

Parallel beam search

za problem trgovačkog putnika

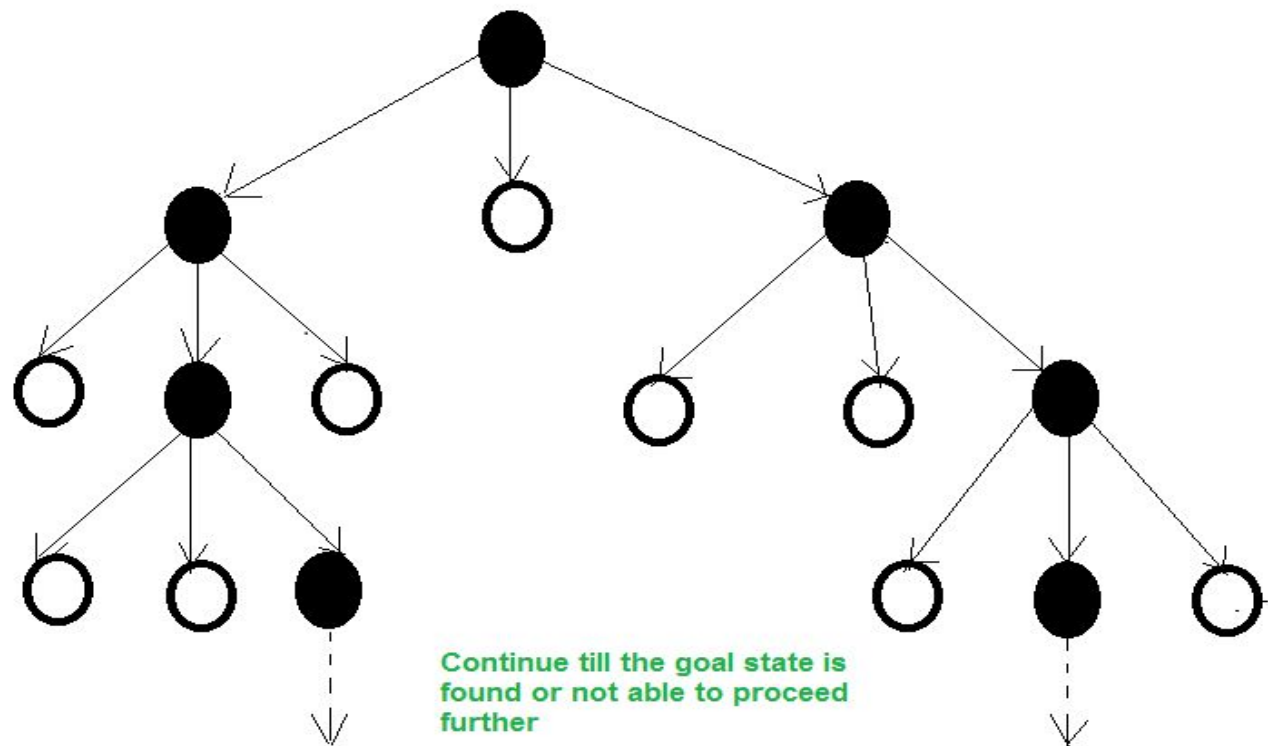
Nikola Markov
215/2018

Matematički fakultet
u Beogradu

Uvod

- Problem trgovačkog putnika
 - NP težak problem
 - najkraći put koji obilazi sve čvorove i vraća se u početni
- Parallel beam search
 - Algoritam pretrage stabla nivo po nivo
 - Slično BFS algoritmu





Pseudokod

Input: Set of cities and their distances

Output: Shortest found path

```
function parallel_beam_search(distnaces, beam_width, num_processes):  
    beam = initial beam with starting node  
    for 1 to num_nodes:  
        create num_processes thread  
        for each thread start search  
  
        wait for all threads  
        Sort beam  
        Select beam_width nodes for next iteration  
  
    return best beam
```

Rezultati

Hardver:
i5-1135G7 2.40GHz
8GB RAM
Linux
Jupyter notebook

					Beam width	
att48	Scale	Optimal solution	Best	100	1000	5000
Tabu search	48	33522	34198			
GAs	48	33522	34572			
PSO	48	33522	34759			
ACO	48	33522	34357			
PBS	48	33522	36184	42979	36184	37381

					Beam width	
berlin52	Scale	Optimal solution	Best	100	1000	5000
Tabu search	52	7542	7976			
GAs	52	7542	8201			
PSO	52	7542	8197			
ACO	52	7542	7647			
PBS	52	7542	8719	9851	8719	9709

					Beam width	
pr76	Scale	Optimal solution	Best	100	1000	5000
Tabu search	76	108159	110941			
GAs	76	108159	115329			
PSO	76	108159	118038			
ACO	76	108159	110517			
PBS	76	108159	145599	151976	145599	150913

					Beam width	
eil101	Scale	Optimal solution	Best	100	1000	5000
Tabu search	101	629	667			
GAs	101	629	682			
PSO	101	629	687			
ACO	101	629	649			
PBS	101	629	809	819	809	817

att48	Time (s)	Best	Optimal
Ant system	66,425	35250	33522
PBS	2,274	36184	33522

berlin52	Time (s)	Best	Optimal
Ant system	72,961	7681	7542
PBS	2,663	8719	7542

pr76	Time (s)	Best	Optimal
Ant system	116,907	118693	108159
PBS	6,035	145599	108159

st70	Time (s)	Best	Optimal
Ant system	104,988	721	675
PBS	33,403	871	675

Zaključak

Iz dobijenih rezultata može se zaključiti da algoritam Beam search nije konkurentan vodećim algoritmima koji se bave ovim problemom.

Razlog tome je jako brzo konvergiranje ka rešenju, jer se odseca veliki broj mogućnosti koje mogu biti pregledane.

Kako bi se povećala efikasnost može se uzeti još neka mera koja će sa nekom određenom verovatnoćom birati čvorove.

