Branch and bound approach

For each velocity $k \in \{0...N\}$ on the mesh, we can compute the pair of scalar velocities u_k and v_k as linear functions of the model parameters. We want to ensure that the ratio is between 0 and 1. If we fix the sign of either u_k or v_k, this is a convex constraint.

Let $K \subset \{0...n\} \times \{-1, 1\} = \mathcal{K}$ be a set of indices and signs (not necessarily for all k).

Then we can use a convex optimizer to minimize our objective function, such that our ratio constraint is satisfied for all $(k, s) \in K$, and possibly unsatisfied for other k. Thus, each K represents a convex relaxation of the original problem. Let $M(K)$ be the value of the objective for the solution of the relaxed problem.

In the end, we want to find $M = \min_{K \subset \mathcal{K}} M(K)$.

First, we can note that for any $K_1 \subset K_2$, $M(K_1) \leq M(K_2)$. This can allow us to dismiss entire sets of indices and signs. If we know for instance that $M \leq z$, and we know that $M(K) > z$, then we can dismiss K and all its supersets, because adding constraints will only make the value larger.

This is the basis of the branch and bound algorithm. We start with $K = \emptyset$. We then solve the relaxed problem, and get a value $M(\emptyset)$. We can then check for which indices the ratio constraint is violated. If there is none, we have found a solution. Otherwise, we try to pick on intelligently (maybe for instance starting with indices where both velocities are large and have the same sign?). This gives rise to two new sets $K_1 = \{(k, 1)\}$ and $K_2 = \{(k, -1)\}$. We pick the sub-problem that we think most likely contains the true solution, and solve it, but remember the other one to check later.

We can then repeat the process, until we have found an admissible solution. This process can take up to N iterations, but hopefully it will usually be much faster.

The admissible solution we found is an upper bound on M. We can then go back to the previous steps, and solve the other sub-problems. Hopefully, we can dismiss those based on the upper bound we found already. If not, we can again split those problems into two and continue recursively.

During this process we can maintain a lower and an upper bound on M, so we should at least have some idea of how far we are from the true solution.

Initially, we know that $M \geq M(\emptyset)$. Once we have found an admissible solution, we also have a lower bound on M. And while we check past sub-problems, we can update both bounds.

McCormick envelopes

Instead of just completely ignoring the non-convex constraints, we can also approximate them using McCormick envelopes.

The basic idea of the branch-and-bound algorithm can remain the same if we want to use those.

In order to be able to use McCormick envelopes, we need upper and lower bounds for all v_k and u_k. I'm actually not sure how to get those. We could just use relatively large numbers, but it seems the quality of the envelope will get worse if we use too extreme bounds, so we will have to experiment a bit.

We first introduce new variables for the velocities v_k and u_k, and add an equality constraint to fix them to the linear function of the parameter vectors.

We then use the fact that $0 \leq \frac{v_k}{u_k} \leq 1$ is equivalent to $v_k^2 - v_k u_k \leq 0$.

We now introduce yet another variable z, which we want to be approximately $v_k u_k$.

We can achieve this approximation by adding bounds

$$
\begin{aligned}
z &\geq u_{\text{lower}} v + u v_{\text{lower}} - u_{\text{lower}} b_{\text{lower}} \\
z &\geq u_{\text{upper}} v + u v_{\text{upper}} - u_{\text{upper}} b_{\text{upper}} \\
z &\leq u_{\text{upper}} v + u v_{\text{lower}} - u_{\text{upper}} b_{\text{lower}} \\
z &\leq u_{\text{lower}} v + u v_{\text{upper}} - u_{\text{lower}} b_{\text{upper}}
\end{aligned}
$$