F23: N-Queens Problem

This analysis examines the fixed-budget plots *'F23_100D_Final'* and *'F23_100D_Inset'* for the N-Queens problem, with the optimum set at 10. The N-Queens problem is a combinatorial optimisation task with a rugged search landscape, making it a useful benchmark for comparing Random Search (RS), Randomised Local Search (RLS), and the (1+1) Evolutionary Algorithm (EA).

The plot shows that RS has a higher starting mean and spread, indicating that, on average, RS begins with a better value than the other algorithms. This reflects that random sampling can occasionally stumble upon moderately good placements by chance. However, all algorithms start far from the optimum, with values below -1200. This indicates that a randomly filled board typically contains many conflicts. RS improves sharply in the first ~10 evaluations, as even random moves are very likely to reduce the large number of initial conflicts. However, its progress slows quickly, and improvements soon become irregular due to the fact that RS cannot use information from previous solutions. Therefore, its ability to refine placements on the board is limited. After ~200 evaluations, the standard deviation of the RS mean curve narrows considerably, suggesting that most runs converge to similarly poor results. By the end of the 100,000 evaluations, RS finishes with a *'best-so-far'* value of approximately -650, which is far from the optimum of 10, and clearly inferior to the other approaches in this problem space.

Both the RLS and the (1+1) EA start at similar initial values and are much lower than RS because they begin from a single random configuration rather than sampling broadly. For the first 10 evaluations, their performance is almost identical, that is, until RLS gains momentum, improving steadily up to the 200th evaluation. This rapid improvement occurs because flipping a single queen's position often eliminates several conflicts at once when the board is heavily constrained. After this point, progress stalls since escaping local optima requires multi-bit changes, which RLS cannot produce. The accompanying inset plot makes this plateau evident, as RLS converges to solutions with *'best-so-far'* values of 7-8, with very low standard deviation. This consistency showcases both the algorithm's strength in finding acceptable local optima and its weakness in failing to move beyond them.

As stated above, the (1+1) EA mirrors RLS's performance early on; however, it soon diverges due to its difference in mutation operator. The (1+1) EA flips each bit with a probability of 1/n, resulting in mostly single-bit flips per iteration, but this can occasionally produce multi-bit flips. As a result, the (1+1) EA follows RLS into the positive values, where this unique characteristic allows it to continue improving beyond the point where RLS stalls. Additionally, the difference between RLS and the (1+1) EA's mutation operator is what causes higher standard deviation in the (1+1) EA during the early and mid-phase, as some runs make lucky multi-bit jumps earlier than others. Eventually, almost all runs converge to the near-optimal *'best-so-far'* value of 9, with standard deviation collapsing after ~10,000 evaluations. Despite the convergence to near-optimal values, the plot suggests that no runs of the (1+1) EA reach the true optimum of 10. This makes intuitive sense since the probability of generating the exact multi-bit flip needed to escape the final local optima is extremely small.

Overall, the N-Queens results captured in this plot highlight that RS makes quick initial gains, but its growth deteriorates early, leading it to fail to find acceptable solutions. In contrast, both RLS and the (1+1) EA improve rapidly and consistently. However, RLS plateaus at suboptimal values, highlighting its inability to overcome local optima, while the (1+1) EA leverages rare multi-bit mutations to escape these local optima and reach near-optimal solutions, thereby outperforming both algorithms.