

Constrained Global Optimization Using Exterior and Interior Methods

Ross B. Alexander

Department of Aeronautics and Astronautics

Stanford University

Stanford, CA 94305

rbalexan@stanford.edu

I. CONSTRAINED OPTIMIZATION

For each of the problems, we use an exterior method coupled with an interior method. Decomposing the problem into an exterior method, which searches the design space for a solution in the feasible space, and an interior method, which searches the feasible space, makes use of the strengths of each method while guaranteeing a feasible solution so long as the exterior method converges.

From our initial design point, we iterate using the exterior method and terminate once a feasible solution is found. The solution from the exterior method is then used as the initial design point for the interior method, which runs until the number of function, gradient, and constraint evaluations reaches the maximum number of allowed evaluations.

A. Exterior Method

For the exterior method, we use the quadratic penalty method with a forward finite-difference penalty gradient approximation on top of gradient descent with Nesterov momentum.

While zero-order methods can be useful in situations where the gradient of the objective function is not available, first-order methods can improve the search performance in situations where this information is available (as is the case for us). Additionally, in constrained optimization, gradient information can be useful in guiding the search toward feasibility. As a result, we use a penalty method with a quadratic penalty, which introduces a component of objective function proportional to the sum of squares of constraint violations, and accordingly, a component of the gradient proportional to the sum of twice the magnitude of the constraint violations. This additional component of the gradient is an improvement over a count penalty, which only introduces a discontinuity at constraint boundaries and does not help guide the search toward feasibility but rather "corral" the solution to remain in the feasible space. As such, a mixed penalty consisting of a linear combination of a count penalty and a quadratic penalty does not confer much of an advantage over a quadratic penalty alone.¹

As mentioned earlier, we use gradient descent with Nesterov momentum to descend the penalized objective function. Gradient descent with Nesterov momentum exploits the benefits of traditional momentum but employs a look-ahead gradient

to reduce momentum near minima, which makes for more efficient gradient descent. Since we are using a first-order method on the penalized objective function to descend into the feasible space, we need to explicitly compute the gradient of the penalty function. For this, we choose a forward finite-difference approximation. While the linear convergence ($\mathcal{O}(h)$) of the forward-difference approximation is clearly inferior to the quadratic convergence ($\mathcal{O}(h^2)$) of the central-difference approximation, we use the forward-difference approximation since after evaluating the penalty at the current design point, we only need to compute its variation at the forward step along each dimension (n function evaluations) versus needing to compute the variations at both the forward and backward step along each dimension ($2n$ function evaluations). While this is a trade-off, we choose the first-order method since the gradient of the penalty function does not need to be known with high precision.

B. Interior Method

For the interior method, we use the inverse-barrier interior point method on top of the Hooke-Jeeves method.

The interior point method utilizes a similar approach to the penalty method by penalizing near-constraint violation using a barrier penalty function. This barrier function adds a penalty that approaches infinity as the design point approaches the constraint. At each iteration, the strength of the barrier function is decreased by a constant factor, which allows the solution to remain in the feasible space while slowly and safely approaching the constraint boundary, if necessary, using the inner descent method. The process of gradually reducing a penalty applied by the barrier function is similar to simulated annealing, which can help the optimization process avoid local minima. There are two popular choices for barrier functions: an inverse barrier function applies a penalty globally (over the entire feasible space) and a logarithmic barrier function applies a penalty locally (only very close to the constraint boundary), however, to utilize the benefits of simulated annealing, we choose the inverse barrier function over the log barrier function due to the globality of the applied penalty.²

We used the Hooke-Jeeves method to descend the penalized objective function in the feasible space. While we have gradient

¹While we settled on the quadratic penalty, the count, quadratic, and mixed penalties are all implemented.

²Similarly to the penalty method, while we settled on the inverse barrier function, the inverse and log barrier functions are all implemented.

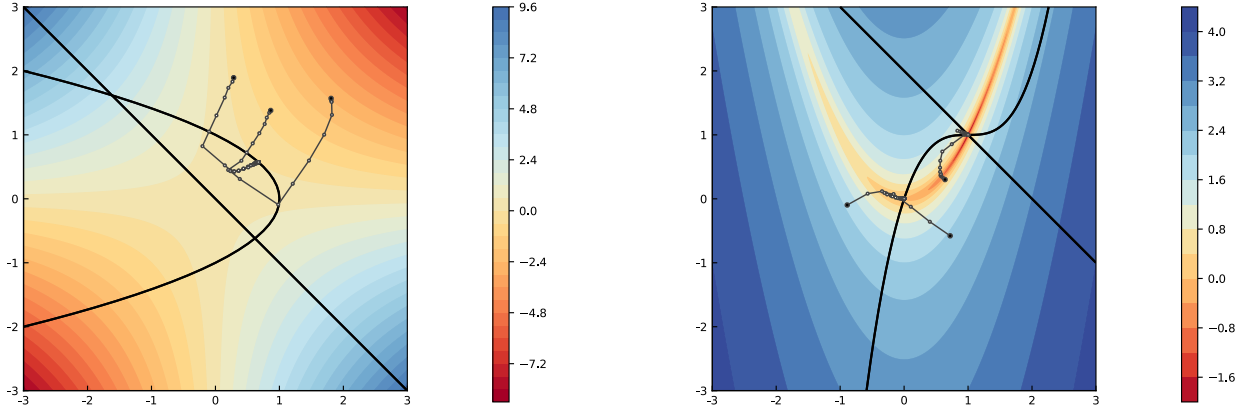


Fig. 1. Three constrained optimization paths on the `simple1` function (*left*) and the `simple2` function (*right*). The feasible space for `simple1` is the semi-elliptic region in the center and the feasible space for `simple2` is the left-half region. For the exterior method, these optimizations use the quadratic penalty method ($\rho = 1, \gamma = 2$) with gradient descent with Nesterov momentum ($\alpha = 1\text{E} - 3, \beta = 0.9$) and for the interior method, these optimizations use the inverse-barrier interior point method ($\rho = 1, \gamma = 2$) on top of the Hooke-Jeeves method ($\alpha_0 = 1\text{E} - 1, \gamma = 0.5, \epsilon = 1\text{E} - 4, k_{max} = 10$). The color gradations for `simple2` are the base-10 logarithm of the function values.

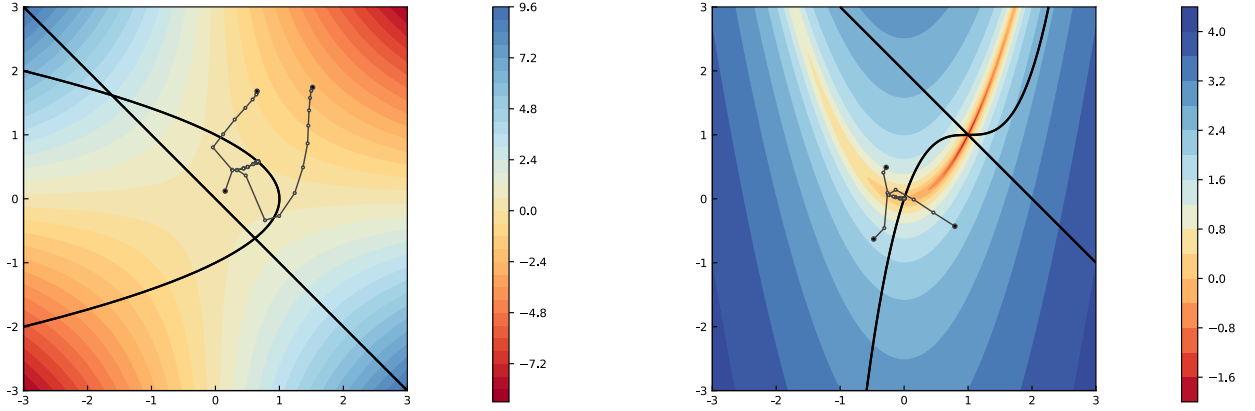


Fig. 2. Three constrained optimization paths on the `simple1` function (*left*) and the `simple2` function (*right*). The feasible space for `simple1` is the semi-elliptic region in the center and the feasible space for `simple2` is the left-half region. For the exterior method, these optimizations use the mixed penalty method ($\rho_1 = 1, \rho_2 = 1, \gamma = 2$) with gradient descent with Nesterov momentum ($\alpha = 1\text{E} - 3, \beta = 0.9$) and for the interior method, these optimizations use the log-barrier interior point method ($\rho = 1, \gamma = 2$) on top of the Hooke-Jeeves method ($\alpha_0 = 1\text{E} - 1, \gamma = 0.5, \epsilon = 1\text{E} - 4, k_{max} = 10$). The color gradations for `simple2` are the base-10 logarithm of the function values.

information, which enables us to use first-order methods, we use a first-order method, namely, the Hooke-Jeeves method, to descend the penalized objective function since it is relatively simple to implement. Unfortunately, the Hooke-Jeeves method is the most expensive among pattern-based zero-order methods in d -dimensional space, requiring $2d$ evaluations per iteration, compared with $d + 1$ evaluations per iteration for methods like generalized pattern search (GPS), the Nelder-Mead simplex method, and mesh-adaptive direct search (MADS). This is something we will have to live with. The Hooke-Jeeves method searches along each positive and negative basis vector of the design space for an improvement and accepts the best one – this behavior can lead the search to get stuck in local minima (as we see in the `simple2` figures).

C. Results & Discussion

In Figs. 1-4, we use two different methods, varying in choice of penalty (quadratic v. mixed) and barrier (inverse v. log).

Deciding on the hyperparameters (see Figs 1-2 for specific values) for each method relied on some typical values provided in the textbook as well as some cyclic search over the hyperparameter space. Using the `localtest.jl`, we iterated over a few values of each hyperparameter and swept the hyperparameter space twice. Several of the hyperparameters were more difficult to tune, including the Nesterov momentum hyperparameters, as these depend on the magnitude of objective function's gradient, which varies from problem to problem and also varies with the strength and choice of penalty function. The typical penalty and barrier hyperparameters worked well, so we did not alter them much. Tuning the Hooke-Jeeves method was much more complicated an involved trade-offs with number of iterations and accuracy since the method itself is run inside of the interior point method loop. Ultimately, we settled on a maximum of 10 inner iterations with an initial step size of 0.1. All of these hyperparameters seemed to work

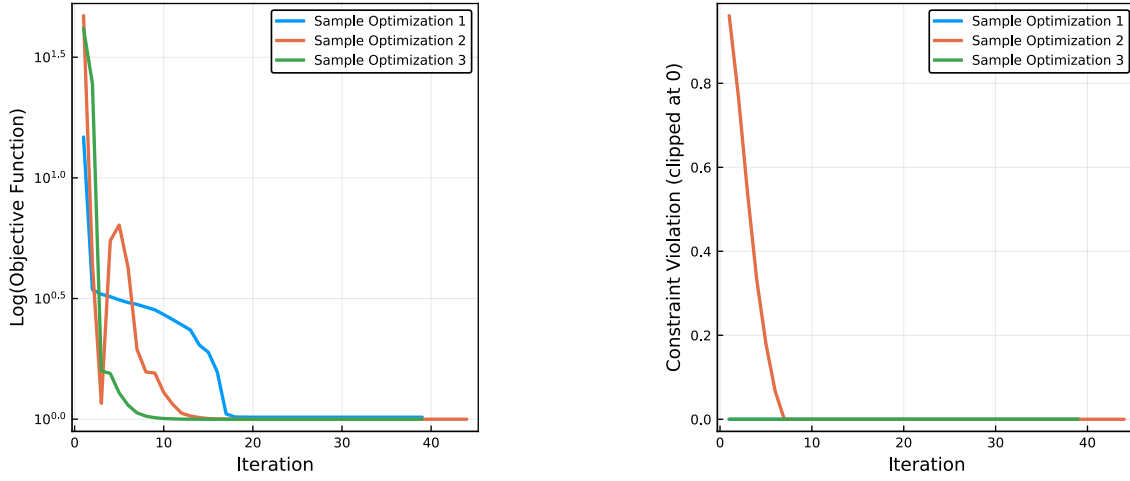


Fig. 3. Convergence of objective function (*left*) and the maximal constraint violation (*right*) for three randomly-initialized constrained optimizations on the `simple2` function. We use the first algorithm (and hyperparameters) described in Fig. 1.

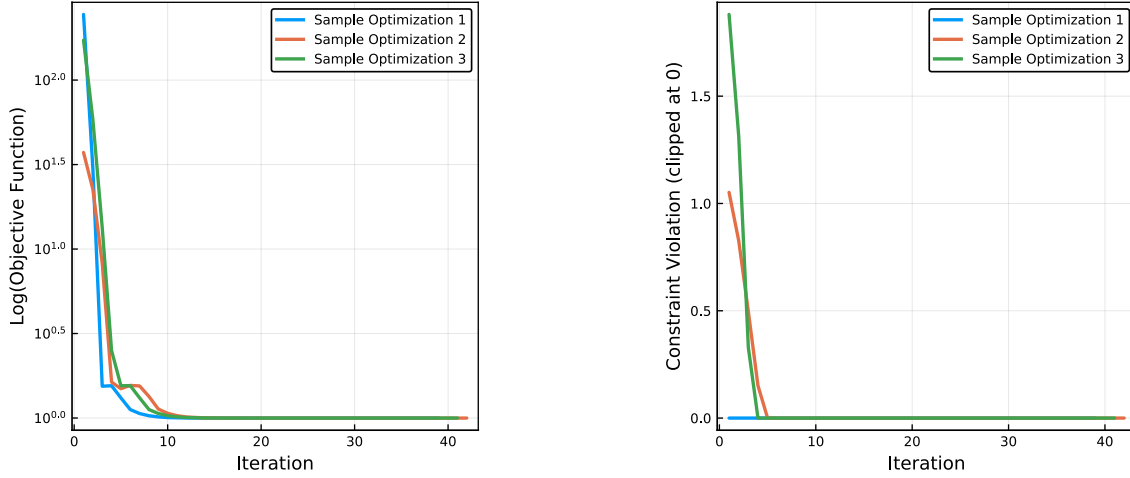


Fig. 4. Convergence of objective function (*left*) and the maximal constraint violation (*right*) for three randomly-initialized constrained optimizations on the `simple2` function. We use the second algorithm (and hyperparameters) described in Fig. 2.

reasonably well for all functions.

Looking at Figures 1 and 2, we see no remarkable difference between the performance on each function using the two different algorithms. Each of the methods can be seen to first obtain a solution in the feasible space and then optimize the solution in the interior of the feasible space. For `simple1` we do achieve the global minimum on nearly every iteration, which is in contrast to `simple2` where we nearly never achieve the global minimum despite always returning a feasible solution. In part this is due to the fact that the combination of exterior and interior methods is not well-suited to the topology of the function. The exterior method forces the optimization to the feasible space, and then the interior method must overcome a massive gradient to reach the global minimum at $(1, 1)$, which makes reaching the global minimum incredibly challenging. As an alternative, a method that performs efficient descent should be able to reach the minimum relatively quickly and afterwards an exterior method could be applied to guide the

solution to feasibility.

In Figures 3 and 4, we observe relatively clean convergence characteristics of the objective function and maximal constraint violation on `simple2` (albeit to a non-global minimum) for both variations of the algorithm. While the mixed penalty with logarithmic barrier appears smoother, it is hard to generalize that either algorithm is better than another without looking at a larger statistical estimate of the performance characteristics. Overall, we see a quick reduction in the maximal constraint violation as the solution is guided to feasibility and then a follow-on reduction in the objective function as the interior point method begins to optimize the solution within the feasible space.

Ultimately, the methods we chose were sufficient to achieve near-optimal feasible solutions across a variety of constrained optimization benchmarks. It would be interesting to implement some first-order descent methods in the interior point method and see what performance gains can be achieved.