Večnivojski usmerjeni grafi za analizo prostorskih podatkov

Boris Petelin

doktorska disertacija

predana

Fakulteti za računalništvo in informatiko

kot del izpolnjevanja pogojev za pridobitev naziva

doktor znanosti

s področja

računalništva in informatike

Ljubljana, 2014

IZJAVA

Izjavljam, da sem avtor dela in da slednje ne vsebuje materiala, ki bi ga kdorkoli predhodno že objavil ali oddal v obravnavo za pridobitev naziva na univerzi ali na drugem visokošolskem zavodu, razen v primerih, kjer so navedeni viri.

— Boris Petelin —

marec 2014

Oddajo so odobrili

dr. Igor Kononenko

redni profesor za računalništvo in informatiko

MENTOR IN ČLAN OCENJEVALNE KOMISIJE

dr. Matjaž Kukar

docent za računalništvo in informatiko

SOMENTOR IN ČLAN OCENJEVALNE KOMISIJE

dr. Marko Bajec

izredni profesor za računalništvo in informatiko

PREDSEDNIK OCENJEVALNE KOMISIJE

dr. Vlado Malačič

izredni profesor za področje Fizika

ZUNANJI ČLAN OCENJEVALNE KOMISIJE

Nacionalni inštitut za biologijo

PREDHODNA OBJAVA

Izjavljam, da so bili rezultati obravnavane raziskave predhodno objavljeni/sprejeti za objavo v recenzirani reviji ali javno predstavljeni v naslednjih primerih:

[1] B. Petelin, I. Kononenko, V. Malačič and M. Kukar. Multi-level association rules and directed graphs for spatial data analysis. *Expert Systems with Applications*, 40(12):4957–4970, 2013. doi: 10.1016/j.eswa.2013.03.004

[2] B. Petelin, V. Malačič, A. Malej, M. Kukar, I. Kononenko. Multi-level association rules and directed graphs for the Lagrangian analysis of the Mediterranean ocean forecasting system (MFS). V zborniku *Geophys. res. abstr.*, 14:4877, 2012. EGU2012

[3] B. Petelin. *Multi-level association rules and directed graphs for the Lagrangian analysis of oceanographic data* . Vabljeno predavanje, Istituto Nazionale di Oceanografia e di Geofisica Sperimentale, Dipartimento di Oceanografia - OGA, Zgonik (Sgonico), Trst, 11/09/2012.

[4] B. Petelin, I. Kononenko, M. Kukar, V. Malačič and A. Malej. Spatial-temporal directed graphs for modeling the dispersion of plankton in the Mediterranean sea. V zborniku *CIESM Congress Proceedings n°*40, 2013. CIESM2013

Potrjujem, da sem pridobil pisna dovoljenja vseh lastnikov avtorskih pravic, ki mi dovoljujejo vključitev zgoraj navedenega materiala v pričujočo disertacijo. Potrjujem, da zgoraj navedeni material opisuje rezultate raziskav, izvedenih v času mojega podiplomskega študija na Univerzi v Ljubljani.

IN MEMORIAM

Žiga Dobrajc
1979–2010

"The bigger you want to grow, the deeper you have to get into your brain."

— Lenyn Nunez

— Shutterstock, *Man Adrift in tiny boat in binary ocean*.

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Boris Petelin

Večnivojski usmerjeni grafi za analizo prostorskih podatkov

POVZETEK

V disertaciji podajamo svoj prispevek na področju prostorsko-časovnega podatkovnega rudarjenja, ki se je uveljavilo kot odgovor na ogromno količino podatkov, zbranih v operativnih in raziskovalnih podatkovnih zbirkah po vsem svetu. Prednost prostorsko-časovnega podatkovnega rudarjenja v primerjavi s tradicionalnimi metodami je v ustrezni obravnavi prostorskih in časovnih atributov in posledično sposobnost odkrivanja skritih informacij, ki jih vsebujejo omenjene podatkovne zbirke.

V disertaciji smo se omejili na prostorsko-podatkovno rudarjenje v zemeljskih znanostih oziroma bolj natančno, razvili smo metodologijo, ki nadgrajuje metode Lagrangevega sledenja premikajočih se navideznih delcev vodnih mas v oceanih. Pri klasični Lagrangevi analizi proučujemo poti oziroma trajektorije simuliranih delcev z vizualnim pregledom ali pa uporabimo razne statistične in druge metode (npr. teorija dinamičnih sistemov, stohastično modeliranje itd.). Na področju oceanografije, ki je poglavitni izvor podatkov za našo disertacijo, se vedno bolj uveljavljajo kvalitetni numerični modeli. Na osnovi hitrostnih polj v rezultatih takšnega modela najprej tvorimo veliko število trajektorij navideznih delcev (tipično okoli 100.000), nato razdelimo domeno numeričnega modela na manjša območja in poiščemo prostorsko-časovna povezovalna pravila, ki povedo, kolikšne so verjetnosti prehodov navideznih delcev iz posameznih območij v sosednja območja v določenem časovnem intervalu. Dobljena pravila prikažemo v obliki večnivojskih usmerjenih grafov z različno granulacijo v prostoru in času. Povezavam in vozliščem v takšnih grafih lahko dodamo poljubne atribute, ki predstavljajo agregirane ali statistične podatke, oziroma oceanografsko ali drugo znanje.

Dobljeni večnivojski usmerjeni grafi so primerni za analizo s številnimi algoritmi, ki se uporabljajo za rudarjenje grafov. Naš prispevek predstavljajo algoritmi za iskanje značilnih struktur (poti in ciklov) v teh grafih. V posameznih grafih najprej poiščemo preproste cikle, ki se pojavljajo v krajših obdobjih (en mesec), vendar se bolj realistične

poti in cikli pojavljajo v daljših obdobjih, npr. več mesecev in celo več let. V disertaciji se ukvarjamo s potmi in cikli, ki se raztezajo v obdobjih, dolgih nekaj mesecev vendar krajših od enega leta. Simulacije, ki jih obravnavamo, so namreč prekratke za odkrivanje dolgotrajnejših procesov in posledica tega so dinamične strukture (poti in cikli). Pri konstrukciji teh poti in ciklov moramo upoštevati, da se uteži grafov s časom spreminjajo, zato te poti in cikle imenujemo dinamične. Dobljene dinamične poti in cikle hierarhično združujemo na osnovi medsebojne razdalje v dinamične mehke poti in cikle ter jih primerjamo s strukturami, ki jih poznajo oceanografski eksperti iz opazovanj v dani domeni. Rezultati v disertaciji kažejo precejšnje ujemanje dobljenih dinamičnih mehkih poti in ciklov z opazovanji oziroma s predhodnim znanjem.

Metodologija, ki jo opisujemo v disertaciji, je dobra osnova za razvoj aplikacij, ki nadgrajujejo uveljavljene metode, ki jih uporabljajo oceanografski eksperti. V 6. poglavju disertacije prikazujemo nekaj primerov uspešne uporabe večnivojskih usmerjenih grafov. Tako pokažemo, da je gibanje vodnih mas v Sredozemskem morju periodično s ciklom 12 mesecev oziroma izkazuje svoj sezonski značaj. Poleg tega s pomočjo večnivojskih usmerjenih grafov prikažemo dolgoročne prehodne pojave, kot je obrat cirkulacije v Jonskem morju približno vsakih 10 let. Z uporabo dodatnih atributov (npr. moč vetra) v večnivojskih usmerjenih grafih prikažemo povezanost verjetnosti premikov vodnih mas s temi atributi. Ne nazadnje so večnivojski usmerjeni grafi solidna osnova za modeliranje širjenja bioloških vrst.

Ključne besede: prostorsko-časovno podatkovno rudarjenje, Lagrangeva analiza, prostorsko-časovna povezovalna pravila, večnivojski usmerjeni grafi, oceanografija

University *of Ljubljana*
Faculty *of Computer and Information Science*
Boris Petelin
Multi-level directed graphs for analysis of spatial data

ABSTRACT

The dissertation provides a contribution to the field of spatial-temporal data mining, which is a response to the enormous amount of data collected in operational and research databases worldwide. The advantage of spatial-temporal data mining compared to traditional methods is the appropriate treatment of spatial and temporal attributes and consequently the ability to discover hidden information which is contained in databases.

The dissertation is limited to spatial-temporal data mining in the earth sciences, or more precisely, we have developed a methodology that upgrades methods of the Lagrangian particle tracking of moving virtual water parcels in the ocean. By using classic Lagrangian analysis, we examine paths or trajectories of simulated water parcels by visual inspection or by using various statistical methods as well as other approaches (i.e. dynamic system theory, stochastic modeling etc.). In the field of oceanography, which is the primary data source for the dissertation, high-quality numerical modeling is becoming increasingly important. Based on the velocity fields in the results of such a model, we first produce a number of trajectories of virtual particles (typically around 100,000). In the next step, we subdivide the model domain into smaller areas and search for spatial-temporal association rules that enable us to obtain the probability of the transition of virtual particles from individual sea areas to neighboring ones within the specified time interval. We visualize the resulting rules in the form of multi-level directed graphs with different granulation in space and time. We can add any attributes to the edges and vertices in such graphs, which represent aggregated or statistical information, or oceanographic or other material.

The resulting multi-level directed graphs are open to numerous algorithms that are used for graph mining. Our contribution is the algorithms for searching significant structures (paths and cycles) in these graphs. First we uncover simple cycles that occur

in short periods of time (one month) within one graph, though more realistic paths and cycles occur over longer periods, that is, several months or even years. In the dissertation, we deal with paths and cycles that extend into periods of several months but less than one year. We deal with simulations which are too short to detect longer term processes and this results in dynamic paths. For the construction of these paths and cycles, we must take into account that the weights of graphs change over time, so the resulting paths and cycles are called *dynamic*. We perform hierarchical clustering of the resulting dynamic paths and cycles based on the distance between them and obtain *dynamic fuzzy paths* and *cycles* and compare them with the structures that are known from oceanographic observations provided by domain experts. The results in the dissertation show the significant similarity of obtained dynamic fuzzy paths and cycles with the observations and prior knowledge of oceanographic experts.

The methodology, described in the dissertation, is a solid basis for the development of applications that upgrade the established methods used by oceanographic experts. In chapter 6 of the dissertation, we present some examples of successful applications of multi-level directed graphs. Thus, we show that the movement of water masses in the Mediterranean Sea has a seasonal nature with a period of 12 months. In addition, by using multi-level directed graphs we present long-term transient phenomena such as the circulation reversal in the Ionian Sea, which occurs approximately every 10 years. By using additional attributes (e.g. wind power) in multi-level directed graphs, we show the correlation of the probability of movements of water masses with these attributes. Finally, multi-level directed graphs are a solid basis for modeling the dispersal of biological species.

Key words: spatial-temporal data mining, Lagrangian analysis, spatial–temporal association rules, multi-level directed graphs, oceanography

ZAHVALA

Najprej se zahvaljujem svojemu mentorju, prof. dr. Igorju Kononenku, in somentorju, doc. dr. Matjažu Kukarju, za vodenje in usmerjanje mojega doktorskega študija ter za številne ideje, ki so se tudi uresničile in s pomočjo katerih je nastala ta doktorska disertacija.

Posebna zahvala gre vodji Morske biološke postaje Nacionalnega inštituta za biologijo (NIB-MBP), prof. dr. Vladu Malačiču, in vodji raziskovalnega programa ARRS "Raziskave obalnega morja" prof. dr. Alenki Malej, ki sta mi omogočila doktorski študij v okviru NIB-MBP in ga s strani NIB-MBP tudi finančno podprla. Obema se zahvaljujem tudi za nasvete v zvezi z oceanografskimi, biološkimi in ekološkimi znanji, ki so povezana z mojo doktorsko disertacijo.

Doktorski študij je delno sofinancirala Evropska Unija, in sicer iz Evropskega socialnega sklada. Sofinanciranje se izvaja v okviru Operativnega programa razvoja človeških virov za obdobje 2007-2013, 1. razvojne prioritete Spodbujanje podjetništva in prilagodljivosti; prednostne usmeritve_1. 3: Štipendijske sheme.

Hvaležen sem vsem sodelavcem Laboratorija za kognitivno modeliranje, ki so v času doktorskega študija poslušali moje seminarje in brainstorminge ter mi dali marsikateri nasvet, ki je koristil mojemu doktorskemu delu. Poleg tega se zahvaljujem še ostalim sodelavcem iz Nacionalnega inštituta za biologijo in še posebej Morske biološke postaje Piran za vsakršno pomoč, ki sem je bil deležen v času študija.

Na koncu bi se rad zahvalil še svoji mami za vso skrb in potrpežljivost, ker sem zaradi obilice dela v zvezi z doktorskim študijem imel premalo časa, da bi kaj postoril tudi doma.

I would like to thank oceanographic experts from institutions outside Slovenia for any help and advice I needed in my doctoral study. Special thanks go to Pierre-Marie Poulain, PhD, Milena Menna, PhD and Miro Gačić, PhD from Istituto Nazionale di Oceanografia e Geofisica Sperimentale (OGS), Trieste, Italy, Sandro Carniel, PhD from ISMAR-CNR,

Venezia, Italy, Nadia Pinardi, PhD from Istituto Nazionale di Geofisica e Vulcanologia (INGV), Bologna, Italy and many others.

— Boris Petelin, Ljubljana, marec 2014.

KAZALO

I

Uvod

Prostorsko-časovno podatkovno rudarjenje (ang. *spatial-temporal data mining*) in odkrivanje geografskega znanja (ang. *geographic knowledge discovery*) – ti izrazi se lahko uporabljajo izmenično – se je v zadnjem desetletju hitro razvijalo zaradi ogromne rasti količine geografskih podatkov. Omenjeni podatki se zbirajo s pomočjo sodobnih tehnik pridobivanja podatkov, kot so sistemi satelitskega določanja položaja (GPS), daljinsko zaznavanje v visoki ločljivosti (satelitski posnetki) in merilne naprave, ki poleg samih podatkov posredujejo tudi njihovo prostorsko in časovno informacijo. Takšni podatki so ponavadi prostorsko, časovno ali pa kar prostorsko-časovno odvisni. Tipični primer je sledenje premikajočim se objektom, ki se v danem trenutku nahajajo na točno določenem mestu. Prostorsko-časovno podatkovno rudarjenje imenujemo tudi geografsko podatkovno rudarjenje in odkrivanje znanja in se uporablja na mnogih področjih, npr. v meteorologiji (gibanje tornadov), biologiji (gibanje živali), gozdarstvu (širjenje požarov), zdravstvu (širjenje epidemij), ekologiji (sledenje onesnaženj), prometu (sledenje vozil) itd.

V nasprotju s klasičnim podatkovnim rudarjenjem metode prostorsko-časovnega podatkovnega rudarjenja upoštevajo prostorsko-časovne lastnosti tj. geografske meritve, prostorsko odvisnost, heterogenost in kompleksnost objektov ter pravil in različne tipe podatkov. Tako imamo namesto klasičnih metod uvrščanja, razvrščanja, povezovalnih pravil itd. sedaj prostorsko uvrščanje in ugotavljanje prostorskih odvisnosti, prostorsko razvrščanje, prostorske trende, prostorsko generalizacijo, prostorska povezovalna pravila in geografsko vizualizacijo. Zainteresirani bralec si lahko ogleda izčrpne opise metod prostorskega podatkovnega rudarjenja v [1] in [2]. Zaradi svojih prostorsko-časovnih lastnosti so te metode hitro našle svoje mesto v zemeljskih znanostih. Prišlo je do znatnega povečanja uporabe teh metod pri reševanju okoljskih problemov in v meteorologiji ter oceanografiji. V tej disertaciji se omejimo na ožji del prostorsko-časovnega podatkovnega rudarjenja in ga uporabimo na področju oceanografije. Opišemo novo metodologijo rudarjenja trajektorij Lagrangeovih delcev [3] s pomočjo prostorsko-časovnih povezovalnih pravil in večnivojskih usmerjenih grafov. Prikažemo algoritme za rudarjenje omenjenih grafov, ki so zmožni poiskati najbolj verjetne poti in cikle pri gibanju Lagrangeovih delcev. Na koncu podkrepimo uporabnost omenjene metodologije z aplikacijami, namenjenimi oceanografskim ekspertom.

1.1 Motivacija

Zgodovina oceanografskih meritev je stara že nekaj stoletij; v naši bližini pa se je začela že pred več kot sto leti. Kot primer lahko navedemo meritve temperature in slanosti morja, ki jih je v letih 1905–1908 v Tržaškem zalivu opravil avstrijski geograf in oceanograf Alfred Merz [4]. V zadnjem času se meritve opravljajo s sofisticiranimi instrumenti na različnih platformah (križarjenja z raziskovalnimi plovili, oceanografske boje, plovci različnih vrst, satelitski posnetki itd.). Količina oceanografskih podatkov je dandanes ogromna in shranjena v številnih podatkovnih zbirkah nacionalnih oceanografskih centrov po vsem svetu. V Evropski uniji se izvajajo različni projekti npr. MyOcean [5] in SeaDataNet [6], katerih namen je povezati vse te baze podatkov na način, da postanejo prosto dostopne partnerjem na teh projektih in tudi drugim uporabnikom v EU. Ti projekti so bili do sedaj uspešno izvedeni v tolikšni meri, da lahko uporabniki za svoje potrebe dostopajo do željene podmnožice podatkov preko spletnih aplikacij. V praksi zaradi velikih količin podatkov uporabniki vizualno analizirajo le majhen delež teh podatkov, zato je potreba po metodah prostorsko-časovnega podatkovnega rudarjenja še toliko večja. Do sedaj so se v oceanografiji in meteorologiji uveljavile že številne metode, od katerih nekatere omenimo v razdelku 2.1.

Pomemben del razpoložljivih podatkov se nanaša na morske tokove oziroma gibanje vodnih mas v morjih. Ti podatki so bili pridobljeni bodisi z meritvami s pomočjo plovcev različnih vrst in merilnih instrumentov na zasidranih oceanografskih bojah in različnih plovilih, bodisi predstavljajo rezultate simulacij s pomočjo oceanografskih numeričnih modelov. Slednji nam dajo najbolj celovito sliko o gibanju vodnih mas (glej primer na sliki 5.1). Ti modeli so gnani s pomočjo rezultatov meteoroloških modelov in robnih ter začetnih pogojev, ki jih dobijo od bolj grobih oceanografskih modelov, v katere so vgnezdeni. Robni pogoji predstavljajo podatke o hitrosti morskih tokov, višini gladine, temperaturi, slanosti itd. na odprtih robovih modela in tudi podatke o vetrni napetosti, sončnem sevanju, toplotnem toku, padavinah itd. na morski gladini. Začetni pogoji predstavljajo razporeditev temperature in slanosti v celotni domeni modela na začetku simulacije. Pri simulacijah preteklih situacij s pomočjo numeričnih modelov oceanografski eksperti v slednje asimilirajo izmerjene podatke, da bi dobili rezultate, ki so čim bližje realni situaciji. Slabost numeričnih modelov je še vedno njihova parametrizacija, zato ne morejo popolnoma reproducirati gibanja vodnih mas, kakršno se v morju v resnici pojavlja, in še vedno potrebujejo verifikacijo s pomočjo

oceanografskih meritev.

Z vizualnim pregledom hitrostnih polj morskih tokov v rezultatih numeričnih modelov oceanografski eksperti ugotavljajo splošne slike gibanja vodnih mas v morju. Pri tem se uporabljajo tudi Lagrangeovo sledenje majhnih delčkov vodnih mas [3], kjer v določenih točkah domene numeričnega modela spuščajo navidezne delce in opazujejo njihove trajektorije v prostoru in času. Poleg vizualnega pregleda trajektorij strokovnjaki uporabljajo še razne statistične in druge metode (teorija dinamičnih sistemov, stohastično modeliranje itd.) [3]. Uporabljajo se tudi nekatere metode prostorsko-časovnega podatkovnega rudarjenja t.j. razvrščanje trajektorij (ang. *trajectory clustering*), ki pa so se bolj uveljavile v meteorologiji [7, 8] kot pa v oceanografiji, kjer je gibanje vodnih mas dosti bolj zapleteno.

V disertaciji se lotimo prostorsko-časovnega podatkovnega rudarjenja hitrostnih polj v rezultatih omenjenih oceanografskih numeričnih modelov in kasneje tudi rezultatov meritev s pomočjo plovcev. Pri tem uporabimo metodologijo, ki smo jo razvili posebej v ta namen in je sestavljena iz več korakov (poglavje 3). Najprej v hitrostnih poljih rezultatov numeričnega modela ustvarimo veliko število Lagrangeovih trajektorij navideznih delcev (okoli 100.000). Pri tem vzamemo rezultate numeričnih modelov take kot so, ker samo ocenjevanje kvalitete rezultatov numeričnih modelov in tudi same metode za tvorbo trajektorij navideznih delcev presega okvir te disertacije in to prepustimo oceanografskim ekspertom. Osnovno pravilo, iz katerega izhajamo, je: "*Če se delec v določenem časovnem intervalu nahaja na območju A dane prostorske domene, potem se isti delec v naslednjem časovnem intervalu nahaja na območju B z določeno verjetnostjo.*", iz česar pridemo do uporabe prostorsko-časovnih povezovalnih pravil. Pri tem moramo domeno numeričnega modela diskretizirati oziroma jo razdeliti na primerno velika območja in na njihovi osnovi poiskati prostorsko-časovna povezovalna pravila. Iz množice omenjenih pravil tvorimo usmerjene grafe. Pri tem lahko domeno razdelimo na poljuben način (na večja ali manjša območja) in pri tem uporabimo ustrezen časovni interval, zato dobljene grafe imenujemo *večnivojski usmerjeni grafi*. Njihova večnivojskost se torej odraža v dejstvu, da pokrivajo različne granulacije prostorskih območij domene in različne časovne razpone. Svoj pravi pomen večnivojski usmerjeni grafi dobijo šele, ko na njih izvedemo algoritme rudarjenja (t.j. iskanje značilnih struktur – najbolj verjetnih poti in ciklov v grafih – glej poglavji 4 in 5) in pokažemo, da je možno na osnovi omenjene metodologije razviti številne nove programe oziroma aplikacije, ki so v pomoč oceanografskim ekspertom (poglavje 6).

1.2 Prispevki k znanosti

Poglavitni prispevki doktorske disertacije k znanosti so:

- *metodologija, ki uporablja prostorsko-časovna povezovalna pravila in večnivojske usmerjene grafe za analizo velike množice Lagrangeovih trajektorij objektov*: v poglavju 3 je predstavljena nova metodologija, s pomočjo katere iz velike množice Lagrangeovih trajektorij dobimo t. i. večnivojske usmerjene grafe. S pomočjo rudarjenja teh grafov pridemo do značilnih vzorcev gibanja objektov.

- *algoritmi za rudarjenje večnivojskih usmerjenih grafov*: v poglavju 4 so opisani nekateri algoritmi za rudarjenje oziroma iskanje struktur v večnivojskih usmerjenih grafih. To so *dinamične mehke poti* in *dinamični mehki cikli*, ki jih dobimo s sprehodom v smeri najbolj verjetnih povezav v časovni vrsti večnivojskih usmerjenih grafov.

- *specifične aplikacije, zasnovane na večnivojskih usmerjenih grafih*: v poglavju 6 je opisanih nekaj aplikacij, ki uporabljajo večnivojske usmerjene grafe in služijo kot ideja za razvoj bolj kompleksnih aplikacij. Slednje so lahko v veliko pomoč oceanografskim ekspertom pri interpretaciji velikih količin oceanografskih podatkov z namenom potrditve obstoječih in nastavka novih hipotez.

1.3 Pregled naloge

Disertacija je sestavljena iz sedmih poglavij. Drugo poglavje našteje nekatere novejše primere razvoja in uporabe metod prostorsko-časovnega podatkovnega rudarjenja na področju oceanografije in meteorologije. V istem poglavju podajamo tudi ozadje, ki ga predstavljajo prostorsko-časovna povezovalna pravila in večnivojski usmerjeni grafi, ki so osnova za metodologijo, ki smo jo razvili v okviru te disertacije. V tretjem poglavju omenjeno metodologijo podrobno opišemo. V četrtem poglavju so podani nekateri algoritmi, ki smo jih razvili za potrebe rudarjenja večnivojskih usmerjenih grafov, ki so rezultat omenjene metodologije. Omenjene algoritme smo razvili z namenom iskanja pogostih struktur (poti in ciklov), ki se pojavljajo v zvezi z gibanjem vodnih mas v morju. Peto poglavje opisuje ovrednotenje metodologije na rezultatih oceanografskega numeričnega modela Mediterranean Ocean Forecasting System (MFS) [9, 10], podaja rezultate algoritmov in jih primerja s strukturami, ki jih poznajo oceanografski eksperti.

V šestem poglavju podrobneje opišemo nekaj uporab večnivojskih usmerjenih grafov, ki koristijo oceanografskim ekspertom, in na osnovi katerih lahko razvijemo nove (bolj kompleksne) aplikacije. Zadnje, sedmo poglavje podaja zaključke, razpravo in načrte za nadaljnje delo.

2

Ozadje

V tem poglavju najprej opišemo nekaj reprezentativnih metod prostorsko-časovnega podatkovnega rudarjenja. Pri tem se omejimo na metode, ki so se v zadnjih letih uveljavile v oceanografiji, ekologiji in meteorologiji in se navezujejo na rezultate metodologije, ki jo opisujemo v disertaciji. Omenjene metode so opisane v številnih člankih v revijah, ki pokrivajo bodisi področja zemeljskih znanosti kot tudi računalniške znanosti. Kot vidimo, večina opisanih metod uporablja nenadzorovano učenje npr. prostorsko-časovno razvrščanje, prostorsko-časovna povezovalna pravila, vizualizacija s pomočjo samoorganizirajoče se mreže (SOM) itd. V disertaciji smo se pri razvoju metodologije in algoritmov v glavnem oprli na metode, ki uporabljajo nenadzorovano učenje. Čeprav je uporaba metod, ki uporabljajo nadzorovano učenje, v oceanografiji redkejša, v nadaljevanju podajamo tudi primer prostorske klasifikacije. Vzrok za pomanjkanje uporabnih metod za nadzorovano učenje je med drugim tudi kompleksnost problemov, ki otežuje uporabo nadzorovanega učenja. V drugem razdelku opisujemo ozadje prostorsko-časovnih povezovalnih pravil in večnivojskih usmerjenih grafov, ki predstavljajo temelje metodologije, ki jo opisuje doktorska disertacija.

2.1 Pregled del

2.1.1 Prostorska klasifikacija

Prostorska klasifikacija je metoda nadzorovanega učenja, ki obravnava preslikavo prostorskih in drugih podatkov v določene razrede, kjer je število slednjih veliko manjše od števila samih podatkov. To metodo predstavlja gradnja klasifikacijskih in regresijskih dreves, ki so zelo uporabna za analizo kompleksnih kategoričnih in/ali numeričnih podatkov, s katerimi imamo opravka v oceanografiji in ekologiji.

Dea'th in Fabricius [11] sta uporabila klasifikacijska in regresijska drevesa za analizo podatkov iz raziskave o abundanci (številčnosti) mehkih koral iz podrazreda Octocoralia (Cnidaria) na avstralskem Velikem koralnem grebenu. Pri tem sta analizirala širok nabor podatkov, pridobljenih z opazovanji in meritvami na 374 mestih vzdolž koralnega grebena. Podatki so zajemali številčnost koral, fizikalne parametre kot npr. debelina sedimenta, vidljivost v morju, vpliv valov in naklon dna ter prostorske parametre, kot so lokacija na kontinentalni polici (šelfu), vrsta grebena, lokacija v grebenu in globina opazovanja. Numerične podatke, kot so npr. abundanca, vidljivost in naklon dna, sta razdelila v intervale (diskretizirala), medtem ko sta druge podatke obravnavala kot kategorične, npr. vpliv valov je lahko zanemarljiv, zmeren ali pa močan. Avtorja sta

uporabila omenjene parametre bodisi kot odvisne ali pa kot neodvisne spremenljivke. Analiza regresijskih dreves je pokazala, da gosto naseljene združbe treh vrst koral zasedajo različne habitatne tipe, kjer je vsak izmed njih opredeljen s 3–4 okoljskimi spremenljivkami. Številčnost treh vrst mehkih koral je bila klasificirana v območju od 54 % do 68 % pojasnjene variance. Med fizikalnimi spremenljivkami sta dobila najboljši rezultat pri vidljivosti v morju s 66,7 % pojasnjene variance, sledijo ji debelina sedimenta, naklon dna in vpliv valov, ki imajo po vrsti pojasnjene variance 56,4 %, 39,3 % in 34,1 %. Avtorja sta primerjala rezultate klasifikacije s pomočjo regresijskih dreves z rezultati alternativnih metod, kot so analiza variance in linearni regresijski modeli, in pokazala, da omenjene metode ne morejo razkriti določenih vzorcev, ki jih lahko pokažejo regresijska drevesa. Če povzamemo, klasifikacijska in regresijska drevesa so zelo primerna za analizo oceanografskih in ekoloških podatkov, vendar je potrebno pred tem prostorsko informacijo pretvoriti v opisne atribute.

2.1.2 Prostorsko-časovno razvrščanje

Razvrščanje (ang. *clustering*) je pomembna metoda za rudarjenje prostorskih podatkov, ki organizira podatke v skupine (ang. *cluster*) tako, da so podatki v isti skupini med seboj čim bolj podobni in se hkrati čim bolj razlikujejo od podatkov v drugih skupinah. Poznamo množico metod za prostorsko razvrščanje in večino lahko uvrstimo v naslednje kategorije: metode z razdelitvijo (ang. *partitioning*), hierarhične metode, metode na osnovi gostote (ang. *density-based metods*) in metode, osnovane na mrežah (ang. *grid based methods*) [1]. Za analizo skupin v morju na osnovi fizikalnih podatkov se zadnji dve metodi zdita najbolj obetavni. Metode, osnovane na gostoti, se uporabljajo za iskanje gruč poljubnih oblik (gosta področja, ki so med seboj ločena s področji z nizko gostoto – slednja predstavljajo šum). Reprezentativni algoritmi, ki se lahko uporabljajo v ta namen, so DBSCAN, OPTICS, DENCLUE, Wavecluster in CURD (glej [1] in [12]). V tem razdelku se osredotočimo na algoritem DBSCAN (ang. *density-based spatial clustering of applications with noise*). Podrobnosti o tem algoritmu lahko najdemo v [13]. Številne metode razvrščanja temeljijo na tem algoritmu ali pa ga uporabljajo kot takega. DBSCAN učinkovito obravnava prostorske podatke, vendar pa ni uporaben pri prostorsko-časovnih podatkih, ki jih srečujemo v oceanografiji.

Birant in Kut [12] predlagata nadgradnjo tega algoritma (dobljeni algoritem imenujeta ST-DBSCAN), ki temelji na gostoti in poleg svojih drugih prednosti ne zahteva vnaprej določenega števila skupin. Vendar pa potrebuje dva pomembna vhodna pa-

rametra: velikost okolice (*Eps*) in minimalno število točk v soseščini (*MinPts*). Za določanje teh parametrov avtorja uporabljata primerne hevristike, ki so opisane v [13]. S pomočjo omenjenih hevristik, avtorja najprej določita *MinPts*, ki je velikostnega reda $\ln n$, kjer je n število podatkov (točk), ki jih razvrščata. Potem izračunata razdalje od vseh točk do pripadajočih *MinPts* sosedov, jih uredita v padajočem vrstnem redu in prikažeta v obliki grafa. Primerna razdalja *Eps* se nahaja v prvi "dolini" grafa. Posledica tega je, da se vse točke, ki imajo razdaljo večjo od *Eps*, smatrajo za šum.

Avtorja sta omenjeni algoritem uporabila za razvrščanje morskih območij okoli Turčije na osnovi fizikalnih oceanografskih parametrov, kot so temperatura na površini morja, anomalija višine morske gladine in značilna višina morskih valov. Omenjeni oceanografski parametri so bili pridobljeni s pomočjo satelitov in shranjeni v dobro zasnovano prostorsko-časovno podatkovno bazo. To bazo sta avtorja uporabila tudi za shranjevanje rezultatov razvrščanja. Razvila sta tudi uporabniku prijazen vmesnik, ki omogoča delo z aplikacijo tudi neizkušenim uporabnikom.

Za potrebe razvrščanja prostorsko-časovnih podatkov sta avtorja rešila nekatere probleme, ki so bili značilni za obstoječe pristope. Najprej predlagata dve metriki za izračun razdalje namesto ene, ki jo uporablja algoritem DBSCAN. Prva definira geografsko bližino točk (zemljepisna dolžina in širina), druga pa obravnava podobnost na osnovi ne-prostorskih podatkov, ki so v bistvu oceanografski podatki (temperatura morja itd.). Drugič, da bi lahko opredelila šum pri skupinah z različnimi gostotami, predlagata t. i. faktor gostote, ki se dodeli vsaki skupini. Tretjič, problem identifikacije sosednjih skupin sta rešila tako, da sta primerjala povprečno gostoto v skupini z novimi podatki in dodala nove podatke tisti skupini, kjer je bila razlika med njeno povprečno gostoto in novimi podatki dovolj majhna. Na koncu sta avtorja dodala še t. i. časovno soseščino, kar pomeni, da algoritem združuje podatke, ki so bili izmerjeni v zaporednih dneh v istemu letu ali pa isti dan v letu v različnih letih.

Zaključimo lahko, da je razvrščanje točkovnih prostorskih podatkov na osnovi gostote zelo obetavna metoda za rudarjenje prostorskih podatkov v oceanografiji. Vendar ima skrbna izbira parametrov za razvrščanje tukaj bistven pomen.

2.1.3 *Razvrščanje trajektorij*

Posebne metode razvrščanja so bile razvite v primeru trajektorij (Definicija 1), tj. podatkov, ki so bili zbrani v zvezi s pojavi, kjer se v času pogosto spreminja geografska lokacija [1]. Do sedaj je bilo razvitih le nekaj metod za razvrščanje trajektorij. Prvotne

metode so obravnavale trajektorije v celoti, novejše metode pa že omogočajo razvrščanje delov trajektorij. Obstajata dva pristopa za razvrščanje celotnih trajektorij: na osnovi verjetnosti in na osnovi gostote. Za razvrščanje delov trajektorij pa se uporablja t. i. koncept delitve in združevanja (ang. *partition-and-group framework*). Sestoji se iz dveh faz: v prvi se vsaka trajektorija razdeli na odseke, v drugi pa se podobni odseki združijo v skupine.

Camargo in sodelavci [7] so izvedli analizo trajektorij tropskih ciklonov na območju zahodnega severnega Pacifika. Uporabili so verjetnostno metodo razvrščanja (ang. *probabilistic clustering technique*), osnovano na modelu zmesi regresije (ang. *regression mixture model*). S to metodo so sestavili zmes polinomskih regresijskih modelov (tj. krivulj), ki so jih prilagodili geografskim oblikam trajektorij. Uporabili so podatke tistih tropskih ciklonov iz obdobja 1950–2002, ki so imeli najmanj značilnost tropskih neviht, in tako našli sedem izrazito ravnih in tudi ukrivljenih skupin trajektorij. Avtorji so proučili različne lastnosti tropskih ciklonov v vsaki skupini, med drugim porazdelitev trajektorij, lokacijo nastanka, intenzivnost, življenjsko dobo, prehod iz morja na kopno in prehod med skupinami.

Predvidevamo, da bo razvoj metod razvrščanja trajektorij dosegel stopnjo, da bodo omenjene metode postale zelo uporabne tudi v oceanografiji. Gibanje vodnih mas v morju je namreč zelo kompleksno in je zato potrebno metode razvrščanja trajektorij še izboljšati.

2.1.4 Prostorsko-časovna povezovalna pravila

Iskanje povezovalnih pravil je bilo prvotno zasnovano za ugotavljanje odvisnosti med atributi v velikih transakcijskih podatkovnih bazah [14, 15]. Tradicionalne aplikacije za analizo nakupovalnih košaric vključujejo znane metode za iskanje povezovalnih pravil med izdelki znotraj ene transakcije. Omenjeni pristop je znan kot rudarjenje podatkov znotraj transakcij (ang. *intra-transaction itemset*) in se v ta namen uporabljajo klasični algoritmi tj. APriori [14]. Da lahko vključimo bogate prostorske in časovne odvisnosti med oceanografskimi podatki, je potrebno v iskanje povezovalnih pravil vključiti koncept, ki upošteva atribute v različnih transakcijah, ki so med odvisne oziroma povezane v prostoru in času (ang. *inter-transaction itemset*).

Huang in sodelavci [16] predlagajo učinkovit algoritem za iskanje povezovalnih pravil med podatki o slanosti in temperaturi, izmerjenimi v morjih, ki obkrožajo otok Tajvan. Primer takšnega povezovalnega pravila ima npr. obliko: “če je slanost v bližini

severnega Tajvana narasla za 0,1–0,2 psu, potem bo temperatura v srednji oddaljenosti od severovzhodnega Tajvana v naslednjem mesecu narasla za 0–0,8 ^{0}C". V ta namen so opredelili t. i. referenčno-centrični model in razdelili morje okrog Tajvana v sektorje, ki so bili določeni z dvema koncentričnima krogoma in štirimi premicami skozi središče teh krogov (mesto Taipei). Tako sta bili prostorski razsežnosti definirani z oddaljenostjo in smerjo od središča krogov. Da bi opredelili "transakcijske atribute" za oceanografske podatke, so intervale podatkov preslikali v kvalitativne opise (npr. "temperatura je rahlo narasla", "slanost je znatno padla" itd.), slednje pa so potem preslikali v binarne atribute. Da bi upoštevali časovno odvisnost, so uporabili drseče okno s šestimi zaporednimi mesečnimi vrednostmi temperature in slanosti. Da bi našli pogoste transakcijske skupine atributov (ang. *itemset*) z največjo dolžino šest, avtorji predlagajo t. i. algoritem Reduced Prefix-Projected Itemset (RPPI). Avtorji so analizirali tudi učinkovitost omenjenega algoritma in ga primerjali z algoritmoma FITI [17] in APriori. Tako algoritem FITI kot RPPI imata bistveno manjšo časovno zahtevnost kot tradicionalni algoritem APriori. Pomembna prednost RPPI pred FITI je tudi v tem, da RPPI generira le pogoste skupine atributov namesto vseh možnih, kar drastično zmanjša časovno kompleksnost algoritma. Menimo, da so avtorji s predlaganim konceptom pomembno prispevali k iskanju povezovalnih pravil med oceanografskimi podatki.

2.1.5 Prostorska vizualizacija

Kot že rečeno, imamo dandanes na voljo velike količine oceanografskih podatkov, ki so bili pridobljeni z opazovanji na merilnih mestih v morju (lat. *in situ*) in s satelitskimi posnetki, ter rezultatov, ki so bili izračunani s pomočjo oceanografskih numeričnih modelov. Da lahko omenjene podatke interpretiramo, moramo imeti na voljo metode podatkovnega rudarjenja in vizualizacije podatkov. Takšne podatke same po sebi težko analiziramo zgolj vizualno, zato moramo imeti na voljo dovolj učinkovite metode. Najbolj razširjena metoda za ta namen je samoorganizirajoča se mreža (ang. *Self-Organizing Map - SOM*) [18, 19]. Na področju oceanografije je nastala že velika skupnost uporabnikov te metode, ki je v bistvu nevronska mreža, ki temelji na nenadzorovanem učenju [19]. SOM je orodje, ki se pogosto uporablja za preslikavo večdimenzionalnih podatkov v prostor z manjšim številom dimenzij, pri kateri se podobni podatkovni vzorci organizirajo v sosednje nevrone SOM. Vendar pa je potrebno pri tej metodi določiti parametre, ki so potrebni za natančno preslikavo oceanograf-

skih količin v nevrone SOM. Liu in sodelavci [20] so izvedli študijo občutljivosti in učinkovitosti SOM tako, da so spreminjali število nevronov, strukturo mreže, način inicializacije, funkcijo sosednosti in velikost šuma v podatkih. Pri tem so uporabili umetne sintetične podatke, ki opisujejo različne oblike valovanja (sinusno, žagasto, stopničasto itd.) in tudi oceanografske podatke o morskih tokovih. V omenjenem delu je možno najti koristne napotke za določanje parametrov SOM.

SOM se uporablja za različne vrste oceanografskih podatkov, kot so satelitski posnetki barve oceana in klorofila, *in situ* biološki in geokemijski podatki, temperatura na morski gladini in višina morske gladine, ki ju dobimo s pomočjo satelitskih posnetkov, morski tokovi, ki so lahko izmerjeni *in situ* ali pa so rezultat numeričnih modelov, vetrna napetost, hrapavost morskega dna, slanost na morski gladini in razlitje nafte. Liu in sodelavci [18] podajajo izčrpen seznam aplikacij SOM v oceanografiji. Vzorci, ugotovljeni s pomočjo SOM, so se izkazali za bolj natančne in intuitivne od tistih, ki so bili ugotovljeni z alternativnimi metodami t.j. z empirično ortogonalno funkcijo (EOF), analizo glavnih komponent (PCA) in z razvrščanjem v skupine z metodo voditeljev (ang. *k-means clustering*) [21–23]. Poleg tega SOM najde tudi vzorce, ki jih omenjene metode ne razkrijejo. SOM se je izkazal kot zelo robustna in zanesljiva metoda za določanje pripadnosti podatkov v skupinah.

Telszewski in sodelavci [24] so uporabili SOM za izdelavo zemljevidov porazdelitev delnega tlaka ogljikovega dioksida (pCO_2) za območje Severnega Atlantskega oceana. Oznaka pCO_2 pomeni količino raztopljenega CO_2 v morski vodi. Avtorji so uporabili SOM za rekonstrukcijo nelinearnih odvisnosti med tremi biokemijskimi količinami tj. koncentracijo klorofila, površinsko temperaturo morja in globino mešane plasti (ang. *mixed layer depth*). Biokemijski podatki so bili pridobljeni s pomočjo satelitskih opazovanj ter z reanalizo in asimilacijo podatkov iz različnih merilnih mest in plovil v Severnem Atlantskem oceanu. SOM je skupno 389.000 trojic teh podatkov razvrstila v 2220 nevronov. Podatki o pCO_2 so bili redkejši od biokemijskih podatkov (njihovo število je 137.000) in so bili pridobljeni s pomočjo plovil, ki so redno prečkala Severni Atlantski ocean. Avtorji so dobljene nevrone označili s povprečnimi vrednostmi pCO_2, ki so bile izmerjene na merilnih mestih, ki pripadajo posameznim nevronom. Povprečna napaka je bila 3,2 % povprečja vseh izmerjenih podatkov pCO_2. S pomočjo naučene SOM z označenimi nevroni so avtorji izdelali mesečne in sezonske karte porazdelitev pCO_2 za celotno območje Severnega Atlantskega oceana.

Solidoro in sodelavci [23] so uporabili SOM na multivariatnih biokemijskih po-

datkih, zbranih v severnem delu Jadranskega morja. Cilj te raziskave je bil poiskati časovno in prostorsko spremenljivost biokemičnih lastnosti (tj. parametrov kakovosti vode) v obalnem območju severnega Jadrana. Uporabili so SOM za identifikacijo manjšega števila razredov (tipov) morske vode z namenom, da bi razložili biokemijske in ekološke pojave, ki so povezani s temi razredi, in obravnavali prostorsko porazdelitev in časovni razvoj omenjenih razredov. Avtorji v svojem delu podajajo tudi napotke za metodo učenja in izbor parametrov pri SOM (tj. število nevronov, parametri učenja itd.).

Mihanović in sodelavci [25] so s pomočjo SOM prikazali vzorce površinskih morskih tokov, ki so bili izmerjeni s pomočjo mreže visokofrekvenčnih radarjev v severnem Jadranu. Pri analizi so dobili 12 različnih vzorcev na pravokotni mreži SOM z dimenzijami 4×3, kjer so trije nevroni pripadali podatkom morskih tokov pod vplivom burje, nadaljnji trije tokovom pod vplivom juga, ostali pa šibkim vetrovom in brezvetrju. Podobne rezultate so dobili tudi z upoštevanjem podatkov o vetru na površini morja iz modela ALADIN/HR, ki so jih dodali vhodnim vektorjem SOM. Tako so dobili preprosto, vendar učinkovito možnost uporabe operativnih meteoroloških produktov (npr. ALADIN/HR) za napovedovanje značilnih vzorcev površinskih morskih tokov s pomočjo prepoznavanja vzorcev, dobljenih s SOM.

Iz naštetega lahko zaključimo, da se samoorganizirajoča se mreža pogosto uporablja za razvrščanje in vizualizacijo oceanografskih podatkov. Vendar pa je izbira parametrov SOM še vedno velik izziv za oceanografske eksperte, ker lahko različna izbira parametrov vodi v različne vzorce SOM. Eksperti določajo te parametre večinoma eksperimentalno oziroma na podlagi ocene dobljenih rezultatov. To dejstvo preprečuje potencialnim novim uporabnikom, da bi bolje izkoristili to metodo. Na voljo imamo namreč obilje oceanografskih podatkov (predvsem rezultatov numeričnih modelov), ki jih je še potrebno analizirati s pomočjo SOM.

2.1.6 Zaključne misli

Lahko zaključimo, da so se metode prostorsko-časovnega podatkovnega rudarjenja zelo dobro uveljavile na področju oceanografije in ekologije. Nekatere (npr. samoorganizirajoča se mreža) se pogosto uporabljajo, druge pa imajo velik potencial za uporabo v omenjenih področjih. Večina pristopov, opisanih v tem razdelku, temelji na nenadzorovanem učenju (razvrščanje, povezovalna pravila). Prvi razlog za to je verjetno pomanjkanje orodij za nadzorovano učenje, ki bi jih lahko neposredno uporabili na

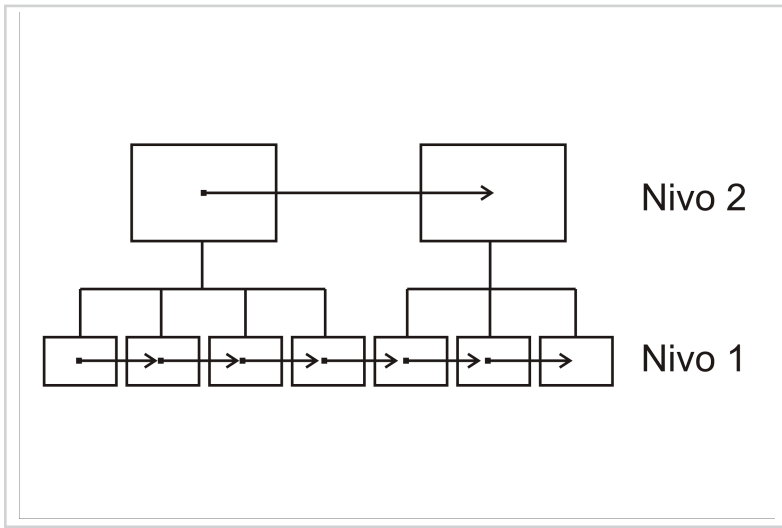
prostorskih podatkih, drugi razlog pa je kompleksnost problemov, ker je nadzorovano učenje tukaj težko opredeliti.

Naše mnenje je, da se razvoj metod prostorskega podatkovnega rudarjenja v oceanografiji nadaljuje z veliko intenzivnostjo. V prihodnosti pričakujemo še bolj učinkovite algoritme za nenadzorovano učenje in še bolj napredne hevristike za izbiro njihovih parametrov. Velik izziv predstavljajo metode razvrščanja trajektorij v oceanografiji, ker je gibanje vodnih mas v morju precej zapleteno in je odvisno od številnih dejavnikov npr. topografija morskega dna, vetrovi, rečni pritoki itd. Menimo, da je pristop s pomočjo razvrščanja delov trajektorij [1] v tej smeri zelo obetaven.

2.2 *Prostorsko-časovna povezovalna pravila in večnivojski usmerjeni grafi*

Koncept večnivojskih prostorskih povezovalnih pravil in grafov je dobro opisan v [26]. Avtorja uporabljata vizualizacijo prostorskih povezovalnih pravil s pomočjo grafov pri različnih nivojih granulacije in z različno razdrobljenostjo pravil (tj. število atomov v pogojnem in sklepnem delu pravila). Njuna metoda je zelo uporabna pri iskanju pravil, ki odražajo strukturo prostorskih objektov, na primer *okrožje*, *mesto*, *cesta* itd., in prostorske/ne-prostorske odvisnosti, ki vsebujejo prostorske predikate, kot na primer *seka*, *prečka*, *znotraj* itd. Avtorja dodatno opredelita še operatorja za posplošitev in specializacijo povezovalnih pravil. Za vizualizacijo vloge vozlišč in povezav v grafih in tudi podpore in zaupanja v povezovalnih pravilih avtorja uporabljata nasičenost barve in dolžino povezav. Avtorja podkrepita uporabnost predlagane metodologije s primerom študije podatkov s popisa prebivalstva.

V disertaciji obravnavamo metodologijo, ki razkriva bolj splošne vzorce v prostoru in času in uspešno nadgrajuje obstoječe metode Lagrangeove analize. Metoda je osnovana na omenjenih večnivojskih prostorsko-časovnih povezovalnih pravilih in usmerjenih grafih. Oceanografski podatki izkazujejo izrazite prostorske in časovne odvisnosti, zato moramo klasična povezovalna pravila [14] razširiti v prostorska-časovna povezovalna pravila [27]. Pri tem se ne ukvarjamo s strukturiranimi objekti in prostorskimi razmerji, kot je to opisano v [26, 27], ampak se osredotočimo na prostorsko in časovno granulacijo v smislu velikosti območij in dolžin časovnih intervalov. V našem primeru torej zgradimo večnivojske usmerjene grafe (slika 2.1) z različnimi nivoji granulacije v prostoru in času, kjer vozlišča predstavljajo prostorska območja različnih dimenzij (npr. od majhnih morskih območij do velikih morij), medtem ko povezave med njimi



Slika 2.1

Shematski prikaz večnivojskega usmerjenega grafa, v katerem kvadratki predstavljajo geografska območja, vodoravne puščice pa ponazarjajo prehode navideznih delcev med območji.

predstavljajo različne relacije med temi območji, na primer verjetnost prehoda Lagrangeovih delcev iz enega območja v drugo v danem časovnem intervalu. "Večnivojskost" naših grafov se torej odraža v prostorski razdelitvi problemske domene in v časovnem intervalu, v katerem opazujemo določen pojav.

3

Metodologija

V tem poglavju opisujemo metodologijo za prostorsko-časovno podatkovno rudarjenje velikega števila Lagrangeovih trajektorij oziroma poti [28, 29]. Metodologija je sestavljena iz več korakov. Najprej moramo imeti na voljo zadostno število trajektorij, predstavljenih v obliki zaporedja pozicij delcev v prostoru in času.

Definicija 1: Trajektorija je pot, ki jo opiše določen objekt (npr. namišljeni delec tekočine), ki se giblje skozi prostor kot funkcija časa. Matematično jo lahko opišemo bodisi z geometrijo poti ali pa kot položaj objekta skozi čas. Formalno jo zapišemo takole:

$$\vec{x} = (x, y, z, t) \tag{3.1}$$

kjer so x, y in z prostorske koordinate objekta v času t.

Trajektorije se pojavljajo na številnih področjih npr. sledenje vozilom, plovilom, živalim, ki so opremljene z GPS napravami itd., kar omogoča široke možnosti za podatkovno rudarjenje nastalih trajektorij. V disertaciji obravnavamo kot trajektorije bodisi poti navideznih delcev, ki smo jih spustili v izbranih točkah v hitrostnem polju oceanografskega modela, ali pa poti, ki smo jih dobili z opazovanji resničnih plovcev v morju. Pri slednjih smo zelo omejeni s številom razpoložljivih meritev, z navideznimi delci pa lahko ustvarimo veliko število (običajno več tisoč) trajektorij v domeni numeričnega modela. Iz dobljenih trajektorij dobimo prostorsko-časovna povezovalna pravila, na osnovi katerih potem konstruiramo večnivojske usmerjene grafe. V slednje vključimo še dodatne atribute različnih tipov, ki predstavljajo znanje o obravnavani domeni. V našem primeru je to oceanografsko ekspertno znanje. V naslednjih razdelkih podamo tudi smernice za izbor parametrov, ki so potrebni za tvorbo prostorsko-časovnih povezovalnih pravil in večnivojskih usmerjenih grafov. Na dobljenih grafih potem izvajamo rudarjenje z uporabo različnih metod in algoritmov, kot je opisano v poglavju 4.

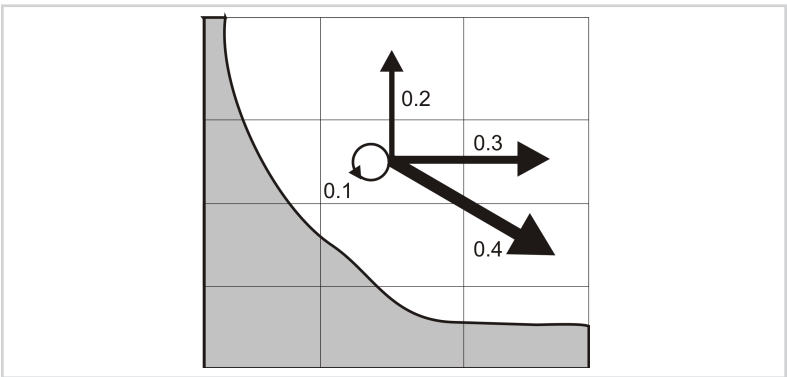
3.1 Prostorsko-časovna povezovalna pravila

Osnovno pravilo, ki ga obravnavamo, lahko formuliramo takole: "*Če se delec v določenem časovnem intervalu nahaja na območju A dane prostorske domene, potem se isti delec v naslednjem časovnem intervalu nahaja na območju B z določeno verjetnostjo*". Pri tem sta območji A in B bodisi neposredna soseda (se dotikata), bodisi leži med njima neko dru-

go območje, lahko pa tudi sovpadata (gre za isto območje). Za reševanje tega problema najprej razdelimo domeno na manjša območja in izberemo ustrezen časovni interval s pomočjo hevristike, ki jo opišemo kasneje. Nato določimo verjetnosti premikov delcev iz danega območja v sosednja s pomočjo prostorsko-časovnih povezovalnih pravil [27]. Te verjetnosti predstavlja mera zaupanja v dobljenih pravilih. Ob predpostavki, da se določen delež delcev, ki se nahajajo v območju A v času t, premakne v danem intervalu Δt iz območja A v območje B, potem sta podpora in zaupanje podana kot:

$$s(\%) = N_A/N * 100\ \% \qquad c(\%) = N_B/N_A * 100\ \% \tag{3.2}$$

kjer je N_A število delcev, ki se nahajajo na območju A v času t, medtem ko je N_B število delcev, ki so se premaknili iz območja A v območje B v časovnem intervalu Δt, N pa je skupno število delcev v celotni domeni v času t. Predlagani koncept prikazuje slika 3.1.



Slika 3.1

Prostorsko-časovna povezovalna pravila, ki se uporabljajo v predlagani metodologiji. Puščice kažejo prehode delcev iz izbranega v sosednja območja. Številke poleg puščic podajajo verjetnosti prehodov v danem časovnem intervalu (debelina puščic je sorazmerna z dano verjetnostjo). Zanka označuje verjetnost, da delci ostanejo v tistem območju v danem časovnem intervalu. Ker lahko delci bodisi zapustijo dano območje ali pa ostanejo v njem, je vsota verjetnosti za dano območje enaka 1. Območje, kjer se delci ne morejo gibati, je obarvano sivo.

3.2 Večnivojski usmerjeni grafi

Celotno množico prostorsko-časovnih povezovalnih pravil, ki so opisana v prejšnjem razdelku, lahko prikažemo v obliki večnivojskih usmerjenih grafov.

Definicija 2: Naj bo N množica vozlišč, ki predstavljajo območja v domeni modela in E množica povezav med temi vozlišči. Povezave imajo uteži w, ki imajo vrednosti mere zaupanja v prostorsko-časovnih povezovalnih pravilih oziroma verjetnosti prehodov delcev med območji (vozlišči) v danem časovnem intervalu. Večnivojski usmerjeni graf $G = (N, E, w)$ ima naslednje lastnosti:

1. G je markovski graf, kjer za vsako vozlišče velja, da je vsota uteži vseh njegovih izhodnih povezav, vključno s povezavo vozlišča samega vase, enaka 1. G je povezan z matriko prehodov markovskega procesa, kjer vozlišča grafa predstavljajo stanja, uteži povezav pa verjetnosti prehodov delcev med vozlišči (območji). Trenutna stanja so odvisna samo od stanj v prejšnjem časovnem intervalu.

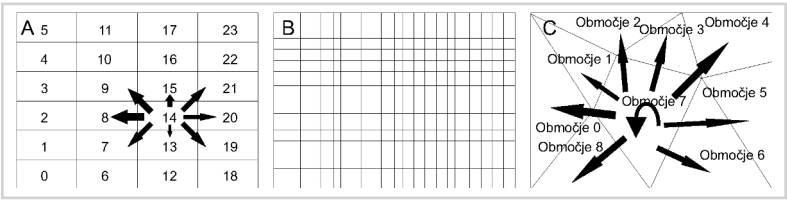
2. Časovni atributi so povezani s samim grafom G in predstavljajo njegov časovni razpon.

3. Prostorski in ne-prostorski atributi so povezani z vozlišči N. Prostorski atributi opredeljujejo obliko, velikost in lokacijo območij, ne-prostorski pa vsebujejo domensko specifično (oceanografsko) in drugo znanje, povezano s temi območji.

4. Poleg obveznega atributa w (verjetnost prehoda), imajo povezave E lahko še vrsto drugih atributov, ki predstavljajo odvisnosti med verjetnostjo prehoda in domensko specifičnimi (oceanografskimi) in drugimi količinami.

5. G je večnivojski graf, kar pomeni, da lahko domeno, ki jo pokriva, poljubno razdelimo in uporabimo ustrezen časovni interval.

Primer takšnega grafa je prikazan na sliki 5.3, kjer so vrednosti uteži povezav vozlišč samih vase sorazmerne z nasičenostjo rdeče barve, debelina puščic pa ponazarja verjetnost prehoda med območji. Na takšnem grafu lahko uporabljamo različne metode in algoritme podatkovnega rudarjenja.



Slika 3.2

Načini razdelitve domen: A) enakomerna pravokotna, B) neenakomerna pravokotna, C) poljubna

3.3 Razdelitev domene

Pred konstrukcijo večnivojskih usmerjenih grafov je potrebno definirati prostorsko razdelitev domene, kar pomeni, da moramo določiti velikosti in oblike območij, ki pripadajo posameznim vozliščem. Pri tem lahko uporabimo enakomerno pravokotno mrežo (Slika 3.2 A), neenakomerno pravokotno mrežo (slika 3.2 B) ali pa poljubno mrežo, sestavljeno iz mnogokotnikov (slika 3.2 C). Razdelitev domene na območja je odvisna od problema, ki ga želimo rešiti z uporabo večnivojskih usmerjenih grafov. Če želimo imeti zelo fino razdelitev, je najbolje uporabiti enakomerno pravokotno mrežo (slika 3.2 A) ali pa neenakomerno pravokotno mrežo (slika 3.2 B) v primeru, ko želimo, da so nekatera območja razdeljena bolj fino, druga pa bolj grobo. Poljubna razdelitev se uporablja za manjše število velikih območij (morij), kjer je možno opraviti tako razdelitev ročno. Uporabnik mora definirati tako razdelitev, ki je najbolj prilagojena njegovim potrebam.

Potem ko smo razdelili domeno na območja, je potrebno vsakemu vozlišču (območju) določiti enolično oznako. V naši raziskavi uporabimo enakomerno pravokotno mrežo in vozlišča označimo z zaporednimi številkami od 0 to $n-1$, kjer je n število vseh vozlišč. Z označevanjem začnemo v spodnjem levem kotu in se pomikamo od leve proti desni po stolpcih od spodaj navzgor (glej sliko 3.2 A). Z namenom, da zmanjšamo število vozlišč, upoštevamo le tista, ki so v našem interesu tj. vozlišča, ki delno ali v celoti vsebujejo morska (“mokra”) območja. Na podoben način lahko označimo vozlišča v neenakomerni pravokotni mreži. Za poljubno razdelitev v oceanografiji, kjer imamo opraviti z majhnim številom območij (ponavadi so to posamezna morja), predlagamo njihovo označitev oziroma poimenovanje z zemljepisnimi imeni. Uporaba tako označenih vozlišč je enaka kot v prej opisanih primerih.

3.4 Tvorba prostorsko-časovnih povezovalnih pravil

Za tvorbo prostorsko-časovnih povezovalnih pravil lahko uporabimo algoritem APriori [14], ki so ga prvotno uporabili za analizo nakupovalnih navad kupcev in sicer za odkrivanje pogostih podmnožic artiklov v končni množici transakcij. Za našo uporabo moramo najprej pretvoriti premike delcev iz enega območja v drugo v danem časovnem intervalu Δt v primere oziroma transakcije, ki jih potem obdela algoritem APriori. Transakcije definiramo v obliki četverk *(t, pozicija(t), globina(t), pozicija(t+Δt))*, kjer je spremenljivka *pozicija(t)* tipa *točka(x, y)* s prostorskimi koordinatami *x* in *y*, *t* pa podaja informacijo o času. V tem delu se brez izgube na splošnosti metodologije, omejimo na *t = leto, mesec*. Δt je časovni interval, ki je ponavadi merjen v dnevih, v katerem delci bodisi ostanejo na določenem območju ali pa se premaknejo v drugo območje. Optimalna izbira časovnega intervala je opisana razdelku 3.5. Transakcije oziroma primere izluščimo iz podatkovne baze trajektorij, kar pomeni, da v trajektorijah vzorčimo pare točk, ki se razlikujejo za časovni interval Δt. Tako je število primerov N_e, ki jih dobimo na takšen način, enako:

$$N_e = n_t(l_t - \Delta t) \tag{3.3}$$

kjer je n_t skupno število trajektorij in l_t dolžina posamezne trajektorije. Dolžina l_t je enako kot Δt izražena v časovnih enotah (ponavadi v dnevih). V nadaljevanju preslikamo pozicije v primerih (transakcijah) v območja in dobimo četverke *(leto, mesec, območje(t), območje(t+Δt))* s pomočjo postopka, ki je znan v računalniški geometriji kot iskanje mnogokotnika, v katerem se nahaja dana točka (ang. *point location*). To pomeni, da moramo za dano razdelitev domene na disjunktna območja določiti območje, kjer leži izbrana točka [30]. Te naloge se lahko lotimo z "grobo silo" tj. z zaporednim izčrpnim preiskovanjem vseh območij, da bi našli tisto, kateremu pripada dana točka. Ta pristop je uporaben, kjer imamo opraviti z majhnim številom po obliki razmeroma kompleksnih območij, kot so npr. v oceanografiji velika morja z zapleteno obalno črto. V primeru, ko imamo domeno razdeljeno na mnogokotnike kot v primeru na sliki 3.2 C), je problem iskanja mnogokotnika, v katerem se nahaja točka, rešljiv z algoritmi, za katere podajamo angleška poimenovanja, npr. *slab decomposition* [31], *monotone subdivisions* [32], *triangulation refinement* in *trapezoidal decomposition*. Vsi ti algoritmi imajo časovno kompleksnost *O(log n)*, kjer je *n* število vseh mnogokotnikov v dani domeni. Pri neenakomerni pravokotni razdelitvi lahko uporabimo binarno iskanje ali bisekcijo na obeh prostorskih oseh (časovna kompleksnost *O(log n)*). Za enakomerno

pravokotno razdelitev se iskanje lokacije točke zreducira na preslikavo (glej sliko 3.2 A):

$$i = \lfloor x - x_0 \rfloor / \Delta x \quad j = \lfloor y - y_0 \rfloor / \Delta y \quad L_k = n_y i + j \tag{3.4}$$

kjer sta x in y prostorski koordinati dane točke, x_0 in y_0 sta koordinati spodnjega levega kota domene, $i \in 0 \ldots n_x - 1$ je indeks območja v smeri osi x, $j \in 0 \ldots n_y - 1$ je indeks v smeri osi y, Δx in Δy pa sta dimenziji posameznih območij v smeri x in y. L_k je oznaka (zaporedna številka) območja, če predpostavimo ureditev vozlišč, kot je prikazano na Sliki 3.2 A, kjer so L_k, $k \in 0 \ldots n - 1$, oznake vozlišč. Tako je dobljena konstantna časovna kompleksnost O(1). Skupna časovna kompleksnost pri tvorbi N primerov je potem *NO(f(n))*, kjer je *f(n)* bodisi *n* za poljubno razdelitev, *log(n)* za neenakomerno pravokotno razdelitev ali pa 1 za enakomerno pravokotno razdelitev.

Iz dobljenih primerov oziroma transakcij tvorimo prostorsko-časovna povezovalna pravila ob izbiri primerne minimalne podpore. Za potrebe aplikacije v oceanografiji, ki je opisana v 5. poglavju, se izkaže, da je primerna vrednost minimalne podpore enaka 10^{-6}. Posledica tega je, da zanemarimo povezave, ki imajo zelo majhno podporo in se le redko pojavljajo ter se zaradi tega se ne poslabša natančnost analiz, ki jih pozneje opravimo s pomočjo večnivojskih usmerjenih grafov. Iz številnih dobljenih povezovalnih pravil pa izluščimo le tista, iz katerih lahko zgradimo večnivojske usmerjene grafe, kjer posamezen graf pokriva obdobje enega meseca, leta ali celotnega obdobja (več let). Uporabimo torej prostorsko-časovna pravila v eni izmed naslednjih oblik:

$$year(t) \wedge month(t) \wedge area(t) \Longrightarrow area(t + \Delta t) \quad (s(\%), c(\%)) \tag{3.5}$$

$$year(t) \wedge area(t) \Longrightarrow area(t + \Delta t) \quad (s(\%), c(\%)) \tag{3.6}$$

$$area(t) \Longrightarrow area(t + \Delta t) \quad (s(\%), c(\%)) \tag{3.7}$$

kjer sta s (%) in c (%) podpora in zaupanje povezovalnega pravila, ki ju izračunamo s pomočjo enačb (3.2).

3.5 Izbira časovnega intervala

Naslednja naloga, ki se je lotimo, je izbira ustreznega časovnega intervala Δt, za katerega ugotavljamo verjetnost prehodov delcev med območji. Časovni interval Δt je odvisen od velikosti območij in od hitrosti delcev v teh območjih. Zaradi enostavnosti in bolj

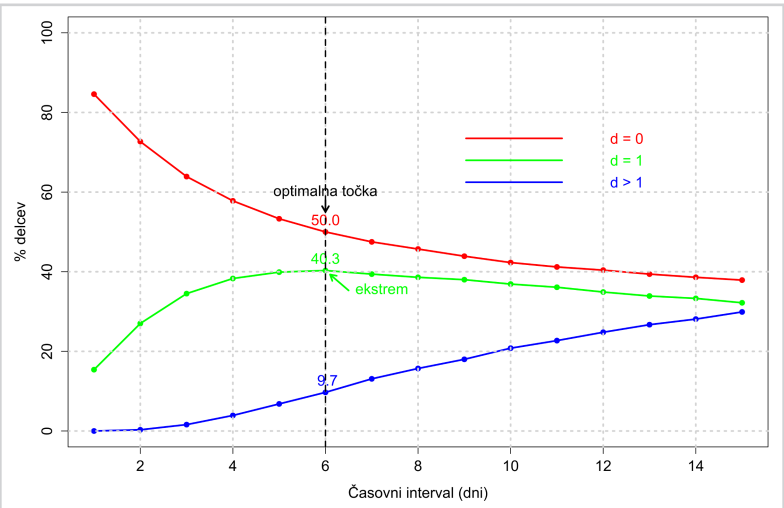
nazorne vizualizacije je za večnivojski usmerjeni graf zaželeno, da ima čim več povezav neposredno v sosednja in čim manj povezav v bolj oddaljena območja. Idealno naj bi bil večnivojski usmerjeni graf čimbolj podoben "grafu sosedov", v katerem definiramo povezave na naslednji način:

$$e_{ij} = \begin{cases} 1 & i = j \quad \vee \quad dotika(v_i, v_j) \\ 0 & sicer \end{cases} \tag{3.8}$$

kjer predikat $dotika(v_i, v_j)$ pomeni, da imata vozlišči v_i in v_j bodisi skupno stranico ali pa vogal pripadajočih mnogokotnikov. Pravilo "preko palca" za določitev primernega časovnega intervala za aplikacijo v oceanografiji, ki je opisana v 5. poglavju (ki je zlahka lahko posplošimo na katerokoli drugo aplikacijsko domeno), bi lahko bilo podano v obliki enačbe:

$$\Delta t\,(dni) = \frac{d_{povp}\,(m)}{v_{povp}\,(m/s) \times 3600\,(s/h) \times 24\,(h/dan)} \tag{3.9}$$

kjer je d_{povp} povprečna razdalja med pari sosednjih vozlišč v metrih, v_{povp} povprečna hitrost (m/s) delcev ali morskih tokov v domeni, Δt pa je podan v dnevih. Drug način bi bil izčrpno generiranje vseh možnih večnivojskih usmerjenih grafov z enako prostorsko razdelitvijo z različnimi časovnimi intervali Δt, kar bi bilo časovno potratno. Sami uporabimo drugačen pristop. Najprej izberemo primeren interval $(\Delta t_{min}, \Delta t_{max})$ in za vsak Δt iz tega intervala naključno izberemo manjšo podmnožico primerov in izračunamo število delcev, ki 1) ostanejo v istem območju, 2) se premaknejo v sosednje območje in 3) se premaknejo dlje kot v sosednja območja. S povečevanjem Δt se zmanjšuje delež delcev, ki ostanejo v istem območju; delež delcev, ki preidejo v sosednja območja in še dlje, pa narašča. Število delcev, ki v danem časovnem intervalu preidejo v sosednja območja in tam ostanejo, narašča do določene maksimalne vrednosti, nato pa začne upadati. Vrednost Δt pri tem maksimumu je optimalna izbira, ki je v našem primeru 6 dni (glej sliko 3.3).



Slika 3.3

Odstotek števila delcev, ki: 1) ostanejo v istih območjih (rdeča), 2) se premaknejo v sosednja območja (zelena) in 3), ki se premaknejo v oddaljena območja (modra), v odvisnosti od časovnega intervala. Količina d ponazarja število korakov po povezavah grafa, ki jih opravijo delci. Optimalna izbira Δt za dano prostorsko razdelitev 60 x 30 (glej razdelek 5.1) je 6 dni, pri katerem je največji odstotek delcev, ki se premaknejo v sosednja območja (zelena krivulja). Slika je bila dobljena pri ovrednotenju metodologije s pomočjo rezultatov oceanografskega numeričnega modela Mediterranean Ocean Forecasting System (MFS).

4

Algoritmi na večnivojskih usmerjenih grafih

Na večnivojskih usmerjenih grafih, ki so opisani v poglavju 3, je možno razviti ali uporabiti številne algoritme za rudarjenje teh grafov. Slednje je zelo široko področje, katerega namen je iskanje znanja v podatkih, ki so predstavljeni v obliki grafa. Omenjeno področje obravnava številne metode in pristope pri rudarjenju grafov, npr. analiza povezav (ang. *link analysis*), rudarjenje pogostih podgrafov (ang. *frequent subgraph mining*), razvrščanje grafov (ang. *cluster analysis*), uvrščanje (ang. *classification*), redukcija dimenzij (ang. *dimensionality reduction*), detekcija anomalij na osnovi grafov (ang. *graph based anomaly detection*) itd. Poleg številnih člankov si lahko bralec ogleda izčrpen opis omenjenih metod v [33] in [34]. V disertaciji se omejimo na iskanje pogostih podstruktur v večnivojskih usmerjenih grafih in prikažemo algoritme za iskanje najbolj verjetnih poti in ciklov v omenjenih grafih. V razdelku 4.1 opišemo algoritem, ki v večnivojskem usmerjenem grafu, ki velja za določeno obdobje npr. en mesec, poišče enostavne cikle, pri katerih preučujemo njihovo pogostost v različnih obdobjih (mesecih) [28]. V razdelku 4.2 ta algoritem nadgradimo z namenom iskanja t. i. *dinamičnih mehkih poti* in *ciklov*, ki se raztezajo skozi daljše obdobje oziroma skozi časovno vrsto večnivojskih usmerjenih grafov [35].

Naštete postopke lahko med drugim uporabimo v oceanografiji za iskanje značilnih vzorcev gibanja vodnih mas tj. poti, ki se razvijejo v daljših obdobjih (več mesecev, vendar manj kot eno leto) in ciklov (kroženje vodnih mas, ki se pojavlja vzdolž omenjenih poti).

4.1 Iskanje enostavnih ciklov v grafih

Kot idejo za rudarjenje večnivojskih usmerjenih grafov, kar v našem primeru pomeni iskanje značilnih poti in ciklov v grafih, vzamemo algoritem za odkrivanje skupin vozlišč na osnovi izmenjave oznak (ang. *label propagation clustering*), ki se pogosto uporablja za rudarjenje socialnih omrežij [36, 37]. V tem algoritmu ima vsako vozlišče na začetku enolično oznako, ki jo algoritem na vsakem koraku spremeni v oznako, ki jo ima večina sosedov danega vozlišča. Algoritem iterativno ponavlja ta postopek, dokler ni dosežen konsenz med vozlišči in se oznake več ne spreminjajo. Na osnovi tega smo razvili algoritem za odkrivanje skupin vozlišč na osnovi izmenjave oznak v večnivojskih usmerjenih grafih. Algoritem vozlišču namesto oznake večine sosednjih vozlišč dodeli oznako sosednjega vozlišča, iz katerega izhaja vhodna povezava z največjo utežjo v tekoče vozlišče ali pa izhodna povezava v sosednje vozlišče, ki ima ravno tako največjo utež. V našem primeru algoritem po vrsti pregleduje vozlišča in vsakemu do-

deli oznako sosednjega vozlišča, iz katerega izhaja vhodna povezava v tekoče vozlišče, ki ima največjo utež (verjetnost prehoda). Nastali algoritem ima za našo specifično vrsto grafa pričakovano časovno zahtevnost *O(nd)* oziroma *O(m)*, kjer je *n* število vozlišč in *m* število povezav v grafu. Iz tega algoritma izpeljemo algoritme za iskanje pogostih poti in ciklov v večnivojskih usmerjenih grafih. Algoritem 1, ki poišče vse posamezne cikle v določenem večnivojskem usmerjenem grafu, je prikazan na sliki 4.1 in ga opišemo v nadaljevanju. Kasneje poiščemo pogoste cikle s pomočjo principa APriori, za kar smo dobili idejo pri algoritmih za iskanje pogostih podgrafov [38, 39].

Na začetku so vsa vozlišča kandidati za pripadnost v ciklih ali poteh in nobeno od vozišč še ni bilo obiskano. K atributom povezav dodamo zaporedno številko cikla (*cycle.number*) in tip cikla (*cycle.type*), ki sta na začetku nedefinirana. Tip cikla je lahko *ciklonski* (vrti se v nasprotni smeri urinega kazalca), anticiklonski (v smeri urinega kazalca)[1] ali pa "nedefiniran". Slednji je rezerviran za cikel, ki vsebuje samo dve vozlišči, in zato algoritem ne more ugotoviti smeri cikla, ki se določa z izračunom psevdo-površine[2] poligona, ki ga obkroža cikel. Če ima psevdo-površina pozitiven predznak, je cikel ciklonski, sicer je anticiklonski. Algoritem obišče vsako od vozlišč, ga označi kot obiskano (*visited*) in potisne na sklad. Nato izbere naslednika tega vozlišča, do katerega kaže povezava z največjo utežjo (verjetnostjo prehoda). Če naslednik obstaja in še ni bil obiskan, ga algoritem ravno tako označi kot obiskan (*visited*) in potisne na sklad. Algoritem ponavlja ta postopek, dokler ne naleti na vozlišče, ki je bilo že obiskano, kar pomeni, da je našel cikel. Nato jemlje vozlišča (predhodnike) iz sklada in jih označi z zaporedno številko cikla, ki ga je našel. Pri tem algoritem koraka po ciklu v nasprotni smeri, dokler ponovno ne naleti na vozlišče *cycle.begin* in tako zaključi cikel. Pri tem izračuna psevdo-površino poligona, ki obdaja cikel, s pomočjo formule [40]:

$$A = \frac{1}{2}\left[(x_1 - x_2)\left(y_1 + y_2\right) + (x_2 - x_3)\left(y_2 + y_3\right) + \ldots + (x_n - x_1)\left(y_n + y_1\right)\right] \tag{4.1}$$

kjer točke $P_1(x_1,y_1)$, $P_2(x_2,y_2)$, …, $P_n(x_n,y_n)$ predstavljajo masna središča vozlišč (območij), vsebovanih v ciklu, x_i in y_i pa so njihove prostorske koordinate. Točka $P_1(x_1,$

[1] To velja za severno poloblo; na južni polobli se ciklonski vrtinci vrtijo v smeri urinega kazalca, anticiklonski pa v nasprotni smeri.

[2] To ni prava površina, ker je izračunana neposredno iz geografskih koordinat namesto iz dolžinskih enot.

y_1) predstavlja začetek in konec cikla (*cycle.begin*). Če ima izračunana psevdo-površina pozitiven predznak, je cikel ciklonski, sicer je anticiklonski. Če je psevdo-površina enaka 0, potem je cikel "nedefiniran". To se zgodi pri tvorbi cikla iz dveh vozlišč, ki sta prostorsko blizu in med katerima sta dve vzajemni povezavi z največjo utežjo. Povezave preostalih vozlišč v skladu ne tvorijo cikla ampak pot.

Če za dano vozlišče (kandidata) algoritem po opisani metodi ne najde cikla, je to posledica enega od naslednjih dogodkov: ali je naletel na vozlišče, ki nima izhodnih povezav, ali pa je prišel do vozlišča, ki ga je že prej obdelal.

Algoritem obišče vsako vozlišče samo enkrat in najde povezave z največjimi utežmi, ki kažejo na naslednike. Povprečna časovna zahtevnost algoritma je *O(nd)* = *O(m)*, kjer je *n* število vozlišč, *d* povprečno število izhodnih povezav vozlišč in *m* število povezav. Poleg tega pa vseh *n* vozlišč algoritem natanko enkrat shrani na sklad in vzame iz njega in istočasno dodeli povezavam atributa *cycle.number* in *cycle.type*. Tako je skupna pričakovana časovna zahtevnost *O(m+n)*. Prej omenjeni algoritmi za odkrivanje skupin vozlišč na osnovi izmenjave oznak [36, 37] so bolj splošni in dosti bolj časovno zahtevni za uporabo na večnivojskih usmerjenih grafih. To nalogo opravijo v *t* iteracijah in imajo časovno zahtevnost *O(nt)*, kjer je *n* število vozlišč v grafu.

Opisani algoritem lahko najde številne cikle na istih območjih v različnih časovnih obdobjih. Ti cikli so si podobni, vendar niso povsem enaki, kar pomeni, da se razlikujejo v nekaterih vozliščih in pripadajočih povezavah. Da bi našli pogoste identične cikle, uporabimo princip APriori. Pri tem najprej pretvorimo cikle v grafih v "transakcije". Za cikel, ki vsebuje vozlišča v_1, v_2, $\ldots v_n$, zapišemo transakcijo kot:

$$\left\{cycleInfo, (v_1 v_2), (v_2 v_3), \ldots, (v_{n-1} v_n), (v_n v_1)\right\} \tag{4.2}$$

kjer so $(v_i v_j)$ po vrsti povezave med vozlišči v ciklu z dolžino *n*, koda *cycleInfo* pa zagotavlja informacijo o dolžini in tipu cikla tj. ciklonski, anticiklonski, "nedefiniran" ali pa enostavna pot.

Algoritem APriori tvori pogoste skupine povezav, ki se pojavljajo v ciklih, in jih uredi v padajočem vrstnem redu vrednosti podpore. Te skupine so sestavljene tako iz pogostih delov ciklov kot tudi iz popolnih ciklov, kjer je njihovo število je odvisno od dane minimalne podpore. Naš namen je najti samo popolne cikle, zato v ta namen uporabimo kodo *cycleInfo*. Pri tem primerjamo dolžino vsakega rezultata (skupine povezav) z dolžino pripadajočega cikla, ki je zapisana v *cycleInfo*. Če se ti dolžini ujemata, potem je rezultat pogosti popolni cikel. Časovna zahtevnost te faze brez predhodnega

Algoritem 1 - poišče cikle v večnivojskem usmerjenem grafu

Vhod: Graf $G\,(N, E)$ z utežmi w
Izhod: Graf $G_{out}\,(N, E_{out})$ z utežmi w_{out}

```
1:  for all n ∈ N do
2:      visited_n ← False
3:      candidate_n ← True
4:  end for
5:  for all e ∈ E do
6:      cycle.number_e ← NA
7:      cycle.type_e ← NA
8:      // "ciklonski", "anticiklonski", "nedefiniran", "pot"
9:  end for
10: s ← stack()
11: cycle.seq ← 1
12: for all n ∈ N do
13:     if candidate_n then
14:         visited_n ← True
15:         found.visited ← False
16:         found.blind ← False
17:         begin.cycle ← NA
18:         push(s, n)
19:         curr ← n
20:         // n je tekoče vozlišče
21:         while ¬ (found.visited ∨ found.blind) do
22:             succ ← argmax_j (w_{j,curr}) ,  j ≠ curr
23:             // naslednik z največjo utežjo
24:             if exists_succ then
25:                 if candidate_succ then
26:                     if visited_succ then
27:                         found.visited ← True
28:                         begin.cycle ← succ
29:                     else
30:                         visited_succ ← True
31:                         push(s, succ)
32:                     end if
33:                     curr ← succ
34:                 else
35:                     found.blind ← True
36:                     // naslednik je bil že uporabljen
37:                     push(s, succ)
38:                 end if
39:             else
40:                 found.blind ← True
41:                 // ne najde naslednika
42:             end if
43:         end while
```

Slika 4.1

Algoritem 1 – poišče vse enostavne cikle v večnivojskem usmerjenem grafu (se nadaljuje na naslednji strani).

```
        found.begin ← False
        area ← 0
        prepred ← begin.cycle
        // pred-predhodnik predstavlja začetek cikla
        while ¬empty(s) do
            pred ← pop(s)
            if ¬(found.blind ∨ found.begin) then
                cycle.number_{pred,prepred} ← cycle.seq
                segment ← izračun površine segmenta
                area ← addSegment(area, segment)
                // segment doda skupni površini cikla
                if pred == begin.cycle then
                    found.begin ← True
                    if area > 0 then
                        cycle.type ← "cyclonic"
                    else if area > 0 then
                        cycle.type ← "anticyclonic"
                    else
                        cycle.type ← "undefined"
                        // area == 0
                    end if
                    cycle.seq ← cycle.seq + 1
                    // naslednji cikel
                else
                    if ¬found.blind then
                        cycle.type_{e_{pred,prepred}} ← "path"
                    else
                        if exists_{(pred,preped)} then
                            cycle.type_{e_{pred,prepred}} ← "path"
                        end if
                    end if
                end if
            end if
            eout_{pred,prepred} ← e_{pred,prepred}
            candidate_{pred} ← False
            prepred ← pred
        end while
    end if
end for
return G_out
```

Algoritem 1 – nadaljevanje s prejšnje strani

izvajanja algoritma APriori je $O(m)$, kjer je m število rezultatov algoritma APriori. Slednji zagotavlja tudi seznam transakcij, tj. obdobja (mesece), v katerih se pojavljajo ti cikli.

4.2 Iskanje dinamičnih mehkih poti in ciklov

V prejšnjem razdelku opišemo algoritem, ki poišče enostavne cikle znotraj posameznih večnivojskih usmerjenih grafov. Vendar pa dobljeni cikli veljajo samo za krajše obdobje (en mesec). Razlog za to je, da se hitrostno polje v numeričnem modelu stalno spreminja, in zato moramo s pomočjo povezovalnih pravili združevati dnevne rezultate modela v mesečne večnivojske usmerjene grafe s konstantnimi utežmi povezav. Zato je potrebno algoritem iz prejšnjega razdelka nadgraditi z namenom, da bi našli poti in cikle, ki se razvijejo v daljših obdobjih, in sicer več mesecev, vendar manj kot eno leto. Poleg tega združujemo nove poti in cikle v skupine (ang. *clusters*), ki jih imenujemo *dinamične mehke poti* in *cikli*. Postopek iz razdelka 4.1 razširimo tudi tako, da poti iz vsakega vozlišča v vsakem mesečnem grafu v časovni vrsti grafov začnemo v vsakem časovnem intervalu Δt znotraj mesecev in s tem pridobimo še večje število poti (in tudi ciklov). Po drugi strani pa namesto iskanja poti v smeri povezave z največjo utežjo, kar ustreza požrešnemu iskanju (ang. *greedy search*), uporabimo iskanje nekaj najboljših poti v smeri povezav z največjimi utežmi. Uporabimo torej iskanje v snopu (ang. *beam search*).

V tem razdelku opisujemo algoritme, ki smo jih razvili z namenom iskanja dinamičnih poti in ciklov v večnivojskih usmerjenih grafih [35]. S pomočjo teh algoritmov iščemo najbolj verjetne poti z začetkom v vsakem vozlišču v časovni vrsti grafov. V poglavju 5 je opisana aplikacija, kjer je časovna vrsta grafov tvorjena iz rezultatov oceanografskega numeričnega modela za določeno obdobje. Te poti nastajajo v daljših obdobjih in so zato sestavljene iz povezav, ki pripadajo različnim (zaporednim) grafom v časovnih vrsti. Zaradi slednjega te poti imenujemo *dinamične*. Iz dobljenih dinamičnih poti potem z algoritmi izločimo cikle in slednje uvrstimo glede na smisel vrtenja v ciklonske (proti smeri urinega kazalca), anticiklonske (v smeri urinega kazalca) in t. i. "nedefinirane" cikle s po dvema vozliščema in dvema povezavama. Slednji so skrčeni v daljico in pri njih ne moremo definirati smisla vrtenja. Dobljene dinamične poti združujemo na osnovi skupnih povezav v t. i. *dinamične mehke poti*, dinamične cikle pa na osnovi površin območij, ki ustrezajo skupnim vozliščem, v t. i. *dinamične mehke cikle*.

Definicija 3: Dinamične mehke strukture predstavljajo najbolj značilne poti in cikle v daljšem časovnem obdobju. Dobimo jih s sprehodom po najverjetnejših povezavah v časovni vrsti večnivojskih usmerjenih grafov in hierarhičnim razvrščanjem dobljenih poti in ciklov v skupine. Strukture so *dinamične*, ker se uteži povezav, iz katerih so sestavljene, s časom spreminjajo in *mehke*, ker strukture združujemo v skupine na osnovi izbrane medsebojne razdalje.

4.2.1 Ozadje

Pri opisovanju algoritmov v naslednjih razdelkih se opremo na ovrednotenje metodologije, ki je opisano kasneje v razdelku 5.1. Pri konstrukciji dinamičnih mehkih poti in ciklov uporabimo 156 večnivojskih usmerjenih grafov, ki smo jih dobili iz rezultatov modela MFS [5, 9, 10] za vsak mesec v obdobju 1999–2011 (glej razdelek 5.1 in sliko 5.3). Omenjeni grafi imajo po 848 vozlišč in v povprečju okoli 4.000 povezav. Slednje so utežene z verjetnostmi prehodov med vozlišči v časovnem intervalu $\Delta t = 6$ dni. Ker vzamemo konstantno dolžino meseca 30 dni, so torej vrednosti uteži konstantne za pet prehodov v obdobju enega meseca. To moramo upoštevati pri konstrukciji dinamičnih poti in ciklov, kjer moramo pri vsakem koraku vzdolž poti uporabiti uteži ustreznega večnivojskega usmerjenega grafa, ki velja za določen mesec. Vse tako dobljene dinamične mehke strukture uporabimo za primerjavo s strukturami, ki so jih definirali oceanografski eksperti na osnovi opazovanj ali rezultatov numeričnih modelov. Vidimo, da so dobljene strukture zelo podobne tistim, ki jih opisuje oceanografska literatura, poleg tega pa z našimi metodami dobimo tudi strukture, ki jih oceanografski eksperti morajo še potrditi.

4.2.2 Konstruiranje najbolj verjetnih poti

Preden se lotimo konstrukcije dinamičnih mehkih poti in ciklov, moramo najprej poiskati vse najbolj verjetne dinamične poti v časovni vrsti usmerjenih grafov, ki jih imamo na voljo. V ta namen smo razvili algoritem, ki konstruira takšne poti z začetkom v vsakem vozlišču in v petih časovnih intervalih znotraj vsakega meseca. Časovni intervali so med seboj razmaknjeni za $\Delta t = 6$ dni in ob predpostavki, da vzamemo za vsak mesec konstantno dolžino 30 dni, začnemo s konstrukcijo poti vsak 1., 7., 13., 19. in 25. dan v mesecu. Pri tem se pomikamo v smereh najverjetnejših povezav grafih in pri tem uporabimo *iskanje v snopu* [15]. Na ta način hkrati iščemo M najverjetnejših poti,

ki se nadaljujejo iz danega vozlišča v danem časovnem intervalu (M je širina snopa). Kasneje dobljene poti hierarhično razvrstimo z namenom, da dobimo primerno število skupin (dinamičnih mehkih poti), ki jih primerjamo s strukturami, ki jih poznajo oceanografski eksperti.

Kot rečeno, so najbolj verjetne poti zgrajene tako, da imajo svoje začetke v vsakem vozlišču v časovnih intervalih $t_{init} = 0,\ \Delta t,\ 2 \times \Delta t,\ \ldots,\ T \times \Delta t$, kjer je T število vseh časovnih intervalov Δt v celotnem obdobju, za katerega imamo na voljo rezultate numeričnega modela MFS v obdobju 1999–2011. V tem primeru je T enak 156×5 6-dnevnih intervalov na mesec = 780. Pri konstrukciji poti se v vsakem časovnem koraku iz posameznega vozlišča pomaknemo v M najbolj verjetnih smereh in tako hkrati iščemo M najbolj verjetnih poti. Zaradi boljše učinkovitosti algoritma predstavimo večnivojske usmerjene grafe v obliki seznama M sosedov, ki so urejeni padajoče po velikostih uteži povezav (verjetnostih prehoda iz tekočega v sosednje vozlišče). Seznam sosedov predstavimo na naslednji način:

$$v_i : ((v_{j1}, w_{j1}), (v_{j2}, w_{j2}), \ldots, (v_{jM}, w_{jM})) \tag{4.3}$$

kjer je v_i i-to vozlišče grafa, $i = 1 \ \ldots \ N$, N je število vozlišč, v_{jk}, $k = 1 \ \ldots \ M$, pa so nasledniki i-tega vozlišča, ki so urejeni padajoče po velikostih uteži povezav w_{jk}. Tukaj izpustimo prehod iz danega vozlišča v samo vase tj. $v_i \neq v_{jk}$, $k = 1 \ \ldots \ M$. Na ta način je časovna zahtevnost dostopa do vsakega vozlišča in njegovih naslednikov enaka $O(1)$. Vozlišča brez naslednikov imenujemo "zastojna" vozlišča, ker pri njih velja večja verjetnost, da delci v danem časovnem intervalu tam ostanejo kot pa da se premaknejo v sosednja vozlišča.

Algoritem 2 (slika 4.2) najde vse dinamične poti in jih vrne bodisi kot zaporedje[3] vozlišč P^V ali pa kot zaporedje povezav P^E. Pri prehodih iz enega vozlišča v drugo algoritem upošteva uteži w povezav grafa (verjetnosti prehoda), ki pripada določenemu mesecu t_{curr}. Pri konstrukciji določene poti se algoritem ustavi bodisi v primeru, ko naleti na "zastojno" vozlišče ali pa je pretekel že vse časovne intervale od 0 do $(T-1)\times\Delta t$. Največje število poti, ki jih algoritem lahko najde, znaša $N \times T \times M$. Pričakovana časovna zahtevnost algoritma je $O(NTMl_p)$, kjer je l_p povprečna dolžina vseh poti, ki jih algoritem najde.

[3]Nadpisana V in E tukaj označujeta način predstavitve poti (kot zaporedje vozlišč ali povezav) in ne pomenita eksponent.

Algoritem 2 Iskanje dinamičnih poti v snopu

Vhod: Množica mesečnih večnivojskih usmerjenih grafov $G(V, E, t)$, $t = 1 \ \ldots \ t_{max}$; širina snopa M
Izhod: Seznam dinamičnih poti kot zaporedje vozlišč P^V ali povezav P^E

1: $hops.per.month \leftarrow 30/\Delta t$ // Δt je podan v dnevih
2: $P^V \leftarrow \emptyset$, $P^E \leftarrow \emptyset$
3: $n_{int} \leftarrow t_{max} \times hops.per.month$
4: **for** $int_0 = 0 \rightarrow (n_{int} - 1)$ **do**
5: $t_0 \leftarrow floor(int_0 \,/\, hops.per.month)$ // začetni mesec
6: **for all** $n \in V$ **do**
7: $curr \leftarrow \{n\}$, $p^v \leftarrow curr$, $p^e \leftarrow \emptyset$, $path.len \leftarrow 0$
8: **while** $\neg empty(curr)$ & $(int_0 + path.len) < (n_{int} - 1)$ **do**
9: $t_{curr} \leftarrow t_0 + floor(path.len \,/\, hops.per.month)$ // tekoči mesec
10: $path.len \leftarrow path.len{+}1$
11: $succ \leftarrow successors(curr, t_{curr})$ // nasledniki za tekoči mesec
12: $succ \leftarrow sort.descend(succ)$ // padajoče po vrednosti uteži
13: $best \leftarrow succ_{1,\ldots,M}$
14: **if** $\neg empty(best)$ **then**
15: $p^v \leftarrow p^v \cup best$, $p^e \leftarrow p^e \cup e_{curr,best}(t_{curr})$
16: **end if**
17: $curr \leftarrow best$
18: **end while**
19: **if** $\neg empty(p^e)$ **then**
20: $P^V \leftarrow P^V \cup p^v$, $P^E \leftarrow P^E \cup p^e$
21: **end if**
22: **end for**
23: **end for**
24: **return** P^V, P^E

Slika 4.2

Algoritem 2 – iskanje najbolj verjetnih dinamičnih poti v večnivojskih usmerjenih grafih. Algoritem uporablja iskanje v snopu.

4.2.3 *Dinamične mehke poti*

Definicija 4: Dinamična mehka pot je množica poti, ki so si med seboj podobne. Kot mero podobnosti uporabimo utežena najdaljša skupna podzaporedja [41].

V nadaljevanju opišemo postopek za določanje dinamičnih mehkih poti s pomočjo hierarhičnega razvrščanja posameznih poti, ki smo jih dobili s pomočjo Algoritma 2 (slika 4.2). Tukaj uporabimo razdaljo ROUGE-W, ki temelji na iskanju uteženih najdaljših skupnih podzaporedij (ang. *weighted longest common subsequence – WLCS*) povezav med posameznimi pari poti [41, 42]. Znotraj posameznih poti lahko obstaja več takih zaporedij (glej sliko 4.3). Najprej izračunamo razdaljo WLCS med danim parom poti, kot to opisujejo Lin-Och in sodelavci [41]. Vhod v algoritem Lin-Och so pari poti, kjer so slednje predstavljene kot zaporedja povezav grafov P^E (glej konstrukcijo P^E v algoritmu 2). Želeno mero različnosti d_{WLCS} dobimo na koncu s pomočjo izrazov [41]:

$$R_{WLCS} = f^{-1}\left(\frac{WLCS(X,Y)}{f(m)}\right) \tag{4.4}$$

$$P_{WLCS} = f^{-1}\left(\frac{WLCS(X,Y)}{f(n)}\right) \tag{4.5}$$

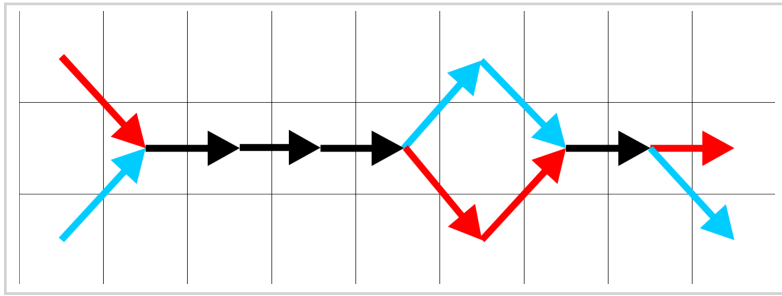
$$F_{WLCS} = \frac{(1+\beta^2)R_{WLCS}P_{WLCS}}{R_{WLCS}+\beta^2 P_{WLCS}} \tag{4.6}$$

$$d_{WLCS} = 1 - F_{WLCS} \tag{4.7}$$

kjer je $f(k) = k^2$.

Časovna zahtevnost algoritma Lin-Och je $O(mn)$, kjer sta m and n dolžini posameznih poti v paru. Tako je pričakovana časovna zahtevnost za izračun mere različnosti med n_p potmi s povprečno dolžino l_p enaka $O(n_p^2 l_p^2)$.

Pri hierarhičnem razvrščanju poti uporabimo Ward-ovo metodo [43], ki upošteva minimalno varianco med potmi, ki se nahajajo v istih skupinah (dinamične mehke poti). Ta metoda nam da kompaktne skupine poti, ki imajo primerljivo varianco, kar je zelo zaželena lastnost pri združevanju poti, ki imajo skupne povezave. Na osnovi poti v skupini tako vzpostavimo t. i. mehko pot v smislu prostorske podobnosti, ki jo določa WLCS.



Slika 4.3

Primer dveh poti (rdeče in modre barve) s skupnimi podzaporedji (črne barve).

4.2.4 Detekcija ciklov v poteh

Da bi lahko konstruirali dinamične mehke cikle, je potrebno v posameznih dinamičnih poteh (glej razdelek 4.2.2) najprej odkriti posamezne cikle, ki so lahko različnega tipa. Za nas najbolj zanimivi ciklonski (v nasprotni smeri urinega kazalca na severni polobli) in anticiklonski cikli (v smeri urinega kazalca na severni polobli).

Definicija 5: Cikel je tako podzaporedje vozlišč na poti, ki se začne in konča z istim vozliščem, ki ga imenujemo "vozelna točka".

Izločanje ciklov iz poti je sestavljeno iz več faz:

1. V poteh, ki so rezultat algoritma 2 (slika 4.2) in so predstavljene kot zaporedja vozlišč P^V, je potrebno najprej najti vse cikle.

   V danih poteh lahko najdemo različne tipe ciklov: ravninske ciklonske, anticiklonske in t. i. " nedefinirane" cikle, ki obdajajo poligone, katerih ploščine so enake 0. Poleg teh dobimo še t. i. ne-ravninske cikle, kjer se povezave lahko sekajo zunaj vozlišč. Algoritem 3 (slika 4.4) v dani poti najde vse cikle različnih tipov. Dobljeni cikli lahko vsebujejo še druge vgnezdene cikle ali pa se delno prekrivajo z drugimi cikli. Algoritem 3 ima časovno zahtevnost $O(l_p^2)$, kjer je l_p dolžina dane poti oziroma število vozlišč na tej poti.

2. Dobljene dinamične cikle, ki vsebujejo druge vgnezdene cikle, poenostavimo v enostavne cikle tako, da zamenjamo vsa podzaporedja, ki so tudi cikli, z vozlišči, ki se nahajajo na začetku in na koncu podzaporedja oziroma "vozelnimi

Algoritem 3 Iz dane poti izloči vse cikle

Vhod: Pot p^v v obliki zaporedja vozlišč
Izhod: Množica ciklov C_{cycle}

```
1:  C_cycle ← ∅
2:  path.len ← length(p^v)
3:  // najkrajši cikel vsebuje 3 vozlišča
4:  if path.len ≥ 3 then
5:      for i = 1 → path.len do
6:          for j = path.len → i do
7:              if p_i^v == p_j^v  &  i ≠ j then
8:                  C_cycle ← C_cycle ∪ p_{i ... j}^v
9:              end if
10:         end for
11:     end for
12: end if
13: return C_cycle
```

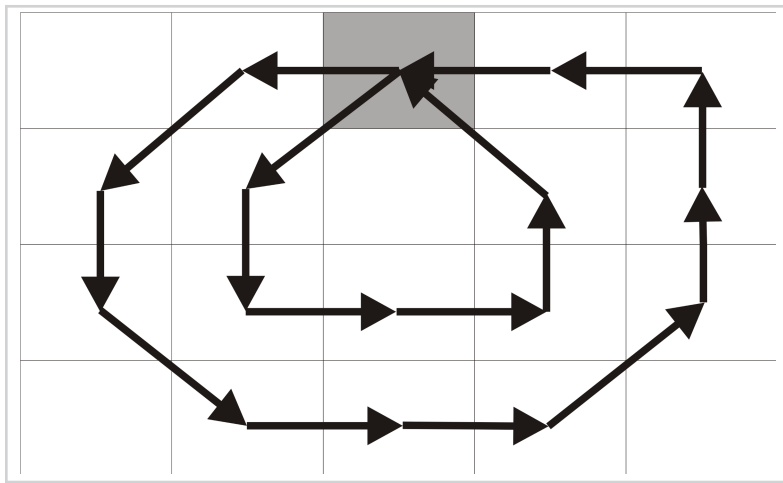
Slika 4.4

Algoritem 3 – ekstrakcija vseh ciklov iz dane poti

točkami" (glej sliko 4.5). Omenjena poenostavitev je naloga algoritma 4 (slika 4.6), ki primerja vse pare ciklov in če je potrebno, opravi to zamenjavo. Funkcija $find(C_i, C_j)$ išče krajši cikel C_i v daljšem ciklu C_j (vrstica 8). Če ga najde, potem funkcija $substitute(C_i, C_j, C_{i_1})$ nadomesti C_i v C_j z "vozelno točko", ki je prvo (in zadnje) vozlišče v C_i. Poleg tega algoritem doda na seznam poenostavljenih ciklov še enostavne cikle, kjer ni bilo možno opraviti nobene zamenjave (vrstice 15–19). Algoritem se izvaja rekurzivno, dokler ni možno narediti več nobene zamenjave. Potrebno število rekurzivnih klicev je odvisno od števila in kompleksnosti vgnezdenih ciklov v dani poti. Časovna zahtevnost algoritma 4 pri vsakem rekurzivnem klicu je enaka $O(|C_{cycles}|^2)$, ker algoritem primerja vse možne pare ciklov z namenom, da zamenja krajše cikle v daljših z "vozelnimi točkami".

Funkcionalnost algoritma 4 najbolje ponazorimo s primerom, ki je prikazan v tabeli 4.1. Za poenostavitev ciklov so bili v tem primeru potrebni štirje rekurzivni klici.

3. Enostavne cikle, ki smo jih dobili s pomočjo algoritma 4, klasificiramo glede na smisel vrtenja:



Slika 4.5

Dva vgnezdena cikla z "vozelno točko", ki je na sliki osenčena.

Tabela 4.1

Primer poenostavitve ciklov. V prvem stolpcu je seznam ciklov, dobljenih z algoritmom 3, v zadnjem pa seznam poenostavljenih ciklov, ki jih vrne algoritem 4 po štirih rekurzivnih klicih. Pot in cikli so podani kot zaporedja vozlišč, označenih z zaporednimi številkami od 1 do 8.

Pot	(3,2,5,6,8,7,8,2,1,2,1,2,5,7)			
Cikli (rezultat Algoritma 3)	Reducirani cikli število rekurzivnih klicev Algoritma 4			
	1	2	3	4 (rezultat)
(8,7,8)	(8,7,8)	(8,7,8)	(8,7,8)	(8,7,8)
(2,1,2)	(2,1,2)	(2,1,2)	(2,1,2)	(2,1,2)
(1,2,1)	(1,2,1)	(1,2,1)	(1,2,1)	(1,2,1)
(2,1,2,1,2)	(2,5,6,8,2)	(2,5,6,8,2)	(2,5,6,8,2)	(2,5,6,8,2)
(2,5,6,8,7,8,2)	(7,8,2,5,7)	(7,8,2,5,7)	(7,8,2,5,7)	(7,8,2,5,7)
(2,5,6,8,7,8,2,1,2)	(2,1,2,1,2)	(5,6,8,2,5)	(5,6,8,2,5)	(5,6,8,2,5)
(7,8,2,1,2,1,2,5,7)	(2,5,6,8,2,1,2)	(2,1,2,1,2)	(2,1,2,1,2)	
(2,5,6,8,7,8,2,1,2,1,2)	(2,5,6,8,7,8,2)	(2,5,6,8,1,2)	(2,5,6,8,2,1,2)	
(5,6,8,7,8,2,1,2,1,2,5)	(7,8,2,1,2,5,7)	(5,6,8,2,1,2,5)	(5,6,8,2,1,2,5)	
	(5,6,8,7,8,2,5)	(2,5,6,8,2,1,2)	(2,5,6,8,7,8,2)	
	(2,5,6,8,7,8,1,2)	(5,6,8,7,8,2,5)	(5,6,8,7,8,2,5)	
	(2,5,6,8,2,1,2,1,2)	(2,5,6,8,7,8,2)		
	(5,6,8,2,1,2,1,2,5)	(7,8,2,1,2,5,7)		
	(5,6,8,7,8,2,1,2,5)	(2,5,6,8,2,1,2,1,2)		
	(2,5,6,8,7,8,2,1,2)	(5,6,8,2,1,2,1,2,5)		
	(7,8,2,1,2,1,2,5,7)	(2,5,6,8,7,8,2,1,2)		
	(2,5,6,8,7,8,2,1,2,1,2)	(5,6,8,7,8,2,1,2,5)		
	(5,6,8,7,8,2,1,2,1,2,5)			

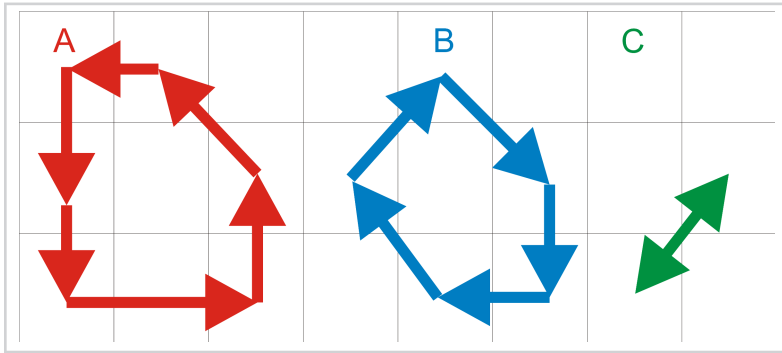
Algoritem 4 Procedura REDUCECYCLES(C_{cycles}), ki poenostavi cikle - pri prvem klicu je vhod začetna množica ciklov C_{cycles}, ki še niso bili poenostavljeni.

Vhod: Množica ciklov C_{cycles}
Izhod: Množica enostavnih ciklov $C_{reduced}$

```
1: C_reduced ← ∅
2: reduction ← False
3: for all C ∈ C_cycles do
4:     reduced_C ← False
5: end for
6: for all C_i ∈ C_cycles do
7:     for all C_j ∈ C_cycles do
8:         if find(C_i, C_j) & i ≠ j & length(C_j) > length(C_i) then
9:             C_reduced ← C_reduced ∪ substitute(C_i, C_j, C_i1)
10:            reduced_Cj ← True
11:            reduction ← True
12:        end if
13:    end for
14: end for
15: for all C ∈ C_cycles do
16:    if ¬reduced_C then
17:        C_reduced ← C_reduced ∪ C
18:    end if
19: end for
20: if reduction then
21:    return REDUCECYCLES(C_reduced)
22: else
23:    return C_reduced
24: end if
```

Slika 4.6

Algoritem 4 – poenostavitev ciklov



Slika 4.7

Primeri ciklov: A) ciklonski, B) anticiklonski, C) "nedefiniran"

(a) "Nedefinirani" cikli, ki vsebujejo samo dve vozlišči in dve povezavi, kar pomeni, da je psevdo-površina, ki jo zajema takšen cikel, enaka 0.

(b) Z zaključenimi (ne-ravninskimi) krivuljami, ki sekajo same sebe in zahtevajo posebno obravnavo, se v tem delu ne ukvarjamo. Detektiramo jih s pomočjo postopka iz računalniške geometrije (ang. *self intersect*). Za krivulje, ki sekajo same sebe, je možno razviti bolj zapleten algoritem, ki omenjene krivulje razstavi na enostavne cikle. Kasneje iz rezultatov razberemo, da omenjene krivulje predstavljajo manj kot 10 % vseh ciklov.

(c) Če dobljeni cikel ni opisan niti pod (a) niti pod (b), potem je bodisi ciklonski (proti smeri urinega kazalca) ali pa anticiklonski (v smeri urinega kazalca). To ugotovimo z izračunom psevdo-površine poligona, ki ga definira cikel [28, 40]. Če je psevdo-površina pozitivna, potem je cikel ciklonski, če pa je negativna, imamo opravka z anticiklonskim ciklom. V primeru, ko je psevdo-površina enaka 0, smisla vrtenja ne moremo ugotoviti in je dani cikel "nedefiniran". Primeri opisanih ciklov so prikazani na sliki 4.7.

Poti in cikli, ki jih obravnavajo algoritmi 2–4, vsebujejo za vsako vozlišče tudi časovno informacijo, ki je po navadi v obliki $\langle leto, mesec, dan \rangle$. To je potrebno zaradi nadaljnjih analiz pogostosti dinamičnih poti in ciklov v različnih časovnih obdobjih.

Definicija 6: Enoličen cikel je trojka:

$$C = \langle V_C, tip, \#pojavitev \rangle \tag{4.8}$$

kjer je V_C množica vozlišč vsebovanih v ciklu, *tip* je tip cikla (ciklonski, anticiklonski ali "nedefiniran") in *#pojavitev* je število pojavitev tega cikla v danem časovnem obdobju.

Definicija 7: Enolična pojavitev cikla je četvorka:

$$C_i = \langle V_C, tip, t_{zacetek}, t_{konec} \rangle \tag{4.9}$$

kjer $t_{zacetek}$ in t_{konec} predstavljata začetni in končni čas cikla in sta po navadi zapisana kot $\langle leto, mesec, dan \rangle$.

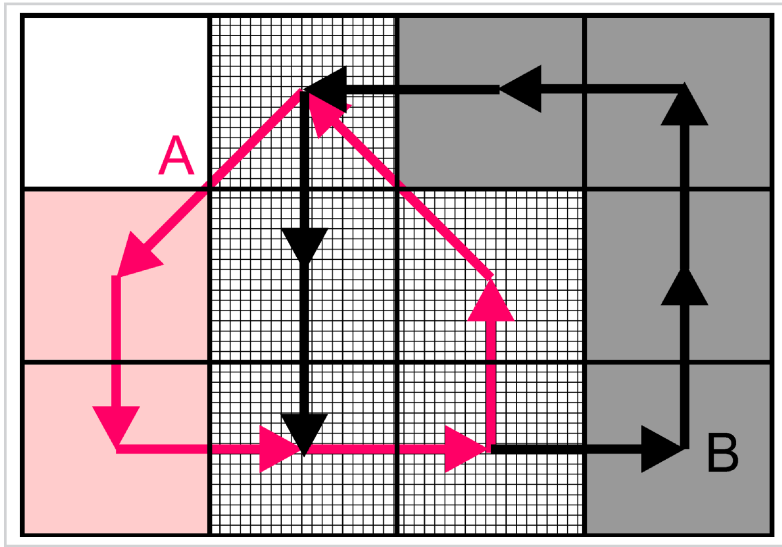
4.2.5 *Dinamični mehki cikli*

Z uporabo algoritmov, ki jih opišemo v razdelku 4.2.4, dobimo množici enoličnih ciklonskih in anticiklonskih ciklov (skupaj s številom pojavitev za posamezni cikel), ki se med seboj prekrivajo (glej primere na sliki 5.14). Zato je smiselno prekrivajoče se (posebej ciklonske in posebej anticiklonske) cikle med seboj združevati in dobimo t. i. *dinamične mehke cikle*.

Definicija 8: Dinamični mehki cikli so množice ciklov istega tipa, ki jih dobimo kot rezultat hierarhičnega razvrščanja posameznih dinamičnih ciklov v skupine z uporabo primerne razdalje in kasneje z rezanjem dobljenega dendrograma na ustrezni višini. Dinamični mehki cikel vizualiziramo v obliki konveksne ogrinjače povprečja ciklov (poligonov) v dani skupini.

Da določimo razdaljo, ki je potrebna za hierarhično razvrščanje dinamičnih ciklov, se opremo na indeks Tverskega [44], ki je definiran takole:

$$Tversky(A; B; \alpha; \beta) = \frac{|A \cap B|}{|A \cap B| + \alpha |A - B| + \beta |B - A|}, \tag{4.10}$$



Slika 4.8

Primera dveh ciklonskih ciklov C_A (rdeč) and C_B (črn). Skupna vozlišča v preseku ciklov so šrafirana; vozlišča, ki pripadajo razliki $A - B$, so svetlo rdeče barve; tista, ki pripadajo razliki $B - A$, pa so sive barve.

kjer sta A in B množici objektov, med katerimi ugotavljamo podobnost. V našem primeru sta A in B množici vozlišč grafa, ki jih obdajata cikla C_A in C_B (slika 4.8). Tukaj uporabimo ploščine območij vozlišč, ki se prekrivajo, ter njihove razlike in izračunamo razdaljo Tverskega kot sledi:

$$D_{tversky}(A;B;\alpha;\beta) = 1 - \frac{S(A \cap B)}{S(A \cap B) + \alpha S(A - B) + \beta S(B - A)}, \qquad (4.11)$$

kjer S predstavlja ploščino preseka in razlik med množicama ciklov C_A in C_B. V primeru, ko cikli nimajo skupnega preseka, je $D_{tversky}$ enak 1, sicer zavzame vrednost med 0 in 1. Vrednost razdalje Tverskega je odvisna od ploščine preseka in razlik med A in B ter od izbire parametrov α in β. Manjše vrednosti teh parametrov pomenijo, da bolj poudarimo skupno območje (presek) obeh ciklov in bolj zanemarimo njune razlike. Po drugi strani pa večja vrednost parametrov bolj poudari razlike med cikli v primerjavi z njihovim presekom.

Razdalja Tverskega v splošnem ni simetrična, ker razlika množic ni simetrična. Podobnost med objektoma A in B ni enaka podobnosti med B in A, zato velja

$$D_{tversky}(A;B;\alpha;\beta) \neq D_{tversky}(B;A;\alpha;\beta) \tag{4.12}$$

Da bi za našo uporabo razdalja Tverskega postala simetrična, vzamemo za C_A cikel z večjo ploščino in za C_B tistega z manjšo. Če pri izračunu razdalje Tverskega uporabimo večji koeficient α, potem bolj poudarimo razliko med ploščino večjega cikla in ploščino manjšega cikla. Nasprotno, uporaba večjega koeficienta β bolj poudari razliko med manjšim in večjim ciklom. V nadaljevanju izvedemo hierarhično razvrščanje ciklov z uporabo različnih vrednosti koeficientov α in β.

Za hierarhično razvrščanje je na voljo več različnih pristopov in metod [15]. V naših primerih uporabimo pristop *od spodaj navzgor*, kjer na začetku vsak cikel predstavlja svojo skupino. Algoritem nato iterativno združuje podobne skupine, dokler ne ostane le nekaj v prostoru ločenih skupin. Pri tem obstaja več metod za povezovanje: minimalna, povprečna in maksimalna, kjer se po vrsti cikli združujejo v skupine na osnovi minimalne, povprečne in maksimalne razdalje med cikli v različnih skupinah. Poleg teh je na voljo še Ward-ova metoda, ki minimizira varianco med cikli znotraj skupin.

4.2.6 Vizualizacija dinamičnih mehkih ciklov

Rezultat hierarhičnega razvrščanja so množice ciklov (mehki cikli), kjer se posamezni cikli v ravnini v celoti ali vsaj deloma prekrivajo, kar pomeni, da imajo skupno območje (presek) znotraj mehkega cikla. Z namenom, da bi mehke cikle čim bolj nazorno prikazali, združimo posamezne cikle v enoten cikel s pomočjo izračuna "povprečja" teh ciklov, ki so v bistvu mnogokotniki v ravnini. Na ta način dobimo kot rezultat večji mnogokotnik, ki obdaja cikle v skupini. V ta namen uporabimo vsoto Minkowskega [45] za mnogokotnike (v našem primeru cikle), s pomočjo katere izpeljemo izraz za izračun povprečja ciklov v skupini. Povprečje dveh ciklov, kjer prvi vsebuje n_1, drugi pa n_2 vozlišč, izračunamo s seštevanjem parov ravninskih koordinat njihovih vozlišč in deljenjem te vsote z 2. Na ta način dobimo mnogokotnik z največ $n_1 \times n_2$ vozlišči. Dobljenemu povprečju dodajamo nove cikle s pomočjo izraza za rekurzivno povprečje [46]:

$$\overline{C_{i+1}} = (\overline{C_i}i + C_{i+1}1)/(i+1) \tag{4.13}$$

Algoritem 5 (slika 4.9) prikazuje celoten postopek tvorbe dinamičnih mehkih ciklov. Vhod v algoritem predstavljata množica ciklov danega tipa in želena višina rezanja v

Algoritem 5 Tvorba mehkih ciklov

Vhod:

- Množica ciklov C_{ctype}
- Tip ciklov *ctype* - "C" (ciklonski), "A" (anticiklonski)
- Višina rezanja dendrograma h_{cut}

Izhod: Množica mehkih ciklov C_{fuzzy}

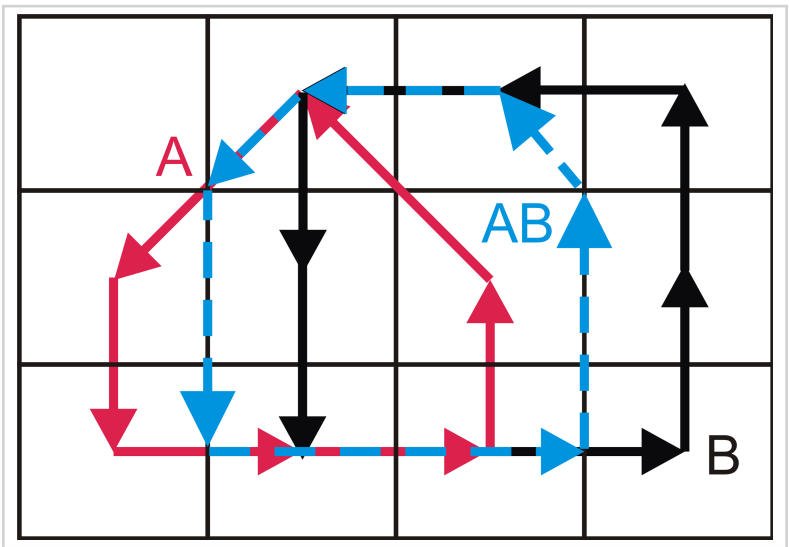
1: $d_{tversky} \leftarrow tversky(C_{ctype}, \alpha, \beta)$
2: $dendrogram_1 \leftarrow hierarchicalClustering(d_{tversky})$
3: $dendrogram_2 \leftarrow lowerTree(cut(dendrogram_1, h_{cut}))$
4: $C_{fuzzy} \leftarrow \emptyset$
5: **for all** $dendrogram \in dendrogram_2$ **do**
6: **if** $\neg is.leaf(dendrogram)$ **then**
7: $C_{curr} \leftarrow minkowski(ctype, dendrogram)$
8: $C_{curr} \leftarrow convexHull(C_{curr})$
9: $C_{curr} \leftarrow concatenate(C_{curr}, C_{curr_1})$
10: **if** $ctype == "C"$ **then**
11: $C_{curr} \leftarrow reverse(C_{curr})$
12: **end if**
13: **end if**
14: **end for**
15: $C_{fuzzy} \leftarrow C_{fuzzy} \cup C_{curr}$
16: **return** C_{fuzzy}

Slika 4.9

Algoritem 5 – tvorba dinamičnih mehkih ciklov.

končnem dendrogramu (vrstice 2–3). Pri večji višini rezanja dobimo manjše število mehkih ciklov, ki vsebujejo večje število posameznih ciklov. Za potrebe vizualizacije mehkih ciklov izračunamo *konveksno ogrinjačo* (ang. *convex hull*) [47] poligona (vrstica 8), ki smo ga dobili z izračunom "povprečja Minkowskega" (vrstica 7). Na koncu moramo dobljeni konveksni ogrinjači dodati še prvo vozlišče, da zaključimo cikel (vrstica 9). Ker je dobljena konveksna ogrinjača podana v smeri urinega kazalca, moramo v primeru ciklonskih ciklov (nasprotna smer urinemu kazalcu) obrniti zaporedje vozlišč v ciklu (vrstice 10–12).

Primer mehkega cikla, ki ga na opisan način dobimo iz ciklonskih ciklov A in B s slike 4.8, je prikazan na sliki 4.10.



Slika 4.10

Konveksna ogrinjača (modra črtkana črta), ki jo dobimo, če izračunamo "povprečje Minkowskega" ciklonskih ciklov A in B s slike 4.8.

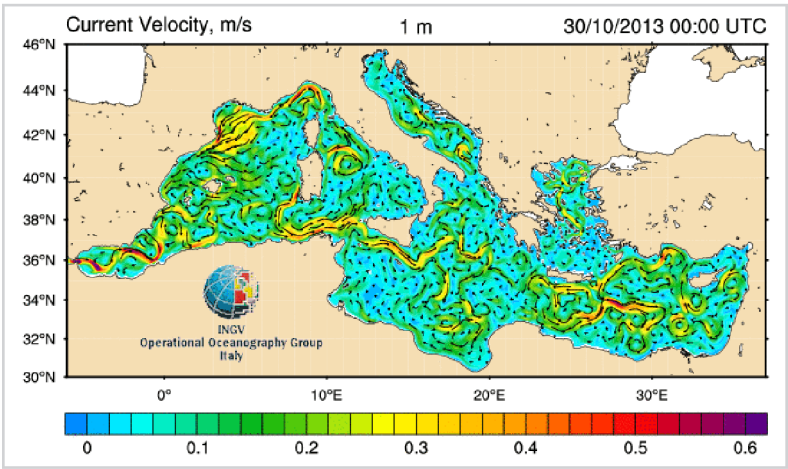
5 Rezultati

V tem poglavju prikazujemo rezultate uporabe metodologije in algoritmov, ki so opisani v poglavjih 3 in 4, na rezultatih oceanografskega numeričnega modela Mediterranean Ocean Forecasting System (MFS). Metodologijo, opisano v poglavju 3, najprej ovrednotimo s pomočjo rezultatov omenjenega numeričnega modela. V nadaljevanju prikažemo rezultate algoritma za iskanje enostavnih ciklov v večnivojskih usmerjenih grafih, ki so opisani v razdelku 4.1, nato pa še rezultate iskanja dinamičnih mehkih poti in ciklov (razdelek 4.2). Vse tako dobljene dinamične mehke strukture uporabimo za primerjavo s strukturami, ki so jih definirali oceanografski eksperti na osnovi opazovanj ali rezultatov numeričnih modelov. Dobljene dinamične mehke poti primerjamo z znanimi strukturami, ki so jih opazili pri gibanju vodnih mas v Jadranskem morju, dobljene dinamične mehke cikle pa z reprezentativnimi vrtinci v celotnem Sredozemskem morju. Vidimo, da so dobljene strukture zelo podobne tistim, ki jih opisuje oceanografska literatura, poleg tega pa z našimi metodami dobimo tudi strukture, ki jih oceanografski eksperti morajo še potrditi. Na koncu tega poglavja podamo še čase izvajanja omenjenih algoritmov na uporabljeni strojni in programski opremi.

5.1 Ovrednotenje metodologije

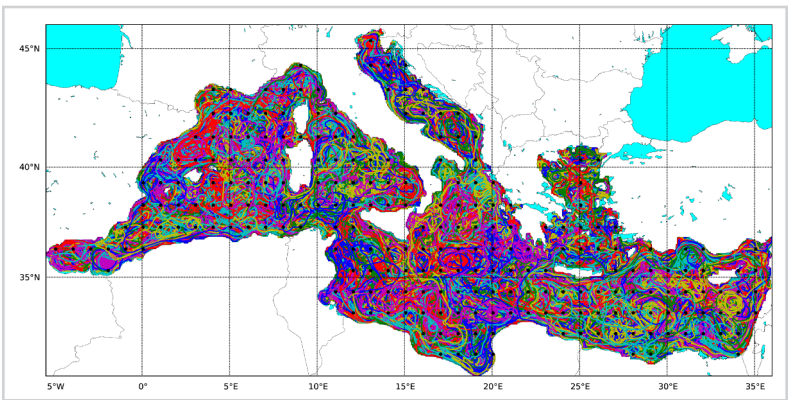
Metodologijo, ki smo jo opisali v poglavju 3, ovrednotimo s pomočjo hitrostnega polja v numeričnem oceanografskem modelu Mediterranean Ocean Forecasting System (MFS) [5, 9, 10] v obdobju 1999–2011, za katerega izračunamo Lagrangeove trajektorije s pomočjo orodja Ariane [48]. Primer slike hitrostnega polja omenjenega numeričnega modela je prikazan na sliki 5.1. Numerični model ima horizontalno resolucijo $1/16^0 \times 1/16^0$ (253×677 celic) in 71 neenakomerno razporejenih vertikalnih nivojev. Horizontalna velikost posamezne celice je v tem primeru približno 6 km. V tej simulaciji smo spustili navidezne (numerične) delce iz 239 horizontalno enakomerno porazdeljenih začetnih točk (vsaka 16. celica v x in y smeri numeričnega modela) v globini 1 m v začetku vsakega meseca do vključno januarja 2011. Tako smo v trajanju 145 mesecev dobili skupno 34.655 trajektorij, kjer ima vsaka dolžino 365 dni. Pri tem je vertikalna hitrost navideznih delcev enaka 0 podobno kot pri plovcih, ki ves čas potujejo na isti globini (v našem primeru 1 m). Slika 5.2 prikazuje nastalo množico trajektorij, iz katerih pa ne moremo razbrati bistvene cirkulacije v Sredozemskem morju in zato uporabimo metode prostorsko-časovnega podatkovnega rudarjenja s pomočjo večnivojskih usmerjenih grafov.

Domeno (slika 5.2) smo razdelili na 60 x 30 = 1800 enako velikih območij (velikost



Slika 5.1

Primer hitrostnega polja v numeričnem modelu Mediterranean Ocean Forecasting System (MFS). Slika prikazuje površinske morske tokove v Sredozemskem morju za 30. 10. 2013 in je povzeta po [10].



Slika 5.2

Trajektorije površinskih virtualnih delcev, spuščenih v hitrostnem polju numeričnega modela Mediterranean Ocean Forecasting System (MFS) v obdobju 1999–2011. Črne točke predstavljajo začetne pozicije delcev. Trajektorije so različnih barv zaradi bolj nazorne predstavitve.

Tabela 5.1

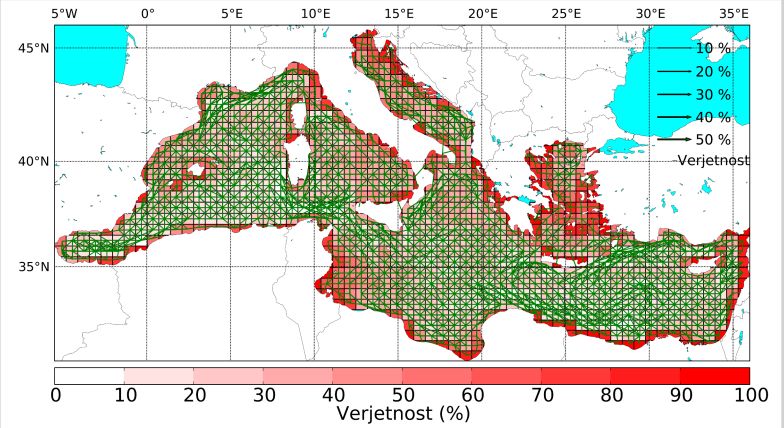
Opisna statistika povezav večnivojskih usmerjenih grafov z različno časovno granulacijo (mesečna, letna in celotna).

Časovna granulacija	Število povezav Povprečje±SD	Min.	Maks.
Mesečna	4224±766	718	5372
Letna	11193±758	9464	11877
Celotna	18770±0	18770	18770

posameznega območja je približno 60 km) in med njimi je 848 takih, ki so delno ali v celoti prekrita z morjem. V tem primeru optimalni "graf sosedov" (glej enačbo 3.8) vsebuje 848 vozlišč in 7006 povezav. S pomočjo metode, opisane v razdelku 3.5, smo določili optimalni časovni interval Δt, ki je enak 6 dni in v katerem 40,3 % delcev (maksimum) preide v sosednja območja, 50 % jih ostane v istih območjih, 9,7 % pa prečka sosednja območja in preide v bolj oddaljena območja (slika 3.3). Na ta način smo dobili 12.415.413 primerov in na njihovi osnovi konstruirali 156 mesečnih grafov (199901, 199902, …, 201112), 13 letnih grafov (1999, 2000, …, 2011) in en graf, ki pokriva celotno obdobje 1999–2011. Pri tem smo uporabili prostorsko-časovna povezovalna pravila, ki jih po vrsti prikazujejo enačbe 3.5, 3.6 in 3.7. V tabeli 5.1 povzemamo število povezav v dobljenih grafih. Iz tabele je razvidno, da grafi, ki pokrivajo daljša časovna obdobja, vsebujejo tudi večje število povezav, npr. letni grafi imajo več kot dvakrat večje število povezav kot mesečni grafi. To je mogoče pripisati različnim povezavam z nizko podporo in zaupanjem, ki kažejo v ne-sosednja (oddaljena) območja. Večnivojski usmerjeni graf za celotno obdobje 1999–2011 (slika 5.3) vsebuje 18.770 povezav.

5.2 *Iskanje enostavnih ciklov*

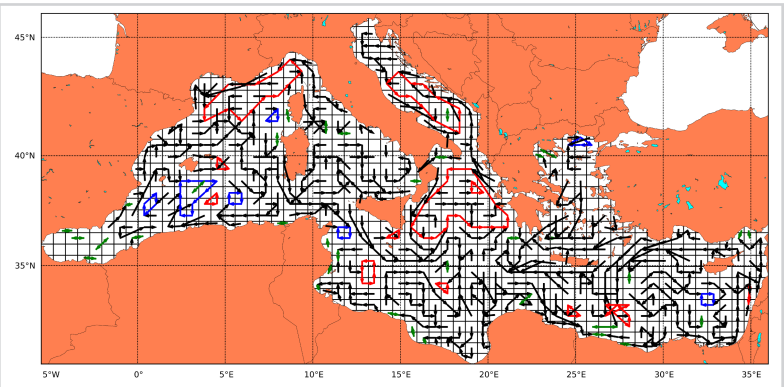
Algoritem 1 za iskanje enostavnih ciklov (glej razdelek 4.1 in [28]) smo uporabili na 156 mesečnih grafih, dobljenih iz numeričnega modela Mediterranean Ocean Forecasting System (MFS) za obdobje 1999–2011, in tako našli skupno 1361 ciklonskih, 774 anticiklonskih in 5197 "nedefiniranih" ciklov. Sliki 5.4 in 5.5 c prikazujeta primer ciklov, ki smo jih našli v grafu za september 2000. Ta primer je zanimiv tudi zato, ker vsebuje tudi nekaj večjih ciklonskih ciklov, ki se sicer razvijejo v obdobjih, daljših od



Slika 5.3

Večnivojski usmerjeni graf, konstruiran iz verjetnosti prehodov med različnimi (zelene puščice) in istimi morskimi območji (rdeči pravokotniki) za obdobje 1999–2011. Uteži zank v usmerjenem grafu so sorazmerne z nasičenostjo rdeče barve, debelina zelenih povezav v grafu je sorazmerna z verjetnostjo prehoda med različnimi morskimi območji. Za lažjo predstavitev so prikazani le prehodi z minimalnim zaupanjem 0,05.

enega meseca.

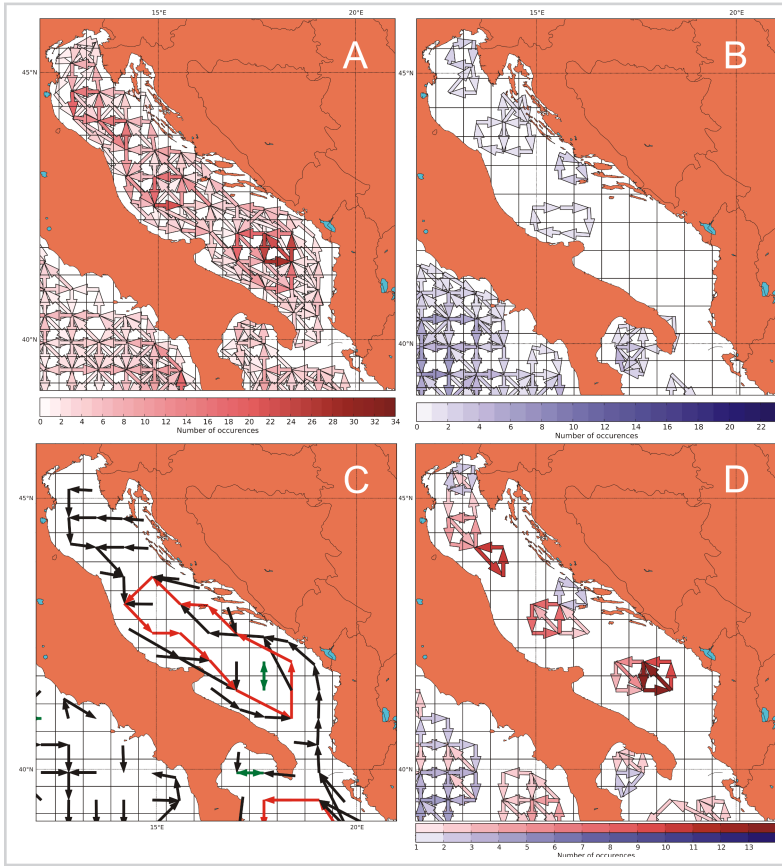


Slika 5.4

Primer enostavnih ciklov, ki jih dobimo s pomočjo algoritma 1 (slika 4.1) v grafu za september 2000. Ciklonski cikli so obarvani rdeče, anticiklonski pa modro. "Nedefinirani" cikli, ki vsebujejo le po dve vozlišči, so obarvani zeleno. Poti so v črni barvi.

Sliki 5.5 A in 5.5 B prikazujeta pokritost Jadranskega morja s ciklonskimi in anticiklonskimi cikli. Oboji so bili dobljeni z nalaganjem bodisi ciklonskih ali pa anticiklonskih ciklov drug na drugega. Tako imajo bolj pogoste povezave tudi bolj nasičeno barvo. To daje uporabniku prvi vtis o tem, kako so cikli porazdeljeni in kje so bolj pogosti.

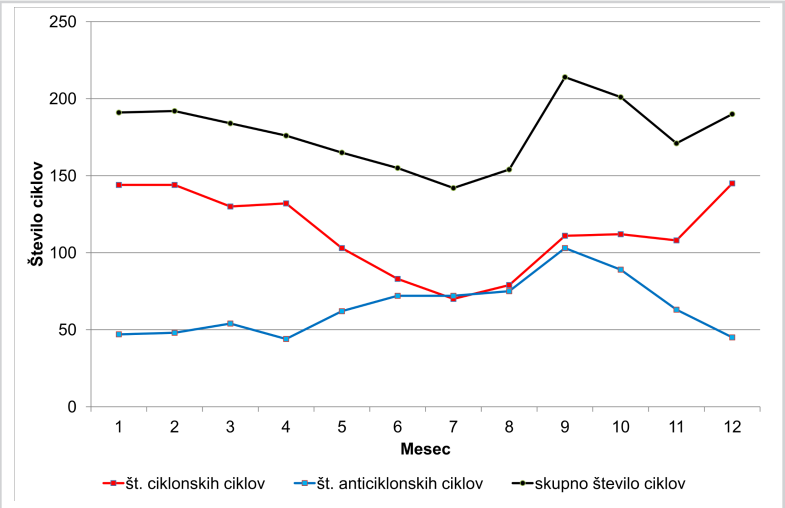
Analizirali smo tudi časovni razvoj števila in velikosti ciklov v obdobju 1999–2011



Slika 5.5

Prikaz ciklonskih (A) in anticiklonskih ciklov (B), ki ga dobimo, če naložimo posamezne povezave vseh ciklonskih (anticiklonskih) ciklov v obdobju 1999–2011 v območju Jadranskega morja in delno Jonskega morja; (C) primeri ciklov za september 2000; (D) pogosti cikli, dobljeni s pomočjo algoritma APriori z uporabo minimalne podpore 0,0005. Nasičenost rdeče in modre barve na slikah A, B in D je sorazmerna pogostosti posameznih povezav.

na mesečni ravni, ki kaže zanimiva sezonska nihanja. Iz slike 5.6 lahko razberemo, da je število ciklonskih ciklov pozimi skoraj dvakrat večje kot poleti. Njihovo število doseže minimum v juliju in maksimum v januarju. Število anticiklonskih ciklov narašča od decembra proti toplejšim letnim časom in doseže maksimum v septembru. Skupno število ciklov je največje v zimskih mesecih, nato pa začne upadati in doseže globalni minimum v juliju. Po tem začne hitro naraščati in doseže znatno konico v septembru in nato znova upada in doseže lokalni minimum v novembru.

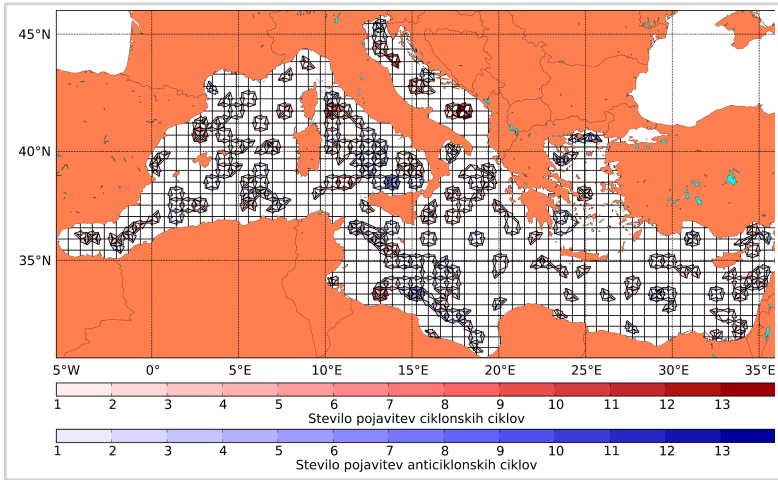


Slika 5.6

Število vseh ciklonskih (rdeče), anticiklonskih (modro) in skupno število ciklov po mesecih.

Sliki 5.5 D in 5.7 prikazujeta pogoste cikle, ki smo jih našli s pomočjo algoritma APriori z uporabo minimalne podpore 0,0005. Z drugimi besedami, vsak od dobljenih pogostih ciklov se mora pojaviti najmanj dvakrat. Če ne upoštevamo "nedefiniranih" ciklov, potem skupno število ciklov znaša 2135 (1361 ciklonskih in 774 anticiklonskih). Tako mora biti minimalno število pojavitev dobljenih pogostih enako 2 (2 > (2135*0,0005 = 1,07)). Na ta način smo dobili število pojavitev dobljenih pogostih ciklov v obdobju 1999–2011 na mesečni ravni na intervalu od 2 do 13. To pomeni, da se identični cikli ponovijo kvečjemu enkrat na leto ali celo manj pogosto.

Nekateri cikli, ki smo jih dobili s pomočjo opisanega algoritma, so obravnavani v številnih študijah, na primer [49–51]. Za območje Jadranskega morja ta metoda jasno razkrije najbolj pogoste ciklonske vrtince (rdeče barve) v južnem Jadranu, manj pogoste pa srednjem Jadranu, kjer so v bližini obalne črte prisotni tudi anticiklonski vrtinci (slika 5.5). Ciklonske vrtince opazimo tudi med srednjim in severnim Jadranom, medtem, ko so anticiklonski vrtinci najbolj pogosti v najbolj severnem kotu s ciklonskimi vrtinci, ki se nahajajo južno od njih. Drugi primeri ciklov, ki so bili dobljeni s pomočjo opisanega algoritma in so v skladu z znanimi vrtinci, so: ciklonski Northern Tyrrhenian Gyre, ki ga povzroča veter, ciklonska vrtinca Cretan Gyre in Rhodes Gyre, anticiklonski Mersa-Matruh Eddy južno od Cipra, itd. (slika 5.7).



Slika 5.7

Pogosti cikli, dobljeni z algoritmom APriori z uporabo minimalne podpore 0,0005.

5.3 *Iskanje dinamičnih mehkih poti in ciklov*

V tem razdelku opisujemo rezultate iskanja dinamičnih mehkih poti in ciklov, ki smo jih dobili s pomočjo algoritmov na večnivojskih usmerjenih grafih, opisanih v razdelku 4.2 (glej tudi [35]). Omenjene večnivojske usmerjene grafe smo tvorili iz rezultatov numeričnega modela Mediterranean Ocean Forecasting System (MFS) v obdobju 1999–2011 in pri tem upoštevali gibanje vodnih mas samo na površini oceana. Ker nam je algoritem 2 (slika 4.2) dal veliko število posameznih dinamičnih poti (okoli pol milijona), se pri iskanju dinamičnih mehkih poti raje omejimo na manjši del domene tj. Jadransko morje, za katero je na voljo tudi bolj preprosto in razumljivo predznanje oceanografskih ekspertov. Po drugi strani pa je število posameznih dinamičnih ciklov v Sredozemlju bistveno nižje od števila poti, tako da lahko iščemo dinamične mehke cikle kar v celotnem Sredozemlju.

5.3.1 *Dinamične mehke poti*

Algoritem 2 (slika 4.2) smo izvedli nad 156 mesečnimi grafi pridobljenimi iz numeričnega modela MFS 1999–2011 in našli skupno 551.430 najverjetnejših dinamičnih poti. Pri tem smo uporabili širino snopa $M = 2$. Iskanje dveh najverjetnejših poti v snopu je potrebno zato, ker se lahko v določenih vozliščih pojavita dve izhodni po-

Tabela 5.2

Opisna statistika dinamičnih poti, ki se nahajajo v Jadranskem morju (numerični model MFS 1999–2011) in imajo dolžino najmanj 3 povezave.

Število poti	Dolžina poti					
	Min.	1. četrt.	Mediana	Povp.	3. četrt.	Maks.
10.143	3	4	5	6,407	8	35

vezavi z največjo verjetnostjo, katerih vrednosti se le malo razlikujeta in ju zato lahko smatramo kot enakovredni. Možno je tudi, da naletimo na vozlišča, ki imajo po tri ali več približno enakovrednih povezav, vendar smo se zaradi skrajšanja časa izvajanja algoritma omejili na širino snopa $M = 2$, kar po našem mnenju ne spremeni bistveno rezultatov. Za lažjo primerjavo z obstoječim oceanografskim znanjem smo uporabili za hierarhično razvrščanje le podmnožico vseh dobljenih poti v Sredozemskem morju tj. uporabili smo samo poti ali njihove odseke, ki se v celoti nahajajo samo v Jadranskem morju (glej sliki 5.8 and 5.9) in imajo najmanjšo dolžino tri povezave. S temi omejitvami smo našli 10.143 poti ali njihovih odsekov, katerih opisna statistika je prikazana v tabeli 5.2.

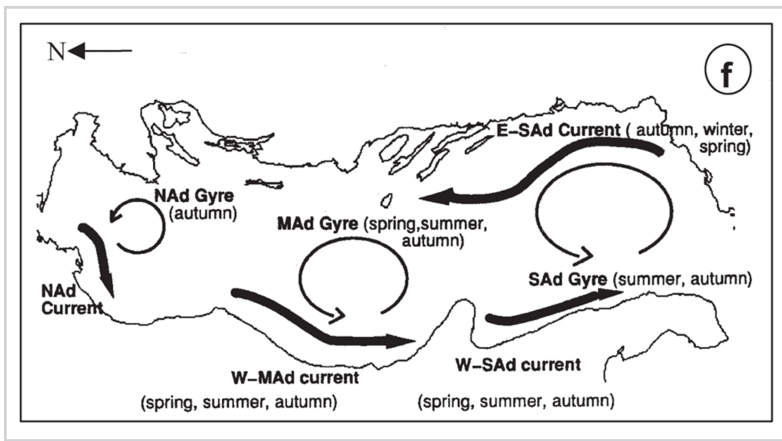
Z uporabo Ward-ove razdalje smo poti v Jadranskem morju hierarhično razvrstili v 12 skupin, za katere se je izkazalo, da so primerljive s strukturami iz oceanografske literature [52] (slika 5.8). Slika 5.9 prikazuje štiri reprezentativne skupine, ki jih prepoznamo po bolj pogostih povezavah, ki so prikazane z bolj nasičeno rdečo barvo. Te skupine zajemajo vse pomembnejše strukture, znane iz oceanografske literature npr. tok ob italijanski obali od severnega Jadrana proti Otrantskim vratom (W-MAd Current in W-Sad Current), tok ob vzhodni obali Jadranskega morja (E-SAd Current), ki teče od juga proti severu in znani vrtinci v srednjem in južnem Jadranu (MAd Gyre in SAd Gyre). V tabeli 5.4 so naštete vse vse te strukture skupaj z letnimi časi, v katerih se pojavljajo, in pripadajoče skupine (dinamične mehke poti). Histogrami na sliki 5.10 prikazujejo pogostost posameznih skupin po mesecih. Opredelitev oceanografskih letnih časov na osnovi mesecev, kakršno podaja Artegiani [52], je prikazana v tabeli 5.3. Iz histogramov je razvidno, da se poti v Jadranskem morju z vsaj tremi povezavami razvijajo predvsem v obdobju od pomladi do pozne jeseni, kar je v skladu z večino struktur na sliki 5.8. Izjema je tok vzdolž vzhodne obale južnega Jadranskega morja (*E-SAd current*), ki se običajno pojavlja jeseni, pozimi in spomladi. V našem prime-

Tabela 5.3

Opredelitev oceanografskih letnih časov na osnovi mesecev po Artegiani-ju [52]).

Letni čas	Meseci
Zima	Januar, Februar, Marec, April
Pomlad	Maj, Junij
Poletje	Julij, Avgust, September, Oktober
Jesen	November, December

ru ga predstavljata skupini 6 in 9, ki se v glavnem pojavljata od spomladi do jeseni z vrhom v oktobru.



Slika 5.8

Tipične poti in cikli v Jadranskem morju (povzeto po [52]).

5.3.2 Dinamični mehki cikli

S pomočjo algoritmov 3 in 4 (sliki 4.4 in 4.6) smo iz vseh 550.981 poti, ki smo jih našli v celotnem Sredozemlju, izločili 222.065 poti, ki imajo dolžino najmanj tri povezave, in v njih odkrili 6877 enoličnih ciklonskih, 5625 anticiklonskih and 2378 "nedefiniranih" ciklov in dodatno še 1231 zaključenih krivulj, ki sekajo same sebe. Uporabili smo različne metode hierarhičnega razvrščanja ciklonskih in anticiklonskih ciklov pri različnih kombinacijah parametrov α in β za izračun razdalje Tverskega. Rezultate smo vizualno pregledali in primerjali dobljene dinamične mehke cikle s posameznimi

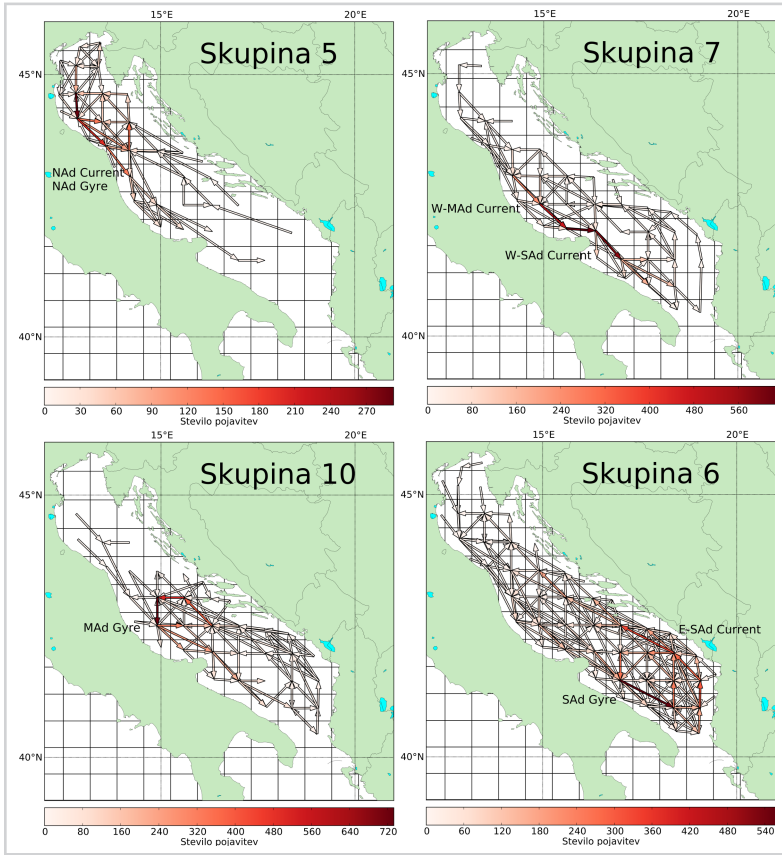
Tabela 5.4

Tipične strukture v Jadranskem morju in pripadajoče dinamične mehke poti.

Struktura	Pojavljanje	Skupine
NAd current		5
NAd gyre	jesen	5
W-MAd current	pomlad, poletje, jesen	3, 4, 7, 8, 11
MAd gyre	pomlad, poletje, jesen	3, 8, 10, 11, 12
W-SAd current	pomlad, poletje, jesen	3, 4, 7, 8, 11
SAd gyre	poletje, jesen	1, 2, 3, 4, 6
E-SAd current	jesen, zima, pomlad	6, 9

znanimi cikli (ang. *gyres*, *eddies*) iz oceanografske literature [50, 53, 54]. Na slikah iz omenjene literature smo znane cikle digitalizirali in ugotavljali njihovo ujemanje z dobljenimi dinamičnimi mehkimi cikli. Slike 5.11, 5.12 in 5.13 prikazujejo rezultate digitalizacije.

V prvem poskusu smo primerjali rezultate treh metod hierarhičnega razvrščanja (minimalna, povprečna in maksimalna) pri uporabi razdalje Tverskega s koeficienti $\alpha = 1.0$ in $\beta = 1.0$ (kar ustreza Jaccardovi razdalji). Višino rezanja dendrogramov h_{cut} smo izbrali tako, da so bili dobljeni dinamični mehki cikli čim bolj podobni vrtincem iz oceanografske literature (glej slike 5.11, 5.12 in 5.13). Iskanje optimalne višine rezanja smo začeli pri $h_{cut} = 0$, pri kateri imamo največje število mehkih ciklov, kjer vsak vsebuje le nekaj članov, in postopoma povečevali h_{cut} s korakom 0,1. Pri tem se je velikost mehkih ciklov in število njihovih članov povečevalo, po drugi strani pa se je skupno število mehkih ciklov zmanjševalo. Z iskanjem smo prenehali, ko so se posamezni cikli združili v prevelike mehke cikle, v katerih nismo mogli več prepoznati opazovanih vrtincev iz slik 5.11, 5.12 in 5.13. Optimalna vrednost h_{cut} je potemtakem zadnja vrednost zmanjšana za 0,1. Rezultati pokažejo, da minimalna metoda združi preveč ciklov tudi pri majhni višini rezanja ($h_{cut} = 0{,}20$). Število dobljenih mehkih ciklov v tem primeru znaša 1.527, kar je ogromno v primerjavi s povprečno metodo ($h_{cut} = 0{,}70$; 224 mehkih ciklov) in maksimalno metodo ($h_{cut} = 0{,}99$; 186 mehkih ciklov). Zato se zdi, da sta povprečna in maksimalna metoda bolj primerni za iskanje dinamičnih mehkih ciklov. Pri uporabi maksimalne metode dendrogram odrežemo tik pod korenom, medtem ko je pri povprečni metodi višina rezanja nekje

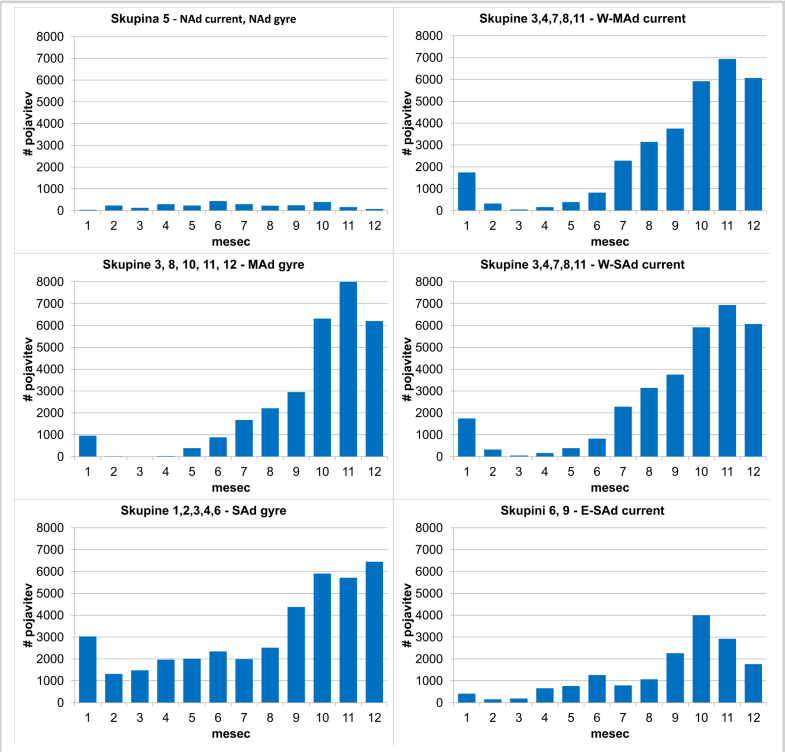


Slika 5.9

Reprezentativne skupine poti (tj. dinamične mehke poti) v Jadranskem morju. Pogostost posameznih povezav v poteh je sorazmerna z nasičenostjo rdeče barve.

med sredino in korenom dendrograma.

V drugem poskusu smo iskali dinamične mehke cikle z uporabo vseh kombinacij parametrov α in β z vrednostmi $\alpha, \beta \in \{0, 0.5, 1\}$. Za vsako od $3 \times 3 = 9$ kombinacij smo izvedli hierarhično razvrščanje ciklov z uporabo povprečne metode in odrezali dobljene dendrograme na višinah $h_{cut} = \{0, 0.1, \ldots, 0.9, 0.99\}$. Na ta način smo dobili $3 \times 3 \times 11 = 99$ rezultatov, ki smo jih vizualno pregledali in ocenili njihovo ujemanje z z nekaterimi znanimi vrtinci iz oceanografske literature (glej slike 5.11, 5.12 in 5.13). Omenjeni znani cikli so našteti v tabeli 5.5.

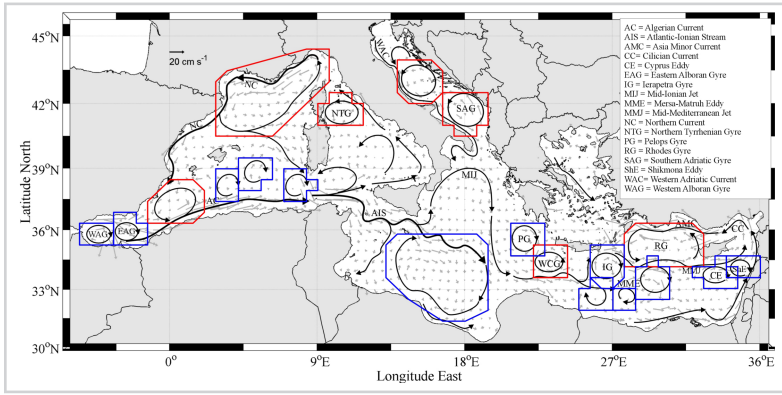


Slika 5.10

Histogrami pogostosti povezav sezonskih skupin poti iz tabele 5.4

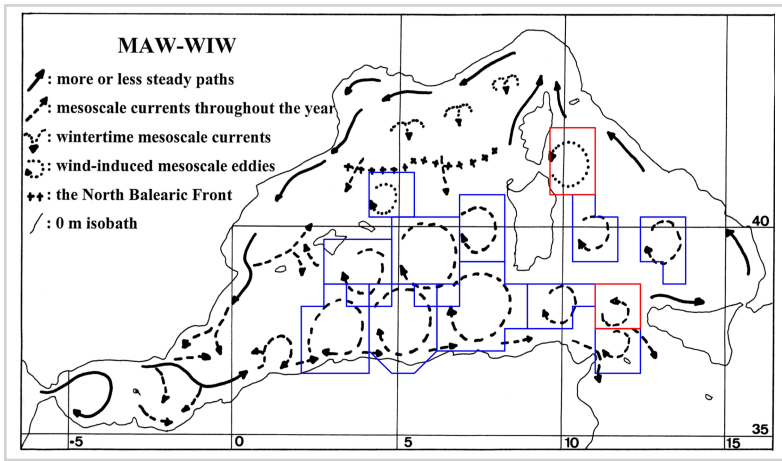
Pri ocenjevanju smo upoštevali ujemanje posameznih dinamičnih mehkih ciklov z opazovanimi v velikosti, legi in številu posameznih ciklov, ki jih vsebujejo. Uporabili smo naslednje ocene: 1 – slabo ujemanje, 2 – srednje ujemanje in 3 – dobro ujemanje. Če za določen opazovani cikel nismo dobili ustreznega mehkega cikla, potem smo ga ocenili z 0. Na koncu smo izračunali seštevek točk za vsak opazovani cikel.

Rezultati, ki se najbolje ujemajo z opazovanji iz oceanografske literature, so povzeti v tabelah 5.6 in 5.7. Iz prve tabele je razvidno, da je potrebna višina rezanja dendrograma h_{cut} močno odvisna od koeficienta α in le malo odvisna od β. Pearsonov koeficient korelacije ρ med α in h_{cut} je enak 0,92 ($p = 0{,}0002$), medtem, ko je ρ med β in h_{cut} enak 0,23 ($p = 0{,}28$). Iz tabele 5.7 je razvidno, da je skupna ocena povezana s



Slika 5.11

Znani vrtinci v Sredozemlju (povzeto iz [53]). Digitalizirani ciklonski cikli so obarvani z rdečo barvo, anticiklonski pa z modro.



Slika 5.12

Znani vrtinci v zahodnem Sredozemlju (povzeto iz [50]). Digitalizirani ciklonski cikli so obarvani z rdečo barvo, anticiklonski pa z modro.

koeficientom α ($\rho = 0{,}70$; $p = 0{,}02$), ni pa povezana z β ($\rho = 0{,}31$; $p = 0{,}21$), kar pomeni, da je pri iskanju podobnosti dinamičnih mehkih ciklov in opazovanih ciklov pomembna razlika med večjimi in manjšimi cikli, medtem ko razliki med manjšimi in večjimi cikli ne pripisujemo velikega pomena.

Najboljši rezultati, ki so navedeni v tabelah 5.6 and 5.7, so predstavljeni na sliki 5.14, kjer prikazujemo samo pogoste dinamične mehke cikle, ki vsebujejo vsaj 270 posameznih ciklov (članov). S tem se izognemo pretirani gneči na slikah zaradi ciklov,

Tabela 5.5

Nekateri vrtinci, znani iz oceanografske literature.

Ime	Kratica	Tip
Northern Tyrrhenian Gyre	NTG	ciklonski
Southern Adriatic Gyre	SAG	ciklonski
Pelops Gyre	PG	anticiklonski
West Cretan Gyre	WCG	ciklonski
Mersa-Matruh Eddy	MME	anticiklonski
Rhodes Gyre	RG	ciklonski
West Cyprus	WCy	ciklonski
Cyprus Eddy, Shikmona Eddy	CE, ShE	anticiklonski

Tabela 5.6

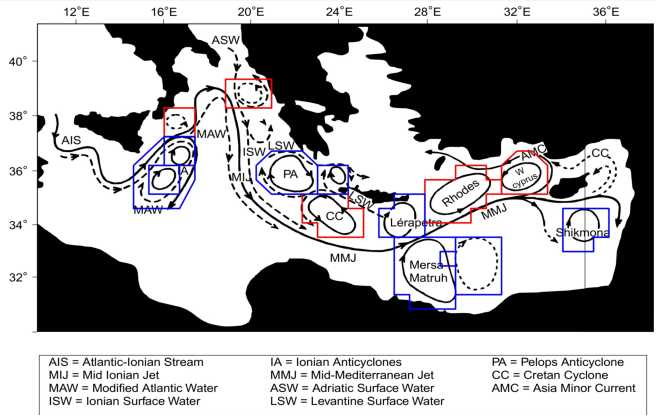
Število ciklonskih in anticiklonskih mehkih ciklov v odvisnosti od parametrov α in β pri ustrezni višini reza dendrogramov h_{cut}. Zadnji stolpec prikazuje število mehkih ciklov, ki vsebujejo najmanj 270 posameznih ciklov.

α	β	h_{cut}	#ciklonskih	#anticiklonskih	#skupaj	#skupaj(≥ 270)
0,00	0,00	0,0	174	144	318	52
0,00	0,50	0,2	228	212	440	52
0,00	1,00	0,2	381	332	713	37
0,50	0,00	0,5	202	171	373	54
0,50	0,50	0,5	281	239	520	47
0,50	1,00	0,6	227	189	416	54
1,00	0,00	0,6	267	225	492	54
1,00	0,50	0,7	199	173	372	57
1,00	1,00	0,7	224	196	420	53

Tabela 5.7

Ocena ujemanja mehkih ciklov z izbranimi cikli iz oceanografske literature (glej slike 5.11, 5.12 in 5.13) v odvisnosti od parametrov α in β. Zadnji stolpec prikazuje seštevek točk za vse posamezne opazovane cikle.

α	β	NTG	SAG	PG	WCG	MME	RG	WCy	CE, ShE	Ocena
0,00	0,00	3	1	2	2	2	1	0	2	13
0,00	0,50	3	1	2	2	2	2	0	2	14
0,00	1,00	3	1	2	2	1	1	0	1	11
0,50	0,00	3	2	3	3	2	2	1	1	17
0,50	0,50	3	2	2	3	2	3	0	2	17
0,50	1,00	3	2	3	3	2	3	1	2	19
1,00	0,00	3	2	2	3	2	1	0	1	14
1,00	0,50	3	3	3	3	2	3	0	2	19
1,00	1,00	3	3	3	3	2	3	1	3	21



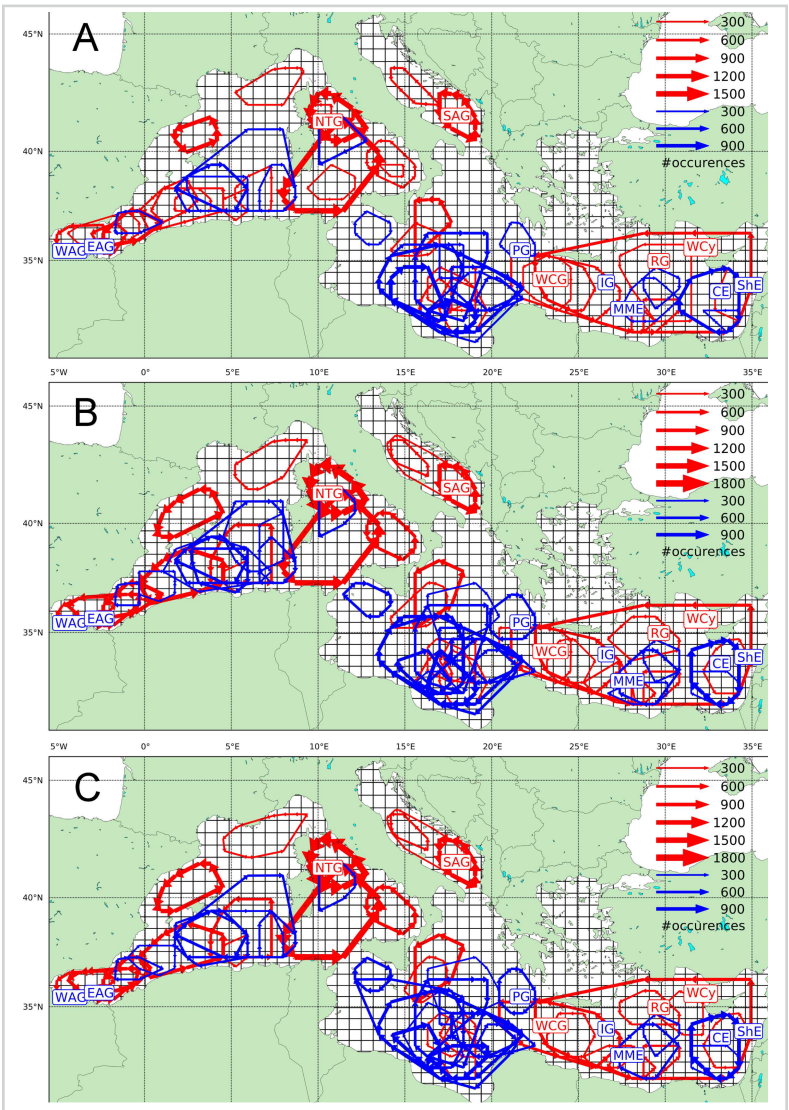
Slika 5.13

Znani vrtinci v vzhodnem Sredozemlju (povzeto iz [54]). Digitalizirani ciklonski cikli so obarvani z rdečo barvo, anticiklonski pa z modro.

ki niso preveč pogosti. Poleg tega slika 5.15 ilustrira še primerjavo mehkih ciklov s slike 5.14 c z opazovanimi cikli iz [52] v Jadranskem morju. Treba pa je poudariti, da tudi cikli, ki so jih podali oceanografski eksperti, niso popolnoma točni, ampak bolj kvalitativne narave, zato naši rezultati bistveno ne odstopajo od opazovanj. Vsi rezultati pokažejo, da sta najbolj pogosta cikla Northern Tyrrhenian Gyre (NTG) in Southern Adriatic Gyre (SAG). Prvega v glavnem povzroči veter [50], slednjega pa v prvi vrsti nadzoruje topografija morskega dna. Rezultati jasno pokažejo tudi druge pomembne cikle, čeprav niso povsem skladni z opazovanimi cikli. Nekaterih znanih ciklov, kot so npr. Eastern Alboran Gyre (EAG), Western Alboran Gyre (EAG) in Ierapetra Gyre (IG), s to metodo nismo dobili, zato jih nismo vključili v ovrednotenje rezultatov.

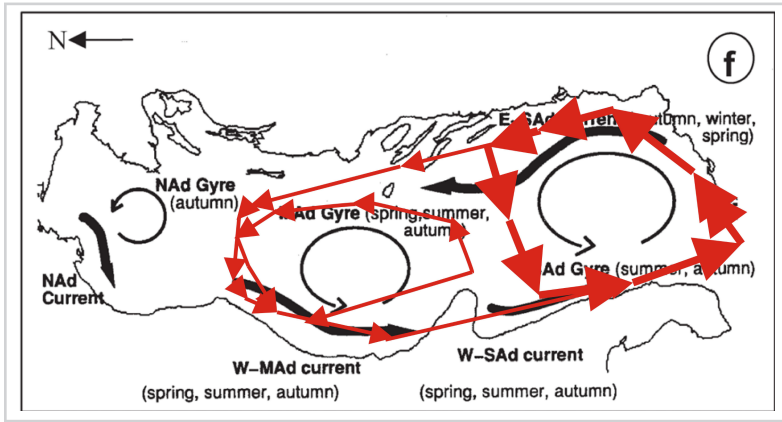
5.3.3 *Samodejno določanje višine rezanja dendrogramov*

V razdelku 5.3.2 smo iskali dinamične mehke cikle s pomočjo povprečne metode hierarhičnega razvrščanja posameznih dinamičnih ciklov in pri tem uporabili kombinacije parametrov α in β z vrednostmi $\alpha, \beta \in \{0, 0.5, 1\}$. Dobljene dendrograme smo odrezali na višinah $h_{cut} = \{0, 0.1, \ldots, 0.9, 0.99\}$ in z vizualnim pregledovanjem iskali takšne kombinacije parametrov α, β in h_{cut}, pri katerih so se dobljeni dinamični mehki cikli po velikosti in legi najbolje ujemali z nekaterimi opazovanimi vrtinci iz oceanografske literature (tabela 5.5). Pri vrednotenju dobljenih dinamičnih mehkih ciklov smo upoštevali tudi število vsebovanih posameznih ciklov.



Slika 5.14

Mehki cikli, dobljeni s povprečno metodo hierarhičnega razvrščanja pri uporabi razdalje Tverskega s parametri $\alpha = 1.00$: A) $\beta = 0.00$ ($h_{cut} = 0.60$)B) $\beta = 0.50$ ($h_{cut} = 0.70$), C) $\beta = 1.00$ ($h_{cut} = 0.70$). Kratice imen opazovanih ciklov, ki ustrezajo mehkim ciklov, so navedene v tabeli 5.5. Ciklonski cikli so obarvani rdeče, anticiklonski pa modro.



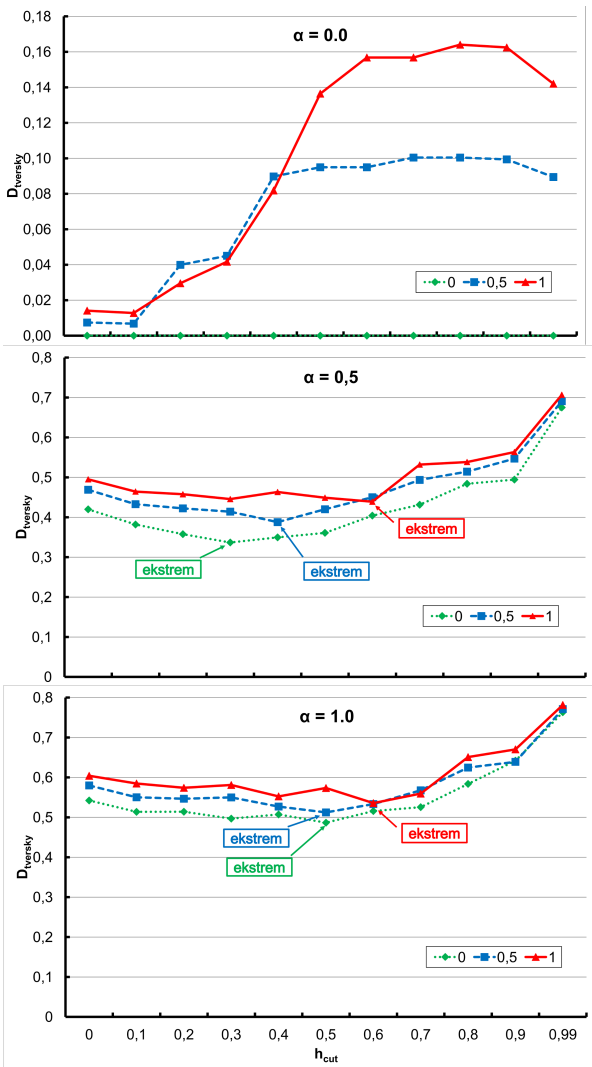
Slika 5.15

Mehki cikli, ki odgovarjajo opazovanim ciklom MAd Gyre in SAd Gyre in so bili dobljeni s povprečno metodo hierarhičnega razvrščanja pri uporabi razdalje Tverskega s parametri $\alpha = 1.00$ in $\beta = 1.00$ ($h_{cut} = 0.70$) (glej sliko 5.14 c) skupaj z opazovanimi strukturami iz [52] v Jadranskem morju.

V tem razdelku opišemo postopek, ki za določeno kombinacijo parametrov α in β samodejno določi ožji interval višine rezanja h_{cut}, na osnovi katerega oceanografski ekspert poišče najboljšo vrednost h_{cut}. Pri tem za vsako kombinacijo parametrov α, β in h_{cut} in za vsak opazovani vrtinec iz tabele 5.5 poiščemo dinamični mehki cikel, ki ima najmanjšo razdaljo Tverskega do omenjenega opazovanega vrtinca pri danih parametrih α in β. Za potrebe tega postopka smo opazovane vrtince iz iz tabele 5.5 digitalizirali (glej slike 5.11, 5.12 in 5.13). Pri opazovanih vrtincih upoštevamo vse možne vire iz literature, npr. v primeru opazovanega vrtinca Pelops Gyre vzamemo vrtinec PG iz slike 5.11 in tudi PA iz slike 5.13. Za posamezne kombinacije parametrov α, β in h_{cut} izračunamo povprečje dobljenih minimalnih razdalj Tverskega za vse opazovane vrtince iz tabele 5.5.

Razdalj Tverskega pri različnih vrednostih parametrov α in β ne moremo med seboj neposredno primerjati, lahko pa primerjamo razdalje pri isti kombinaciji α in β pri različnih vrednostih h_{cut}. Slika 5.16 prikazuje odvisnost razdalje Tverskega od višine rezanja dendrograma h_{cut} pri danih kombinacijah parametrov α in β.

Iz slik 5.16 B ($\alpha = 0.50$)in 5.16 C ($\alpha = 1.00$) je razvidno, da se pri $\alpha > 0.00$ razdalja Tverskega v odvisnosti od h_{cut} zmanjšuje do določenega minimuma (na sliki je označen kot "ekstrem") in potem narašča do $h_{cut} = 0.99$. Najprimernejša izbira višine rezanja dendrograma h_{cut} se torej nahaja v okolici dobljenega ekstrema. Za kombinacije parametrov α in β na slikah 5.16 B in 5.16 C predlagamo intervale najboljših vrednosti



Slika 5.16

Odvisnost povprečja razdalj Tverskega med opazovanimi cikli (tabela 5.5) in dobljenimi dinamičnimi mehkimi cikli (razdelek 5.3.2) od višine rezanja dendrograma h_{cut} pri danih koeficientih α in β. Slika A prikazuje vrednosti pri $\alpha = 0.00$, slika B pri $\alpha = 0.50$ in slika C pri $\alpha = 1.00$. Zelena pikčasta črta prikazuje vrednosti pri $\beta = 0.00$, modra prekinjena črta pri $\beta = 0.50$ in polna rdeča črta pri $\beta = 1.00$. Na posameznih krivuljah so označene tudi minimalne vrednosti razdalje Tverskega (ekstremi).

Tabela 5.8

Samodejno in ročno določanje višine rezanja dendrograma pri povprečni metodi hierarhičnega razvrščanja ciklov v odvisnosti od izbire koeficientov α in β v razdalji Tverskega. Pri samodejnem določanju je podan interval vrednosti h_{cut}, pri ročnem pa je podana vrednost h_{cut}, ki smo jo izbrali na podlagi vizualnega pregleda rezultatov. V zadnjih dveh stolpcih je podano povprečno število posameznih ciklov, ki jih vsebujejo obravnavani dinamični mehki cikli.

α	β	h_{cut} (samodejno)	h_{cut} (ročno)	#ciklov (samodejno)	#ciklov (ročno)
0,00	0,00	n/a	0,0	n/a	304
0,00	0,50	n/a	0,2	n/a	403
0,00	1,00	n/a	0,2	n/a	147
0,50	0,00	(0,2-0,4)	0,5	(51-265)	403
0,50	0,50	(0,3-0,5)	0,5	(102-411)	411
0,50	1,00	(0,5-0,7)	0,6	(373-624)	540
1,00	0,00	(0,4-0,6)	0,6	(112-298)	298
1,00	0,50	(0,4-0,6)	0,7	(92-310)	496
1,00	1,00	(0,5-0,7)	0,7	(166-487)	487

h_{cut}, ki so podani v tabeli 5.8.

Iz tabele 5.8 je razvidno, da smo pri vizualnem pregledu rezultatov izbrali nekoliko višje vrednosti za h_{cut} oziroma vrednosti zgornje meje intervalov, ki so podani v tretjem stolpcu tabele. Priporočljivo je, da oceanografski ekspert pregleda rezultate v okolici ekstremov, ker je ujemanje povprečja razdalj dobljenih dinamičnih mehkih ciklov do opazovanih vrtincev odvisno od tega, kako kvalitetno so eksperti podali omenjene opazovane vrtince. Poleg tega smo pri vrednotenju uporabili različne vire iz literature za opazovane vrtince, v katerih so omenjeni vrtinci precej različni.

Iz slike 5.16 A je razvidno, da je opisani postopek neuporaben pri $\alpha = 0.00$, kjer ne moremo določiti izrazitih ekstremov. V tem primeru se posamezni dinamični cikli pri $\beta > 0.00$ združujejo v prevelike dinamične mehke cikle že pri nizkih vrednostih h_{cut}. Pri $\alpha = 0.00$ in $\beta = 0.00$ pa se združijo vsi cikli, ki se vsaj malo prekrivajo.

Če imamo na voljo kvalitetno oceanografsko predznanje na določenih delih domene (npr. dobro poznamo nekaj opazovanih vrtincev), potem lahko s pomočjo opisane metode dobimo dinamične mehke cikle tudi na drugih delih domene, kjer omenjenega predznanja nimamo na voljo. Problem nastopi pri domenah, kjer oceanografskega predznanja praktično ni. V tem primeru lahko oceanografski ekspert oceni (npr. na osnovi topografije morskega dna) značilno dolžino vrtincev, ki jih tam pričakuje. Pri tem ekspert za dano domeno oceni spodnjo in zgornjo mejo dolžine vrtincev, ki ju po vrsti označimo z D_{min} in D_{max}. Značilne dolžine dinamičnih mehkih ciklov podamo

s pomočjo izraza:

$$D = \sqrt{D_x^2 + D_y^2} \tag{5.1}$$

kjer sta D_x in D_y po vrsti dolžini cikla v x in y smeri. Za dobljene dendrograme pri hierarhičnem razvrščanju posameznih dinamičnih ciklov potem poiščemo intervale vrednosti $(h_{cut_{min}}, h_{cut_{max}})$, za katere velja $D_{min} \leq D \leq D_{max}$ za vsak dobljeni dinamični mehki cikel z dolžino D. S pomočjo opisanega kriterija preverimo vse dobljene dinamične mehke cikle ali pa samo tiste, ki vsebujejo minimalno število posameznih ciklov, vendar moramo to število še določiti.

5.3.4 Diskusija

V tem poglavju smo prikazali veliko uporabnost rudarjenja večnivojskih usmerjenih grafov za analizo prostorskih podatkov. Začeli smo iskanjem enostavnih ciklov v posameznih (mesečnih) večnivojskih usmerjenih grafih. Za vse te grafe velja, da so verjetnosti prehoda med vozlišči (uteži povezav) konstantne v okviru enega meseca in veljajo za časovni korak $\Delta t = 6$ dni. Zaradi tega imamo v vsakem mesecu 5 prehodov med vozlišči grafa. V teh grafih smo poiskali ciklonske in anticiklonske cikle in preučevali njihovo pogostost v različnih obdobjih znotraj leta. Cikli, ki smo jih dobili, so bili v veliki večini krajši od petih povezav, kar pomeni, da imajo pri časovnem koraku Δt=6 dni dolžino krajšo od enega meseca, kar opravičuje iskanje ciklov znotraj enega grafa.

Za potrebe iskanja vzorcev gibanja vodnih mas, ki bi bili bolj realistični in primerljivi z vzorci, ki jih navaja oceanografska literatura, smo razvili metodo odkrivanja dinamičnih mehkih poti in ciklov, ki se razvijejo v obdobjih, dolgih po več mesecev vendar krajših od enega leta. Pri tem smo morali upoštevati, da se omenjene poti in cikli raztezajo preko časovne vrste večnivojskih usmerjenih grafov in se zato spreminjajo uteži povezav omenjenih grafov.

Po ustrezni nastavitvi parametrov α in β, ki omogočata fleksibilno prilagajanje procesa rudarjenja, dobimo mehke cikle, ki so primerljivi z znanimi strukturami (tj. NTG, SAG itd.) iz oceanografske literature. Tukaj moramo upoštevati dejstvo, da se učimo iz rezultatov numeričnega modela in ne iz opazovanih podatkov. Oceanografski numerični modeli imajo namreč določene omejitve zaradi svoje parametrizacije. Kljub temu lahko za tipične cikle v Sredozemskem morju (tj. NTG, SAG, itd.) dobimo s predlagano metodo rudarjenja dobre rezultate.

Vendar pa ta metoda ne more samodejno določiti ustrezne višine rezanja dendrogra-

mov. To velja tako za dinamične mehke poti kot za cikle. Celo najpreprostejše pravilo tj. rezanje dendrograma, kjer je razlika višin med posameznimi poddrevesi največja, tukaj ni uporabno. Zato moramo višino rezanja določiti sami s pomočjo vizualnega pregleda rezultatov za različne kombinacije parametrov α in β in za različne višine rezanja h_{cut}. Pri ocenjevanju velikosti, oblike in pogostosti dobljenih mehkih struktur se zanašamo na predhodno oceanografsko ekspertno znanje. Če slednjega ni na voljo, potem ta metoda omogoča uporabniku, da analizira različne kandidate med strukturami, dobljenimi z različnimi vrednostmi parametrov. Naše priporočilo je, da vzamemo višjo vrednost parametra α (ugotovili smo, da je $\alpha = 1.0$ najprimernejša privzeta vrednost) in postopoma pregledujemo rezultate pri postopnem povečevanju h_{cut} od $h_{cut} = 0$ proti višjim vrednostim.

Da bi odpravili pomanjkljivosti, opisane v prejšnjem odstavku, si pri dinamičnih mehkih ciklih pomagamo s postopkom samodejnega določanja višine rezanja dendrogramov (glej razdelek 5.3.3). S pomočjo tega postopka dobimo ožji interval višine rezanja h_{cut} okoli ekstrema, kjer je razdalja Tverskega med dobljenimi dinamičnimi mehkimi cikli in opazovanimi cikli najmanjša. Dobljeni interval je precej blizu vrednostim višine rezanja, ki smo jih določili z vizualnim pregledom ujemanja izračunanih in opazovanih ciklov. Na ta način oceanografski ekspert pregleda samo rezultate na dobljenem intervalu rezanja, kar pomeni znaten časovni prihranek pri pregledu rezultatov.

5.4 Čas izvajanja algoritmov

Rezultate, ki smo jih opisali v tem poglavju, smo dobili s pomočjo algoritmov, opisanih v poglavju 4, ki so bili v večini implementirani na prenosni delovni postaji HP EliteBook 8540w (v nadaljevanju HP EliteBook). Omenjena delovna postaja vsebuje dvojedrni procesor Intel® Core™ i7 2.67 GHz in 8 GB pomnilnika ter deluje pod operacijskim sistemom Windows 7 Professional. V zvezi z izmerjenimi časi izvajanja algoritmov velja opozoriti, da so omenjeni algoritmi trenutno v fazi prototipov in še niso primerni za široko uporabo. Pri implementaciji smo uporabili programski orodji R verzija 3.0.1 (64-bit) [55] in Python 2.7 [56] s pripadajočimi knjižnicami, ki so bile potrebne za izvedbo hierarhičnega razvrščanja in operacij računske geometrije. Raziskovalci pogosto uporabljajo omenjeni orodji, ker je mogoče s pomočjo njiju elegantno in učinkovito reševati probleme z uveljavljenimi metodami strojnega učenja in statistike z uporabo pripadajočih knjižnic. Nekatere algoritme, ki ne zahtevajo posebnih

knjižnic (npr. algoritem Lin-Och – glej razdelek 4.2.3), smo zaradi krajšega časa izvajanja implementirali na računalniku s procesorjem Intel® Xeon® CPU E5440 2.83 GHz s predpomnilnikom velikosti 6 MB in 8 GB glavnega pomnilnika. Omenjeni računalnik pripada t. i. numerični gruči računalnikov (v nadaljevanju numerični računalnik). Algoritem Lin-Och smo implementirali v okolju Linux v programskem jeziku C++ proizvajalca Portland Group.

Opisana implementacija se je izkazala za učinkovito pri razvoju algoritmov, opisanih v poglavju 4, vendar, kot rečeno, še ni primerna za končnega uporabnika. V tabeli 5.9 so navedeni časi izvajanja opravil, ki smo jih opisali v razdelkih 5.2, 5.3.1 in 5.3.2, pripadajoči algoritmi ter uporabljena strojna in programska oprema.

Iz tabele 5.9 je razvidno, da imajo algoritmi 1–4 in algoritem Lin-Och solidne čase izvajanja. Daljši čas izvajanja ima algoritem 5, kjer ima najslabšo učinkovitost računanje povprečja Minkowskega. Za n prekrivajočih se ciklov, ki po vrsti vsebujejo n_1, n_2, ... , n_n vozlišč, ima izračun povprečja Minkowskega s pomočjo enačbe (4.13) časovno zahtevnost $O(n_1 \times n_2 \times ... \times n_n)$, kar močno podaljša čas izvajanja pri združevanju velikega števila posameznih ciklov v mehki cikel. V disertaciji nismo optimizirali algoritma za izračun povprečja Minkowskega za cikle zaradi težav pri določitvi pravilne lege mehkega cikla, ki ga dobimo kot rezultat, in se bomo temu posvetili pri bodočem delu. Zaradi slabše učinkovitosti algoritma 5, ki se izvaja za vsako kombinacijo parametrov α in β okoli 35 minut v primeru ciklonskih in 17 minut v primeru anticiklonskih ciklov, je izračun vseh primerov, navedenih v razdelku 5.3.2, trajal več ur.

Tabela 5.9

Časi izvajanja posameznih algoritmov, opisanih v poglavju 4, ki se nanašajo na rezultate v tem poglavju. V prvem stolpcu so v strnjeni obliki opisana opravila, v drugem stolpcu pa so navedeni pripadajoči algoritmi, ki so označeni na enak način kot v poglavju 4. Poleg tega je navedena še uporabljena strojna in programska oprema s pripadajočimi knjižnicami.

Opravilo	Algoritem	Programsko orodje	Strojna oprema	Skupni čas CPE (s)	Skupni čas CPE (mm:ss)
Iskanje enostavnih ciklov v 156 mesečnih grafih	Algoritem 1	R	HP EliteBook	160,86	02:41
Iskanje dinamičnih poti v snopu (M=2) z začetkom v vsakem od 156 * 5 = 780 časovnih intervalov iz vsakega vozlišča posameznih mesečnih grafov	Algoritem 2	R	HP EliteBook	16733,92	38:54
Izračun razdalj WLCS med 222.065 potmi, dobljenimi z Algoritmom 2, ki imajo dolžino vsaj tri povezave	Algoritem Lin-Och	C++	numerični računalnik	1930,00	32:10
Hierarhično razvrščanje (Ward-ova metoda) podmnožice poti, ki se nahajajo v Jadranskem morju (10143 poti)	hclust - R knjižnica stats	R	HP EliteBook	7,17	00:07
Rezanje dendograma in tvorba mehkih poti iz 10.143 posameznih poti	hclust - R knjižnica stats	R	HP EliteBook	292,81	04:53
Iskanje dinamičnih ciklov v 222.065 poteh	Algoritem 3	R	HP EliteBook	109,61	01:50
Poenostavljanje dobljenih dinamičnih ciklov	Algoritem 4	R	HP EliteBook	553,67	09:14
Izračun razdalje Tverskega med 6877 poenostavljenimi ciklonskimi cikli za eno kombinacijo parametrov α in β	Algoritem 5	Python, knjižnica shapely	HP EliteBook	74,40	01:14
Izračun razdalje Tverskega med 5625 poenostavljenimi anticiklonskimi cikli za eno kombinacijo parametrov α in β	Algoritem 5	Python, knjižnica shapely	HP EliteBook	45,60	00:46
Hierarhično razvrščanje 6877 ciklonskih ciklov s povprečno metodo za eno kombinacijo parametrov α in β, rezanje dendrograma na 11 višinah, izračun povprečja Minkowskega in konveksne ogrinjače za vsako višino rezanja	Algoritem 5, hclust - R knjižnica stats	R	HP EliteBook	2106,67	35:07
Hierarhično razvrščanje 5625 anticiklonskih ciklov s povprečno metodo za eno kombinacijo parametrov α in β, rezanje dendrograma na 11 višinah, izračun povprečja Minkowskega in konveksne ogrinjače za vsako višino rezanja	Algoritem 5, hclust - R knjižnica stats	R	HP EliteBook	1020,00	17:00

Uporaba večnivojskih usmerjenih grafov

6

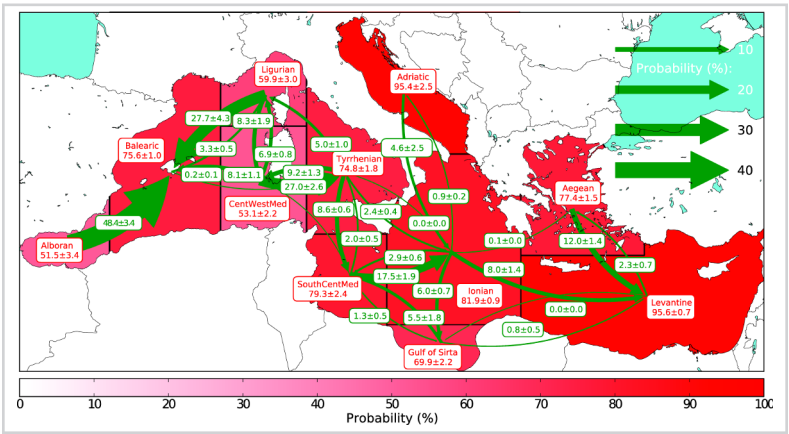
V tem poglavju opisujemo nekaj uspešnih uporab večnivojskih usmerjenih grafov, ki so nastale kot posledica raziskovalnih vprašanj strokovnjakov s področja oceanografije. Aplikacije večnivojskih usmerjenih grafov zajemajo tako njihovo rudarjenje kot tudi uporabo statistike in metod oceanografske podatkovne analize na atributih vozlišč in povezav v grafih. Pri prikazu posameznih aplikacij se še vedno omejimo na domeno, ki zajema gibanje površinskih vodnih mas v Sredozemskem morju. Pri tvorbi večnivojskih usmerjenih grafov se opremo tako na rezultate numeričnega modela Mediterranean Ocean Forecasting System (MFS) [9] kot tudi na opazovanja s pomočjo plovcev [53].

6.1 Periodičnost prehodov vodnih mas med morskimi območji

Čeprav so bile že opravljene številne analize gibanja vodnih mas v različnih območjih Sredozemskega morja in tudi med samimi območji [49–51], se vseeno pokaže potreba po bolj preprostem pogledu na izmenjavo vodnih mas med območji v Sredozemskem morju v krajših časovnih obdobjih npr. en mesec. To vprašanje je zelo pomembno npr. zaradi širjenja onesnaževal in potovanja ladijskih razbitin v morju in je zato tudi motivacija za uporabo večnivojskih usmerjenih grafov. Koristno je npr. vedeti, kolikšna je sezonska amplituda verjetnosti izmenjave vodnih mas med območji v primerjavi s "stabilno" (letno) komponento, če analiziramo premike navideznih delcev na mesečni ravni. Prav raziskovanje te sezonskosti je motivacija za uporabo Fourierjeve analize uteži povezav v večnivojskih usmerjenih grafih.

V tem razdelku opisujemo primer, ki prikazuje periodičnost prehodov površinskih vodnih mas med območji v Sredozemskem morju v obdobju 1999–2011 [28, 29]. V ta namen smo domeno numeričnega modela MFS razdelili na večja območja, ki v grobem ustrezajo geografskim morjem (Jadransko morje, Jonsko morje itd.). V tem primeru smo izbrali časovni interval $\Delta t = 30$ dni, da bi lahko pozneje primerjali rezultate te aplikacije še z drugimi mesečnimi oceanografskimi analizami. Zato smo v tem primeru izpustili postopek za iskanje optimalnega Δt, ki je opisan v razdelku 3.5.

S pomočjo prostorsko-časovnih povezovalnih pravil smo dobili 156 (13 let po 12 mesecev) usmerjenih grafov za vsak par (leto, mesec) v obdobju 1999–2011, z vozlišči, ki predstavljajo posamezna morja in povezavami, uteženimi z verjetnostmi prehodov med posameznimi morji. Opazovanje časovnega zaporedja omenjenih grafov nam daje namig, da vrednosti uteži povezav v grafih periodično nihajo in smo zato izvedli Fourierevo analizo uteži enakoležnih povezav v časovni vrsti grafov. Izkaže se, da je prva perioda enaka 12 mesecev, kar nakazuje, da ima hitrostno polje v numeričnem mode-



Slika 6.1

Usmerjeni graf, dobljen s Fourierjevo analizo časovne vrste verjetnosti prehodov med geografskimi morskimi območji (zelene vrednosti) in znotraj istih morskih območij (rdeče vrednosti) za obdobje 1999–2011. Prva vrednost znotraj okvirja označuje srednjo verjetnost prehoda, druga vrednost pa amplitudo nihanja okoli srednje vrednosti s periodo 12 mesecev. Vrednosti uteži zank v usmerjenem grafu so sorazmerne z nasičenostjo rdeče barve, debeline puščic pa so sorazmerne z verjetnostjo prehoda med morskimi območji.

lu MFS sezonske značilnosti. Usmerjeni graf, dobljen s pomočjo Fourierjeve analize, prikazuje slika 6.1. Iz slike lahko razberemo, da je povprečna verjetnost prehoda npr. znotraj Jadranskega morja 95,4 %, amplituda nihanja okoli te vrednosti (obdobje 12 mesecev) pa 2,5 %; povprečna verjetnost prehoda med Jadranskim in Jonskim morjem je samo 4,6 %, amplituda nihanja pa tako kot prej 2,5 %. Opazimo lahko tudi, da je doseženo največje letno nihanje verjetnosti prehodov (4,3 %) med Ligurskim in Balearskim morjem. Največje verjetnosti, da vodne mase ostanejo na istih območjih (95,4 %–95,6 %), ugotovimo v primerih Jadranskega in Levantinskega bazena (vzhodno Sredozemlje).

6.2 Korelacija verjetnosti prehodov z vetrno energijo

Za vsa morska območja na sliki 6.1 smo za vsak mesec v obdobju 1999–2011 izračunali delo, ki ga opravijo vetrovi in površinski morski tokovi, kot skalarni produkt med vektorji vetrne napetosti in površinskih morskih tokov, ki smo jih dobili iz rezultatov modela MFS [29]. Izračunano delo smo dodali kot atribut vozliščem obstoječih grafov in izračunali Pearsonov korelacijski koeficient med opravljenim delom v danih območjih in verjetnostjo prehoda iz teh območij v sosednja. Pri tem smo upoštevali stopnjo značilnosti $\alpha = 0.05$ in izločili tiste korelacije, ki niso bile statistično značilne. Tabela 6.1 povzema rezultate izračuna korelacij. Rezultat tega izračuna je graf z identičnimi vozlišči, kot jih vsebujejo vhodni grafi in povezavami, ki so utežene z vrednostjo

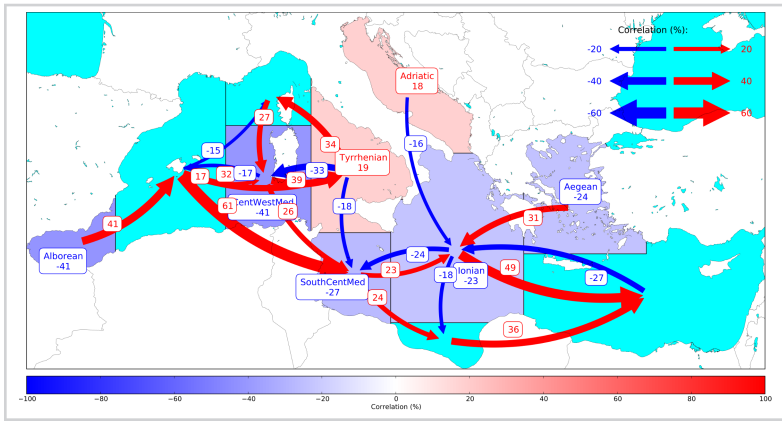
Tabela 6.1

Opisna statistika korelacije z vetrno energijo

Št. grafov	Št. vozlišč	Št. povezav			Korelacija		
		Min.	Mediana	Maks.	Veljavnih	Min.	Maks.
156	11	21	42	48	28	-41,4%	61,1%

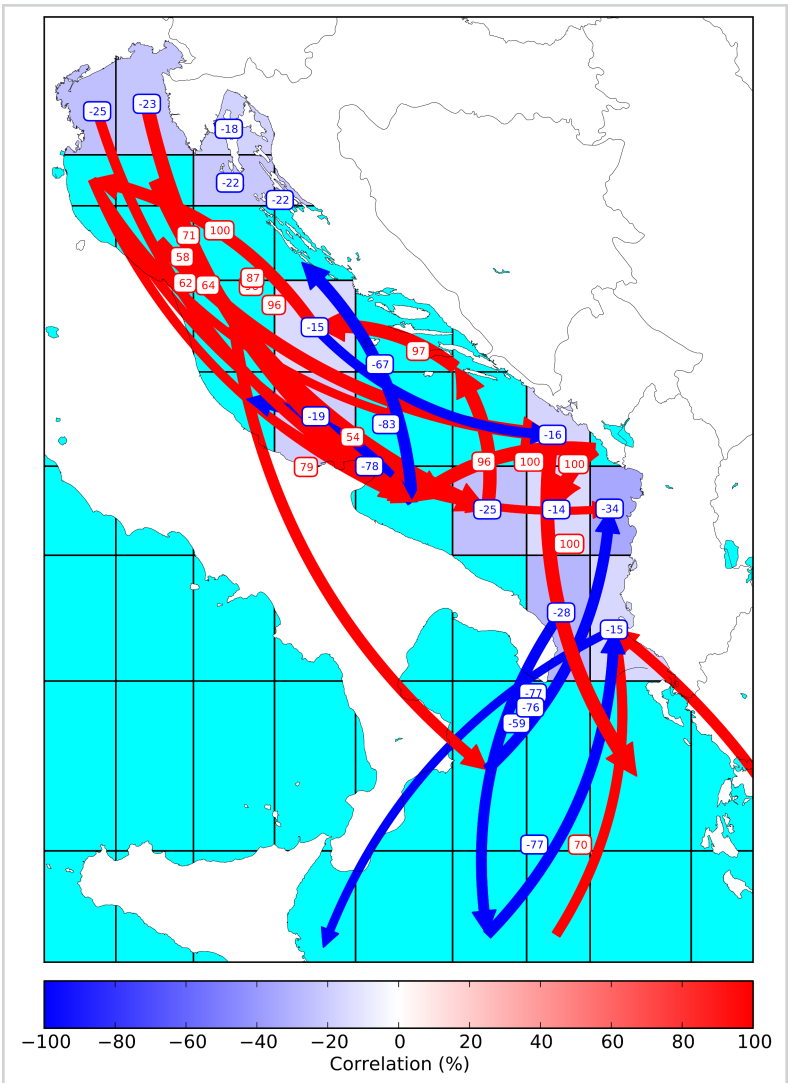
Slika 6.2

Usmerjeni graf korelacij med povprečno vetrno energijo, ki je bila prenesena v ocean v posameznih morskih območjih in verjetnost prehoda med temi območji v obdobju 1999–2011. Pozitivne korelacije so prikazane z rdečo, negativne pa z modro barvo. Nasičenost barve posameznih morskih območij je sorazmerna s korelacijo med vetrno močjo in verjetnostjo, da delci ostanejo v tistem območju v določenem časovnem intervalu. Korelacija med vetrno energijo, oddano v ocean, in verjetnostjo prehoda med različnimi morskimi območji, je sorazmerna z debelino povezav.



Pearsonove korelacije. Nastali graf je predstavljen na sliki 6.2.

Iz slike 6.2 je razvidno, da večja povprečna moč vetra, ki se prenese v morje, povzroči večjo verjetnost, da se delci premaknejo iz danega območja v sosednja. Izjemi sta Jadransko in Tirensko morje, ki se obnašata kot zaprti območji in v katerih veter povzroča kroženje površinskih delcev samo znotraj njih. Za podrobnejšo analizo zadnje trditve smo domeno numeričnega modela MFS razdelili bolj podrobno in dobili s pomočjo prostorsko-časovnih povezovalnih pravil 156 usmerjenih grafov, kjer je imel vsak po 235 vozlišč in približno 2.000 povezav. Na teh bolj podrobnih grafih smo potem izračunali korelacijo. Slika 6.3 prikazuje povečan del grafa, ki pokriva Jadransko morje in del Jonskega morja. Gibanje delcev znotraj Jadranskega morja je zelo povezano z vetrno energijo, kar pa ne velja za delce, ki prečkajo mejo med Jadranskim in Jonskim morjem.



Slika 6.3

Korelacija med povprečno vetrno energijo in verjetnostjo premika delcev v Jadranskem in Jonskem morju, prikazana v podrobnejšem grafu. Slika prikazuje le korelacije, ki so po absolutni vrednosti večje ali enake 50 % in pri katerih se začetna ali končna vozlišča nahajajo v Jadranskem morju.

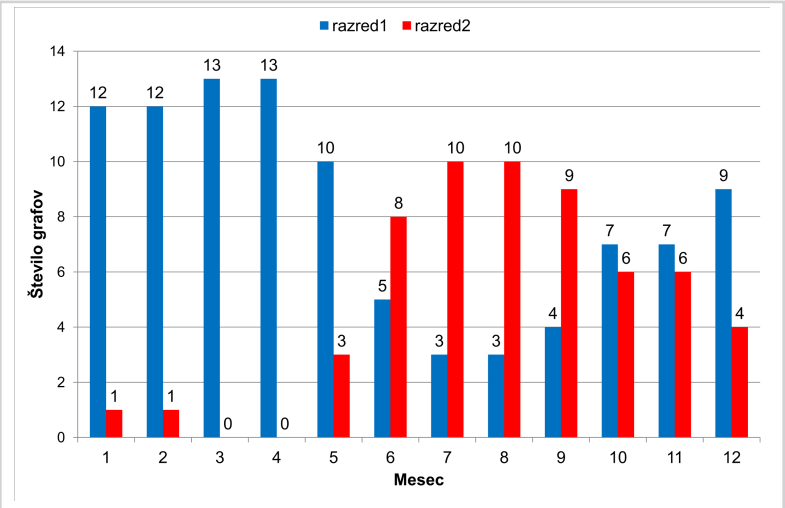
6.3 Sezonskost večnivojskih usmerjenih grafov

Številne študije [57–60] so pokazale, da ima izmenjava vodnih mas v Sredozemskem morju izrazito sezonski značaj. Naš cilj je, da to potrdimo s pomočjo hierarhičnega razvrščanja 156 mesečnih večnivojskih usmerjenih grafov, dobljenih iz rezultatov numeričnega modela MFS v obdobju 1999–2011 [28]. V tem primeru smo tako razvrstili matrike prehodov, ki vsebujejo uteži (verjetnosti prehodov delcev) iz 156 mesečnih večnivojskih usmerjenih grafov, dobljenih iz rezultatov numeričnega modela MFS (glej razdelek 5.1 in sliko 5.3). Omenjeni grafi vsebujejo po 848 vozlišč, verjetnosti prehodov pa so izračunane za časovni interval $\Delta t = 6$ dni. Tukaj smo uporabili Ward-ovo metodo hierarhičnega razvrščanja, ki za razliko od ostalih pristopov (minimalna, povprečna in maksimalna metoda) minimizira skupno varianco znotraj skupine. Druge omenjene metode uporabljajo Evklidsko razdaljo med skupinami. Hierarhično razvrščanje, ki uporablja Ward-ovo metodo, razdeli 156 grafov v dva razreda, ki sta med seboj primerljiva po številu grafov, ki jih vsebujeta. V našem primeru smo dobili 98 grafov v prvem in 58 grafov v drugem razredu. Druge omenjene metode nam dajo kot rezultat dva razreda, od katerih eden vsebuje veliko večino grafov, drugi pa peščico le-teh. Slika 6.4 prikazuje število grafov, ki so razvrščeni v enega od obeh razredov, po mesecih. Iz slike lahko razberemo, da *razred1* predstavlja grafe, ki pripadajo predvsem hladnejšim mesecim, *razred2* pa je v glavnem sestavljena iz grafov iz toplejših obdobij. Hierarhično razvrščanje v dve glavni skupini tako razdeli gibanje vodnih mas v Sredozemskem morju na zimsko in poletno situacijo, kar potrjuje sezonskost gibanja vodnih mas v Sredozemskem morju. Podoben poskus na bolj grobem grafu s 235 vozlišči je tudi pokazal podobno razdelitev, kar potrjuje robustnost te metode.

6.4 Obrat površinske cirkulacije

V literaturi [61, 62] najdemo poglobljene analize eksperimentalnih opazovanj, satelitskih podatkov, poti plovcev in temperature ter slanosti v različnih globinah Jonskega morja. Te analize so privedle do ugotovitve, da pride do obrata pri kroženju vodnih mas v Jonskem morju od anticiklonske smeri vrtenja (v smeri urinega kazalca) do ciklonske (proti smeri urinega kazalca) in obratno približno vsakih 10 let. Avtorji menijo, da je prišlo do obrata iz anticiklonskega v ciklonski režim v letu 1997 in obratno v letih 2006–2007.

To hipotezo smo raziskali tudi s pomočjo večnivojskih usmerjenih grafov [28], ki

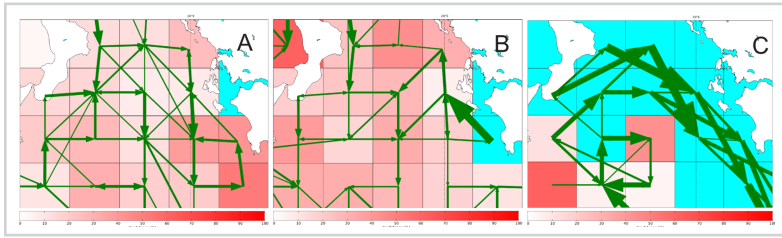


Slika 6.4

Število grafov, ki so razvrščeni v vsakega od dveh razredov po mesecih. Stolpci so obarvani rdeče (poletno stanje) in modro (zimsko stanje). Številke na vrhu stolpcev označujejo število grafov v vsakem mesecu.

smo jih dobili iz podatkov o poteh plovcev [53]. Z namenom, da pokažemo morebitni obrat v Jonskem morju, smo uporabili razpoložljive podatke o poteh plovcev (ang. *drifters*) v globini od 0 do 15 m v obdobju 1994–2007. V skladu z napotki iz omenjene oceanografske literature smo razdelili obdobje 1994–2007 na tri krajša obdobja: do januarja 1994 do junija 1997, od julija 1997 do decembra 2006 in leto 2007 v celoti. V naštetih obdobjih so po vrsti prevladovali anticiklonski, ciklonski in ponovno anticiklonski režim v Jonskem morju. Za vsako od teh obdobij smo skonstruirali večnivojski usmerjeni graf z 246 vozlišči (enakomerna pravokotna razdelitev domene), ki so približno dvakrat večja od vozlišč, opisanih v razdelku 5.1. S pomočjo omenjene prostorske granulacije dosežemo dovolj nazoren prikaz režima cirkulacije v Jonskem morju. Slika 6.5 prikazuje dobljene večnivojske usmerjene grafe za vsako od teh obdobij. Z namenom, da naredimo sliko verjetnosti prehodov plovcev bolj razločno, prikazujemo na sliki za prvo in tretje obdobje samo povezave med sosednjimi območji, ki imajo minimalno zaupanje 10 % in minimalno podporo 0,01 %, za drugo obdobje pa tiste, ki imajo minimalno podporo 0,1 %.

Na sliki 6.5 A lahko opazimo izrazit anticiklonski režim pred sredino leta 1997, manj izrazit ciklonski režim od sredine 1997 do 2006 6.5 B in očiten anticiklonski režim v letu 2007 6.5 C, kar potrjuje študije o obratih cirkulacije v Jonskem morju v zadnjih



Slika 6.5

Večnivojski usmerjeni grafi, konstruirani iz podatkov o poteh plovcev v Jonskem morju: A) od januarja 1994 do junija 1997, B) od julija 1997 do konca 2006, in C) leto 2007

dveh desetletjih [61, 62].

6.5 Modeliranje širjenja bioloških vrst

Metodologijo, ki uporablja večnivojske usmerjene grafe, smo uporabili tudi za modeliranje širjenja bioloških vrst v dani domeni [29, 63]. V tem primeru smo proučevali širjenje meduz vrste mesečinka (znanstveno ime *Pelagia noctiluca*, Scyphozoa), ki je zelo pogosta v Sredozemskem morju. Številne študije obravnavajo transport in abundanco (ang. *abundance*, slovensko *številčnost*) omenjene vrste (glej npr. [64, 65]). Opaženi so ponavljajoči še izbruhi velikega števila osebkov te vrste v Sredozemskem morju in vzhodnem Atlantiku [64], kar ima precejšen vpliv na turizem in obalno industrijo. Za to vrsto meduze je značilno, da v vodi prosto lebdi ali plava in v svojem razvoju nima t. i. polipne faze, v kateri bi bila pritrjena na dno morja kot pri drugih vrstah meduz. Naše predhodne študije [66] so pokazale zelo majhno genetsko diferenciacijo *P. noctiluca* v velikem prostorskem merilu (Sredozemsko morje in vzhodni Atlantik), kar kaže, da je ta vrsta demografsko zelo odprta in ima velike možnosti širjenja. Vse to nam omogoča, da lahko modeliramo njeno širjenje s pomočjo večnivojskih usmerjenih grafov, ki jih dobimo na osnovi hitrostnega polja numeričnih modelov ali pa izmerjenih poti plovcev. Matrike sosednosti omenjenih grafov lahko uporabimo kot prehodne matrike za modeliranje abundance biološke vrste s pomočjo Markovskih procesov. Abundanca je mera za številčnost vrste na določenem območju in je ponavadi podana kot število osebkov na prostorninsko ali površinsko enoto. Abundanca vrste na morskih območjih (vozliščih grafa) predstavlja stanje Markovskega procesa. Izračunamo jo v zaporednih časovnih korakih s pomočjo enačbe:

$$a(t + \Delta t) = M^T(m)a(t) \tag{6.1}$$

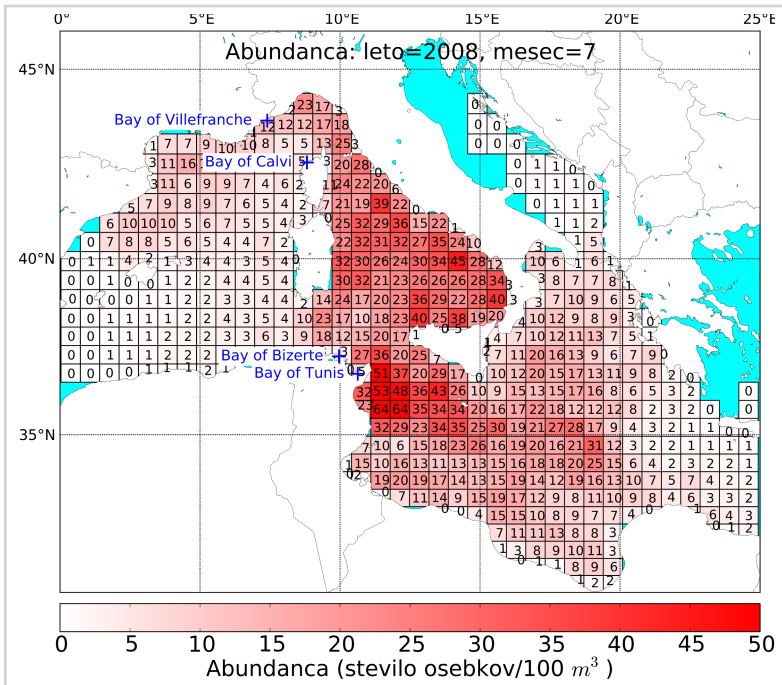
kjer sta $a(t)$ and $a(t + \Delta t)$ vektorja, ki po vrsti vsebujeta abundanco v tekočem in naslednjem časovnem koraku; $M^T(m)$ je transponirana matrika sosednosti grafa, ki pripada mesecu m, t pa je trenutni časovni interval, ki lahko zavzame vrednosti od 1 do t_{max}. Slednjega izračunamo na naslednji način:

$$t_{max} = 30 m_{max}/\Delta t \tag{6.2}$$

kjer je m_{max} trajanje simulacije v mesecih, Δt pa časovni korak, merjen v dnevih. Pri tem privzamemo konstantno dolžino 30 dni za vsak mesec. Matrike sosednosti dobimo iz mesečnih večnivojskih usmerjenih grafov, elementi teh matrik pa predstavljajo verjetnosti premikov virtualnih delcev iz enega morskega območja v drugo zaradi gibanja vodnih mas (advekcija in turbulentna difuzija). Bioloških lastnosti vrste *Pelagia noctiluca*, kot so npr. rast, razmnoževanje, umiranje itd., v tem modelu ne upoštevamo.

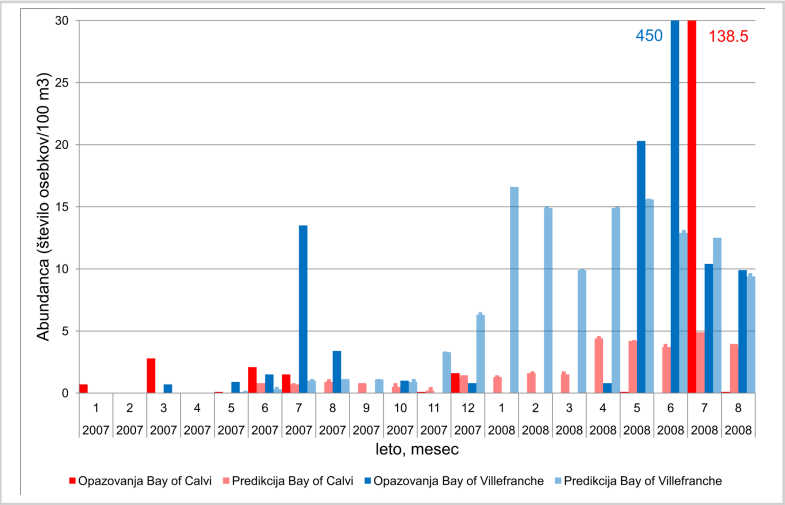
Pri simulaciji smo uporabili podatke o pojavih izredno povečanega števila meduz na določenih mestih v zahodnem Sredozemlju v letih 2007 in 2008 [64]. Iz omenjenega vira smo pridobili povprečno število meduz na prostorninsko enoto za vsak mesec v obdobju 2007–2008 za štiri merilna mesta: Bay of Tunis, Bay of Bizerte, Bay of Calvi in Bay of Villefranche (glej sliko 6.6 za lokacije teh mest). S simulacijo smo pričeli v januarju 2007 z začetnimi vrednostmi številčnosti meduz v vozliščih grafa, ki zajemata območji Bay of Tunis in Bay of Bizerte (izvorna območja). V vseh ostalih vozliščih je bila vrednost začetne abundance enaka 0. Pri simulaciji smo uporabili mesečne večnivojske usmerjene grafe, ki smo jih skonstruirali na podlagi numeričnega modela MFS in so opisani v razdelku 5.1. Pri vsakem koraku simulacije smo posodabljali abundance z opazovanji v Bay of Tunis in Bay of Bizerte in izračunavali abundance v celotni domeni. Porazdelitev abundanc po posameznih območjih po 18 mesecih simulacije nam prikazuje slika 6.6.

Pod drobnogled smo vzeli izračunane abundance v Bay of Calvi in Bay of Villefranche (v našem primeru ciljna območja) in jih primerjali z izmerjenimi [64]. Slika 6.7 prikazuje primerjavo med izmerjenimi in izračunanimi abundancami na ciljnih območjih po 18 mesecih simulacije (julij 2008). Izračunane abundance nam povedo, da *Pelagia noctiluca* iz izvornih območij doseže ciljna območja nekaj mesecev pred opazovanimi izbruhi v juniju in juliju 2008. Vrednosti izračunanih abundanc v maju, juliju in avgustu 2008 v Bay of Villefranche so celo primerljive z izmerjenimi vrednostmi. Vendar pa so izbruhi meduz odvisni od bioloških dejavnikov in ugodnih okoliščin, ki nastanejo na določenih območjih tj. višje temperature morja, odsotnost plenilcev in



Slika 6.6

Porazdelitev abundance *P. noctiluca* v morskih območjih po 18 mesecih simulacije z začetkov v januarju 2007 in zaključkom v juliju 2008, kjer smo številčnost meduz v Bay of Tunis in Bay of Bizerte sproti posodabljali z mesečnimi povprečnimi meritvami v obdobju 2007–2008. Abundanca meduz je prikazana s števili in je sorazmerna z nasičenostjo rdeče barve.



Slika 6.7

Časovne vrste abundance v ciljnih območjih Bay of Calvi (rdeči stolpci) in Bay of Villefranche (modri stolpci). Stolpci temnejše barve predstavljajo opazovanja, svetlejši pa rezultate simulacije. Zelo visoke izmerjene abundance v juniju (Bay of Villefranche) in juliju (Bay of Calvi) 2008 so še dodatno označene z njihovim številom.

konkurentov, evtrofikacija itd., zato je za simulacijo takšnih izbruhov potrebno opisani model dopolniti z biološkimi procesi. Metodologija, ki smo jo opisali, je zelo obetavna za modeliranje porazdelitev abundance bioloških vrst (v našem primeru *P. noctiluca*) kot posledico gibanja vodnih mas na določenem območju, kljub temu pa ne more identificirati masivnih izbruhov populacij meduz. Le-ti so lokalne narave in se ne nanašajo na morska območja (ki jih predstavljajo vozlišča v večnivojskih usmerjenih grafih) kot celote. Poleg tega se metodologija ne ukvarja z biološkimi procesi npr. razmnoževanje in umiranje meduz ter okoliščinami, kot je temperatura morja itd. V model je potrebno vključiti še fiziološke lastnosti *P. noctiluca* in ekološke parametre, kar prestavlja izziv za naše prihodnje delo.

Zaključki

7

7.1 Zaključki

V disertaciji je predstavljen nov pristop, ki omogoča prostorsko-časovno rudarjenje velikega števila trajektorij, ki se pojavljajo na številnih področjih, kjer imamo opraviti s sledenjem objektov v prostoru in času. Omenjeni pristop uporablja prostorsko-časovna povezovalna pravila in večnivojske usmerjene grafe. Razvili smo metode in algoritme za analizo in rudarjenje večnivojskih usmerjenih grafov. V disertaciji smo se omejili na področje oceanografije tj. iskanje značilnih struktur (poti in ciklov), ki nastanejo pri gibanju vodnih mas v morju. Kljub temu je metodologija večnivojskih usmerjenih grafov odprta za razvoj širokega nabora novih in adaptacijo obstoječih algoritmov za rudarjenje grafov (glej [33, 34]). Obstoječe algoritme za rudarjenje splošnih grafov lahko priredimo za naš poseben tip grafa (tj. *večnivojski usmerjen graf za analizo prostorskih podatkov*) in jim hkrati lahko izboljšamo tudi časovno zahtevnost. Algoritem za iskanje enostavnih ciklov v posameznih (mesečnih) večnivojskih usmerjenih grafih (glej razdelek 4.1) ima časovno zahtevnost $O(m+n)$ (n je število vozlišč, m pa število povezav). Omenjeni algoritem ne zadošča za iskanje daljših (več mesecev) in kompleksnejših poti in ciklov, zato smo v ta namen razvili vrsto algoritmov, ki jih opisujemo v razdelku 4.2. Rezultat so bili dinamične mehke poti in cikli, ki smo jih dobili z združevanjem posameznih poti in ciklov. Algoritmi, ki smo jih uporabili za ta namen, imajo večinoma kvadratno časovno zahtevnost, za poenostavitev kompleksnih ciklov (algoritem 4) pa je potrebno nekaj rekurzivnih klicev omenjenega algoritma. Število rekurzivnih klicev pa je odvisno od kompleksnosti vgnezdenih ciklov.

Omenjeni pristop nadgrajuje obstoječe metode Lagrangeove analize v oceanografiji [3] in jim dodaja novo vrednost. S tem pomaga uporabniku, da odkriva in vizualizira skrite vzorce in zakonitosti v rezultatih oceanografskih opazovanj in numeričnih modelov. V ta namen uporabimo prostorsko-časovna povezovalna pravila in večnivojske usmerjene grafe, ki so se izkazali za zelo obetavno izbiro za tovrstno nalogo. Algoritmi, ki smo jih razvili, so se izkazali za zelo učinkovite pri odkrivanju sezonskih vzorcev prehodov vodnih mas med morskimi območji. Ta pristop nam je omogočil potrditev spoznanja, da ima verjetnost prehoda vodnih mas med območji v Sredozemskem morju periodo 12 mesecev. S pomočjo hierarhičnega razvrščanja večnivojskih usmerjenih grafov smo ločili sliko gibanja vodnih mas v celotni domeni na zimsko in poletno situacijo. Poleg tega je predlagana metodologija primerna za vizualizacijo dolgoročnih prehodnih pojavov, kot je obrat cirkulacije v Jonskem morju. Ta pojav je opisan v

oceanografski literaturi [53, 61, 62]. Večnivojske usmerjene grafe smo uporabili na Lagrangeovih trajektorijah površinskih plovcev in tako potrdili obrat cirkulacije v Jonskem morju v letih 1997 in 2006–2007. Nenazadnje smo večnivojske usmerjene grafe uporabili tudi za simulacijo širjenja bioloških vrst v morju s tem, da smo omenjene grafe povezali z markovskimi procesi. S pomočjo tega postopka smo potrdili hipotezo, da je širjenje populacije meduz vrste *Pelagia noctiluca* odvisno od gibanja vodnih mas v morju zaradi advekcije in turbulentne difuzije. Omenjeno metodo je mogoče izboljšati s tem, da večnivojske usmerjene grafe nadgradimo z atributi in metodami, ki so potrebne za biološko in ekološko modeliranje.

Metodologija, opisana v disertaciji, je splošno uporabna na poljubnih trajektorijah za sledenje npr. vozilom, plovilom, živalim itd. Ko imamo na voljo dovolj veliko število trajektorij, ki opisujejo pozicijo določenih objektov v odvisnosti od časa, potem je postopek gradnje prostorsko-povezovalnih pravil in večnivojskih usmerjenih grafov identičen tistemu, ki smo ga opisali v 3. poglavju. Pri tem moramo prostorsko razdelitev domene primerno prilagoditi problemu, ki ga rešujemo, in določiti ustrezen časovni interval Δt. Pri sledenju vozil lahko npr. namesto prostorske razdelitve domene na mnogokotnike, ki jih potem pripišemo vozliščem večnivojskega usmerjenega grafa, uporabimo geografske enote tj. mesta, okrožja itd. in le-te pripišemo vozliščem grafa. Na nastalih večnivojskih usmerjenih grafih je uporaba algoritmov, opisanih v 4. poglavju, podobna kot v primerih iz oceanografije. Ne nazadnje opisana metodologija in algoritmi omogočajo razvoj številnih aplikacij, ki bi bile uporabne na drugih področjih.

7.2 Razprava in nadaljnje delo

V poglavju 3 smo opisali metodologijo konstrukcije večnivojskih usmerjenih grafov in jo ovrednotili s pomočjo rezultatov numeričnega modela MFS. Prikazali smo različne načine prostorske razdelitve domene. Pri tem se takoj pojavi vprašanje, na kakšen način lahko optimalno razdelimo domeno, da bi algoritmi na dobljenih večnivojskih usmerjenih grafih dali najbolj relevantne rezultate. Enakomerna razdelitev domene, kakršno smo večinoma uporabljali v disertaciji, ni najbolj primerna za iskanje značilnih vzorcev pri gibanju vodnih mas v kompleksni domeni, kot je Sredozemsko morje. Vzrok za to je heterogenost hitrostnega polja tj. hitrosti gibanja vodnih mas so zelo odvisne od lokacije oziroma morskih območij. Kjer so hitrosti v povprečju nižje, bi bilo potrebno definirati večja morska območja in obratno. Kljub temu smo z enakomerno razdeli-

tvijo domene pri izbrani velikosti območij dobili dobre rezultate pri iskanju struktur v srednjem merilu (ang. *mesoscale variability*). V nadaljnjem delu bi bilo potrebno razviti algoritem, ki bi ob dani povprečni velikosti območij poiskal optimalno delitev na neenaka območja in poiskal ustrezen časovni interval za izračun verjetnosti prehoda med območji.

V razdelku 5.3 smo se pri določanju metode hierarhičnega razvrščanja dinamičnih poti in ciklov in izbire ustrezne višine rezanja oprli na oceanografsko ekspertno predznanje. Postopek za samodejno določanje parametrov hierarhičnega razvrščanja tako še vedno ostaja odprt problem. Podobne pomanjkljivosti smo omenili že pri opisih nekaterih uveljavljenih metod prostorsko-časovnega podatkovnega rudarjenja npr. DBSCAN, SOM itd. V našem primeru smo si pomagali s postopkom, s pomočjo katerega smo določili ožji interval višine rezanja dendrograma okoli ekstrema, kjer je bila razdalja med dobljenimi dinamični mehki cikli in opazovanimi cikli najmanjša. Podobno so si npr. avtorji algoritma DBSCAN pomagali z grafom, s pomočjo katerega so določali parametre za omenjeni algoritem. Dinamične mehke poti in cikle smo v bistvu dobili z združevanjem posameznih dinamičnih poti in ciklov. Tudi iz drugih področij oziroma domen je znano, da združevanje prostorskih objektov ni trivialno in pogosto zahteva domensko predznanje v obliki hierarhičnih geografskih konceptov [glej 1, str. 14]. Izboljšanje metode iskanja dinamičnih mehkih poti in ciklov je torej velik izziv za naše nadaljnje delo.

Algoritme, opisane v disertaciji, smo implementirali kot prototip na prenosni delovni postaji HP EliteBook 8540w z vgrajenim dvojedrnim procesorjem Intel® Core™ i7 2.67 GHz in 8 GB pomnilnika ter nameščenim 64-bitnim operacijskim sistemom Windows 7 Professional. Programski orodji, ki smo ju v glavnem uporabljali, sta R verzija 3.0.1 (64-bit) [55] in Python 2.7 [56]. Z uporabo navedenih orodij na dani strojni opremi so se opisani algoritmi izvajali tudi po več ur. Pri tem smo bili tudi omejeni s pomnilnikom, zaradi tega smo lahko hkrati izvajali hierarhično razvrščanje največ okoli 10.000 dinamičnih poti (glej razdelek 5.3.1). Večino opravil, opisanih v disertaciji, je mogoče izvajati vzporedno na poljubnem številu procesorjev. Implementacija vzporednega izvajanja omenjenih algoritmov je tako ena od prioritetnih nalog, preden bi se opisana metodologija lahko učinkovito uporabljala. Poleg tega bi bilo potrebno postaviti učinkovito prostorsko podatkovno bazo in razviti uporabniški vmesnik, ki bi omogočil oceanografskim ekspertom, ki niso vešči računalniškega programiranja, učinkovito uporabo te metodologije.

Izboljšano implementacijo opisane metodologije in algoritmov bi bilo potrebno testirati še na drugih domenah npr. na drugih regionalnih oceanografskih numeričnih modelih in seveda tudi na globalnem oceanografskem modelu. Pri slednjem se množica rezultatov v zadnjih letih povečuje, čeprav njihova zanesljivost trenutno ne doseže tiste, ki jo ima model za Sredozemlje (MFS). Z naraščajočim številom podatkov, ki opisujejo trajektorije na drugih področjih, pa imamo še več možnosti za testiranje opisane metodologije.

LITERATURA

[1] H. J Miller and J. Han. *Geographic Data Mining and Knowledge Discovery*. Data Mining and Knowledge Discovery Series. Chapman-Hall/CRC, second edition edition, 2009.

[2] J. Mennis and D. Guo. Spatial data mining and geographic knowledge discovery - An introduction. *Computers Environment and Urban Systems*, 33(6): 403–408, 2009.

[3] A. Griffa, A. D. Kirwan Jr, A. J. Mariano, T. Özgökmen, and H. T. Rossby, editors. *Lagrangian Analysis and Prediction of Coastal and Ocean Dynamics*. Cambridge University Press, 2007.

[4] A. Merz. Hydrographische Untersuchungen im Golf von Triest, 1911.

[5] MyOcean. Ocean monitoring and forecasting, 2013. URL www.myocean.eu. (zadnji dostop december 2013).

[6] SeaDataNet, 2013. URL www.seadatanet.org. (zadnji dostop december 2013).

[7] S. J. Camargo, A. W. Robertson, S. J. Gaffney, and P. Smyth. Cluster Analysis of Western North Pacific Tropical Cyclone Tracks. Technical Report 05-03, The International Research Institute for Climate and Society, Columbia University, New York, December 2005.

[8] R. Žabkar, J. Rakovec, and S. Gaberšek. A trajectory analysis of summertime ozone pollution in Slovenia. *Geofizika*, 25(2):24 pp., 2008.

[9] M Tonani, N Pinardi, S Dobricic, I Pujol, and C Fratianni. A high-resolution free-surface model of the Mediterranean Sea. *Ocean Science*, 4(1):1–14, 2008.

[10] GNOO, INGV. Mediterranean ocean Forecasting System, 2009. URL http://gnoo.bo.ingv.it/mfs/myocean/. (zadnji dostop december 2013).

[11] G. De'ath and K. E. Fabricius. Classification and regression trees: A powerful yet simple technique for ecological data analysis. *Ecology*, 81(11):3178–3192, 2000.

[12] D. Birant and A. Kut. An algorithm to discover spatial temporal distributions of physical seawater characteristics and a case study in Turkish seas. *Journal of Marine Science and Technology*, 11:183–192, 2006.

[13] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, OR*, pages 226–231, August 1996.

[14] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In Jorge B Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, pages 487–499. Morgan Kaufmann, 1994. ISBN 1-55860-153-8.

[15] I. Kononenko and M. Kukar. *Machine Learning and Data Mining*. Horwood Publishing, 2007. ISBN 97819042752l3.

[16] Y. P. Huang, L. J. Kao, and F. E. Sandnes. Efficient mining of salinity and temperature association rules from ARGO data. *Expert Syst Appl*, 35(1-2):59–68, 2008. ISSN 0957-4174.

[17] A. K. H. Tung, H. Lu, J. Han, and L. Feng. Efficient mining of intertransaction association rules. *IEEE Transactions on Knowledge and Data Engineering*, 15: 43–56, January 2003. ISSN 1041-4347.

[18] Y. Liu and R. H. Weisberg. *Self Organizing Maps - Applications and Novel Algorithm Design*, chapter A Review of Self-Organizing Map Applications in Meteorology and Oceanography, pages 253–271. InTech, January 2010.

[19] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, New York, 1995, 1997, 2001., third extended edition edition, 2001.

[20] Y. Liu, R. H. Weisberg, and C. N. K. Mooers. Performance evaluation of the self-organizing map for feature extraction. *Journal of Geophysical Research: Oceans*, 111 (C5), 2006. ISSN 2156-2202.

[21] Y. Liu and R. H. Weisberg. Patterns of ocean current variability on the West Florida Shelf using the self-organizing map. *Journal of Geophysical Research*, 110: C06003, 12 pp., 2005.

[22] Y. Liu and R. H. Weisberg. Ocean currents and sea surface heights estimated across the West Florida Shelf. *Journal of Physical Oceanography*, 37:1697–1713, 2007.

[23] C. Solidoro, V. Bandelj, P. Barbieri, G. Cossarini, and S. F. Umani. Understanding dynamic of biogeochemical properties in the northern Adriatic Sea by using selforganizing maps and k-means clustering. *Journal of Geophysical Research*, 112:C07S90, 13 pp., 2007.

[24] M. Telszewski, A. Chazottes, U. Schuster, A. J. Watson, C. Moulin, D. C. E. Bakker, M. González-Dávila, T. Johannessen, A. Körtzinger, H. Lüger, A. Olsen, A. Omar, X. A. Padin, A. F. Ríos, T. Steinhoff, M. Santana-Casiano, D. W. R. Wallace, and R. Wanninkhof. Estimating the monthly pCO_2 distribution in the North Atlantic using a self-organizing neural network. *Biogeosciences*, 6(8):1405–1421, 2009.

[25] H. Mihanovic, S. Cosoli, I. Vilibic, D. Ivankovic, V. Dadic, and M. Gacic. Surface current patterns in the northern Adriatic extracted from high-frequency radar data using self-organizing map analysis. *Journal of Geophysical Research*, 116, 2011.

[26] A. Appice and P. Buono. Analyzing Multi-level Spatial Association Rules Through a Graph-Based Visualization. In Moonis Ali and Floriana Esposito, editors, *Innovations in Applied Artificial Intelligence*, volume 3533 of *Lecture Notes in Computer Science*, pages 67–92. Springer Berlin / Heidelberg, 2005. ISBN 978-3-540-26551-1.

[27] K. Koperski and J. Han. Discovery of spatial association rules in geographic information databases. In M. J. Egenhofer and J. R. Herring, editors, *Advances in Spatial Databases*, volume 951 of *Lecture Notes in Computer Science*, pages 47–66. Springer Berlin Heidelberg, 1995. ISBN 978-3-540-60159-3.

[28] B. Petelin, I. Kononenko, V. Malačič, and M. Kukar. Multi-level association rules and directed graphs for spatial data analysis. *Expert Systems with Applications*, 40:4957–4970, September 2013. ISSN 0957-4174.

[29] B. Petelin, V. Malačič, A. Malej, M. Kukar, and I. Kononenko. Multi-level Association Rules and Directed Graphs for the Lagrangian Analysis of the Mediterranean Ocean Forecasting System (MFS). In A. Abbasi and N. Giesen, editors, *EGU General Assembly Conference Abstracts*, volume 14 of *EGU General Assembly Conference Abstracts*, page 4877, April 2012.

[30] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. Point location. In *Computational Geometry*, chapter 6, pages 121–146. Springer-Verlag, 2nd revised edition, 2000. ISBN 3-540-65620-0.

[31] D. Dobkin and R. J. Lipton. Multidimensional searching problems. *SIAM J Sci Comput*, 5(2):181–186, 1976.

[32] H. Edelsbrunner, L. J. Guibas, and J. Stolfi. Optimal point location in a monotone subdivision. *SIAM J Sci Comput*, 15(2):317–340, 1986.

[33] D. J. Cook and L. B. Holder. *Mining Graph Data*. John Wiley & Sons, 2006. ISBN 0471731900.

[34] N. F. Samatova. *Practical Graph Mining With R*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Chapman and Hall/CRC, July 15 2013.

[35] B. Petelin, I. Kononenko, V. Malačič, and M. Kukar. Dynamic fuzzy paths and cycles in multi-level directed graphs. Technical report, University of Ljubljana, Faculty of Computer and Information Science, 2013. (poslano v objavo).

[36] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):12, September 2007.

[37] L. Šubelj and M. Bajec. Unfolding communities in large complex networks: Combining defensive and offensive label propagation for core extractione. *Physical Review E*, 83(3):036103, 2011.

[38] K. Lakshmi and T. Meyyappan. Frequent Subgraph Mining Algorithms -A Survey And Framework For Classification. In *The First International Conference on Information Technology Convergence and Services (ITCS 2012)*, 2012.

[39] T. Washio and H. Motoda. State of the art of graph-based data mining. *SIGKDD Explor. Newsl.*, 5(1): 59–68, 2003. ISSN 1931-0145.

[40] B. Braden. The Surveyor's Area Formula. *The College Mathematics Journal*, 17(4):326–337, 1986.

[41] C. J. Lin and F. J. Och. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[42] M. Poženel, V. Mahnič, and M. Kukar. Separation of interleaved web sessions with heuristic search. In *2010 IEEE 10th International Conference on Data Mining (ICDM)*, pages 411–420, 2010.

[43] J. H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[44] A. Tversky. Features of similarity. *Psychological Reviews*, 84(4):327–352, 1977.

[45] E. Oks and M. Sharir. Minkowski sums of monotone and general simple polygons. *Discrete & Computational Geometry*, 35(2):223–240, 2006. ISSN 0179-5376.

[46] K. Teknomo. Recursive average and variance, 2006. URL http://people.revoledu.com/kardi/tutorial/RecursiveStatistic/index.html. (zadnji dostop december 2013).

[47] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, 2000.

[48] B. Blanke and N. Grima. Ariane, 2010. URL http://stockage.univ-brest.fr/~grima/Ariane/. (zadnji dostop december 2013).

[49] N. Hamad, C. Millot, and I. Taupier-Letage. The surface circulation in the eastern basin of the Mediterranean Sea. *Sci. Mar.*, 70(3):457–503, 2006.

[50] C. Millot. Circulation in the Western Mediterranean Sea. *J Mar Syst*, 20(1–4):423–442, 1999. ISSN 0924-7963.

[51] A. R. Robinson, W. G. Leslie, A. Theocharis, and A. Lascaratos. Mediterranean Sea Circulation, 2001.

[52] A. Artegiani, D. Bregant, E. Paschini, N. Pinardi, F. Raicich, and A. Russo. The Adriatic Sea general circulation. part ii. baroclinic circulation structure. *J. Phys. Oceanogr.*, 27:1515–1532, 1997.

[53] P. M. Poulain, M. Menna, and E. Mauri. Surface Geostrophic Circulation of the Mediterranean Sea Derived from Drifter and Satellite Altimeter Data. *J Phys Oceanogr*, 42(6):973–990, February 2012. ISSN 0022-3670.

[54] P. Malanotte-Rizzoli, B. Manca, M. Ribera D'Alcala, A. Theocharis, A. Bergamasco, D. Bregant, G. Budillon, G. Civitarese, D. Georgopoulos, A. Michelato, E. Sansone, P. Scarazzato, and E. Souvermezoglou. A synthesis of the Ionian Sea hydrography, circulation and water mass pathways during POEM-Phase I. *Prog. Oceanogr.*, 39:153–204, 1997.

[55] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL http://www.r-project.org/. (zadnji dostop december 2013).

[56] Guido van Rossum et al. *The Python Language Reference*. Python Software Foundation, July 2010. URL http://docs.python.org/release/2.7/reference/index.html. (zadnji dostop december 2013).

[57] V. Roussenov, E. Stanev, V. Artale, and N. Pinardi. A seasonal model of the Mediterranean Sea general circulation. *J. Geophys. Res.*, 100(C7):13515–13538, 1995. ISSN 2156-2202.

[58] I. M. Ovchinnikov. Circulation in the surface and intermediate layers of the Mediterranean. *Oceanology*, 6:48–59, 1966.

[59] C. Millot. Mesoscale and seasonal variabilities of the circulation in the western Mediterranean. *Dynamics of Atmospheres and Oceans*, 15(3–5):179–214, 1991. ISSN 0377-0265.

[60] E. Tziperman and P. Malanotte-Rizzoli. The climatological seasonal circulation of the Mediterranean Sea. *Journal of Marine Research*, 49:411–434, 1991.

[61] G. L. E. Borzelli, M. Gačić, V. Cardin, and G. Civitarese. Eastern Mediterranean Transient and reversal of the Ionian Sea circulation. *Geophys Res Lett*, 36(15): L15108, August 2009. ISSN 0094-8276.

[62] M. Gačić, G. L. E. Borzelli, G. Civitarese, V. Cardin, and S. Yari. Can internal processes sustain reversals of the ocean upper circulation? The Ionian Sea example. *Geophys Res Lett*, 37(9):L09608, May 2010. ISSN 0094-8276.

[63] B. Petelin, I. Kononenko, M. Kukar, V. Malačič, and A. Malej. Spatial-temporal directed graphs for modeling the dispersion of plankton in the Mediterranean Sea. In *CIESM Congress Proceedings n°40*, 40th CIESM Congress, Marseille, France, 28 Oct. - 01 Nov. 2013.

[64] P. Licandro, D. V. P. Conway, M. N. Daly Yahia, M. L. Fernandez de Puelles, S. Gasparini, J. H. Hecq, P. Tranter, and R. R. Kirby. A blooming jellyfish in the northeast Atlantic and Mediterranean. *Biology Letters*, 6(5):688–691, 2010.

[65] L. Berline, B. Zakardjian, A. Molcard, Y. Ourmieres, and K. Guihou. Modeling jellyfish *Pelagia noctiluca* transport and stranding in the Ligurian Sea. *Marine Pollution Bulletin*, 70(1–2):90 – 99, 2013. ISSN 0025-326X.

[66] K. Stopar, A. Ramšak, P. Trontelj, and A. Malej. Lack of genetic structure in the jellyfish *Pelagia noctiluca* (Cnidaria: Scyphozoa: Semaeostomeae) across European seas. *Molecular Phylogenetics and Evolution*, 57(1): 417–428, 2010. ISSN 1055-7903.