Genetic Algorithms: Theory and Applications

Alexandru Agapie

National Institute of Microtechnology
PO Box 38-160, 72225, Bucharest, Romania
E-mail: agapie@oblio.imt.pub.ro

Abstract

Genetic Algorithms (GAs) are robust probabilistic algorithms for optimization, relying strongly on parallel computation. Their power comes from multi-point exploiting of the searching space, avoiding the stagnation in local optima. First we review the state of art in GA theory. Next, two illustrative original applications highlight the efficiency of GA on multi-parameter optimization tasks: on solving systems of fuzzy relational equations, and on optimizing the parameters involved in an economic forecasting task.

Keywords: Genetic Algorithm's convergence, Markov chain, Dependence with complete connections, System of fuzzy relational equations, Economic forecasting.

1 Introduction

Genetic Algorithms (GAs), as well as Evolution Strategies and Evolutionary Programming, are part of a steadily growing optimization methodology, usually referred to as Evolutionary Computation, (Bäck et al., 1997).

Introduced by Holland (1975), GAs are imitating - in an algorithmic formalization - the mechanics of natural genetics and natural selection by starting the search for an optima with an initial population of likely problem solutions and then evolve towards better and better solutions.

According to Goldberg (1989), GAs differ in some very fundamental ways from the classical optimization methods, namely:

- GAs work with a coding of the parameter set, not with the parameters themselves;
- GAs search from a population of points, not from a single point;
- GAs use objective function information, but not derivatives or other auxiliary knowledge;
- GAs use probabilistic transition rules, not deterministic ones.

A simple GA requires the definition of the following components:

1. a genetic representation of potential problem solutions,
2. a function verifying the fitness of the solution (called *objective/fitness function*),
3. *genetic operators*, like: *crossover, mutation, selection*,
4. some constant values for the parameters used by the algorithm (such as *population size*, *probability* of applying an operator etc.).

International Journal of Computing Anticipatory Systems, Volume 7, 2000
Edited by D. M. Dubois, CHAOS, Liège, Belgium, ISSN 1373-5411 ISBN 2-9600179-9-4

The natural parameters of the optimization problem - representing a potential solution - have to be coded as a finite-length string over some finite alphabet; within this paper we shall tackle the binary case only - as it is most representative, both by current practice and illustrative reasons. This string is called *chromosome* and its components are called *genes*. A *population* consists in a set of chromosomes. The iterates of a GA involve only copying, swapping and comparing binary strings - this simplicity of operations, together with their proved efficiency made GAs so attractive for optimization tasks.

A simple GA that works well on many practical problems will be described in the following. As stated above, this will be composed of three operators: selection (appealed also as *reproduction*, in some references), crossover and mutation.

Selection is the process in which individual strings are copied proportional to their objective function values: in case of a *maximization* problem, strings with a higher value have a higher probability of contributing by one/more offspring in the next generation.

Crossover is a two-step operator. First, members of the newly reproduced strings are mated at random and second, each pair of strings undergoes crossing over by swapping some segments of genes with same size and position. Mathematically, the crossover combines the features of two randomly selected chromosomes, making use of a template, yielding two offspring, like in the following example: let A and B be the initial chromosomes, T the template, and T'= *not* T:

A = [1 1 0 1 0], T = [0 0 0 1 1],

B = [0 0 1 0 0], T'= [1 1 1 0 0],

then, the offspring are given by the formulas:

C_1 = MIN {T', A}+ MIN {T, B}

C_2 = MIN {T, A}+ MIN {T', B}

where *MIN* and *negation* are w.r.t. the binary logic. We thus obtain:

C_1 = [1 1 0 0 0]

C_2 = [0 0 1 1 0]

The last operator, *mutation*, is performed on a bit by bit basis, by negating a single bit position of a string. For example, if we apply the mutation operator on the chromosome A, with the mutation site #3, we obtain the offspring:

C = [1 1 1 1 0].

In our case, the GA's parameters are the *population size* and the probabilities p_c and p_m

of applying crossover and mutation, respectively. The chromosome's length is rather depending on the function to be optimized than a parameter at our choice. Summing up, the algorithm can be represented in the following form:

Procedure Simple Genetic Algorithm

choose an initial population
determine the fitness of each individual
perform selection
repeat
    perform crossover, according to a mutation probability, p_c
    perform mutation, according to a mutation probability, p_m
    determine the fitness of each chromosome
    perform selection
until *some stopping criterion applies*

2 Theory of Genetic Algorithms

As announced in the previous section, we shall consider a general optimization task for the simple GA, that is:

$max\ \{f(x) \mid x \in \{0, 1\}^{\ell}\}$, with $f(x) > 0$ for all x, where ℓ is a (fixed) positive integer (1)

When the *schema theorem* - a probabilistic inequality for estimating the expected number of offspring generated by an instance of a particular string-structure, established in (Holland, 1975), and consecutively revised (Uesaka, 1995), (Wright, 1999) - proved to be insufficient for analyzing the GAs, most of the theoretical approaches moved onto the convergence theorems of stochastic processes, and especially to the ergodic theorems from Markov chain (MC) theory.

In this regard, the first attempts were concerning the theory of finite, homogenous MC only (Goldberg et al., 1987; Aarts et al., 1989; Eiben et al. 1990; Horn, 1993; Suzuki, 1993; Rudolph, 1994), but they evolved also to the study of inhomogeneous (Davis et al., 1993), or infinite models (Qi et al., 1994). Within this section we shall survey the main convergence results for the *finite, homogenous* algorithm (w.r.t. its MC model) on the optimization problem (1), indicating also the major flows of this theory and some future research directions. For an extensive tour d'horizon on finite MC results in Evolutionary Computation we recommend the recent paper of Rudolph (1998) and the references within.

The history of finite, homogeneous MC modeling of GAs originated in 1987, with the simple model introduced by Goldberg et al. (1987). Aarts et al. (1989) proved the first sufficient condition for the GA convergence; despite its unpolished formalism (based on basic probability theory only) that result was holding the essence of all further convergence theorems. Nix et al. (1992) introduced a simplified computational

framework for the GA's model, relying on non-negative matrices and combinatorics, while, extrapolating the simulated annealing theory onto GAs, Davis et al. (1991, 1993) provided a complete formalization of the genetic operators, including both the homogeneous and inhomogeneous Markovian case. They made use of the Perron-Frobenius and ergodicity theorems associated to non-negative matrices and finite MCs, but they could not avoid the primitive form for the transition matrix, yielding non-convergence results only. This conclusion was independently drawn by Fogel (1994-1995) – by proving the absorption of the canonical algorithm into the set of uniform populations.

The matrix analysis of GAs came to a head when Rudolph (1994) proved that a canonical GA does not converge, but its *elitist* variant does (by 'elitist' we denote an algorithm which maintains inside the current population the best solution found so far, during its whole previous evolution). This result enlightened the ergodic theorem for reducible stochastic matrix, which made the difference between this paper and the previous similar works. This basic convergence theorem was revisited later – first by improving the accuracy and generality of the original approach (Rudolph, 1997), and second by relaxing the positive assumption on the mutation matrix (Agapie, 1998a). Finally, the convergence of a *general* GA modeled as a finite homogeneous MC showed to be equivalent to the condition below - satisfied in particular by *elitist* GAs, but not exclusively (Agapie, 1998b):

Each sub-optimal state is inessential

where 'sub-optimal' means 'worse than the global optima', and 'inessential' means that 'there is at least an one-way path from that state to another' in the associated Markov transition matrix.

Despite their indubitable correctness, we can not omit a major weakness of the homogeneous convergence results proved up to this moment: they are too general. Practically, these theorems make no difference between *elitist GA* and *elitist random walk*, for example. Both are convergent under the circumstances; but this does not correspond to the real case, where GA performs better. On the other hand, the Markovian models designed up to this moment were not handling the *'premature convergence'* (that is, stagnation of the algorithm in local optima, with all chromosomes in the current population being copies of a single individual) of the GA, which still remains unsolved for common applications. The theoretical trick for surpassing this practical problem was relying on the mutation capability of escaping from uniform populations. Though correct in principle, the escape time from such a uniform population might be extremely large, yielding a non-convergent behavior for the GA on practical problems.

Analyzing the GA's stagnation, one will see this occurring due to the small mutation rate (p_m) commonly used in GA applications. Although, one would not recommend the usage of a large (initial) p_m – this would enhance the exploration capability of the algorithm, making stagnation impossible, but would also diminish its exploitation capability, making GA resemble to a simple random walk. One might suggest progressive decreasing p_m from a generation to another - from a theoretical point of view, this means passing to the inhomogeneous Markov model. But this

solution is not satisfactory either, as long as the generations index itself does not contain sufficient information in order to avoid stagnation.

At this point it is worth noticing the analysis on premature convergence of canonical GA provided in (Leung et al., 1997). Building on the concept of 'degree of population diversity', the authors conclude their approach by proving that premature convergence at a chromosome position - that is, the probability for allele loss to occur at that position - decreases with the population size, and increases with $|p_m - 1/2|$. The suggestion is straightforward, even if it goes beyond the MC framework of the canonical GA: Use the population diversity as a quantitative measure to prevent premature convergence by *adaptively varying mutation probability*.

However, this is the case in practice, where many of the GA designers allow p_m to be (self)-adaptive, related to the diversity of the current population or to the algorithm's stagnation during several generations. But these 'smart', *adaptive* GAs have not been modeled yet by means of stochastic processes - even if the adaptation principle receives a steadily growing attention in the Evolutionary Computation area, see e.g. (Bäck, 1992).

At a closer look, the previous MC-based approaches owe their deficits to the very definition of the MC: transition at time $n+1$ is conditioned by the chain's position at time n, alone. Referring to GA, this means that evolution from a population to another is conditioned by the current population only - in the homogeneous case, with the unpredictable (thus useless) aid of time changes - in the inhomogeneous case. Such a model seems to offer insufficient information for avoiding stagnation in sub-optimal points; consequently, one can expect *no* new results to the convergence issue from MCs. Making a step further in the theory of stochastic processes, we expect better theoretical results from the *random systems with complete connections*, a non-trivial extension of the Markovian dependence, accounting for a complete, rather than recent, history of a stochastic evolution (Iosifescu et al., 1990).

3 Applications

3.1 Solving Systems of Fuzzy Relational Equations

Following (Agapie et al., 1997b), we illustrate first the efficiency of GAs on the problem of solving a system of fuzzy relational equations.

The basic assumptions are the following: each equation has at least a solution, but the entire system has none. Solving this kind of systems is of real interest - on one hand because of their wide applicability (e.g. in fuzzy logic inference, in modelling fuzzy control systems) and on the other hand because of the lack of exact solutions in most real applications.

Let us consider a set of pairs of fuzzy sets (X_k, Y_k), for $k=1,6$

$X_k, Y_k : \{x_1, x_2, x_3, x_4\} \rightarrow [0,1]$ given by

$$X = \begin{bmatrix} 1.0 & 0.4 & 0.3 & 0.1 \\ 0.4 & 1.0 & 0.6 & 0.4 \\ 0.5 & 0.5 & 0.8 & 0.9 \\ 0.6 & 0.4 & 0.7 & 1.0 \\ 0.9 & 0.6 & 0.4 & 0.3 \\ 0.5 & 0.6 & 0.8 & 1.0 \end{bmatrix} \qquad Y = \begin{bmatrix} 0.5 & 0.5 & 0.7 & 1.0 \\ 0.3 & 0.3 & 1.0 & 0.9 \\ 0.4 & 0.2 & 0.2 & 0.6 \\ 0.5 & 0.6 & 1.0 & 0.3 \\ 0.4 & 0.3 & 0.5 & 0.7 \\ 1.0 & 0.6 & 0.3 & 0.2 \end{bmatrix}$$

We examine the system of induced equations $\{X_k \circ R = Y_k \ , \ k=1,6\}$. One can easily check that each equation, treated separately, has a solution, yet the whole system has no solution. Applying a classical (probabilistic) algorithm for obtaining an *approximate solution* for the system, the following fuzzy relation has been obtained:

$$R = \begin{bmatrix} 0.40 & 0.30 & 0.70 & 0.70 \\ 0.40 & 0.30 & 1.00 & 0.90 \\ 0.40 & 0.30 & 1.00 & 0.60 \\ 0.40 & 0.60 & 1.00 & 0.60 \end{bmatrix}$$

with the performance index calculated like an averaged Hamming distance between fuzzy sets $Q = 0.70$. Applying a GA - for each column of the matrix R - we obtained, after a relative small number of iterations ($\sim 10^3$, while the size of the searching space is $\sim 10^7$) the following matrix, with a better performance index: $Q = 0.60$.

$$R = \begin{bmatrix} 0.40 & 0.37 & 0.70 & 1.00 \\ 0.37 & 0.30 & 1.00 & 0.90 \\ 0.35 & 0.00 & 0.25 & 0.37 \\ 0.50 & 0.60 & 0.25 & 0.60 \end{bmatrix}$$

3.2 Economic Forecasting

The second GAs' application presented here is from a more difficult area: the one of time series forecasting, with case study of an economic time series (Agapie et al., 1997a).

When one deals with short-length, non-stationary time series with seasonal components, the statistical procedures or even the neural networks approaches may prove to be unsatisfactory. We present in this section an alternative method, based on Genetic Algorithms, stressing their applicability on prediction tasks, especially on those requireing a large number of parameters.

Without introducing the specific forecasting methodology (we only notice that the prediction methods we used are usually appealed as *Holt*, and *Holt-Winters*), we confine ourselves to illustrate the GAs' performance in economic forecasting by two examples of micro-economic time series.

In fig.1 is depicted a GA-forecast on a classical example, while in fig.2 is represented a concrete example from an Airlines company, with data corresponding to the year 1995. In both cases, the data was split in two parts: a training set and a test set – as is the standard procedure in machine learning, neural networks etc.

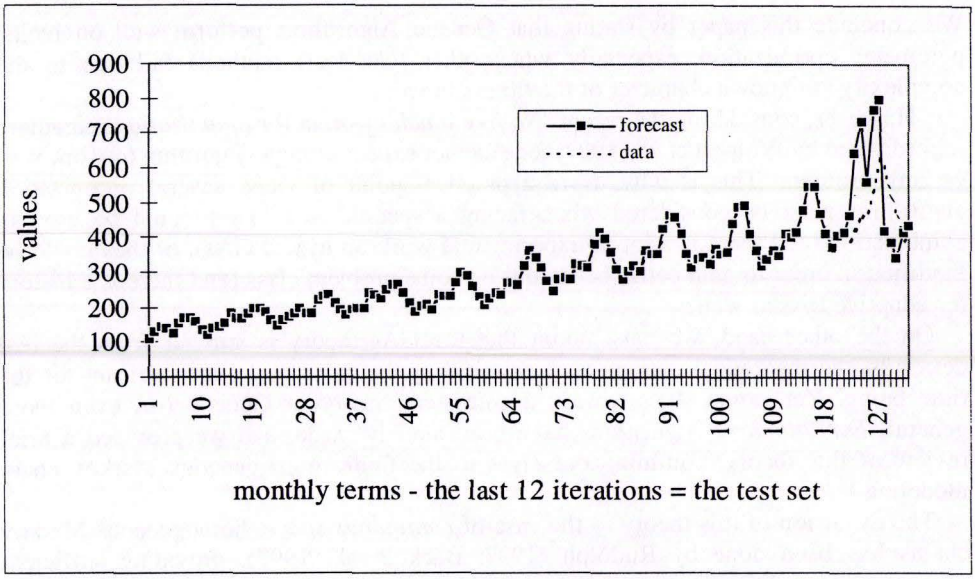

Fig. 1: GA forecast on a classical time series - © *1997 IEEE, reprinted by permission.*

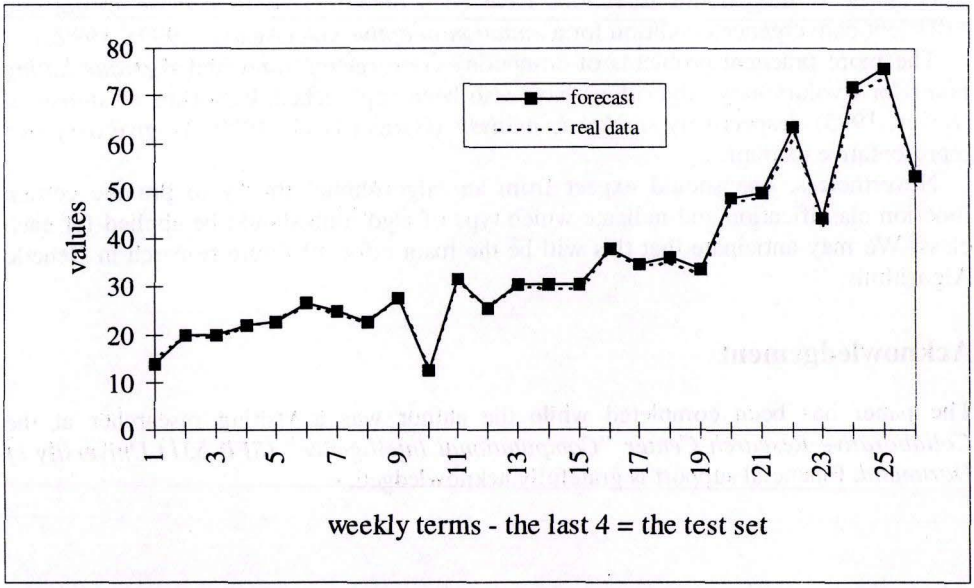

Fig. 2: GA forecast for an Airlines company - © *1997 IEEE, reprinted by permission.*

4 Conclusions

We conclude this paper by stating that Genetic Algorithms perform well on multi-parameter optimization, especially when other (classical) methods fail due to the complexity / unknown character of the fitness function.

However, considering the recent *'No free lunch theorem for optimization'* paradigm – introduced by Wolpert et al. (1997), one cannot expect a single algorithm to work well on any problem. This is why, from a practical point of view, several optimization algorithms must be considered when facing a specific task. They could be used in combination (and there is a lot of experimental work on hybrid GAs), or they could be modified in order to gain better behaviour on some problem class (and there is a history for adaptive EAs as well).

On the other hand, we must admit that the GAs theory is still far from the real demands, the theoretical outputs onto practical applications being insignificant for the time being. Yet, some steps toward a consistent theory of Genetic (or, even more general, *Evolutionary*) Algorithms have been already made, and we provided a brief review of this theory, confining ourselves to the finite, homogeneous Markov chain modeling.

The extension of this theory to the case of *continuous-space*, homogeneous Markov chains has been done by Rudolph (1997; Bäck et al., 1997), providing sufficient convergence conditions in terms of *'irreducible kernels'* (Nummelin, 1984).

Not far from the Markov theory, new results could come by introducing the *'random systems with complete connections'* (Iosifescu et al., 1990) as a new analysis tool for GAs; some results in this directions have been obtained, up to a close model and sufficient convergence condition for a *mutation-adaptive* GA (Agapie, 1999a, 1999b).

The more practical problems of computing *convergence rates* and *rigorous hitting times* for Evolutionary Algorithms have also been approached, by means of *statistical* (Beyer, 1995), respectively *stochastic analysis* (Garnier et al., 1999), to give only two representative examples.

Nevertheless, one should expect from an 'algorithmic' theory to provide certain function classification and indicate which type of algorithm should be applied for each class. We may anticipate that this will be the main effort of future research in Genetic Algorithms.

Acknowledgement

The paper has been completed while the author was a visiting researcher at the *Collaborative Research Center "Computational Intelligence" (SFB 531) University of Dortmund*. Financial support is gratefully acknowledged.

References

Aarts, E.H.L., Eiben, A.E., Van Hee, K.M. (1989). A General Theory of Genetic Algorithms, *Computing Science Notes*, Eindhoven University of Technology.

Agapie, A. (1998a). Genetic Algorithms: Minimal Conditions for Convergence, in Hao, J.-K., Lutton, E., Ronald, E., Schoenauer, M., Snyers, D. (Eds.), *Artificial Evolution, Third European Conference; selected papers / AE'97*, Springer, Berlin, pp.183-193.

Agapie, A. (1998b). Modeling Genetic Algorithms: From Markov Chains to Dependence with Complete Connections, in A.Eiben, T.Bäck, M.Schoenauer, H.-P.Schwefel (Eds.), *Parallel Problem Solving from Nature V*, Springer, pp.3-12.

Agapie, A. (1999a). Adaptive Genetic Algorithms – Modeling and Convergence, in *Proc. of Congress on Evolutionary Computation*, vol.1, IEEE Press, Piscataway, NJ, pp. 729-735.

Agapie, A. (1999b). Genetic Algorithms as Random Systems with Complete Connections, in *Proc. of Genetic and Evolutionary Computation Conference*, vol.1, Morgan Kaufman, San Francisco, CA, p.770.

Agapie, Ad. Agapie, A. (1997a). Forecasting the Economic Cycles Based on an Extension of the Holt-Winters Model. A Genetic Algorithms Approach, in *Proc. of IEEE/IAFE Int. Conf. on Comp. Intell. for Financial Eng.*, IEEE Press, pp.96-99.

Agapie, A., Giuclea, M. (1997b). Genetic Algorithms for Solving Systems of Fuzzy Relational Equations, in *Proc. Seventh Int. Fuzzy Systems Asociation's World Congress*, Prague, vol.2, pp.379-382.

Bäck, T., Fogel, D.B., Michalewicz, Z. (1997). *Handbook of Evolutionary Computation*, Oxford University Press.

Bäck, T. (1992). The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm, in *Parallel Problem Solving from Nature 2*, North Holland, Amsterdam, pp.85-94.

Beyer, H.-G. (1994). Toward a Theory of Evolution Strategies: The (μ, λ)-Theory, *Evolutionary Computation*, 2 (4), pp. 381-408.

Davis, T.E. (1991). Toward an extrapolation of the simulated annealing convergence theory onto the simple genetic algorithm, PhD thesis, Univ. of Florida, Gainesville.

Davis, T.E., Principe, J.C. (1993). A Markov Chain Framework for the Simple Genetic Algorithm, *Evolutionary Computation* 1, (3), pp. 269-288.

Fogel, D.B. (1994). Asymptotic convergence properties of genetic algorithms and evolutionary programming: Analysis and experiments, *Cybernetics and Systems*, 25 (3), pp. 389-407.

Fogel, D.B. (1995). *Evolutionary Computation – Toward a New Philosophy of Machine Intelligence,* IEEE Press.

Eiben, A.E., Aarts E.H.L., Van Hee, K.M. (1990). Global Convergence of Genetic Algorithms: a Markov Chain Analysis, in *Parallel Problem Solving from Nature*, Springer, Berlin, pp. 4-12.

Garnier, J., Kallel, L., Schoenauer, M. (1999). Rigorous Hitting Times for Binary Mutations, *Evolutionary Computation,* 7 (2), pp. 173-203.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading (MA): Addison Wesley.

Goldberg, D.E., Segrest, P. (1987). Finite Markov chain analysis of genetic algortihms, in: *Proc. Int. Conf. on Genetic Algorithms*, Hillsdale, NJ: Lawrence Elbraum, pp.1-8.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*, Ann. Arbor, Univ. of Michigan Press.

Horn, J. (1993). Finite Markov Chain Analysis of Genetic Algorithms with Niching, in *Proc. Fifth Int. Conf. on Genetic Algorithms*, Morgan Kaufman, San Mateo, CA, pp.110-117.

Iosifescu, M., Grigorescu, S. (1990). *Dependence with complete connections and its applications,* Cambridge Univ. Press.

Leung, Y., Gao, Y., Xu, Z.-B. (1997). Degree of Population Diversity - A Perspective on Premature Convergence in Genetic Algorithms and its Markov Chain Analysis, *IEEE Trans. on Neural Networks*, 8 (5), pp.1165-1176.

Nix, A.E., Vose, M.D. (1992). Modeling genetic algorithms with Markov chains, *Ann. Math. and Artificial Intelligence*, 5, pp.79-88.

Nummelin, E. (1984). *General irreducible Markov chains and non-negative operators,* Cambridge Univ. Press.

Rudolph, G. (1994). Convergence Analysis of Canonical Genetic Algorithms, *IEEE Trans. on Neural Networks,* 5 (1), pp.98-101.

Rudolph, G. (1997). *Convergence Properties of Evolutionary Algorithms,* Kovac.

Rudolph, G. (1998). Finite Markov Chain Results in Evolutionary Computation: A Tour d'Horizon, *Fundamenta Informaticae,* 35, IOS Press, pp. 67-89.

Qi, X., Palmieri, F. (1994). Theoretical Analysis of Evolutionary Algorithms, parts I-II, *IEEE Trans. on Neural Networks*, 1 (5), pp. 102-129.

Suzuki, J. (1995). A Markov Chains Analysis on Simple Genetic Algorithms, *IEEE Trans. on System, Man, and Cybernetics*, 25 (4), pp.655-659.

Wolpert, D.H., Macready, W.G. (1997). No free lunch theorems for optimization, *IEEE Trans. on Evolutionary Computation*, 1 (1), pp.67-82.

Wright, A.H. (1999). The Exact Schema Theorem, Techn. Rep., Univ. of Montana, Missoula.