

Määrittelydokumentti

1. Työn esittely

Kivi-sakset-paperi-tekolyn toiminta perustuu erilaisten ennustusalgoritmien ja metastrategioiden yhteistyöhön. Työssä käytetään Markovin malleihin perustuvia algoritmeja (MarkovFirst ja MarkovSecond), jotka pitävät kirjaa yleisimmistä siirtoyhdistelmistä hyvin tehokkaasti, sekä kuvionetsintään perustuvia algoritmeja (PatternMatching ja PatternMatchingPlayer), jotka etsivät siirtohistoriasta nykyistä pelitilannetta vastaavia tilanteita.

Ongelmana on siis etsiä ja hyödyntää tehokkaasti vastustajan mahdollisia toistuvuuksia ja käyttää niitä sitä vastaan. Tähän tarkoitukseen Markovin mallit ovat ideaaleja, koska ne ovat nopeita mutta myös toimivia ihmispelaajaa vastaan. Niiden lisäksi tarvitaan kuitenkin muita algoritmeja pidempien toistuvuuksien löytämiseksi, koska testeissä kävi ilmi että pelkästään markovin malleihin perustuvat tekoälyt olivat alttiita häviämään tietyille toistuville kuviolle joka kerta. "Syötteinä" algoritmit saavat aina molempien pelaajien joko viimeisimmät siirrot, tai koko siirtohistorian.

2. Käytettävät algoritmit ja tietorakenteet

Markov-algoritmit rakentavat siirtohistoriasta kaksiulotteisen taulukon, joissa y-akseli on edellinen siirto tai siirtoyhdistelmä ja x-akseli siirtojen jälkeen tullut siirto. Tästä taulukosta on siis helppo laskea mikä on yleisin siirto tietyn siirron tai siirtoyhdistelmän jälkeen. Algoritmien aikavaativuus on $O(n)$.

PatternMatching-algoritmit saavat syötteenään molempien pelaajien siirtohistorian, mistä ne luovat n :stä viimeisestä siirrosta tai siirtoyhdistelmästä kuvion, jota ne sitten etsivät aikasemmista siirroista. Algoritmin aikavaativuus on $O(n^2)$ vaikka jotkin tehokkaammat algoritmit mahdollistaisivat jopa $O(n)$ aikavaativuuden.

Itse tekoäly koostuu pisteytys- ja valintataulukosta, joiden avulla eri strategioiden suoritusta voidaan tarkastella ja valita paras strategia. Nämä operaatiot ovat vakioaikaisia.

3. Lähteet

<http://www.dllu.net/programming/rps/>

http://en.wikipedia.org/wiki/Markov_chain