

Publications of Timo Kötzing

This document lists all peer-reviewed publications of [Timo Kötzing](#), Chair for Algorithm Engineering, Hasso Plattner Institute, Potsdam, Germany. This listing was automatically generated on November 28, 2020. An up-to-date version is available online at hpi.de/friedrich/docs/publist/koetzing.pdf.

Journal articles

- [1] Khazraei, A., Kötzing, T., Seidel, K., [Learning Half-Spaces and other Concept Classes in the Limit with Iterative Learners](#). In: pp. 22, 2020.

In order to model an efficient learning paradigm, iterative learning algorithms access data one by one, updating the current hypothesis without regress to past data. Past research on iterative learning analyzed for example many important requirements and their impact on iterative learners. In this paper, our results are twofold. First, we analyze the relative learning power of various settings of iterative learning, including learning from text and from informant, as well as various further restrictions, for example we show that strongly non-U-shaped learning is restrictive for iterative learning from informant. Second, we investigate the learnability of the concept class of half-spaces and provide a constructive iterative algorithm to learn the set of half-spaces from informant.

- [2] Kötzing, T., Seidel, K., [Learning Languages in the Limit from Positive Information with Finitely Many Memory Changes](#). In: pp. 14, 2020.

We investigate learning collections of languages from texts by an inductive inference machine with access to the current datum and its memory in form of states. The bounded memory states (*BMS*) learner is considered successful in case it eventually settles on a correct hypothesis while exploiting only finitely many different states. We give the complete map of all pairwise relations for an established collection of learning success restrictions. Most prominently, we show that non-U-shapedness is not restrictive, while conservativeness and (strong) monotonicity are. Some results carry over from iterative learning by a general lemma showing that, for a wealth of restrictions (the semantic restrictions), iterative and bounded memory states learning are equivalent. We also give an example of a non-semantic restriction (strongly non-U-shapedness) where the two settings differ.

- [3] Doerr, B., Kötzing, T., Lagodzinski, J. A. G., Lengler, J., [The impact of lexicographic parsimony pressure for ORDER/MAJORITY on the run time](#). In: *Theoretical Computer Science* 816, pp. 144–168, 2020.

While many optimization problems work with a fixed number of decision variables and thus a fixed-length representation of possible solutions, genetic programming (GP) works on variable-length representations. A naturally occurring problem is that of bloat, that is, the unnecessary growth of solution lengths, which may slow down the optimization process. So far, the mathematical runtime analysis could not deal well with bloat and required explicit assumptions limiting bloat. In this paper, we provide the first mathematical runtime analysis of a GP algorithm that does not require any assumptions on the bloat. Previous performance guarantees were only proven conditionally for runs in which no strong bloat occurs. Together with improved analyses for the case with bloat restrictions our results show that such assumptions on the bloat are not necessary and that the algorithm is efficient without explicit bloat control mechanism. More specifically, we analyzed the performance of the $(1 + 1)$ GP on the two benchmark functions Order and Majority. When using lexicographic parsimony pressure as bloat control, we show a tight runtime estimate of $O(T_{init} + n \log n)$ iterations both for Order and Majority. For the case without bloat control, the bounds $O(T_{init} \log T_{init} + n(\log n)^3)$ and $\Omega(T_{init} + n \log n)$ (and $\Omega(T_{init} + \log T_{init})$ for $n = 1$) hold for Majority.

- [4] Kötzing, T., Lagodzinski, J. A. G., Lengler, J., Melnichenko, A., [Destructiveness of lexicographic parsimony pressure and alleviation by a concatenation crossover in genetic programming](#). In: *Theoretical Computer Science* 816, pp. 96–113, 2020.

For theoretical analyses there are two specifics distinguishing GP from many other areas of evolutionary computation: the variable size representations, in particular yielding a possible bloat (i.e. the growth of individuals with redundant parts); and also the role and the realization of crossover, which is particularly central in GP due to the tree-based representation. Whereas some theoretical work on GP has studied the effects of bloat, crossover had surprisingly little share in this work. We analyze a simple crossover operator in combination with randomized local search, where a preference for small solutions minimizes bloat (lexicographic parsimony pressure); we denote the resulting algorithm Concatenation Crossover GP. We consider three variants of the well-studied Majority test function, adding large plateaus in different ways to the fitness landscape and thus giving a test bed for analyzing the interplay of variation operators and bloat control mechanisms in a setting with local optima. We show that the Concatenation Crossover GP can efficiently optimize these test functions, while local search cannot be efficient for all three variants independent of employing bloat control.

- [5] Shi, F., Schirneck, M., Friedrich, T., Kötzing, T., Neumann, F., [Correction to: Reoptimization Time Analysis of Evolutionary Algorithms on Linear Functions Under Dynamic Uniform Constraints](#). In: *Algorithmica* 82, pp. 3117–3123, 2020.

In the article "Reoptimization Time Analysis of Evolutionary Algorithms on Linear Functions Under Dynamic Uniform Constraints", we claimed a worst-case runtime of $O(nD \log D)$ and $O(nD)$ for the Multi-Objective Evolutionary Algorithm and the Multi-Objective $(\mu + (\lambda, \lambda))$ Genetic Algorithm, respectively, on linear profit functions under dynamic uniform constraint, where $D = |B - B^*|$ denotes the difference between the original constraint bound B and the new one B^* . The technique used to prove these results contained an error. We correct this mistake and show a weaker bound of $O(nD^2)$ for both algorithms instead.

- [6] Friedrich, T., Kötzing, T., Lagodzinski, J. A. G., Neumann, F., Schirneck, M., [Analysis of the \(1+1\) EA on Subclasses of Linear Functions under Uniform and Linear Constraints](#). In: *Theoretical Computer Science* 832, pp. 3–19, 2020.

Linear functions have gained great attention in the run time analysis of evolutionary computation methods. The corresponding investigations have provided many effective tools for analyzing more complex problems. So far, the runtime analysis of evolutionary algorithms has mainly focused on unconstrained problems, but problems occurring in applications frequently involve constraints. Therefore, there is a strong need to extend the methods for analyzing unconstrained problems to a setting involving constraints. In this paper, we consider the behavior of the classical (1+1) evolutionary algorithm on linear functions under linear constraint. We show tight bounds in the case where the constraint is given by the OneMax function and the objective function is given by either the OneMax or the BinVal function. For the general case we present upper and lower bounds.

- [7] Kötzing, T., Krejca, M. S., [First-hitting times under drift](#). In: *Theoretical Computer Science* 796, pp. 51–69, 2019.

For the last ten years, almost every theoretical result concerning the expected run time of a randomized search heuristic used drift theory, making it the arguably most important tool in this domain. Its success is due to its ease of use and its powerful result: drift theory allows the user to derive bounds on the expected first-hitting time of a random process by bounding expected local changes of the process – the drift. This is usually far easier than bounding the expected first-hitting time directly. Due to the widespread use of drift theory, it is of utmost importance to have the best drift theorems possible. We improve the fundamental additive, multiplicative, and variable drift theorems by stating them in a form as general as possible and providing examples of why the restrictions we keep are still necessary. Our additive drift theorem for upper bounds only requires the process to be lower-bounded, that is, we remove unnecessary restrictions like a finite, discrete, or bounded state space. As corollaries, the same is true for our upper bounds in the case of variable and multiplicative drift. By bounding the step size of the process, we derive new lower-bounding multiplicative and variable drift theorems. Last, we also state theorems that are applicable when the process has a drift of 0, by using a drift on the variance of the process.

- [8] Doerr, B., Doerr, C., Kötzing, T., [Solving Problems with Unknown Solution Length at Almost No Extra Cost](#). In: *Algorithmica* 81, pp. 703–748, 2019.

Following up on previous work of Cathabard et al. (in: Proceedings of foundations of genetic algorithms (FOGA'11), ACM, 2011) we analyze variants of the (1 + 1) evolutionary algorithm (EA) for problems with unknown solution length. For their setting, in which the solution length is sampled from a geometric distribution, we provide mutation rates that yield for both benchmark functions ONEMAX and LEADINGONES an expected optimization time that is of the same order as that of the (1 + 1) EA knowing the solution length. More than this, we show that almost the same run times can be achieved even if no a priori information on the solution length is available. We also regard the situation in which neither the number nor the positions of the bits with an influence on the fitness function are known. Solving an open problem from Cathabard et al. we show that, for arbitrary natural numbers s , such ONEMAX and LEADINGONES instances can be solved, simultaneously for all natural numbers n , in expected time $O(n(\log(n))^2 \log \log(n) \dots \log^{(s-1)}(n)(\log^{(s)}(n))^{(1+\varepsilon)})$ and $O(n^2 \log(n) \log \log(n) \dots \log^{(s-1)}(n)(\log^{(s)}(n))^{(1+\varepsilon)})$, respectively; that is, in almost the same time as if n and the relevant bit positions were known. For the LEADINGONES case, we prove lower bounds of same asymptotic order of magnitude apart from the $(\log^{(s)}(n))^\varepsilon$ factor. Aiming at closing this arbitrarily small remaining gap, we realize that there is no asymptotically best performance for this problem. For any algorithm solving, for all n , all instances of size n in expected time at most $T(n)$, there is an algorithm doing the same in time $T'(n)$ with $T' = o(T)$. For ONEMAX we show results of similar flavor.

- [9] Trubenova, B., Kötzing, T., Krejca, M. S., Lehre, P. K., [Surfing on the seascape: Adaptation in a changing environment](#). In: *Evolution: International Journal of Organic Evolution* 73, pp. 1356–1374, 2019.

The environment changes constantly at various time scales and, in order to survive, species need to keep adapting. Whether these species succeed in avoiding extinction is a major evolutionary question. Using a multilocus evolutionary model of a mutation-limited population adapting under strong selection, we investigate the effects of the frequency of environmental fluctuations on adaptation. Our results rely on an "adaptive-walk" approximation and use mathematical methods from evolutionary computation theory to investigate the interplay between fluctuation frequency, the similarity of environments, and the number of loci contributing to adaptation. First, we assume a linear additive fitness function, but later generalize our results to include several types of epistasis. We show that frequent environmental changes prevent populations from reaching a fitness peak, but they may also prevent the large fitness loss that occurs after a single environmental change. Thus, the population can survive, although not thrive, in a wide range of conditions. Furthermore, we show that in a frequently changing environment, the similarity of threats that a population faces affects the level of adaptation that it is able to achieve. We check and supplement our analytical results with simulations.

- [10] Friedrich, T., Kötzing, T., Krejca, M. S., [Unbiasedness of Estimation-of-Distribution Algorithms](#). In: *Theoretical Computer Science* 785, pp. 46–59, 2019.

In the context of black-box optimization, black-box complexity is used for understanding the inherent difficulty of a given optimization problem. Central to our understanding of nature-inspired search heuristics in this context is the notion of unbiasedness. Specialized black-box complexities have been developed in order to better understand the limitations of these heuristics – especially of (population-based) evolutionary algorithms (EAs). In contrast to this, we focus on a model for algorithms explicitly maintaining a probability distribution over the search space: so-called estimation-of-distribution algorithms (EDAs). We consider the recently introduced n -Bernoulli- λ -EDA framework, which subsumes, for example, the commonly known EDAs PBIL, UMDA, λ -MMAS_{IB}, and cGA. We show that an n -Bernoulli- λ -EDA is unbiased if and only if its probability distribution satisfies a certain invariance property under isometric automorphisms of $[0, 1]^n$. By restricting how an n -Bernoulli- λ -EDA can perform an update, in a way common to many examples, we derive conciser characterizations, which are easy to verify. We demonstrate this by showing that our examples above are all unbiased.

- [11] Shi, F., Schirneck, M., Friedrich, T., Kötzing, T., Neumann, F., [Reoptimization Time Analysis of Evolutionary Algorithms on Linear Functions Under Dynamic Uniform Constraints](#). In: *Algorithmica* 81, pp. 828–857, 2019.

Rigorous runtime analysis is a major approach towards understanding evolutionary computing techniques, and in this area linear pseudo-Boolean objective functions play a central role. Having an additional linear constraint is then equivalent to the NP-hard Knapsack problem, certain classes thereof have been studied in recent works. In this article, we present a dynamic model of optimizing linear functions under uniform constraints. Starting from an optimal solution with respect to a given constraint bound, we investigate

the runtimes that different evolutionary algorithms need to recompute an optimal solution when the constraint bound changes by a certain amount. The classical $(1 + 1)$ EA and several population-based algorithms are designed for that purpose, and are shown to recompute efficiently. Furthermore, a variant of the $(1 + (\lambda, \lambda))$ GA for the dynamic optimization problem is studied, whose performance is better when the change of the constraint bound is small.

- [12] Doerr, B., Fischbeck, P., Frahnw, C., Friedrich, T., Kötzing, T., Schirneck, M., [Island Models Meet Rumor Spreading](#). In: *Algorithmica* 81, pp. 886–915, 2019.

Island models in evolutionary computation solve problems by a careful interplay of independently running evolutionary algorithms on the island and an exchange of good solutions between the islands. In this work, we conduct rigorous run time analyses for such island models trying to simultaneously obtain good run times and low communication effort. We improve the existing upper bounds for both measures (i) by improving the run time bounds via a careful analysis, (ii) by balancing individual computation and communication in a more appropriate manner, and (iii) by replacing the usual communicate-with-all approach with randomized rumor spreading. In the latter, each island contacts a randomly chosen neighbor. This epidemic communication paradigm is known to lead to very fast and robust information dissemination in many applications. Our results concern island models running simple $(1+1)$ evolutionary algorithms to optimize the classic test functions OneMax and LeadingOnes. We investigate binary trees, d -dimensional tori, and complete graphs as communication topologies.

- [13] Aschenbach, M., Kötzing, T., Seidel, K., [Learning from Informants: Relations between Learning Success Criteria](#). In: *ArXiv*, pp. 37, 2018, [ArXiv preprint](#).

Learning from positive and negative information, so-called informants, being one of the models for human and machine learning introduced by Gold [1967], is investigated. Particularly, naturally arising questions about this learning setting, originating in results on learning from solely positive information, are answered. By a carefully arranged argument learners can be assumed to only change their hypothesis in case it is inconsistent with the data (such a learning behavior is called conservative). The deduced main theorem states the relations between the most important delayable learning success criteria, being the ones not ruined by a delayed in time hypothesis output. Additionally, our investigations concerning the non-delayable requirement of consistent learning underpin the claim for delayability being the right structural property to gain a deeper understanding concerning the nature of learning success criteria. Moreover, we obtain an anomalous hierarchy when allowing for an increasing finite number of anomalies of the hypothesized language by the learner compared with the language to be learned. In contrast to the vacillatory hierarchy for learning from solely positive information, we observe a duality depending on whether infinitely many vacillations between different (almost) correct hypotheses are still considered a successful learning behavior.

- [14] Kötzing, T., Sudholt, D., [Preface to the Special Issue on Theory of Genetic and Evolutionary Computation](#). In: *Algorithmica* 80, pp. 1575–1578, 2018.

Evolutionary algorithms (EAs) are randomized search heuristics that can be employed to solve complex optimization problems, including multimodal or highly constrained problems. EAs work by mimicking principles from natural evolution: maintaining a collection of possible solutions (a population) and iteratively creating variants of the individuals (the offspring) and then choosing a new set of individuals for the next iteration (selection). EAs are popular because they represent general-purpose optimizers that can be easily applied to various problems, even in cases where little or no in-depth knowledge about the problem is available. In order to guide practitioners devising new and effective algorithms, theoretical computer scientists employ methods from the field of randomized algorithms to analyze the working principles of EAs with mathematical rigor. Key questions concern the impact of parameter choices (such as, for example, the offspring size or the choice of variation operators) as well as foundational work on developing powerful analysis methods. The theory track of the annual ACM Genetic and Evolutionary Computation Conference (GECCO) is the first tier event for advances in this direction. In this special issue six selected papers from the 2016 edition of the GECCO theory track are collected, each one of them carefully revised and extended to meet the high quality standards of *Algorithmica*.

- [15] Doerr, B., Doerr, C., Kötzing, T., [Static and Self-Adjusting Mutation Strengths for Multi-valued Decision Variables](#). In: *Algorithmica* 80, pp. 1732–1768, 2018.

The most common representation in evolutionary computation are bit strings. With very little theoretical work existing on how to use evolutionary algorithms for decision variables taking more than two values, we study the run time of simple evolutionary algorithms on some OneMax-like functions defined over $\Omega = 0, 1, \dots, r - 1^n$. We observe a crucial difference in how we extend the one-bit-flip and standard-bit mutation operators to the multi-valued domain. While it is natural to modify a random position of the string or select each position of the solution vector for modification independently with probability $1/n$, there are various ways to then change such a position. If we change each selected position to a random value different from the original one, we obtain an expected run time of $\Theta(nr \log n)$. If we change each selected position by $+1$ or -1 (random choice), the optimization time reduces to $\Theta(nr + n \log n)$. If we use a random mutation strength $i \in 0, 1, \dots, r - 1$ with probability inversely proportional to i and change the selected position by $+i$ or $-i$ (random choice), then the optimization time becomes $\Theta(n \log(r)(\log n + \log r))$, which is asymptotically faster than the previous if $r = \omega(\log(n) \log \log(n))$. Interestingly, a better expected performance can be achieved with a self-adjusting mutation strength that is based on the success of previous iterations. For the mutation operator that modifies a randomly chosen position, we show that the self-adjusting mutation strength yields an expected optimization time of $\Theta(n(\log n + \log r))$, which is best possible among all dynamic mutation strengths. In our proofs, we use a new multiplicative drift theorem for computing lower bounds, which is not restricted to processes that move only towards the target.

- [16] Dang, D.-C., Friedrich, T., Kötzing, T., Krejca, M. S., Lehre, P. K., Oliveto, P. S., Sudholt, D., Sutton, A. M., [Escaping Local Optima Using Crossover with Emergent Diversity](#). In: *Transactions on Evolutionary Computation* 22, pp. 484–497, 2018.

Population diversity is essential for the effective use of any crossover operator. We compare seven commonly used diversity mechanisms and prove rigorous run time bounds for the $(\mu + 1)$ GA using uniform crossover on the fitness function *Jump_k*. All previous results in this context only hold for unrealistically low crossover probability $p_c = O(k/n)$, while we give analyses for the setting of constant $p_c < 1$ in all but one case. Our bounds show a dependence on the problem size n , the jump length k , the population size μ , and the crossover probability p_c . For the typical case of constant $k > 2$ and constant p_c , we can compare the resulting expected optimisation times for different diversity mechanisms assuming an optimal choice of μ : $O(n^{k-1})$ for duplicate elimination/minimisation, $O(n^2 \log n)$ for maximising the convex hull, $O(n \log n)$ for det. crowding (assuming $p_c = k/n$), $O(n \log n)$ for maximising the Hamming distance, $O(n \log n)$ for fitness sharing, $O(n \log n)$ for the single-receiver island model. This proves a sizeable advantage of all variants of the $(\mu + 1)$ GA compared to the $(1+1)$ EA, which requires $\Theta(n^k)$. In a short empirical study we confirm that the asymptotic differences can also be observed experimentally.

- [17] Friedrich, T., Kötzing, T., Krejca, M. S., Sutton, A. M., [The Compact Genetic Algorithm is Efficient under Extreme Gaussian Noise](#). In: *Transactions on Evolutionary Computation* 21, pp. 477–490, 2017.

Practical optimization problems frequently include uncertainty about the quality measure, for example due to noisy evaluations. Thus, they do not allow for a straightforward application of traditional optimization techniques. In these settings, randomized search heuristics such as evolutionary algorithms are a popular choice because they are often assumed to exhibit some kind of resistance to noise. Empirical evidence suggests that some algorithms, such as estimation of distribution algorithms (EDAs) are robust against a scaling of the noise intensity, even without resorting to explicit noise-handling techniques such as resampling. In this paper, we want to support such claims with mathematical rigor. We introduce the concept of graceful scaling in which the run time of an algorithm scales polynomially with noise intensity. We study a monotone fitness function over binary strings with additive noise taken from a Gaussian distribution. We show that myopic heuristics cannot efficiently optimize the function under arbitrarily intense noise without any explicit noise-handling. Furthermore, we prove that using a population does not help. Finally we show that a simple EDA called the compact Genetic Algorithm can overcome the shortsightedness of mutation-only heuristics to scale gracefully with noise. We conjecture that recombinative genetic algorithms also have this property.

- [18] Jain, S., Kötzing, T., Stephan, F., [Enlarging learnable classes](#). In: *Information and Computation* 251, pp. 194–207, 2016.

We study which classes of recursive functions satisfy that their union with any other explanatorily learnable class of recursive functions is again explanatorily learnable. We provide sufficient criteria for classes of recursive functions to satisfy this property and also investigate its effective variants. Furthermore, we study the question which learners can be effectively extended to learn a larger class of functions. We solve an open problem by showing that there is no effective procedure which does this task on all learners which do not learn a dense class of recursive functions. However, we show that there are two effective extension procedures such that each learner is extended by one of them.

- [19] Case, J., Kötzing, T., [Strongly non-U-shaped language learning results by general techniques](#). In: *Information and Computation* 251, pp. 1–15, 2016.

In learning, a semantic or behavioral U-shape occurs when a learner first learns, then unlearns, and, finally, relearns, some target concept. This paper introduces two general techniques and applies them especially to syntactic U-shapes in learning: one technique to show when they are necessary and one to show when they are unnecessary. The technique for the former is very general and applicable to a much wider range of learning criteria. It employs so-called self-learning classes of languages which are shown to characterize completely one criterion learning more than another. We apply these techniques to show that, for set-driven and rearrangement-independent learning, any kind of U-shapes is unnecessary. Furthermore, we show that U-shapes are necessary in a strong way for iterative learning, contrasting with an earlier result by Case and Moelius that semantic U-shapes are unnecessary for iterative learning.

- [20] Gießen, C., Kötzing, T., [Robustness of Populations in Stochastic Environments](#). In: *Algorithmica* 75, pp. 462–489, 2016.

We consider stochastic versions of OneMax and LeadingOnes and analyze the performance of evolutionary algorithms with and without populations on these problems. It is known that the (1+1) EA on OneMax performs well in the presence of very small noise, but poorly for higher noise levels. We extend these results to LeadingOnes and to many different noise models, showing how the application of drift theory can significantly simplify and generalize previous analyses. Most surprisingly, even small populations (of size $\Theta(\log n)$) can make evolutionary algorithms perform well for high noise levels, well outside the abilities of the (1+1) EA. Larger population sizes are even more beneficial; we consider both parent and offspring populations. In this sense, populations are robust in these stochastic settings.

- [21] Kötzing, T. [Concentration of First Hitting Times Under Additive Drift](#). In: *Algorithmica* 75, pp. 490–506, 2016.

Recent advances in drift analysis have given us better and better tools for understanding random processes, including the run time of randomized search heuristics. In the setting of multiplicative drift we do not only have excellent bounds on the expected run time, but also more general results showing the concentration of the run time. In this paper we investigate the setting of additive drift under the assumption of strong concentration of the "step size" of the process. Under sufficiently strong drift towards the goal we show a strong concentration of the hitting time. In contrast to this, we show that in the presence of small drift a Gambler's-Ruin-like behavior of the process overrides the influence of the drift. Finally, in the presence of sufficiently strong negative drift the hitting time is superpolynomial with high probability; this corresponds to the so-called negative drift theorem, for which we give new variants.

- [22] Case, J., Kötzing, T., [Topological Separations in Inductive Inference](#). In: *Theoretical Computer Science* 620, pp. 33–45, 2016.

Re learning in the limit from positive data, a major concern is which classes of languages are learnable with respect to a given learning criterion. We are particularly interested herein in the reasons for a class of languages to be unlearnable. We consider two types of reasons. One type is called topological where it does not help if the learners are allowed to be uncomputable (an example of Gold's is that no class containing an infinite language and all its finite sub-languages is learnable - even by an uncomputable learner). Another reason is called computational (where the learners are required to be algorithmic). In particular, two learning criteria might allow for learning different classes of languages from one another - but with dependence on whether the unlearnability is of type topological or computational. In this paper we formalize the idea of two learning criteria separating topologically in learning power. This allows us to study more closely why two learning criteria separate in learning power. For a variety of learning criteria, concerning vacillatory, monotone, (several kinds of) iterative and feedback learning, we show that certain learning criteria separate topologically, and certain others, which are known to separate, are shown not to separate topologically. Showing that learning criteria do not separate topologically implies that any known separation must necessarily exploit algorithmicity of the learner.

- [23] Kötzing, T., Palenta, R., [A map of update constraints in inductive inference](#). In: *Theoretical Computer Science* 650, pp. 4–24, 2016.

We investigate how different learning restrictions reduce learning power and how the different restrictions relate to one another. We give a complete map for nine different restrictions both for the cases of complete information learning and set-driven learning. This completes the picture for these well-studied delayable learning restrictions. A further insight is gained by different characterizations of conservative learning in terms of variants of cautious learning. Our analyses greatly benefit from general theorems we give, for example showing that learners with exclusively delayable restrictions can always be assumed total.

- [24] Friedrich, T., Kötzing, T., Krejca, M. S., Sutton, A. M., [Robustness of Ant Colony Optimization to Noise](#). In: *Evolutionary Computation* 24, pp. 237–254, 2016.
- Recently Ant Colony Optimization (ACO) algorithms have been proven to be efficient in uncertain environments, such as noisy or dynamically changing fitness functions. Most of these analyses focus on combinatorial problems, such as path finding. We analyze an ACO algorithm in a setting where we try to optimize the simple OneMax test function, but with additive posterior noise sampled from a Gaussian distribution. Without noise the classical $(\mu + 1)$ -EA outperforms any ACO algorithm, with smaller μ being better; however, with large noise, the $(\mu + 1)$ -EA fails, even for high values of μ (which are known to help against small noise). In this paper we show that ACO is able to deal with arbitrarily large noise in a graceful manner, that is, as long as the evaporation factor p is small enough dependent on the parameter δ^2 of the noise and the dimension n of the search space ($p = o(1/(n + \delta \log n)^2 \log n)$), optimization will be successful.
- [25] Jain, S., Kötzing, T., Ma, J., Stephan, F., [On the Role of Update Constraints and Text-Types in Iterative Learning](#). In: *Information and Computation* 247, pp. 152–168, 2016.
- The present work investigates the relationship of iterative learning with other learning criteria such as decisiveness, caution, reliability, non-U-shapedness, monotonicity, strong monotonicity and conservativeness. Building on the result of Case and Moelius that iterative learners can be made non-U-shaped, we show that they also can be made cautious and decisive. Furthermore, we obtain various special results with respect to one-one texts, fat texts and one-one hypothesis spaces.
- [26] Doerr, B., Doerr, C., Kötzing, T., [Unbiased Black-Box Complexities of Jump Functions](#). In: *Evolutionary Computation* 23, pp. 641–670, 2015.
- We analyze the unbiased black-box complexity of jump functions with small, medium, and large sizes of the fitness plateau surrounding the optimal solution. Among other results, we show that when the jump size is $(1/2 - \epsilon)n$, that is, only a small constant fraction of the fitness values is visible, then the unbiased black-box complexities for arities 3 and higher are of the same order as those for the simple OneMax function. Even for the extreme jump function, in which all but the two fitness values $n/2$ and n are blanked out, polynomial-time mutation-based (i.e., unary unbiased) black-box optimization algorithms exist. This is quite surprising given that for the extreme jump function almost the whole search space (all but a $\Theta(n^{-1/2})$ fraction) is a plateau of constant fitness. To prove these results, we introduce new tools for the analysis of unbiased black-box complexities, for example, selecting the new parent individual not by comparing the fitnesses of the competing search points, but also by taking into account the (empirical) expected fitnesses of their offspring.
- [27] Freydenberger, D. D., Kötzing, T., [Fast Learning of Restricted Regular Expressions and DTDs](#). In: *Theory of Computing Systems* 57, pp. 1114–1158, 2015.
- We study the problem of generalizing from a finite sample to a language taken from a predefined language class. The two language classes we consider are subsets of the regular languages and have significance in the specification of XML documents (the classes corresponding to so called chain regular expressions, Chares, and to single occurrence regular expressions, Sores). The previous literature gave a number of algorithms for generalizing to Sores providing a trade off between quality of the solution and speed. Furthermore, a fast but non-optimal algorithm for generalizing to Chares is known. For each of the two language classes we give an efficient algorithm returning a minimal generalization from the given finite sample to an element of the fixed language class; such generalizations are called descriptive. In this sense, both our algorithms are optimal.
- [28] Kötzing, T., Sutton, A. M., Neumann, F., O’Reilly, U.-M., [The Max problem revisited: The importance of mutation in genetic programming](#). In: *Theoretical Computer Science* 545, pp. 94–107, 2014.
- This paper contributes to the rigorous understanding of genetic programming algorithms by providing runtime complexity analyses of the well-studied Max problem. Several experimental studies have indicated that it is hard to solve the Max problem with crossover-based algorithms. Our analyses show that different variants of the Max problem can probably be solved using simple mutation-based genetic programming algorithms. Our results advance the body of computational complexity analyses of genetic programming, indicate the importance of mutation in genetic programming, and reveal new insights into the behavior of mutation-based genetic programming algorithms.
- [29] Kötzing, T. [Iterative learning from positive data and counters](#). In: *Theoretical Computer Science* 519, pp. 155–169, 2014.
- We analyze iterative learning in the limit from positive data with the additional information provided by a counter. The simplest type of counter provides the current iteration number (counting up from 0 to infinity), which is known to improve learning power over plain iterative learning. We introduce five other (weaker) counter types, for example only providing some unbounded and non-decreasing sequence of numbers. Analyzing these types allows for understanding what properties of a counter can benefit learning. For the iterative setting, we completely characterize the relative power of the learning criteria corresponding to the counter types. In particular, for our types, the only properties improving learning power are unboundedness and strict monotonicity. Furthermore, we show that each of our types of counter improves learning power over weaker ones in some settings, and that, for iterative learning criteria with one of these types of counter, separations of learning criteria are necessarily witnessed by classes containing only infinite languages.
- [30] Doerr, B., Doerr, C., Kötzing, T., [The unbiased black-box complexity of partition is polynomial](#). In: *Artificial Intelligence* 216, pp. 275–286, 2014.
- Unbiased black-box complexity was introduced as a refined complexity model for randomized search heuristics (Lehre and Witt (2012) [24]). For several problems, this notion avoids the unrealistically low complexity results given by the classical model of Droste et al. (2006) [10]. We show that for some problems the unbiased black-box complexity remains artificially small. More precisely, for two different formulations of an NP-hard subclass of the well-known Partition problem, we give mutation-only unbiased black-box algorithms having complexity $O(n \log n)$. This indicates that also the unary unbiased black-box complexity does not give a complete picture of the true difficulty of this problem for randomized search heuristics.
- [31] Case, J., Kötzing, T., [Memory-limited non-U-shaped learning with solved open problems](#). In: *Theoretical Computer Science* 473, pp. 100–123, 2013.

In empirical cognitive science, for human learning, a semantic or behavioral U-shape occurs when a learner first learns, then unlearns, and, finally, relearns, some target concept. Within the formal framework of Inductive Inference, for learning from positive data, previous results have shown, for example, that such U-shapes are unnecessary for explanatory learning, but are necessary for behaviorally correct and non-trivial vacillatory learning. Herein we also distinguish between semantic and syntactic U-shapes. We answer a number of open questions in the prior literature as well as provide new results regarding syntactic U-shapes. Importantly for cognitive science, we see more of a previously noticed pattern that, for parameterized learning criteria, beyond very few initial parameter values, U-shapes are necessary for full learning power. We analyze the necessity of U-shapes in two memory-limited settings. The first setting is Bounded Memory State (BMS) learning, where a learner has an explicitly-bounded state memory, and otherwise only knows its current datum. We show that there are classes learnable with three (or more) memory states that are not learnable non-U-shapedly with any finite number of memory states. This result is surprising, since, for learning with one or two memory states, U-shapes are known to be unnecessary. This solves an open question from the literature. The second setting is that of Memoryless Feedback (MLF) learning, where a learner may ask a bounded number of questions about what data has been seen so far, and otherwise only knows its current datum. We show that there is a class learnable memorylessly with a single feedback query such that this class is not learnable non-U-shapedly memorylessly with any finite number of feedback queries. We employ self-learning classes together with the Operator Recursion Theorem for many of our results, but we also introduce two new techniques for obtaining results. The first is for transferring inclusion results from one setting to another. The main part of the second is the Hybrid Operator Recursion Theorem, which enables us to separate some learning criteria featuring complexity-bounded learners, employing self-learning classes. Both techniques are not specific to U-shaped learning, but applicable for a wide range of settings.

- [32] Doerr, B., Johannsen, D., Kötzing, T., Neumann, F., Theile, M., [More effective crossover operators for the all-pairs shortest path problem](#). In: *Theoretical Computer Science* 471, pp. 12–26, 2013.

The all-pairs shortest path problem is the first non-artificial problem for which it was shown that adding crossover can significantly speed up a mutation-only evolutionary algorithm. Recently, the analysis of this algorithm was refined and it was shown to have an expected optimization time (w.r.t. the number of fitness evaluations) of $\Theta(n^{3.25}(\log n)^{0.25})$. In contrast to this simple algorithm, evolutionary algorithms used in practice usually employ refined recombination strategies in order to avoid the creation of infeasible offspring. We study extensions of the basic algorithm by two such concepts which are central in recombination, namely repair mechanisms and parent selection. We show that repairing infeasible offspring leads to an improved expected optimization time of $O(n^{3.2}(\log n)^{0.2})$. As a second part of our study we prove that choosing parents that guarantee feasible offspring results in an even better optimization time of $O(n^3 \log n)$. Both results show that already simple adjustments of the recombination operator can asymptotically improve the runtime of evolutionary algorithms.

- [33] Doerr, B., Kötzing, T., Lengler, J., Winzen, C., [Black-box complexities of combinatorial problems](#). In: *Theoretical Computer Science* 471, pp. 84–106, 2013.

Black-box complexity is a complexity theoretic measure for how difficult a problem is to be optimized by a general purpose optimization algorithm. It is thus one of the few means trying to understand which problems are tractable for genetic algorithms and other randomized search heuristics. Most previous work on black-box complexity is on artificial test functions. In this paper, we move a step forward and give a detailed analysis for the two combinatorial problems minimum spanning tree and single-source shortest paths. Besides giving interesting bounds for their black-box complexities, our work reveals that the choice of how to model the optimization problem is non-trivial here. This in particular comes true where the search space does not consist of bit strings and where a reasonable definition of unbiasedness has to be agreed on.

- [34] Case, J., Kötzing, T., [Computability-Theoretic Learning Complexity](#). In: *Philosophical Transactions of the Royal Society A* 370, pp. 3570–3596, 2012.

Initially discussed are some of Alan Turing’s wonderfully profound and influential ideas about mind and mechanism-including regarding their connection to the main topic of the present study, which is within the field of computability-theoretic learning theory. Herein is investigated the part of this field concerned with the algorithmic, trial-and-error inference of eventually correct programs for functions from their data points. As to the main content of this study: in prior papers, beginning with the seminal work by Freivalds et al. in 1995, the notion of intrinsic complexity is used to analyse the learning complexity of sets of functions in a Gold-style learning setting. Herein are pointed out some weaknesses of this notion. Offered is an alternative based on epitomizing sets of functions-sets that are learnable under a given learning criterion, but not under other criteria that are not at least as powerful. To capture the idea of epitomizing sets, new reducibility notions are given based on robust learning (closure of learning under certain sets of computable operators). Various degrees of epitomizing sets are characterized as the sets complete with respect to corresponding reducibility notions! These characterizations also provide an easy method for showing sets to be epitomizers, and they are then employed to prove several sets to be epitomizing. Furthermore, a scheme is provided to generate easily very strong epitomizers for a multitude of learning criteria. These strong epitomizers are the so-called self-learning sets, previously applied by Case & Kötzing in 2010. These strong epitomizers can be easily generated and employed in a myriad of settings to witness with certainty the strict separation in learning power between the criteria so epitomized and other not as powerful criteria!

- [35] Alcaraz, N., Friedrich, T., Kötzing, T., Krohmer, A., Müller, J., Pauling, J., Baumbach, J., [Efficient Key Pathway Mining: Combining Networks and OMICS Data](#). In: *Integrative Biology* 4, pp. 756–764, 2012.

Systems biology has emerged over the last decade. Driven by the advances in sophisticated measurement technology the research community generated huge molecular biology data sets. This comprises rather static data on the interplay of biological entities, for instance protein-protein interaction network data, as well as quite dynamic data collected for studying the behavior of individual cells or tissues in accordance to changing environmental conditions, such as DNA microarrays or RNA sequencing. Here we bring the two different data types together in order to gain higher level knowledge. We introduce a significantly improved version of the KeyPathwayMiner software framework. Given a biological network modelled as graph and a set of expression studies, KeyPathwayMiner efficiently finds and visualizes connected sub-networks where most components are expressed in most cases. It finds all maximal connected sub-networks where all nodes but k exceptions are expressed in all experimental studies but at least l exceptions. We demonstrate the power of the new approach by comparing it to similar approaches with gene expression data previously used to study Huntington’s disease. In addition, we demonstrate KeyPathwayMiner’s flexibility and applicability to non-array data by analyzing genome-scale DNA methylation profiles from colorectal tumor cancer patients. KeyPathwayMiner release 2 is available as a Cytoscape plugin and online at <http://keypathwayminer.mpi-inf.mpg.de>.

- [36] Kötzing, T., Neumann, F., Röglin, H., Witt, C., [Theoretical analysis of two ACO approaches for the traveling salesman problem](#). In: *Swarm Intelligence* 6, pp. 1–21, 2012.

Bioinspired algorithms, such as evolutionary algorithms and ant colony optimization, are widely used for different combinatorial optimization problems. These algorithms rely heavily on the use of randomness and are hard to understand from a theoretical point of view. This paper contributes to the theoretical analysis of ant colony optimization and studies this type of algorithm on one of the most prominent combinatorial optimization problems, namely the traveling salesperson problem (TSP). We present a new construction graph and show that it has a stronger local property than one commonly used for constructing solutions of the TSP. The rigorous runtime analysis for two ant colony optimization algorithms, based on these two construction procedures, shows that they lead to good approximation in expected polynomial time on random instances. Furthermore, we point out in which situations our algorithms get trapped in local optima and show where the use of the right amount of heuristic information is provably beneficial.

- [37] Case, J., Kötzing, T., [Learning secrets interactively. Dynamic modeling in inductive inference](#). In: *Information and Computation* 220, pp. 60–73, 2012.

Introduced is a new inductive inference paradigm, dynamic modeling. Within this learning paradigm, for example, function h learns function g iff, in the i -th iteration, h and g both produce output, h gets the sequence of all outputs from g in prior iterations as input, g gets all the outputs from h in prior iterations as input, and, from some iteration on, the sequence of h 's outputs will be programs for the output sequence of g . Dynamic modeling provides an idealization of, for example, a social interaction in which h seeks to discover program models of g 's behavior it sees in interacting with g , and h openly discloses to g its sequence of candidate program models to see what g says back. Sample results: every g can be so learned by some h ; there are g that can only be learned by an h if g can also learn that h back; there are extremely secretive h which cannot be learned back by any g they learn, but which, nonetheless, succeed in learning infinitely many g ; quadratic time learnability is strictly more powerful than linear time learnability. This latter result, as well as others, follows immediately from general correspondence theorems obtained from a unified approach to the paradigms within inductive inference. Many proofs, some sophisticated, employ machine self-reference, a.k.a., recursion theorems.

- [38] Heinz, J., Kasprzik, A., Kötzing, T., [Learning in the limit with lattice-structured hypothesis spaces](#). In: *Theoretical Computer Science* 457, pp. 111–127, 2012.

We define a collection of language classes which are TxtEx-learnable (learnable in the limit from positive data). The learners map any data input to an element of a fixed lattice, and keep the least upper bound of all lattice elements thus obtained as the current hypothesis. Each element of the lattice is a grammar for a language, and the learner climbs the lattice searching for the right element. We call these classes in our collection lattice classes. We provide several characterizations of lattice classes and their learners, which suggests they are very natural. In particular, we show that any class of languages is a lattice class iff it is TxtEx-learnable consistently, conservatively, set-drivenly, and strongly monotonically. We show several language classes previously discussed in the literature to be lattice classes, including the locally k -testable classes, the piecewise k -testable classes, the k -reversible languages and the pattern languages. We also show that lattice classes contain three previously known collections of language classes: string extension language classes, function-distinguishable language classes, and closed-set systems. Finally, the lattice perspective helps analyze the learning of these classes. Illustrations include query-learning results in dependence on the lattice structure, characterizations of closure properties and the VC-dimension of lattice classes in terms of lattice properties.

Conference papers

- [39] Doskoč, V., Kötzing, T., [Cautious Limit Learning](#). In: *Algorithmic Learning Theory (ALT)*, pp. 251–276, 2020.

We investigate language learning in the limit from text with various *cautious* learning restrictions. Learning is *cautious* if no hypothesis is a proper subset of a previous guess. While dealing with a seemingly natural learning behaviour, cautious learning does severely restrict explanatory (syntactic) learning power. To further understand why exactly this loss of learning power arises, Kötzing and Palenta (2016) introduced weakened versions of cautious learning and gave first partial results on their relation. In this paper, we aim to understand the restriction of cautious learning more fully. To this end we compare the known variants in a number of different settings, namely full-information and (partially) set-driven learning, paired either with the syntactic convergence restriction (explanatory learning) or the semantic convergence restriction (behaviourally correct learning). To do so, we make use of normal forms presented in Kötzing et al. (2017), most notably strongly locking and consistent learning. While strongly locking learners have been exploited when dealing with a variety of syntactic learning restrictions, we show how they can be beneficial in the semantic case as well. Furthermore, we expand the normal forms to a broader range of learning restrictions, including an answer to the open question of whether cautious learners can be assumed to be consistent, as stated in Kötzing et al. (2017).

- [40] Kötzing, T., Witt, C., [Improved Fixed-Budget Results via Drift Analysis](#). In: *Parallel Problem Solving From Nature (PPSN)*, pp. 648–660, 2020.

Fixed-budget theory is concerned with computing or bounding the fitness value achievable by randomized search heuristics within a given budget of fitness function evaluations. Despite recent progress in fixed-budget theory, there is a lack of general tools to derive such results. We transfer drift theory, the key tool to derive expected optimization times, to the fixed-budget perspective. A first and easy-to-use statement concerned with iterating drift in so-called greed-admitting scenarios immediately translates into bounds on the expected function value. Afterwards, we consider a more general tool based on the well-known variable drift theorem. Applications of this technique to the LeadingOnes benchmark function yield statements that are more precise than the previous state of the art.

- [41] Fokina, E., Kötzing, T., San Mauro, L., [Limit Learning Equivalence Structures](#). In: *Algorithmic Learning Theory (ALT)*, pp. 383–403, 2019.

While most research in Gold-style learning focuses on learning formal languages, we consider the identification of computable structures, specifically equivalence structures. In our core model the learner gets more and more information about which pairs of elements of a structure are related and which are not. The aim of the learner is to find (an effective description of) the isomorphism type of the structure presented in the limit. In accordance with language learning we call this learning criterion InfEx-learning (explanatory learning from informant). We start with a discussion and separations of different variants of this learning criterion, including learning from text (where the only information provided is which elements are related, and not which elements are not related) and finite learning (where the first actual conjecture of the learner has to be correct). This gives first intuitions and examples for what (classes of) structures are learnable and which are not. Our main contribution is a complete characterization of the learnable classes of structures in terms of a combinatorial property of the classes. This property allows us to derive a bound of \mathcal{O}'' on the computational complexity required to learn uniformly enumerable classes of structures. Finally, we show how learning classes of structures relates to learning classes of languages by mapping learning tasks for structures to equivalent learning tasks for languages.

- [42] Doerr, B., Kötzing, T., [Multiplicative Up-Drift](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, 2019.

Drift analysis aims at translating the expected progress of an evolutionary algorithm (or more generally, a random process) into a probabilistic guarantee on its run time (hitting time). So far, drift arguments have been successfully employed in the rigorous analysis of evolutionary algorithms, however, only for the situation that the progress is constant or becomes weaker when approaching the target. Motivated by questions like how fast fit individuals take over a population, we analyze random processes exhibiting a multiplicative growth in expectation. We prove a drift theorem translating this expected progress into a hitting time. This drift theorem gives a simple and insightful proof of the level-based theorem first proposed by Lehre (2011). Our version of this theorem has, for the first time, the best-possible linear dependence on the growth parameter δ (the previous-best was quadratic). This gives immediately stronger run time guarantees for a number of applications.

- [43] Peters, J., Stephan, D., Amon, I., Gawendowicz, H., Lischoid, J., Salabarría, J., Umland, J., Werner, F., Krejca, M. S., Rothenberger, R., Kötzing, T., Friedrich, T., [Mixed Integer Programming versus Evolutionary Computation for Optimizing a Hard Real-World Staff Assignment Problem](#). In: *International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 541–554, 2019.

Assigning staff to engagements according to hard constraints while optimizing several objectives is a task encountered by many companies on a regular basis. Simplified versions of such assignment problems are NP-hard. Despite this, a typical approach to solving them consists of formulating them as mixed integer programming (MIP) problems and using a state-of-the-art solver to get solutions that closely approximate the optimum. In this paper, we consider a complex real-world staff assignment problem encountered by the professional service company KPMG, with the goal of finding an algorithm that solves it faster and with a better solution than a commercial MIP solver. We follow the evolutionary algorithm (EA) metaheuristic and design a search heuristic which iteratively improves a solution using domain-specific mutation operators. Furthermore, we use a flow algorithm to optimally solve a subproblem, which tremendously reduces the search space for the EA. For our real-world instance of the assignment problem, given the same total time budget of 100 hours, a parallel EA approach finds a solution that is only 1.7% away from an upper bound for the (unknown) optimum within under five hours, while the MIP solver Gurobi still has a gap of 10.5 %.

- [44] Friedrich, T., Kötzing, T., Quinzan, F., Sutton, A. M., [Improving the Run Time of the \(1+1\) Evolutionary Algorithm with Luby Sequences](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 301–308, 2018.

In the context of black box optimization, the most common way to handle deceptive attractors is to periodically restart the algorithm. In this paper, we explore the benefits of combining the simple (1+1) Evolutionary Algorithm (EA) with the Luby Universal Strategy - the (1+1) EA_U , a meta-heuristic that does not require parameter tuning. We first consider two artificial pseudo-Boolean landscapes, on which the (1+1) EA exhibits exponential run time. We prove that the (1+1) EA_U has polynomial run time on both instances. We then consider the Minimum Vertex Cover on two classes of graphs. Again, the (1+1) EA yields exponential run time on those instances, and the (1+1) EA_U finds the global optimum in polynomial time. We conclude by studying the Makespan Scheduling. We consider an instance on which the (1+1) EA does not find a $(4/3 - \epsilon)$ -approximation in polynomial time, and we show that the (1+1) EA_U reaches a $(4/3 - \epsilon)$ -approximation in polynomial time. We then prove that the (1+1) EA_U serves as an Efficient Polynomial-time Approximation Scheme (EPTAS) for the Partition Problem, for a $(1 + \epsilon)$ -approximation with $\epsilon > 4/n$.

- [45] Frahnow, C., Kötzing, T., [Ring Migration Topology Helps Bypassing Local Optima](#). In: *Parallel Problem Solving From Nature (PPSN)*, pp. 129–140, 2018.

Running several evolutionary algorithms in parallel and occasionally exchanging good solutions is referred to as island models. The idea is that the independence of the different islands leads to diversity, thus possibly exploring the search space better. Many theoretical analyses so far have found a complete (or sufficiently quickly expanding) topology as underlying migration graph most efficient for optimization, even though a quick dissemination of individuals leads to a loss of diversity. We suggest a simple fitness function Fork with two local optima parametrized by $r \geq 2$ and a scheme for composite fitness functions. We show that, while the (1+1) EA gets stuck in a bad local optimum and incurs a run time of $\Theta(n^{2^r})$ fitness evaluations on Fork, island models with a complete topology can achieve a run time of $\Theta(n^{1.5^r})$ by making use of rare migrations in order to explore the search space more effectively. Finally, the ring topology, making use of rare migrations and a large diameter, can achieve a run time of $\tilde{\Theta}(n^r)$, the black box complexity of Fork. This shows that the ring topology can be preferable over the complete topology in order to maintain diversity.

- [46] Kötzing, T., Krejca, M. S., [First-Hitting Times Under Additive Drift](#). In: *Parallel Problem Solving From Nature (PPSN)*, pp. 92–104, 2018.

For the last ten years, almost every theoretical result concerning the expected run time of a randomized search heuristic used drift theory, making it the arguably most important tool in this domain. Its success is due to its ease of use and its powerful result: drift theory allows the user to derive bounds on the expected first-hitting time of a random process by bounding expected local changes of the process – the drift. This is usually far easier than bounding the expected first-hitting time directly. Due to the widespread use of drift theory, it is of utmost importance to have the best drift theorems possible. We improve the fundamental additive, multiplicative, and variable drift theorems by stating them in a form as general as possible and providing examples of why the restrictions we keep are still necessary. Our additive drift theorem for upper bounds only requires the process to be nonnegative, that is, we remove unnecessary restrictions like a finite, discrete, or bounded search space. As corollaries, the same is true for our upper bounds in the case of variable and multiplicative drift.

- [47] Kötzing, T., Krejca, M. S., [First-Hitting Times for Finite State Spaces](#). In: *Parallel Problem Solving From Nature (PPSN)*, pp. 79–91, 2018.

One of the most important aspects of a randomized algorithm is bounding its expected run time on various problems. Formally speaking, this means bounding the expected first-hitting time of a random process. The two arguably most popular tools to do so are the fitness level method and drift theory. The fitness level method considers arbitrary transition probabilities but only allows the process to move toward the goal. On the other hand, drift theory allows the process to move into any direction as long as it moves closer to the goal in expectation; however, this tendency has to be monotone and, thus, the transition probabilities cannot be arbitrary. We provide a result that combines the benefit of these two approaches: our result gives a lower and an upper bound for the expected first-hitting time of a random process over $\{0, \dots, n\}$ that is allowed to move forward and backward by 1 and can use arbitrary transition probabilities. In case that the transition probabilities are known, our bounds coincide and yield the exact value of the expected first-hitting time. Further, we also state the stationary distribution as well as the mixing time of a special case of our scenario.

- [48] Kötzing, T., Lagodzinski, J. A. G., Lengler, J., Melnichenko, A., [Destructiveness of Lexicographic Parsimony Pressure and Alleviation by a Concatenation Crossover in Genetic Programming](#). In: *Parallel Problem Solving From Nature (PPSN)*, pp. 42–54, 2018.

For theoretical analyses there are two specifics distinguishing GP from many other areas of evolutionary computation. First, the variable size representations, in particular yielding a possible bloat (i.e. the growth of individuals with redundant parts). Second, the role and realization of crossover, which is particularly central in GP due to the tree-based representation. Whereas some theoretical work on GP has studied the effects of bloat, crossover had a surprisingly little share in this work. We analyze a simple crossover operator in combination with local search, where a preference for small solutions minimizes bloat (lexicographic parsimony pressure); the resulting algorithm is denoted ConcatenationCrossover GP. For this purpose three variants of the well-studied Majority test function with large plateaus are considered. We show that the Concatenation Crossover GP can efficiently optimize these test functions, while local search cannot be efficient for all three variants independent of employing bloat control.

- [49] Kötzing, T., Schirneck, M., Seidel, K., [Normal Forms in Semantic Language Identification](#). In: *Algorithmic Learning Theory (ALT)*, pp. 493–516, 2017.

We consider language learning in the limit from text where all learning restrictions are semantic, that is, where any conjecture may be replaced by a semantically equivalent conjecture. For different such learning criteria, starting with the well-known TxtGBC-learning, we consider three different normal forms: strongly locking learning, consistent learning and (partially) set-driven learning. These normal forms support and simplify proofs and give insight into what behaviors are necessary for successful learning (for example when consistency in conservative learning implies cautiousness and strong decisiveness). We show that strongly locking learning can be assumed for partially set-driven learners, even when learning restrictions apply. We give a very general proof relying only on a natural property of the learning restriction, namely, allowing for simulation on equivalent text. Furthermore, when no restrictions apply, also the converse is true: every strongly locking learner can be made partially set-driven. For several semantic learning criteria we show that learning can be done consistently. Finally, we deduce for which learning restrictions partial set-drivenness and set-drivenness coincide, including a general statement about classes of infinite languages. The latter again relies on a simulation argument.

- [50] Gao, W., Friedrich, T., Kötzing, T., Neumann, F., [Scaling up Local Search for Minimum Vertex Cover in Large Graphs by Parallel Kernelization](#). In: *Australasian Conference on Artificial Intelligence (AUSAI)*, pp. 131–143, 2017.

We investigate how well-performing local search algorithms for small or medium size instances can be scaled up to perform well for large inputs. We introduce a parallel kernelization technique that is motivated by the assumption that graphs in medium to large scale are composed of components which are on their own easy for state-of-the-art solvers but when hidden in large graphs are hard to solve. To show the effectiveness of our kernelization technique, we consider the well-known minimum vertex cover problem and two state-of-the-art solvers called NuMVC and FastVC. Our kernelization approach reduces an existing large problem instance significantly and produces better quality results on a wide range of benchmark instances and real world graphs.

- [51] Friedrich, T., Kötzing, T., Wagner, M., [A Generic Bet-and-Run Strategy for Speeding Up Stochastic Local Search](#). In: *Conference on Artificial Intelligence (AAAI)*, pp. 801–807, 2017.

A common strategy for improving optimization algorithms is to restart the algorithm when it is believed to be trapped in an inferior part of the search space. However, while specific restart strategies have been developed for specific problems (and specific algorithms), restarts are typically not regarded as a general tool to speed up an optimization algorithm. In fact, many optimization algorithms do not employ restarts at all. Recently, "bet-and-run" was introduced in the context of mixed-integer programming, where first a number of short runs with randomized initial conditions is made, and then the most promising run of these is continued. In this article, we consider two classical NP-complete combinatorial optimization problems, traveling salesperson and minimum vertex cover, and study the effectiveness of different bet-and-run strategies. In particular, our restart strategies do not take any problem knowledge into account, nor are tailored to the optimization algorithm. Therefore, they can be used off-the-shelf. We observe that state-of-the-art solvers for these problems can benefit significantly from restarts on standard benchmark instances.

- [52] Friedrich, T., Kötzing, T., Quinzan, F., Sutton, A. M., [Resampling vs Recombination: a Statistical Run Time Estimation](#). In: *Foundations of Genetic Algorithms (FOGA)*, pp. 25–35, 2017.

Noise is pervasive in real-world optimization, but there is still little understanding of the interplay between the operators of randomized search heuristics and explicit noise-handling techniques, such as statistical resampling. In this paper, we report on several statistical models and theoretical results that help to clarify this reciprocal relationship for a collection of randomized search heuristics on noisy functions. We consider the optimization of pseudo-Boolean functions under additive posterior Gaussian noise and explore the trade-off between noise reduction and the computational cost of resampling. We first perform experiments to find the optimal parameters at a given noise intensity for a mutation-only evolutionary algorithm, a genetic algorithm employing recombination, an estimation of distribution algorithm (EDA), and an ant colony optimization algorithm. We then observe how the optimal parameter depends on the noise intensity for the different algorithms. Finally, we locate the point where statistical resampling costs more than it is worth in terms of run time. We find that the EA requires the highest number of resamples to obtain the best speed-up, whereas crossover reduces both the run time and the number of resamples required. Most surprisingly, we find that EDA-like algorithms require no resampling, and can handle noise implicitly.

- [53] Friedrich, T., Kötzing, T., Lagodzinski, J. A. G., Neumann, F., Schirneck, M., [Analysis of the \(1+1\) EA on Subclasses of Linear Functions under Uniform and Linear Constraints](#). In: *Foundations of Genetic Algorithms (FOGA)*, pp. 45–54, 2017.

Linear functions have gained a lot of attention in the area of run time analysis of evolutionary computation methods and the corresponding analyses have provided many effective tools for analyzing more complex problems. In this paper, we consider the behavior of the classical (1+1) Evolutionary Algorithm for linear functions under linear constraint. We show tight bounds in the case where both the objective function and the constraint is given by the OneMax function and present upper bounds as well as lower bounds for the general case. Furthermore, we also consider the LeadingOnes fitness function.

- [54] Friedrich, T., Kötzing, T., Melnichenko, A., [Analyzing Search Heuristics with Differential Equations](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 313–314, 2017.

Drift Theory is currently the most common technique for the analysis of randomized search heuristics because of its broad applicability and the resulting tight first hitting time bounds. The biggest problem when applying a drift theorem is to find a suitable potential function which maps a complex space into a single number, capturing the essence of the state of the search in just one value. We discuss another method for the analysis of randomized search heuristics based on the Theory of Differential Equations. This method considers the deterministic counterpart of the randomized process by replacing probabilistic outcomes by their expectation, and then bounding the error with good probability. We illustrate this by analyzing an Ant Colony Optimization algorithm (ACO) for the Minimum Spanning Tree problem (MST).

- [55] Doerr, B., Kötzing, T., Lagodzinski, J. A. G., Lengler, J., [Bounding Bloat in Genetic Programming](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 921–928, 2017.

While many optimization problems work with a fixed number of decision variables and thus a fixed-length representation of possible solutions, genetic programming (GP) works on variable-length representations. A naturally occurring problem is that of bloat (unnecessary growth of solutions) slowing down optimization. Theoretical analyses could so far not bound bloat and required explicit assumptions on the magnitude of bloat. In this paper we analyze bloat in mutation-based genetic programming for the two test functions ORDER and MAJORITY. We overcome previous assumptions on the magnitude of bloat and give matching or close-to-matching upper and lower bounds for the expected optimization time. In particular, we show that the $(1+1)$ GP takes (i) $\Theta(T_{\text{init}} + n \log n)$ iterations with bloat control on ORDER as well as MAJORITY; and (ii) $O(T_{\text{init}} \log T_{\text{init}} + n(\log n)^3)$ and $\Omega(T_{\text{init}} + n \log n)$ (and $\Omega(T_{\text{init}} \log T_{\text{init}})$ for $n = 1$) iterations without bloat control on MAJORITY.

- [56] Shi, F., Schirneck, M., Friedrich, T., Kötzing, T., Neumann, F., [Reoptimization Times of Evolutionary Algorithms on Linear Functions Under Dynamic Uniform Constraints](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 1407–1414, 2017.

The investigations of linear pseudo-Boolean functions play a central role in the area of runtime analysis of evolutionary computing techniques. Having an additional linear constraint on a linear function is equivalent to the NP-hard knapsack problem and special problem classes thereof have been investigated in recent works. In this paper, we extend these studies to problems with dynamic constraints and investigate the runtime of different evolutionary algorithms to recompute an optimal solution when the constraint bound changes by a certain amount. We study the classical $(1+1)$ EA and population-based algorithms and show that they recompute an optimal solution very efficiently. Furthermore, we show that a variant of the $(1 + (\lambda, \lambda))$ GA can recompute the optimal solution more efficiently in some cases.

- [57] Doerr, B., Doerr, C., Kötzing, T., [Unknown Solution Length Problems With No Asymptotically Optimal Run Time](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 1367–1374, 2017.

We revisit the problem of optimizing a fitness function of unknown dimension; that is, we face a function defined over bit-strings of large length N , but only $n \ll N$ of them have an influence on the fitness. Neither the position of these relevant bits nor their number is known. In previous work, variants of the $(1+1)$ evolutionary algorithm (EA) have been developed that solve, for arbitrary $s \in N$, such OneMax and LeadingOnes instances, simultaneously for all $n \in N$, in expected time $O(n(\log(n))^2 \log \log(n) \dots \log^{s-1}(n)(\log^{(s)}(n))^{1+\epsilon})$ and $O(n^2 \log(n) \log \log(n) \dots \log^{s-1}(n)(\log^{(s)}(n))^{1+\epsilon})$, respectively; that is, in almost the same time as if n and the relevant bit positions were known. In this work, we prove the first, almost matching, lower bounds for this setting. For LeadingOnes, we show that, for every $s \in N$, the $(1+1)$ EA with any mutation operator treating zeros and ones equally has an expected run time of $\omega(n^2 \log(n) \log \log(n) \dots \log^{(s)}(n))$ when facing problem size n . Aiming at closing the small remaining gap, we realize that, quite surprisingly, there is no asymptotically best performance. For any algorithm solving, for all n , all instances of size n in expected time at most $T(n)$, there is an algorithm doing the same in time $T'(n)$ with $T' = o(T)$. For OneMax we show results of similar flavor.

- [58] Doerr, B., Fischbeck, P., Frahnw, C., Friedrich, T., Kötzing, T., Schirneck, M., [Island Models Meet Rumor Spreading](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 1359–1366, 2017.

Island models in evolutionary computation solve problems by a careful interplay of independently running evolutionary algorithms on the island and an exchange of good solutions between the islands. In this work, we conduct rigorous run time analyses for such island models trying to simultaneously obtain good run times and low communication effort. We improve the existing upper bounds for the communication effort (i) by improving the run time bounds via a careful analysis, (ii) by setting the balance between individual computation and communication in a more appropriate manner, and (iii) by replacing the usual communicate-with-all-neighbors approach with randomized rumor spreading, where each island contacts a randomly chosen neighbor. This epidemic communication paradigm is known to lead to very fast and robust information dissemination in many applications. Our results concern islands running simple $(1+1)$ evolutionary algorithms, we regard d -dimensional tori and complete graphs as communication topologies, and optimize the classic test functions OneMax and LeadingOnes.

- [59] Friedrich, T., Kötzing, T., Krejca, M. S., Sutton, A. M., [The Benefit of Recombination in Noisy Evolutionary Search](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 161–162, 2016.

Practical optimization problems frequently include uncertainty about the quality measure, for example due to noisy evaluations. Thus, they do not allow for a straightforward application of traditional optimization techniques. In these settings, randomized search heuristics such as evolutionary algorithms are a popular choice because they are often assumed to exhibit some kind of resistance to noise. Empirical evidence suggests that some algorithms, such as estimation of distribution algorithms (EDAs) are robust against a scaling of the noise intensity, even without resorting to explicit noise-handling techniques such as resampling. In this paper, we want to support such claims with mathematical rigor. We introduce the concept of graceful scaling in which the run time of an algorithm scales polynomially with noise intensity. We study a monotone fitness function over binary strings with additive noise taken from a Gaussian distribution. We show that myopic heuristics cannot efficiently optimize the function under arbitrarily intense noise without any explicit noise-handling. Furthermore, we prove that using a population does not help. Finally we show that a simple EDA called the Compact Genetic Algorithm can overcome the shortsightedness of mutation-only heuristics to scale gracefully with noise. We conjecture that recombinative genetic algorithms also have this property.

- [60] Friedrich, T., Kötzing, T., Quinzan, F., Sutton, A. M., [Ant Colony Optimization Beats Resampling on Noisy Functions](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 3–4, 2016.

Despite the pervasiveness of noise in real-world optimization, there is little understanding of the interplay between the operators of randomized search heuristics and explicit noise-handling techniques such as statistical resampling. Ant Colony Optimization (ACO) algorithms are claimed to be particularly well-suited to dynamic and noisy problems, even without explicit noise-handling techniques. In this work, we empirically investigate the trade-offs between resampling and the noise-handling abilities of ACO algorithms. Our main focus is to locate the point where resampling costs more than it is worth.

- [61] Dang, D.-C., Friedrich, T., Krejca, M. S., Kötzing, T., Lehre, P. K., Oliveto, P. S., Sudholt, D., Sutton, A. M., [Escaping Local Optima with Diversity Mechanisms and Crossover](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 645–652, 2016.

Population diversity is essential for the effective use of any crossover operator. We compare seven commonly used diversity mechanisms and prove rigorous run time bounds for the $(\mu + 1)$ GA using uniform crossover on the fitness function $Jump_k$. All previous results in this context only hold for unrealistically low crossover probability $p_c = O(k/n)$, while we give analyses for the setting of constant $p_c < 1$ in all but one case. Our bounds show a dependence on the problem size n , the jump length k , the population size μ , and the crossover probability p_c . For the typical case of constant $k > 2$ and constant p_c , we can compare the resulting expected optimisation times for different diversity mechanisms assuming an optimal choice of μ : $O(n^{k-1})$ for duplicate elimination/minimisation, $O(n^2 \log n)$ for maximising the convex hull, $O(n \log n)$ for det. crowding (assuming $p_c = k/n$), $O(n \log n)$ for maximising the Hamming distance, $O(n \log n)$ for fitness sharing, $O(n \log n)$ for the single-receiver island model. This proves a sizeable advantage of all variants of the $(\mu + 1)$ GA compared to the $(1+1)$ EA, which requires $\Theta(n^k)$. In a short empirical study we confirm that the asymptotic differences can also be observed experimentally.

- [62] Friedrich, T., Kötzing, T., Krejca, M. S., Nallaperuma, S., Neumann, F., Schirneck, M., [Fast Building Block Assembly by Majority Vote Crossover](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 661–668, 2016.

Different works have shown how crossover can help with building block assembly. Typically, crossover might get lucky to select good building blocks from each parent, but these lucky choices are usually rare. In this work we consider a crossover operator which works on three parent individuals. In each component, the offspring inherits the value present in the majority of the parents; thus, we call this crossover operator majority vote. We show that, if good components are sufficiently prevalent in the individuals, majority vote creates an optimal individual with high probability. Furthermore, we show that this process can be amplified: as long as components are good independently and with probability at least $1/2 + \delta$, we require only $O(\log 1/\delta + \log \log n)$ successive stages of majority vote to create an optimal individual with high probability! We show how this applies in two scenarios. The first scenario is the Jump test function. With sufficient diversity, we get an optimization time of $O(n \log n)$ even for jump sizes as large as $O(n^{(1/2-\epsilon)})$. Our second scenario is a family of vertex cover instances. Majority vote optimizes this family efficiently, while local searches fail and only highly specialized two-parent crossovers are successful.

- [63] Friedrich, T., Kötzing, T., Krejca, M. S., [EDAs cannot be Balanced and Stable](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 1139–1146, 2016.

Estimation of Distribution Algorithms (EDAs) work by iteratively updating a distribution over the search space with the help of samples from each iteration. Up to now, theoretical analyses of EDAs are scarce and present run time results for specific EDAs. We propose a new framework for EDAs that captures the idea of several known optimizers, including PBIL, UMDA, λ -MMASIB, cGA, and $(1, \lambda)$ -EA. Our focus is on analyzing two core features of EDAs: a balanced EDA is sensitive to signals in the fitness; a stable EDA remains uncommitted under a biasless fitness function. We prove that no EDA can be both balanced and stable. The LeadingOnes function is a prime example where, at the beginning of the optimization, the fitness function shows no bias for many bits. Since many well-known EDAs are balanced and thus not stable, they are not well-suited to optimize LeadingOnes. We give a stable EDA which optimizes LeadingOnes within a time of $O(n \log n)$.

- [64] Doerr, B., Doerr, C., Kötzing, T., [The Right Mutation Strength for Multi-Valued Decision Variables](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 1115–1122, 2016.

The most common representation in evolutionary computation are bit strings. This is ideal to model binary decision variables, but less useful for variables taking more values. With very little theoretical work existing on how to use evolutionary algorithms for such optimization problems, we study the run time of simple evolutionary algorithms on some OneMax-like functions defined over $\Omega = \{0, 1, \dots, r-1\}^n$. More precisely, we regard a variety of problem classes requesting the component-wise minimization of the distance to an unknown target vector $z \in \Omega$. For such problems we see a crucial difference in how we extend the standard-bit mutation operator to these multi-valued domains. While it is natural to select each position of the solution vector to be changed independently with probability $1/n$, there are various ways to then change such a position. If we change each selected position to a random value different from the original one, we obtain an expected run time of $\Theta(nr \log n)$. If we change each selected position by either $+1$ or -1 (random choice), the optimization time reduces to $\Theta(nr + n \log n)$. If we use a random mutation strength $i \in \{0, 1, \dots, r-1\}^n$ with probability inversely proportional to i and change the selected position by either $+i$ or $-i$ (random choice), then the optimization time becomes $\Theta(n \log(r)(\log(n) + \log(r)))$, bringing down the dependence on r from linear to polylogarithmic. One of our results depends on a new variant of the lower bounding multiplicative drift theorem.

- [65] Dang, D.-C., Lehre, P. K., Friedrich, T., Kötzing, T., Krejca, M. S., Oliveto, P. S., Sudholt, D., Sutton, A. M., [Emergence of Diversity and its Benefits for Crossover in Genetic Algorithms](#). In: *Parallel Problem Solving From Nature (PPSN)*, pp. 890–900, 2016.

Population diversity is essential for avoiding premature convergence in Genetic Algorithms (GAs) and for the effective use of crossover. Yet the dynamics of how diversity emerges in populations are not well understood. We use rigorous runtime analysis to gain insight into population dynamics and GA performance for a standard $(\mu + 1)$ GA and the $Jump_k$ test function. By studying the stochastic process underlying the size of the largest collection of identical genotypes we show that the interplay of crossover followed by mutation may serve as a catalyst leading to a sudden burst of diversity. This leads to improvements of the expected optimisation time of order $\Omega(n/\log n)$ compared to mutation-only algorithms like the $(1 + 1)$ EA.

- [66] Friedrich, T., Kötzing, T., Sutton, A. M., [On the Robustness of Evolving Populations](#). In: *Parallel Problem Solving From Nature (PPSN)*, pp. 771–781, 2016.

Most theoretical work that studies the benefit of recombination focuses on the ability of crossover to speed up optimization time on specific search problems. In this paper, we take a slightly different perspective and investigate recombination in the context of evolving solutions that exhibit mutational robustness, i.e., they display insensitivity to small perturbations. Various models in population genetics have demonstrated that increasing the effective recombination rate promotes the evolution of robustness. We show this result also holds in the context of evolutionary computation by proving crossover promotes the evolution of robust solutions in the standard $(\mu + 1)$ GA. Surprisingly, our results show that the effect is present even when robust solutions are at a selective disadvantage due to lower fitness values.

- [67] Friedrich, T., Kötzing, T., Krejca, M. S., Sutton, A. M., [Graceful Scaling on Uniform versus Steep-Tailed Noise](#). In: *Parallel Problem Solving From Nature (PPSN)*, pp. 761–770, 2016.

Recently, different evolutionary algorithms (EAs) have been analyzed in noisy environments. The most frequently used noise model for this was additive posterior noise (noise added after the fitness evaluation) taken from a Gaussian distribution. In particular, for this setting it was shown that the $(\mu + 1)$ -EA on OneMax does not scale gracefully (higher noise cannot efficiently be compensated by higher μ). In this paper we want to understand whether there is anything special about the Gaussian distribution which makes the $(\mu + 1)$ -EA not scale gracefully. We keep the setting of posterior noise, but we look at other distributions. We see that for exponential tails the $(\mu + 1)$ -EA on OneMax does also not scale gracefully, for similar reasons as in the case of Gaussian noise. On the other hand, for uniform distributions (as well as other, similar distributions) we see that the $(\mu + 1)$ -EA on OneMax does scale gracefully, indicating the importance of the noise model.

- [68] Doerr, B., Doerr, C., Kötzing, T., [Provably Optimal Self-Adjusting Step Sizes for Multi-Valued Decision Variables](#). In: *Parallel Problem Solving From Nature (PPSN)*, pp. 782–791, 2016.

We regard the problem of maximizing a OneMax-like function defined over an alphabet of size r . In previous work [GECCO 2016] we have investigated how three different mutation operators influence the performance of Randomized Local Search (RLS) and the $(1+1)$ Evolutionary Algorithm. This work revealed that among these natural mutation operators none is superior to the other two for any choice of r . We have also given in [GECCO 2016] some indication that the best achievable run time for large r is $\Theta(n \log r (\log n + \log r))$, regardless of how the mutation operator is chosen, as long as it is a static choice (i.e., the distribution used for variation of the current individual does not change over time). Here in this work we show that we can achieve a better performance if we allow for adaptive mutation operators. More precisely, we analyze the performance of RLS using a self-adjusting mutation strength. In this algorithm the size of the steps taken in each iteration depends on the success of previous iterations. That is, the mutation strength is increased after a successful iteration and it is decreased otherwise. We show that this idea yields an expected optimization time of $\Theta(n(\log n + \log r))$, which is optimal among all comparison-based search heuristics. This is the first time that self-adjusting parameter choices are shown to outperform static choices on a discrete multi-valued optimization problem.

- [69] Kötzing, T., Schirneck, M., [Towards an Atlas of Computational Learning Theory](#). In: *Symposium on Theoretical Aspects of Computer Science (STACS)*, pp. 47:1–47:13, 2016.

A major part of our knowledge about Computational Learning stems from comparisons of the learning power of different learning criteria. These comparisons inform about trade-offs between learning restrictions and, more generally, learning settings; furthermore, they inform about what restrictions can be observed without losing learning power. With this paper we propose that one main focus of future research in Computational Learning should be on a structured approach to determine the relations of different learning criteria. In particular, we propose that, for small sets of learning criteria, all pairwise relations should be determined; these relations can then be easily depicted as a map, a diagram detailing the relations. Once we have maps for many relevant sets of learning criteria, the collection of these maps is an Atlas of Computational Learning Theory, informing at a glance about the landscape of computational learning just as a geographical atlas informs about the earth. In this paper we work toward this goal by providing three example maps, one pertaining to partially set-driven learning, and two pertaining to strongly monotone learning. These maps can serve as blueprints for future maps of similar base structure.

- [70] Kötzing, T., Lissovoi, A., Witt, C., [\(1+1\) EA on Generalized Dynamic OneMax](#). In: *Foundations of Genetic Algorithms (FOGA)*, pp. 40–51, 2015.

Evolutionary algorithms (EAs) perform well in settings involving uncertainty, including settings with stochastic or dynamic fitness functions. In this paper, we analyze the $(1+1)$ EA on dynamically changing OneMax, as introduced by Droste (2003). We re-prove the known results on first hitting times using the modern tool of drift analysis. We extend these results to search spaces which allow for more than two values per dimension. Furthermore, we make an anytime analysis as suggested by Jansen and Zarges (2014), analyzing how closely the $(1+1)$ EA can track the dynamically moving optimum over time. We get tight bounds both for the case of bit strings, as well as for the case of more than two values per position. Surprisingly, in the latter setting, the expected quality of the search point maintained by the $(1+1)$ EA does not depend on the number of values per dimension.

- [71] Doerr, B., Doerr, C., Kötzing, T., [Solving Problems with Unknown Solution Length at \(Almost\) No Extra Cost](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 831–838, 2015.

Most research in the theory of evolutionary computation assumes that the problem at hand has a fixed problem size. This assumption does not always apply to real-world optimization challenges, where the length of an optimal solution may be unknown a priori. Following up on previous work of Cathabard, Lehre, and Yao [FOGA 2011] we analyze variants of the $(1+1)$ evolutionary algorithm for problems with unknown solution length. For their setting, in which the solution length is sampled from a geometric distribution, we provide mutation rates that yield an expected optimization time that is of the same order as that of the $(1+1)$ EA knowing the solution length. We then show that almost the same run times can be achieved even if no a priori information on the solution length is available. Finally, we provide mutation rates suitable for settings in which neither the solution length nor the positions of the relevant bits are known. Again we obtain almost optimal run times for the OneMax and LeadingOnes test functions, thus solving an open problem from Cathabard et al.

- [72] Friedrich, T., Kötzing, T., Krejca, M. S., Sutton, A. M., [Robustness of Ant Colony Optimization to Noise](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 17–24, 2015. [Best-Paper Award \(ACO/SI Track\)](#).

Recently Ant Colony Optimization (ACO) algorithms have been proven to be efficient in uncertain environments, such as noisy or dynamically changing fitness functions. Most of these analyses focus on combinatorial problems, such as path finding. We analyze an ACO algorithm in a setting where we try to optimize the simple OneMax test function, but with additive posterior noise sampled from a Gaussian distribution. Without noise the classical $(\mu + 1)$ -EA outperforms any ACO algorithm, with smaller μ being better; however, with large noise, the $(\mu + 1)$ -EA fails, even for high values of μ (which are known to help against small noise). In this paper

we show that ACO is able to deal with arbitrarily large noise in a graceful manner, that is, as long as the evaporation factor μ is small enough dependent on the parameter δ^2 of the noise and the dimension n of the search space ($p = o(1/(n(n + \delta \log n)^2 \log n))$), optimization will be successful.

- [73] Friedrich, T., Kötzing, T., Krejca, M. S., Sutton, A. M., [The Benefit of Recombination in Noisy Evolutionary Search](#). In: *International Symposium of Algorithms and Computation (ISAAC)*, pp. 140–150, 2015.

Practical optimization problems frequently include uncertainty about the quality measure, for example due to noisy evaluations. Thus, they do not allow for a straightforward application of traditional optimization techniques. In these settings meta-heuristics are a popular choice for deriving good optimization algorithms, most notably evolutionary algorithms which mimic evolution in nature. Empirical evidence suggests that genetic recombination is useful in uncertain environments because it can stabilize a noisy fitness signal. With this paper we want to support this claim with mathematical rigor. The setting we consider is that of noisy optimization. We study a simple noisy fitness function that is derived by adding Gaussian noise to a monotone function. First, we show that a classical evolutionary algorithm that does not employ sexual recombination (the $(\mu + 1)$ -EA) cannot handle the noise efficiently, regardless of the population size. Then we show that an evolutionary algorithm which does employ sexual recombination (the Compact Genetic Algorithm, short: cGA) can handle the noise using a graceful scaling of the population.

- [74] Kötzing, T., Palenta, R., [A Map of Update Constraints in Inductive Inference](#). In: *Algorithmic Learning Theory (ALT)*, pp. 40–54, 2014.

We investigate how different learning restrictions reduce learning power and how the different restrictions relate to one another. We give a complete map for nine different restrictions both for the cases of complete information learning and set-driven learning. This completes the picture for these well-studied delayable learning restrictions. A further insight is gained by different characterizations of conservative learning in terms of variants of cautious learning. Our analyses greatly benefit from general theorems we give, for example showing that learners with exclusively delayable restrictions can always be assumed total.

- [75] Jain, S., Kötzing, T., Ma, J., Stephan, F., [On the Role of Update Constraints and Text-Types in Iterative Learning](#). In: *Algorithmic Learning Theory (ALT)*, pp. 55–69, 2014.

The present work investigates the relationship of iterative learning with other learning criteria such as decisiveness, caution, reliability, non-U-shapedness, monotonicity, strong monotonicity and conservativeness. Building on the result of Case and Moelius that iterative learners can be made non-U-shaped, we show that they also can be made cautious and decisive. Furthermore, we obtain various special results with respect to one-one texts, fat texts and one-one hypothesis spaces.

- [76] Doerr, B., Doerr, C., Kötzing, T., [Unbiased black-box complexities of jump functions: how to cross large plateaus](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 769–776, 2014.

We analyze the unbiased black-box complexity of jump functions with large jump sizes. Among other results, we show that when the jump size is $(1/2 - \epsilon)n$, that is, only a small constant fraction of the fitness values is visible, then the unbiased black-box complexities for arities 3 and higher are of the same order as those for the simple OneMax function. Even for the extreme jump function, in which all but the two fitness values $n/2$ and n are blanked out, polynomial-time mutation-based (i.e., unary unbiased) black-box optimization algorithms exist. This is quite surprising given that for the extreme jump function almost the whole search space (all but a $\Theta(n^{-1/2})$ fraction) is a plateau of constant fitness. To prove these results, we introduce new tools for the analysis of unbiased black-box complexities, for example, selecting the new parent individual not by comparing the fitnesses of the competing search points, but also by taking into account the (empirical) expected fitnesses of their offspring.

- [77] Kötzing, T. [Concentration of first hitting times under additive drift](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 1391–1398, 2014.

Recent advances in drift analysis have given us better and better tools for understanding random processes, including the run time of randomized search heuristics. In the setting of multiplicative drift we do not only have excellent bounds on the expected run time, but also more general results showing the concentration of the run time. In this paper we investigate the setting of additive drift under the assumption of strong concentration of the "step size" of the process. Under sufficiently strong drift towards the goal we show a strong concentration of the hitting time. In contrast to this, we show that in the presence of small drift a Gambler's-Ruin-like behavior of the process overrides the influence of the drift. Finally, in the presence of sufficiently strong negative drift the hitting time is superpolynomial with high probability; this corresponds to the so-called negative drift theorem, for which we give new variants.

- [78] Gießen, C., Kötzing, T., [Robustness of populations in stochastic environments](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 1383–1390, 2014.

We consider stochastic versions of OneMax and LeadingOnes and analyze the performance of evolutionary algorithms with and without populations on these problems. It is known that the $(1+1)$ EA on OneMax performs well in the presence of very small noise, but poorly for higher noise levels. We extend these results to LeadingOnes and to many different noise models, showing how the application of drift theory can significantly simplify and generalize previous analyses. Most surprisingly, even small populations (of size $\Theta(\log n)$) can make evolutionary algorithms perform well for high noise levels, well outside the abilities of the $(1+1)$ EA. Larger population sizes are even more beneficial; we consider both parent and offspring populations. In this sense, populations are robust in these stochastic settings.

- [79] Kötzing, T. [A Solution to Wiehagen's Thesis](#). In: *Symposium on Theoretical Aspects of Computer Science (STACS)*, pp. 494–505, 2014.

Wiehagen's Thesis in Inductive Inference (1991) essentially states that, for each learning criterion, learning can be done in a normalized, enumerative way. The thesis was not a formal statement and thus did not allow for a formal proof, but support was given by examples of a number of different learning criteria that can be learned by enumeration. Building on recent formalizations of learning criteria, we are now able to formalize Wiehagen's Thesis. We prove the thesis for a wide range of learning criteria, including many popular criteria from the literature. We also show the limitations of the thesis by giving four learning criteria for which the thesis does not hold (and, in two cases, was probably not meant to hold). Beyond the original formulation of the thesis, we also prove stronger versions which allow for many corollaries relating to strongly decisive and conservative learning.

- [80] Case, J., Kötzing, T., [Topological Separations in Inductive Inference](#). In: *Algorithmic Learning Theory (ALT)*, pp. 128–142, 2013.

Re learning in the limit from positive data, a major concern is which classes of languages are learnable with respect to a given learning criterion. We are particularly interested herein in the reasons for a class of languages to be unlearnable. We consider two types of reasons. One type is called topological where it does not help if the learners are allowed to be uncomputable (an example of Gold's is that no class containing an infinite language and all its finite sub-languages is learnable - even by an uncomputable learner). Another reason is called computational (where the learners are required to be algorithmic). In particular, two learning criteria might allow for learning different classes of languages from one another - but with dependence on whether the unlearnability is of type topological or computational. In this paper we formalize the idea of two learning criteria separating topologically in learning power. This allows us to study more closely why two learning criteria separate in learning power. For a variety of learning criteria, concerning vacillatory, monotone, (several kinds of) iterative and feedback learning, we show that certain learning criteria separate topologically, and certain others, which are known to separate, are shown not to separate topologically. Showing that learning criteria do not separate topologically implies that any known separation must necessarily exploit algorithmicity of the learner.

- [81] Feldmann, M., Kötzing, T., [Optimizing expected path lengths with ant colony optimization using fitness proportional update](#). In: *Foundations of Genetic Algorithms (FOGA)*, pp. 65–74, 2013.

We study the behavior of a Max-Min Ant System (MMAS) on the stochastic single-destination shortest path (SDSP) problem. Two previous papers already analyzed this setting for two slightly different MMAS algorithms, where the pheromone update fitness-independently rewards edges of the best-so-far solution. The first paper showed that, when the best-so-far solution is not reevaluated and the stochastic nature of the edge weights is due to noise, the MMAS will find a tree of edges successfully and efficiently identify a shortest path tree with minimal noise-free weights. The second paper used reevaluation of the best-so-far solution and showed that the MMAS finds paths which beat any other path in direct comparisons, if existent. For both results, for some random variables, this corresponds to a tree with minimal expected weights. In this work we analyze a variant of MMAS that works with fitness-proportional update on stochastic-weight graphs with arbitrary random edge weights from $[0, 1]$. For δ such that any suboptimal path is worse by at least δ than an optimal path, then, with suitable parameters, the graph will be optimized after $O(n^3 \ln n / \delta \text{over} \delta^3)$ iterations (in expectation). In order to prove the above result, the multiplicative and the variable drift theorem are adapted to continuous search spaces.

- [82] Benz, F., Kötzing, T., [An effective heuristic for the smallest grammar problem](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 487–494, 2013.

The smallest grammar problem is the problem of finding the smallest context-free grammar that generates exactly one given sequence. Approximating the problem with a ratio of less than 8569/8568 is known to be NP-hard. Most work on this problem has focused on finding decent solutions fast (mostly in linear time), rather than on good heuristic algorithms. Inspired by a new perspective on the problem presented by Carrascosa et al. (2010), we investigate the performance of different heuristics on the problem. The aim is to find a good solution on large instances by allowing more than linear time. We propose a hybrid of a max-min ant system and a genetic algorithm that in combination with a novel local search outperforms the state of the art on all files of the Canterbury corpus, a standard benchmark suite. Furthermore, this hybrid performs well on a standard DNA corpus.

- [83] Freydenberger, D. D., Kötzing, T., [Fast learning of restricted regular expressions and DTDs](#). In: *International Conference on Database Theory (ICDT)*, pp. 45–56, 2013.

We study the problem of generalizing from a finite sample to a language taken from a predefined language class. The two language classes we consider are subsets of the regular languages and have significance in the specification of XML documents (the classes corresponding to so called chain regular expressions, Chares, and to single occurrence regular expressions, Sores). The previous literature gave a number of algorithms for generalizing to Sores providing a trade off between quality of the solution and speed. Furthermore, a fast but non-optimal algorithm for generalizing to Chares is known. For each of the two language classes we give an efficient algorithm returning a minimal generalization from the given finite sample to an element of the fixed language class; such generalizations are called descriptive. In this sense, both our algorithms are optimal.

- [84] Croitoru, C., Kötzing, T., [A Normal Form for Argumentation Frameworks](#). In: *Theorie and Applications of Formal Argumentation (TAFa)*, pp. 32–45, 2013.

We study formal argumentation frameworks as introduced by Dung (1995). We show that any such argumentation framework can be syntactically augmented into a normal form (having a simplified attack relation), preserving the semantic properties of original arguments. An argumentation framework is in normal form if no argument attacks a conflicting pair of arguments. An augmentation of an argumentation framework is obtained by adding new arguments and changing the attack relation such that the acceptability status of original arguments is maintained in the new framework. Furthermore, we define join-normal semantics leading to augmentations of the joined argumentation frameworks. Also, a rewriting technique which transforms in cubic time a given argumentation framework into a normal form is devised.

- [85] Bailly, G., Oulasvirta, A., Kötzing, T., Hoppe, S., [MenuOptimizer: interactive optimization of menu systems](#). In: *User Interface Software and Technology (UIST)*, pp. 331–342, 2013.

Menu systems are challenging to design because design spaces are immense, and several human factors affect user behavior. This paper contributes to the design of menus with the goal of interactively assisting designers with an optimizer in the loop. To reach this goal, 1) we extend a predictive model of user performance to account for expectations as to item groupings; 2) we adapt an ant colony optimizer that has been proven efficient for this class of problems; and 3) we present MenuOptimizer, a set of inter-actions integrated into a real interface design tool (QtDesigner). MenuOptimizer supports designers' abilities to cope with uncertainty and recognize good solutions. It allows designers to delegate combinatorial problems to the optimizer, which should solve them quickly enough without disrupting the design process. We show evidence that satisfactory menu designs can be produced for complex problems in minutes.

- [86] Jain, S., Kötzing, T., Stephan, F., [Enlarging Learnable Classes](#). In: *Algorithmic Learning Theory (ALT)*, pp. 36–50, 2012.

An early result in inductive inference shows that the class of Ex-learnable sets is not closed under unions. In this paper we are interested in the following question: For what classes of functions is the union with an arbitrary Ex-learnable class again Ex-learnable, either effectively (in an index for a learner of an Ex-learnable class) or non-effectively? We show that the effective case and the non-effective case separate, and we give a sufficient criterion for the effective case. Furthermore, we extend our notions to considering unions with classes of single functions, as well as to other learning criteria, such as finite learning and behaviorally correct learning. Furthermore, we consider the possibility of (effectively) extending learners to learn (infinitely) more functions. It is known that all Ex-learners learning a dense set of functions can be effectively extended to learn infinitely more. It was open whether the learners learning a non-dense set of functions can be similarly extended. We show that this is not possible, but we give an alternative

split of all possible learners into two sets such that, for each of the sets, all learners from that set can be effectively extended. We analyze similar concepts also for other learning criteria.

- [87] Baumbach, J., Friedrich, T., Kötzing, T., Krohmer, A., Müller, J., Pauling, J., [Efficient Algorithms for Extracting Biological Key Pathways with Global Constraints](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 169–176, 2012.

The integrated analysis of data of different types and with various interdependencies is one of the major challenges in computational biology. Recently, we developed KeyPathwayMiner, a method that combines biological networks modeled as graphs with disease-specific genetic expression data gained from a set of cases (patients, cell lines, tissues, etc.). We aimed for finding all maximal connected sub-graphs where all nodes but K are expressed in all cases but at most L , i.e. key pathways. Thereby, we combined biological networks with OMICS data, instead of analyzing these data sets in isolation. Here we present an alternative approach that avoids a certain bias towards hub nodes: We now aim for extracting all maximal connected sub-networks where all but at most K nodes are expressed in all cases but in total (!) at most L , i.e. accumulated over all cases and all nodes in a solution. We call this strategy GLONE (global node exceptions); the previous problem we call INES (individual node exceptions). Since finding GLONE-components is computationally hard, we developed an Ant Colony Optimization algorithm and implemented it with the KeyPathwayMiner Cytoscape framework as an alternative to the INES algorithms. KeyPathwayMiner 3.0 now offers both the INES and the GLONE algorithms. It is available as plugin from Cytoscape and online at <http://keypathwayminer.mpi-inf.mpg.de>.

- [88] Kötzing, T., Sutton, A. M., Neumann, F., O’Reilly, U.-M., [The max problem revisited: the importance of mutation in genetic programming](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 1333–1340, 2012.

This paper contributes to the rigorous understanding of genetic programming algorithms by providing runtime complexity analyses of the well-studied Max problem. Several experimental studies have indicated that it is hard to solve the Max problem with crossover-based algorithms. Our analyses show that different variants of the Max problem can provably be solved using simple mutation-based genetic programming algorithms. Our results advance the body of computational complexity analyses of genetic programming, indicate the importance of mutation in genetic programming, and reveal new insights into the behavior of mutation-based genetic programming algorithms.

- [89] Doerr, B., Hota, A., Kötzing, T., [Ants easily solve stochastic shortest path problems](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 17–24, 2012.

The first rigorous theoretical analysis (Horoba, Sudholt (GECCO 2010)) of an ant colony optimizer for the stochastic shortest path problem suggests that ant system experience significant difficulties when the input data is prone to noise. In this work, we propose a slightly different ant optimizer to deal with noise. We prove that under mild conditions, it finds the paths with shortest expected length efficiently, despite the fact that we do not have convergence in the classic sense. To prove our results, we introduce a stronger drift theorem that can also deal with the situation that the progress is faster when one is closer to the goal.

- [90] Kötzing, T., Molter, H., [ACO Beats EA on a Dynamic Pseudo-Boolean Function](#). In: *Parallel Problem Solving from Nature (PPSN)*, pp. 113–122, 2012.

In this paper, we contribute to the understanding of the behavior of bio-inspired algorithms when tracking the optimum of a dynamically changing fitness function over time. In particular, we are interested in the difference between a simple evolutionary algorithm (EA) and a simple ant colony optimization (ACO) system on deterministically changing fitness functions, which we call dynamic fitness patterns. Of course, the algorithms have no prior knowledge about the patterns. We construct a bit string optimization problem where we can show that the ACO system is able to follow the optimum while the EA gets lost.

- [91] Croitoru, C., Kötzing, T., [Deliberative Acceptability of Arguments](#). In: *Starting AI Researcher Symposium (STAIRS)*, pp. 71–82, 2012.

State of the art extension based argument acceptance is currently biased toward attacks: while the defending extension of an argument a is internally coherent, no such requirement is imposed on its attacking set. On the other hand, if we restrict ourselves only to conflict-free sets of attacking arguments, then we could have different attacking sets for a specified argument a (any two conflicting attackers of a must belong to different a ’s attacking sets). Having only one defending extension for all these attacking sets would contradict the deliberative nature of argumentation in the real world, where only the coherent sets of attacks matter and the defending sets of arguments depend on the former. In this paper we introduce a new type of acceptability of an argument, in which its attacking and defending sets of arguments are uniformly treated. We call it deliberative acceptance, discuss how this and the classical acceptance notions interrelate and analyze its computational properties. In particular, we prove that the corresponding decision problem is Π_2^P -complete, but its restrictions on bipartite or co-chordal argumentation frameworks are in P .

- [92] Kötzing, T. [Iterative Learning from Positive Data and Counters](#). In: *Algorithmic Learning Theory (ALT)*, pp. 40–54, 2011.

We analyze iterative learning in the limit from positive data with the additional information provided by a counter. The simplest type of counter provides the current iteration number (counting up from 0 to infinity), which is known to improve learning power over plain iterative learning. We introduce five other (weaker) counter types, for example only providing some unbounded and non-decreasing sequence of numbers. Analyzing these types allows one to understand which properties of a counter learning can benefit from. For the iterative setting, we completely characterize the relative power of the learning criteria corresponding to the counter types. In particular, for our types, the only properties improving learning power are unboundedness and strict monotonicity. Furthermore, we show that each of our types of counter improves learning power over weaker ones in some settings; to this end, we analyze transductive and non-U-shaped learning. Finally we show that, for iterative learning criteria with one of our types of counter, separations of learning criteria are necessarily witnessed by classes containing only infinite languages.

- [93] Doerr, B., Johannsen, D., Kötzing, T., Lehre, P. K., Wagner, M., Winzen, C., [Faster black-box algorithms through higher arity operators](#). In: *Foundations of Genetic Algorithms (FOGA)*, pp. 163–172, 2011.

We extend the work of Lehre and Witt (GECCO 2010) on the unbiased black-box model by considering higher arity variation operators. In particular, we show that already for binary operators the black-box complexity of LeadingOnes drops from $\Theta(n^2)$ for unary operators to $O(n \log n)$. For OneMax, the $\Omega(n \log n)$ unary black-box complexity drops to $O(n)$ in the binary case. For k -ary operators, $k \leq n$, the OneMax-complexity further decreases to $O(n/\log k)$.

- [94] Kötzing, T., Neumann, F., Sudholt, D., Wagner, M., [Simple max-min ant systems and the optimization of linear pseudo-boolean functions](#). In: *Foundations of Genetic Algorithms (FOGA)*, pp. 209–218, 2011.

With this paper, we contribute to the understanding of ant colony optimization (ACO) algorithms by formally analyzing their runtime behavior. We study simple MAX-MIN ant systems on the class of linear pseudo-Boolean functions defined on binary strings of length n . Our investigations point out how the progress according to function values is stored in the pheromones. We provide a general upper bound of $O((n^3 \log n) \rho)$ on the running time for two ACO variants on all linear functions, where ρ determines the pheromone update strength. Furthermore, we show improved bounds for two well-known linear pseudo-Boolean functions called ONEMAX and BINVAL and give additional insights using an experimental study.

- [95] Doerr, B., Kötzing, T., Winzen, C., [Too fast unbiased black-box algorithms](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 2043–2050, 2011.

Unbiased black-box complexity was recently introduced as a refined complexity model for randomized search heuristics (Lehre and Witt, GECCO 2010). For several problems, this notion avoids the unrealistically low complexity results given by the classical model of Droste, Jansen, and Wegener (Theor. Comput. Sci. 2006). In this work, we show that for two natural problems the unbiased black-box complexity remains artificially small. For the classical JumpK test function class and for a subclass of the well-known Partition problem, we give mutation-only unbiased black-box algorithms having complexity $O(n \log n)$. Since the first problem usually needs $\Theta(n^k)$ function evaluations to be optimized by standard heuristics and the second is even NP-complete, these black-box complexities seem not to indicate the true difficulty of the two problems for randomized search heuristics.

- [96] Kötzing, T., Sudholt, D., Theile, M., [How crossover helps in pseudo-boolean optimization](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 989–996, 2011.

Understanding the impact of crossover on performance is a major problem in the theory of genetic algorithms (GAs). We present new insight on working principles of crossover by analyzing the performance of crossover-based GAs on the simple functions OneMax and Jump. First, we assess the potential speedup by crossover when combined with a fitness-invariant bit shuffling operator that simulates a lineage of independent evolution on a function of unitation. Theoretical and empirical results show drastic speedups for both functions. Second, we consider a simple GA without shuffling and investigate the interplay of mutation and crossover on Jump. If the crossover probability is small, subsequent mutations create sufficient diversity, even for very small populations. Contrarily, with high crossover probabilities crossover tends to lose diversity more quickly than mutation can create it. This has a drastic impact on the performance on Jump. We complement our theoretical findings by Monte Carlo simulations on the population diversity.

- [97] Doerr, B., Lengler, J., Kötzing, T., Winzen, C., [Black-box complexities of combinatorial problems](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 981–988, 2011.

Black-box complexity is a complexity theoretic measure for how difficult a problem is to be optimized by a general purpose optimization algorithm. It is thus one of the few means trying to understand which problems are tractable for genetic algorithms and other randomized search heuristics. Most previous work on black-box complexity is on artificial test functions. In this paper, we move a step forward and give a detailed analysis for the two combinatorial problems minimum spanning tree and single-source shortest paths. Besides giving interesting bounds for their black-box complexities, our work reveals that the choice of how to model the optimization problem is non-trivial here. This in particular comes true where the search space does not consist of bit strings and where a reasonable definition of unbiasedness has to be agreed on.

- [98] Kötzing, T., Neumann, F., Spöhel, R., [PAC learning and genetic programming](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 2091–2096, 2011.

Genetic programming (GP) is a very successful type of learning algorithm that is hard to understand from a theoretical point of view. With this paper we contribute to the computational complexity analysis of genetic programming that has been started recently. We analyze GP in the well-known PAC learning framework and point out how it can observe quality changes in the evolution of functions by random sampling. This leads to computational complexity bounds for a linear GP algorithm for perfectly learning any member of a simple class of linear pseudo-Boolean functions. Furthermore, we show that the same algorithm on the functions from the same class finds good approximations of the target function in less time.

- [99] Case, J., Kötzing, T., [Measuring Learning Complexity with Criteria Epitomizers](#). In: *Symposium on Theoretical Aspects of Computer Science (STACS)*, pp. 320–331, 2011.

In prior papers, beginning with the seminal work by Freivalds et al. 1995, the notion of intrinsic complexity is used to analyze the learning complexity of sets of functions in a Gold-style learning setting. Herein are pointed out some weaknesses of this notion. Offered is an alternative based on epitomizing sets of functions - sets, which are learnable under a given learning criterion, but not under other criteria which are not at least as powerful. To capture the idea of epitomizing sets, new reducibility notions are given based on robust learning (closure of learning under certain classes of operators). Various degrees of epitomizing sets are characterized as the sets complete with respect to corresponding reducibility notions! These characterizations also provide an easy method for showing sets to be epitomizers, and they are, then, employed to prove several sets to be epitomizing. Furthermore, a scheme is provided to generate easily very strong epitomizers for a multitude of learning criteria. These strong epitomizers are so-called self-learning sets, previously applied by Case & Koetzing, 2010. These strong epitomizers can be generated and employed in a myriad of settings to witness the strict separation in learning power between the criteria so epitomized and other not as powerful criteria!

- [100] Case, J., Kötzing, T., [Solutions to Open Questions for Non-U-Shaped Learning with Memory Limitations](#). In: *Algorithmic Learning Theory (ALT)*, pp. 285–299, 2010.

A U-shape occurs when a learner first learns, then unlearns, and, finally, relearns, some target concept. Within the framework of Inductive Inference, previous results have shown, for example, that U-shapes are unnecessary for explanatory learning, but are necessary for behaviorally correct learning. This paper solves the following two problems regarding non-U-shaped learning posed in the prior literature. First, it is shown that there are classes learnable with three memory states that are not learnable non-U-shapedly with any finite number of memory states. This result is surprising, as for learning with one or two memory states, U-shapes are known to be unnecessary. Secondly, it is shown that there is a class learnable memorylessly with a single feedback query such that this class is not learnable non-U-shapedly memorylessly with any finite number of feedback queries. This result is complemented by the result that any class of infinite languages learnable memorylessly with finitely many feedback queries is so learnable without U-shapes - in fact, all classes of infinite languages learnable with complete memory are learnable memorylessly with finitely many feedback queries and without U-shapes. On the other hand, we show that there is a class of infinite languages learnable memorylessly with a single feedback query, which is not learnable without U-shapes by any particular bounded number of feedback queries.

- [101] Case, J., Kötzing, T., [Strongly Non-U-Shaped Learning Results by General Techniques](#). In: *Conference On Learning Theory (COLT)*, pp. 181–193, 2010.
- In learning, a semantic or behavioral U-shape occurs when a learner first learns, then unlearns, and, finally, relearns, some target concept (on the way to success). Within the framework of Inductive Inference, previous results have shown, for example, that such U-shapes are unnecessary for explanatory learning, but are necessary for behaviorally correct and non-trivial vacillatory learning. Herein we focus more on syntactic U-shapes. This paper introduces two general techniques and applies them especially to syntactic U-shapes in learning: one technique to show when they are necessary and one to show when they are unnecessary. The technique for the former is very general and applicable to a much wider range of learning criteria. It employs so-called self-learning classes of languages which are shown to characterize completely one criterion learning more than another. We apply these techniques to show that, for set-driven and partially set-driven learning, any kind of U-shapes are unnecessary. Furthermore, we show that U-shapes are not unnecessary in a strong way for iterative learning, contrasting an earlier result by Case and Moelius that semantic U-shapes are unnecessary for iterative learning.
- [102] Kötzing, T., Lehre, P. K., Neumann, F., Oliveto, P. S., [Ant colony optimization and the minimum cut problem](#). In: *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 1393–1400, 2010.
- Ant Colony Optimization (ACO) is a powerful metaheuristic for solving combinatorial optimization problems. With this paper we contribute to the theoretical understanding of this kind of algorithm by investigating the classical minimum cut problem. An ACO algorithm similar to the one that was proved successful for the minimum spanning tree problem is studied. Using rigorous runtime analyses we show how the ACO algorithm behaves similarly to Karger and Stein’s algorithm for the minimum cut problem as long as the use of pheromone values is limited. Hence optimal solutions are obtained in expected polynomial time. On the other hand, we show that high use of pheromones has a negative effect, and the ACO algorithm may get trapped in local optima resulting in an exponential runtime to obtain an optimal solution. This result indicates that ACO algorithms may be inappropriate for finding minimum cuts.
- [103] Kötzing, T., Neumann, F., Röglin, H., Witt, C., [Theoretical Properties of Two ACO Approaches for the Traveling Salesman Problem](#). In: *International Conference on Swarm Intelligence (ANTS)*, pp. 324–335, 2010.
- Ant colony optimization (ACO) has been widely used for different combinatorial optimization problems. In this paper, we investigate ACO algorithms with respect to their runtime behavior for the traveling salesperson (TSP) problem. We present a new construction graph and show that it has a stronger local property than the given input graph which is often used for constructing solutions. Later on, we investigate ACO algorithms for both construction graphs on random instances and show that they achieve a good approximation in expected polynomial time.
- [104] Kasprzik, A., Kötzing, T., [String Extension Learning Using Lattices](#). In: *Language and Automata Theory and Applications (LATA)*, pp. 380–391, 2010.
- The class of regular languages is not identifiable from positive data in Gold’s language learning model. Many attempts have been made to define interesting classes that are learnable in this model, preferably with the associated learner having certain advantageous properties. Heinz ’09 presents a set of language classes called String Extension (Learning) Classes, and shows it to have several desirable properties. In the present paper, we extend the notion of String Extension Classes by basing it on lattices and formally establish further useful properties resulting from this extension. Using lattices enables us to cover a larger range of language classes including the pattern languages, as well as to give various ways of characterizing String Extension Classes and its learners. We believe this paper to show that String Extension Classes are learnable in a very natural way, and thus worthy of further study.
- [105] Doerr, B., Johannsen, D., Kötzing, T., Neumann, F., Theile, M., [More Effective Crossover Operators for the All-Pairs Shortest Path Problem](#). In: *Parallel Problem Solving from Nature (PPSN)*, pp. 184–193, 2010.
- The all-pairs shortest path problem is the first non-artificial problem for which it was shown that adding crossover can significantly speed up a mutation-only evolutionary algorithm. Recently, the analysis of this algorithm was refined and it was shown to have an expected optimization time (w.r.t. the number of fitness evaluations) of $\Theta(n^{3.25}(\log n)^{0.25})$. In contrast to this simple algorithm, evolutionary algorithms used in practice usually employ refined recombination strategies in order to avoid the creation of infeasible offspring. We study extensions of the basic algorithm by two such concepts which are central in recombination, namely repair mechanisms and parent selection. We show that repairing infeasible offspring leads to an improved expected optimization time of $O(n^{3.2}(\log n)^{0.2})$. As a second part of our study we prove that choosing parents that guarantee feasible offspring results in an even better optimization time of $O(n^3 \log n)$. Both results show that already simple adjustments of the recombination operator can asymptotically improve the runtime of evolutionary algorithms.
- [106] Case, J., Kötzing, T., [Difficulties in Forcing Fairness of Polynomial Time Inductive Inference](#). In: *Algorithmic Learning Theory (ALT)*, pp. 263–277, 2009.
- There are difficulties obtaining fair feasibility from polynomial time updated language learning in the limit from positive data. Pitt 1989 noted that unfair delaying tricks can achieve polynomial time updates but with no feasibility constraint on the whole learning process. In this context Yoshinaka 2009 makes a useful list of properties or restrictions towards true feasibility. He also provides interesting examples of fair polynomial time algorithms featuring particular uniformly polynomial time decidable hypothesis spaces, and each of his algorithms satisfies several of his properties. Yoshinaka claims that the combination of the three restrictions on polynomial time learners of consistency (which we call herein postdictive completeness), conservativeness and prudence is restrictive enough to stop Pitt’s delaying tricks from working. The present paper refutes the claim of the previous paragraph in three settings. In the setting of uniformly polynomial time decidable hypothesis spaces with a few effective closure properties, the three restrictions allow maximal unfairness. The other two settings involve certain other uniformly decidable hypothesis spaces and general language learning hypothesis spaces. In each of these settings, the three restrictions forbid some, but not all Pitt-style delaying tricks. Inside the proofs of each of our theorems asserting that the three restrictions do not forbid some or all delaying tricks, the witnessing learners can be seen to explicitly employ delaying tricks.
- [107] Case, J., Kötzing, T., [Dynamic Modeling in Inductive Inference](#). In: *Algorithmic Learning Theory (ALT)*, pp. 404–418, 2008.

Introduced is a new inductive inference paradigm, Dynamic Modeling. Within this learning paradigm, for example, function h learns function g iff, in the i -th iteration, h and g both produce output, h gets the sequence of all outputs from g in prior iterations as input, g gets all the outputs from h in prior iterations as input, and, from some iteration on, the sequence of h 's outputs will be programs for the output sequence of g . Dynamic Modeling provides an idealization of, for example, a social interaction in which h seeks to discover program models of g 's behavior it sees in interacting with g , and h openly discloses to g its sequence of candidate program models to see what g says back. Sample results: every g can be so learned by some h ; there are g that can only be learned by an h if g can also learn that h back; there are extremely secretive h which cannot be learned back by any g they learn, but which, nonetheless, succeed in learning infinitely many g ; quadratic-time learnability is strictly more powerful than linear-time learnability. This latter result, as well as others, follow immediately from general correspondence theorems obtained from a unified approach to the paradigms within inductive inference. Many proofs, some sophisticated, employ machine self-reference, a.k.a., recursion theorems.

- [108] Case, J., Kötzing, T., [Dynamically Delayed Postdictive Completeness and Consistency in Learning](#). In: *Algorithmic Learning Theory (ALT)*, pp. 389–403, 2008.

In computational function learning in the limit, an algorithmic learner tries to find a program for a computable function g given successively more values of g , each time outputting a conjectured program for g . A learner is called postdictively complete iff all available data is correctly postdicted by each conjecture. Akama and Zeugmann presented, for each choice of natural number δ , a relaxation to postdictive completeness: each conjecture is required to postdict only all except the last δ seen data points. This paper extends this notion of delayed postdictive completeness from constant delays to dynamically computed delays. On the one hand, the delays can be different for different data points. On the other hand, delays no longer need to be by a fixed finite number, but any type of computable countdown is allowed, including, for example, countdown in a system of ordinal notations and in other graphs disallowing computable infinitely descending counts. We extend many of the theorems of Akama and Zeugmann and provide some feasible learnability results. Regarding fairness in feasible learning, one needs to limit use of tricks that postpone output hypotheses until there is enough time to “think” about them. We see, for polytime learning, postdictive completeness (and delayed variants): 1. allows some but not all postponement tricks, and 2. there is a surprisingly tight boundary, for polytime learning, between what postponement is allowed and what is not. For example: 1. the set of polytime computable functions is polytime postdictively completely learnable employing some postponement, but 2. the set of exptime computable functions, while polytime learnable with a little more postponement, is not polytime postdictively completely learnable! We have that, for w a notation for *mega*, the set of exptime functions is polytime learnable with w -delayed postdictive completeness. Also provided are generalizations to further, small constructive limit ordinals.

- [109] Case, J., Kötzing, T., Paddock, T., [Feasible Iteration of Feasible Learning Functionals](#). In: *Algorithmic Learning Theory (ALT)*, pp. 34–48, 2007.

For learning functions in the limit, an algorithmic learner obtains successively more data about a function and calculates trials each resulting in the output of a corresponding program, where, hopefully, these programs eventually converge to a correct program for the function. The authors desired to provide a feasible version of this learning in the limit — a version where each trial was conducted feasibly and there was some feasible limit on the number of trials allowed. Employed were basic feasible functionals which query an input function as to its values and which provide each trial. An additional tally argument $0 \leq i$ was provided to the functionals for their execution of the i -th trial. In this way more time resource was available for each successive trial. The mechanism employed to feasibly limit the number of trials was to feasibly count them down from some feasible notation for a constructive ordinal. Since all processes were feasible, their termination was feasibly detectable, and, so, it was possible to wait for the trials to terminate and suppress all the output programs but the last. Hence, although there is still an iteration of trials, the learning was a special case of what has long been known as total Fin-learning, i.e., learning in the limit, where, on each function, the learner always outputs exactly one conjectured program. Our general main results provide for strict learning hierarchies where the trial count down involves all and only notations for infinite limit ordinals. For our hierarchies featuring finitely many limit ordinal jumps, we have upper and lower total run time bounds of our feasible Fin-learners in terms of finite stacks of exponentials. We provide, though, an example of how to regain feasibility by a suitable parameterized complexity analysis.