Chapter 7: REPLACEMENT

7.1 Need of Replacement

A genetic algorithm operates on population of constant size. An initial population of individuals is generated randomly or heuristically. Selection operator is used to improve the quality of the population by selecting the fittest individuals to form the mating pool. Crossover operator takes two individuals with higher fitness values from the mating pool and randomly chooses the position and length of the portion to be exchanged and performs this operation at either single or multiple points. Mutation introduces new genetic structures in the population by randomly modifying some of the genes, helping the search algorithm to escape from local optimum by reaching new points in the search space. Current generation of individuals is replaced by newly generated offsprings by the specific replacement strategy. Genetic algorithms are stochastic iterative algorithms, so the algorithm iterates till maximum number of generations is reached or the cycle of genetic algorithm continues until the optimal solution is achieved.

When a new generation of offsprings is produced, the next question is which of these newly generated offsprings would move forward to the next generation and would replace which chromosomes of the current generation. The answer to this question is based on Darwin's principle of "Survival of Fittest". So better fit individuals have more chances to survive and carried forward to next generation leaving behind the less fit ones. The process of forming next generation of individuals by replacing or removing some offsprings or parent individuals is done by replacement operator. This process in evolution is known as replacement scheme.
Replacement strategy helps to find out the individuals that would replace the current generation to form next generation of population.

7.2 Types of Replacement

Replacement is the last step in breeding step of any genetic algorithm cycle. The technique used to decide which individuals stay in a population and which are replaced in on a par with selection in influencing convergence [Sivanandam *et al.* 2007]. Basically, there are two kinds of replacement strategies for maintaining the population – generational replacement and steady state replacement.

A. Generational Replacement

In generational replacement, entire population of genomes is replaced at each generation. It means newly generated offsprings would move forward to the next generation and replace all the chromosomes of the current generation [Fogel 1995]. Two consecutive generations are non-overlapping using generational replacement.

Module for Generational Replacement is :

```
Replace(P,S,n)
    //S is set of chromosomes generated as offspring
    //P is parent generation of chromosomes
    P:=S
End
```

Derived forms of generational replacements are μ+λ replacement and μ,λ replacement.

i) **μ+λ Replacement:** In (μ+λ) replacement strategy, both parent and offspring set of chromosomes is grouped into one unit. Both parents and offsprings compete for survival. They are sorted on the basis of fitness value. Then the best set of chromosomes equivalent to population size are chosen to form the next generation of population [Smith *et al.* 1999].

Module for μ+λ Replacement is :

```
Replace(P,S,n)
    //S is set of chromosomes generated as offspring
    //P is parent generation of chromosomes
    //n is population size
    npool:= R + S
    Sort npool on the basis of fitness value and store in snpool
    Choose n best chromosomes from snpool and store in P
End
```

ii) **μ,λ Replacement:** In (μ,λ) replacement strategy, the number of offsprings created is far greater than the number of parents. The offsprings created are ranked according to their fitness and the best μ offsprings are selected to replace the population of parent chromosomes [Schwefel 1981]. In this case, only the offspring population is being used in replacing the parent population.

Module for μ,λ Replacement is :

```
Replace(P,S,n)
    //S is set of chromosomes generated as offspring
    //P is parent generation of chromosomes
    //n is population size
    //number of chromosomes in S is greater than n
    npool:= S
    Sort npool on the basis of fitness value and store in snpool
    Choose n best chromosomes from snpool and store in P
End
```

B. Steady State Replacement

Steady state replacement involves overlapping population in which only a small fraction of the population is replaced during each iteration. In a steady state replacement, new individuals are inserted in the population as soon as they are created [Cavicchio 1970]. This gives an advantage that highly fit offsprings can participate in further genetic operations without waiting for a generation to complete and get selected in mating pool. So when a highly fit offspring is inserted in the population, then the worst individual is selected to be replaced. In steady state replacement, the two consecutive populations are non-overlapping. The term generation gap refers to the amount of overlap between parents and offsprings.

Module for Steady State Replacement is :

```
Replace(P,O,n)
  //O is chromosome generated as offspring
  //P is parent generation of chromosomes
  //n is population size
```

```
   Evaluate offspring O with fitness function
   Select a chromposome from P to be replaced and store in L
   if (O is better)
        replace L with O
   endif
End
```

Steady state replacement has been implemented in GENITOR. Whitley introduced GENITOR in which worst λ individuals were deterministically replaced every iteration. This led to very rapid improvements in the mean population fitness, but in certain cases also lead to premature convergence as the population focused on fittest member currently present [Mitchell 1996]. Goldberg & Deb analysed GENITOR and observed that it has high selective pressure even if parents are selected randomly and suggested that deletion of worst individuals was major factor in selection intensity [Goldberg *et al.* 1991, Affenzeller *et al.* 2009].

7.3 Other Related Replacement Strategies

- Elitism: In case of elitism, complete population of genome is replaced except for the best member of each generation which is carried over to next generation without modification.
- Delete n-last: In Delete-n-last, the n weakest individuals are replaced by n descendants.
- Delete n: In this case, arbitrarily chosen n individuals of old population is replaced by n individuals from new population. It reduces convergence speed of genetic algorithm and also avoids premature convergence.
- Random Replacement: In Random Replacement, the children replace two randomly chosen individuals in the population. The parents are also candidates for selection. This can be useful for continuing the search in small populations, since weak individuals can be introduced into the population.
- Weak Parent Replacement: In weak parent replacement, a weaker parent is replaced by a strong child. With the four individuals only the fittest two, parent or child, return

to population. This process improves the overall fitness of the population when paired with a selection technique that selects both fit and weak parents for crossing, but if weak individuals and discriminated against in selection the opportunity will never raise to replace them [Sivanandam *et al.* 2007].

- Both parents replacement: In this case, each individual is allowed to breed only once. Both the parent chromosomes get replaced in each generation by the offsprings. It helps in change in genetic material in each breeding cycle and introduces diversity. But, this scheme has a drawback that it does not retain fit chromosomes in forthcoming generations. Thus, it may lose strongly favored fit parents selected in future generations.

7.4 Research Motive

It has been observed from the literature review that the performance of genetic algorithm is also influenced by the type of replacement strategy used. In the previous chapter, proposed annealed selection operator was compared with Roulette wheel selection and Rank Selection keeping all other parameters constant. Replacement is also important genetic operator that has an impact on evolution. So, the research was carried out to study and analyse various replacement strategies and their effect. In this chapter, the effect of different replacement strategies have been analysed in conjunction with Roulette wheel selection, Rank selection and Proposed Annealed selection on the performance of genetic algorithm.

7.5 Genetic Algorithm Implementing Replacement

```
Procedure GA(tourlength, θ, n, r, m,ngen)
//tourlength is fitness function to evaluate chomosomes
//θ is the fitness threshold that determines when to halt
//Since objective of test problem is minimization, best fitness is minimum value of tourlength
// n is the population size in each generation (say 100)
// r is fraction of population generated by crossover (say 0.7)
// m  is the mutation rate ( say 0.01)
 //ngen is total number of generations
```

```
P := generate n individuals at random
// initial generation is generated randomly
// h represents the chromosome in the population P
i:=1

while min(tourlength($h_i$)) < $\theta$ or i <=ngen do
{
    //define the next generation S of size n
    //Reproduction step:
        Select $h_i$ having min(tourlength) and compare with pb and store the best in $k_1$
        //best individual in generation selected as parent
        //Select n/2 individuals of P as per any of the selection methods
        Call Select(P,n,r) and store in L
        //Crossover step:
        Probabilistically select (1-r)n individuals from L and store in $k_2$

        foreach pair selected ($k_1$, $k_2$), produce two offspring by applying the PMX
            crossover operator and add these offspring to S
        //Mutation step:
        Choose m% of S and Mutate chromosomes by inversion
        //Replacement Step
        // Generate new population as per any of the replacement strategies
        Call Replace(S,n) and store in P
        i:=i+1
}

Find b such that tourlength(b) = min(tourlength($h_i$))
return(b)
end proc
```

7.6 Implementation and Observation

Genetic algorithm was developed using MATLAB for two test problems – Benchmark TSP-Eil51 problem and Benchmark sphere function (F1) to compare the effect of different selection operators in combination with two different replacement schemes on algorithm's performance . Various parameters used for this implementation are listed in Table 7.1.

Parameter	**Eil51 Benchmark TSP instance**	**Benchmark DeJong's Sphere Function (F1)**
Population size	51 chromosomes	3,5 , 10 and 20
Number of generations	100	10
Fitness Function	Tourlength of Path	F1 objective function
Encoding	Permutation encoding	Real Value encoding
Selection method	Roulette wheel selection, Rank Selection and Proposed annealed selection	Roulette wheel selection, Rank Selection and Proposed annealed selection
Crossover operator	PMX crossover	Arithmetic Crossover
Mutation	Inversion with mutation probability 5%	Inversion with mutation probability 5%
Algorithm ending criteria	Execution stops on reaching maximum generations	Execution stops on reaching maximum generations

Table 7.1: Parameters Used to Compare Performance of Replacement Schemes

The code used the same initial population to compare the performance of genetic algorithm in various cases of selection and replacement. Test runs have been carried out to compare the performance of genetic algorithm affected by varying selection operator and replacement scheme. Here, three different selection methods, namely, Roulette wheel selection, Rank selection and Proposed Annealed selection [Kumar *et al.* 2011] have been considered with two replacement schemes that are Generational replacement and μ+λ replacement [Kumar *et al.* 2012c]. Figure 7.1 and Figure 7.2 illustrate the comparison of average and minimum tourlength respectively of Eil51 TSP problem for 100 generations. Figure 7.3 and Figure 7.4 illustrate the comparison of average and minimum tour length selection wise using column chart. Figure 7.5 and Figure 7.6 illustrate the comparison of average and minimum fitness respectively of De Jong's benchmark test function F1 for 10 generations.

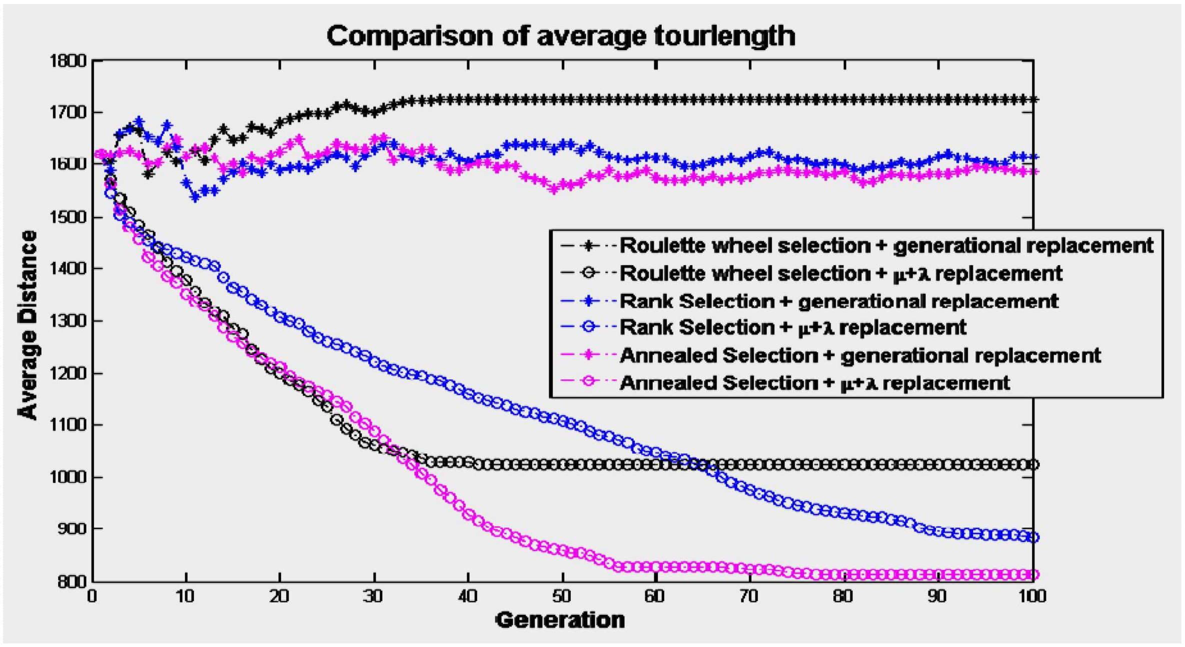

Figure 7.1: Comparison of Average Tour Length of Eil51 TSP Using Generational and μ+λ Replacement

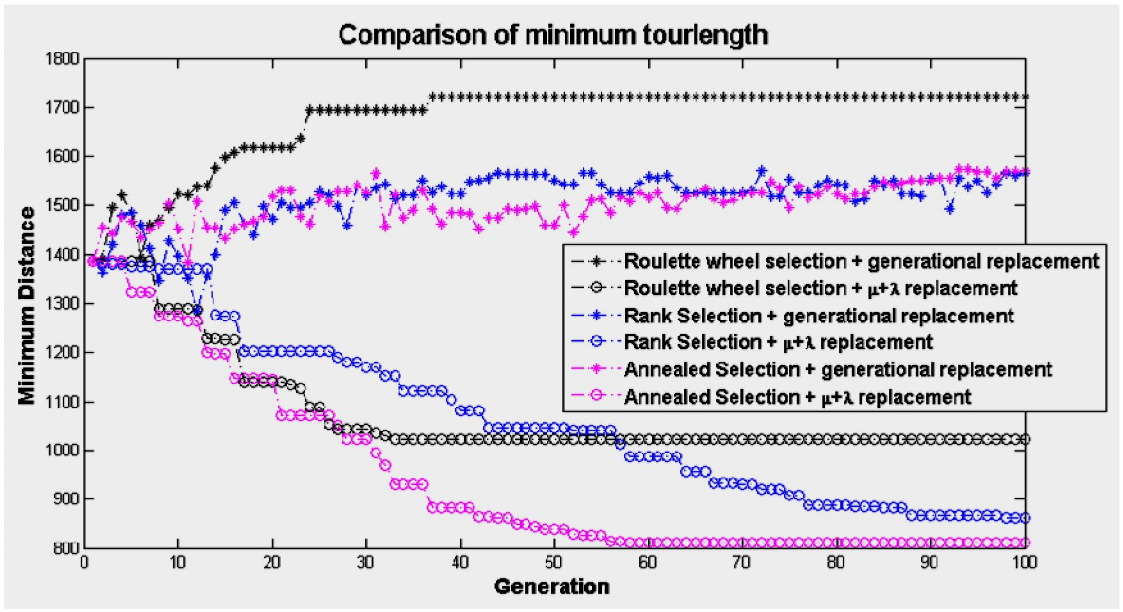

Figure 7.2 : Comparison of Minimum Tour Length of Eil51 TSP Using Generational and μ+λ Replacement

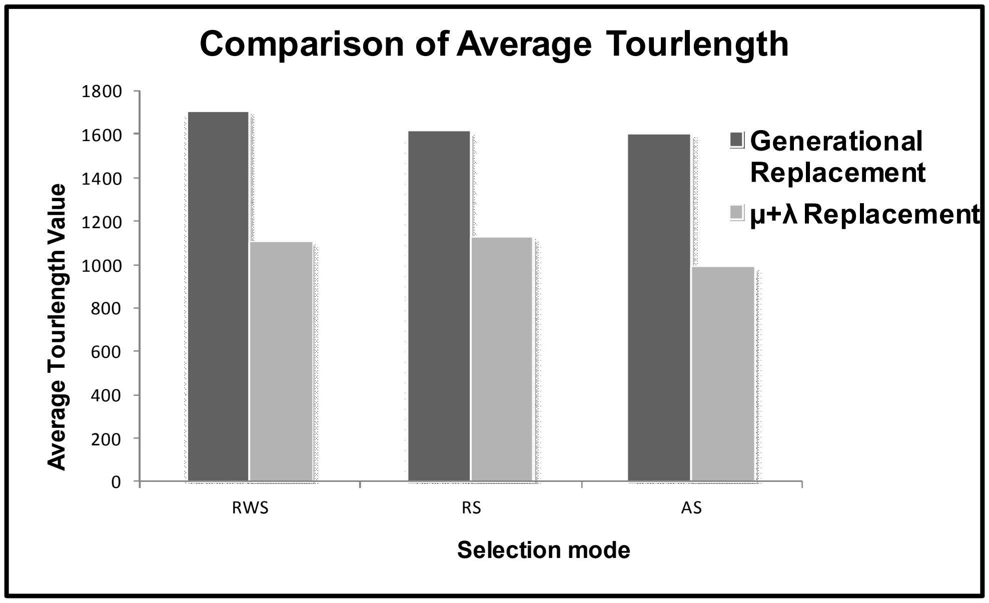

Figure 7.3 : Comparison of Average fitness of Different Selection Approaches Using Generational and μ+λ Replacement

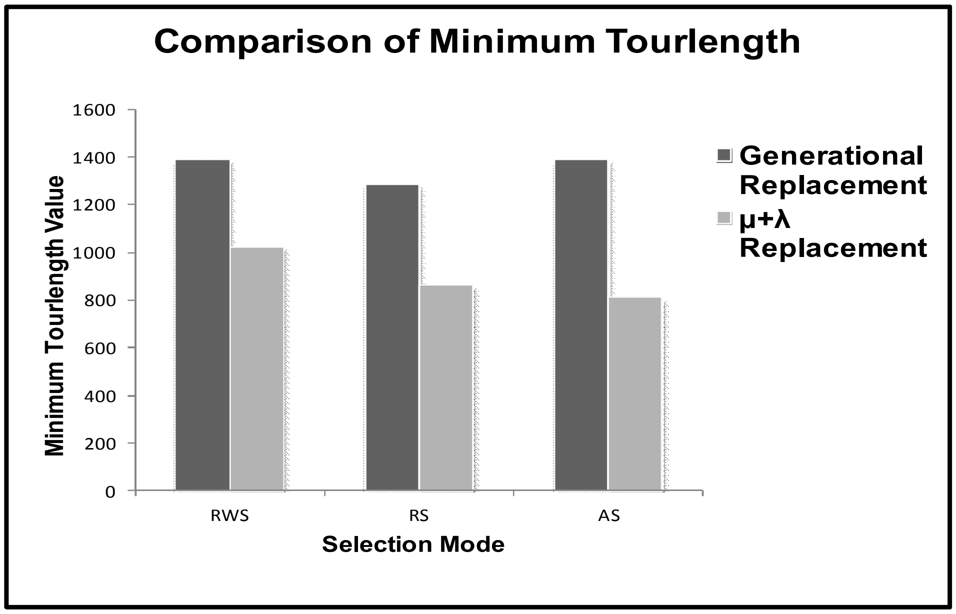

Figure 7.4 : Comparison of Best Fitness of Different Selection Approaches Using Generational and μ+λ Replacement

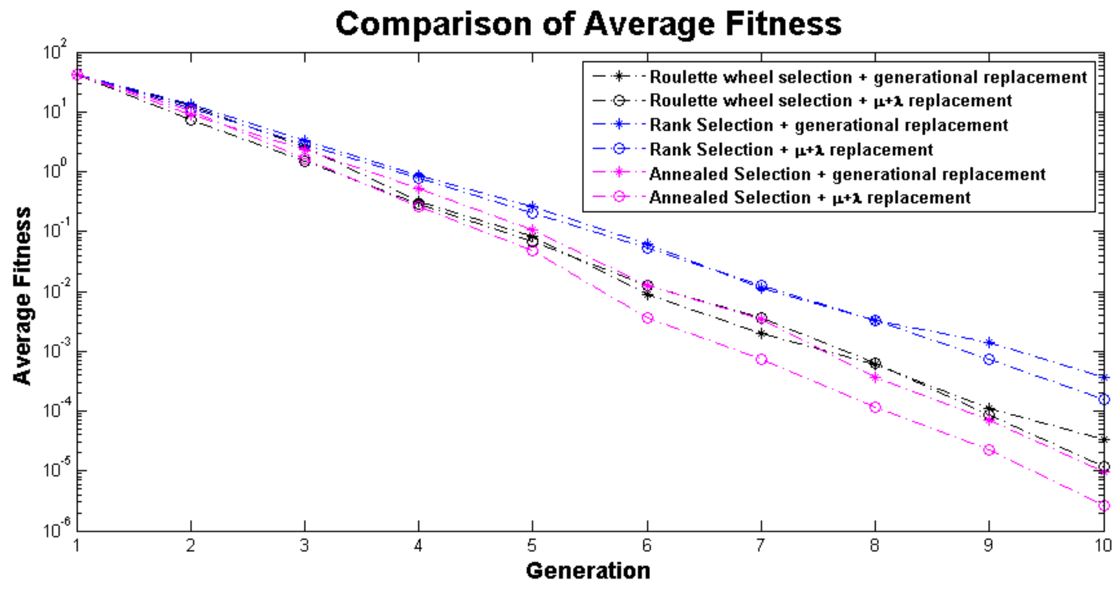

Figure 7.5 : Comparison of Average Fitness of Benchmark F1 Function Using Generational and μ+λ Replacement

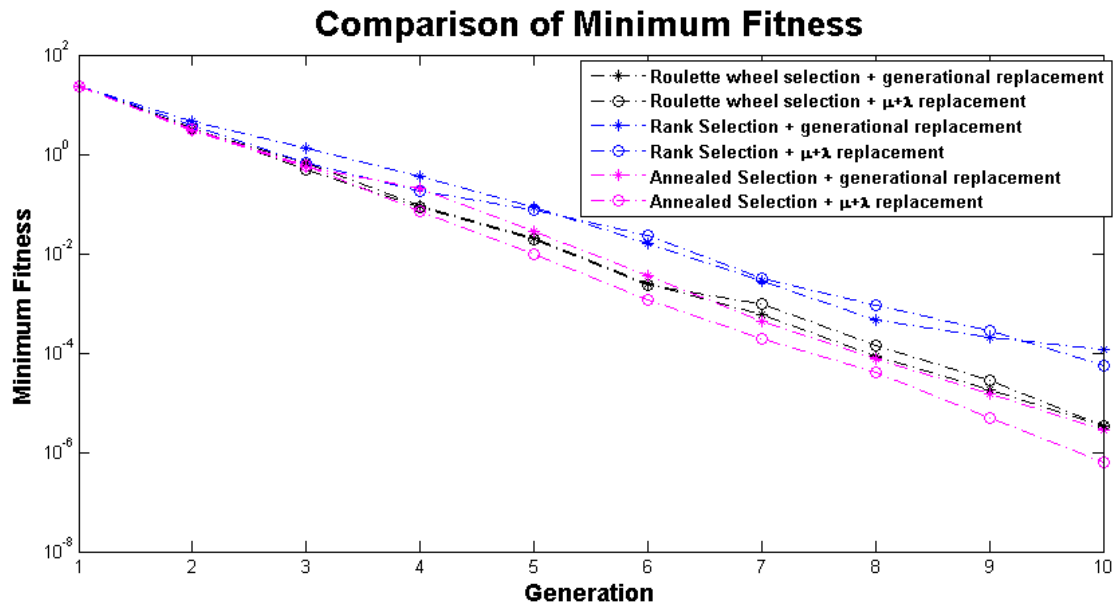

Figure 7.6 : Comparison of Minimum Fitness of Benchmark F1 Function Using Generational and μ+λ Replacement

The performance of annealed selection operator that has been proposed in this research is analysed in combination with two different replacement methods - generational replacement and μ+λ replacement. The results clearly demonstrated that the genetic algorithm implementing μ+λ replacement is better than the one implementing generational replacement. Out of the three selection operators, proposed annealed selection outperformed roulette wheel selection and rank selection. It has been experimentally proved that proposed annealed selection operator has a blend of exploration as well as exploitation and gives better results in both replacement strategies. Test runs confirmed optimistic results in both the test problems – Eil51 and De Jong Sphere Function (F1) in favour of Annealed selection and μ+λ replacement.

7.7 Summary

Diversity in population is fundamental requirement for evolution. Replacement operator in genetic algorithm adds its contribution in maintaining population diversity. Different replacement strategies are discussed in the chapter. On conjunction of different selection methods with two replacement strategies, it is clear that both selection and replacement have their impact on the performance of genetic algorithm. Combination of appropriate selection and replacement strategy can lead to better results in comparison to other techniques and also help in maintaining population diversity over generations.