Provide an in-depth mathematical analysis of an evolutionary algorithm or something related to it.

This task, i want to from the probabilistic view to analysis the (1+1)-Evolutionary algorithm.

The last two decades or so have seen an explosion of application areas of evolutionary algorithms (EAs) in diverse scientific or engineering disciplines. An EA is a random search heuristic, using evolutionary mechanisms such as crossover and mutation, for finding a solution that often aims at optimizing an objective function. EAs proved to be extremely useful for combinatorial optimization problems because they can find good solutions for complicated problems using only basic mathematical modeling and simple operators with reasonable efficiency.
Although EAs have been widely applied in solving practical problems, the analysis of their performance and efficiency, which often provides better modeling prediction for potential uses in practice, is much less developed, and only computer simulation results are available for most of the EAs in use.

We are concerned in this article with a precise probabilistic analysis of a simple algorithm called (1+1)-EA.

A typical EA comprises several ingredients:
the coding of solution, the population of individuals, the selection for reproduction, the operations for generating new individuals, and the fitness function to evaluate the new individual.
The mathematical analysis of the time complexity is often challenging mostly because the stochastic dynamics is difficult to capture. It proves more insightful to look instead at simplified versions of the algorithm, seeking for a compromise between mathematical tractability and general predictability.
Such a consideration was first attempted in Bäck (1992) and Mühlenbein (1992) in the early 1990s for the (1+1)-EA, using only one individual with a single mutation operator at each stage.
An outline of the procedure is as follows.

1. Choose an initial string uniformly at random
2. Repeat until a terminating condition is reached
    - Create y by flipping each bit of x (with probability $p% of flipping), each bit independently of the others
    - Replace x by y iff $f(y)f(x)$

Step 1 is often realized by tossing a fair coin for each of the n bits, one independently of the others, and the terminating condition is either exhausting the number of assigned iterations or reaching a state when no further improvement has been observed for a given amount of time.

Mühlenbein (1992) considered in detail the complexity of (1+1)-EA under the fitness function $OneMax$, which counts the number of 1s, namely, $f(x) = \sum_{1jn} x_j$.
More precisely, let X_n denote the time needed to reach the optimum value (often referred to as the optimization time of $OneMax$).
Then the expected time $\mathrm{E}(X_n)$ was argued to be of order $n \log n$, indicating the efficiency of the (1+1)-EA.
Bäck (1992) derived expressions for the success, failure, and stagnation probabilities of mutation. A finer asymptotic approximation of the form

$$\mathrm{E}(X_n) = en \log n + c_1 n + o(n) \tag{1}$$

was derived by Garnier et al. (1999), where $c_1 \approx -1.9$ when the mutation rate p equals $\frac{1}{n}$.
They went further by characterizing the limiting distribution of $\frac{X_n - en \log_n}{en}$ in terms of a log-exponential distribution (which is indeed a double exponential or a Gumbel distribution).
However, some of their proofs, notably the error analysis, seem incomplete (as indicated in their paper). Thus a precise result such as (1) has remained obscure in the EA literature.

While the (1+1)-EA under simple fitness functions may seem too simplified to be of much practical value, the study of its complexity continues to attract the attention in the literature for several reasons.

- First, the (1+1)-EA (under some fitness functions) represents one of the simplest models whose behavior is mathematically tractable.
- Second, the stochastic behaviors under such a simple formulation often have, although hard to prove, a wider range of applicability or predictability, either for more general models or for other meta-heuristics.
  Such a complexity robustness can on the other hand be checked by simulations, and in such a case, the theoretical results are useful in directing good guesses.
- Third, although tractable, most of the analyses of the (1+1)-EAs are far from being trivial, and different mathematical tools have been introduced or developed for such a purpose, leading to more general methodological developments, which may also be useful for the analysis of other EAs or heuristics.
- Fourth, from a mathematical point of view, few models can be solved precisely, and those that can be often exhibit additional structural properties that are fundamental and may be of interest for further investigation.
- Finally, understanding the expected complexity of algorithms may help in identifying hard inputs and in improving the efficiency of the algorithm; see, for example, Doerr and Doerr (2014) and

Doerr et al. (2013).

In this task, we focus on the mutation rate $p = \frac{1}{n}$ and prove that the expected number of steps taken by the (1+1)-EA to reach the optimum of $OneMax$ function satisfies.

$$\mathrm{E}(X_n) = en \log n + c_1 n + \frac{1}{2} e \log n + c_2 + O(n^{-1} \log n)$$

where c_1 and c_2 are explicitly computable constants.

$$c_1 = -e(\log 2 - \gamma - \phi_1(\frac{1}{2}))$$
$$\phi_1(z) := \int_0^z (\frac{1}{S_1(i)} - \frac{1}{t}) dt$$
$$S_1(z) := \sum_{\epsilon 1} \frac{z^l}{l!} \sum (l-j) \frac{(1-z)^j}{j!}$$

From the expression of c_1, it is clear that its characterization lies much deeper than the dominant term $en \log n$.
Numerically, such a characterization is also important because $\log n$ is close to being a constant for moderate values of n, so that the overshoot (c_1 being negative) from the leading term $en \log n$ is not small in such a situation.

Briefly, due to the recursive nature of the algorithm, we first derive the corresponding recurrence relation satisfied by the random variables that capture the remaining optimization time from different states of the algorithm.
In the case when the recurrence can be solved by techniques from analytic combinatorics through the use of generating functions, the corresponding asymptotic approximations can often be obtained by suitable complex-analytic tools such as singularity analysis and saddle-point method.
The analysis of (1+1)-EA under LeadingOnes belongs to such a case.
However, when such a generating function-based approach fails to provide more manageable forms (in terms of functional or differential equations), a different route through “matched asymptotics” may be attempted, which is the one we adopt for the analysis of the (1+1)-EA under OneMax.
Roughly, we identify terms with the largest contribution on the right-hand side and guess the right form (the Ansatz) by matching the asymptotic expansions on both sides of the recurrence.
The Ansatz is, once postulated, often easily checked by direct numerical calculations. The final stage is to justify the Ansatz by a proper error analysis, which often involves a delicate asymptotic analysis.
Our recurrences are, however, of a nonlinear nature, and involve two parameters.
Note that these two approaches are not exclusive, but instead complementary in many cases. For example, we rely on generating functions and complex-analytic tools for the proof of several auxiliary results in this article.

Reference

1. Evolution strategy(wikipad)
2. Cruz P A F, Torres D F M. Evolution strategies in optimization problems[J]. arXiv preprint arXiv:0709.1020, 2007.
3. Raisbeck J C, Allen M, Weissleder R, et al. Evolution Strategies Converges to Finite Differences[J]. arXiv preprint arXiv:2001.01684, 2019.
4. Salimans T, Ho J, Chen X, et al. Evolution strategies as a scalable alternative to reinforcement learning[J]. arXiv preprint arXiv:1703.03864, 2017.
5. K. O. Stanley and R. Miikkulainen, "Evolving Neural Networks through Augmenting Topologies," in Evolutionary Computation, vol. 10, no. 2, pp. 99-127, June 2002, doi: 10.1162/106365602320169811.
6. In William B. Langdon and Erick Cantu-Paz and Keith E. Mathias and Rajkumar Roy and David Davis and Riccardo Poli and Karthik Balakrishnan and Vasant Honavar and G{"u}nter Rudolph and Joachim Wegener and Larry Bull and Mitchell A. Potter and Alan C. Schultz, editors, Proceedings of the Genetic and Evolutionary Computation Conference, 1757-1762, Piscataway, NJ, 2002. San Francisco, CA: Morgan Kaufmann.
7. Mathematics Shows How to Ensure Evolution
8. Hsien-Kuei Hwang, Alois Panholzer, Nicolas Rolin, Tsung-Hsi Tsai, Wei-Mei Chen; Probabilistic Analysis of the (1+1)-Evolutionary Algorithm. Evol Comput 2018; 26 (2): 299–345. doi: https://doi.org/10.1162/evco_a_00212