

# Heuristic Algorithms using Tree Decompositions

Report 3 06/10/2021

De afgelopen weken ben ik vooral bezig geweest met literatuur studie, met het oog op een toepassing om een algoritme voor te schrijven.

Het eerste wat mij interessant leek was de rank-based approach (<https://github.com/LouisCarpentier42/HeuristicAlgorithmsUsingTreeDecompositions/blob/main/documents/literature/rankBasedApproach/frameworkExplanation.pdf>). Dit is een optimalisatie voor standaard dynamic programming algoritmes. Na het processen van een node in de tree decomposition roep je een methode op die de dynamic programming table van die node zal prunen. De rank-based approach garandeert dat de table nog altijd een optimale oplossing bevat, maar deze blijft van exponentiele grootte ifv de treewidth. De rank-base approach werkt via gaussian elimination en er zijn al implementaties voor Hamiltonian Cycle en Steiner Tree die gebruik maken van de rank-based approach. Het framework zelf is redelijk complex en ik wil dit nog beter begrijpen zodat ik kan inschatten hoe praktisch het is voor het schrijven van een approximatie algoritme.

Verder heb ik ook nog gekeken naar andere problemen waarvoor ik eventueel een algoritme zou kunnen schrijven zoals het Maximum Happy Vertex problem. Hiervoor heb ik echter nog geen idee hoe ik een algoritme zou kunnen schrijven en moet ik dus meer voor lezen.

Een andere strategie die volgens mij relevant is wordt beschreven in <https://github.com/LouisCarpentier42/HeuristicAlgorithmsUsingTreeDecompositions/blob/main/documents/literature/2combinatorialOptimizationExamplesUsingTreeDecompositions/TSPTreeDecomposition.pdf>. Voor TSP bestaat er een heuristic K-opt die voor een bepaalde cycle  $k$  edges vervangt door  $k$  andere zodat de nieuwe cycle zo kort mogelijk is. In de paper beschrijven de auteurs een algoritme om de K-opt heuristic efficient uit te voeren met behulp van een tree decomposition. Ik ben niet zeker of het eenvoudig zal zijn hun algoritme aan te passen door een benaderings-algoritme, maar het kan volgens mij wel interessant zijn om een subcomponent van een bestaand algoritme te berekenen met behulp van een tree decomposition.