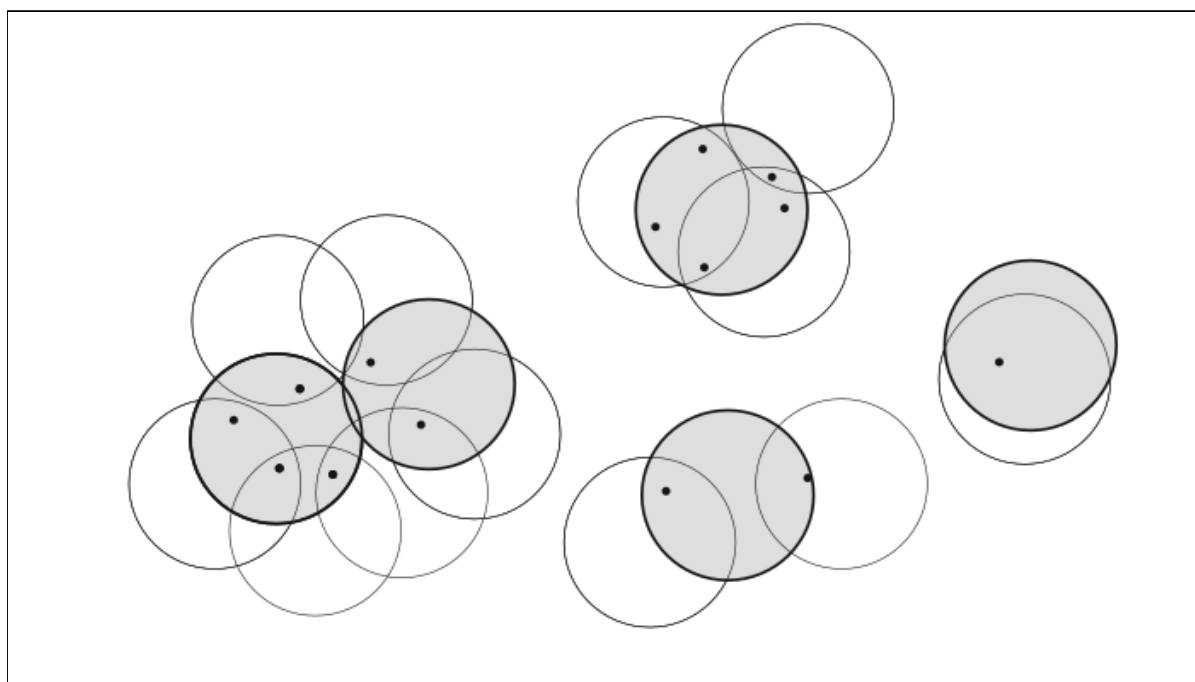


Algoritmi za pokrivanje skupa tačaka jediničnim krugovima

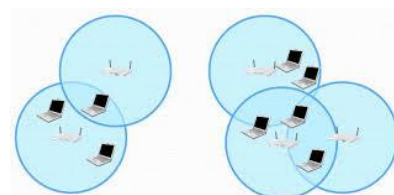
Ana Jakovljević 1025/2020



Opis problema

Problem pokrivanja jediničnim krugovima (*engl. Unit Disk Cover problem*) predstavlja problem izračunavanja minimalnog skupa jediničnih krugova potrebnih za pokrivanje svih tačaka zadatog skupa, pri čemu jedinični krugovi predstavljaju krugove poluprečnika jedan. Problem je *NP težak*, a u ovom radu su implementirana četiri aproksimaciona algoritma različitih vremenskih složenosti i faktora aproksimacije za primenu na tačke zadate u ravni.

Primena: Najpopularnija primena je lociranje najmanjeg broja neophodnih objekata tako da se svi potencijalni korisnici nalaze unutar razumno male udaljenosti od jednog od objekata.



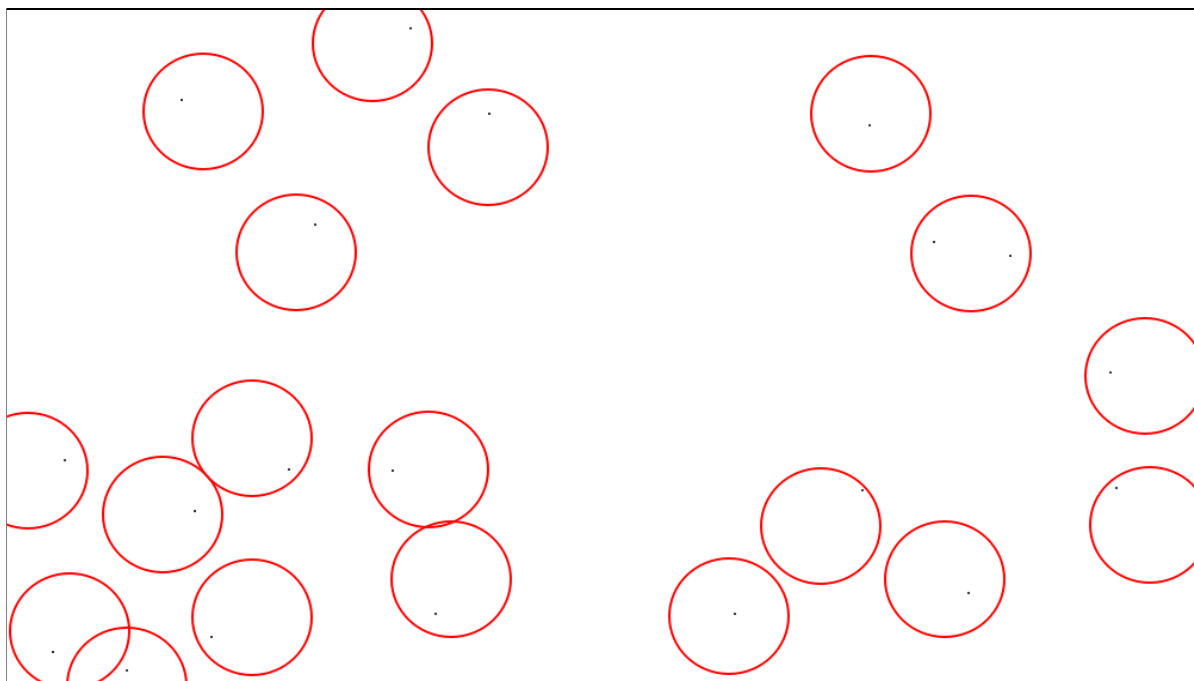
Ulaz: skup od n tačaka ravni

Izlaz: skup tačaka koje predstavljaju centre krugova fiksiranog poluprečnika za pokrivanje ulaznog skupa od n tačaka

Naivno rešenje problema

Ograničenja za pozicije krugova ne postoje, pa kandidati za centre krugova mogu biti bilo koji parovi realnih brojeva. Zbog toga je iscrpna pretraga za rešavanje ovog problema nemoguća i kao prvi naivni pristup korišćeno je slučajno generisanje jednog po jednog kruga uz proveravanje da li svaki novi generisani krug pokriva neku od do tada nepokrivenih tačaka.

Međutim, besmisleno je proveravati krugove koji se ne nalaze u okolini bar neke tačke datog skupa, a ovaj algoritam gubi značajno vreme na proveru takvih krugova. Iz tog razloga implementiran je "manje" naivan algoritam koji se takođe zasniva na slučajnom generisanju krugova, ali se slučajnost odnosi na izbor pozicije u okolini neke od tačaka koju treba pokriti.



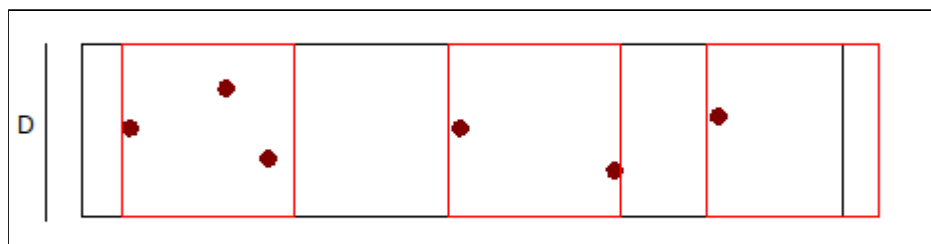
Vizualizacija naivnog algoritma

Glavni problem ovih algoritama je nepostojanje teorijskog dokaza o faktoru aproksimacije. U nastavku će biti predstavljeni algoritmi sistematičnog pristupa za koje postoje takvi teorijski dokazi. Problem će biti uopšten i algoritmi će biti opisani pretpostavljajući opštu vrednost poluprečnika R .

Algoritam1: Gonzalez [1]

Autor u ovom radu opisuje algoritam za rešavanje problema pokrivanja skupa tačaka kvadratima fiksirane veličine, stranica paralelnih osama. Rešenje ovog problema će biti iskorišćeno za dobijanje rešenja početnog problema pokrivanja krugovima.

Pre svega, predstavljen je jednostavniji problem ploča (*engl. slab problem*) u kom su sve tačke locirane unutar pravougaonika visine D , stranica paralelnih osama. Algoritam za problem ploča iterativno pokriva najlevlju nepokrivenu tačku kvadratom dimenzija $D \times D$ i na taj način kreira optimalno rešenje.

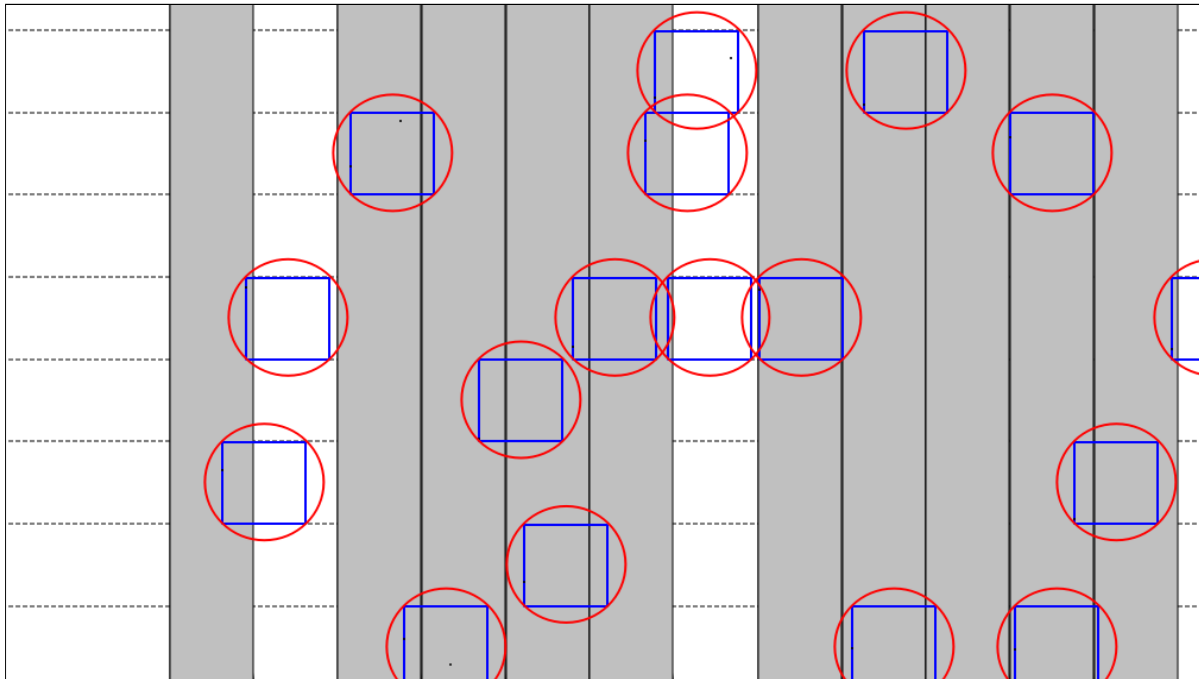


Rešavanje problema ploča

Problem ploča se dalje uopštava na problem pokrivanja šireg skupa tačaka, a ideja je da se prostor podeli na ploče visine D i da se svaka ploča nezavisno obradi datim algoritmom. Na osnovu toga, rešenje početnog problema pokrivanja krugovima se dobija postavljanjem dužine ivice kvadrata na $R\sqrt{2}$, rešavanjem međuproblema sa kvadratima i konačno, opisivanjem krugova poluprečnika R oko svakog kvadrata koji čini rešenje međuproblema. Navedeni algoritam ima vremensku složenost izvršavanja $O(n \cdot \log|OPT|)$, dok faktor aproksimacije iznosi 4.

Vizuealizacija:

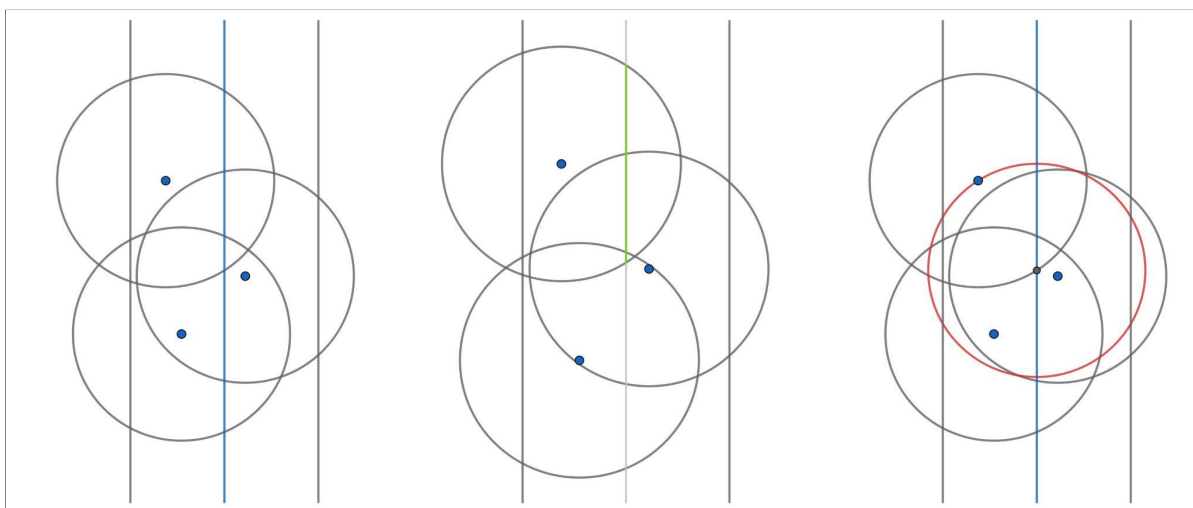
Vizuealizacija je izvršena u skladu sa navedenim načinom rada algoritma. Tačke su prvo podeljene po horizontalnim trakama, a zatim su parne i neparne horizontalne trake nezavisno obrađene na isti način. Tačke se, za obe grupe horizontalnih traka, dalje dele po vertikalnim trakama prikazanim sivo obojenim pravougaonicima. Algoritam obrađuje dve po dve susedne trake i iscrta se postupak pokrivanja tačaka kvadratima. Na kraju se opisuju krugovi oko svakog kvadrata koji čini rešenje.



Vizuelizacija algoritma G1991

Algoritam2: Liu Lu [2]

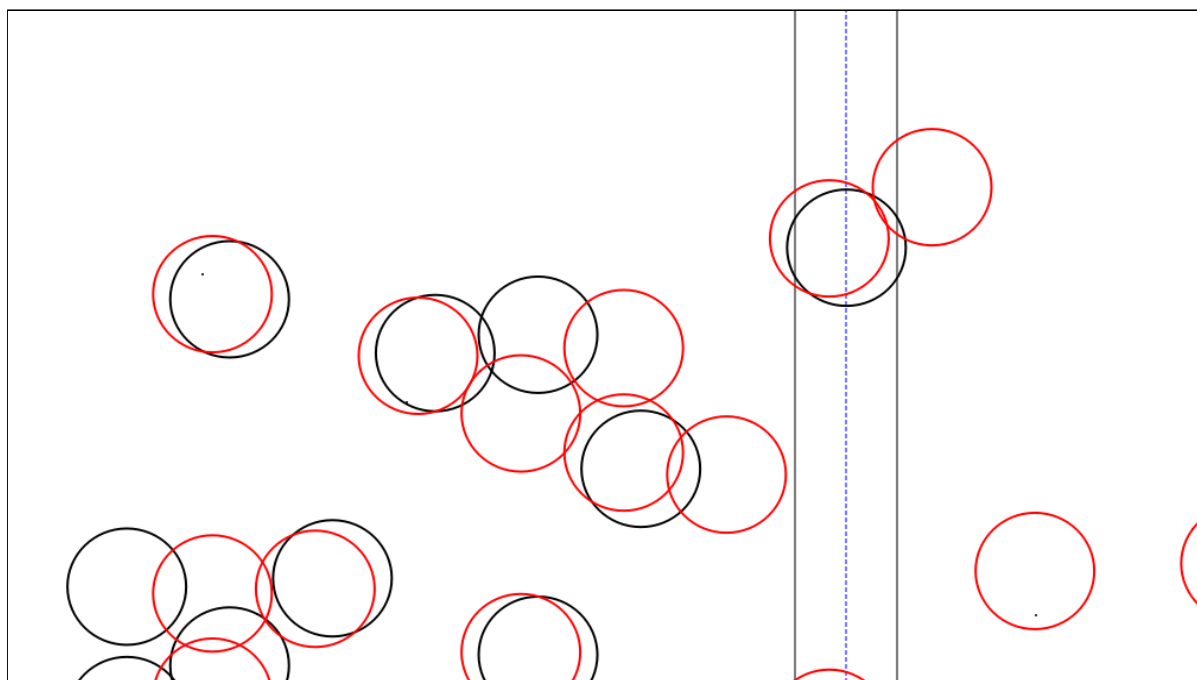
Autori su u radu predstavili $\frac{25}{6}$ - aproksimacioni algoritam za pronalaženje pokrivača skupa tačaka koji se izvršava u vremenu $O(n \cdot \log n)$. Algoritam se zasniva na podeli prostora na trake širine $R\sqrt{3}$, gde se za svaku nepraznu traku koristi jednostavna pohlepna procedura koja postavlja krug što je niže moguće tako da on pokriva najvišu nepokrivenu tačku. Zbog jednostavnosti, za svaku traku se pretpostavlja da su u njoj sve tačke nepokrivene. Za preciznije objašnjenje pohlepnog pristupa postavljanja krugova u traci uvodi se pojam *prave ograničenja* koja predstavlja pravu koja prolazi kroz sredinu trake i na koju se postavljaju centri krugova koji ulaze u rešenje. Pohlepni pristup se tada zasniva na konstrukciji segmenta na pravoj ograničenja na kojoj se mogu nalaziti centri krugova tako da ti krugovi pokrivaju datu tačku. Segmenti se kreiraju za svaku tačku, zatim se sortiraju neopadajuće i za svaki segment se bira njegova najniža tačka za centar kruga. Proverava se koji su dodatno segmenti obezbeđeni ovakvim izborom, i smatra se da su njima odgovarajuće tačke pokrivene. Osnovna verzija algoritma ima aproksimacioni faktor 5, ali korišćenjem strategije pomeranja (*engl. shifting strategy*) dobija se navedeni aproksimacioni faktor od $\frac{25}{6}$. Ova strategija se zasniva na pomeranju svake trake udesno za $R\frac{\sqrt{3}}{6}$ i traženju najboljeg rešenja među svim pronađenim.



Pohlepni pristup izbora centra kruga u traci (algoritam LL2014)

Vizuealizacija:

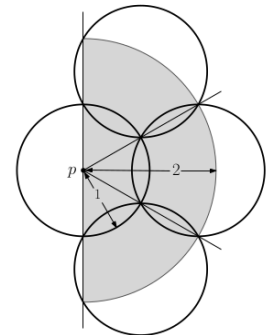
Pretraga se vrši po traci širine $R\sqrt{3}$ koja se iscrtava svakim pomerajem, a u sredini se nalazi isprekidana prava ograničenja na kojoj se pozicioniraju centri krugova. Prilikom izvršavanja algoritma postoji šest iteracija i tokom svake se iscrtavaju krugovi u crnoj boji koji predstavljaju rešenje za trenutnu iteraciju, dok je trenutno najbolje pronađeno rešenje iscrtano krugovima crvene boje.



Vizuealizacija algoritma LL2014

Algoritam3: Biniaz Liu Maheshwari Smid [3]

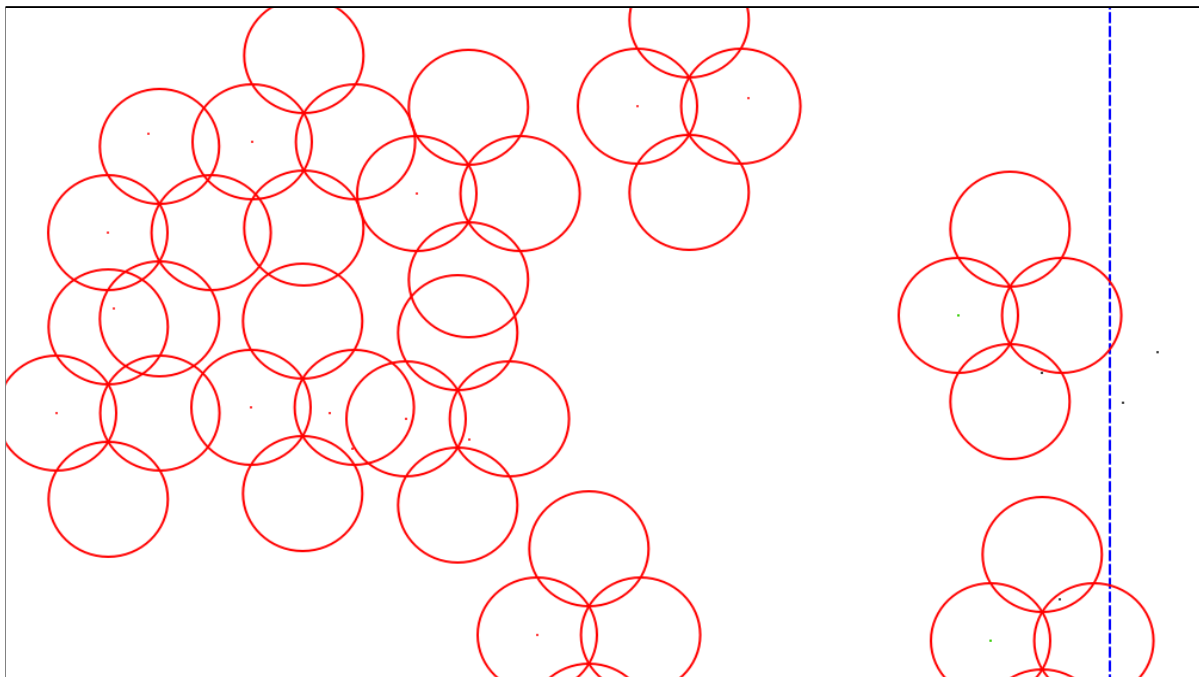
U ovom radu prikazan je 4-aproksimacioni algoritam koji se zasniva na činjenici da je za pokrivanje polukruga poluprečnika 2 potrebno i dovoljno 4 jedinična kruga. Ideja za algoritam se krije u tehnici brišuće prave i obilasku tačaka u sortiranom redosledu. Tokom izvršavanja algoritma održavaju se dve strukture podataka: red događaja koji sadrži događaje položaja i događaje brisanja i pomoću kog se svaka tačka ulaznog skupa obrađuje na adekvatan način kada se na nju naiđe; i binarno stablo pretrage koje sadrži centre krugova, kandidate čiji polukrug poluprečnika 2 potencijalno pokriva tačku koja se u tom trenutku obrađuje. Algoritam se izvršava u vremenu $O(n \cdot \log n)$.



Mana ovog algoritma je što za nepokrivene tačke uvek postavlja 4 nova kruga. To obezbeđuje fiksiran aproksimacioni faktor, međutim ukoliko optimalno rešenje sadrži onoliko krugova koliko ulazni skup ima tačaka, rešenje dobijeno ovim algoritmom će uvek sadržati 4 puta više krugova od optimalnog rešenja.

Vizuealizacija:

Tokom izvršavanja algoritma iscrtava se kretanje brišuće prave po tačkama događaja i dodavanje 4 nova kruga pokrivača kad god brišuća prava naiđe na nepokrivenu tačku.



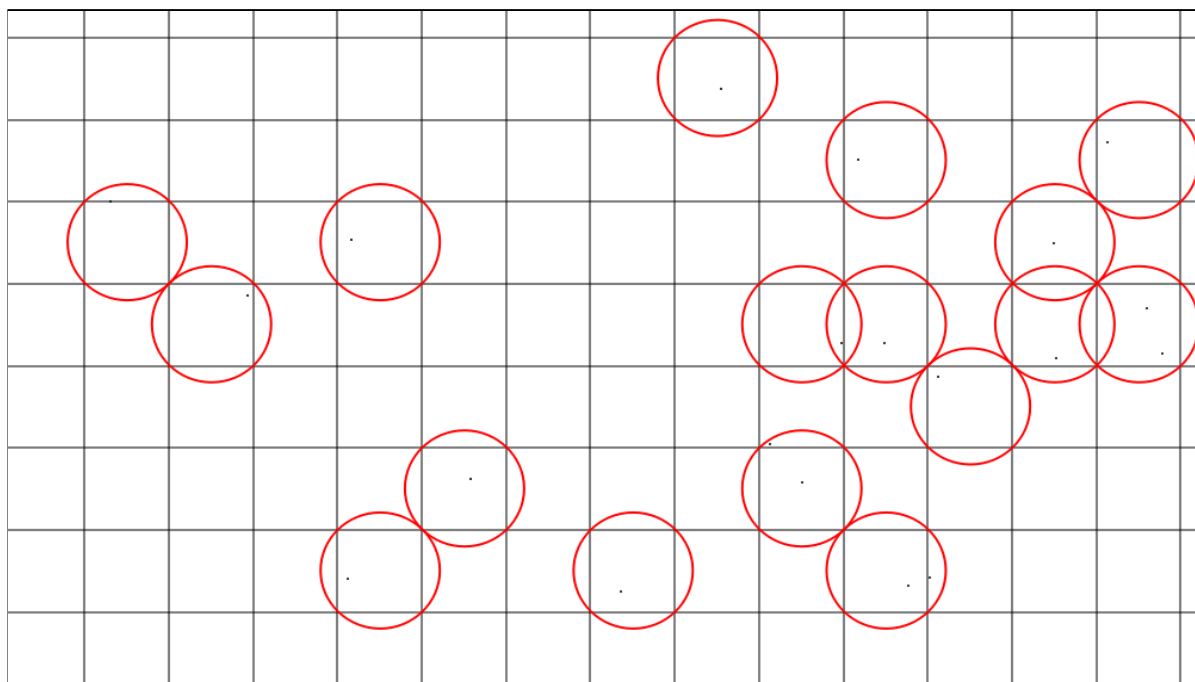
Algoritam4: Ghosh Hicks Shevchenko [4]

Algoritam predstavljen u ovom radu predstavlja online algoritam koji se u odnosu na ostale algoritme izvršava brže, a da pri tome zadržava dovoljan kvalitet pronađenog rešenja. Brzo izvršavanje algoritma omogućeno je nedostatkom potrebe za prethodnom pripremom podataka i čuvanjem podataka u heš tabeli. Autori su u navedenom radu uporedili algoritam sa nekoliko drugih algoritama (uključujući ovde navedene algoritme) i dosli do zaključaka o kvalitetu algoritma i izložili su dokaz o aproksimacionom faktoru koji je jednak 7.

Algoritam se zasniva na kreiranju mreže kvadrata i lociranjem tačaka unutar mreže. Tačke se obilaze u proizvoljnom redosledu (nezavisnom od njihovog položaja) i za svaku se proverava da li je ona sigurno nepokrivena ili je pokrivena krugovima koji odgovaraju kvadratima levo, desno, gore i dole, ukoliko ti krugovi postoje u pokrivaču. Na osnovu toga donosi se odluka o postavljanju novog kruga u centar kvadrata u kom je tačka locirana. Kvadrati oko kojih su opisani krugovi rešenja se čuvaju u heš tabeli.

Vizuealizacija:

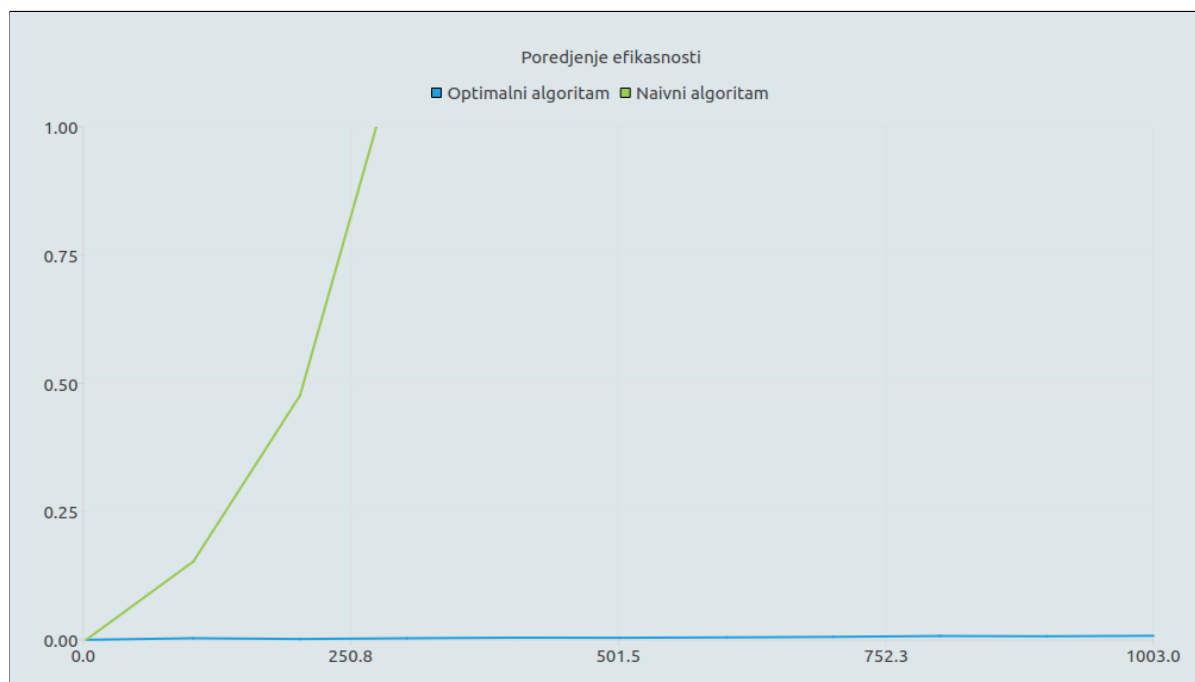
Prilikom izvršavanja algoritma iscrtava se mreža kvadrata nad kojima se vrši pretraga pokrivača i iscrtavaju se jedan po jedan krugovi opisani oko pojedinačnih kvadrata koji odgovaraju konačnom rešenju.



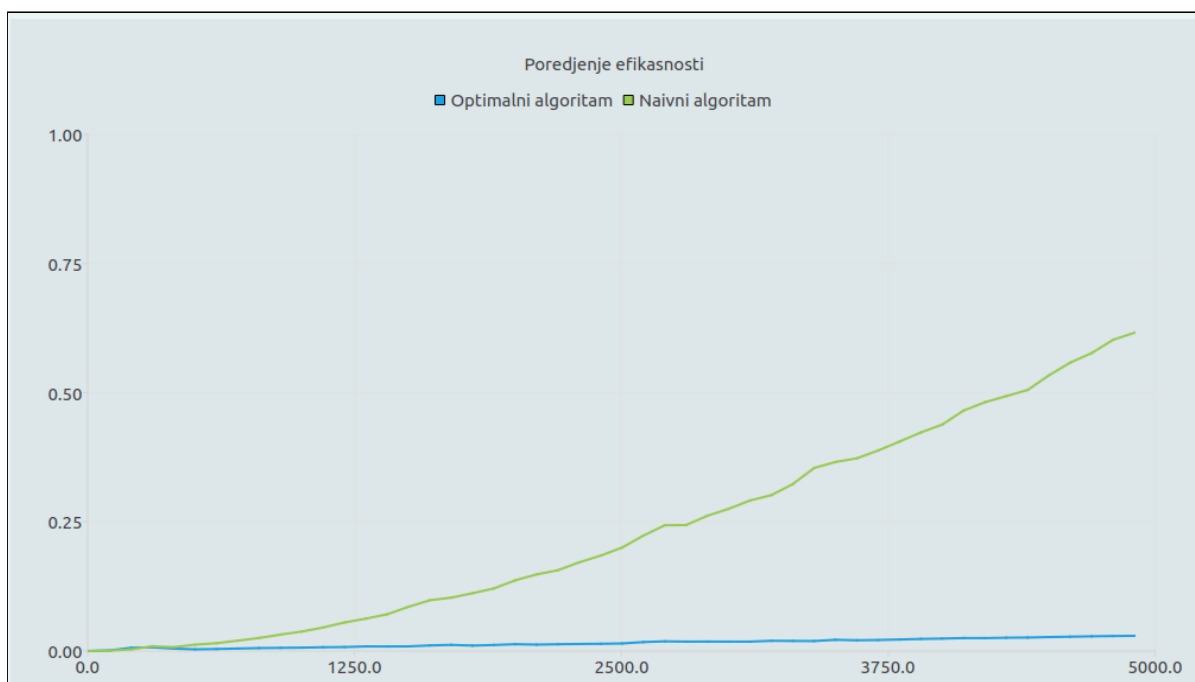
Vizuealizacija algoritma GHS2019

Poređenje efikasnosti naivnog i naprednog algoritma

Izvršeno je poređenje efikasnosti naivnih i svih ostalih algoritama. Može se uočiti značajno povećanje vremena izvršavanja naivnih algoritama sa porastom ulaznog skupa tačaka.



Poređenje efikasnost prvog naivnog algoritma i algoritma LL2014



Poređenje efikasnost prvog naivnog algoritma i algoritma LL2014

Testiranje ispravnosti algoritama

Testovi su pokretani nad svakim od implementiranih algoritama, uključujući i naivne. Testirani su slučajevi sa specifičnim tačkama i nasumično generisanim tačkama. Kao najbitnija osobina, u svakom testu se proverava da li algoritam zaista pokriva sve tačke. Za svaki algoritam je napravljen zaseban test sa nazivom sastavljenim od naziva testa datog u tabeli i oznakom algoritma koji se testira.

Naziv testa	Opis testa	Ulaz	Očekivani izlaz
NoPoints	Ulaz ne sadrži nijednu tačku	Niz dimenzija 0	Pokrivač veličine 0
SamePoints	Ulaz sadrži više tačaka na istoj poziciji	Niz dimenzije 4 iz datoteke file1.txt gde su sve tačke iste	Pokrivač određenje veličine
RandomInput1	Slučajno generisan niz od 20 tačaka	Niz dimenzije 20	Nema nepokrivenih tačaka
RandomInput2	Slučajno generisan niz od 500 tačaka	Niz dimenzije 500	Nema nepokrivenih tačaka

Dodatak

Osnovnu literaturu predstavljao je rad koji sumira ponašanje nekoliko poznatih algoritama za ovaj problem [4]. Literaturu su činili i originalni radovi u kojima su predstavljeni algoritmi, radi lakšeg razumevanja istih. Autori u radu [4] opisuju algoritme, upoređuju kvalitet rešenja i vreme izvršavanja algoritama nad velikim skupovima tačaka i donose zaključke o njima: najbolje vreme izvršavanja pokazao je algoritam GHS2019, dok su algoritmi LL2014 i G1991 pokazali najkvalitetnija rešenja sa najmanjim brojem krugova u pokrivaču.

	G-1991	CCFM-1997	LL-2014	BLMS-2017	DGT-2018	GHS	α
birch3	99990, 0.33	99993, 204.66	99989, 0.76	399912, 10.94	99993, 19.86	99996, 0.08	1.01
monalisa	100000, 0.32	100000, 283.85	100000, 0.65	400000, 15.74	100000, 28.98	100000, 0.08	1
usa	115475, 0.39	115475, 269.99	115475, 0.78	461900, 20.57	115475, 33.4	115475, 0.1	1
KDDCU2D	1288, 0.34	1459, 5.29	1147, 1.59	2652, 2.97	1626, 5.12	1418, 0.03	2.14
europe	168087, 0.59	168079, 546.16	168253, 1.19	670524, 40.11	168069, 66.72	168333, 0.14	1.01
wildfires	623, 3.52	710, 65.24	622, 26.84	1256, 26.72	787, 69.42	663, 0.35	2.12
world	7098, 17.3	8597, 69.98	6667, 26.01	14188, 70.42	10980, 74.01	7874, 0.34	2.22
china	562, 4.72	639, 40.81	565, 24.12	1124, 26.9	723, 44.35	599, 0.29	2.14
nyctaxi	31, 3.31	29, 28.7	25, 53.36	84, 38.68	31, 30.86	34, 0.55	1.62
uber	4, 3.53	5, 41.39	3, 79.5	8, 8.51	5, 29.16	5, 0.78	2.5
hail2015	867, 29.14	998, 399.54	888, 174.67	1724, 234.64	1128, 410.03	901, 1.91	2.1

Tabela poređenja izvršavanja algoritama

[1] [Teofilo F. Gonzalez: Covering a set of points in multidimensional space](#)

[2] [Paul Liu, Daniel Lu: A fast 25/6-approximation for the minimum unit disk cover problem](#)

[3] [Ahmed Biniarz, Paul Liu, Anil Maheshwari, Michiel Smid: Approximation algorithms for the unit disk cover problem in 2D and 3D](#)

[4] [Anirban Ghosh, Brian Hicks, Ronald Shevchenko: Unit Disk Cover for Massive Point Sets](#)