Minimum Multiprocessor Scheduling

Racunarska inteligencija

Matematicki fakultet

Nenad Dobrosavljevic

mi20088@alas.matf.bg.ac.rs

Sadržaj:

1. Uvod
2. Formalizacija problema
3. Primena grube sile
4. Optimizacije
   1. Lokalna pretraga
   2. Simulirano kaljenje
   3. Genetski algoritam
   4. Optimizacija rojem čestica
5. Poredjenje rešenja
6. Zaključak
7. **Uvod**

Problem *Minimum Multiprocessor Scheduling* predstavlja jedan od klasičnih problema u oblasti optimizacije i raspodele zadataka u računarstvu. Cilj ovog problema je raspodela skupa zadataka na više procesora kako bi se minimizovalo vreme završetka (makespan). Ovaj problem je NP-težak, što znači da optimalno rešenje može biti teško pronaći u razumnom vremenu za velike instance problema.

U ovom radu ćemo opisati i uporediti nekoliko pristupa za rešavanje ovog problema:

* Brute-force metod
* Local Search algoritam
* Simulated Annealing (SA)
* Genetski algoritam (GA)
* Optimizacija rojem čestica

1. **Opis problema**

**Ulazni podaci:**

* Skup zadataka (T)
* Broj procesora (m)
* Vreme izvršavanja svakog zadatka na određenom procesoru (execution times)

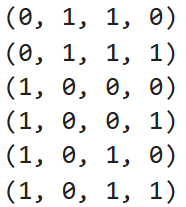
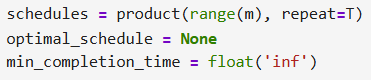
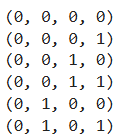
**Izlaz**:

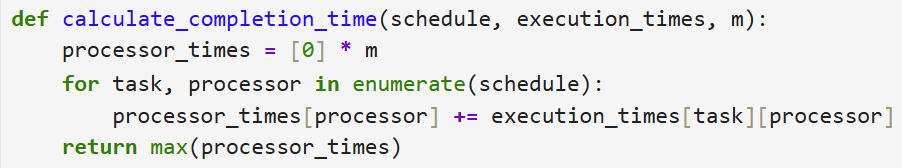
* Funkcija dodele koja zadatke dodeljuje procesorima

Cilj:

* Minimizovati vreme završetka:

Ispisujemo sve rasporede za m(broj procesora) i T(broj zadataka)

****

****

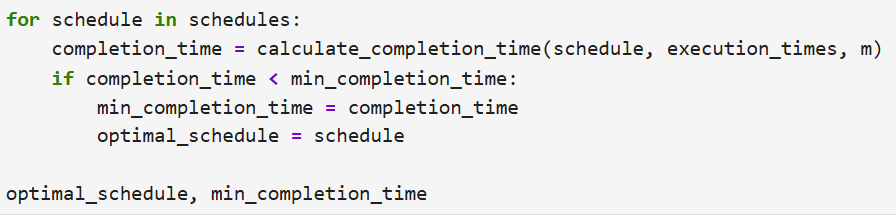
Ovako za svaki schedule racunamo vreme izvrsavanja. Uzimamo maximum jer je to vreme za koje ce oba taska da se zavrse..

1. **Metode resavanja**

**3.1 Brute-force pristup**

Brute-force metoda generiše sve moguće dodele zadataka procesorima i bira onu koja daje minimalni makespan. Ovaj pristup je računski neizvodljiv za veće instance zbog eksponencijalnog rasta mogućih rešenja.

Vreme izvršavanja: (eksponencijalno)

****

**3.2 Local Search algoritam**

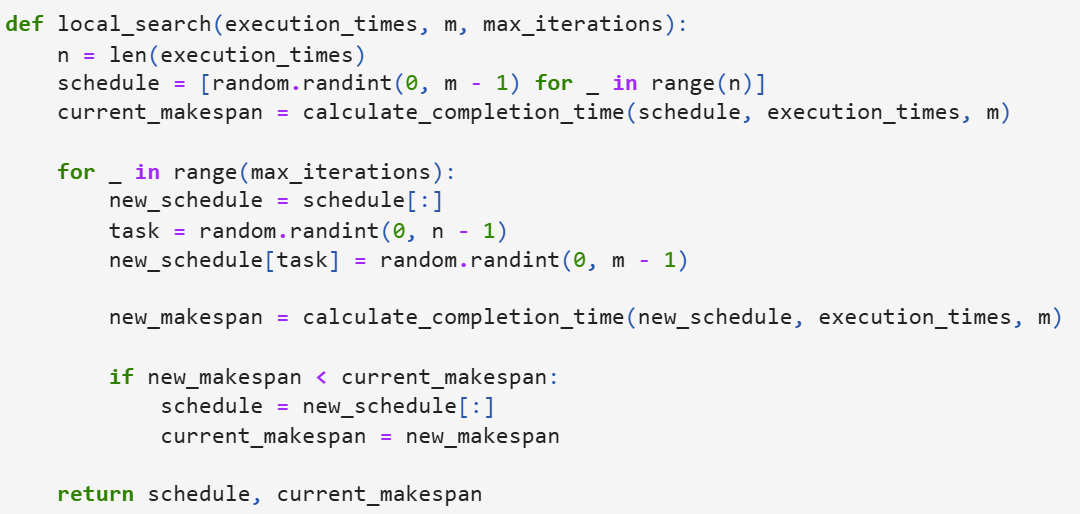
Local Search je heuristički metod koji poboljšava trenutno rešenje malim promenama, tražeći bolju raspodelu zadataka. Koristi se strategija poboljšanja susednih rešenja dok se ne dođe do lokalnog minimuma.

Prednosti:

* Brži od brute-force metode
* Daje dobra rešenja u razumnom vremenu

Nedostaci:

* Može zapeti u lokalnom optimumu

****

**3.3 Simulated Annealing (SA)**

Simulated Annealing je probabilistički algoritam koji simulira proces hlađenja metala. U početku dozvoljava lošija rešenja kako bi izbegao lokalne minimume, ali vremenom smanjuje verovatnoću prihvatanja lošijih rešenja.

**Prednosti**:

* Bolji od Local Search-a jer može izbeći lokalne minimume
* Fleksibilan i primenjiv na razne optimizacione probleme

**Nedostaci**:

* Potrebno pažljivo podešavanje parametara (temperatura, stopa hlađenja)

****

**3.4 Genetski algoritam (GA)**

Genetski algoritam je inspirisan prirodnom evolucijom i koristi operacije selekcije, ukrštanja i mutacije da iterativno poboljša rešenja.

Koraci:

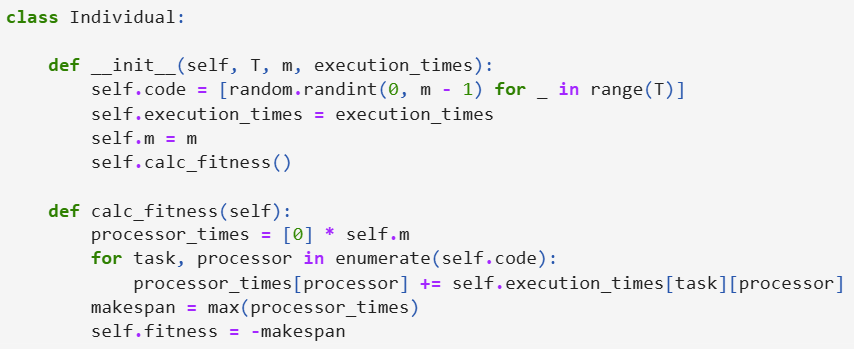
1. Generiše se inicijalna populacija mogućih rešenja.
2. Rešenja se ocenjuju pomoću fitness funkcije (makespan).
3. Selekcija najboljih rešenja.
4. Primena operatora ukrštanja i mutacije.
5. Ponavljanje procesa dok se ne postigne zadovoljavajuće rešenje.

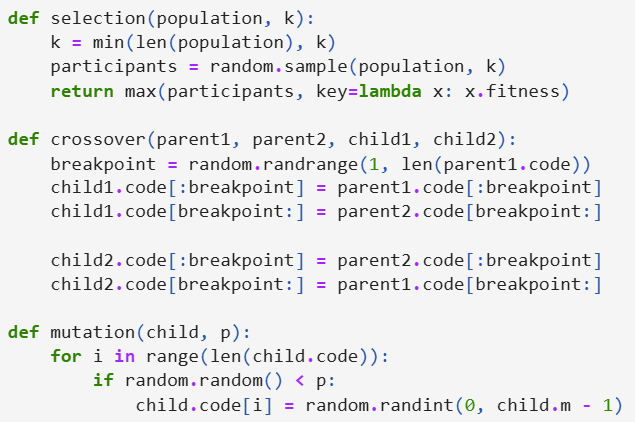
**Prednosti**:

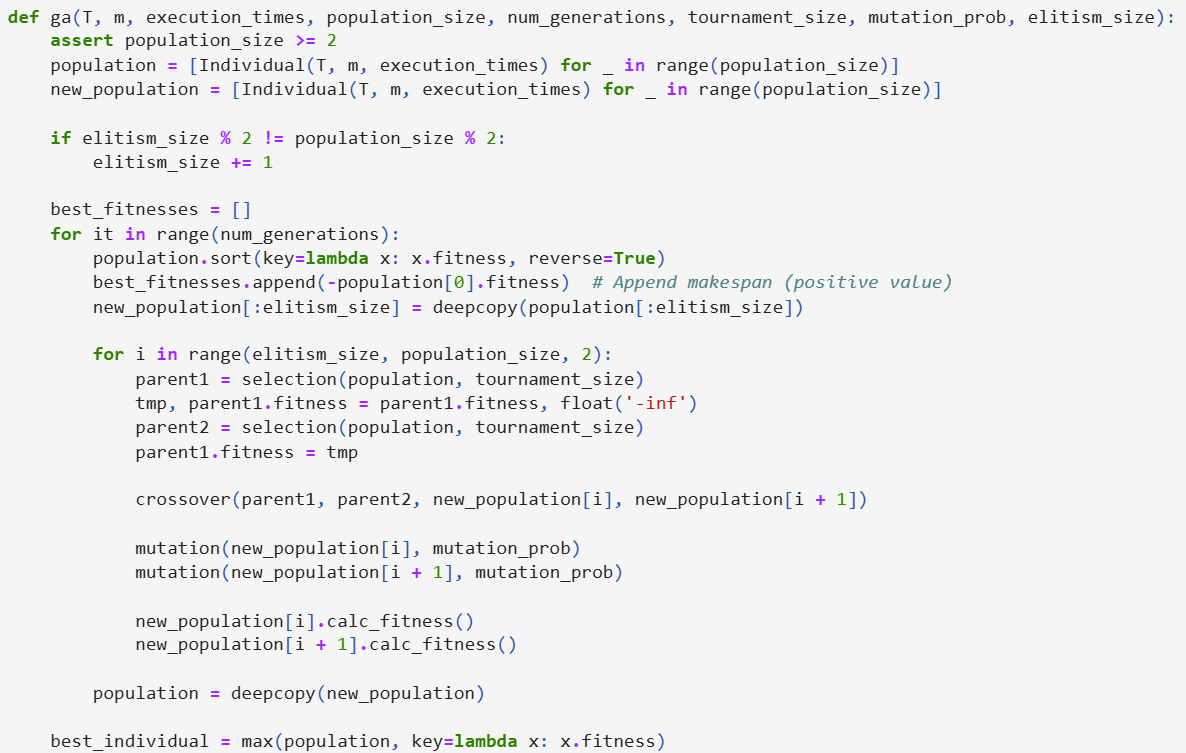
* Dobar balans između eksploatacije i istraživanja rešenja
* Može pronaći blizu optimalna rešenja za velike instance

**Nedostaci**:

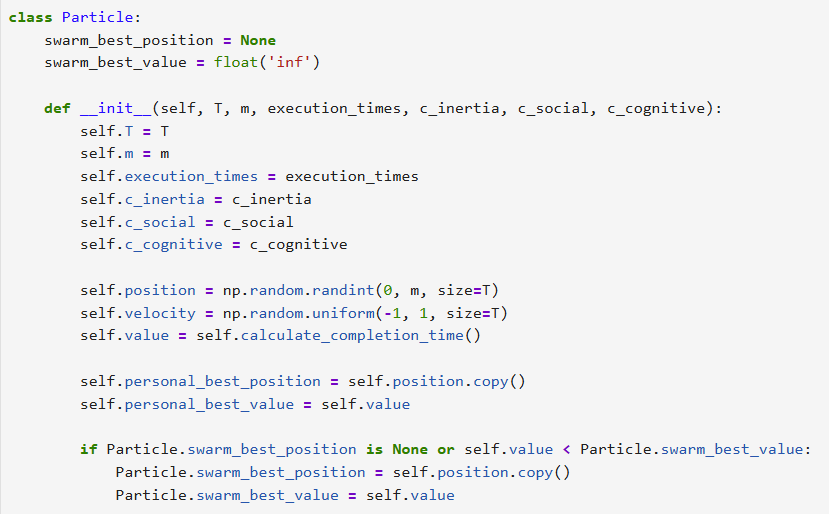
* Potreban veći broj iteracija
* Podešavanje parametara (populacija, mutacija, ukrštanje)

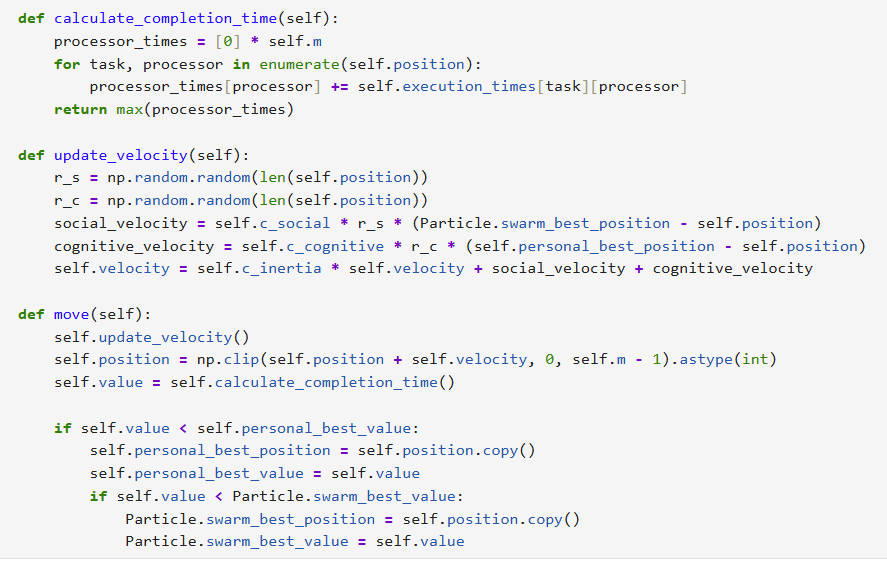
****

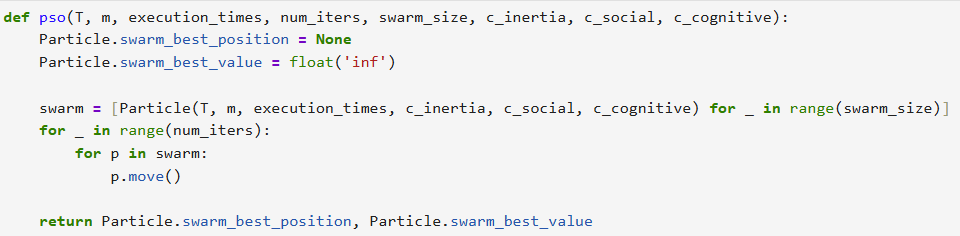
****

****

**Optimizacija rojem čestica**

****

****

****

**4. Eksperimentalni rezultati**

Izvršili smo eksperimentalnu evaluaciju svih metoda na različitim instancama problema. Rezultati pokazuju sledeće:

|  |  |  |
| --- | --- | --- |
| Metoda | Kvalitet resenja | Brzina izvrsavanja |
| Brute-force | Optimalno | Veoma sporo |
| Local Search | Dobra resenja | Brzo |
| Simulated Annealing | Vrlo brza resenja | Srednje brzo |
| Genetski algoritam | Najbolja resenja | Sporije od SA, ali efikasno |

Zaključak je da je genetski algoritam najefikasniji metod za veće instance problema, dok su Local Search i Simulated Annealing dobri kompromisi između kvaliteta rešenja i brzine izvršavanja.

**Zaključak:**

Minimum Multiprocessor Scheduling je složen optimizacioni problem za koji smo isprobali više pristupa. Brute-force je dao optimalna rešenja, ali nije praktičan za veće instance. Local Search je efikasan, ali može zapeti u lokalnim optimumima. Simulated Annealing je pokazao bolje performanse zahvaljujući probabilističkom izlasku iz lokalnih minimuma, dok je Genetski algoritam dao najbolji balans između pretrage i eksploatacije rešenja.

Za buduće radove, moguće je istražiti hibridne metode koje kombinuju GA sa SA ili Local Search, kao i primenu dubokog učenja za predikciju optimalnih strategija raspodele.

**Reference**

[1] Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman. [2] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). *Optimization by Simulated Annealing*. Science. [3] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems.* University of Michigan Press.