Wed, Apr 21

HW 6 assign. Fri, due last day of class

Optional assign. on last day, due during exam week

HW 5: blue lights graph part.

good sol: 21s 1300s
best sol: 21.2514 1351
21.25139

blue lights: multiple tweaks

pick a random light
replace it with a new random light
a lot of exploration
not great for exploitation

tweaks = picking a random solution

GP unpenalized score

allow the two sides to have different sizes, but penalize it

$$|502 - 498| = 4$$
$$\text{score} \cdot \left(1 + \frac{4}{1000}\right)$$

Topic 18 - Variations on Local Search

"Local Search" = MHs where we look nearby, like H-C, S.A., T.S, etc.

Today: two variations to give you a taste of what's possible

#1 Variable Neighborhood Search

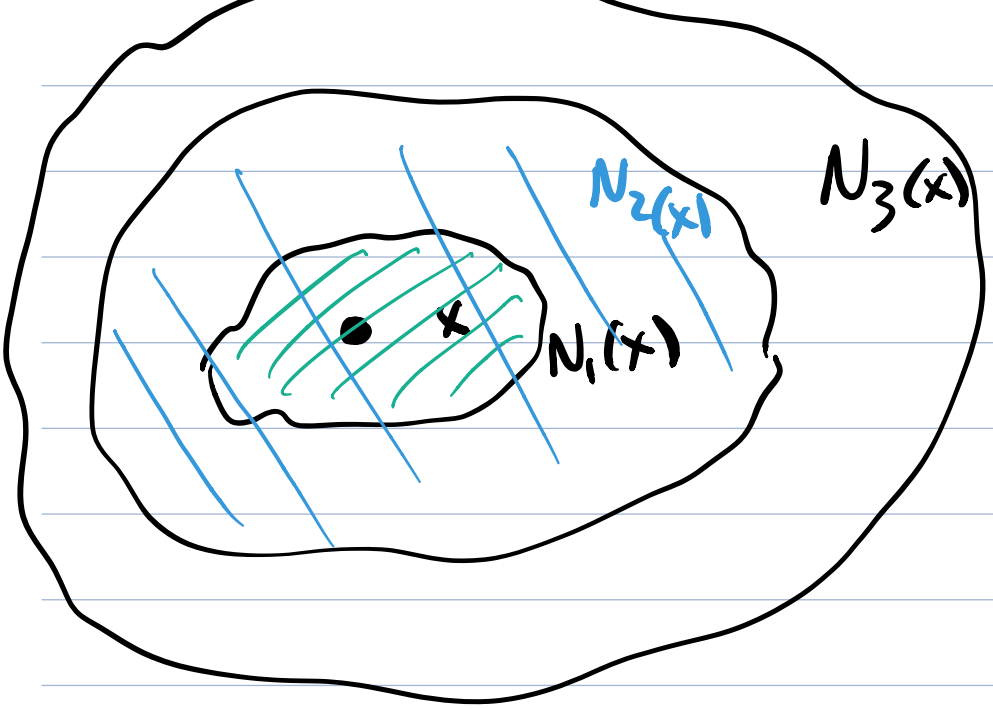
Idea: Define different kinds of tweaks (different neighborhoods).

$$N_1(x),\ N_2(x),\ N_3(x),\ \ldots,\ N_d(x)$$

move and move dramatic →

Often, but not always:

$$N_1(x) \subseteq N_2(x) \subseteq N_3(x) \subseteq \cdots \subseteq N_d(x)$$



With the blue lights:

N_1 = move one light a tiny amount

N_2 = delete a light and randomly replace it

$N_1(x) \subseteq N_2(x)$ ✓

Pseudocode:

```
Let x be a random solution.
While True:
    Pick s randomly from N1(x)
    if s is better than x:
        keep it and start the while loop over
    else:
        Pick s randomly from N2(x)
        if s is better than x:
            keep and start over
```

else:

Pick s randomly from $N_3(x)$

.... and so on

If you go through all neighborhoods with no improvement, just start over with $N_1(x)$.

* You can try a neighborhood move than once before moving onto the next one.

We need:

* a neighborhood structure

Ex: Knapsack

$N_k(x)$ = delete k random items, then greedily add items back until full

Ex: "Metaheuristics: From Design to Implementation" by Talbi

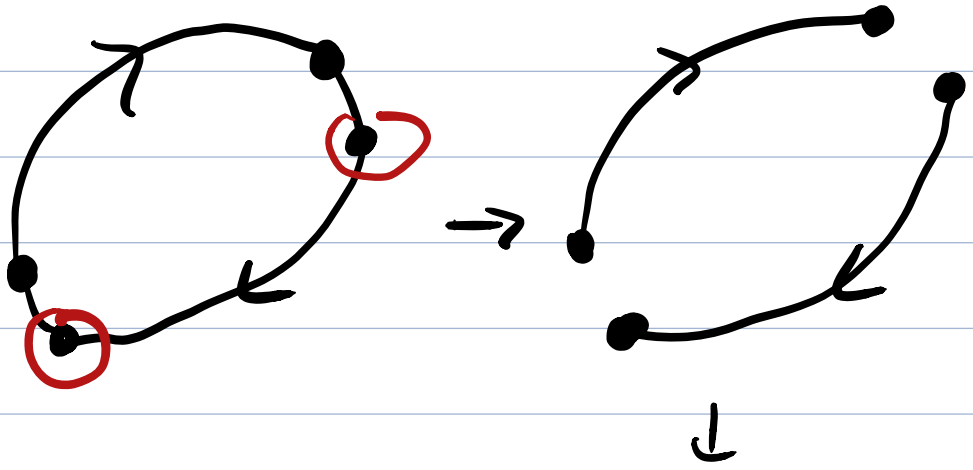
Example 2.35 Many neighborhoods in a real-life application. This example illustrates the use of many neighborhoods ($k = 9$) to solve a scheduling problem [26]. Many oil wells in onshore fields rely on artificial lift methods. Maintenance services such as cleaning, stimulation, and others are essential to these wells. They are performed by workover rigs. Workover rigs are available in a limited number with respect to the number of wells demanding service. The decision which workover rig should be sent to perform some maintenance services is based on factors such as the well production, the current location of the workover rig in relation to the demanding well, and the type of service to be performed. The problem of scheduling workover rigs consists in finding the best schedule S^* for the available m workover rigs, so as to minimize the production loss associated with the wells awaiting service. A schedule is represented by an ordered set of wells serviced by workover rigs. A variable neighborhood search metaheuristic has been proposed for this problem using the following neighborhoods in the shaking procedure:

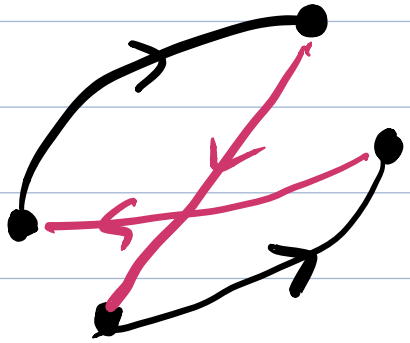
1. Swapping of routes (SS) where the wells associated with two workover rigs are exchanged.
2. Swapping of wells from the same workover rig (SWSW) where two wells serviced by the same workover rig are exchanged.
3. Swapping of wells from different workover rigs (SWDW) where two wells affected by two different workover rigs are exchanged.
4. Add/drop (AD) where a well affected by a workover rig is reassigned to any position of the schedule of another workover rig.
5. Two applications of the SWSW transformation.
6. Two applications of the SWDW transformation.
7. Three applications of the SWDW transformation.
8. Successive application of two AD transformations.
9. Successive application of three AD transformations.

Ex 3: TSP

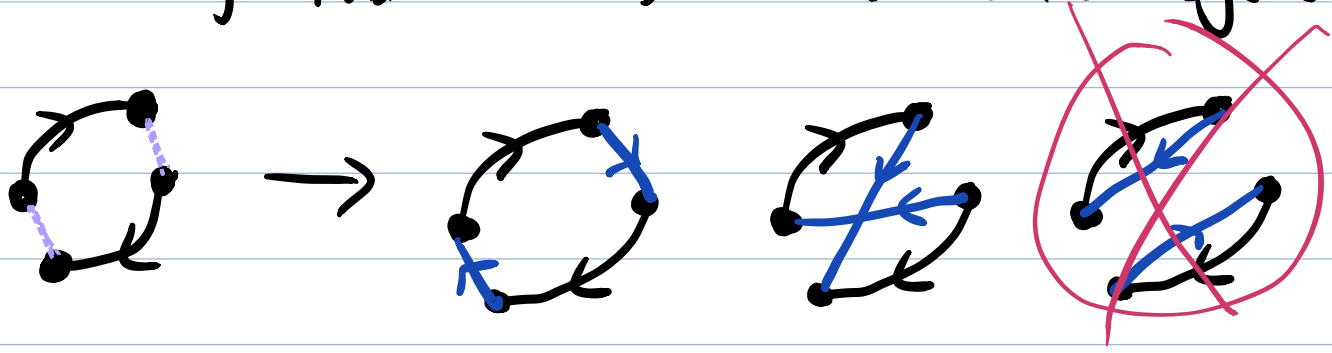
Good tweak we've used: "pick two cities, and reverse the path in between".

As a picture:





Another way to phrase this:
"delete two random edges, then reconnect the tour in the cheapest way that is still a valid cycle"



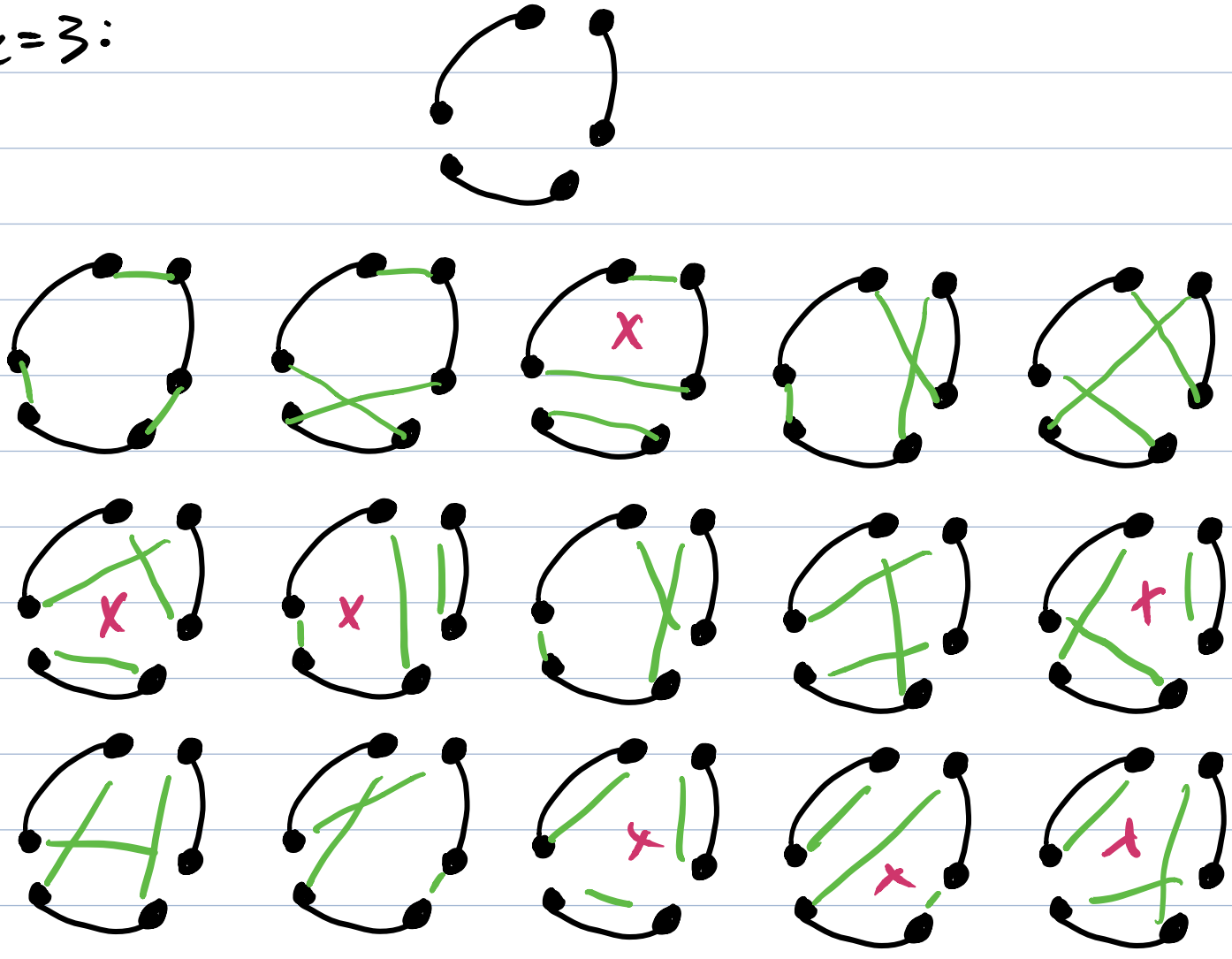
not a valid TSP solution

"k-opt": delete k edges, consider all valid ways of reconnecting

the tour, pick the cheapest

3-opt: most popular for H-C

k=3:



8 valid possibilities to check.