



KOMBINATORNI ALGORITMI- PROBLEM TRGOVAČKOG PUTNIKA I RASPOREĐIVANJE PROCESA PROCESORA

COMBINATORIAL ALGORITHMS- TRAVELLING SALESMAN PROBLEM AND SCHEDULING CPU PROCESSES

Nevena Dovičin, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Problem trgovačkog putnika spada u veliku klasu problema koji se nazivaju kombinatorni problemi optimizacije. Ovo je jedan od najviše proučavanih problema optimizacije, a njegova rešenja i primene će biti opisane u ovom radu. S druge strane raspoređivanje procesa u procesoru je zadatak u kom se bira proces iz reda čekanja kom se dodeljuje procesor nakon toga (od strane dispečera). Postoji više različitih algoritama za biranje procesa koji će se izvršavati, od kojih će najznačajniji biti opisani.

Ključne reči: kombinatorni algoritmi, problem trgovačkog putnika, raspoređivanje procesa

Abstract – The travelling salesman problem is one of the problems called combinatorial optimization problems. This is one of the most studied optimization problems, and its solutions and applications will be described in this paper. On the other hand, scheduling a process in a processor is the task in which the process is selected from the queue to which processor is assigned after that (by dispatcher). There are several different algorithms for selecting the process to be executed, and most significant will be described.

Keywords: combinatorial algorithms, travelling salesman problem, scheduling processes

1. UVOD

Razvojem računarstva i informatike i povećanjem procesne moći računara, ljudi su počeli rešavati kompleksne probleme koji su do tada bili nerešivi. S obzirom da su danas računari ekstremno brzi, nije retka situacija da se njihovom upotrebom u sekundi mogu istražiti milioni potencijalnih rešenja, i time vrlo brzo pronaći ono najbolje.

Postoji čitav niz problema koje srećemo u svakodnevnom životu, a jedan od primera je problem trgovačkog putnika. U ovom radu biće opisani neki od algoritama koji rešavaju ovaj problem.

U informatici, raspoređivanje je metoda po kojoj se procesima i tokovima podataka daje pristup vremenskim resursima. Potreba za algoritmima raspoređivanja procesa je proizašla iz potrebe modernih sistema da u isto vreme obavljaju više poslova.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinac, red. prof.

U tipičnim slučajevima procesor i periferije nisu ravnomerno iskorišćeni. Takođe u poslednjim poglavljima biće opisani algoritmi koji se bave problemima rasporedom procesa u procesoru.

2. ISTORIJA PROBLEMA TRGOVAČKOG PUTNIKA

Problem trgovačkog putnika (TSP) je dugo privlačio pažnje sa svih strana i jedan je od najintenzivnije proučavanih, ako ne i najproučavaniji problem iz područja matematike. Tačan izvor problema je nepoznat, kao i informacija ko je ovaj problem imenovao tako. Problem se prvi put spominje 1832. godine u priručniku jednog nemačkog trgovačkog putnika. Mnoge trgovce tog doba je zanimalo strateško planiranje što ekonomičnije rute do područja svojih klijenata. Zvanično je prvi put formulisano 1930. godine.

3. DEFINICIJA I PREDSTAVLJANJE

Problem trgovačkog putnika je predstavnik velike klase problema poznatih kao kombinatorni problemi optimizacije. U regularnom obliku TSP-a, mapa gradova je dodeljena prodavcu i on mora da poseti sve gradove samo jednom kako bi završio obilazak tako da je dužina turneje najkraća među svim mogućim putovanjima za ovu mapu. S obzirom na skup gradova i troškova putovanja između svakog mogućeg para, cilj je da se pronađe najbolji mogući način posete svih gradovima i vraćanje na početnu tačku, s najmanjim mogućim troškovima putovanja.

4. EVOLUTIVNI ALGORITMI

U prirodi preživljavaju jedinke koje se prilagođavaju novim uslovima i okolnostima. Jedinke sa karakteristikama kao što su bolje prilagođavanje okolini imaju veće šanse za preživljavanje i povećanje svoje vrste, dok jedinke koje se sporije prilagođavaju izumiru, uzrokujući tako izumiranje čitavih vrsta. Na taj način se u prirodi sprovodi prirodna selekcija i osigurava da geni bolje prilagođenih jedinki imaju veće šanse za preživljavanje od onih koje su se slabije prilagodile. Upotrebom tog osnovnog biološkog modela, nastali su evolutivni algoritmi (*Evolutionary Algorithms*) čiji su deo genetski algoritmi. Kada kažemo evolutivni algoritmi mislimo na one algoritme koju podrazumevaju procese biološke evolucije. Ove metode pretraživanja su našle inspiraciju u prirodi i biološkom svetu. U svakoj iteraciji ovakvog algoritma postoji konkurentni izbor među svim

rešenjima u populaciji; rezultat je najbolji od njih, a loša rešenja se brišu iz populacije. Vršiti se rekombinacija i formira se novo rešenje koje je bolje od prethodnih. Rekombinacija i mutacija se koriste u evoluciji populacije.

5. GENETSKI ALGORITMI

Genetski algoritmi (*Genetic Algorithms, GA*) su najpopularnija vrsta evolutivnih algoritama, a predstavljaju model optimizacije čije ponašanje potiče iz procesa evolucije koji se odvija u prirodi. Osnovni cilj genetskih algoritama je pronalaženje rešenja nekog problema koje tradicionalne determinističke metode ne mogu rešiti. Za razliku od većine determinističkih algoritama, genetski algoritmi pretragu ne započinju od jedne tačke nego od celog niza potencijalnih rešenja. U ovim algoritmima razlikujemo ukrštanje i mutaciju. Ukrštanje se često naziva i rekombinacija. Ovo je proces kojim iz dve jedinke roditelja nastaje nova jedinka dete. Najvažnije karakteristika ukrštanja je da deca nasleđuju osobine roditelja. Novonastalo dete poseduje genetski materijal oba roditelja i ima jednak broj gena kao i oni. Mutacija je slučajna promena jednog ili više gena u cilju dobijanja genetske raznolikosti.

5.1. Rešavanje TSP-a korišćenjem genetskih algoritama

Postoji veliki broj algoritama i mogućih rešenja, a GA se smatraju još boljim pristupom, naročito kada su praćeni pažljivo osmišljenim genetskim operaterima. Garantuju veliki prostor rešenja za najbolji odgovor, a operateri im pomažu da postanu brži. Poslednjih godina je ovo privuklo veliku pažnju. Spovedene su mnoge studije i istraživači pokušavaju da doprinesu različitim delovima procesa razmišljanja i rešavanja.

6. OPTIMIZACIJA KOLONIJOM MRAVA

Algoritam optimizacije kolonijom mrava proučava veštačke sisteme koji su našli inspiraciju po ugledu na ponašanje pravih mravljih kolonija, koji se koriste za rešavanje optimizacionih problema, najčešće pronalaženja najkraćeg puta u zadatom grafu. Ovo je heuristički metod koji je inspirisan stvarnim kretanjem i ponašanjem mrava u pronalaženju najkraće putanje između gnezda i hrane.

Ovo se postiže supstancom koja se naziva feromon koja pokazuje trag mrava. U svom traganju, mravi koriste heurističke informacije koje je već njihovo znanje odakle miris hrane dolazi.

6.1. Ponašanje veštačkih mrava

Za objašnjenje ponašanje veštačkih mrava i izbor nekog od mogućih puteva koristi se jednačina :

$$p_{ij}^k(t) = \left\{ \begin{array}{l} \frac{[\tau_{ij}(t)]^\alpha * [\mu_{ij}]^\beta}{\sum_{k \in \text{dopusteno}_k} [\tau_{ik}(t)]^\alpha * [\mu_{ik}]^\beta} \text{ ako } (j \in \text{dopusteno}_k) \\ 0 \text{ inace} \end{array} \right\}$$

P predstavlja koeficijent isparavanja feromona, τ predstavlja feromon.

6.2. Mravlji algoritam za rešavanje TSP-a

Veštački mrav je agent koji se kreće po grafu TSP. Njegova putna strategija se zasniva na probablističkoj funkciji koja uvažava dve stvari. Prvo, agent beleži puteve kojima je prošao idući od grada do grada i sabira njihove dužine. Drugo, on oseća miris feromona koje su iza sebe ostavili drugi agenti. Svaki agent menja svoje okruženje na dva različita načina:

- Lokalna promena traga: Dok se kreće između gradova, mrav menja količinu feromona na svakoj pređenoj deonici između dva grada.

- Globalna promena traga: Kada svi mravi završe putovanje, mrav koji je prešao najkraći put dodaje količinu feromona na svim deonicama kojima je prošao. Svrha lokalnog pravila promene traga je da se izbegne da svaki mrav izabere deonice na kojima je položeno mnogo feromona.

Na taj način se povećava istraživačka sposobnost mrava, a u suštinskom smislu izbegava da se proces traženja najboljeg puta okonča u nekom lokalnom minimumu. Globalno pravilo promene traga daje jače osveženje feromonskog traga na najkraćem pređenom putu, odnosno svaka njegova deonica dobija jači miris pri sledećim iteracijama procesa.

7. OPTIMIZACIJA ROJEM ČESTICA

Optimizacija rojem čestica pripada skupu algoritama koji koriste ponašanje inspirisano jatom ptica, riba ili pčela. To je u suštini stohastički algoritam čiji su temelj sociološko-psihološki principi i pružaju uvid u sociološka ponašanja te pomoću njih doprinose rešavanju inženjerskih problema i aplikacija.

7.1. Rešavanje problema TSP-a korišćenjem optimizacije roja čestica

Smatra se da je jedan od odličnih algoritama koji se zasniva na PSO za rešavanje TSP objasnjen u *Pang et al 2004*. U ovom radu je predožen algoritam koji koristi *fuzzi* matrice za vektore brzine i položaja. Pored toga, oni koriste operatore *fuzzi* množenja i sabiranja za formule brzine i položaja.

Pomenuti PSO u prethodnim odeljcima modifikovan je na algoritam koji radi na osnovu *fuzzi* sredstava. U svakoj iteraciji položaj generisanog rešenja se koristi da bi se odredio trošak (vrednost) pojedinca. Ovaj trošak će biti korišćen za ažuriranje najboljeg lokalnog položaja.

8. OPTIMIZACIJA KOLONIJOM PČELA

Kao i svi ostali algoritmi inspirisani prirodom, i ovi algoritmi su zasnovani na kolektivnoj inteligenciji, kao što je PSO preuzet iz života ptica, ACO zasnovan na društvenom životu kolonije mrava -ovo je još jedna vrsta sistema veštačke inteligencije koji može biti koristan u rešavanju mnogih problema i aplikacija što se tiče inženjeringa, upravljanja, kontrole i računskih problema.

Algoritam inspirisan pčelinjim kolonijama u prirodi je jedan od interesantnih metaheurističkih algoritama koji predstavljaju drugi pravac u oblasti inteligencije roja. Zasnovana je na biološkoj potrebi jedinki da se udruže radi ostvarenja zajedničkih ciljeva.

8.1. BCO algoritam

Osnovna ideja je konstruisanje sistema od više agenata što znači da na početku svaka pčela pripada koloniji koja traži hranu pojedinačno.

Ovakva kolonija (kolonija veštačkih pčela) se sastoji od manjeg broja individua – što ne narušava osnovne koncepte modelovanja zasnovane na ponašanju pčela u prirodi. Kada pčela iz kolonije pronađe hranu, ona plesom obavestava druge pčele. Ostale pčele sakupljaju i prenose hranu u košnicu.

U ovom algoritmu postoje dve faze – let unapred i let unazad. Ove dve faze se smenjuju NC puta u toku jedne iteracije. Smenjuju se dok se ne izgenerišu konstruktivna rešenja ili dok se početna rešenja ne poprave. Tokom leta unapred, sve pčele se bave istraživanjem okolnog prostora.

Po analogiji sa prirodnim procesima, (parcijalno) rešenje se može poistovetiti sa količinom sakupljenog nektara i/ili sa udaljenosti košnice od izvora hrane, a proces sakupljanja nektara u prirodi odgovara fazi leta unapred u BCO algoritmu.

Nakon generisanja svojih (parcijalnih) rešenja veštačke pčele započinju drugu fazu, let unazad. U fazi leta unazad veštačke pčele razmenjuju informacije o kvalitetu svojih pronađenih rešenja.

9. PRIMENA TSP-A I VEZE SA DRUGIM PROBLEMIMA

Osim što predstavlja veliki izazov za rešavanje, problem trgovačkog putnika je široko primenjiv na različite probleme usmeravanja i upoređivanja. Direktno primene pronađene su u problemima bušenja štamparskih ploča, postavljanje lopatica na turbine aviona, analize strukture jednog kristala, izbora naloga u skladištima, planiranje misije u vojsci.

Ovaj problem ima mnogo primena, a ovde su navedene samo neke od njih. Postoji još niz primera u kojima se problem trgovačkog putnika pojavljuje kao potproblem ili se drugi problemi mogu svesti na njega.

10. RASPOREĐIVANJE PROCESA PROCESORA

Pored problema trgovačkog putnika, algoritmi koji su takođe bili zanimljivi i proučavani su algoritmi raspoređivanja procesa u procesoru. Raspoređivanje procesa procesora se bavi problemom odlučivanja- u smislu koji procesi u redu čekanja će biti dodeljeni procesoru. Postoji mnogo različitih algoritama za raspoređivanje procesora. U ovom odeljku SU opisani neki od njih.

10.1. *First come, first served* algoritam

Jedan od najjednostavnijih algoritama raspoređivanja je upravo ovaj, *first-come, first-served* algoritam raspoređivanja (u bukvalnom prevodu *prvi došao, prvi uslužen*). Sa ovom shemom, proces koji procesor prvo zahteva, prvo je i dodeljen procesoru.

Implementacija sa FCFS algoritmom se lako upravlja pomoću FIFO reda. Kada proces uđe u red čekanja, njegova PCB je povezana na rep reda. Pokrenut proces se zatim uklanja iz reda čekanja. Kod za FCFS algoritam raspoređivanja je jednostavan za pisanje i razumevanje.

Sa druge strane, prosečno vreme čekanja kod FCFS je često veliko.

10.2. *Shortest job first* algoritam

Drugačiji pristup raspoređivanju rada procesora od FCFS, je tzv. *shortest-job-first* (u bukvalnom prevodu-prvo najkraći posao, u daljem tekstu SJF). Ovaj algoritam povezuje sa svakim procesom dužinu narednog procesa procesora. Kada je procesor slobodan, dodeljen je procesu koji ima najkraće vreme izvršavanja. Ukoliko postoje dva procesa sa istim vremenom izvršavanja, FCFS algoritam je tu da odluči koji će se izvršiti sledeći.

10.3. Raspoređivanje s prioritetom

SJF algoritam je ustvari specijalan slučaj tzv. *priority scheduling* (raspoređivanje sa prioritetom). Svakom procesu je zadat prioritet, a procesor je dodeljen procesu sa najvećim prioritetom. Procesi sa jednakim prioritetom su dodeljeni po FCFS algoritmu raspoređivanja. SJF spada u algoritme raspoređivanja sa prioritetom, gde je prioritet inverzija predviđenog sledećeg vremena izvršavanja. Što je veće vreme izvršavanja, manji je prioritet i obrnuto. U ovom tipu raspoređivanja kao što je već navedeno, postoje procesi niskog i visokog prioriteta. To obično bude opseg brojeva.

10.4. Round robin raspoređivanje

Ovaj algoritam je dizajniran za *time-sharing systems*. Vrlo je sličan FCFS algoritmu, ali je *preemption* (pretpražnjenje) dodata da omogućiti da se sistem premešta sa procesa na proces. U ovom algoritmu postoji nešto specifično, a to je mala jedinica vremena koja je definisana. Zove se time *quantum* ili *time-slice*. Obično traje od 10 do 100 milisekundi. Spreman red se tretira kao kružni red. *CPU scheduler* ide oko spremnog reda i dodeljuje procesor svakom procesu za vremenski interval do 1 vremenski kvant. Da bismo implementirali RR planiranje, ponovo tretiramo red spremnosti kao FIFO red čekanja procesa. Novi procesi se dodaju u rep spremnog reda. Raspoređivač procesa u procesoru bira prvi proces iz reda spremnosti, postavlja tajmer da se prekine nakon 1 vremenskog kvanta, i raspodeli proces.

11. ZAKLJUČAK

Kao što je već navedeno, TSP je jedan od najproučavanijih problema u istoriji problema optimizacije. Naime, zadatak ne zvuci teško. Najjednostavniji postupak bi generalno bio da se korišćenjem iscrpne pretrage prebroje sve moguće putanje i na kraju da se odabere najoptimalnija. Moglo bi se reći da se vrlo trivijalnim putem stiže do rešenja. Takođe postoji metoda čija je ideja da se krene iz bilo kog grada i da se bira uvek najbliži – što nije najpouzdaniji algoritam. Pored toga, ovde su opisani evolucionarni algoritmi, algoritmi inspirisani prirodom i slično. Ove vrste algoritama daju nešto malo manje lošije rešenje od optimalnog. Ali nijedno od ovih rešenja ne predstavlja univerzalno koje garantuje najbolji rezultat. Razlog je složenost algoritma, jer povećanjem broja gradova raste ukupan broj putanja ogromnom brzinom. Zato se ovaj zadatak svrstava u nerešive u opštem smislu takoreći, odnosno u zadatke NP klase. Ovaj problem ima mnogo primena, a ovde su navedene neke od njih. Postoji

još niz primerau kojima se problem trgovačkog putnika pojavljuje kao potproblem ili se drugi problemi mogu svesti na njega. S druge strane raspoređivanje procesa u procesoru je zadatak u kom se bira proces iz reda čekanja kom se dodeljuje procesor nakon toga (od strane dispečera).

FCFS je najjednostavniji algoritam za raspoređivanje, ali može da uzrokuje da kratki procesi čekaju duge procese da se izvrše. Za SJF algoritam bi se moglo reći da je optimalan, međutim njegova implementacija je dosta teška jer treba pretpostaviti i predvideti dužinu sledećeg procesa.

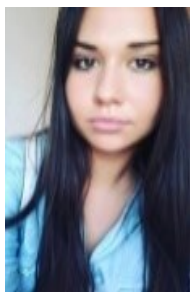
SJF algoritam je poseban slučaj algoritama sa prioritetom, koji alocira procesoru proces sa najvećim prioritetom. RR raspoređivanje je pogodno za interaktivne sisteme, gde postoji vremenska jedinica koja ima najveći značaj u ovom algoritmu. Postoji širok opseg algoritama za raspoređivanje, a ovo su samo neke od njih. Analitičke metode koriste matematičke algoritme da odrede učinak algoritma.

Simulacione metode određuju performanse tako što naprave imitaciju algoritma po uzorku procesa i izračunavaju i programiraju performanse. Simulacijom se najbolje aproksimiraju performanse stvarnog sistema. Jedina pouzdana tehnika za procenu algoritma raspoređivanja je da se primeni algoritam na stvarnom sistemu i da se nadgleda njegovo funkcionisanje u realnom okruženju.

LITERATURA

- [1] U-Turning Ant Colony Algorithm for Solving Symmetric Traveling Salesman Problem- Saman M Al-Mufti
- [2] Programming of processes, Scheduling Algorithms - Uldis Sukovskis
- [3] Wikipedia, Regular expression, 2019, Wikipedia
- [4] Mohammad Reza Bonyadi, Mostafa Rahimi Azghadi, and Hamed Shah-Hosseini (2008). Population-Based Optimization Algorithms for Solving the Travelling Salesman Problem,
- [5] Travelling Salesman Problem –Jose Guevara Ilatoma

Kratka biografija:



Nevena Dovičin rođena je 27.09. 1994. godine u Novom Sadu. Završila je Gimnaziju "20. oktobar" u Bačkoj Palanci, a osnovne četvorogodišnje studije na odseku elektrotehnika, smeru računarstvo i automatika 2017. godine. Ispunila je sve obaveze i položila je sve ispite predviđene studijskim programom.