

## **1 Artikel 1**

### **1.1 Wat is de vraagstelling van dit artikel?**

(hoe) kan het geheugengebruik van graph-search algoritmen beperkt worden?

### **1.2 Wat is de voornaamste claim die in het artikel wordt gemaakt? Wat is het resultaat dat hier wordt gepresenteerd?**

Met hun techniek hoeft er minder geheugen te worden gebruikt. De tabel met resultaten laat zien dat alleen in één van de drie gevallen de gepresenteerde techniek beter is dan de al bestaande.

### **1.3 Hoe ziet het met de implementatie? Is de implementatie relevant voor het antwoord op de vraagstelling?**

Ja, om de vraag te kunnen beantwoorden moet je het in de praktijk kunnen testen.

### **1.4 Hoe beargumenteren de auteurs dat hun werk origineel is?**

Er worden een aantal eerdere manieren genoemd om geheugengebruik te beperken. De auteurs hebben echter een flexibere manier gevonden, zeggen ze zelf.

### **1.5 Wat claimen de auteurs over praktische relevantie?**

Vooraf bij het vergelijken van grote DNA sequenties is dit algoritme nuttig. In het algemeen bij zoekproblemen met een hoge branching factor: veel vertakkingen in de zoekboom.

### **1.6 Geeft de titel goed weer waar het artikel over gaat? Weet je een betere titel?**

Als je je een beeld kunt vormen bij het woord “sparse” wel, anders zou “Increasing memory efficiency of graph search” een goede zijn.

### **1.7 Stel dat het artikel met 20% ingekort zou moeten worden. Hoe zou dat kunnen?**

Volgens mij kan het eigenlijk niet. Maar als het echt moet zou ik de achtergronden en beschrijvingen van eerdere algoritmen eruit halen; en alle plaatjes en grafieken.

### **1.8 Stel dat het artikel 50% langer zou mogen zijn. Wat zou toegevoegd kunnen worden?**

Er zou beschreven kunnen worden voor welk soort zoekproblemen dit algoritme juist niet werkt, en waarom niet. Er zou verwezen kunnen worden naar een echte implementatie. Maar 50% is wel wat veel, dan zou je het onderwerp breder moeten maken; bijvoorbeeld met een nieuw zoekalgoritme wat weer voor hele andere problemen geschikt is.

## **2 Artikel 2**

### **2.1 Wat is de vraagstelling van dit artikel?**

Kan het mini-max algoritme verbeterd worden qua snelheid en geheugengebruik.

### **2.2 Wat is de voornaamste claim die in het artikel wordt gemaakt? Wat is het resultaat dat hier wordt gepresenteerd?**

Ze hebben een nieuwe versie van minimax die beter werkt dan alphabeta-pruning als de diepte niet te hoog is. Een combinatie van bestfirst minimax en alphabeta geeft echter nog betere resultaten.

### **2.3 Hoe ziet het met de implementatie? Is de implementatie relevant voor het antwoord op de vraagstelling?**

De implementatie is wederom belangrijk om het resultaat te kunnen meten.

### **2.4 Hoe beargumenteren de auteurs dat hun werk origineel is?**

Er worden een aantal meldingen genoemd van bestfirst minimax in de literatuur, maar die zijn nooit uitgewerkt.

### **2.5 Wat claimen de auteurs over praktische relevantie?**

Daar wordt niet over gesproken; het gaat om spelletjes dus deze zal ver te zoeken zijn.

## **2.6 Geeft de titel goed weer waar het artikel over gaat? Weet je een betere titel?**

De titel is prima.

## **2.7 Stel dat het artikel met 20% ingekort zou moeten worden. Hoe zou dat kunnen?**

Veel stap-voor-stap uitleg kan ingekort worden. Verder had het bewijs van correctheid ook in de appendix kunnen staan - als het algoritme in de praktijk werkt dan kan je aannemen dat het correct is.

## **2.8 Stel dat het artikel 50% langer zou mogen zijn. Wat zou toegevoegd kunnen worden?**

Het artikel is al zeer uitgebreid dus dat lijkt me geen goed idee. Maar de vermelding van Go in de conclusie vond ik wel zeer interessant, zou dit algoritme hiervoor nuttig zijn?

## **3 Bronnen**

- (1) Rong Zhou en Eric A. Hansen, Sparse-Memory Graph Search, gepresenteerd op het IJCAI congres van 2003
- (2) Richard E. Korf en David M. Chickering, Best-first minimax search, Artificial Intelligence. Vol. 84, no. 1-2, pp. 299-337. 1996.