

# PARALLEL MODEL OF GENETIC ALGORITHM FOR KNAPSACK PROBLEM

**Martin Majtán**

Bachelor Degree Programme (III), FEEC BUT

E-mail: xmajta02@stud.feec.vutbr.cz

Supervised by: Jan Karásek

E-mail: karasekj@feec.vutbr.cz

**Abstract:** This paper describes use of a parallel model for reducing time of a genetic algorithm for the knapsack problem. First part describes a genetic algorithm for the knapsack problem. Next part describes a parallel model and its use with a genetic algorithm. Last part describes setting of a genetic algorithm and results of measured times of a genetic algorithm are depicted in a table. A genetic algorithm is implemented with Java programming language.

**Keywords:** parallel algorithm, genetic algorithm, knapsack problem

## 1 ÚVOD

Táto práca sa zaoberá porovnaním jednovláknového modelu a paralelného modelu pre použitie v genetickom algoritme z hľadiska rýchlosti výpočtu. V tejto práci je popísaný paralelný model a jeho využitie na zredukovanie času potrebného na beh genetického algoritmu, ktorý rieši problém naplnenia batohu. Pokračovaním tejto práce bude vytvorenie knihovny na paralelizáciu genetického algoritmu.

## 2 POPIS TESTOVACIEHO PROBLÉMU

Genetický algoritmus [1] patrí do triedy evolučných algoritmov. Evolučné algoritmy sú pravdepodobnostné prehľadavacie algoritmy založené na mechanizme prirodzeného výberu a princípoch evolučnej biológie. Tieto princípy sú prevzaté z Darwinovej evolučnej teórie [2] a aplikované v informatike.

V každom genetickom algoritme sa na začiatku inicializuje počiatočná populácia. Populáciu tvoria chromozómy, tie sa ďalej vytvárajú z génov. Po inicializácii populácie sa na populáciu aplikuje fitness funkcia, je to funkcia ktorá ohodnotí jednotlivých jedincov populácie, podľa veľkosti hodnoty fitness funkcie sú vybratí jedinci do procesu vývoja populácie. V tomto procese sa aplikujú genetické operátory, sú to operátory kríženia, mutácie a elitizmu. Operátor elitizmu predstavuje funkciu ktorá skopíruje určitý počet najlepších jedincov do novej populácie. Operátor mutácie náhodne vyberie jeden alebo viacej génov a zmutuje ich. Operátor kríženia náhodne vyberie bod kríženia a v tomto bode spraví kríženie dvoch chromozómov. Po aplikovaní operátorov sa vytvorí nová populácia z potomkov, ktorí vznikli aplikovaním genetických operátorov, a aplikuje sa fitness funkcia. Tento proces sa opakuje čím sa dosahuje postupný vývoj jedincov v populácii. Algoritmus sa ukončí keď nájde jedinca ktorý vyhovuje podmienkam ktoré užívateľ nastaví v algoritme.

V tejto práci je použitý typický problém z oblasti evolučných algoritmov, problém naplnenia batohu [3]. Jedná sa o problém pri ktorom je potrebné naplniť batoh rôznymi predmetami. Každý predmet má rôznu cenu a hmotnosť a môže byť v batohu len raz, algoritmus prehľadáva kombinácie týchto predmetov. V genetickom algoritme ktorý rieši problém naplnenia batohu predstavujú jednotlivé predmety gény v genetickom algoritme. Zoznamy týchto predmetov predstavujú chromozómy, z ktorých sa ďalej vytvárajú populácie.

Fitness funkcia v použítom genetickom algoritme zoradí jedincov podľa súčtu cien predmetov v batohu od najväčšej ceny, potom sa aplikuje elitizmus a operátory mutácie a kríženia s určitou pravdepodobnosťou. Ukončenie algoritmu je v prípade keď je splnená podmienka čo najväčšej celkovej sumy predmetov v batohu a zároveň nie je prekročená maximálna nosnosť batohu alebo keď je splnená obmedzujúca podmienka maximálneho počtu evolučných krokov.

Algoritmus bol analyzovaný a ako miesta vhodné na paralelizáciu boli detekované metódy ktoré slúžia na vytváranie populácie a aplikovanie operátorov mutácie a kríženia. V týchto metódach musí algoritmus robiť veľký počet opakovaní v závislosti od veľkosti populácie, preto sú vhodné na paralelizáciu.

## **2.1 PARALELNÝ MODEL**

Paralelný model [4] predstavuje model pri ktorom sa robí paralelizácia medzi viacej vlákien na jednom počítači. Výpočet ktorí sa má paralelizovať sa rozdelí na menšie časti a tie sa potom jednotlivito vykonávajú vo vláknach. Po dokončení výpočtov v jednotlivých vláknach sa medzi výsledky vrátia do hlavného vlákna kde sa spoja a ďalej sa s nimi pracuje. Hlavnou výhodou použitia paralelného modelu je zredukovanie času potrebného na beh algoritmu. Výhody paralelného modelu sa prejavajú už pri výpočtoch ktoré trvajú rádovo desiatky sekúnd, pretože inicializácia a režia vlákien trvá rádovo mikrosekundy. Pri paralelnom modeli je vhodné používať len toľko vlákien, koľko podporuje procesor, pri väčšom počte vlákien sa výkon nezlepší, dokonca môže byť horší, čo je spôsobené zvýšenou réziou vlákien, jedno jadro procesoru spracováva striedavo viac vlákien.

Paralelizácia genetického algoritmu je spravená na miestach kde sa vytvára populácia a aplikujú genetické operátory mutácie a kríženia. Paralelizácia metódy na vytváranie populácie je spravená tak, že veľkosť populácie ktorá sa má vytvoriť sa rozdelí na menšie časti a tie sa potom jednotlivito vytvárajú vo vláknach. Po skončení vytvárania menších populácií sa vytvorí jedna o zadanej veľkosti s ktorou sa ďalej pracuje. Paralelizácia metód na aplikovanie genetických operátorov je spravená rovnako ako paralelizácia vytvárania populácie. Populácia sa rozdelí na menšie časti, na tie sa potom aplikujú genetické operátory.

## **3 VÝSLEDKY TESTOVANIA**

### **3.1 NASTAVENIE ALGORITMU**

Meranie času genetického algoritmu sa robilo pre veľkosť populácie 100, 1 000, 10 000, 20 000 a 50 000 jedincov. Meranie sa robilo 20 krát a potom sa spravil aritmetický priemer. Meranie času genetického algoritmu bolo robené pomocou systémového času, ktorí sa na začiatku a na konci algoritmu zapísal do premenných. Výsledný čas sa vypočítal ako rozdiel týchto časov.

Nastavenie parametrov genetického algoritmu:

Maximálny počet generácií: 20

Maximálna hmotnosť predmetov v batohu: 5 kg

Minimálna hmotnosť predmetov v batohu: 4,8 kg

Minimálna cena predmetov v batohu: 21 200 CZK

Pomer kríženia: 30 %

Pomer mutácie: 10 %

Hodnota elitizmu: 5 (počet jedincov, ktorí sa bez zmeny skopírujú do novej generácie)

### 3.2 POROVNANIE MODELOV

Meranie genetického algoritmu sa robilo na počítači s dvoj-jadrový procesor Intel i5 2410M 2,3GHz s podporou spracovávania štyroch vlákien naraz s 64-bitovým operačným systémom Microsoft Windows 7 a 4 GB pamäte RAM.

| Veľkosť populácie         | 100     | 1 000 | 10 000 | 20 000 | 50 000 |
|---------------------------|---------|-------|--------|--------|--------|
|                           | Čas [s] |       |        |        |        |
| Jednovláknový model       | 0,093   | 0,836 | 9,759  | 28,288 | 91,247 |
| Paralelný model-2 vlákna  | 0,115   | 0,797 | 7,977  | 20,974 | 65,505 |
| Paralelný model-4 vlákna  | 0,117   | 0,766 | 8,739  | 20,767 | 62,713 |
| Paralelný model-8 vlákien | 0,140   | 0,794 | 8,453  | 21,998 | 65,158 |

**Tabulka 1:** Porovnanie jednovláknového a paralelného modelu

Podľa nameraných časov v Tabuľke 1 je vidieť, že použitie paralelného modelu nie je vhodné na všetky veľkosti populácie. Pri veľkosti populácie 100 jedincov bol čas s použitím paralelného modelu horší ako v jednovláknovom spracovaní, čo je spôsobené tým, že režia vlákien trvá príliš dlho v pomere k času výpočtu. Pri veľkosti populácie 1 000, 10 000, 20 000 a 50 000 jedincov bol čas lepší pri použití dvoch, štyroch a ôsmich vlákien.

## 4 ZÁVER

Pomocou paralelného modelu sa podarilo zredukovať čas potrebný na beh genetického algoritmu ktorí rieši problém naplnenia batohu. Výhody paralelného modelu v genetickom algoritme sa prejaví až pri veľkosti populácie 1 000 jedincov a viac. Pri menšom počte jedincov sa čas zhoršil. Z nameraných časov je vidieť, že s rastúcou náročnosťou výpočtu sa viac prejavuje rozdiel v použití jednovláknového a paralelného modelu. Najlepšieho zredukovania času sa dosiahlo pri veľkosti populácie 50 000 jedincov, 31,27 %. Zvýšená režia vlákien sa najviac prejavila pri veľkosti populácie 100 jedincov, pri väčších populáciách nie je režia vlákien tak výrazná. Z nameraných hodnôt je vidieť, že zvyšovanie počtu vlákien neprináša lepšie výsledky. Pokračovaním tejto práce bude vytvorenie distribuovaného modelu pre problém naplnenia batohu.

## POĎAKOVANIE

Tento príspevok vznikol za podpory špecifického výskumného zámeru FEKT-S-11-17.

## REFERENCE

- [1] Obitko, M. Genetic Algorithms, [online]. [cit. 10.2.2014]. Dostupné z: <http://www.obitko.com/tutorials/genetic-algorithms/index.php>.
- [2] Peter, M. Evolúcia a evolučná teória, [online]. [cit. 10.2.2014]. Dostupné z: <http://www.evolutionrevolution.eu/sk/>
- [3] Nils, H. Genetic Algorithm vs. 0-1-KNAPSACK, [online]. [cit. 19.3.2014]. Dostupné z: <http://www.nils-haldenwang.de/computer-science/computational-intelligence/genetic-algorithm-vs-0-1-knapsack>
- [4] Blaise, B. Introduction to Parallel Computing, [online]. [cit. 19.3.2014]. Dostupné z: [https://computing.llnl.gov/tutorials/parallel\\_comp/](https://computing.llnl.gov/tutorials/parallel_comp/)