Baseline Chips & Circuits

Op het moment passen we een Breadth First Search algoritme toe dat voor elke gevraagde connectie tussen gates elke richting op zoekt tot er een pad naar de bestemming wordt gevonden. Het eerste pad wat wordt gevonden wordt toegevoegd aan het bord, en vanaf dat bord rekent het algoritme vervolgens verder. Komt dit later in het algoritme niet uit op een correct configuratie, dan wordt er een ander pad voor die gates genomen (ook als dat pad langer is).
Een perfecte baseline voor dit probleem zou rekening moeten houden met elk mogelijk pad van de eerste gate naar de tweede, en voor elk van die opties een nieuwe set opties moeten maken van de tweede gate naar de volgende, enzovoorts. Zelfs bij een kleine input loopt dit snel uit de hand.
Onze versie van het algoritme is deterministisch en geeft voor dezelfde input ook altijd dezelfde output. We zouden het algoritme zo kunnen omvormen dat er telkens een willekeurig pad tussen gates wordt gegeven i.p.v. de kortste die eerst wordt gevonden. Maar de vraag is dan: welk pad kies je? En hoe valt er te garanderen dat bij het kiezen van een willekeurig pad het algoritme überhaupt nog tot een oplossing kan komen? Als hiervoor gebacktrackt moet worden, en er gekozen moet worden uit alle mogelijke opties kom de situatie weer neer op het al eerder genoemde perfecte geval. Namelijk een waar alle mogelijke opties bekend zijn en er willekeurig eentje gekozen kan worden.
Om toch een iets univormere output van de baseline te geven hebben we het algoritme aangepast om willekeurig te kiezen in welke volgorde de nets worden gelegd. Hiermee komen we op iets meer gevarieerde outputs. Verder hebben we ook code geschreven die al tot een optimale oplossing weet te komen voor de eerste chip en netlist, door rekening te houden met alle kortste paden naar een gate, maar wel elk van dezelfde (en dus de kortste) lengte. We zouden deze code zo kunnen herschrijven dat het algoritme een van deze kortste paden willekeurig kiest en op dat punt verder rekent. Door tijdgebrek zijn we hier nog niet aan toegekomen. Als dit nog nuttig is zouden we hier wel over kunnen discussiëren.