

trip 53 bus 55 bus 77 taxi (see file 'optimization improvement example 1')

The solution is potentially resulted from how we handling the candidates resulted from local search `swap_area()`

Recall in our optimizer, we handles trips as followed:

**step 1:** generate candidates

[see codes](#)

```
for any trips pair t1, t2:  
    candidates <- swap any two compatible areas in between t1 and  
    t2  
    candidates <- try to combine t2 and a third trip into t1  
    candidates <- try to disspate t1 to other trips (exhaust)
```

**step 2:** sort the candidate and yield a best trip

**step 3:** For best trip's old trip, we only retain the best trip and kill all the other potentials from this old trip. (even a third swap might result in an optimal).

[see codes](#)

```
for any old trip resulted in current best trip:  
    remove old trip itself  
    remove any candidates originated from this old_trip
```

From the beginning, we know we can get to the optimal through two area swaps. For example, if we first `swap_area(trip 53, trip 55)` AND RETAIN THE RESULT which swap Maria Studios (from 55) with Altantica Eleon Grand (from 53). Then in the next optimization run, we have another area swap `swap_area(trip 53, trip 77)` and might yield a solution which swap Plessas Palace (from 77) with TUI BLUE Zante

Maris (from 53). But if as our optimize does, if the first swap between 53 and 55 yield a best trip which is not the same as the above first swap, then the solution with the above first swap would be killed in step 3.

(In other cases, if the best trip from the first swap doesn't originate from swap between trip 53 and trip 55, that's safe.)

Generally speaking, if an optimal solution is not monotonically cost-reducing from pair-wise swapping, there is a risk that we kill the paths to the optimal.

PS: we can't say the risk is high, in some cases that there are many alternative pairwise swaps that lead to the optimal or closed-optimal, killing some of them might not be an issue.

