

Posudek diplomové práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce Petr Zábaj
Název práce Efektivní paralelizace evolučních algoritmů
Rok odevzdání 2020
Studijní program Informatika **Studijní obor** Umělá inteligence

Autor posudku Mgr. Martin Pilát, Ph.D. **Role** vedoucí
Pracoviště KTIML MFF UK

Text posudku:

Student se ve své práci zabývá problémem paralelizace evolučních algoritmů v případě, kdy vyhodnocení fitness funkce je pomalé a navíc trvá pro každého jedince jinak dlouho. V takových případech tradiční paralelní evoluční algoritmy trpí tím, že v každé generaci musí čekat na vyhodnocení pomalých jedinců. Tento problém se dá řešit použitím tzv. asynchronních algoritmů, ty ale zase trpí tím, že jsou zatížené na části prohledávaného prostoru, kde je vyhodnocení fitness rychlejší. Student tedy v práci rozšířil algoritmus, který používá tzv. prolínání generací, kde se noví jedinci generují jakmile jsou k dispozici jejich rodiče. Díky tomu se v jednu chvíli mohou vyhodnocovat jedinci z několika generací, jinak je ale algoritmus ekvivalentní s tradičním evolučním algoritmem. Rozšíření spočívalo především v přidání spekulativního vyhodnocování – pokud není možné dostatečně vytížit CPU vytvořením jedinců na základě známých rodičů, vytvoří se nějaký odhad, jaký jedince se stane rodičem.

Celá práce je rozdělena do čtyř kapitol. První kapitola poskytuje úvod do problému paralelizace evolučních algoritmů včetně popisu moderních verzí paralelních algoritmů a jejich problémů v případě optimalizace funkcí, kde vyhodnocení každého jedince trvá jinak dlouho. Kapitola je napsána s rozumným množstvím detailů v jednotlivých částech a zaměřuje se na informace potřebné ve zbytku práce.

Další tři kapitoly obsahují vlastní přínos práce. Druhá kapitola popisuje základní myšlenku navrženého algoritmu s prolínáním generací a spekulativním vyhodnocováním. Ta je nejdříve vhodně demonstrována na jednodušší verzi typu (μ, μ) (tedy počet rodičů je stejný jako počet potomků a environmentální selekce vybírá právě všechny potomky) a až potom je zobecněna na mnohem komplikovanější verze (μ, λ) a $(\mu + \lambda)$. Popis algoritmu zde je na rozumné úrovni detailu a dobře vysvětluje rozhodnutí učiněná během jeho vytváření. To, že se nakonec implementačně podařilo sjednotit verze (μ, λ) a $(\mu + \lambda)$ tak, že používají stejný kód jen s několika málo podmínkami svědčí o tom, že analýza problému byla provedena důkladně.

Třetí kapitola potom obsahuje další implementační detaily, které byly v popisu v předchozí kapitole vynechány, aby příliš nezastřely hlavní myšlenku algoritmu. Toto rozdělení považuji vzhledem k relativně velké komplexnosti algoritmu za velmi vhodné.

Čtvrtá kapitola potom popisuje provedené experimenty. Těch je celá řada v závislosti na různých fitness funkcích, konfiguracích evolučního algoritmu a počtech CPU. Všechny tyto experimenty byly navíc několikrát opakovány pro ukázání statistické významnosti. Ukazuje se, že navržený algoritmus dosahuje lepšího využití CPU než existující algoritmy. Bohužel se také v některých konfiguracích ukazuje, že algoritmy nejsou úplně ekvivalentní s generačním evolučním algoritmem, což jsme nečekali, vzhledem k tomu, že celý algoritmus byl navržen tak, aby ekvivalentní byl (tj. aby na konci běhu dosáhl stejné fitness jako generační algoritmus bez jakéhokoliv zatížení vlivem různých dob vyhodnocení fitness funkce). Tyto konfigurace jsou ale relativně málo časté. Proč k těmto rozdílům dochází není zřejmé a nepodařilo se to zatím vysvětlit.

Celkově je práce relativně dobře napsaná, algoritmy jsou i přes jejich složitost a množství okrajových případů vysvětleny srozumitelně. Nejsložitější částí práce se ukázala implementace, kde bylo mnohem více obtíží a okrajových případů, které bylo potřeba vyřešit, než bylo očekáváno. Student se nicméně se všemi těmito problémy vypořádal a tím ukázal, že je schopen samostatné práce na relativně složitých problémech.

Práci doporučuji k obhajobě.

Práci nenavrhuji na zvláštní ocenění.

V Praze dne 3. září 2020

Podpis: