Feasible Approaches to Convergence Results for Evolutionary Algorithms Part II: Runtime analysis of evolutionary algorithms and summary.

Stefan Droste* and Lothar M. Schmitt+

*LS Informatik 2, Universität Dortmund, 44221 Dortmund, Germany
stefan.droste@udo.edu

+School of Computer Science and Engineering, The University of Aizu
Aizu-Wakamatsu City, Fukushima Prefecture 965-8580, Japan
lothar@u-aizu.ac.jp

Abstract. Despite many successes of evolutionary algorithms (EAs) in real-world applications, theoretical knowledge in regard to these algorithms is still in its infancy. In this work, we discuss a number of approaches to theory for EAs in regard to strengths and weaknesses of statements for convergence-speed obtained with these methods. This includes the general convergence-analysis of a broad class of EAs in an arbitrary-fitness-function black-box scenario similar to the setting for the simulated annealing algorithm, and the runtime-analysis of specific EAs on limited classes of fitness-functions within the framework of asymptotic runtime-analysis for randomized algorithms.

We propose that a suitable merger of ideas put forward through the latter two types of convergence-analysis may yield substantial progress towards understanding convergence behavior of EAs. In particular, this may yield a unified theoretical framework for EAs as well as probabilistic estimates for runtimes of EAs used in real-world applications.

Introduction

This work continues the discussion begun in "L.M. Schmitt & S. Droste, *Feasible Approaches to Convergence Results for Evolutionary Algorithms. Part I: Introductory overview and analysis of scaled genetic algorithms.*" which is also published in these workshop proceedings and contains a longer introductory chapter.

* Stefan Droste was supported in part by the Deutsche Forschungsgemeinschaft through the Collaborative Research Center "*Computational Intelligence*" (SFB 531). This work was done while he was visiting The University of Aizu whose financial support is also kindly acknowledged.

II.1. Overview of Runtime Analysis

In the analysis of classical algorithms, one wants to determine the number of steps the algorithm makes until a solution to the problem instance (here: an optimum) is found. Since EAs are most often used in situations where the knowledge about the problem instance is very limited, there exists usually no criterion for stopping an EA that guarantees that an optimum has been found. Hence, one often assumes that the mixing-selection cycle of the EA is repeated forever, and one is interested in the number of steps until an optimum is attained for the first time. Consequently, one does not analyze a finite algorithm but an infinite random process.

Asymptotically exact estimates. The analysis of efficient deterministic or randomized algorithms would not be successful, if the runtime were to be analyzed in complete detail. Specifically designed and well-defined simplification techniques and assumptions are applied, that make the analysis much easier, but do not affect the correctness and significance of the results [38, 59]:

- Constant factors are not taken into account. Hence, *e.g.*, runtimes of n^2 and $3n^2$ are subsumed as functions of quadratic growth. This is motivated by the fact, that constant factors depend on many issues such as computer speed or implementation details that are out of the scope of regular analysis.
- One only cares about the fastest growing term, *i.e.*, $n^2 + n$ is simplified to n^2, since the quadratic term dominates the linear term for growing n.
- Since one deals with large n in many applications, one does not care about small n, *i.e.*, in many estimates one does not consider inputs with dimension bounded by some constant n_o.

These assumptions are the basis of the well-known O-calculus used for the analysis of algorithms (see [38] for the formal definition). This gives a precise mathematical foundation for the exactness of asymptotic runtime analysis results while allowing to simplify computations.

Homogeneous Markov chain models. Since the next step of the mixing-selection cycle of an EA, usually, only depends upon the actual population, it is possible to describe such EAs using Markov chains. If we model such an EA in exact fashion, then the state space of the Markov chain is the set of all possible populations. However, the size of this state space increases exponentially fast in the dimension (size) n of the problem instance and the size s of the population. Thus, it is often easier and necessary but also sufficient to consider a simplified version of the state space. One possibility would be to use the best, *i.e.*, fittest search point in the population to denote the state.

Even if one follows the latter plan for reduction of the size of the state space, the size still may grow exponentially fast in n. Thus, one tries to find an even smaller state space depending on the fitness function. A typical example for further state space reduction techniques is incorporated in the analysis[1] of an EA on the OneMax-function on bit-strings whose output is the number of ones

[1] See [21] for further details.

in its input. Since the position of the bits in a search point now has no relevance for the analysis, one just identifies a search point with the number of its ones.

The previous, very simple example can be extended by using a *potential function*, a common tool in the analysis of algorithms [12]: the potential function maps the state of the algorithm to a range of numbers where the optimal state of the algorithm is usually mapped to the maximal value of the potential function. If we can determine an upper bound for the range of values of the potential function, and a lower bound for the (expected) gain during a finite number of steps of the EA, then we can determine an upper bound for the (expected) number of steps until an optimum is found. This very flexible method was used, *e.g.*, in [21] where the expected runtime of the $(1+1)$ EA on linear functions was analyzed.

Simplification techniques for stochastic matrices. If one has found an appropriate state space $\{0,\ldots,N\}$, $N \in \mathbf{N}$, then the EA under consideration is modeled by a Markov chain $M = (M_{j,i})_{j,i\in\{1,\ldots,N\}}$ where $M_{j,i}$ denotes the probability of moving from state i to state j in one step (cycle) of the algorithm. A further simplification technique is to assume without loss of generality, that the optimal state is state N. Now, one only takes the transition probabilities into consideration in order to estimate the expected runtime of the EA asymptotically. Finding methods to estimate the expected runtime of the occurring Markov chains M is then the key for a runtime analysis of the corresponding EA.

Probably the most simple Markov chains are those that can only move from a state i to a state j with $j > i$, *i.e.*, they can only *improve* towards the optimal state N. The simplest method to obtain an upper bound for the expected runtime of such a process is to find a lower bound for the probability $M_{i+1,i}$ of making an improving step from state i to state $i+1$. The expected time to arrive at state $i+1$ is then at most $1/M_{i+1,i}$; and the expected time to reach state N starting from i is bounded from above by

$$\frac{1}{M_{i+1,i}} + \frac{1}{M_{i+2,i+1}} + \ldots + \frac{1}{M_{N,N-1}}.$$

It is intuitively clear that disregarding possible transitions from i to states $j \geq i+2$ makes the process under consideration only slower (see [18] for a proof). Application of this simple observation often yields asymptotically exact results, if the EA under consideration prefers small improvements which makes $M_{i+1,i}$ larger than the combined probability for all other transitions.

If the process under consideration is also allowed to make steps that reduce the fitness value, *i.e.*, move away from the optimal state N, then we cannot ignore any such steps in estimates for an upper bound of the runtime. The simplest case of such a process is a so-called $\{-1,0,1\}$-*process* which, by definition, can only move from state $i \in [0,N]$ to $\{i-1,i,i+1\} \cap [0,N]$. Let $E(T_{i+1,i})$ be the expected time to reach state $i+1$ under such a process for the first time when starting in state i. In the case of a $\{-1,0,1\}$-process, one can compute the expected

runtime very easily by using the following recursive formula for $E(T_{i+1,i})$:

$$E(T_{i+1,i}) = 1 + M_{i+1,i} + M_{i,i} \cdot E(T_{i+1,i}) + M_{i,i-1} \cdot (E(T_{i,i-1}) + E(T_{i+1,i}))\,.$$

This is based on the simple fact that the first step is followed either by no additional step (if state $i+1$ has been reached), by $E(T_{i+1,i})$ steps (if the process remains in state i), or by $E(T_{i,i-1}) + E(T_{i+1,i})$ steps (if the process has moved to state $i-1$). Recursively using the above equation and using that $E(T_{1,0}) = 1/M_{1,0}$, one gets a rather simple equation for $E(T_{i+1,i})$, *cf.* [20].

If the process under consideration is only allowed to make steps that reduce the fitness value by one, *i.e.*, steps from state i to state $j < i-1$ are not possible, then one can simplify the process by omitting all steps that improve the fitness value by more than one and get a $\{-1,0,1\}$-process whose runtime stochastically dominates the runtime of the original process. Upper bounds of the expected runtime for $(1+1)$ EAs were achieved in this way in [17–19].

If the process under consideration is possibly given by a fully positive stochastic matrix, then one has to apply a more carefully designed simplification procedure. As a first step, one would omit all improvements in the process by more than one, *i.e.*, one would replace all entries $M_{j,i}$ for $j > i+1$ by 0 and $M_{i,i}$ by $M_{i,i} + \sum_{j>i+1} M_{j,i}$. This yields a so-called $\leq$1-process which shall be denoted by $M^{(1)}$ in what follows below. $M^{(1)}$can only move from state i to a state in $[0, i+1]$.

The goal is now to find a $\{-1,0,1\}$-process $M^{(2)}$ whose runtime dominates the runtime of the $M^{(1)}$. This can be achieved by using the following ideas (see [18] for a formal description): Let us assume that we have a partition

$$S_1^i \cup S_0^i \cup \ldots \cup S_{-i}^i$$

of the set $\{-1,0,1\}^l$ of all sequences of length l of steps of $M^{(2)}$ starting from state i such that every sequence in S_k^i leads the process $M^{(2)}$ from state i to state $i+k$. We can also deal with sequences of varying length but omit this case for the sake of brevity. If the joint probability of all sequences in S_k^i is higher than $M_{i+k,i}^{(1)}$ for negative k and lower than $M_{i+k,i}^{(1)}$ for positive k, then the process $M^{(2)}$ has a "*a stronger tendency towards worse states*". By making these arguments more formal, we can prove that the runtime of $M^{(2)}$ stochastically dominates the runtime of $M^{(1)}$. This technique of partitioning transition sequences has been used in [18, 19] to obtain upper bounds for the expected runtime of EAs that are allowed to make steps that reduce the fitness value by more than one.

Special lower bound techniques. For lower bounds, in particular, superpolynomial lower bounds, one has to analyze the Markov process associated with an EA in much more detail. Often it is the case that the probability for moving away from the optimum is larger the closer the algorithm gets to the optimum. In the associated Markov chain model this is equivalent to

$$M_{0,i} + \cdots + M_{i-1,i} > \alpha \cdot (M_{i+1,i} + \cdots + M_{N,i})\,, \quad \alpha > 1,$$

if i is in a particular set of states close to the optimal state N. Depending upon the range of values of i where the latter inequality holds and the value of the fixed constant α, one can distinguish different cases where the runtime is super-polynomial or not.

In [18], it was shown that the runtime is super-polynomial with high probability if this inequality holds for all $i \in [N - \log(N)\beta(N), N]$ and $\alpha > \beta(N)$. Here, $\beta(N)$ is an unbounded function of N. Consequently, a region of more than logarithmic length $\log(N)$ cannot be successfully bridged, if the tendency α towards worse states is more than a constant. (Essentially, α is a lower bound for the quotient of the probability for worsening over the probability for improvement).

Similarly, it has been shown in [19] that the runtime of an EA is super-polynomial with high probability, if the above inequality holds for all $i \in [N - N^{\varepsilon}, N]$ and $\alpha > 1 + \gamma$ where $\varepsilon \in]0, 1[$ and $\gamma > 0$ are constant.

The proofs for the above results in [18, 19] rely heavily on Chernoff bounds (see [38]). Consider a random variable X which is the sum of m binary random variables $X_1, \ldots, X_m$. Intuitively, it becomes very unlikely that this random variable varies largely from its expected value, if m is large. Chernoff bounds formalize this by showing that the probability $P(X > (1 + c) \cdot E(X))$ decreases exponentially in $E(X)$ for a constant c. Hence, Chernoff bounds can be applied, in cases one is able to achieve the following two objectives: first, one must determine lower and upper bounds for the expected values of the binary random variables X_i; and secondly, one must show that the event one wants to show to be unlikely implies that X has to be by a constant factor > 1 larger than its expected value on the set of states pertaining to that event.

The methods presented here are only a small portion of a wide range of methods that can be used for the runtime analysis of EAs (see [65] for an overview). Often a combination of these methods has to be used to get asymptotically matching upper and lower bounds. But they all have in common that the Markov process representing the EA is simplified in a well-defined way in order to make it more easily analyzable.

Applications to general convergence analysis of EAs. The authors think that the methods discussed above can also be applied to the case of general convergence analysis of EAs (GAs) as described in section 2. It seems feasible that in the case of general convergence analysis, the underlying inhomogeneous Markov chain can be replaced by an asymptotically equivalent, simpler inhomogeneous Markov chain in the spirit of the above discussion. Thus through simplification of the mutation operator and the associated neighborhood structure and/or the fitness landscape, one is able to replace one inhomogeneous Markov chain by a simpler one, similar to the process of replacing a general homogeneous Markov chain by a more manageable one in the above discussion. This may, at least for certain sets of fitness functions, lead to a more successful and thorough analysis of scaled EAs (in particular, scaled GAs). Compare [25] in this regard, where the proof of the main result is first obtained for special landscapes.

II.2. Summary

Although there are quite a number of different approaches to EA theory, none of these has yet proven itself to be of outstanding value for practitioners. Hence, theory of EAs should be considered as being still in its very beginning stage.

In the present paper, the authors have argued why trying to merge two specific approaches to EA theory, *i.e.*, general convergence analysis using inhomogeneous Markov chains and asymptotic runtime analysis for EAs using homogeneous Markov chains, is a promising route of endeavor that may increase the knowledge about asymptotic behavior of certain families of EAs. Both approaches use well-founded mathematical models and assumptions that are established methods in recognized fields of science. Both approaches also have proven themselves as being successful in the analysis of EAs to a non-trivial extent.

The main ideas that have emerged in the discussion in this paper are:

1. Can variations of the techniques to obtain the mutation-flow inequality and the steady-state inequality in [50–52, 54–57] yield fruitful applications in the runtime analysis of EAs on certain limited classes of fitness functions investigated in [17–19, 21, 29, 23, 65, 66]?
2. Can variations of the techniques for simplification of homogeneous Markov chains that arise in the runtime analysis of EAs in, *e.g.*, [17–19, 21, 29, 23, 65, 66] yield fruitful applications in the general convergence analysis of scaled GAs. Thus, can those techniques be made applicable to the setting of inhomogeneous Markov chains considered in, *e.g.*, [50, 51]?
3. Is there an interplay of simplifications and improvements in regard to positively answering both questions in (1) and (2) by developing new mathematical techniques?

We have given overviews of the different methods and ideas used in global convergence analysis resp. asymptotic runtime analysis to emphasize common ground. We also have indicated where such common ground may yield new insights by applying ideas obtained through one of the approaches within the framework of the other. Future research in this regard must follow in order to make these ideas more explicit. Although one cannot hope to keep the strengths of both approaches (general applicability resp. exactness of runtime bounds), a merger of ideas associated with the two approaches might lead to quite general convergence results on non-trivial classes of fitness-functions that are interesting for practical applicability of the EAs under consideration. This may yield faster annealing schedules for certain scaled genetic algorithms on certain classes of fitness-functions while preserving asymptotic convergence to global optima. This may also lead to distinctions between, *e.g.*, polynomial *vs.* super-polynomial convergence time for certain EAs on distinct classes of problem instances.

References

1. E.H.L. Aarts, P.J.M. van Laarhoven: Simulated Annealing: An Introduction. *Statist. Neerlandica* **43** (1989), pp. 31–52

2. H. Aytug, G.J. Koehler: Stopping Criteria for Finite Length Genetic Algorithms. *INFORMS J. on Computing* **8** (1996), pp. 183–191
3. H.-G. Beyer, H.-P. Schwefel, I. Wegener: How to analyse evolutionary algorithms. *Theoret. Comput. Sci.* **287** (2002), pp. 101–130
4. H.J. Bremermann, J. Rogson, S. Salaff: Global Properties of Evolution Processes. In: H.H. Pattee *et al.* (eds.). *Natural Automata and Useful Simulations*. Spartan Books, Washington, DC, USA (1966), pp. 3–42
5. E. Cantu-Paz *et al.* (eds.): *Proc. GECCO 2003*, July 12–16, 2003, Chicago, Il, USA. LNCS 2723, Springer Verlag, Berlin, Germany (2003)
6. O. Catoni: *Large Deviations for Annealing*. Ph.D. Dissertation, Université Paris XI, Paris, France (1990)
7. O. Catoni: Sharp Large Deviations Estimates for Simulated Annealing Algorithms. *Ann. de l'Inst. Henri Poincaré. Prob. & Stat.* **27** (1991), pp. 291–383
8. O. Catoni: Applications of Sharp Large Deviations Estimates to Optimal Cooling Schedules. *Ann. de l'Inst. Henri Poincaré. Prob. & Stat.* **27** (1991), pp. 493–518
9. O. Catoni: Rough Large Deviations Estimates for Simulated Annealing — Application to Exponential Schedules. *Ann. of Probability* **20** (1992), pp. 1109–1146
10. R. Cerf: An Asymptotic Theory for Genetic Algorithms. In: J.M. Alliot *et al.* (eds.). *Artificial Evolution. European Conference AE 95*, Brest, France, September 4–6, 1995, selected papers. LNCS 1063, Springer Verlag, Berlin, Germany (1996), pp. 37–53
11. R. Cerf: Asymptotic Convergence of Genetic Algorithms. *Adv. in Appl. Probability* **30** (1998), pp. 521–550
12. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein: *Introduction to Algorithms*. MIT Press, Boston, MA, USA (2001)
13. T.E. Davis: *Toward an Extrapolation of the Simulated Annealing Convergence Theory onto the Simple Genetic Algorithm*. Ph.D. Dissertation, University of Florida (1991)
14. T.E. Davis, J.C. Principe: A Simulated Annealing-like Convergence Theory for the Simple Genetic Algorithm. In: R.K. Belew, L.B. Booker (eds.), *Proc. 4*th *Int. Conf. on Genetic Algorithms '91*, Morgan Kaufmann Publ., San Francisco, CA, USA (1991), pp. 174–181
15. T.E. Davis, J.C. Principe: A Markov Chain Framework for the Simple Genetic Algorithm. *Evol. Comput.* **1** (1993), pp. 269–288
16. K. De Jong: Lecture on Analysis of Evolutionary Algorithms using Price's Theorem. In: H.-G. Beyer *et al.* (chairs). "*Theory of Evolutionary Computation 2004*", Schloß Dagstuhl, Saarland, Germany (2004)
17. S. Droste: Analysis of the (1 + 1) EA for a Dynamically Changing OneMax-Variant. In: *Proc. of the Congress on Evolut. Comput. (CEC 2002)*, May 12-17, 2002, Honolulu, HI, USA (2002), pp. 55–60
18. S. Droste: Analysis of the (1+1) EA for a Dynamically Bitwise Changing OneMax. In: [5], pp. 909–921
19. S. Droste: Analysis of the (1 + 1) EA for a Noisy OneMax. In: [42]
20. S. Droste, T. Jansen, I. Wegener: Dynamic Parameter Control in Simple Evolutionary Algorithms. In: W.N. Martin, W.M. Spears (eds.), *Proc. of FOGA 6*, July 21-23, 2000, Charlottesville, VA, USA. Morgan Kaufmann Publ., San Francisco, CA, USA (2000), pp. 275-294
21. S. Droste, T. Jansen, I. Wegener: On the analysis of the (1 + 1) evolutionary algorithm. *Theoret. Comput. Sci.* **276** (2002), pp. 51–81
22. M.R. Garey, D.S. Johnson: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, New York, NY, USA (1979)

23. O. Giel, I. Wegener: Evolutionary algorithms and the maximum matching problem. In: H. Alt, M. Habib (eds.), *Proc. 20th Symp. on Theoret. Aspects of Computer Sci. (STACS 2003)*, February 27 - March 1, 2003, FU Berlin, Germany. LNCS 2607, Springer Verlag, Berlin, Germany (2003), pp. 415-426
24. D.E. Goldberg: *Genetic Algorithms, in Search, Optimization & Machine Learning.* Addison-Wesley Publ., Boston, MA, USA (1989)
25. B. Hajec: Cooling Schedules for Optimal Annealing. *Math. of Operations Research* **13** (1988), pp. 311–329
26. J.H. Holland: *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI, USA(1975), Extended new edition, MIT Press, Cambridge, MA, USA (1992)
27. D.L. Isaacson, R.W. Madsen: *Markov Chains: Theory and Applications.* Prentice-Hall Publ., Upper Saddle River, NJ, USA (1961)
28. J. Jägersküpper: Analysis of a Simple Evolutionary Algorithm for Minimization in Euclidean Spaces. In: *Proc. of the 30th Int. Coll. on Automata, Languages, and Programming (ICALP 2003)*, June 30 - July 4, 2003, Eindhoven, The Netherlands. LNCS 2719, Springer Verlag, Berlin, Germany (2003), pp. 1068–1079
29. T. Jansen, I. Wegener: Real royal road functions — where crossover provably is essential. In: L. Spector, E.D. Goodman (eds.), *Proc. of GECCO 2001*, July 7–11, 2001, San Francisco, CA, USA. Morgan Kaufmann, San Francisco, CA, USA (2001), pp. 375–382
30. D.E. Knuth: *The Art of Computer Programming. Volume 1: Fundamental Algorithms.* Third edition. Addison-Wesley Publ., Boston, MA, USA (1997)
31. D.E. Knuth: *The Art of Computer Programming. Volume 2: Semi-Numerical Algorithms.* Third edition. Addison-Wesley Publ., Boston, MA, USA (1997)
32. D.E. Knuth: *The Art of Computer Programming. Volume 3: Sorting and Searching.* Second edition. Addison-Wesley Publ., Boston, MA, USA (1997)
33. J.A. Lozano, P. Larrañaga, M. Graña, F.X. Albizuri: Genetic Algorithms: bridging the Convergence Gap. *Theoret. Comput. Sci.* **229** (1999), pp. 23–39
34. T. Lux, S. Schornstein: Genetic learning as an Explanation of Stylized Facts of Foreign Exchange Markets. Manuscript, Universität Kiel, London School of Economics. PDF-file (2002), 1–26. `http://www.bwl.uni-kiel.de/vwlinstitute/gwrp/publications/publications.htm`
35. S.W. Mahfoud, D.E. Goldberg: A Genetic Algorithm for Parallel Simulated Annealing. In: R. Männer, B. Manderick (eds.), *Parallel Problem Solving from Nature*, **2**. Elsevier (1992), 301–310
36. S.W. Mahfoud, D.E. Goldberg: Parallel Recombinative Simulated Annealing: a Genetic Algorithm. *Parallel Computing* **21** (1995), 1–28
37. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller: Equations of State Calculations by Fast Computing Machines. *J. of Chemical Physics* **21** (1953), 1087–1091
38. R. Motwani, P. Raghavan: *Randomized Algorithms.* Cambridge University Press, Cambridge, United Kingdom (1995)
39. A.E. Nix, M.D. Vose: Modeling Genetic Algorithms with Markov Chains. *Ann. Math. Artif. Intell.* **5** (1992), pp. 79–88
40. G. Paun, G. Rozenberg, A. Salomaa: *DNA Computing: New Computing Paradigms.* Springer Verlag-g, Berlin, Germany (1998)
41. R. Poli: Exact Schema Theory for Genetic Programming and Variable-Length Genetic Algorithms with One-Point Crossover. *Genetic Programming and Evolvable Machines* **2** (2001), pp. 123–163

42. R. Poli *et al.* (chairs): *Proc. GECCO 2004*, June 26–30, 2004, Seattle, WA, USA. LNCS, Springer Verlag, Berlin, Germany ((2004)
43. A. Poli, M. Langdon: Schema Theory for Genetic Programming with One-Point Crossover and Point Mutation. *Evolut. Computation* **6** (1998), pp. 231–252
44. I. Rechenberg: *Evolutionsstrategie. Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Frommann-Holzboog Verlag, Stuttgart, Germany (1973)
45. G. Rudolph: Convergence Analysis of Canonical Genetic Algorithms. *IEEE Trans. on Neural Networks* **5** (1994), pp. 96–101
46. G. Rudolph: *Convergence Properties of Evolutionary Algorithms.* Verlag Dr. Kovac, Hamburg, Germany (1997)
47. H.H. Schaefer: *Banach Lattices and Positive Operators.* Die Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen **215**, Springer Verlag, Berlin, Germany (1974)
48. L.M. Schmitt, C.L. Nehaniv, R.H. Fujii: Linear Analysis of Genetic Algorithms. *Theoret. Comput. Sci.* **200** (1998), pp. 101–134
49. L.M. Schmitt, C.L. Nehaniv: The Linear Geometry of Genetic Operators with Applications to the Analysis of Genetic Drift and Genetic Algorithms using Tournament Selection. Lect. on Math. in the Life Sci. **26**, AMS, Providence, RI, USA (1999), pp. 147–166
50. L.M. Schmitt: Theory of Genetic Algorithms. *Theoret. Comput. Sci.* **259** (2001), pp. 1–61
51. L.M. Schmitt: Theory of Genetic Algorithms II — Models for genetic operators over the string-tensor representation of populations and convergence to global optima for arbitrary fitness function under scaling. *Theoret. Comput. Sci.* **310** (2004), pp. 181–231
52. L.M. Schmitt: Asymptotic Convergence of Scaled Genetic Algorithms to Global Optima —A gentle introduction to the theory. IN: A. Menon (ed.), *Frontiers of Evolutionary Computation.* Kluwer Ser. in Evol. Comput. (D.E. Goldberg, ed.), Kluwer, Dordrecht, The Netherlands (2003), 45p.
53. L.M. Schmitt: Modeling Genetic Algorithms from a Linear Operator Point of View — A Survey of Recent Advances and Future Perspectives. *Proc. of KOTAC Int. Conf. 2003 'Operator Theory and Appl.'*, Chungnam National University, Daejeon City, Korea (June 19–21, 2003). *Trends in Contemporary Math.* **6** (2003), pp. 113–123
54. L.M. Schmitt: Coevolutionary Convergence to Global Optima. IN: [5], pp. 373–375
55. L.M. Schmitt: Optimization with Genetic Algorithms in Multi-Species Environments. IN: L. Jiao *et al.* (eds.), *Proc. Fifth Int. Conf. on Computation Intelligence and Multimedia Appl. (ICCIMA 2003)*, Xidian University, Xi'an, China (Sept. 27–30, 2003). IEEE Computer Society (publisher), Los Alamitos, CA, USA, ISBN 0-7695-1957-1, pp. 194–199
56. L.M. Schmitt: Theory of Coevolutionary Genetic Algorithms. IN: M. Guo, L.T. Yang (eds.), *Proc. 'The 2003 Int. Symp. on Parallel and Distributed Processing and Appl.' ISPA 2003*, The University of Aizu, Aizu-Wakamatsu City, Fukushima, Japan (July 2–4, 2003). LNCS 2745, Springer Verlag, Berlin, Germany (2003), pp. 285–293
57. L.M. Schmitt: Classification with Scaled Genetic Algorithms in a Coevolutionary Setting IN: [42]
58. H.-P. Schwefel: *Evolution and Optimum Seeking.* Wiley Interscience Publ, New York, NY, USA (1995)

59. R. Sedgewick: *Algorithms*. Addison-Wesley Publ., Boston, MA, USA (1988)
60. J. Suzuki: A Further Result on the Markov Chain Model of Genetic Algorithms and Its Application to a Simulated Annealing-like Strategy. In: R.K. Belew, M.D. Vose (eds.), *Proc. Foundations of Genetic Algorithms 4*. Morgan Kaufmann Publ., San Francisco, CA, USA (1997), pp. 53–72
61. J. Suzuki: A Further Result on the Markov Chain Model of Genetic Algorithms and Its Application to a Simulated Annealing-like Strategy. *IEEE Transactions on Systems, Man and Cybernetics — Part B* **28** (1998), pp. 95–102
62. S. Tong, D.J. Powell: Genetic Algorithms: A Fundamental Component of an Optimization Toolkit for Improved Engineering Designs. In: [5], pp. 2346–2359
63. M.D. Vose, G.E. Liepins. Punctuated Equilibria in Genetic Search. *Complex Systems* **5** (1991), pp. 31–44
64. M.D. Vose: *The Simple Genetic Algorithm: Foundations and Theory*. MIT Press, Cambridge, MA, USA (1999)
65. I. Wegener: Methods for the analysis of evolutionary algorithms on pseudo-boolean functions. In: R. Sarker *et al.* (eds.), *Evolutionary Optimization*. Kluwer, Dordrecht, The Netherlands (2002), pp. 349–369
66. I. Wegener, C. Witt: On the Optimization of Monotone Polynomials by Simple Randomized Search Heuristics. To appear in: *Combinatorics, Probability & Computing*
67. Downloadable collection of technical reports. Web-site of Collaborative Research Center "*Computational Intelligence*" (SFB 531), Universität Dortmund, Dortmund, Germany
`http://sfbci.informatik.uni-dortmund.de/home/English/_index.html`