EXERCISE 3 ANALYSIS

In this exercise, a Genetic Algorithm (GA) was implemented using uniform crossover, bit-flip mutation, and a parent population of at least ten individuals. The GA was tested on the same Pseudo-Boolean benchmark problems (F1, F2, F3, F18, F23, F24, F25) under the same evaluation budget as Exercise 2, allowing direct comparison with the baseline algorithms: Random Search, Randomised Local Search (RLS), and the (1+1) Evolutionary Algorithm (EA).

Performance was evaluated using expected target value plots for each benchmark, with particular attention given to the mean trajectory and variability across runs. The aim is to assess the GA's relative strengths, highlight where its population-based search and recombination provide a clear advantage, and identify cases where simpler evolutionary methods achieve comparable or superior results.

F1: OneMax

The objective is to maximise the number of 1s in a 100-bit string, with the global optimum achieved when all bits are set to 1. Since the fitness landscape is smooth and non-deceptive, algorithms that exploit incremental improvements, such as RLS and the (1+1) EA, perform very effectively. Random Search shows some initial gains by chance but quickly stagnates around values in the low 70s. Its trajectory varies significantly between runs, reflecting its reliance on randomness and lack of memory. In contrast, RLS progresses steadily and reaches the optimum the fastest, as every single-bit flip that increases the number of 1s directly improves fitness. The (1+1) EA follows a similar path but lags slightly because its multi-bit flips occasionally slow its progress. Both methods display low variance after the early stage, converging quickly to the global optimum with consistent results.

The Genetic Algorithm behaves differently. In the early stage, progress is slower compared to RLS and (1+1) EA, which is explained by its use of a population of 20 individuals. Evaluations are distributed across multiple candidates, so improvements appear more gradual at the start, and variance is wider since different individuals may advance at different rates. However, after around 200–300 evaluations, the GA shows a clear acceleration. This occurs because uniform crossover starts combining partial solutions from different individuals, allowing the population to accumulate 1s more quickly than mutation alone. The bit-flip mutation with probability 1/n further refines offspring by correcting remaining 0s, while elitism ensures that the best individual is always preserved. This combination of operators steadily reduces variability in the mid-phase and pushes the population closer to the optimum.

In the later phase, the GA consistently converges to the global optimum, but typically after both RLS and (1+1) EA. While its slower delays start convergence, the design choices of population diversity, crossover, mutation, and elitism guarantee robustness and reliable performance. The variance across runs narrows considerably toward the end, showing that once the GA begins to converge, it does so consistently. Overall, while RLS and (1+1) EA are more efficient on this simple, unimodal function, the GA demonstrates how its mechanisms enable strong mid-phase acceleration and eventual success, even if it requires more evaluations.

F2: LeadingOnes

The LeadingOnes problem evaluates solutions based on the length of consecutive 1s starting from the first position. This structure creates a sequential dependency where only flipping the leftmost 0-bit can improve fitness, and any mutation that disrupts an earlier 1 resets progress. As a result, improvements are inherently stepwise and fragile, making the problem more challenging than OneMax despite its unimodal structure.

Random Search performs poorly here, which is expected since the probability of sampling bitstrings with long prefixes of 1s is very low. Its best-so-far values remain limited and plateau well below the optimum, with wide variance across runs. RS essentially never finds long consecutive runs of 1s within the evaluation budget.

RLS and the (1+1) EA both display the characteristic jagged rise typical of LeadingOnes. Their curves begin with a flat segment when the first bit happens to be zero, since no progress is possible until that bit is corrected. Once corrected, improvements appear as discrete steps, where each additional leading 1 is locked in place until the next bit flip occurs. RLS generally shows more efficient progress, reaching the optimum faster, as its strict single-bit mutation avoids disrupting already correct bits. The (1+1) EA occasionally suffers setbacks due to multi-bit flips, which can overwrite progress, although it recovers over time. Variance for both hill-climbers is larger in the early stages, reflecting differing starting conditions, but shrinks as runs align toward longer prefixes of 1s.

The Genetic Algorithm introduces a more complex dynamic. In the early stage, progress is slower than RLS or (1+1) EA because the population of 20 individuals distributes evaluations across multiple candidates, and crossover cannot provide immediate advantages when initial strings have few leading 1s. However, in the mid-phase, the GA begins to accelerate as uniform crossover recombines partial prefixes of 1s from different individuals. This allows the population to accumulate longer leading segments more quickly than mutation alone. The bit-flip mutation with rate 1/n further supports this process by correcting single incorrect bits without overwhelming the leading sequence. Elitism helps preserve strong individuals, ensuring that once long prefixes are discovered, they are not lost to disruptive variation. Variability in GA runs is wider in the early and mid-stages due to differences in population dynamics, but the variance narrows significantly once crossover begins reinforcing shared prefixes across the population.

By the late phase, GA reliably converges to the optimum, but generally later than RLS, which remains the most efficient algorithm on this problem. The GA's trajectory demonstrates its adaptability: although its diversity slows early progress, the combination of crossover, mutation, and elitism enables robust mid-to-late performance. This behaviour highlights that GA is less efficient on purely sequential problems like LeadingOnes, where the value of crossover is limited, but still capable of achieving the optimum consistently within the evaluation budget.

F3: A Linear Function with Harmonic Weights

The F3 function is a weighted linear problem where each bit contributes additively to the solution according to its position. This means that bits with larger indices contribute more to the total fitness, so fixing these high-weight positions early gives much bigger gains compared to lower-weight bits. The global optimum of about 5000 is achieved when all bits are set to 1. Because the landscape is linear and non-deceptive, progress is straightforward: each flipped bit that turns from 0 to 1 increases fitness, but the magnitude of the improvement depends on which bit is flipped.

Random Search struggles in this setup because it samples solutions blindly. While it occasionally flips some high-weight bits to 1, it does not preserve those gains consistently, and with no mechanism for systematic accumulation, its progress is extremely slow. That is why RS curves plateau far below the optimum (around 3500–4000) and also show large variance across runs ; some lucky runs happen to hit higher-weight positions by chance, while others stall with lower totals.

RLS and the (1+1) EA handle the problem much more effectively. Both operate with mutation-driven search, so once a beneficial bit is flipped, it is retained unless directly reversed. For RLS, the strict single-bit mutation ensures that each step only attempts one change, making progress more consistent. This allows RLS to steadily lock in high-weight bits one by one without risking regressions, which explains why its curve rises sharply and reaches the optimum fastest. The (1+1) EA uses probabilistic mutation across all bits, which means that in some iterations multiple flips occur. While this can occasionally give a bigger jump if several useful bits flip at once, it more often leads to wasted evaluations where good bits are undone at the same time as progress is made. This explains why the (1+1) EA follows a nearly identical trajectory to RLS but usually takes more evaluations to reach the plateau. The jaggedness in its curve reflects these occasional regressions and recoveries. Both hill-climbers, however, show very tight variance once they approach the optimum because after enough iterations all high-weight bits are set and locked in place.

The GA displays a different pattern. In the early phase, progress is slower compared to RLS and (1+1) EA because the evaluation budget is spread across 20 individuals. This means improvements are more fragmented one individual might flip a high-weight bit, another might improve a lowweight bit, but the benefits are not combined immediately, which widens variance between runs. The key turning point occurs in the mid-phase (a few hundred to a thousand evaluations), where crossover begins to recombine partial improvements across individuals. For example, if one parent has already set many high-weight bits near the end of the string and another has fixed several mid-weight bits, their offspring can inherit both sets of improvements, creating solutions that are fitter than either parent. This explains the steep acceleration in GA's curve after the initial lag. Mutation complements this by fine-tuning offspring, with a rate of 1/n, it flips about one bit per 100-bit string on average, which is enough to gradually correct the remaining 0s without overwhelming progress. Elitism reinforces the effect: once a strong individual with many high-weight bits is discovered, it is preserved in the population, preventing regression and anchoring future generations.

By the late phase, the GA consistently converges to the global optimum, though generally after RLS and (1+1) EA. The delay comes from its population overhead in the early phase, but its acceleration in the mid-phase allows it to catch up. The standard deviation bands show this clearly: GA begins with higher variance as different individuals explore, but once crossover starts combining strong solutions, variance collapses as the population converges. This behaviour demonstrates both the strength and cost of GA's design choices, diversity slows early progress but provides the raw material for crossover to drive rapid improvement later on.

In summary, RLS is the most efficient on F3 because its single-bit flips directly exploit the linear structure, while (1+1) EA is slightly slower but still reliable. GA eventually matches them in outcome, reaching the optimum within the budget, but does so via a different pathway: slower start, sharp acceleration, then stable convergence. Its design choices of crossover, mutation, and elitism explain exactly why its trajectory looks different, highlighting the trade-off between early efficiency and robustness across runs.

F18: Low Autocorrelation Binary Sequences (LABS)

The LABS problem is far more difficult than the linear and unimodal cases, as it is multimodal and highly deceptive. The objective is to maximise the reciprocal of a sequence's autocorrelation, and because many different local optima exist, algorithms that rely purely on local, single-bit improvements often struggle to escape poor plateaus. This makes LABS a strong benchmark for testing how well algorithms balance exploration with exploitation.

Random Search once again performs weakest, showing some very quick but unsustainable gains in the first handful of evaluations as random samples occasionally hit decent sequences. However, progress stalls almost immediately, and the best-so-far value only creeps upwards very slowly, plateauing at around 2.3. The variance narrows because most runs converge to equally mediocre outcomes, highlighting RS's inability to refine solutions or escape its random ceiling.

RLS improves on RS initially, but its strict single-bit flip strategy makes it extremely vulnerable to the rugged LABS landscape. After some rapid gains between about 10 and 200 evaluations, progress flattens completely. Because escaping deeper local optima requires coordinated multi-bit changes, RLS stagnates at around 3.7, with a wide spread of outcomes depending on which local basin each run happens to get stuck in. The widening variance confirms its inconsistency across runs.

The (1+1) EA performs better, thanks to its mutation rule of flipping each bit with probability 1/n. On average this produces single-bit moves like RLS, but occasionally generates multi-bit flips. These larger jumps provide escape routes from poor local optima, so although progress is slower than the early gains of RLS, the (1+1) EA continues to make incremental improvements over the full budget. Its final best-so-far value of around 4.0 makes it the strongest of the three in Exercise 2, and its variance is more balanced—wider than RS but narrower than RLS—showing a stable trade-off between exploration and exploitation.

The Genetic Algorithm introduces a very different behaviour. In the early phase, progress is again slower than the hill-climbers because evaluations are distributed across 20 individuals, and initial diversity means that improvements are fragmented. However, in the mid-phase the GA shows a striking acceleration. This is explained by uniform crossover recombining promising subsequences from different individuals: if one parent has discovered a low-correlation structure in one region of the bitstring and another has done so elsewhere, their offspring can inherit both advantages simultaneously. This is something mutation-only algorithms cannot achieve. The bit-flip mutation (rate 1/n) then provides fine-grained improvements, polishing the offspring without overwhelming them. Finally, elitism ensures that once especially good individuals are found, they persist in the population and influence future generations.

By the late phase, the GA overtakes both RLS and the (1+1) EA, achieving values above 4.0 and in some runs approaching 4.5. The variance remains moderate: wider than the hill-climbers early on due to diversity, but collapsing later as crossover homogenises the population around strong candidates. This highlights the advantage of GA's population-based design: while it pays the price of slower early gains, its mechanisms allow it to escape the traps that immobilise RLS and eventually surpass the (1+1) EA by leveraging crossover to combine useful building blocks.

In summary, the F18 results emphasise the strength of GA on rugged, multimodal landscapes. Random Search and RLS stagnate early, and even the (1+1) EA is limited to steady but incremental improvement. The GA, by contrast, demonstrates how its combination of diversity, crossover, mutation, and elitism provides the tools needed to escape local optima and achieve the best overall results.

F23: N-Queens Problem

The N-Queens problem presents a highly rugged combinatorial search space where the objective is to place queens such that none can attack each other. For N=100, the problem is extremely constrained, with countless local optima formed by partial arrangements of queens that reduce some conflicts but leave others unresolved. Algorithms that rely on purely local moves often stall, while those that can combine or restructure solutions stand a better chance of progressing toward optimality.

Random Search again begins with the highest variance and slightly better average starting fitness, since broad sampling can occasionally place queens in relatively low-conflict positions by chance. However, its lack of refinement means that after the first 10–20 evaluations, progress slows sharply. The curve stabilises around 650, far from the optimum of 10, and the standard deviation narrows as most runs converge on similarly poor outcomes. This shows RS's inability to exploit the structure of the problem once random improvements are exhausted.

RLS demonstrates its usual strengths in the early phase, where single-bit flips (i.e., moving one queen) are highly effective in reducing multiple conflicts simultaneously on a crowded board. Between 10 and 200 evaluations, RLS improves steadily, often eliminating large groups of conflicts. However, this improvement halts once the board reaches more stable intermediate states. Because RLS only allows single-bit flips, it cannot coordinate the larger rearrangements needed to escape strong local optima. As shown in the inset plot, its runs plateau consistently at values of 7–8. Variance is low, indicating that although RLS reliably finds decent placements, it lacks the flexibility to reach near-optimal solutions.

The (1+1) EA initially mirrors RLS but diverges in the mid-phase because of its mutation operator. While most iterations behave like RLS with single-bit flips, occasional multi-bit flips allow the EA to escape traps where multiple queens need to move simultaneously. This explains its higher variance early on, as some runs find these lucky improvements earlier than others. By 10,000 evaluations, however, almost all runs converge near a best-so-far value of 9, showing that the algorithm consistently approaches near-optimal placements. The fact that it rarely reaches the true optimum of 10 reflects the improbability of generating the exact multi-bit flips required in the limited budget.

The Genetic Algorithm, however, shows a distinct advantage. Its population-based design introduces slower early improvements than RLS and (1+1) EA, as evaluations are distributed across multiple individuals. But once the mid-phase begins, the GA accelerates dramatically. This is due to uniform crossover, which allows the algorithm to combine non-conflicting subsequences from different parents. For instance, one parent may have optimised the placement of queens in the top half of the board, while another excels in the bottom half; crossover enables their offspring to inherit both strengths simultaneously. Bit-flip mutation then provides the fine-grained adjustments needed to polish these merged solutions, while elitism ensures that the best configurations persist across generations and are not lost to random sampling.

As a result, the GA curve climbs steadily past the plateau where RLS stalls, catching up to and eventually overtaking the (1+1) EA. The inset plot highlights that GA does not just approach the optimum but frequently achieves the global optimum of 10, something that the mutation-only methods fail to do. Variance starts wider due to population diversity but collapses later as crossover and elitism concentrate the population around strong solutions, yielding highly consistent final outcomes.

In summary, the N-Queens results highlight the distinct strengths and weaknesses of each approach. Random Search is ineffective beyond quick early gains, RLS stagnates in local optima, and the (1+1) EA pushes further but rarely escapes the final traps. The GA demonstrates the clearest superiority, with its combination of diversity, crossover, and elitism allowing it to integrate partial solutions and consistently reach the true optimum. This reinforces GA's effectiveness on combinatorial problems where local optima dominate the landscape and coordinated improvements are essential.

F24: Concatenated Trap

The Concatenated Trap problem is a deceptive benchmark that deliberately misleads algorithms into local optima. Each sub-block of the bitstring resembles a small trap: greedy improvements appear beneficial in the short term, but they guide the search away from the global optimum. Escaping these traps requires more global coordination of solutions, which makes the problem particularly challenging for local search methods like RLS and the (1+1) EA, and a good test of whether population-based strategies such as the GA can leverage diversity and recombination effectively.

Random Search follows its usual trajectory: quick but shallow gains in the very early evaluations, when random samples occasionally align with trap sub-blocks. Beyond this initial burst, progress becomes irregular and weak. By the end of the budget, RS reaches only a best-so-far value of 12, well below the optimum of 20. The wide variance observed throughout reflects the algorithm's dependence on chance, with some runs stumbling upon moderately good sub-blocks, but most converging on poor, inconsistent results.

RLS performs better in the early phase, as single-bit flips reliably improve solutions within individual trap blocks. Between 10 and 200 evaluations, this leads to a steep rise as the algorithm locks in local progress. However, this progress is deceptive rather than globally optimal. Once local traps are stabilised, RLS stalls completely, plateauing around 16. The narrow spread of runs at this plateau confirms that the algorithm consistently finds the same suboptimal solutions but lacks the ability to escape them due to its restrictive mutation operator.

The (1+1) EA shows a very similar trajectory to RLS, with slightly better long-term results. Because it flips each bit independently with probability 1/n, it occasionally produces multi-bit flips that allow the algorithm to escape some traps. This enables the (1+1) EA to progress marginally further, finishing with a mean best-so-far value of 16.1, just above RLS. However, the improvement is modest, highlighting how unlikely it is to generate exactly the right set of coordinated flips to cross deceptive valleys in the landscape. Variance for the (1+1) EA remains stable throughout, as most runs follow nearly identical paths except for rare lucky multi-bit improvements.

The Genetic Algorithm shows a noticeably different pattern. Its early progress is slightly slower than RLS and the (1+1) EA because evaluations are distributed across a larger population. However, the mid-phase acceleration is striking once the population contains individuals that have solved different parts of the trap sub-blocks, uniform crossover allows the GA to recombine these partial solutions into offspring that inherit multiple correct blocks simultaneously. This is something mutation-only methods cannot achieve efficiently. Bit-flip mutation then fine-tunes these offspring, while elitism preserves the strongest individuals so that progress is never lost.

By 1,000 evaluations, the GA overtakes both RLS and the (1+1) EA, climbing toward the global optimum. The final plateau is very close to the maximum value of 20, indicating that in many runs, the GA successfully escapes the deceptive traps altogether. The variance pattern also reflects this: while initially wide due to population diversity, it narrows sharply once strong sub-block solutions spread through the population via crossover. This convergence shows that GA’s population-based exploration not only avoids stagnation but also provides robustness against the misleading structure of the problem.

In summary, F24 demonstrates the limits of local mutation-based methods on deceptive landscapes. Random Search is weak and inconsistent, RLS consistently falls into traps, and the (1+1) EA improves only slightly by chance. GA, however, leverages recombination and elitism to integrate partial progress across individuals, systematically overcoming the deceptive local optima and approaching the global solution. This highlights why GAs are particularly well-suited to deceptive problems where problem structure must be pieced together from multiple sources of information.

F25: NK Landscapes (NKL)

The NK Landscapes problem is among the hardest benchmarks because of its rugged, epistatic fitness surface. Each variable's contribution to fitness depends not only on itself but also on neighbouring variables, which produces a landscape full of deceptive local optima. For N=100, even strong algorithms struggle, as escaping one basin often requires coordinated flips across multiple variables. Unlike linear or smooth problems, there is no consistent gradient to exploit, making this a critical test of how well algorithms handle ruggedness.

Random Search performs as expected: it improves quickly in the very early evaluations, as blind sampling is likely to find some configurations better than the initial random solution. However, without memory or refinement, RS's growth slows dramatically once the initial gains are exhausted. Its final best-so-far value remains near -0.40, the worst of all methods, although its curve continues to rise slowly throughout the evaluation budget. This ongoing but shallow climb suggests that with infinite time, RS could improve further, but within a fixed budget it remains both inefficient and inconsistent.

RLS fares even worse in the long run. Its strict single-bit mutation operator makes it highly dependent on the initial configuration. Early on, some runs perform relatively well, but many stagnate immediately if the start is far from promising basins. The wide variance at the beginning reflects this sensitivity. After around 200 evaluations, the algorithm plateaus completely, trapped in poor local optima. This behaviour is consistent with the structure of NKL: since interactions between variables require multiple bits to be flipped simultaneously to escape local basins, single-bit mutations cannot generate meaningful progress. By the end, RLS stabilises near -0.38, and its narrow variance indicates that virtually all runs end in the same suboptimal basins.

The (1+1) EA shows stronger performance, with its trajectory diverging positively from RLS after around 500 evaluations. Like RLS, it mainly explores through single-bit flips, but its mutation operator occasionally produces multi-bit flips. These rare events allow the (1+1) EA to escape some of the traps that immobilise RLS, leading to incremental improvements across the entire budget. Its final mean best-so-far value reaches -0.31, the best of the Exercise 2 algorithms. The variance narrows as runs converge on similar suboptimal peaks, showing stability but also the difficulty of making further progress in such a rugged landscape.

The Genetic Algorithm demonstrates a distinct advantage by leveraging its population-based search. Initially, its improvements are slower than the hill-climbers, as evaluations are spread across multiple individuals. However, by the mid-phase, the GA's population begins to diversify across different basins, exploring multiple local regions in parallel. Uniform crossover then plays a critical role: it recombines promising building blocks from different individuals, creating offspring that inherit well-adapted substructures even if the full solution is not optimal. This ability to shuffle partial solutions allows the GA to escape deceptive local optima more effectively than mutation-only methods. Bit-flip mutation (1/n rate) provides the fine-grained exploration needed to refine individuals within basins, while elitism preserves the best solutions to prevent regression.

By the end of the budget, the GA outperforms RLS and nearly matches the (1+1) EA, finishing with values close to -0.31. Importantly, however, its curve shows a more consistent upward trend in the mid- and late phases, indicating that it continues to make progress even after RLS has stalled and (1+1) EA improvements slow. Variance starts wide due to population diversity but narrows later as strong individuals spread through the population, leading to consistent convergence.

In summary, the NK Landscapes results underline the extreme difficulty of rugged, epistatic problems. RS makes shallow stochastic gains, RLS stalls completely, and the (1+1) EA pushes further via occasional multi-bit flips. The GA, however, demonstrates the clearest advantage in the long run: by maintaining a diverse population, leveraging crossover to combine building blocks, and using elitism to preserve progress, it sustains steady improvements where the other methods stagnate. Although all algorithms remain in the negative range after 100,000 evaluations, GA's robustness suggests that it would outperform the hill-climbers even more clearly with extended budgets, making it the most promising approach for navigating deceptive landscapes.