Comparison of parents selection methods of genetic algorithm for TSP

Article · January 2011

CITATIONS
18

READS
206

3 authors, including:

Chetan Chudasama
Madhuben & Bhanubhai Patel Women's Institute of Engineering for Studies & Research in Computer & Communication Technology
5 PUBLICATIONS **18** CITATIONS

SEE PROFILE

Proceedings published by International Journal of Computer Applications® (IJCA)
International Conference on Computer Communication and Networks CSI- COMNET-2011

Comparison of Parents Selection Methods of Genetic Algorithm for TSP

Chetan Chudasama
PG Student
Kalol Institute of Technology and Research Center,kalol

S. M. Shah
Asst. Processor
Government Engineering College, Gandhinagar

Mahesh Panchal
Asst. Processor
Kalol Institute of Technology and Research Center, kalol

ABSTRACT
One of Area of Artificial Intelligence is used to optimize combinatorial problem. Many combinatorial problems like Travelling Salesman, Network Graph problem are optimize by Artificial Intelligence (AI) Searching Methods. One of Searching method of AI is Genetic Algorithm. In Genetic Algorithm parent selection for next generation is very important because solution of problem depends on how much optimized solution derives. In this paper, we present comparative performance of roulette wheel, Elitism and tournament selection method for Travelling Salesman problem. And we found that elitism method is best in all these methods.

Keywords
Combinatorial problem, TSP, Genetic algorithm, Elitism, Roulette wheel selection, Tournament selection

1. INTRODUCTION
As far as the artificial inelegance is concerned, the genetic algorithm is an optimization technique based on natural evolution that is the change over a long period of time. Genetic algorithm (GAs) has been used as a search technique of many NP problems. Genetic algorithms have been successfully applied to many different types of problems, though several factors limit the success of a GA on a specific function. Problem required are good, but optimal solutions are not ideal for GAs. The manner in which points on the search space are represented is an important consideration. An acceptable performance measure or fitness value must be available.

It must also be feasible to test many potential solutions. In nature the fittest individual is most likely to survive and mutate, therefore the next generation should be fitter and healthier because they were bred from healthy parents. The same idea can be applied on the TSP problem by first finding the different solutions and then combine those, which are the fittest solutions among them, in order to create a new and healthy solution and should be optimal or near optimal according to the problem. [1]

Selection is the process of choosing two parents from the population for crossing. The purpose of selection is to emphasize fitter individuals in the population in hopes that their off springs have higher fitness. Chromosomes are selected from the initial population to be parents for reproduction. Selection is a method that randomly picks chromosomes out of the population according to their evaluation function. The higher the fitness function, the more chance an individual has to be selected. [2]

In this paper we are showing the comparison of selection methods (reproduction) which is used for selecting the fittest parent for next generation for TSP. three methods were tested Roulette wheel Selection, Tournament Selection and Elitism selection method.

2. INTRODUTION TO TRAVELLING SALESMAN PROBLEM
The traveling salesman problem (TSP) is perhaps the most well known combinatorial optimization problem. TSP is to find a routing of a salesman who starts from a home location, visits a prescribed set of cities and returns to the original location in such a way that the total distance travelled is minimum and each city is visited exactly once. Although a business tour of a modern day traveling salesman may not seem to be too complex in terms of route planning, the TSP in its generality represents a typical 'hard' combinatorial optimization problem.

3. GENETIC ALOGORITHM
Genetic algorithm is Evolutionary algorithm. Evolutionary algorithms can be applied to any problems that can be formulated as function optimization problems. Genetic algorithm applied which has large solution search space. Search space is space of all feasible solutions (the set of solutions among which the desired solution resides). In the population of individuals, the strong ones survive longer than the weak ones "survival of the fittest". This causes a rise in the overall fitness of population. Based on the fitness value of the individuals, the week ones are terminated. The strong ones are chosen to reproduce themselves by using recombination and/ or mutation on them. Recombination is an action that is applied to two or more of the selected candidate (called parents). It will generate one or more new candidates (called children). Mutation is applied to one candidate and results in one new candidate. Execution of these two operators leads to a set of new candidates that compete with the old ones for a place in the next generation. When this process is repeated, the average fittest of the population will increase until a maximum has been reached.

The selection process of which individuals will be terminated, and which will be parents for the next generation. Evolution is a process of adaption. The fitness function that is used for evaluating the individuals tells us something about the requirements that are needed to survive in the environment. To obtain a higher fitness value, the population needs to adapt increasingly more to the environment.

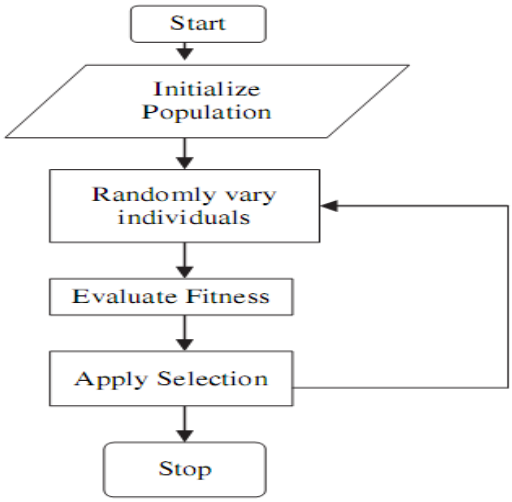

Fig -1 Flow chart of evolution process of genetic algorithm

In genetic algorithm, first the population needs to be initialized with random candidate. Next step evaluate the fitness of individuals. Then we apply the selection operator which selects fittest parents for generations which are parents of next generation. The algorithm is stopped when the population converges toward the optimal solution. Fig -1 show steps of evolution process of genetic algorithm.

3.1 Fitness function

Fitness function is the important parameter of a genetic algorithm that defines the fitness of each chromosome where the values of genetic parameters are adapted as the genetic evolution progresses. At every generation, fitness value of each chromosome is calculated using fitness function. If fitness of two chromosomes is equal, then the mutation rate is increased, in order to help the genetic evolution get out of issues like local maxima or local minima whichever is applicable. Once there is an improvement in the overall fitness, the original mutation rate is restored to continue evolution as normal. If the evolution stabilizes, but the fitness does not seem to be improving for several generations and the search does not find any error, new set of initial population is generated using the initial default parameter values and a new randomly generated seed. [3][4]

TSP is a minimization problem; we consider fitness function calculates cost (or value) of the tour represented by a chromosome.

3.2 Selection Methods

3.2.1 Roulette wheel Selection

Roulette method selection method works similarly to a roulette wheel, where the likelihood that an individual is chosen is proportional to its fitness value. Because a 0 is considered the ideal fitness for individuals in this algorithm, the size of each individuals 'slice' of the roulette wheel will be
inversely proportional to their fitness value. Once the 'slices' have been determined, a number is generated at random. The individual with the range of numbers that contains this randomly generated number will be one parent. This continues until the desired number of parents is found.

Based on the value of fitness function, roulette wheel method selects the next best possible solution chromosome that will create a new generation and be genetic parents for the next generations. It is also allow for parents with low fitness to go to the next generation. [5] Fig-1 shows the pseudo code for roulette wheel selection method. [6]

```
void rouletteParents(numParents, myPopulation, parents){
int high = 0;
for each(chromosome : myPopulation)
if(chromosome.fitness > high)
high = chromosome.fitness;
create vector of integers = size of myPopulation;
for(0 <= i < myPopulation.size())
for(0 <= j < high – myPopulation.elementAt(i).fitness)
add i to vector of integers;
for(0 <= i < numParents)
shuffle vector of integers;
parents.add(myPopulation.elementAt(vector[i]);
 }
```

Fig-2 Pseudo code for Roulette parent selection

3.2.2 Tournament Selection

In tournament selection, every individual in the population is paired at random with another. The fitness values of each pair are compared. The fitter individual of the pair moves on to the next 'round', while the other is
disqualified. This continues until there are a number of winners equal to the desired number of parents. Then this last group of winners is paired as the parents for new individuals. Fig-3 shows the pseudo code for tournament selection method. [6]

```
void tournamentParents(numParents, myPopulation, parents){

create temporary empty population;
create vector of integers = size of myPopulation;
add values 0 to index of last member of myPopulation;
shuffle vector;
if(myPopulation.size <= numParents)
parents = myPopulation;
else
for(0 <= i < myPopulation.size/2){
compare myPopulation element at i and myPopulation.size –
 i
add most fit to temporary population;
}
tournamentParents(numParents, temporary population, parent
s)
}
```

Fig-3: Pseudo code for Tournament parent selection

3.2.3 Elitism

In this method, each individual is assigned a fitness value via the fitness function. Using these scores, a percentage of the best, most fit individuals are used as parents. To start, the first numParents numbers of individuals from the population are chosen as the parents. Then, each member of the population after that is compared one by one to each of the parents. If a member of the population is found to be more fit than an existing parent, that parent is swapped out with that member. This continues until all the members have been compared against the existing parents. Then the original members of the population that were initially selected as parents are compared to the current set of parents to ensure that the fittest individuals are chosen as parents and a more fit individual was not replaced prematurely.

Fig-4 shows the pseudo code for the Elitism method. [6]

```
void eliteParents(numParents, myPopulation, parents)
{
for(0 <= i < numParents)
parents.add(myPopulation.elementAt(i);
for(numParents <= i < myPopulation.size)
for each(chromosome : parents)
if(chromosome.fitness > myPopulation.elementAt(i).fitness)
replace chromosome with myPopulation.elementAt(i);
break;
for(0 <= i < numParents)
for each(chromosome : parents)
if(chromosome.fitness > myPopulation.elementAt(i).fitness)
replace chromosome with myPopulation.elementAt(i);
break;
}
```

Fig-4: Pseudo code for Elitism parent selection

3.3 Crossover

After the completion of the selection process, the chromosomes chosen to be parents for the next generation are recombined to form children that are new chromosomes. This combination can take many forms and the crossover operator can greatly affects the result of the search. [6]

3.4 Mutation

The operation of mutation allow new individual to be created. It begins by selecting an individual from the population based on its fitness. A point along the string is selected at random and the character at that point is randomly changed, the alternate individual is then copied in to the next generation of the population.

Mutation is performed after crossover by randomly choosing a chromosome in the new generation to mutate. We randomly choose a point to mutate and switch that point. Many types of mutation operators exist. Here we are using the bit flip method which is only used with a binary chromosome representation, that changes a particular point on the chromosome to its opposite. [6]

4. IMPLEMENTATION

The population size was set to 100. And Generations were generated up to 100. The initial values are selected from the 1024 pool size. Fitness proportional selection was employed through the evolution function. The fittest solution of TSP is selected and forward to the next generation. The crossover rate and mutation rate is taken70% and 0.2% respectively.

5. RESULT ANALYSIS

When we applied all three selection methods- Roulette wheel, Elitism, Tournament selection methods for finding the optimal solution of TSP problem, this observation is shown fig-5. In this optimal solution of TSP is generated using roulette wheel selection method, tournament selection method and elitism.

We observed that at initial stage all selection method work similarly and in last stage elitism methods gets best fittest population than both tournament and roulette wheel selection method. So we find the best solution using elitism method.

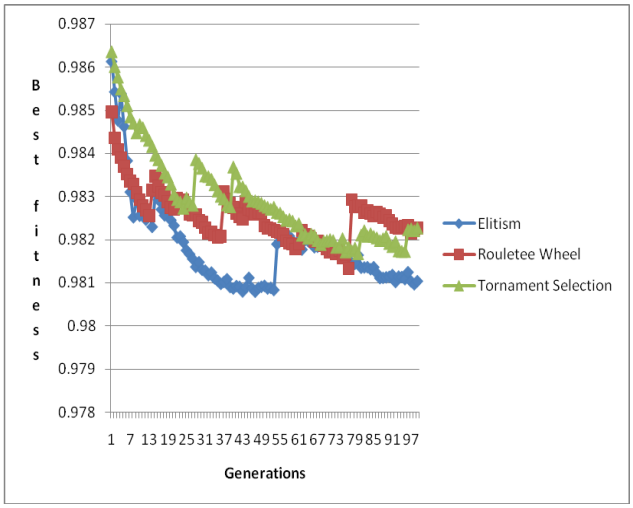

Fig- 5 Best fitness of chromosome of generations

6. CONCLUSION

TSP is combinatorial problem. So, if we taken large set of cities than search space becomes large. So, in early transition steps finding optimize solution from number of feasible solutions we have to select best fittest parents for next generation. It is done by fitness function and selection method for selecting fittest parents.

Comparative result analysis for fitness value using roulette wheel, tournament selection and elitism selection method clearly show that elitism method has generated very fit populations generation compare to former both methods as shows in fig-5.

7. REFERENCES

[1] Fozia Hanif Khan, Nasiruddin Khan, Syed Inyatullah and Shaikh Tajuddin Nizam. Solving TSP problem TSP solving TSP problem by using genetic algorithm. Pages[79-88]. International Journal of Basic & Applied Sciences IJBAS Vol: 9 No: 10.

[2] S.N. Sivananadam, S.N. Deepa, Introduction to Genetic Algorithm, Springer- Verlag Berlin Heidelberg 2008.

[3] G. Kendall and G. Whitwell. An evolutionary approach for the tuning of a chess evaluation function using population dynamics. In Proceedings of the 2001 Congress on Evolutionary Computation, pages 995–1002. IEEE Press, World Trade Center, Seoul, Korea, 2001

[4] Shah, S.M.; Thaker, C. S.; Singh, D.; Multimedia based fitness function optimization through evolutionary game learning, Emerging Trends in Networks and Computer Communications (ETNCC), 2011 International Conference on Publication Year: 2011, page(s): 164-168.

[5] Goldberg, D.E.: Genetic Algorithm in Search, Optimization and Machine Learning, Addison Wesley pub. Co. (1989)

[6] http://www2.stetson.edu/mathcs/people/students/research/pdf/2010/dwells/final.pdf