

Algoritmen en Heuristieken
Experiment
Rein, Michiel, Skip (GridHub)

De verschillende algoritmes hebben 10.000 oplossingen gevonden voor elke instantie van de case. De resultaten van de algoritmes worden hieronder met elkaar vergeleken aan de hand van verschillende statistieken. De beste score en de distributie van oplossingen worden met elkaar vergeleken. De resultaten worden per instantie per district besproken.

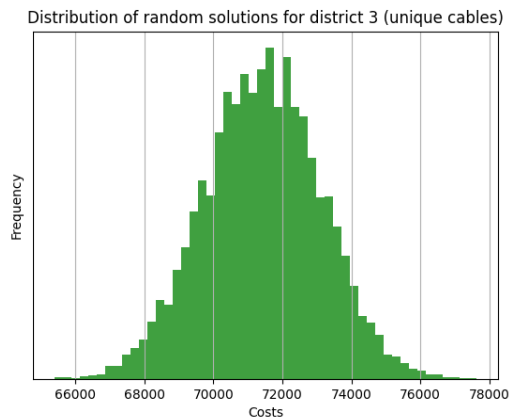
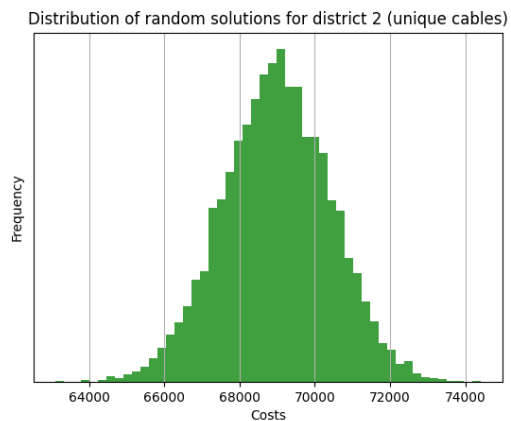
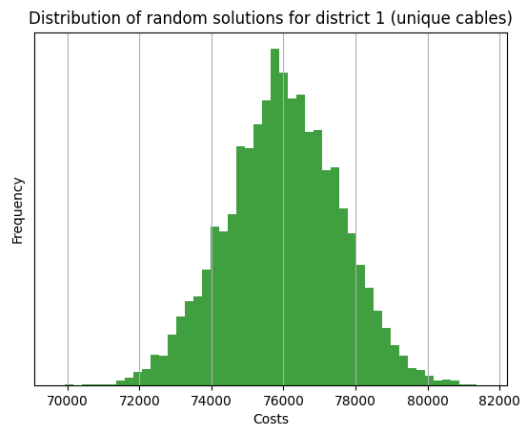
Instantie 1: random algoritme met unieke kabels

Allereerst het random algoritme, hierbij verwachten we dat het vergeleken met andere instanties oplossingen genereert met de hoogste kosten. Dit geeft ons wel iets om de andere oplossingen mee te vergelijken.

District 1: the lowest found cost: 69676

District 2: the lowest found cost: 63097

District 3: the lowest found cost: 65383



Voor deze instantie zien we dat de kosten tussen de 70.000 en 80.000 liggen. De beste random scores liggen tussen de 60.000 en 70.000. Het is te verwachten dat het makkelijk is om voor elk district een score lager dan 70.000 te krijgen, omdat dit algoritme redelijk snel werkt (10.000 runs binnen een kwartier).

Dit algoritme geeft waarschijnlijk niet de beste oplossing, maar komt wel snel met oplossingen.

Instantie 2: greedy algoritme met unieke kabels

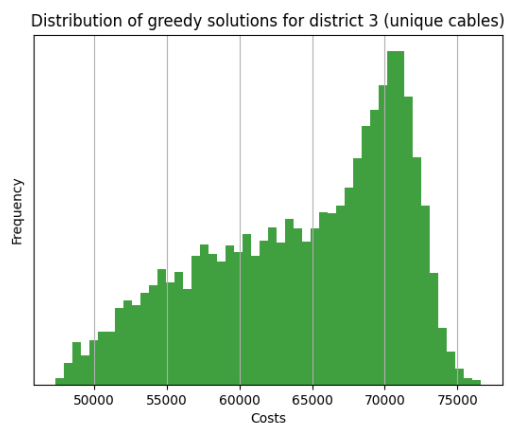
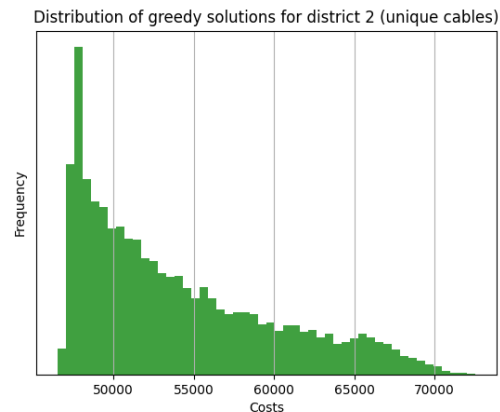
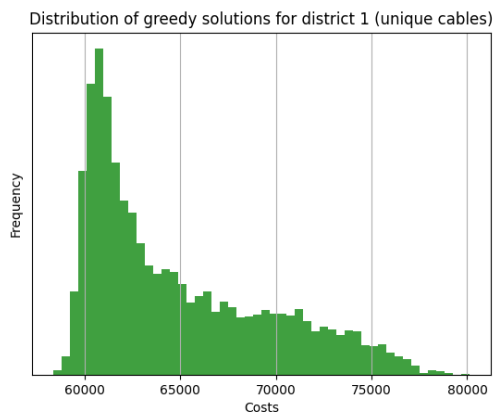
Een algoritme dat beter werkt is het greedy algoritme. Dit algoritme loopt langs alle huizen en bepaalt per huis wat de dichtstbijzijnde beschikbare batterij is. Dit betekent dat het aantal kabels in een district zal afnemen.

De gemiddelde kosten zijn veel lager dan bij het random algoritme. De best gevonden oplossingen zijn dan ook veel beter dan het random algoritme:

District 1: The lowest found cost: 57922

District 2: The lowest found cost: 46429

District 3: The lowest found cost: 46825



De distributie van de oplossingen is interessant. We zien dat het greedy algoritme voor district 1 en district 2 vaker een betere oplossing vindt met lage kosten, terwijl dit andersom is bij district 3. Dit zou kunnen liggen aan de ligging van de batterijen, waardoor het greedy algoritme in dit geval niet optimaal is. De laagst gevonden kosten zijn voor district 3 wel te vergelijken met de andere districts.

Instantie 3: random algoritme met shared cables

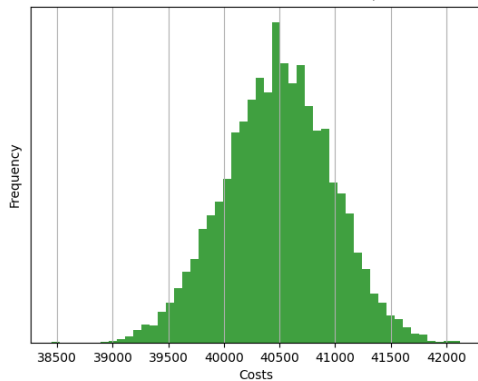
Vanaf nu mogen huizen kabels delen, waardoor er veel minder kabels nodig zijn (een groot gedeelte van de kabels lopen over hetzelfde gridsegment). Daarom wordt verwacht dat dit random algoritme met lagere kosten zal komen vergeleken de algoritmes met unieke kabels.

District 1: The lowest found cost: 38446

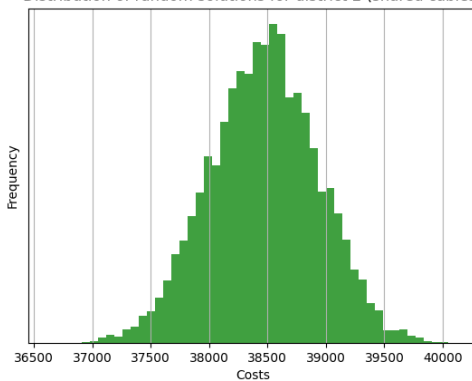
District 2: The lowest found cost: 36628

District 3: The lowest found cost: 37897

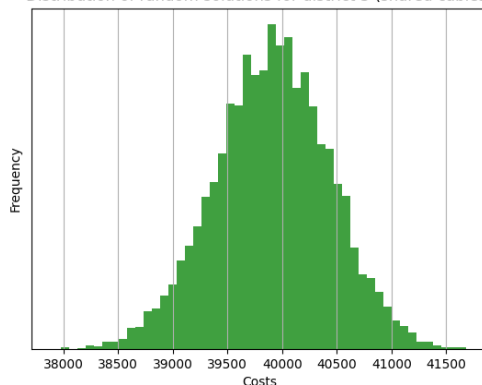
Distribution of random solutions for district 1 (shared cables)



Distribution of random solutions for district 2 (shared cables)



Distribution of random solutions for district 3 (shared cables)



We zien dat de kosten al sterk verminderd zijn, met beste oplossingen tussen de 35.000 en 40.000. De manier waarom huizen aan batterijen verbonden worden is nog steeds random en daarom verre van optimaal. Het is in ieder geval duidelijk dat het erg veel uitmaakt dat de kabels gedeeld kunnen worden. Ook dit algoritme komt redelijk snel tot oplossingen.

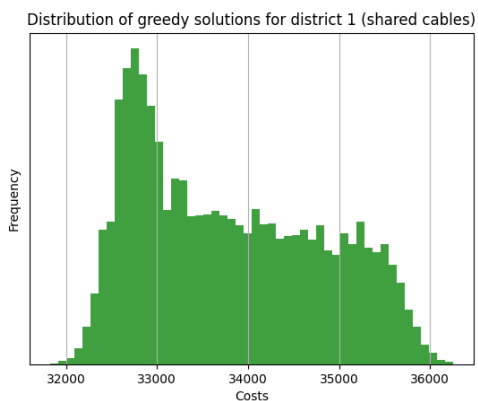
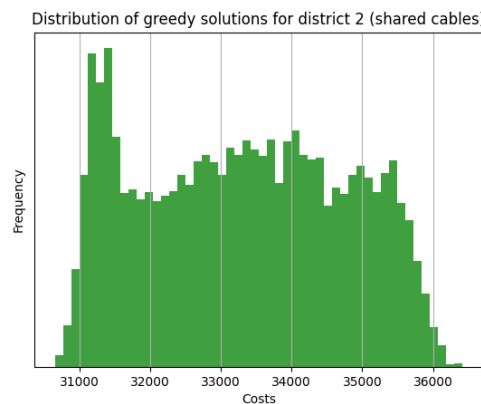
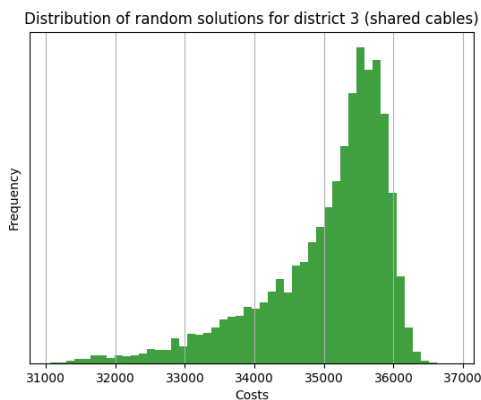
Instantie 4: greedy algoritme met shared cables

Dit laatste algoritme loopt langs alle huizen (in random volgorde) en bepaalt per huis wat de dichtstbijzijnde *kabel* is. De configuratie van huis-batterij combinaties wordt behouden en daarom verbindt het algoritme precies die huizen via een kabel. Dit zal zorgen voor betere oplossingen, maar er is nog steeds kans dat naast elkaar liggende huizen niet aan elkaar verbonden worden, waardoor er langere kabels kunnen ontstaan.

District 1: The lowest found cost: 31894

District 2: The lowest found cost: 30589

District 3: The lowest found cost: 31057



Bovenstaande zijn de best gevonden kosten per district. We weten dat de batterijen bij elkaar al 25.000 kosten, dus een oplossing met kosten rond de 30.000 komt in de buurt van een optimale oplossing.

Aan de distributies is te zien dat het algoritme niet vaak de beste oplossing vindt, maar doordat je het 10.000 oplossingen kunt laten genereren in ~15 minuten kom je redelijk snel tot een

redelijke oplossing. Er is een goede kans dat je nog betere oplossingen kunt vinden wanneer je het algoritme naar nog veel meer oplossingen laat zoeken.

Een andere optie is om hetzelfde algoritme te runnen wanneer de huizen gesorteerd zijn. Dit maakt het algoritme wel 95% deterministisch dus het aantal verschillende oplossingen dat je kunt vinden wordt sterk beperkt. Dit algoritme komt wel snel tot een redelijk goede oplossing wanneer je dit vergelijkt met de andere algoritmes:

District 1: The lowest found cost: 32929

District 2: The lowest found cost: 31390

District 3: The lowest found cost: 31255