AA222 Project 2 - Constrained Optimization

Jeff Robinson

jbrobin@stanford.edu

Method Used: Covariance Matrix Adaptation

Since gradient information does not necessarily bear any relevance to the location of the global minimum in a constrained problem, the optimization methods I considered included exclusively the direct and stochastic methods. I first attempted to use Generalized Pattern Search with basis vectors as the spanning set to descend along the quadratic penalty function to find the feasible set. After discovering that the penalty function would likely not be convex, I instead turned to the Cross-Entropy Method since its stochastic approach would be less likely to become stuck in local minima, whether outside or inside the feasible set. When Cross-Entropy failed to converge on the Secret functions, and proved unreliable on the Simple functions, I attempted to use Particle Swarm instead, and later found that it, too, was unreliable. My final algorithm uses Covariance Matrix Adaptation (CMA) exclusively, since its more sophisticated approach enables more stable convergence. Of particular value to constrained optimization is the weighted update strategy empolyed by CMA, which allows it to distinguish between the optimality of points within the elite set when fitting the distribution, in contrast to the indiscriminate update approach used by Cross-Entropy. This prevented premature convergence issues encountered by Cross-Entropy on the same functions.

Implementation

I primarily relied on *Algorithm 8.9* from *Algorithms for Optimization* to implement CMA, since the implementation of the algorithm in the text worked well with few adaptations to account for constraints. My modifications include:

- The maximum number of loop iterations is calculated from the number of design points sampled per iteration
- Quadratic and count penalties have been added, with the penalty method selected to augment the behavior of the algorithm for certain functions and constraints
- Instead of relying on the mean of the distribution to provide the optimum design point, a potentially optimal design point is tracked throughout the optimizaiton process. New potential optima are selected from among the sampled design points at each iteration, including the mean of the distribution. Only feasible design points are considered potential optima.

Convergence Path Plots

These plots show the path of the potential optimum point discovered by the algorithm at each iteration of the optimization process. Start points are highlighted in red, end points are highlighted in green, and points between are colored in order of decreasing brightness as iteration count increases. The feasible set is within the black lines for Simple 1, and to the left of the black lines for Simple 2.

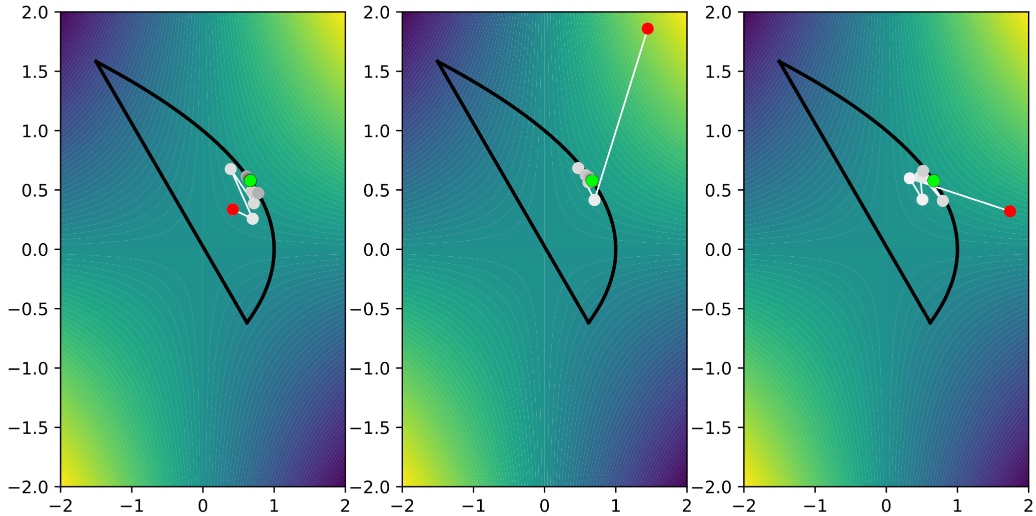

Figure 1: Convergence Paths for Three Random Points on "Simple 1" Function.

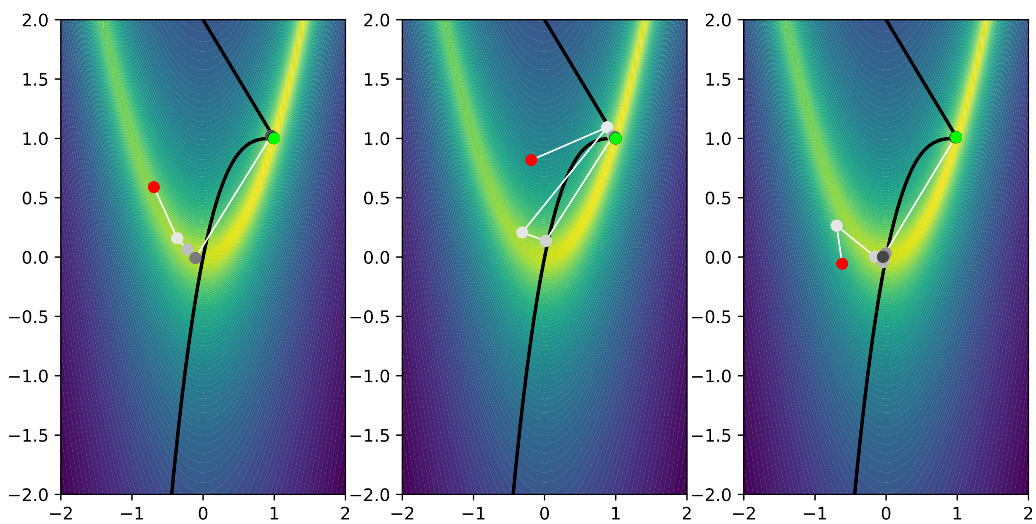

Figure 2: Convergence Paths for Three Random Points on "Simple 2" Function.

Convergence Value Plots

Figure 3 below shows the convergence plots for the six start points shown in the path plots above. The axes to the left represent the Simple 1 function, and the axes to the right represent the Simple 2 function.

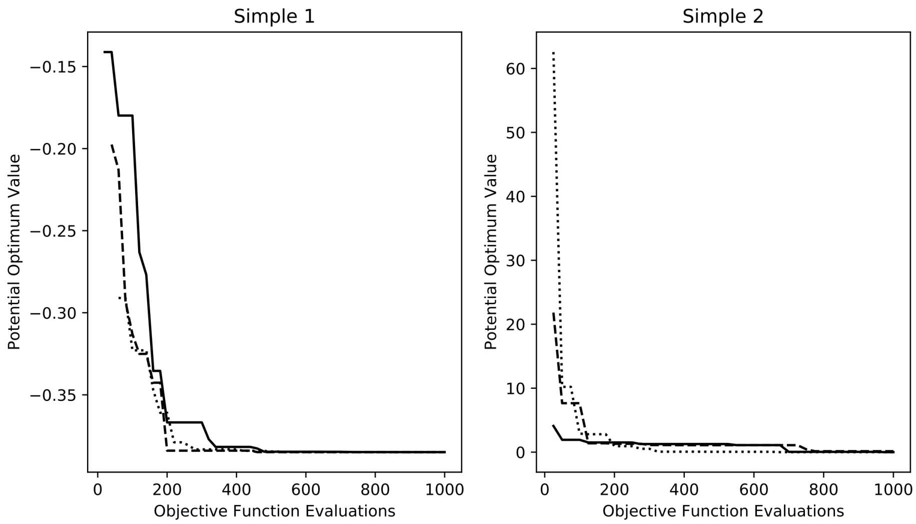

Figure 3: Potential Optimum Function Value vs. Cumulative Number of Function Evaluations.

References

Kochenderfer, M. J., and Wheeler, T. A., *Algorithms for Optimization*, The MIT Press, Cambridge, Massachusetts, 2019.