Review Formulier (max 1 blz.) (Docent/Assistent of Student)

Titel paper : Optimalisatie van het Bewijs 'Sorteernetwerken van Optimale Grootte bij 9 Kanalen'

Auteur paper : Mathias Dekempeneer en Vincent Derkinderen

Een drietal zwakke punten:

1. probleemstelling en eigen bijdrages niet duidelijk

De titel '2. probleemstelling' was misleidend en hebben we gewijzigd naar Terminologie.
Verder hebben we dan een sectie 'Probleemstelling' tussengevoegd die kort aanhaald wat het probleem is (en dus ons doel) en waarom.
In de inleiding is toegevoegd wat de bijdrage van de paper is.

2. geen verklaring van waarom dit zoveel sneller is dan Codish et al

Goed punt, we weten echter niet exact wat onze implementatie sneller maakt.
We maken over het algemeen gebruik van hun concepten dus we veronderstellen dat onze snelheid voornamelijk te wijten valt aan het optimaler implementeren van de details.

1 duidelijk verschil dat ons momenteel bekend is, is dat zij tussenresultaten (de gedeelde lijst) wegschrijven naar files in het geheugen. Dit zorgt ervoor dat ze mogelijk trager zijn door de extra IO overhead, anderzijds hebben ze hierdoor de mogelijkheid tot tussenresultaten en kunnen ze gemakkelijk gebruik maken van meerdere nodes.

3. leest moeilijk zonder achtergrondkennis

Inhoudelijke vragen & suggesties voor verbetering:

Ik vond de abstract en introductie heel moeilijk om te volgen.
Ik stel voor om de abstract rond de volgende puntjes op te bouwen:
1. waar gaat dit over? (optimale sorteernetwerken vinden)
2. wat is de basis / wat doen Codish et al?
3. wat doen jullie anders, en wat zijn de resultaten daarvan?
In de introductie had ik graag eerst wat hoog niveau achtergrond en motivatie gezien ipv direct een soort lijst van definities (zonder dat ik weet waarom jullie het hierover hebben). Wat betekent het om "indirect een bewijs te geven"? Doen jullie dit ook?

Dit is ook door andere reviewers aangehaald. We hebben de abstract ondertussen volledig herschreven en de introductie bijgewerkt.
Het indirect bewijzen hadden we inderdaad nergens uitgelegd. Nu vermelden we de basis van dit bewijs bij de vermelding van lemma 1, dewelke gebruikt werd om voor 10 kanalen te bewijzen.

Sectie 2: dit is een mengeling van achtergrond en related work, maar geen probleemstelling -- wat is jullie taak?

zie uitleg 1ste punt bij "een drietal zwakke punten'.

In Sectie 3 mis ik structuur en overzicht. Hier komen heel veel aspecten aan bod, maar het is niet zo duidelijk hoe ze allemaal samenwerken, en wat jullie bijdragen zijn.
Ik zou proberen om een soort pseudo-code overzicht te geven van het algoritme, en hoe/wanneer de dingen beschreven in die subsecties erin gebruikt worden, en ook waar de verschillen zijn met de gelijkaardige aanpak door Codish.

Sectie 4 geeft een analyse van jullie implementatie, maar ik mis inzicht in wat jullie sneller maakt dan Codish.

zie uitleg 2de punt bij "een drietal zwakke punten'.

details:

- rond Figuur 1: help de lezer door een beetje meer uitleg rond de figuur en de notatie (het was mij bv niet direct duidelijk dat 1-4 in de tekst en de figuur verschillend zijn)

We hebben dit meer verduidelijkt aan de hand van de caption.

- def 1 / lemma 2: let op details notatie, bv C^n_{k,a} vs C_a (verschillend of niet?), C' in lemma 2

Er is nu in footnote vermeldt dat deze inderdaad (vaak) hetzelfde zijn. C^n_{k,a} werd gebruikt om te verduidelijken dat het comparator netwerk C_a n kanalen en k comparatoren.
Dit was een gemakkelijke en korte manier om te zeggen dat bijvoorbeeld C_a en C_b even veel kanalen en comparatoren hebben.
Wat betreft C', deze waren we vergeten verduidelijken in het begin van het lemma, dit is aangepast.

- def 1: wat is "outputs", en hoe kun je daarop \subseteq toepassen? (dit is pas duidelijk in sectie 3.3)

We hebben nu verduidelijkt dat wanneer er gesproken wordt over outputs van een netwerk dit hetzelfde is als spreken over de uitvoerverzameling van een netwerk.

- 3.1: waarom heb je op het einde maar n sequenties?

Er is nu verduidelijkt dat wanneer men een comparator toevoegd uitvoer mogelijk kan veranderen.
Dit zou dan een reductie kunnen betekenen in het aantal unieke outputs.
Aangezien een gesorteerde uitvoer altijd ongewijzigd blijft zal de uitvoerverzameling nooit uit minder dan n sequenties bestaan (n+1 als we 0000 meerekenen, n-1 als we zowel 0000 als 1111 niet meerekenen).

- footnote 3: hiervoor een appendix is wel vrij omslachtig...

De appendix is nu verwijderd.

- tabel 2: leg even uit, dit is niet zo evident

De caption is een beetje aangepast. Het is echter niet zo heel belangerijk om exact te begrijpen waarom de tabel een bepaalde waarde bevat. Het is de bedoeling dat de lezer zeker begrijpt wat precies de structuur is van de tabel. Dit houdt in: dat we een lijst van comparatoren hebben, dat de uitvoer opgedeeld is per aantal 1'en en dat de bits van belang zijn.

- 3.3: ik snap niet goed waar het aantal “lemma’s nagaan” en “beslissingen nemen” over gaat

Dit is nu verduidelijkt in de evaluatie van de 'beslissingen' en enkele eerdere zinnen met beslissingen en 'lemma's nagaan' zijn anders geformuleerd.

- voorbeeld tabellen 4&5: geef meer details

Er zijn enkele zinnen toegevoegd om te verduidelijken hoe de reductie plaats vindt.

Hartelijk bedankt voor de feedback!