

EVO - Aplikované evoluční algoritmy

Simulované žihání - optimalizace řadičů sítě

Číslo zadania: 2

Meno a priezvisko: Richard Seipel

Login: xseipe00

1 Zadanie projektu

Cieľom je optimalizovať radiacu sieť použitím algoritmu simulovaného žihání. Je potrebné zvoliť vhodnú reprezentáciu radiacej siete a simulovať jej použitie. Ďalej implementovať algoritmus simulovaného žihání a prípadne ho vhodne upraviť pre potreby projektu. Súčasťou riešenia je taktiež použitie vyhovujúcej fitness funkcie a procesu chladnutia. Významnou časťou je experimentovanie s nastavením parametrov a výber najvhodnejších z nich. Výsledkom má byť program gerenujúci návrh funkčnej a efektívnej radiacej siete. Nasledujúce sekcie obsahujú pohľad na riešenie problému s mojimi doterajšími znalosťami a finálna implementácia jednotlivých častí sa ešte môže zmeniť. Tento projekt sa chystám implementovať v jazyku Python.

2 Implementácia radiacej siete

Pre implementáciu radiacej siete je potrebné vhodne zvoliť jej reprezentáciu. Zvolil som pole trojíc v tvare (i_1, i_2, u) , kde i_1 a i_2 reprezentujú pozíciu vybraných hodnôt v sieti, ktoré majú byť porovnané komparátorom a prípadne vymenené. Parameter u obsahuje hodnotu **True** alebo **False**, ktorá reprezentuje, či dané prepojenie (komparátor) bude v sieti použité. K pridaniu tohto parametra som sa inšpiroval v článku [1], priloženom v zadaní. Možnosť ignorovania komparátoru dovoľí optimalizačnému algoritmu zostavovať a ohodnocovať radiace siete s rôznym počtom komparátorov. Následne použitím správnej fitness funkcie bude hľadanie riešenia smerované k menšiemu počtu komparátorov, a tým pádom menšej a jednoduchšej sieti. Samotný komparátor bude implementovaný jednoduchou funkciou, ktorá prijme dva para-

metre s hodnotami na vybraných pozíciach v sieti, a ak je prvá väčšia ako druhá vykoná ich výmenu. Simulácia siete prebehne postupným použitím komparátora na všetkých pozíciach určených sieťou nad vstupnými dátami.

3 Implementácia simulovaného žihání

Algoritmus v každej iterácii použije náhodné vstupné hodnoty v určom rozsahu a spočíta nad nimi hodnotu fitness pre aktuálnu inštanciu siete. Následne vymení poradie dvoch náhodne zvolených komparátorov a s určitou pravdepodobnosťou odstráni alebo vráti niektorý komparátor (zmena parametra u). Následne vyhodnotí, či takto zmenená sieť bude použitá, prípadne uloží najlepšiu a pokračuje ďalšou iteráciou.

4 Ďalšie detaily implemetácie

Sieť bude inicializovaná so všetkými možnými kombináciami hodnôt pre danú dĺžku vstupu, kde prvá hodnota (i_1) je vždy väčšia ako druhá (i_2). Fitness funkcia bude zohľadňovať počet použitých komparátorov a počet nesprávne zoradených prvkov, pričom počet nesprávne zoradených prvkov bude mať vyššiu prioritu. Teplota bude v každom kroku násobená konštantou α .

5 Predbežný výstup

Vyššie popísaná implementácia sa pri zjednodušenom manuálnom testovaní javí ako použiteľná, pre jej vylepšenie bude potrebné dôsledné použitie štatistických metód a vyhodnotenie ich výstupov.

Literatúra

- [1] MRAZEK, V. a VASICEK, Z. Automatic design of arbitrary-size approximate sorting networks with error guarantee. In: IEEE. *2016 26th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*. 2016, s. 221–228.