New Modifications of Selection Operator in Genetic Algorithms for the Traveling Salesman Problem

Radovic, Marija; and Milutinovic, Veljko

Abstract—*One of the algorithms used for solving Traveling Salesman Problem is the genetic algorithm. It consists of three important parts: Selection, Crossover, and Mutation. In this paper some of the important concepts and methods of Selection are described. The paper is divided in two sections. In the first one, some of the most popular selection methods are described and in the second one, some new ideas about improving selection methods using the Internet knowledge are presented.*

Index Terms—*Genetic Algorithms, Selection, Traveling Salesman Problem, Semantic Web, Data mining*

1. INTRODUCTION

SELECTION is one of the main operators used in evolutionary computing. 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 can be used in different stages of evolutionary algorithms. Some algorithms (specifically genetic algorithms and genetic programming) usually apply the selection operator first to select good solutions and then apply the recombination and mutation operators on these good solutions, to create a hopefully better set of solutions. Other algorithms (evolution strategies and evolutionary programming) prefer using recombination and mutation operator first to create a set of solutions and then use the selection operator to choose a good set of solutions.

There are two purposes of this report. One is to give a systematic overview of existing approaches. And the other one is to introduce new approaches based on the usage of the Internet.

Manuscript received May, 2006. This work was supported in part by the Faculty of Uppsala, Sweden.

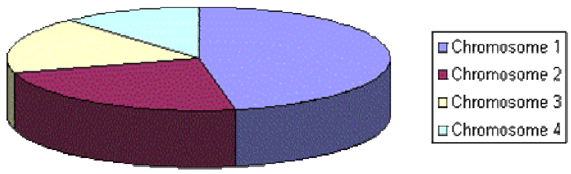
Radovic Marija is with the Faculty of Mathematics, University of Belgrade, Serbia.

Miltinovic Veljko is with the Faculty of Electrical Engineering, University of Belgrade, Serbia.

2. EXISTING SOLUTIONS AND THEIR CRITICISM

2.1 Roulette Wheel Selection

Individuals are selected according to their fitness. The better the chromosomes are, the more chances to be selected they have. Imagine a roulette wheel where all the chromosomes in the population are placed. The size of the section in the roulette wheel is proportional to the value of the fitness function of every individual - the bigger the value is, the larger the section is.



A marble is thrown in the roulette wheel and the chromosome where it stops is selected. Clearly, the chromosomes with bigger fitness value will be selected more times.

Algorithm for roulette wheel selection reads:
Input: population a
Output: population after selection a'

begin
  s_o:=O
  /* forming the roulette wheel */
  for i:= 1 to n do
    $s_i:=s_{i-1}+f_i/n$
  end
  /* simulation of throwing the marble */
  for i:= 1 to n do
    r:= random([0, s_n])
    $a_i' = a_k$ for such k to be accomplished $s_{k-1} < r < s_k$
  end
  return a'
end

Example:
6 random numbers: 0.81, 0.32, 0.96, 0.01, 0.65, 0.42.

Figure 1 shows the selection process of the individuals together with the sample trials.

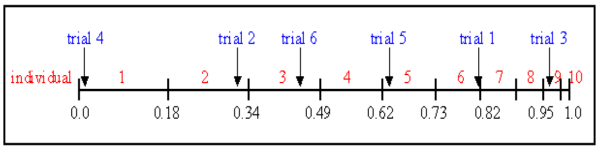


Figure 1: Roulette-wheel selection

After selection the mating population consists of the individuals: 1, 2, 3, 5, 6, 9.

2.2 Rank Selection

The previous type of selection will have problems when there are big differences between the fitness values. For example, if the best chromosome fitness is 90% of the sum of all fitness's then the other chromosomes will have very few chances to be selected.

Rank selection ranks the population first and then every chromosome receives fitness value determined by this ranking. The worst will have the fitness 1, the second worst 2 etc. and the best will have fitness N (number of chromosomes in population).

You can see in following picture, how the situation changes after changing fitness to the numbers determined by the ranking.

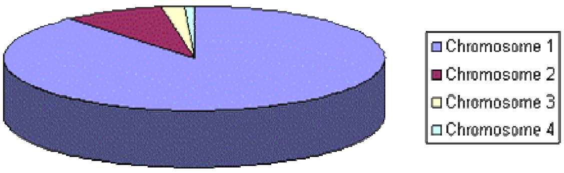


Figure 2: Situation before ranking (graph of fitness's)

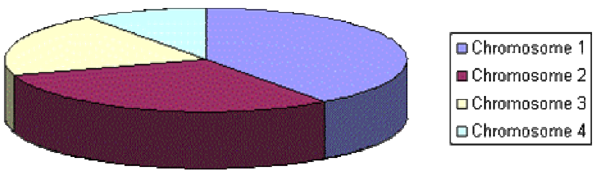


Figure 3: Situation after ranking (graph of order numbers)

Now all the chromosomes have a chance to be selected. However this method can lead to slower convergence, because the best chromosomes do not differ so much from other ones.

There are two types of ranking selection – linear ranking and exponential ranking.

Linear ranking assigns a selection probability to each individual that is proportional to the individual's rank (where the rank of the least fit is defined to be zero and rank of the most fit is defined to be m-1, given a population of size m). This method has one parameter: the degree of reproduction of the least fit individual - r.

Algorithm for linear ranking:

Input: population a, the degree of reproduction of the least fit individual r in interval [0, l]

Output: population after selection a'

begjn
  a' := population a sorted in ascending order by fitness value
  /* forming a roulette wheel */
  s_o:=O
  for i := 1 to n do
  $s_i := s_{i-1} + p_s(a_i)$
  /* value $p_s(a_i)$ is selection probability */
  end
  /* simulation of throwing the marble */
  for i: = 1 to n do
    r := random($[0, s_n]$)
    $a_i' := a_k'$ for such k to be accomplished $s_{k-1} < r < s_k$
    end
  return a'
end

Exponential ranking is different from the linear ranking because an exponential function is used for determination of selection probability.

2.3 Stochastic Universal Sampling 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. Consider NPointer the number of individuals to be selected, then the distance between the pointers are 1/NPointer and the position of the first pointer is given by a randomly generated number in the range [0, 1/NPointer].

For 6 individuals to be selected, the distance between the pointers is 1/6=0.167. Figure 4 shows the selection for the following example.

1 random number in the range [0, 0.167]: 0.1.

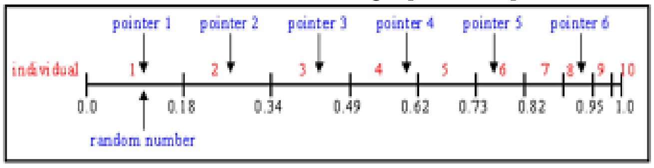


Figure 4: Stochastic universal sampling

After selection the mating population consists of the individuals: 1, 2, 3, 4, 6, 8.

Algorithm for stochastic universal sampling:

Input: population a, the degree of reproduction of the least fit individual R in interval [0,l]

Output: population after selection a'

begin
  sum := 0
  j := 1
  ptr := random([0,1])
  for i := 1 to n do
    sum := sum + R_i
    /* R_i - the degree of reproduction of the individual a_i */
    while (sum > ptr) do
      $a_j' := a_i$
      j:=j+1
      ptr: = ptr + 1
      end
    end
  return a'
end

For more information about the algorithm, see [3].

Stochastic universal sampling ensures a selection of offspring which is closer to what is deserved then roulette wheel selection.

2.4 Tournament Selection

Tournament selection is a rank-based method. The probability that an individual will be selected is based only on the rank of that individual in the population ordered by fitness, and not on the size of the fitness.

In tournament selection, element of population is chosen for passing into next generation if it is better (has better fitness value) than several randomly selected opponents. Tournament size N_{tour} is selection parameter, n is a size of the population and population after selection is a'. Running time for this algorithm is $O(n*N_{tour})$.

Algorithm for tournament selection:

Input: Population a (size of a is n), tournament size N_{tour}, $N_{tour} \in N$

Output: Population after selection a' (size of a' is n)

```
begin
  for i := 1 to n do
    ai' := best fitted item among Ntour elements randomly selected from population;
  return a';
end;
```

An advantage of tournament selection is that it is very easy to implement, and it works very well in a parallel implementation where different individuals are on different processors.

2.5 Fine Grained Tournament Selection

This is a variation of tournament selection. Instead of integer parameter N_{tour} (which represents tournament size), new operator allows real valued parameter F_{tour} – wanted average tournament size. This parameter governs selection procedure, so average tournament size in population should be as close as possible to it.

Algorithm for fine grained tournament selection:

Input: Population a (size of a is n), wished average tournament size F_{tour}, $F_{tour} \in R$

Output: Population after selection a' (size of a' is n)

```
begin
  Ftour- := trunc( Ftour )
  Ftour+ := trunc( Ftour ) + 1
  n- := trunc ( n * ( 1 - ( t - trunc ( Ftour ) ) ) )
  n+ := n - trunc ( n * ( 1 - ( t - trunc ( Ftour ) ) ) )
  /* tournaments with size Ftour- */
  for i := 1 to n- do
    ai' := best fitted item among Ftour- elements randomly selected from population;
  /* tournaments with size Ftour+ */
  for i := n-+1 to n do
    ai' := best fitted item among Ftour+ elements randomly selected from population;
  return a'
end;
```

For more information about this method, see [2].

2.6 Local Selection

In local selection every individual resides inside a constrained environment called the local neighborhood. (In the other selection methods the whole population or subpopulation is the selection pool or neighborhood.) Individuals interact only with individuals inside this region. The neighborhood is defined by the structure in which the population is distributed. The neighborhood can be seen as the group of potential mating partners.

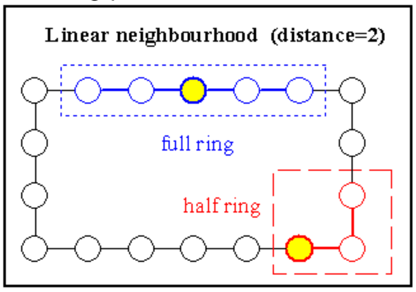


Figure 5: Linear neighborhood: full and half ring

The first step is the selection of the first half of the mating population uniform at random (or using one of the other mentioned selection algorithms, for example, stochastic universal sampling or truncation selection). Now a local neighborhood is defined for every selected individual. Inside this neighborhood the mating partner is selected (best, fitness proportional, or uniform at random).

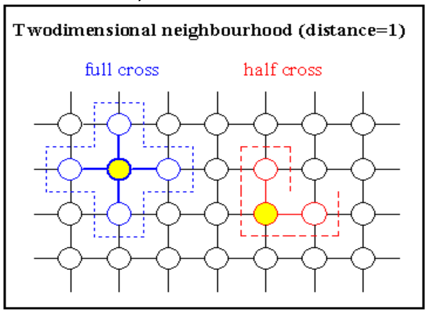


Figure 6: Two-dimensional neighborhood: full and half cross

The structure of the neighborhood can be:

- linear
- full ring, half ring (see figure 5)
- two-dimensional
- full cross, half cross (see figure 6)
- full star, half star (see figure 7)
- three-dimensional and more complex with any combination of the above structures.

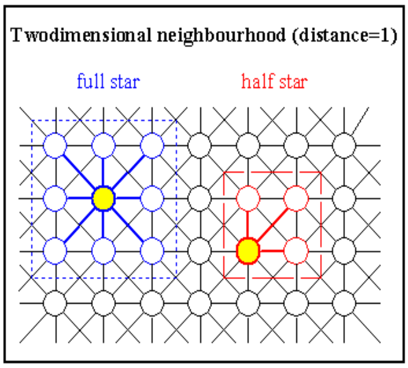


Figure 7: Two-dimensional neighborhood: full and half star

The distance between possible neighbors together with the structure determines the size of the neighborhood. Table 1 gives examples for the size of the neighborhood for the given structures and different distance values.

Between individuals of a population' isolation by distance' exists. The smaller the neighborhood, the bigger the isolation distances. However, because of overlapping neighborhoods, propagation of new variants takes place. This assures the exchange of information between all individuals.

structure	distance 1	distance 2
full ring	2	4
half ring	1	2
full cross	4	8 (12)
half cross	2	4 (5)
full star	8	24
half star	3	8

Table 1: Number of neighbors for local selection

The size of the neighborhood determines the speed of propagation of information between the individuals of a population, thus deciding between rapid propagation or maintenance of a high diversity/variability in the population. A higher variability is often desired, thus preventing problems such as premature convergence to a local minimum. Local selection in a small neighborhood performed better than local selection in a bigger neighborhood. Nevertheless, the interconnection of the whole population must still be provided.

Two-dimensional neighborhood with structure half star using a distance of 1 is recommended for local selection. However, if the population is bigger (>100 individuals) a greater distance and/or another two-dimensional neighborhood should be used.

Algorithm for local selection:
Input: Population a
Output: Population after selection a'

```
begin
  a':= population after local selection
  return a';
end;
```

Before we can use data mining models and algorithms we have to find the most suitable strategy. In order to do that, we have to detect the problem type. Usually, data mining project involves a combination of different problem types, which together solve the problem [1].

At the lower end of the scale of the data mining problems is 'data description' and 'summarization'. It aims at the concise description of characteristics of the data, typically in elementary and aggregated form. This gives us an overview of the structure of the data.

The next data mining problem type is problem of 'segmentation'. It aims at the separation of data into interesting and meaningful subgroups or classes. All member of a subgroup contain common characteristics.

Misunderstanding of term segmentation is caused by it's relation with 'classification', which is another data mining problem type. Classification assumes that there is a set of objects that belong to different classes, where some attributes or features characterize each class. The objective is to build classification model which assign correct class label to previously unseen and unlabeled objects (so called predictive modeling).

Classification and segmentation may introduce new type of data mining problems; it is a 'concept description problem'. It aims at an understandable description of concepts or classes. The purpose is not to develop complete model with high prediction accuracy, but to gain insights.

Another important problem type that occurs in a wide range of application is 'prediction'. The aim of prediction is to find the numerical value of the target attribute for unseen objects.

In close connection to the prediction is another problem type, so called 'dependency analyses'. It consists of finding a model that describes significant dependencies and associations between data items or events.

3. Proposed Solution

There are two types of information that need to be acquired from the Internet.

The first type is information concerning the future. In the process of selection, one more parameter must be acknowledged – what is the weather forecast for the route of the boat (truck) at the moment of his transition. If the weather forecast is bad (rain, storms) than the chance for the boat (truck) to go that way should be very small.

The second type is the past – in what condition is the boat (truck) that is being used. In case of an old boat (truck) chosen routes must be safer (deeper sea for boats or new roads for trucks) than the routes that would be chosen for a new boat (truck). There are some other things that could be considered for the past, such as status of the companies or the maintenance of the boats (trucks).

3.1 Internet Improved Roulette Wheel Selection

First ranking is done using the knowledge from the Internet. After this ranking, we can start forming the roulette wheel and perform the classical method of roulette wheel selection.

Improved algorithm:
Input: Population a
Output: population after selection a'

```
begin
  s_o:=O
  a:=population a after ranking using the knowledge from the Internet
  /* forming the roulette wheel */
  for i:= 1 to n do
    s_i:=s_{i-1}+f_i/n
  end
  I* simulation of throwing the marble */
  for i:= 1 to n do
    r:= random([0, s_n])
    a_i ' = a_k  for such k to be accomplished s_{k-1}<r<s_k
  end
  return a'
end
```

3.2 Internet Improved Rank Selection

Since rank selection is in many ways similar to roulette wheel selection, for improvement of this method we also propose that the knowledge from the Internet is used to rank the individuals before performing the classical rank selection algorithm.

Improved algorithm:
Input: population a, the degree of reproduction of the least fit individual r in interval [0,l]
Output: population after selection a"

```
begin
  a:= population a ranked using the knowledge from the Internet
  a' := population a sorted in ascending order by fitness value
  /* forming a roulette wheel */
  s_o:=O
  for i := 1 to n do
  s_i := s_{i-1} + p_s(a_i')  /* value p_s(a_i') is selection probability */
  end
  /* simulation of throwing the marble */
  for i: = 1 to n do
    r := random( [0,s_n])
    a_i ":= a_k'  for such k to be accomplished s_{k-1}<r<s_k
  end
  return a"
end
```

3.3 Internet Improved Stochastic Universal Sampling Selection

This method is also similar to roulette wheel so the individuals will be ranked and sorted with the usage of the Internet first and than the classical algorithm will be performed on that new population.

Improved algorithm:
Input: population a, the degree of reproduction of the least fit individual R in interval [0, l]
Output: population after selection a'

```
begin
  a:= population a after being ranked with the knowledge from the Internet
  sum := 0
  j := 1
  ptr := random([0,1])
  for i := 1 to n do
    sum := sum + R_i
    /* R_i - the degree of reproduction of the individual a_i */
    while (sum > ptr) do
      a_j':= a_i
      j:=j+1
      ptr: = ptr + 1
    end
  end
  return a'
end
```

3.4 Internet Improved Tournament Selection

This approach enables Internet to be utilized during the process of selecting the individuals.

In every step, after selecting N_{tour} random individuals we can sort them by using the Internet knowledge and then choose the winner among these individuals.

Improved algorithm:
Input: Population a (size of a is n), tournament size N_{tour},, $N_{tour} \in$ N
Output: Population after selection a' (size of a' is n)

```
begin
  for i := 1 to n do
    randomly choose N_tour individuals from a population a
    sort these individuals using the Internet knowledge
    a_i':=best fitted individual among N_tour sorted elements
  end
  return a'
end
```

This can also apply to fine grained tournament selection:
Input: Population a (size of a is n), wished average tournament size F_{tour}, $F_{tour} \in$ R
Output: Population after selection a' (size of a' is n)

```
begin
  F_tour^- := trunc( F_tour )
  F_tour^+ := trunc( F_tour ) + 1
  n^- := trunc ( n * ( 1 - ( t - trunc ( F_tour ) ) ) )
  n^+ := n - trunc ( n * ( 1 - ( t - trunc ( F_tour ) ) ) )
```

/* tournaments with size F_{tour}^{-} */
for i := 1 to n^{-} do
randomly choose F_{tour}^{-} individuals from population a
sort these individuals using the Internet knowledge
a_i' := best fitted item among F_{tour}^{-} sorted elements
end
/* tournaments with size F_{tour}^{+} */
for i := n^{-}+1 to n do
randomly choose F_{tour}^{+} individuals from population a
sort these individuals using the Internet knowledge
a_i' := best fitted item among F_{tour}^{+} sorted elements
end
return a'
end;

3.5 Other Improvements

We described how Internet knowledge can be used before (roulette wheel, rank, stochastic universal sampling) and during (tournament and fine grained tournament) the classical algorithm. In some cases it can be used after the selection algorithm. That can be done using the fitness function that would contain parameters about future and past knowledge (weather forecast, conditions of the boats or trucks, status of the companies, the history of some routes, the maintenance of the tankers, etc.). But how to define that fitness function is yet to be discovered.

4. Problems Concerning the Internet

To gather knowledge from the Internet is another problem we came upon. Information about weather forecast or the condition of the boats (trucks) is not necessarily formatted as text. It can be formatted as pictures or some applications. These kinds of information can not be used in genetic algorithms.

What we can use in genetic algorithms are just numbers that represent parameters that depend on this information. So, we need to have optimal association of numerical values to different semantic entities.

First, we need to have all the information in a form of text. Then we could associate a certain value to every word (storm, rain, snow, old boat, etc.).

But, like we mentioned above, to find these information on the Internet in a form of text is not so common.

The best ways of gathering information would be Semantic Web and Data Mining.

Semantic Web is a concept that enables better machine processing of information on the Web, by structuring documents written for the Web in such a way that they become understandable by machines. More about Semantic web can be found in [8].

Data Mining can be defined as an automated extraction of predictive information from different data sources. It is a powerful technology with great potential to help users focus on the most important information. More information about data mining can be found in [9].

5. Open Problems For Research

How to assign numerical values in interval [0.1] based on the Internet knowledge that is non-numeric, but symbolic or semantic?

What should be considered to create a fitness function for the past knowledge (status of the companies, conditions of the tankers, the history of some routes, the maintenance of the tankers, etc.)?

What should be considered to create a fitness function for the future knowledge (weather forecast, etc.)?

How should we define the fitness function? Some examples are:

Rnew = Rold * Kp * Kf (Rold is some known fitness function, for example Jaccard's Score, Kp is a parameter that depends of the past, Kf is the parameter that depends on the future).

Rnew = Rold * F(Kp) * F(Kf) (F(Kp) and F(Kf) are some predefined functions of the arguments Kp and Kf).

Rnew = Rold o1 F(Kp) o2 F(Kf) (o1 and o2 are predefined operators)

6. Conclusion

We need to find a way to make this information from the Internet understandable to the algorithm we are using (definition of the fitness function and numerical values for the Internet knowledge). This is still a part of our research.

References

[1] Gremlich, R., Hamfelt, A., Valkovsky, V., "Prediction of the optimal decision distribution for the traveling salesman problem,"

[2] Filipovic, V., Kratica, J., Tošić, D., Ljubić, I., "Fine Grained Tournament Selection for the Simple Plant Location Problem," *Proceedings of the 5th Online World Conference on Soft Computing in Industrial Applications WSC5*, pp. 152-158, September 2000.

[3] Filipovic, V., "Predlog poboljsanja operatora turnirske selekcije kod genetskih algoritama,"

[4] Wright, A., "Evolutionary computation, selection methods,"

[5] Sushil, J., L., "Modified GAs for TSPs,"

[6] Obitko, M., "Introduction to Genetic Algorithms,"

[7] Pohlheim, H., "Evolutionary Algorithms: Overview, Methods and Operators," Documentation for: Genetic and Evolutionary Algorithm Toolbox for use with Matlab (November 2001).

[8] Vujovic, Neuhold, E., Fankhauser, P., Nierderee, C., Milutinovic, V., "Semantic Web: A Brief Overview and IPSI Belgrade Research," *Annals Of Mathematics, Computing & Teleinformartics*, Vol 1, No 1, 2003, PP 65-70

[9] Radivojevic, Z., Cvetanovic, M., Milutinovic, V., "Data Mining: A Brief Overview and IPSI Belgrade Research," *Annals Of Mathematics, Computing & Teleinformatics*, Vol 1, No 1, 2003, PP 84-90