

# Heuristic Algorithms using Tree Decompositions

Report 2    21/09/2021

De afgelopen twee weken waren opgesplitst in twee: de eerste week was ik bezig met verder implementatie en de tweede met de literatuurstudie.

Ondertussen is er een adapter toegevoegd om Jdrasil te gebruiken (de software dat Jorik heeft doorgestuurd). Momenteel gebruik ik deze enkel om een (very) nice tree decomposition te genereren, maar ik heb het zo proberen implementeren dat ik relatief eenvoudig het computen van een treedecomposition met Jdrasil kan toevoegen. Om Jdrasil te gebruiken heb ik een apart (java) project gemaakt in een apart git repo (<https://github.com/LouisCarpentier42/JdrasilAdapter>). Dit is apart omdat er volgens mij geen manier is om een nice tree decomposition te berekenen van een reeds bestaande tree decomposition (eg gegenereerd door flow cutter).

Ook heb ik een manier geïmplementeerd om een bepaalde tijd te geven aan de solver voor het berekenen van een tree decomposition: de solver zal telkens meer en meer tijd krijgen om een tree decomposition te berekenen, tot een maximum tijd of tot de treewidth onvoldoende daalt.

Verder heb ik ook wat algemene restructuring gedaan van de code en wat algemene fixes zoals een tree decomposition wrapper (voordien had ik enkel bag-objects)

Bij de literatuurstudie ben ik nog steeds de inleidende documenten aan het herhalen, maar ik ben ook al reeds begonnen aan de papers die een optimalisatie algoritme met behulp van een tree decomposition beschrijven. Hier ben ik echter nog niet zo ver mee omdat de papers redelijk moeilijk zijn. Volgens mij zullen ze wel extreem handig zijn want als ik van de algoritmes/technieken kan identificeren wat de exponentiele term in de running time veroorzaakt en hiervoor een heuristic kan vinden, zou ik al redelijk dicht zitten bij een algoritme. Maar hiervoor moet ik de algoritmes nog veel beter begrijpen eerst.