The Conclusion of the Solve-Jobs-Then-Coverage Method

To solve two-stage optimization, I thought of two methods. The first is to take the entire formulation with both job and coverage vehicle constraints, and perform one massive column generation. The second is to solve jobs first, then solve coverage given those job routes. I have just concluded an exploration on the second. We get a result, and a pretty good one, and it was straightforward!

Using column generation, the best coverage scheme takes a distance of 1'931, giving a full answer of 4'776 along with the 2'845 for jobs.

I first used a direct formulation, described in "DF Coverage Formulation after Jobs Solved". Of particular interest was the second of the "Subtle Points", in which I determined that while in isolation, whether a job starts at its earliest convenience might not matter, when the two are combined, the time windows might have an effect on total cost.

I then coded the direct formulation through "take-best-routes-find-fitting-coverage-df", but this was not fruitful, as the machine did not provide an answer. I tinkered with the code for a bit, but could not find a mistake with that, or my formulation. I would've come back to it, if not for the results provided through column generation.

I next used column generation, the formulation for which is provided in "Column Generation for Coverage". This yielded a nice insight in which, despite a constraint's being worded in terms of i (locations for jobs, NOT coverage) and t, I was able to transform a constraint so that it was written in terms of j (locations for coverage).

The column generation code is provided in "cg-on-coverage.ipynb". It's a standard procedure, and not terribly interesting. In fact, the loop didn't run for very long until it was realized that the best answer is just to have a vehicle sit at a spot for as long as necessary, then go back. It's actually worse to move between coverage nodes, because such an open gap usually doesn't exist.

To see if I could make column generation more appetizing I reduced the number of coverage spots to only those that were strictly necessary. The results, in "cg-on-coverage-leave-suboptimal", was also quick. No negative reduced costs—just stay there and sit.

There are a few little things here and there to test, but otherwise, we rest satisfied with 4'776 and I'm moving on to the first of my two methods.