deviation. An example of sigma scaling is:

$$ExpVal(X_i, t) = \begin{cases} 1 + \frac{F(X_i) - \bar{f}(t)}{2\sigma(t)} & if\ \sigma(t) \neq 0 \\ 1.0 & if\ \sigma(t) = 0 \end{cases} \quad \text{(Eq.xxiii)}$$

where ExpVal(i,t) is the expected value of individual Xi at time t, $f(Xi)$ is the fitness of Xi, $\bar{f}(t)$ is the mean fitness of the population at time t, and σ(t) is the standard deviation of the population fitnesses at time t.

If ExpVal(i,t) was less than 0, Tanese arbitrarily reset it to 0.1, so that individuals with very low fitness had some small chance of reproducing [mit98].

Fine-grained Tournament Selection (FGTS):

FGTS is an improvement of the tournament selection. FGTS is controlled by real value parameter F_{tour} (the desired average tournament size) instead of the integer parameter N_{tour} (the tournament size). Similarly to the tournament selection, an individual is chosen if it is the best individual on the tournament. However, unlike tournament selection, size of the tournament is not unique in the whole population, i.e., tournaments with different number of competitors can be held within one step of the selection. The parameter F_{tour} governs the selection procedure; therefore, average tournament size in population should be as close to F_{tour} as possible. Tournament selection randomly chooses a set of individuals and picks out the best for reproduction [Filipovic 1998]. The number of individuals in the set is mostly equal to two but larger tournament sizes can be used in order to increase the selection pressure.

Stochastic Tournament Selection:

Stochastic tournament selection was stated by Wetzel. In this selection, selection probabilities are calculated normally and successive pairs of chromosomes are drawn using roulette wheel selection [Wetzel 1983].

6.6 Annealed Selection – Proposed Research Work

In maximum optimisation problems, it has been observed and stated in the literature that there is no single cure for all the ills. Sometimes exploration techniques outsmart exploitation techniques and vice-versa. Influenced by these observations a number of

selection operators have been proposed but they were either inclined towards exploitation or exploration. Generally requirements necessitate that in the beginning of evolution cycle exploration is better and in the last exploitation. The focus of the research is to blend the selection operators and generate a new selection operator to obtain perfect fusion of exploration and exploitation. The research considered Roulette Wheel selection as exploitative Selection and Rank Selection as Exploratory selection. The proposed blended selection operator is being coined as **Annealed Selection**.

The name of proposed Annealed selection operator is based on the phenomenon of Simulated Annealing. Annealing is a heat treatment of material with the goal of altering its properties such as hardness. When annealing metal, the initial temperature must not be too low and the cooling must be done sufficiently slow so as to avoid the system getting stuck in a meta-stable, non-crystalline state representing a local minimum of energy [Weise 2007]. Simulated annealing uses a simple cooling schedule that lowers the temperature slowly so that there is no abrupt change in temperature. Similarly, the approach of Proposed Annealed selection is to gradually switch over the selection criteria from exploration to exploitation so as to obtain the perfect blend of the two techniques.

The proposed blending of the two contrasting techniques will lead a new path towards optimal solution effectively and hence better in terms of quality than the existing selection operators. Proposed annealed selection is also fitness dependent as roulette wheel selection. In this method, fitness value of each individual is computed according to both rank and roulette wheel selection. Depending upon the current generation number of genetic algorithm, selection pressure is changed and new fitness contribution, X of each individual is computed. Selection probability of each individual is computed on the basis of X. Fitness X is computed by combining the fitness of rank and roulette wheel selection in such a way that fitness of an individual gradually moves from rank selection to roulette wheel selection, thus, implementing the gradual move from exploration to exploitation. This gradual movement is achieved by multiplying fitness of both rank and roulette wheel selection by their respective factors depending on current generation number. As the generation of population changes, fitness contribution changes and selection probability of each individual also changes. The behaviour of proposed annealed selection operator can be easily modified as per requirement by changing the pressure.

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

$$\boldsymbol{FRW} = \frac{\boldsymbol{F_i}}{\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{F_i}} \quad \text{(Eq.xxiv)}$$

$$\boldsymbol{Frank} = \frac{\boldsymbol{rank_i}}{\boldsymbol{ranksum}} \quad \text{(Eq.xxv)}$$

Annealed Fitness is denoted by FX and is computed as under:

$$\boldsymbol{FX_i} = \boldsymbol{Frank} * \boldsymbol{ra} + \boldsymbol{FRW} * \boldsymbol{rb} \quad \text{(Eq.xxvi)}$$

where *ra* is factor that has initial value 1 and decreases by 1/ngen in each generation

rb is factor that has initial value 0 and increases by 1/ngen in each generation.

ngen is total number of generations

value of i moves from 1 to number of chromosomes in the population

F_i is fitness of and i^{th} individual

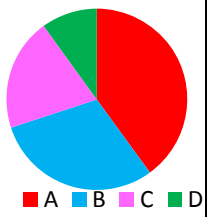
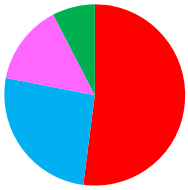
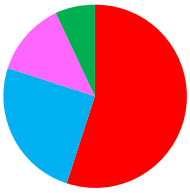
The probability for selecting the i^{th} string is

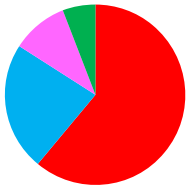
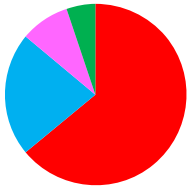
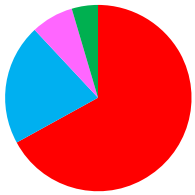
$$\boldsymbol{PX_i} = \frac{\boldsymbol{FXi}}{\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{FXi}} \quad \text{(Eq.xxvii)}$$

Algorithm of 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 $\boldsymbol{c_j=c_{j-1}+FX_{i,j}}$
         If r<=cj, Select the individual j
      End
   g) l=l+1
   End

6.7 Analysis of Fitness based on Annealed Selection Method

Generation	Chromosome A	Chromosome B	Chromosome C	Chromosome D	Pie Chart showing contribution of each chromosome
1	0.4	0.3	0.2	0.1	
2	0.43	0.29	0.186	0.094	
3	0.46	0.28	0.172	0.088	
4	0.49	0.27	0.158	0.082	
5	0.52	0.26	0.144	0.076	
6	0.55	0.25	0.13	0.07	

7	0.58	0.24	0.116	0.064	
8	0.61	0.23	0.102	0.058	
9	0.64	0.22	0.088	0.052	
10	0.67	0.21	0.074	0.046	

The fitness of the chromosome (individual) changes in each generation and similarly its probability of being selected also changes. In initial generations, Selection pressure is low, so the search space is explored. As the generations pass by, the selection pressure increases, thus moving from exploration to exploitation. It is evident from the pie charts that the selection pressure moves from exploration to exploitation gradually. On further experimentations, it has been found that proposed annealed selection has both features of exploration and exploitation i.e. it can be said to be blend of the two techniques.

6.8 Implementation of Annealed Selection

The performance of Proposed Annealed selection was initially tested on Travelling Salesman Problem for four different population sizes – 10 cities, 20 cities, 50 cities and 100 cities using the MATLAB code [Kumar *et al.* 2011]. The code assessed the performance of genetic algorithm by using roulette wheel selection, rank selection and annealed selection using the same initial population. Except selection criteria, all other factors affecting the performance

of genetic algorithm were kept constant. The code was run for 100 generations in each case. Average and minimum fitness in each generation was computed over 100 generations and plotted to compare the performance of three approaches. Figure 6.3 depicts the comparison of average tour length and Figure 6.4 depicts the comparison of minimum tour length in three different selection methods. Table 6.1 lists the detailed data for four different problem sizes and various parameters to analyse performance of the three methods. Comparison of average and minimum tour length is presented in Figure 6.5 and Figure 6.6 respectively.

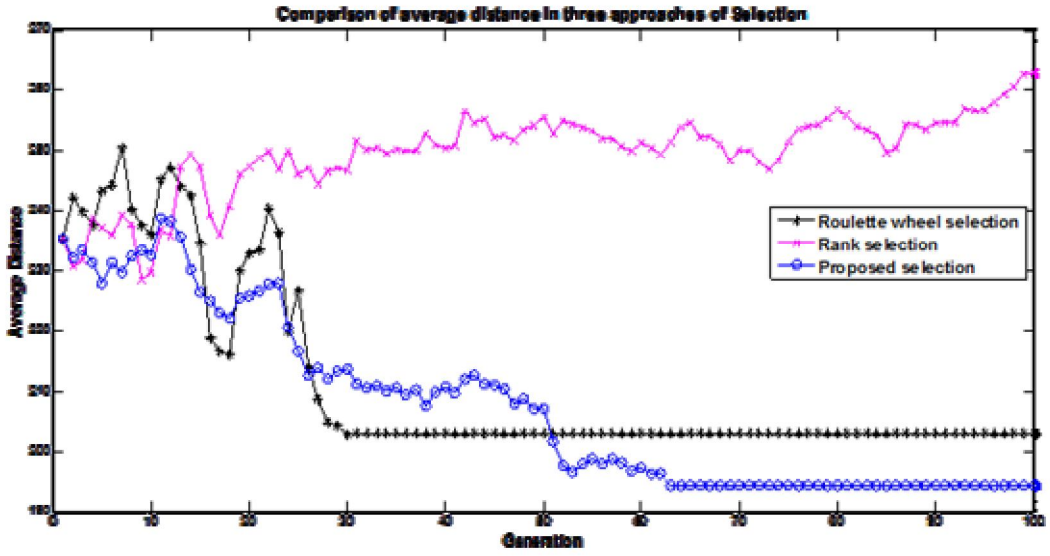


Figure 6.3 Comparison of Average Tour Length for Annealed Selection

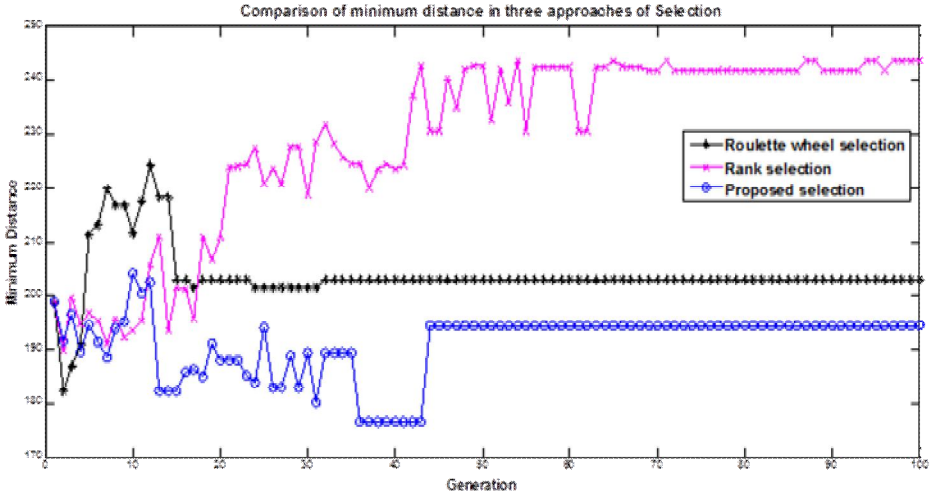


Figure 6.4 Comparison of Minimum Tour Length for Annealed Selection

Table 6.1 Comparison of Three Selection Approaches for TSP

Problem Size (PS) in cities	Rank Selection (RS)			Roulette Wheel Selection (RWS)			Proposed Annealed Selection (PBS)		
	F_{best} (Min)	Convergence Number	(Avg)	F_{best} (Min)	Convergence Number	(Avg)	F_{best} (Min)	Convergence Number	(Avg)
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

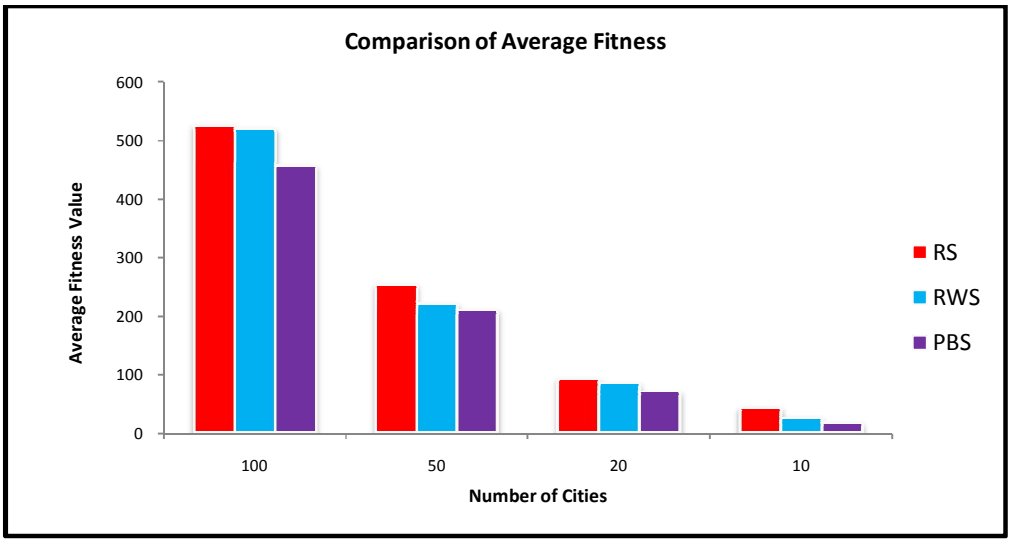


Figure 6.5 Comparison of Average Fitness on different Problem Sizes for Annealed Selection

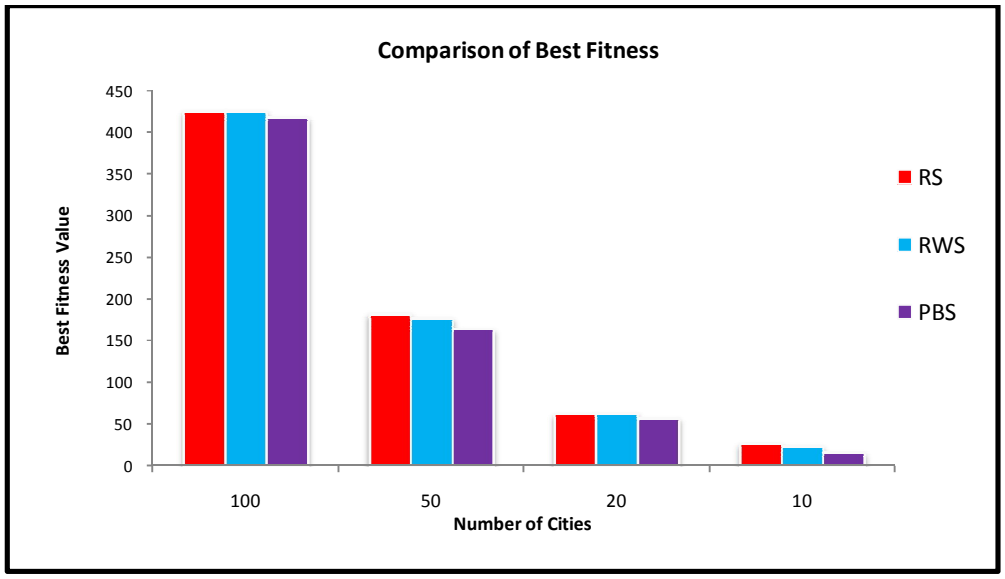


Figure 6.6 Comparison of Minimum fitness on different problem sizes for Annealed selection

Later the code was also tested on set of five benchmark functions stated in chapter 2. Average and minimum value of *fn(x)* in each generation is computed over 50 and 100 generations and plotted to compare the performance of three approaches. The comparison of selection techniques is based on their respective performance estimated as the value of the function. Average and Minimum value of evaluation of each function was recorded and examined for further analysis [Kumat *et al.* 2012d].

The following parameters were used in this implementation:

- Population size (N): 3,5,10 and 20
- Number of generations (ngen) : 50 and 100
- Population selection method: Roulette Wheel Selection (RWS), Rank Selection (RS) and Annealed Selection (AS)
- Crossover Operator: Simple arithmetic crossover
- Mutation: Inversion with mutation probability 5%
- Algorithm ending criteria: Execution stops on reaching ngen generations
- Fitness Function: Objective value of function

Table 6.2 : Average and Minimum Value of fn(x) in F1

N			**3**	**5**	**10**	**20**
Generations =50	**Roulette Wheel Selection (RWS)**	**Min**	4.93E-37	3.09E-34	4.74E-33	1.46E-31
		Avg	0.628	1.179	2.162	5.262
	Rank Selection (RS)	**Min**	5.73E-31	2.85E-30	5.54E-31	2.21E-30
		Avg	0.665	1.1346	2.176	5.122
	Proposed Annealed Selection (AS)	**Min**	8.89E-38	3.32E-35	2.44E-34	1.53E-32
		Avg	0.709	1.0545	2.482	5.155
Generations =100	**Roulette Wheel Selection (RWS)**	**Min**	1.93E-72	1.67E-70	4.87E-66	4.32E-65
		Avg	0.408	0.46	1.374	2.816
	Rank Selection (RS)	**Min**	1.18E-60	1.11E-64	5.20E-60	1.37E-60
		Avg	0.417	0.488	1.353	2.863
	Proposed Annealed Selection (AS)	**Min**	7.21E-74	8.01E-74	2.24E-69	1.51E-66
		Avg	0.411	0.455	1.314	2.71

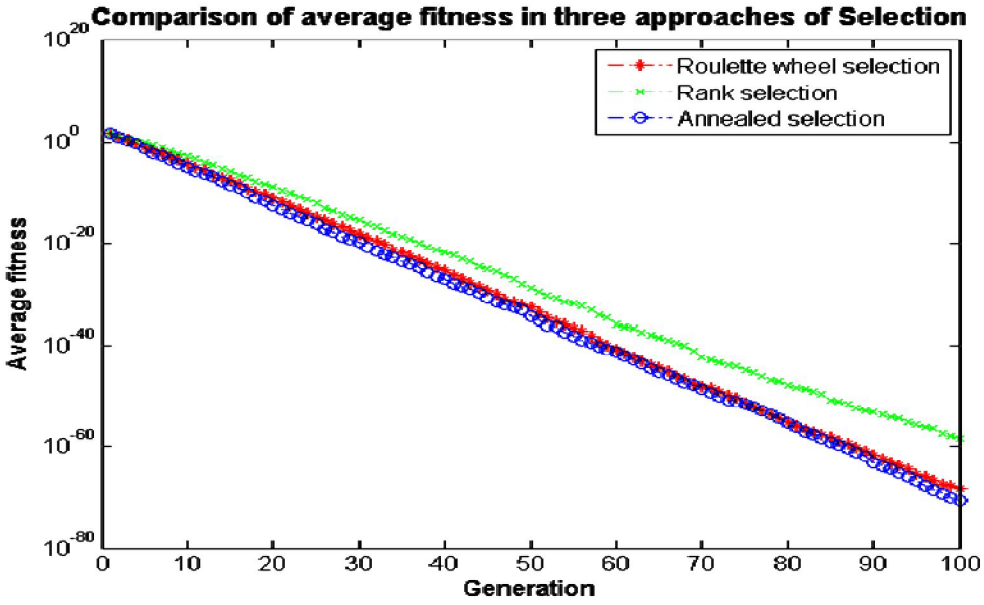


Figure 6.7: Comparison of Average Value of fn(x) in F1

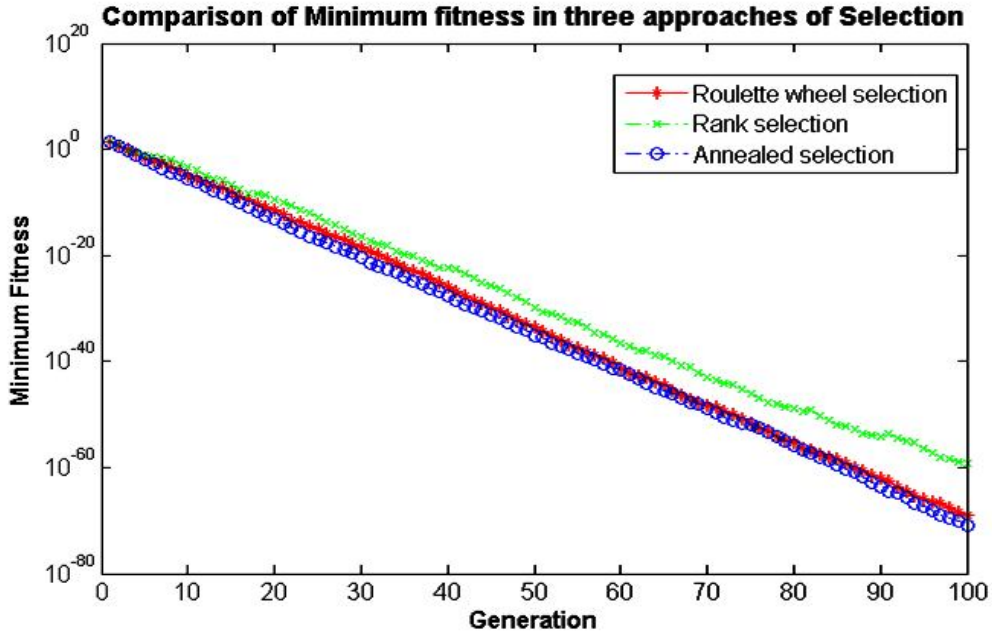


Figure 6.8: Comparison of Minimum Value of fn(x) in F1

Table 6.3 : Average and Minimum Value of fn(x) in F2

N			3	5	10	20
Generations =50	Roulette Wheel Selection (RWS)	Min	1.8951	3.8781	8.9686	18.9999
		Avg	14.3878	34.5157	98.7727	267.7132
	Rank Selection (RS)	Min	1.1274	3.9011	8.9738	18.9973
		Avg	14.3573	38.7858	110.1202	271.9678
	Proposed Annealed Selection (AS)	Min	1.9642	3.8918	8.8512	18.9502
		Avg	14.1299	35.7353	105.7585	257.652
Generations =100	Roulette Wheel Selection (RWS)	Min	1.818	3.9329	8.9625	18.9913
		Avg	14.6431	41.0678	51.9153	118.785
	Rank Selection (RS)	Min	1.9893	3.9999	9	18.9962
		Avg	14.8046	42.8148	54.4672	119.1835
	Proposed Annealed Selection (AS)	Min	1.9411	3.9258	8.9866	18.9985
		Avg	13.7012	39.6808	53.4466	119.699

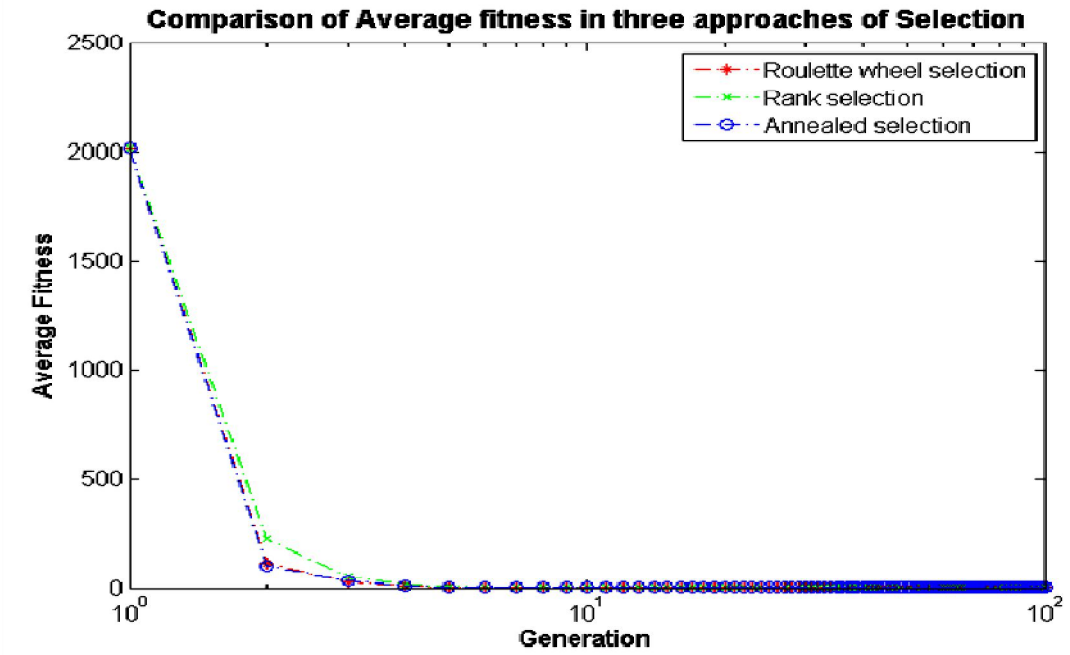


Figure 6.9: Comparison of Average Value of fn(x) in F2

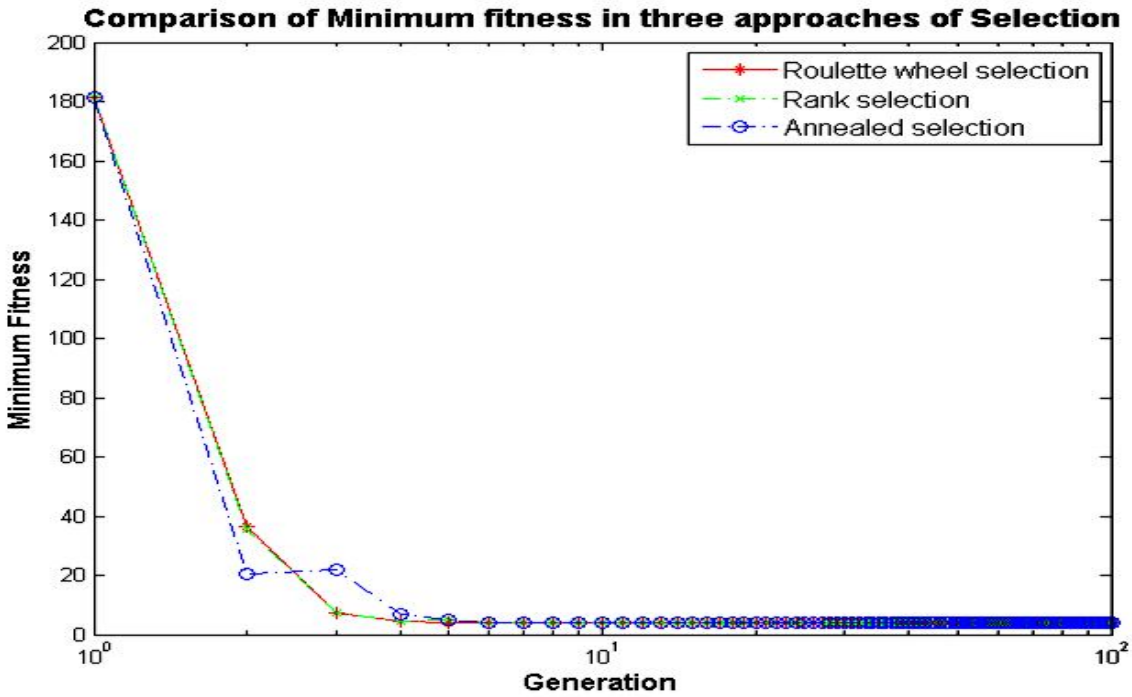


Figure 6.10: Comparison of Minimum Value of fn(x) in F2

Table 6.4 : Average and Minimum Value of fn(x) in F3

N			3	5	10	20
Generations =50	**Roulette Wheel Selection (RWS)**	**Min**	1.09E-16	5.33E-17	5.14E-16	3.99E-15
		Avg	0.312	0.42393	0.98183	2.0637
	Rank Selection (RS)	**Min**	6.57E-16	8.59E-16	9.35E-14	2.63E-13
		Avg	0.27969	0.49485	1.0676	2.065
	Proposed Annealed Selection (AS)	**Min**	1.52E-18	6.72E-18	3.11E-16	1.26E-15
		Avg	0.25408	0.40432	0.93658	1.986
Generations =100	**Roulette Wheel Selection (RWS)**	**Min**	5.93E-35	2.02E-34	7.66E-33	1.60E-32
		Avg	0.11417	0.21006	0.47329	0.98294
	Rank Selection (RS)	**Min**	1.14E-30	5.14E-31	4.02E-30	2.16E-31
		Avg	0.16419	0.25567	0.47437	0.94699
	Proposed Annealed Selection (AS)	**Min**	6.48E-37	3.13E-35	1.95E-33	1.22E-32
		Avg	0.13155	0.23074	0.5237	0.81515

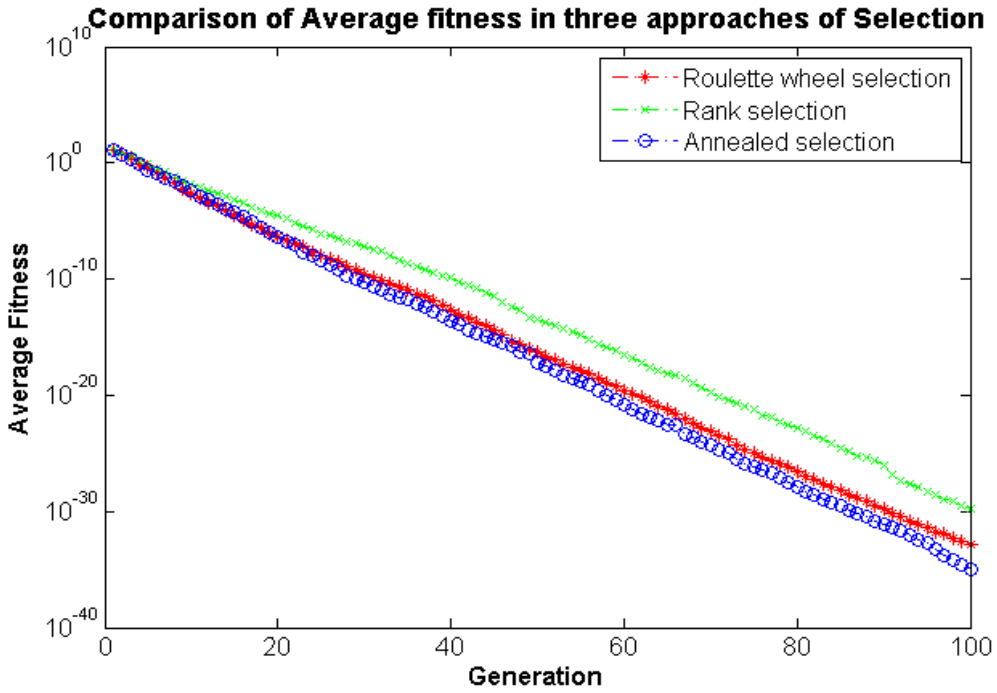


Figure 6.11: Comparison of Average Value of fn(x) in F3

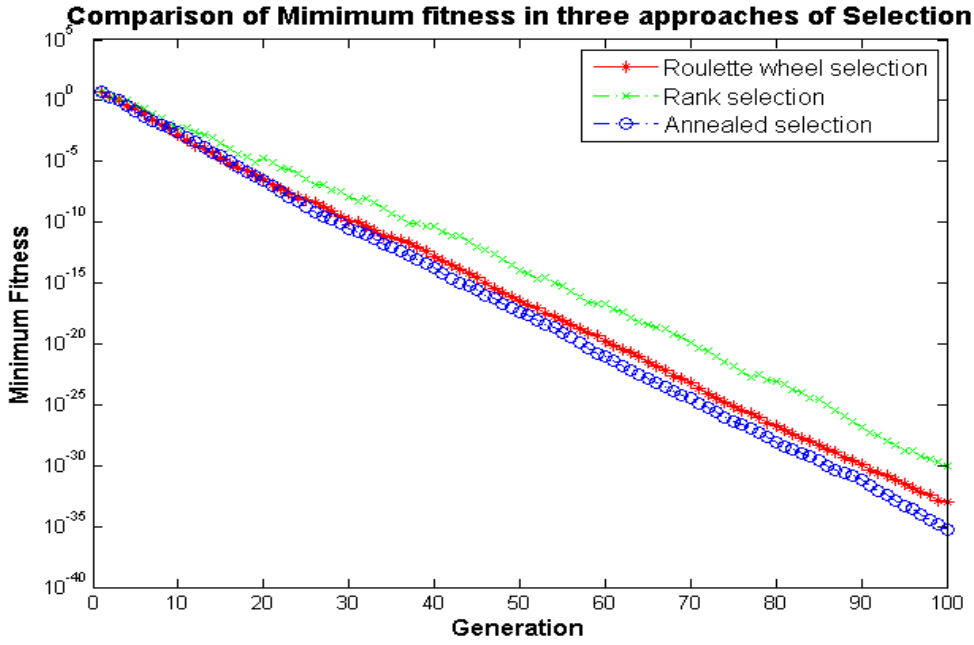


Figure 6.12: Comparison of Minimum Value of fn(x) in F3

Table 6.5 : Average and Minimum Value of fn(x) in F4

N			3	5	10	20
Generations =50	Roulette Wheel Selection (RWS)	Min	70	50	8.9813	-100
		Avg	73.4838	55.4561	99.8336	-78.1042
	Rank Selection (RS)	Min	70	50	9	-100
		Avg	73.0887	56.7271	100.3791	-78.4591
	Proposed Annealed Selection (AS)	Min	70	50	8.9622	-100
		Avg	73.6985	56.8295	96.1036	-77.8765
Generations =100	Roulette Wheel Selection (RWS)	Min	70	50	8.9813	-100
		Avg	72.0275	53.192	95.036	-87.6698
	Rank Selection (RS)	Min	70	50	9	-100
		Avg	71.9712	53.0511	96.1081	-88.4613
	Proposed Annealed Selection (AS)	Min	70	50	8.9622	-100
		Avg	71.9692	52.7239	92.5329	-87.0142

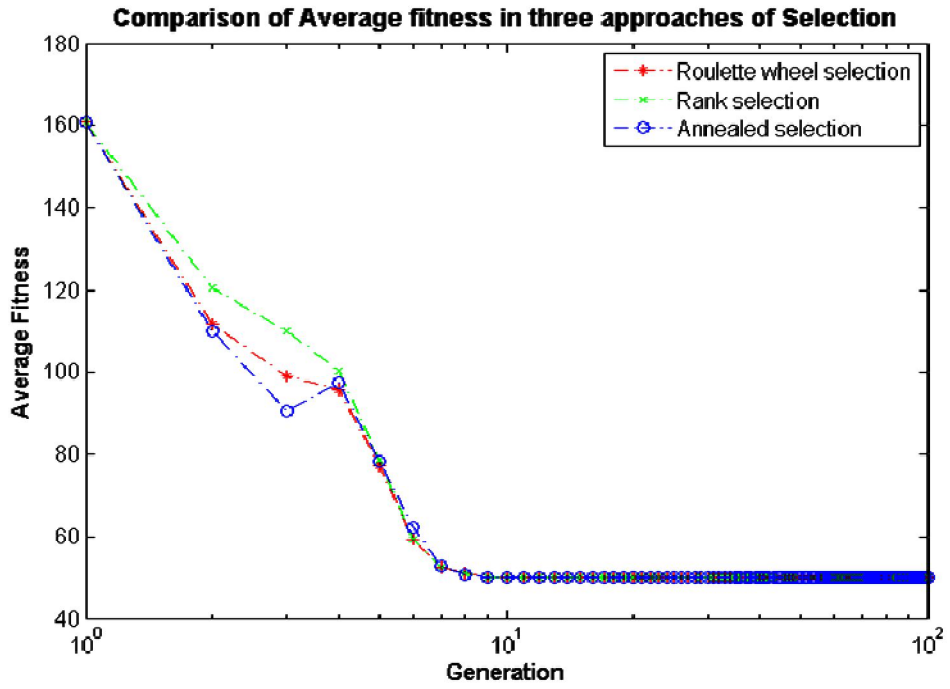


Figure 6.13: Comparison of Average Value of fn(x) in F4

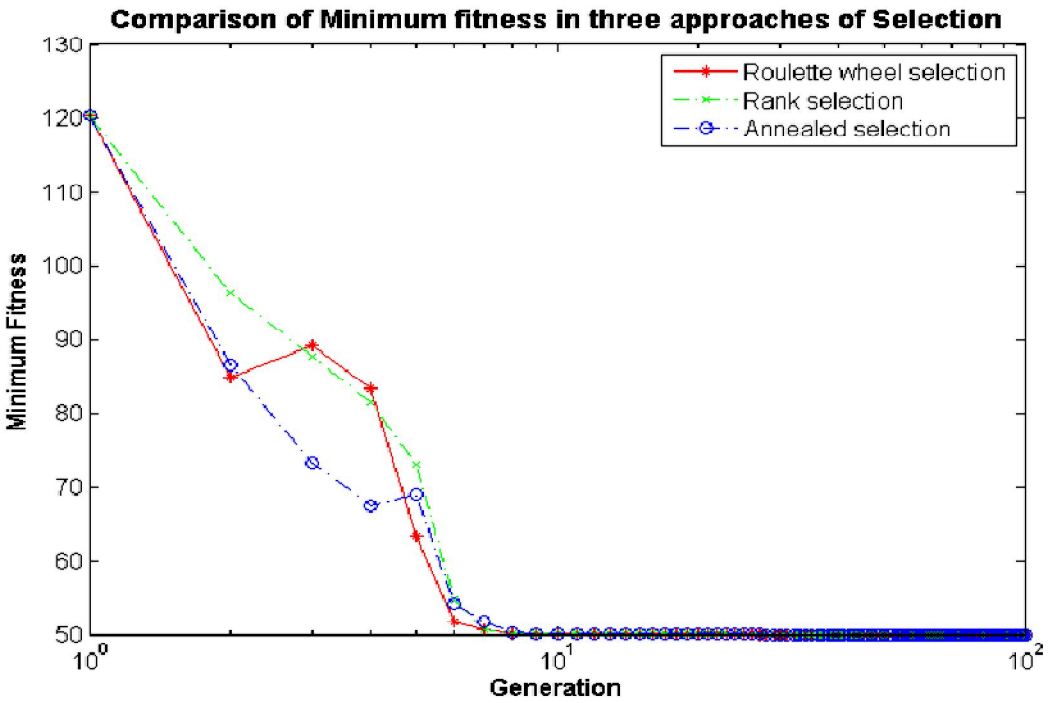


Figure 6.14: Comparison of Minimum Value of fn(x) in F4

Table 6.6 : Average and Minimum Value of fn(x) in F5

N			3	5	10	20
Generations =50	**Roulette Wheel Selection (RWS)**	**Min**	1.3684	1.0696	8.88E-16	-4.6708
		Avg	1.6662	1.4204	0.28187	-2.3732
	Rank Selection (RS)	**Min**	1.3684	1.0696	8.88E-16	-4.6708
		Avg	1.6623	1.4429	0.29853	2.3372
	Proposed Annealed Selection (AS)	**Min**	1.3684	1.0696	8.88E-16	-4.6708
		Avg	1.6066	1.4558	0.27037	-2.2904
Generations =100	**Roulette Wheel Selection (RWS)**	**Min**	1.3684	1.0696	8.88E-16	-4.6708
		Avg	1.4991	1.2506	0.27216	-3.563
	Rank Selection (RS)	**Min**	1.3684	1.0696	8.88E-16	-4.6708
		Avg	1.51	1.2343	0.277	-3.433
	Proposed Annealed Selection (AS)	**Min**	1.3684	1.0696	8.88E-16	-4.6708
		Avg	1.5168	1.2449	0.2574	-3.6287

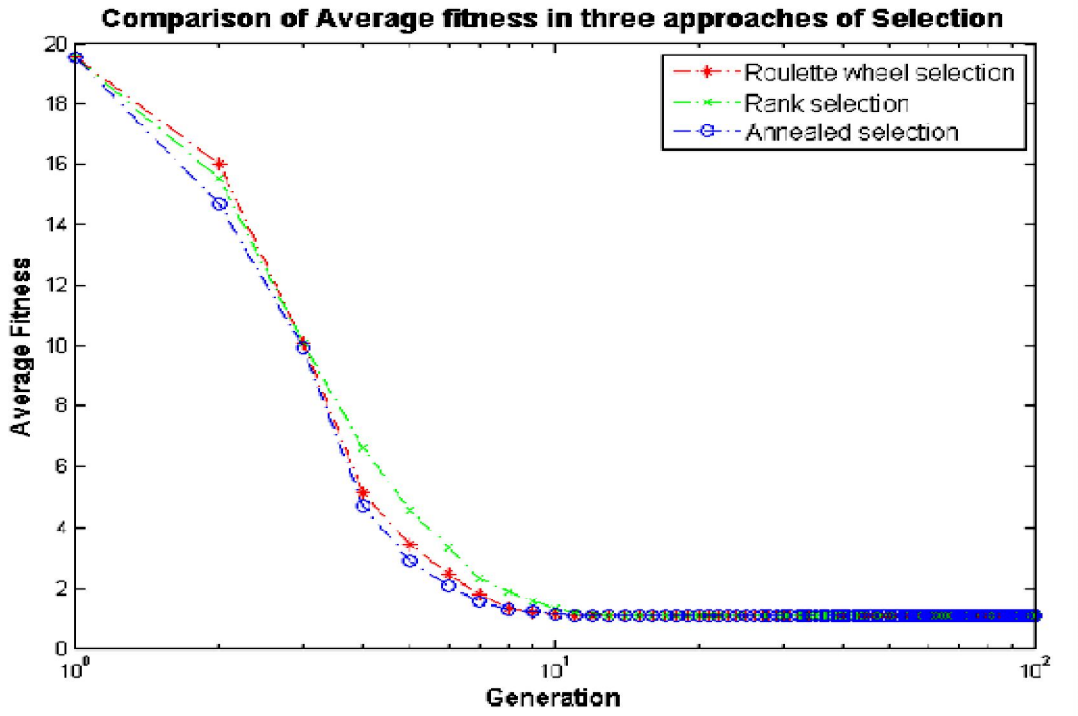


Figure 6.15: Comparison of Average Value of fn(x) in F5

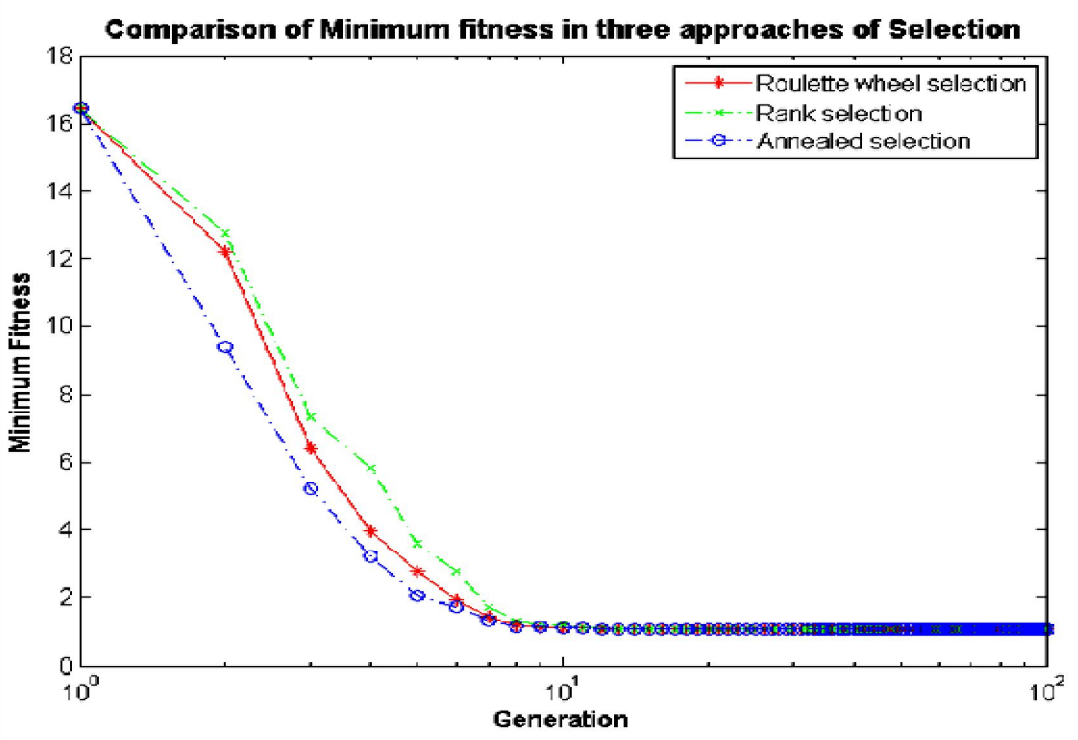


Figure 6.16: Comparison of Minimum Value of fn(x) in F5

6.9 Observations and Findings

In order to study the effect of existing selection operators and proposed annealed selection operator on the performance of genetic algorithm, the implementation has been carried out by keeping the factors initial population, crossover type and its probability, mutation type and mutation rate constant in all the cases. Test runs were carried out for 50 generations as well as 100 generations and four different population sizes. The results for the five benchmark functions in terms of average and minimum value are summarized in Table 6.2, 6.3, 6.4, 6.5 and 6.6. Figures 6.7, 6.8, 6.9, 6.10, 6.11, 6.12, 6.13, 6.14, 6.15 and 6.16 show the performance curves of the three selection operators in terms of average and minimum value of fitness function for population size 20 and 100 generations.

It has been observed that in maximum cases proposed annealed selection has outperformed the roulette wheel selection and rank selection. It can be clearly seen that in early runs of generation, the proposed annealed selection explores the search space and as the number of generations increases, there is increased selection pressure and the annealed selection uses exploitation mechanism in selecting the individual. In early generations, the behaviour of annealed selection is just like the rank selection and gradually it transforms into roulette wheel selection and elitism, this justifies that annealed selection is perfect blend of exploration and exploitation. Test runs have been conducted on Travelling Salesman Problem as well as five different benchmark functions and optimistic results in favour of annealed selection are produced.

6.10 Summary

The primary objective of the selection operator is to emphasize better solutions in a population. The identification of good or bad solutions in a population is usually accomplished according to a solution's fitness. Selection techniques are either based on fitness of an individual or on the ranking of the individuals in a population. Numerous selection techniques have been identified and used in the past. The chapter gave a systematic overview of existing approaches of selection and then discussed the selection operator proposed in the research work. The proposed annealed selection operator is a step ahead of existing operators and considers both exploration as well as exploitation approach.

The proposed annealed selection operator was test on TSP problem as well as benchmark functions resulting in optimistic results. The tests affirmed that the in early generations, the proposed annealed selection operator explored the search space as the selection pressure was low. As the number of generations passes by, selection pressure increases thus transforming the exploratory nature of selection to the exploiting nature. The proposed annealed selection neither gets lost in the search space indefinitely nor results in premature convergence. The research on proposed annealed selection operator is quite satisfactory and the operator is used in conjunction with other genetic operators in forthcoming chapters to substantiate its relevance.