

Implementation of Evolutionary Algorithms in Different Methods of Research- A Analytical Approach with Selection, Recombination, Mutation, Reinsertion and Population Model

Dr. C. Anbalagan

Professor

KLU Business School, KL University, AP, India

e-mail dr.chinlakshanbu@gmail.com

Mr. T. Sugantha

Ph. D Research Scholar

Department of Geology, Periyar University, India

e-mail hydrosuganth@gmail.com

Abstract— Authors should focus on evolutionary Algorithm selection, Recombination, Mutation, Reinsertion and Population Model. Evolutionary Algorithms are the common term used for algorithms based on principles of nature (evolution, genetic). Evolutionary Algorithms contain genetic algorithms, evolution strategies, evolutionary programming and genetic programming. In computer science, evolutionary computation is a subfield of artificial intelligence (more particularly computational intelligence) that involves combinatorial optimization problems. Evolutionary computation uses iterative progress, such as growth or development in a population. This population is then selected in a guided random search using parallel processing to achieve the desired end. Such processes are often inspired by biological mechanisms of evolution. As evolution can produce highly optimized processes and networks, it has many applications in computer science. Here, simulations of evolution using evolutionary algorithms and artificial started with the work of Nils All Barricelli in the 1960s, and was extended by Alex Fraser, who published a series of papers on simulation of artificial selection.[1] Artificial evolution became a widely recognized optimization method as a result of the work of Ingo Rechenberg in the 1960s and early 1970s, who used evolution strategies to solve complex engineering problems. Genetic algorithms in particular became popular through the writing of John Holland. As academic interest grew, dramatic increases in the power of computers allowed practical applications, including the automatic evolution of computer programs.[4] Evolutionary algorithms are now used to solve multi-dimensional problems more efficiently than software produced by human designers, and also to optimise the design of systems.

Keywords- Evolutionary Algorithms - computer science-combinatorial- efficiently-. Genetic algorithms

I. INTRODUCTION

Different main schools of evolutionary algorithms have evolved during the last 40 years. Genetic algorithms, mainly developed in the USA by J. H. Holland Evolutionary strategies, developed in Germany by I. Rechenberg and H.-P. Schwefel Evolutionary programming each of these constitutes a different approach; however, they are inspired by the same principles of

natural evolution. Recombination produces new individuals in combining the information contained in two or more parents (parents - mating population). This is done by combining the variable values of the parents. Depending on the representation of the variables different methods must be used.

The methods for binary valued variables constitute special cases of the discrete recombination. These methods can all be applied to integer valued and real valued variables as well. By mutation individuals are randomly altered. These variations (mutation steps) are mostly small. They will be applied to the variables of the individuals with a low probability (mutation probability or mutation rate). Normally, offspring are mutated after being created by recombination.

For the definition of the mutation steps and the mutation rate two approaches exist:

- Both parameters are constant during a whole evolutionary run. Examples are methods for the mutation of real variables, and mutation of binary variables
- One or both parameters are adapted according to previous mutations. Examples are the methods for the adaptation of mutation step-sizes known from the area of evolutionary strategies.

II. EVOLUTIONARY ALGORITHMS OVERVIEW

- Selection
- Recombination
- Mutation
- Reinsertion
- Population models - parallel implementation of evolutionary algorithms
- Application of multiple/different strategies and competition between subpopulations

Evolutionary algorithms are stochastic search methods that mimic the metaphor of natural biological evolution. Evolutionary algorithms operate on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation.

Evolutionary algorithms model natural processes, such as selection, recombination, mutation, migration, locality and neighborhood. Figure shows the structure of a simple evolutionary algorithm. Evolutionary algorithms work on populations of individuals instead of single solutions. In this way the search is performed in a parallel manner.

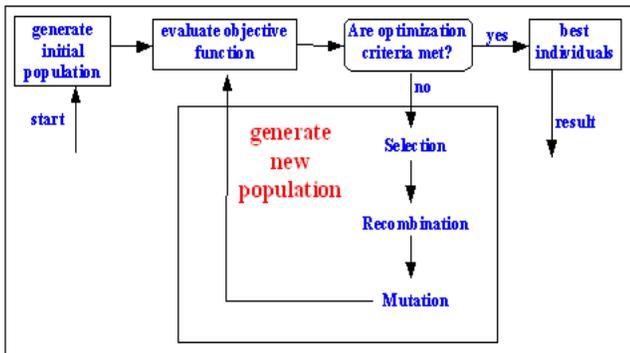


Figure-1: Structure of a single population evolutionary algorithm

At the beginning of the computation a number of individuals (the population) are randomly initialized. The objective function is then evaluated for these individuals. The first/initial generation is produced.

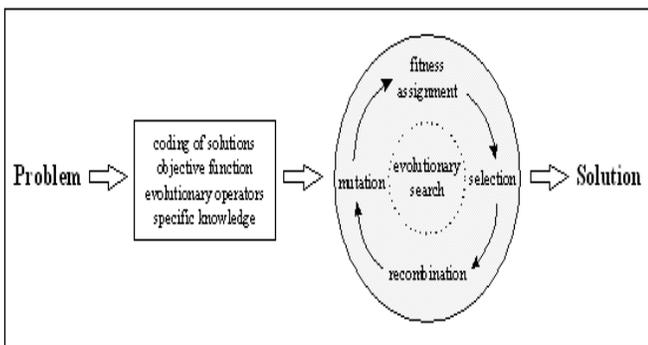


Figure-2: Problem solution using evolutionary algorithms

If the optimization criteria are not met the creation of a new generation starts. Individuals are selected according to their fitness for the production of offspring. Parents are recombined to produce offspring. All offspring will be mutated with a certain probability. The fitness of the offspring is then computed. The offspring are inserted into the population

replacing the parents, producing a new generation. This cycle is performed until the optimization criteria are reached.

Such a single population evolutionary algorithm is powerful and performs well on a wide variety of problems. However, better results can be obtained by introducing multiple subpopulations. Every subpopulation evolves over a few generations isolated (like the single population evolutionary algorithm) before one or more individuals are exchanged between the subpopulation. The multi-population evolutionary algorithm models the evolution of a species in a way more similar to nature than the single population evolutionary algorithm. Figure shows the structure of such an extended multi-population evolutionary algorithm.

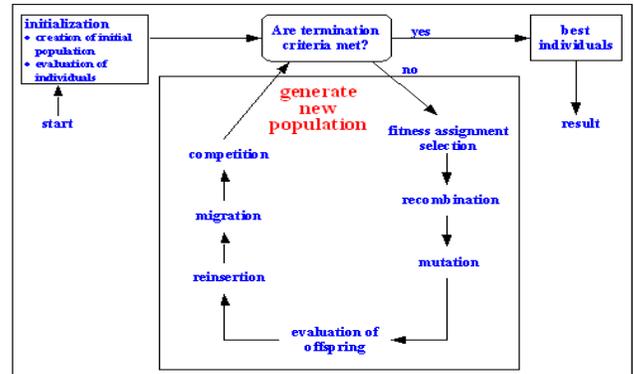


Figure-3: Structure of an extended multi-population evolutionary algorithm

From the above discussion, it can be seen that evolutionary algorithms differ substantially from more traditional search and optimization methods. The most significant differences are:

- Evolutionary algorithms search a population of points in parallel, not just a single point.
- Evolutionary algorithms do not require derivative information or other auxiliary knowledge; only the objective function and corresponding fitness levels influence the directions of search.
- Evolutionary algorithms use probabilistic transition rules, not deterministic ones.
- Evolutionary algorithms are generally more straightforward to apply, because no restrictions for the definition of the objective function exist.
- Evolutionary algorithms can provide a number of potential solutions to a given problem. The final choice is left to the user. (Thus, in cases where the particular problem does not have one individual solution, for example a family of pareto-optimal solutions, as in the case of multi-objective optimization and scheduling problems, then the evolutionary algorithm is potentially useful for identifying these alternative solutions simultaneously.)

The following sections list some methods and operators of the main parts of Evolutionary Algorithms. A thorough explanation of the operators will be given in the following chapters.

a) Selection

Selection determines which individuals are chosen for mating (recombination) and how many offspring each selected individual produces. The first step is fitness assignment by:

- Proportional fitness assignment or
- Rank-based fitness assignment
- Multi-objective ranking

The actual selection is performed in the next step. Parents are selected according to their fitness by means of one of the following algorithms:

- Roulette-wheel selection
- Stochastic universal sampling,
- Local selection,
- Truncation selection
- Tournament selection

b) Recombination

Recombination produces new individuals in combining the information contained in the parents (parents - mating population). Depending on the representation of the variables of the individuals the following algorithms can be applied:

i. All presentation

- Discrete recombination, (known from recombination of real valued variables), corresponds with uniform crossover, (known from recombination of binary valued variables),

ii. Real valued recombination

- Intermediate recombination
- Line recombination
- Extended line recombination

iii. Binary valued recombination

- Single-point / double-point / multi-point crossover
- Uniform crossover
- Shuffle crossover
- Crossover with reduced surrogate

For the recombination of binary valued variables the name 'crossover' is established. This has mainly historical reasons. Genetic algorithms mostly used binary variables and the name 'crossover'. Both notions (recombination and crossover) are equivalent in the area of Evolutionary Algorithms. For consistency, throughout this study the notion 'recombination' will be used (except when referring to specially named methods or operators).

c) Mutation

After recombination every offspring undergoes mutation. Offspring variables are mutated by small perturbations (size of the mutation step), with low probability. The representation of

the variables determines the used algorithm. Two operators are explained:

- Mutation operator for real valued variables
- Mutation for binary valued variables

d) Reinsertion

After producing offspring they must be inserted into the population. This is especially important, if less offspring are produced than the size of the original population. Another case is, when not all offspring are to be used at each generation or if more offspring are generated than needed. By a reinsertion scheme is determined which individuals should be inserted into the new population and which individuals of the population will be replaced by offspring.

The used selection algorithm determines the reinsertion scheme:

- Global reinsertion for all population based selection algorithm (roulette-wheel selection, stochastic universal sampling, truncation selection),
- Local reinsertion for local selection.

e) Population models - parallel implementation of evolutionary algorithms

The extended management of populations (population models) allows the definition of extensions of Evolutionary Algorithms. These extensions can contribute to an increased performance of Evolutionary Algorithms.

The following extensions can be distinguished:

- Global model
- Local model (diffusion model, neighborhood model, fine grained model)
- Regional model (migration model, island model, coarse grained model)

f) Application of multiple/different strategies and competition between subpopulations

Based on the regional population model the application of multiple different strategies at the same time is possible. This is done by applying different operators and parameters for each subpopulation. For an efficient distribution of resources during an optimization competing subpopulations are used.

- Application of multiple strategies
- Competition between subpopulations,

These extensions of the regional population model contribute to an increased performance of Evolutionary Algorithms, especially for large and complex real-world applications.

III. EVOLUTIONARY ALGORITHMS SELECTION

In selection the offspring producing individuals are chosen. The first step is fitness assignment. Each individual in the selection pool receives a reproduction probability depending on the own objective value and the objective value of all other individuals in the selection pool. This fitness is used for the actual selection step afterwards.

Throughout this section some terms are used for comparing the different selection schemes. The definitions of these terms follow

- **Selective pressure:** probability of the best individual being selected compared to the average probability of selection of all individuals
- **Bias:** absolute difference between an individual's normalized fitness and its expected probability of reproduction
- **Spread:** range of possible values for the number of offspring of an individual
- **Loss of diversity:** proportion of individuals of a population that is not selected during the selection phase
- **Selection intensity:** expected average fitness value of the population after applying a selection method to the normalized Gaussian distribution
- **Selection variance:** expected variance of the fitness distribution of the population after applying a selection method to the normalized Gaussian distribution

a) Rank-based fitness assignment

In rank-based fitness assignment, the population is sorted according to the objective values. The fitness assigned to each individual depends only on its position in the individuals rank and not on the actual objective value.

Rank-based fitness assignment overcomes the scaling problems of the proportional fitness assignment. (Stagnation in the case where the selective pressure is too small or premature convergence where selection has caused the search to narrow down too quickly.) The reproductive range is limited, so that no individuals generate an excessive number of offspring. Ranking introduces a uniform scaling across the population and provides a simple and effective way of controlling selective pressure.

Rank-based fitness assignment behaves in a more robust manner than proportional fitness assignment and, thus, is the method of choice.

b) Linear ranking

Consider N_{ind} the number of individuals in the population, Pos the position of an individual in this population (least fit individual has $Pos=1$, the fittest individual $Pos=N_{ind}$) and SP the selective pressure. The fitness value for an individual is calculated as:

Linear ranking:

$$Fitness(Pos) = 2 - SP + 2 \cdot (SP - 1) \cdot \frac{(Pos - 1)}{(N_{ind} - 1)}$$

(3-1)

Linear ranking allows values of selective pressure in [1.0, 2.0].

Non-linear ranking:

A new method for ranking using a non-linear distribution was introduced in [Poh95]. The use of non-linear ranking permits higher selective pressures than the linear ranking method.

Non-linear ranking:

$$Fitness(Pos) = \frac{N_{ind} \cdot X^{Pos-1}}{\sum_{i=1}^{N_{ind}} X^{i-1}} \quad (3-2)$$

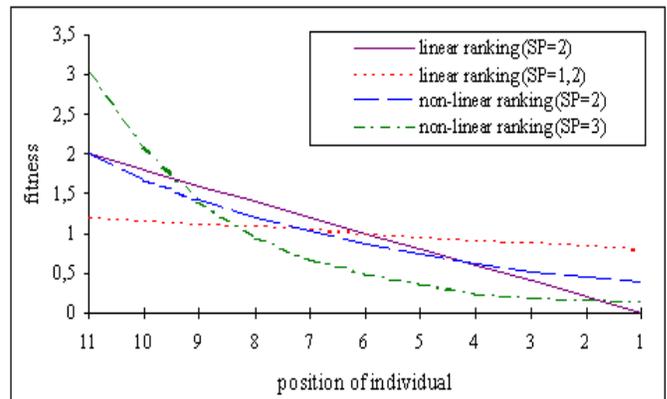
X is computed as the root of the polynomial:

$$0 = (SP - N_{ind}) \cdot X^{N_{ind}-1} + SP \cdot X^{N_{ind}-2} + \dots + SP \cdot X + SP \quad (3-3)$$

Non-linear ranking allows values of selective pressure in [1, $N_{ind} - 2$].

Comparison of linear and non-linear ranking:

Figure compares linear and non-linear ranking graphically.



Graph-1: Fitness assignment for linear and non-linear ranking

The probability of each individual being selected for mating depends on its fitness normalized by the total fitness of the population.

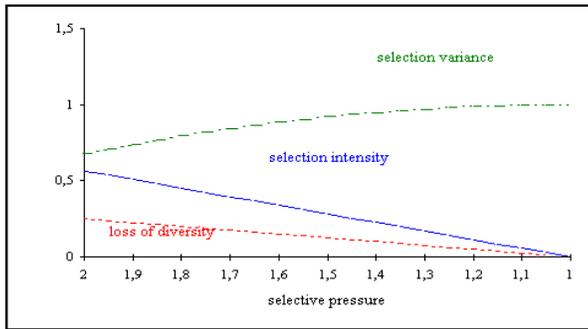
Table contains the fitness values of the individuals for various values of the selective pressure assuming a population of 11 individuals and a minimization problem.

	fitness value (parameter: selective pressure)				
	linear ranking		No rankin g	non-linear ranking	
object ive value	2.0	1.1	1.0	3.0	2.0
1	2.0	1.10	1,0	3.00	2.00
3	1.8	1.08	1,0	2.21	1.69
4	1.6	1.06	1,0	1.62	1.43
7	1.4	1.04	1,0	1.99	1.21
8	1.2	1.02	1,0	0.88	1.03
9	1.0	1.00	1,0	0.65	0.87
10	0.8	0.98	1,0	0.48	0.74
15	0.6	0.96	1,0	0.35	0.62
20	0.4	0.94	1,0	0.26	0.53
30	0.2	0.92	1,0	0.19	0.45
95	0.0	0.90	1,0	0.14	0.38

Table-1: Dependency of fitness value from selective pressure

Analysis of linear ranking:

In an analysis of linear ranking selection can be found.



Graph-2: Properties of linear ranking

Selection intensity

$$SelInt_{LinRank}(SP) = (SP - 1) \cdot \frac{1}{\sqrt{\pi}} \quad (3-4)$$

Loss of diversity

$$LossDiv_{LinRank}(SP) = \frac{(SP - 1)}{4} \quad (3-5)$$

Selection variance

$$SelVar_{LinRank}(SP) = 1 - \frac{(SP - 1)^2}{\pi} = 1 - SelInt_{LinRank}(SP)^2 \quad (3-6)$$

IV. SOLUTION OF SELECTION AND MODEL

a) Multi-objective Ranking

Where proportional and rank-based fitness assignment is concerned it is assumed that individuals display only one objective function value. In many real world problems, however, there are several criteria which have to be considered

in order to evaluate the quality of an individual. Only on the basis of the comparison of these several criteria (thus multi-objective) can a decision be made as to the superiority of one individual over another. Then, as in single-objective problems, an order of individuals within the population can be established from these reciprocal comparisons - multi-objective ranking. After this order has been established the single-objective ranking methods from the subsection can be used to convert the order of the individuals to corresponding fitness values.

Multi-objective fitness assignment (and with it multi-objective optimization) are such an important aspect, that an own chapter contains the description of its different aspects.

b) Roulette wheel selection

The simplest selection scheme is roulette-wheel selection, also called stochastic sampling with replacement. This is a stochastic algorithm and involves the following technique:

The individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness. A random number is generated and the individual whose segment spans the random number is selected. The process is repeated until the desired number of individuals is obtained (called mating population). This technique is analogous to a roulette wheel with each slice proportional in size to the fitness.

Table shows the selection probability for 11 individuals, linear ranking and selective pressure of 2 together with the fitness value. Individual 1 is the most fit individual and occupies the largest interval, whereas individual 10 as the second least fit individual has the smallest interval on the line (see figure). Individual 11, the least fit interval, has a fitness value of 0 and get no chance for reproduction

Number of individual	1	2	3	4	5	6	7	8	9	10	11
fitness value	2	1.8	1.6	1.4	1.2	1	0.8	0.6	0.4	0.2	0
selection probability	0.2	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0	0	0

Table-2: Selection probability and fitness value

For selecting the mating population the appropriate number of uniformly distributed random numbers (uniform distributed between 0.0 and 1.0) is independently generated.

sample of 6 random numbers: 0.81, 0.32, 0.96, 0.01, 0.65, 0.42

Figure shows the selection process of the individuals for the example in table together with the above sample trials.

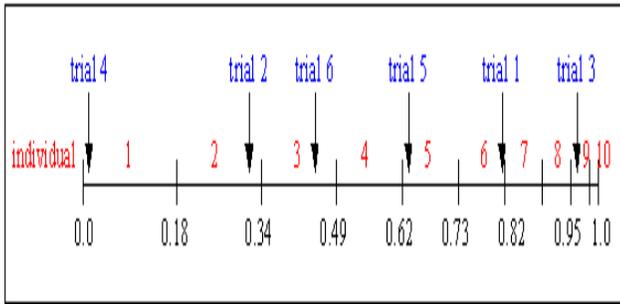


Table-3: Roulette-wheel selection

After selection the mating population consists of the individuals:

1, 2, 3, 5, 6, 9

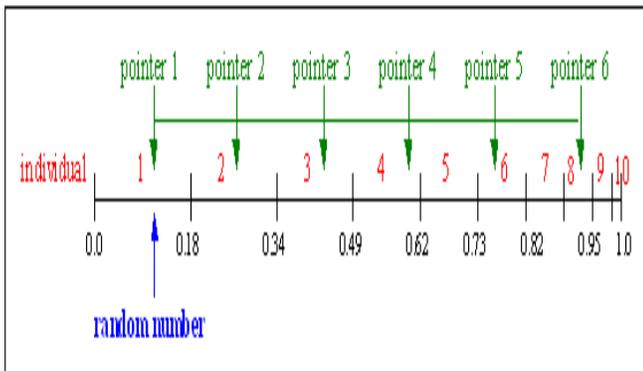
The roulette-wheel selection algorithm provides a zero bias but does not guarantee minimum spread.

c) Stochastic universal sampling

Stochastic universal sampling provides zero bias and minimum spread. 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 N Pointer the number of individuals to be selected, then the distance between the pointers are $1/N\text{Pointer}$ and the position of the first pointer is given by a randomly generated number in the range $[0, 1/N\text{Pointer}]$.

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

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



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

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

d) 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.

e) Local selection is part of the local population model

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).

The structure of the neighborhood can be:

Linear: full ring, half ring (see Figure)

Two-dimensional:

- full cross, half cross (see Figure , left)

- full star, half star (see Figure , right)

Three-dimensional and more complex with any combination of the above structures.

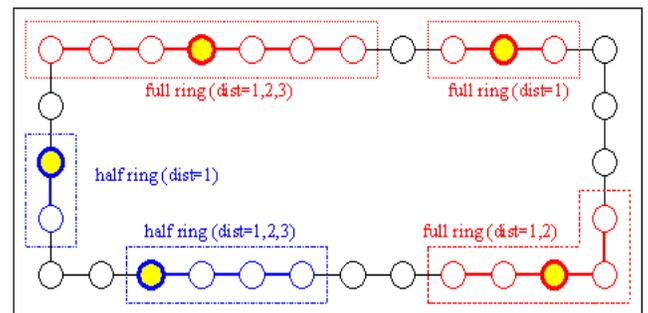
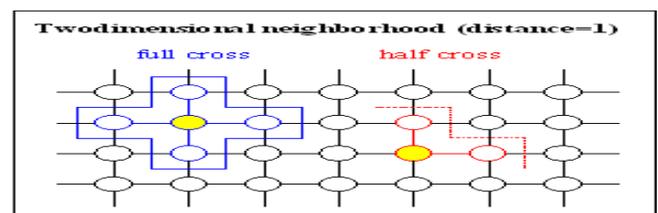
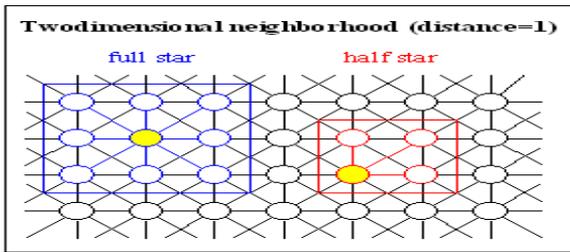


Figure-4: Linear neighborhood: full and half ring

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

Two-dimensional neighborhood; left: full and half cross, right: full and half star





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

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. Similar results were drawn from simulations in [VBS91]. 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.

structure of selection	Distance	
	1	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-4 : Number of neighbors for local selection

f) Truncation selection

Compared to the previous selection methods modeling natural selection truncation selection is an artificial selection method. It is used by breeders for large populations/mass selection. In truncation selection individuals are sorted according to their fitness. Only the best individuals are selected for parents. These selected parents produce uniform at random offspring. The parameter for truncation selection is the truncation threshold *Trunc*. *Trunc* indicates the proportion of the population to be selected as parents and takes values ranging from 50%-10%. Individuals below the truncation threshold do not produce offspring. The term selection intensity is often used in truncation selection. Table shows the relation between both.

truncation threshold	1%	10%	20%	40%	50%	80%
----------------------	----	-----	-----	-----	-----	-----

selection intensity	2.66	1.76	1.2	0.97	0.8	0.34
---------------------	------	------	-----	------	-----	------

Table-5: Relation between truncation threshold and selection intensity

V. PANALYSIS OF TRUNCATION SELECTION

In [BT95] an analysis of truncation selection can be found. The same results have been derived in a different way in [CK70] as well.

Selection intensity

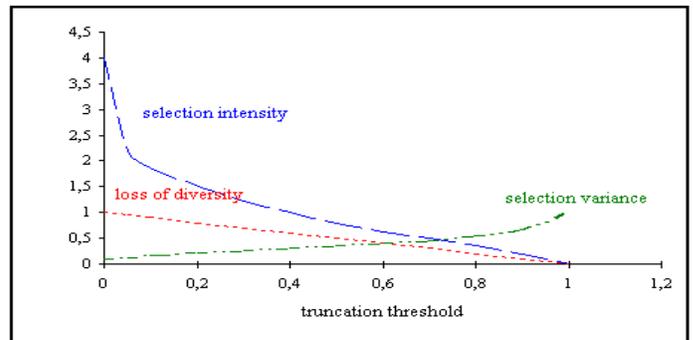
$$SelInt_{Truncation}(Trunc) = \frac{1}{Trunc} \cdot \frac{1}{\sqrt{2 \cdot \pi}} \cdot e^{-\frac{f^2}{2}} \quad (3-7)$$

Loss of diversity

$$LossDiv_{Truncation}(Trunc) = 1 - Trunc \quad (3-8)$$

Selection variance

$$SelVar_{Truncation}(Trunc) = 1 - SelInt_{Truncation}(Trunc) \cdot (SelInt_{Truncation}(Trunc) - f_c) \quad (3-9)$$



Graph-3: Properties of truncation selection

a) Tournament selection

In tournament selection a number *Tour* of individuals is chosen randomly from the population and the best individual from this group is selected as parent. This process is repeated as often as individuals must be chosen. These selected parents produce uniform at random offspring. The parameter for tournament selection is the tournament size *Tour*. *Tour* takes values ranging from 2 to *Nind* (number of individuals in population). Table and figure show the relation between tournament size and selection intensity.

tournament size	1	2	3	5	10	30
selection intensity	0	0.56	0.85	1.15	1.53	2.04

Table-6: Relation between tournament size and selection intensity

b) Analysis of tournament selection

In an analysis of tournament selection can be found.

Selection intensity

$$SelInt_{Tournament}(Tour) \approx \sqrt{2 \cdot (\ln(Tour) - \ln(\sqrt{4.14 \cdot \ln(Tour)}))}$$

(3-10)

Loss of diversity

$$LossDiv_{Tournament}(Tour) = Tour^{\frac{-1}{Tour-1}} - Tour^{\frac{-Tour}{Tour-1}}$$

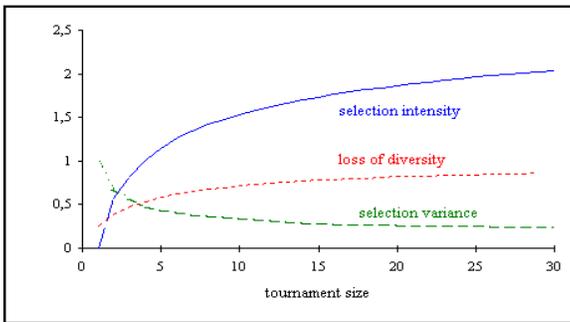
(3-11)

(About 50% of the population are lost at tournament size Tour=5).

Selection variance

$$SelVar_{Tournament}(Tour) \approx \frac{0.918}{\ln(1.186 + 1.328 \cdot Tour)}$$

(3-12)



Graph-4: Properties of tournament selection

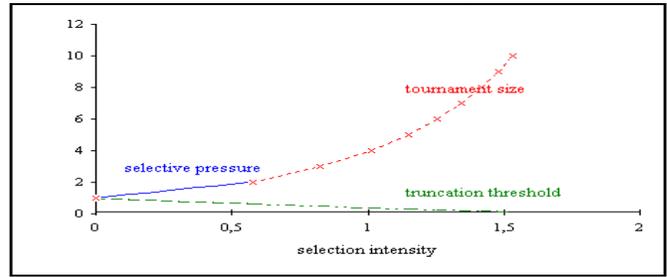
VI. COMPARISON OF SELECTION SCHEMES

As shown in the previous sections of this chapter the selection methods behave similarly assuming similar selection intensity.

a) Selection parameter and selection intensity

Figure shows the relation between selection intensity and the appropriate parameters of the selection methods (selective pressure, truncation threshold and tournament size). It should be stated that with tournament selection only discrete values can be assigned and linear ranking selection allows only a smaller range for the selection intensity.

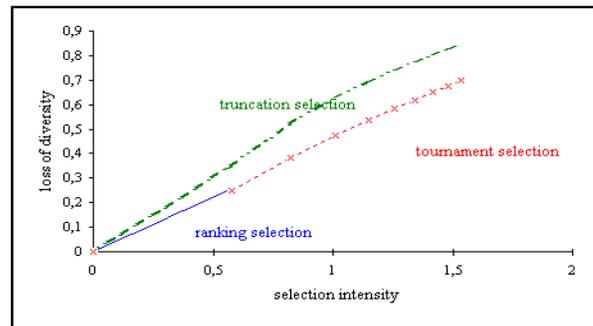
However, the behavior of the selection methods is different. Thus, the selection methods will be compared on the parameters loss of diversity (figure) and selections variance (figure) on the selection intensity.



Graph-5: Dependence of selection parameter on selection intensity

b) Loss of diversity and selection intensity

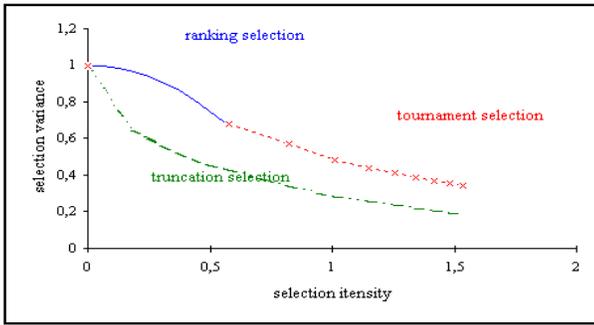
Truncation selection leads to a much higher loss of diversity for the same selection intensity compared to ranking and tournament selection. Truncation selection is more likely to replace less fit individuals with fitter offspring, because all individuals below a certain fitness threshold do not have a probability to be selected. Ranking and tournament selection seem to behave similarly. However, ranking selection works in an area where tournament selection does not work because of the discrete character of tournament selection.



Graph-6: Dependence of loss of diversity on selection intensity

c) Selection variance and selection intensity

For the same selection intensity truncation selection leads to a much smaller selection variance than ranking or tournament selection. As can be seen clearly ranking selection behaves similar to tournament selection. However, again ranking selection works in an area where tournament selection does not work because of the discrete character of tournament selection. In [BT95] was proven that the fitness distribution for ranking and tournament selection for SP=2 and Tour=2 (SelInt=1/sqrt(pi)) is identical.



Graph-7: Dependence of selection variance on selection intensity

Conclusion:

The evolutionary algorithms presented in this discussion provide the basis for their application to systems of various problem classes. After the analysis of the system to be solved the user may select the appropriate evolutionary algorithm. An adaptation is only necessary if there are problem specific requirements. A subdivision into frequently occurring problem classes was undertaken (parameter optimization of real, integer and binary variables as well as sequence or ordering optimization). Details were provided at the beginning of the explanations which can be applied similarly to most of the problem categories (fitness assignment, selection, application of the regional population model with different strategies, competition). Thereafter, a compilation of the operators with robust parameters, which are specific to each problem class, were presented.

There are several possible solutions to problems with variables of different representations. According to each type of variable the advantages and disadvantages of the conversion and application of the different representations were addressed. During application the conversion of real variables into integer representation and its use for the optimization has proved successful and is thus recommended.

REFERENCES

[1] Ackley, D. H.: A connectionist machine for genetic hillclimbing. Boston: Kluwer Academic Publishers, 1987.
 [2] Ackermann, J.: Robuste Regelung. Berlin, Heidelberg, New York: Springer-Verlag, 1993.
 [3] Fogarty, T. C.: Evolutionary Computing. Proceedings of AISB Workshop on Evolutionary Computing 1996, Vol. 1143 of Lecture Notes in Computer Science, Berlin, Heidelberg, New York: Springer-Verlag, 1996.
 [4] Altenberg, L.: The Schema Theorem and Price's Theorem. In [FGA3], pp. 23-49, 1995. <http://dynamics.org/~altenber/PAPERS/STPT/>
 [5] Kinneer, K. E.: Advances in Genetic Programming. Cambridge: MIT Press, 1994.
 [6] Angeline, P. J. and Kinneer, K. E.: Advances in Genetic Programming II. Cambridge: MIT Press, 1996.
 [7] Spector, L., Langdon, W., O'Reilly, U.-M. and Angeline, P. J. (eds.): Advances in 8. Genetic Programming III. Cambridge: MIT Press, 1999.
 [8] T.: Optimal Mutation Rates in Genetic Search. In [ICGA5], pp. 2-8, 1993. <http://lumpi.informatik.uni-dortmund.de/people/baeck/papers/icga93.ps.Z>

[9] T.: Evolutionary Algorithms in Theory and Practice - Evolution Strategies, Evolutionary Programming, Genetic Algorithms. New York, Oxford: Oxford University Press, 1996.
 [10] T. and Hoffmeister, F.: Extended Selection Mechanisms in Genetic Algorithms. In [ICGA4], pp. 92-99, 1991.
 [11] T. and Schwefel, H.-P.: An overview of evolutionary algorithms for parameter optimization. Evolutionary Computation, 1(1), pp. 1-23, 1993.
 [12] T. and Schütz, M.: Intelligent Mutation Rate Control in Canonical Genetic Algorithms. In Ras, Z. W. and Michalewicz, M.: Foundation of Intelligent Systems. 9th International Symposium, ISMIS '96, pp. 158-167, Berlin: Springer-Verlag, 1996.
 [13] J. E.: Adaptive Selection Methods for Genetic Algorithms. In [ICGA1], pp. 101-111, 1985.
 [14] J. E.: Reducing Bias and Inefficiency in the Selection Algorithm. In [ICGA2], pp. 14-21, 1987.
 [15] Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning. Technical Report CMU-CS-94-163, Pittsburgh, Pennsylvania: School of Computer Science, Carnegie Mellon University, 1994.
 [16] Beyer, H.-G.: Toward a Theory of Evolution Strategies: On the Benefits of Sex - the Theory. Evolutionary Computation, 3(1), pp. 81-111, 1995.
 [17] Blickle, T. and Thiele, L.: A Comparison of Selection Schemes used in Genetic Algorithms (2. Edition). TIK Report No. 11, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zürich, Switzerland, 1995.
 [18] T.: Theory of Evolutionary Algorithms and Application to System Synthesis. Ph.D. thesis. TIK-Schriftenreihe Nr. 17., Zürich: vdf Verlag, 1997.
 [19] J. Endrikat, C. and Liu, S.: A Systematic Approach to State Feedback Controller Design for DC/DC Line-Side Traction Converters. Proc. EPE'95, Sevilla, Vol. 1, pp. 314-318, 1995.
 [20] J. and Wu, Z.: Symmetry Properties of Multi-Variable Control Systems. Technical Note, 25/96, Daimler Benz AG, 1996.
 [21] Booker, L.: Improving search in genetic algorithms. In [Dav87], pp. 61-73, 1987.
 [22] G. E. P.: Evolutionary operation: A method for increasing industrial productivity. In Journal of the Royal Statistical Society, C, 6(2), pp. 81-101, 1957.
 [23] [Bra72] Branin, F. K.: A widely convergent method for finding multiple solutions of simultaneous nonlinear equations. IBM J. Res. Develop., pp. 504-522, Sept., 1972.
 [24] [Bre62] Bremermann, H. J.: Optimization through evolution and recombination. In Yovits, M. C. et al.: Self-organizing systems. Washington, DC: Spartan Books, pp. 93-106, 1962.
 [25] [CEC99] Angeline, P. J. (ed.): Proceedings of the Congress on Evolutionary Computation. Piscataway, New Jersey, USA: IEEE Press, 1999.
 [26] [Cha95] Chambers, L. (ed.): Genetic Algorithm Applications Vol. I. New York: CRC Press, 1995.
 [27] [CS88] Caruana, R. A. and Schaffer, J. D.: Representation and Hidden Bias: Gray v. Binary Coding for Genetic Algorithms. In Fifth International Conference on Machine Learning, San 27.
 [28] Mateo, California, USA: Morgan Kaufmann Publishers, pp. 153-161, 1988.
 [29] [CES89] Caruana, R. A., Eshelmann, L. A. and Schaffer, J. D.: Representation and hidden bias II: Eliminating defining length bias in genetic search via shuffle crossover. In Sridharan, N. S. (ed.): Eleventh International Joint Conference on Artificial Intelligence, San Mateo, California,
 [30] USA: Morgan Kaufmann Publishers, Vol. 1, pp. 750-755, 1989.