

## A Subtlety on the Mu Dual Variable

Recently I fixed the largest issue with the colgen that caused it to fail to converge to the correct answer. That (successful) effort is in "Finding the Column Generation Error.pdf": there was a superfluous check for application of the last z-value when no such check should exist.

When I run the new colgen algorithm loop after fixing this, the results mostly match up. That is, except for a few n\_jobs values (and this problem increases in severity as n\_jobs gets higher). There are slight differences (colgen is higher than SP/set-partitioning in all cases, as SP is guaranteed correct) at n = 22, 24, 28, and 29.

The most likely explanation? When I established the colgen, one of the questions I had was whether mu\_it should be added every time t we are at node i, or only when we first enter node i. And this was largely inconsequential, because mu was almost always 0. This could have been a source of error: perhaps we should add mu every time we are at a node, not only when we enter the node.

Fixing this issue gave me a correct answer for n = 29: SP established 3'003 as the objective, and I was printing 3'017 until I implemented the addition of mu\_it every time we are at node i, at which point the loop correctly printed 3'003.

However, this mu fix did not enable n = 22, 24, and 28 to be correct. The printed results indicate that the actual routes, while jobwise-disjoint, had slight job transpositions from the global minimum.

This probably happens because slackness is allowed at the constraint  $\sum_q u_i^q z^q = 1 \forall i$ .

Interestingly, when I forced == instead of >=, these cases became correct. But why is this? After all, with >=, the algorithm still yielded jobwise-disjoint paths! And yet with == the answer is different.

Paradoxically, for n = 29, the answer actually becomes wrong when I use == instead of >=. Instead of 3'003, I get 3'019, and the routes are slightly off.

Something is still wrong with the CG. To summarize, these are the fixes I made so far:

First, I deleted a piece of code which caused the colgen to prematurely terminate.

Second, I fixed the mu variable extraction.

Third and finally, I changed the uniqueness constraint back to a tight response. For some cases, we should use >= instead of ==. (This will be further investigated.)