UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET
SMER: RAČUNARSTVO I INFORMATIKA

MONTE KARLO METODE I PRIMENE U BIOINFORMATICI

-MASTER RAD-

Mentor:
Prof. dr Gordana Pavlović-Lažetić

Student:
Marija Đurić

Beograd, 2017.

SADRŽAJ

1. UVOD

Sam princip rada Monte Karlo simulacije nastaje 1777. godine kada se Žorž Luis Lekler poznatiji kao Kompte de Bufon (Georges Louis Leclerc, Compte de Buffon 1707-1788) zapitao kolika je verovatnoća da drvce dužine l bačeno na rešetku razmaka d ($d > l$) padne na jednu od linija rešetke. Došao je do sledećeg rezultata [1]:

$$p = \frac{2l}{\pi d}$$

Koristeći ovo rešenje, matematičar Laplas (Pierre-Simon Laplace,1749-1827) je došao do jedinstvenog načina određivanja broja π. Neka je Bufonov eksperiment izveden bacanjem drvceta n puta i neka N označava koliko puta je drvce palo na liniju, tada je verovatnoća p jednaka:

$$p = \lim_{n\to\infty}(\frac{N}{n})$$

Iz ovih jednakosti se lako određuje broj π:

$$\pi = \frac{2l/_d}{\lim_{n\to\infty}(\frac{N}{n})}$$

Formalno, Monte Karlo simulaciju razvili su tokom 1940. godine dvojica američkih naučnika Stanislav Ulam (Stanislaw Ulam, 1909-1984) i Džon fon Nojman (John von Neumann, 1903-1957) u Los Alamos Nacionalnoj laboratoriji dok su sarađivali na projektu Menhetn (*eng*. *Manhattan*). Simulaciju su koristili da bi odredili slučajnu difuziju neutrona i nazvali su je Monte Karlo, prema gradu u Monaku i njegovim mnogobrojnim kazinima. Projekat Menheten je bio naziv za tajni program vlada SAD, Kanade i Ujedinjenog kraljevstva čiji je cilj bio razvoj atomske bombe [2].

Danas se Monte Karlo metoda koristi u različitim oblastima, kao što su: biologija, genetika, statistika, itd. Metode Monte Karlo se primenjuju posebno u slučajevima kada bi eksperimenti sa sistemom koji proučavamo bili dugotrajni ili dovodili do oštećenja sistema. Problemi koji se sreću u raznim oblastima se mogu "prevesti" ili svesti na matematičke probleme: rešavanje sistema linearnih jednačina ili nejednačina, računanje integrala (jednostrukih ili višestrukih), rešavanje diferencijalnih jednačina, rešavanje parcijalnih diferencijalnih jednačina, itd.

U ovom radu će pored objašnjenja Monte Karlo metoda biti predstavljena i opisana njihova primena na problem uvijanja proteina kao jedan od važnih problema proteomike i bioinformatike. Prvo poglavlje ovog rada je posvećeno predstavljanju najčešćih primena Monte Kralo metoda kao

i nekih njenih osnovnih tehnika, kao što su uzorkovanje odbacivanje, uzorkovanje po značajnosti i Markovljevi lanci. U trećem i četvrtom poglavlju su opisana dva najvažnija Monte Karlo algoritma, Metropolis-Hestingov algoritam i Gibsov uzorkivač, kao i neke njihove varijacije. Poslednje poglavlje sadrži objašnjenje Monte Karlo algoritma razmene kopija i rezultate primene ovog metoda na problem uvijanja proteina.

2. OSNOVE MONTE KARLO METODE

2.1 MONTE KARLO METODA I NAJČEŠĆE PRIMENE

Monte Karlo metodu predstavlja grupa algoritama čija je suština ponavljanje slučajnih pokušaja u cilju dobijanja numeričkih rezultata. Često se koristi za rešavanje fizičkih i matematičkih problema, posebno u slučajevima kada je nemoguće koristiti druge matematičke metode. Monte Karlo metoda se najčešće koristi za rešavanje problema optimizacije, numeričke integracije i za generisanje uzoraka kod raspodele verovatnoće.

Prilikom upotrebe statističkih metoda često se nailazi na problem izračunavanja integrala oblika:

$$I = \int_a^b h(x)g(x)\,dx.$$

Ovaj problem se može rešiti upotrebom metoda numeričke integracije, ali samo ukoliko se radi o prostoru manjih dimenzija. Za višedimenzione prostore, rešenje predstavlja upoteba Monte Karlo metode kojom se problem izračunavanja integrala svodi na proces uzorkovanja iz određene raspodele verovatnoće i izračunavanje srednje vrednosti dobijenih uzoraka. Neka funkcija $g(x)$ zadovoljava sledeća dva uslova:

$$g(x) \geq 0, x \in (a, b)$$

$$\int g(x) = C < \infty,$$

sada možemo definisati odgovarajuću raspodelu verovatnoće na intervalu (a, b) sa:

$$p(x) = \frac{g(x)}{C}.$$

Uvođenjem ove smene integral I ima sledeću vrednost:

$$I = C\int h(x)p(x)dx = C * E_{p(x)}[h(x)]$$

gde $E_{p(x)}[h(x)]$ predstavlja očekivanu vrednost funkcije $h\,(x)$ izračunatu korišćenjem uzoraka x_i iz pridružene raspodele verovatnoće, a koja se može aproksimirati na sledeći način:

$$E_{p(x)}[h(x)] \approx \frac{1}{N}\sum_i^N h(x_i)$$

gde su x_i=1,…,N nezavisno uzeti iz $p(x)$ [15]. Kako bi izračunavanje integrala i bilo moguće potrebno je odabrati odgovarajuću funkciju raspodele iz koje se jednostavno može vršiti

uzorkovanje, a u nastavku rada će biti opisane neke od osnovnih tehnika uzorkovanja, među kojima su i Markovljevi lanci.

Monte Karlo metode se najčešće koriste pri rešavanju integrala i problema optimizacije u višedimenzionim prostorima kada se oni ne mogu izračunati analitički ili za njih ne postoji efikasan numerički algoritam. Ovi problemi se mogu podeliti u dve grupe: problem izračunavanja očekivanja i problemi različitih varijacija. Objasnićemo ukratko svaki od njih [3]:

Problem izračunavanja očekivanja: Neka je X = $\{X_1, \dots X_n\}$ slučajna promenljiva čije komponente mogu biti diskretene ili neprekidne i neka je raspodela verovatnoće promenljivih data preko nenormalizovane funkcije gustine $f(x)$. Naš zadatak je da odredimo očekivanje funkcije $a(X_1, \dots X_n)$ u zavisnosti od raspodele verovatnoće. U slučaju diskretne raspodele očekivanje se računa na sledeći načih:

$$E(a) = \sum_{x_1} \dots \sum_{x_n} a(x_1, \dots x_n) P(x_1, \dots x_n)$$
$$= \frac{\sum_{x_1} \dots \sum_{x_n} a(x_1, \dots x_n) f(x_1, \dots x_n)}{\sum_{x_1} \dots \sum_{x_n} f(x_1, \dots x_n)} \qquad (1.1)$$

dok se kod neprekidne raspodele suma zamenjuje integralom što dodatno komplikuje izračunavanje. Takođe smo pretpostavili da se za svako x mogu izračunati vrednosti funkcija $f(x)$ i $a(x)$, ali njihovo izračunavanje u nekim slučajevima, naročito u višedimenzionim prostorima, zahteva dosta vremena, pa je naš cilj da minimalizujemo taj broj izračunavanja [3].

Problemi različitih varijacija: Neka problem koji želimo da rešimo ima funkciju gustine $f(x)$ koja značajno varira na svom domenu, sa najvećom verovatnoćom u veoma malim delovima domena čije lokacije nisu poznate *a priori*. Zbog ove karakteristike rešenje ovakvih problema zahteva pronalaženje metoda koje će odrediti delove domena sa najvećom verovatnoćom [3].

U mnogim slučajevima nije moguće dobiti nezavisne uzorke iz raspodele definisane pomoću funkcije $f(x)$, već se koriste zavisni uzorci najbliži zadatoj raspodeli dobijeni korišćenjem metode Markovljevih lanaca [3]. Razvoj ovih metoda i njihovo dalje istraživanje može doprineti pronalaženju boljih ocena očekivanja $E(a)$ sa većom verovatnoćom tačnosti.

2.2 RAZLIČITI PRISTUPI REŠAVANJA PROBLEMA

Koliko su gore navedeni problemi kompleksni svedoči broj različitih metoda za njihovo rešavanje, kao i to da za određene probleme postoje rešenja samo u prostoru manjih dimenzija. Neke od ovih metoda se konstantno razvijaju i njihovo istraživanje traje i danas, što je slučaj sa metodama Markovljevih lanaca. Kako bismo ih lakše opisali podelićemo metode u određene kategorije [3].

Prvoj kategoriji pripadaju **numeričke metode** koje problem izračunavanja očekivanja i različitih varijacija rešavaju direktnim izračunavanjem. Na primer, izračunavanje očekivanja $E(a)$ u konačnom prostoru će se vršiti sumiranjem jednačine (1.1), ali ovakvo izračunavanje je skoro nemoguće u prostoru većih dimenzije zbog eksponencijalnog rasta potrebnog vremena. Slična situacija je i u slučaju neprekidne raspodele koja zahteva izračunavanje integrala. Čak i neke novije metode numeričke integracije se ne mogu primeniti u rešavanju iznetih problema zbog toga što je u određenim delovima prostora vrednost integrala skoro jednaka 0, a u drugim, manjim delovima izuzetno velika.

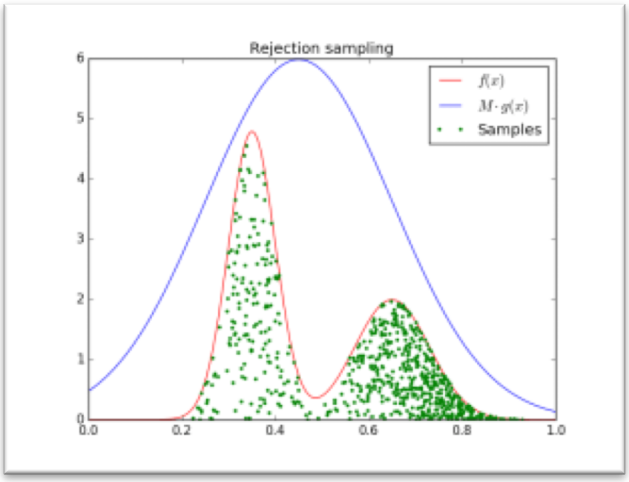
Drugi pristup je ocenjivanje očekivanja $E(a)$ pomoću Monte Karlo formule:

$$E(a) \approx \frac{1}{N}\sum_{t=0}^{N-1} a(x_1^{(t)}, \dots x_n^{(t)})$$

ukoliko možemo vršiti uzorkovanje iz raspodele definisane preko funkcije $f(x)$. U nekim slučajevima uzorkovanje nije moguće izvesti direktno, ali je moguće primeniti tehniku **uzorkovanja odbacivanjem** (*eng. rejection sampling*) koja se zasniva na generisanju nezavisnih uzoraka iz $f(x)$ pomoću uzorkovanja pristupačnije raspodele gde se dobijeni uzorci odbacuju ukoliko ne zadovoljavaju određene uslove. Da bi se primenio ovaj metod, moramo odabrati funkciju $g(x)$ takvu da za svako x i neku konstantu c ($1 < c < \infty$) važi $f(x) \leq cg(x)$. Uzorke iz raspodele sa gustinom $f(x)$ određujemo tako što iteriramo kroz sledeće korake:

- generišemo uzorak x^* iz raspodele sa gustinom $g(x)$ i odredimo broj u~Uniform[0,1]
- izračunamo $\frac{f(x)}{cg(x)}$ i prihvatamo x^* ukoliko važi $u < \frac{f(x)}{cg(x)}$.

Ukoliko uzorak x^* nije prihvaćen postupak nastavljamo sve dok ne naiđemo na uzorak koji će biti prihvaćen.



Slika 1: Metoda uzorkovanja odbacivanjem.
*Posmatrana funcija f(x) je predstavljena crvenom linijom, a pomoćna funkcija M*g(x) plavom, dok zelene tačke predstavljaju uzorke koji se prihvataju.*

Efikasnost ove metode zavisi od toga koliko često se generisani uzorci odbacuju, odnostno koliko dobro funkcija $g(x)$ aproksimira funkciju $f(x)$. Kod nekih kompleksnijih problema skoro da je nemoguće odrediti prikladnu funkciju $g(x)$ i konstantu c za koju će važiti $f(x) \leq cg(x)$, a da se pri tom ne smanji verovatnoća prihvatanja. Zbog toga ova metoda nije primenljiva kod komplikovanijih problema, ali postoje razne njene varijacije koje prevazilaze opisana ograničenja.

Još jedan značajan metod Monte Karlo ocenjivanja očekivanja je **uzorkovanje po značajnosti,** pomoću kojeg se očekivanje izračunava biranjem funkcija gustine $g(x)$, koja ne mora biti normalizovana i iz koje se lako mogu uzimati uzorci, a koja predstavlja aproksimaciju ciljane funkcije $f(x)$. Za razliku od metode uzorkovanja odbacivanjem, jedini uslov kod ove metode je da funkcija $g(x)$ bude različita od 0 kad god je $f(x)$ različita od 0. Sada možemo izračunati očekivanje funkcije a na sledeći način:

$$E_f = \frac{\sum_x a(x) f(x)}{\sum_x f(x)}$$

$$= \frac{\sum_x a(x) \frac{f(x)}{g(x)}}{\sum_x \frac{f(x)}{g(x)}}$$

$$= \frac{E_g\left[a(X)\frac{f(X)}{g(X)}\right]}{E_g\left[\frac{f(X)}{g(X)}\right]}$$

gde E_g predstavlja očekivanje funkcije a u odnosu na raspodelu zadatu preko funkcije $g(x)$, a E_f u odnosu na raspodeli zadatu preko $f(x)$. Monte Karlo ocena očekivanja u zavisnosti od funkcije $g(x)$ bi bila:

$$E_f \approx \frac{\sum_{t=0}^{N-1} a(x^{(t)})\frac{f(x^{(t)})}{g(x^{(t)})}}{\sum_{t=0}^{N-1}\frac{f(x^{(t)})}{g(x^{(t)})}},$$

gde su $x^{(0)}, \ldots, x^{(N-1)}$ uzorci iz raspodele date preko $g(x)$. Metod uzorkovanja po značajnosti izračunava srednju vrednost funkcije a u uzorkovanim tačkama i ocenjuje njihovu značajnost na osnovu toga koliko se predložena raspodela razlikuje od ciljane raspodele što opisuje vrednost težine značajnosti (*eng. importance weight*) $w(x) = \frac{f(x)}{g(x)}$. Ovaj metod takođe ima svoje nedostatke, jer nije uvek lako pronaći funkciju $g(x)$ koja je bliska funkciji $f(x)$, a u nekim slučajevima se čak može dogoditi da ni jedna od tačaka iz predložene raspodele se ne može uzeti kao uzorak za ciljanu raspodelu.

Poslednju grupu čine **Monte Karlo metode zasnovane na Markovljevim lancima** koje predstavljaju kombinaciju opisanih metoda tj. uzorkovanja i potrage za delovima sa najvećom verovatnoćom, a njihov princip rada će biti predstavljen u nastavku.

2.3 MARKOVLJEVI LANCI

Markovljevi lanci spadaju u relativno jednostavnu, ali i veoma interesantnu klasu slučajnih procesa. Oni opisuju slučajne pojave ili sisteme koji se menjaju tokom vremena, a njihova jednostavna struktura omogućava nam da saznamo mnogo toga o njihovom ponašanju u budućnosti. Ako uzmemo u obzir poznavanje nekih informacija o prošlim stanjima, uslovna raspodela budućih stanja sistema zavisi samo od onih najskorijih. Dakle možemo da kažemo da predviđanje budućnosti sistema zavisi samo od sadašnjeg stanja, a ne i od puta kojim je sistem došao do tog stanja.

Lanci Markova predstavljaju jedan od najprostijih matematičkih modela za opisivanje pojava u realnom zivotu. Pravilo da su najjednostavniji modeli često i najkorisniji za analizu praktičnih problema ne zaobilazi ni Markovljeve lance. Njima se mogu modelirati mnoge pojave, kao što su pojave u biologiji, psihologiji, sportu i dr [4].

Prilikom posmatranja određenih sistema, ponekad nam je važno da odredimo kako se stanje nekog sistema menja s vremenom. Ako stanje sistema predstavlja vrednost slučajne promenljive, sledi da nas zanima promena slučajne promenljive u zavisnosti od vremena. Ponašanje posmatranog sistema opisuje se stohastičkim procesom. Poseban oblik stohastičkog procesa je Markovljev proces odnosno Markovljev lanac.

Neka X_t predstavlja veličinu posmatranu u svakom trenutku *t* nekog vremenskog intervala *T,* koja nije unapred određena već se realizuje slučajno, a predstavlja stanje sistema u posmatranom trenutku *t*. Skup svih slučajnih veličina X_t posmatramo kao slučajnu promenljivu koja se menja u zavisnosti od vremena *t.*

Definicija: *Stohastički proces* $\{ X_t : t \in T\}$ *je Markovljev proces prvog reda ako za proizvoljan izbor vremenskih trenutaka* $t_1 < t_2 < \ldots < t_n \in T$ *i proizvoljne vrednosti* $x_1, x_2, x_3, \ldots, x_n$ *važi:*

$$P[\, X_{t_n} \leq x_n \mid X_{t_{n-1}} = x_{n-1}, \ldots, X_{t_1} = x_1] = P[X_{t_n} \leq x_n \mid X_{t_{n-1}} = x_{n-1}\,][6].$$

Predhodna relacija izražava tzv. *Markovljevo svojstvo* koje opisuje činjenicu da raspodela verovatnoće slučajne promenljive X_t u trenutku $t = t_n$ zavisi samo od vrednosti x_{n-1} sistema u trenutku t_{n-1}, a ne zavisi od vrednosti sistema u ranijim trenucima. Radi jednostavnijeg zapisa,

stanje sistema u trenutnku t_n ćemo označavati sa X_n i govoriti o njemu kao o n-tom koraku sistema. U zavisnosti od toga da li je T diskretan ili neprekidan skup, razlikujemo procese Markova sa diskretnim i neprekidnim vremenom. Sve vrednosti koje može da uzme slučajna veličina X_t čine skup stanja ili prostor stanja koji označavamo sa S. Ako je skup S diskretan tj. konačan ili prebrojiv tada se proces Markova zove lanac Markova.

Definicija: *Lanac Markova je slučajni proces* $\{X_t : t \in T\}$ *koji ima svojstvo Markova tj. proces gde za svaki prirodni broj* $n > 2$ *i svaki izbor stanja* $s_1, s_2, \ldots, s_n \in S$ *važi:*

$$P[X_n = s_n \mid X_{n-1} = s_{n-1}, \ldots, X_1 = s_1] = P[X_n = s_n \mid X_{n-1} = s_{n-1}][6].$$

Verovatnoća da se u trenutku n posle jednog koraka pređe iz stanja i u stanje j označavamo sa:

$$p_{ij}^{n,n+1} = P\{X_{n+1} = j \mid X_n = i\}, \qquad i, j \in S,\ S = \{s_1, s_2, \ldots, s_r\}$$

ukoliko verovatnoća $p_{ij}^{n,n+1}$ ne zavisi od trenutka n, tada pišemo samo

$$p_{ij} = P\{X_{n+1} = j \mid X_n = i\}, \qquad i, j \in S$$

Ovakvi lanci se nazivaju homogeni.

Za verovatnoću prelaska iz stanja i u stanje j posle n koraka kosristimo oznaku $p_{ij}(n)$. Sada možemo uvesti pojam matrice verovatnoće prelaska lanca Markova u n-tom koraku.

$$P(n) = \begin{pmatrix} p_{11}(n) & \cdots & p_{1r}(n) \\ p_{21}(n) & & p_{2r}(n) \\ \vdots & \ddots & \vdots \\ p_{r1}(n) & \cdots & p_{rr}(n) \end{pmatrix}$$

Uočimo da za matricu $P(n)$ važi $\sum_{j=1}^{r} p_{ij} = 1$ za svako $i = 1,2, \ldots r$. Matrice sa ovakvim svojstvom nazivaju se stohastičkim matricama [7]**.**

Da bismo Markovljev lanac u potpunosti opisali moramo, osim matrice verovatnoće prelaska, poznavati i vektor početnih vrednosti, koji predstavlja stanje Markovljevog lanca u početnom trenutku posmatranja, tj. vrednost slučajne promenljive X_0.

Vektor početnih vrednosti označimo sa:

$$\pi(0) = \left(\pi_0(0), \pi_1(0), \ldots, \pi_r(0)\right)$$

gde je:

$$\pi_i(0) = P[X_0 = s_i]$$

tj. verovatnoća da se sistem na početku nalazi stanju s_i.

Na isti način definišemo vektor stanja u *n*-tom koraku (ili nakon *n* koraka):

$$\pi(n) = \big(\pi_0(n), \pi_1(n), \dots, \pi_r(n)\big)$$

pri čemu je:

$$\pi_i(n) = P[X_n = s_i].$$

Takođe, jasno je da i ovi vektori stanja moraju biti stohastički, tj. mora važiti:

$$\pi_i(n) \geq 0 \text{ i } \sum_{i=0}^{r} \pi_i(n) = 1.$$

Ako nam je poznat vektor početnih stanja, onda se j-ti element vektora stanja u n-tom koraku računa:

$$\pi_j(n) = P\big[X_n = s_j\big] = \sum_{i=1}^{r} P\big[X_n = s_j | X_0 = s_i\big] * P[X_0 = s_i] = \sum_{i=0}^{r} p_{ij}(n) * \pi_i(0)$$

Izraz na kraju gornje jednakosti predstavlja *j*-ti element umnoška vektora početnih vrednosti matricom P^n. Iz toga sledi da je:

$$\pi(n) = \pi(0)P^n$$

odnosno, ako želimo odrediti vektore stanja u svim koracima imamo

$$\pi(1) = \pi(0)P, \pi(2) = \pi(1)P, \dots, \pi(n) = \pi(n-1)P.$$

Objasnimo ukratko klasifikaciju stanja Markovljevog lanca kako bismo lakše formulisali naredne definicije [6]:

- Za dva stanja *i* i *j*, putanja od *i* do *j* je niz prelaza koji počinju u *i* i završavaju u *j*, tako da je pri svakom prelazu u nizu verovatnoća pozitivna.
- Stanje *j* je dostupno iz stanja *i* ako postoji putanja koja vodi od *i* do *j*.
- Dva stanja su povezana ako je stanje *j* dostupno iz *i*, i ako je stanje *i* dostupno iz *j*
- Skup stanja S u Markoljevom lancu je zatvoren skup ako ne postoji stanje izvan skupa S koje je dostupno iz nekog stanja iz S.
- Stanje *i* je apsorbujuće ako je $p_{ii} = 1$.

- Stanje *i* je prelazno stanje ako postoji stanje *j* koje je dostupno iz *i,* ali stanje *i* nije dostupno iz stanja *j*. Prelazno stanje omogućuje da napustimo stanje *i* ali se više nikada u njega ne vraćamo.
- Stanje koje nije prelazno, naziva se rekurentno stanje (povratno stanje).
- Stanje *i* je periodično sa periodom $k > 1$ ako je *k* najmanji broj takav da sve putanje vode iz stanja *i* ponovo u stanje *i,* i koje imaju dužinu jednaku deliocu broja *k*. Ako rekurentno stanje nije periodično, ono je aperiodično.

Definicija*: Za Markovljev lanac u kome su sva stanja međusobno rekurentna, aperiodična i povezana kaže se da je ergodičan (regularan).*

Regularnom Markovljevom lancu pripada regularna matrica verovatnoće prelaza, odakle sledi da postoji $n > 1$, takav da je $p_{ij}(n) > 0$ za sve $i, j = 1, \dots r$. To znači, da Markovljev lanac iz proizvoljnog stanja, nakon određenog broja koraka može preći u bilo koje drugo stanje. Na osnovu navedenog jasno je da apsorbujući Markovljev lanac ne može biti regularan.

Definicija*: Stohastički vektor $\pi=(\pi_1, \dots . \pi_r)$ nazivamo vektorom stacionarnih vrednosti ako važi:*

$$\pi P = \pi.$$

Ako je $\pi(0) = \pi$, onda je zbog gornje jednakosti $\pi(1) = \pi(0)P = \pi(0) = \pi$, odakle sledi da je i $\pi(n) = \pi$. Regularni Markovljevi lanaci imaju jedinstven vektor stacionarnih vrednosti. Ako je P matrica verovatnoće prelaza regularnog Markovljevog lanca, onda je:

$$\lim_{n\to\infty} P^n = \begin{pmatrix} \pi_1 & \pi_2 & \dots & \pi_r \\ \vdots & \vdots & \ddots & \vdots \\ \pi_1 & \pi_2 & \dots & \pi_r \end{pmatrix}$$

pa iz ove jednakosti sledi:

$$\lim_{n\to\infty} p_{ij} = \pi_j$$

gde je π_j $j-$ta koordinata vektora stacionarnih vrednosti. To znači da nakod određenog broja koraka više nije važno koje je početno stanje sistema već će se sistem bez obzira na to naći u stanju s_j sa verovatnoćom π_j [7].

3. METROPOLIS ALGORITAM

3.1 UVOD

Metrpolis algoritam predstavlja jedan od Monte Karlo algoritama koji se danas najčešće koristi, a ime je dobio po svom autoru Nikolas Metropolisu [5]. Ovaj algoritam ima ogroman uticaj na razvoj nauke i u širokoj je upotrebi u oblasti statistike, fizike, ekonomije i računarskih nauka. Najkritičniji korak pri razvijanju efikasne Monte Karlo metode je simulacija odgovarajuće raspodele verovatnoće $\pi(x)$. Kada nije moguće direktno generisati nezavisne uzorke iz raspodele $\pi(x)$, tada se obično koristi strategija uzorkovanja po značajnosti u kojoj se slučajni uzorci generišu iz predložene probne raspodele i skaliraju prema količniku značaja ili se generišu statistički zavisni uzorci koristeći MKML (Monte Karlo Markovljevi Lanci) metodu.

Pretpostavimo da se sve raspodele verovatnoće mogu zapisati u obliku $\pi(x) = Z^{-1}h(x)$, gde je konstanta normalizacije Z nepoznata. Zapravo Z je poznata i predstavlja vrednost integrala $Z = \int h(x)\,dx$ čije izračunavanje je glavni problem simulacije raspodele π. Kako bi rešio ovaj problem, Metropolis predstavlja algoritam koji nadograđuje Markovljev proces uzorkovanja iz raspodele π [5]. Iako je jednostavan, ovaj algoritam je veoma moćan, a razne njegove varijacije i proširenja su pronašla široku upotrebu u različitim naučnim oblastima.

Metropolis algoritam se može koristiti za generisanje slučajnih uzoraka iz bilo koje ciljane raspodele $\pi(x)$, bez obzira na analitičnu kompleksnost i dimenziju. Iako je ova tvrdnja tačna u teoriji, jedan od potecijalnih problema Monte Karlo metoda baziranih na Markovljevim lancima je to što su rezultujući uzorci često u velikoj korelaciji. Procena razultata kod ovakvih uzoraka znatno varira u odnosu na nezavisne uzorke, pa se zbog toga i dalje prave različiti pokušaji kako bi se ovo ograničenje prevazišlo.

3.2 METROPOLIS ALGORITAM

Osnovna ideja Metropolis algoritma je da simulira Markovljev lanac u prostoru S tako da stacionarna raspodela lanca bude ciljana raspodela π. Kod tradicionalnih analiza Markovljevih lanaca pravila prelaza[1] su obično data, a nepoznata je stacionarna raspodela, dok je kod Monte Karlo Markovljevih lanaca poznata ravnotežna raspodela, a pravila prelaza se opisuju tako da se ta ravnoteža postigne. Metropolis algoritam koristi predloženu raspodelu koja zavisi od trenutnog stanja sistema $x^{(t)}$. Počevši od bilo koje konfiguracije $x^{(0)}$, Metropolis algoritam se sastoji od iteracije kroz sledeća dva koraka:

K1: Predložimo slučajnu nepristrasnu promenu (eng. *unbiased perturbation*) trenutnog stanja $x^{(t)}$ tako da generiše novo stanje x'. x' je generisana iz simetrične funkcije verovatnoće prelaska[2] $T(x^{(t)}, x')$ (koja se često naziva predložena funkcija ili probna funkcija, $T(x^{(t)}, x') = T(x', x^{(t)})$). Nakon ovoga izračunamo promenu:

$\Delta h = h(x') - h(x^{(t)})$.

K2: Generišimo slučajni broj $U \sim \text{Uniform}[0,1]$. Neka je $x^{(t+1)} = x'$ ako važi

$$U \leq \frac{\pi(x')}{\pi(x^{(t)})} \equiv \exp(-\Delta h),$$

a $x^{(t+1)} = x^{(t)}$ inače.

Ova dva koraka zapravo opisuju sledeći postupak: (a) napraviti malu promenu trenutne konfiguracije, (b) a zatim izračunati "poboljšanje" posmatrane funkcije. Nakon ovoga, (c) generisati slučajni broj U, (d) na osnovu kojeg će se nova konfiguracija usvojiti ukoliko je $log(U)$ manji ili jednak od "poboljšanja". Prema tome određivanje M uzoraka iz ciljane raspodele π se postiže izvršavanjem sledećeg pseudokoda [15]:

1. *t = 0*
2. generisati početno stanje x^0
3. dokle god je *t* < *M*:
   t = t + 1

[1] Pravilo prelaza predstavlja uslovnu raspodelu koja određuje koje su šanse prelaska iz jedne tačke prostora u drugu.

[2] Funkcija T(x, y) se naziva funkcijom verovatnoće prelaska ukoliko je ne-negativna i za svako x važi $\sum_{svako\ y} T(x, y) = 1$.

generisati novo stanje x' iz predložene raspodele T

izračunati verovatnoću prihvatanja $\alpha = \frac{\pi(x')}{\pi(x^{(t)})}$

odrediti slučajni broj U iz uniformne raspdele *U(0,1)*

ukoliko je $U < \alpha$, $x^t = x'$

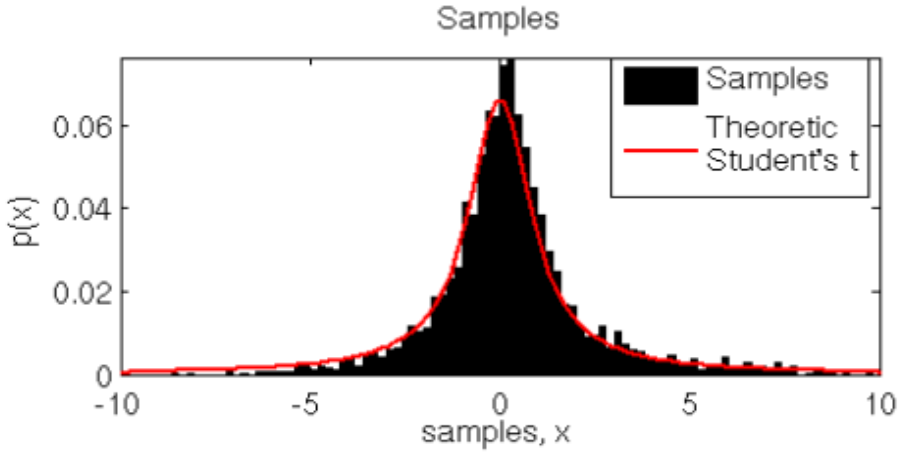
u suprotnom $x^t = x^{t-1}$

Iz navedenog se vidi da je Metropolis algoritam konstruisan na osnovu strategije pokušaja i pogrešaka. Kako su svi prihvaćeni uzorci u korelaciji, jer uzorak $x^{(t)}$ ima verovatnoću raspodele koja zavisi od $x^{(t-1)}$, to se ovaj algoritam mora ponavljati priličan broj puta kako bi se dobili nezavisni uzorci raspodele.

Metropolis, kao i ostali naučnici, su ograničili svoj izbor "pravila promene" samo na simetrične funkcije, što znači da je verovatnoća dobijanje x' perturbacijom x jednaka verovatnoći dobijanja x perturbacijom x'. Matematički se ova restrikcija može zapisati kao:

$$T(x, x') = T(x', x).$$

Primer: Pretpostavimo da želimo da odredimo uzorke iz nenormalizovane Studentove raspodele $p(x) = (1 + x^2)^{-1}$ koristeći Metropolisov algoritam. Prvi korak predstavlja određivanje početnog stanja, i neka to bude broj $x^0 \sim N(0,1)$, dok ćemo za probnu raspodelu $T(x, x')$ uzeti normalnu raspodelu $N(x, 1)$. Na *Slici 2* je predstavljen rezultat nakon 5000 iteracija uzorkovanja (M = 5000), gde su uzorci dobijeni Metropolisovim algoritmom predstavljeni crnom linijom, a ciljana raspodela crvenom, pa se jasno može videti da uzorci prate zadatu raspodelu.



Slika 2: Odnos uzoraka dobijenih Metropolisovim algoritmom i zadate raspodele p(x).

3.3 MATEMATIČKA FORMULACIJA METROPOLIS - HESTINGOVOG ALGORITMA

Metropolis algoritam opisuje pravila prelaska kod Markovljevih lanaca koristeći simetričnu predloženu funkciju $T(X,Y)$, dok Hesting (1970) proširuje ovaj algoritam slučajem u kome T ne mora nužno biti simetrična [17]. Jedina restrikcija Hestingove metode jeste da za predloženu funkciju T mora važiti $T(X,Y) > 0$ ako i samo ako je $T(Y,X) > 0$. Metropolis - Hesting algoritam se može opisati sledećim koracima:

- odabrati y iz predložene raspodele $T(x^{(t)}, y)$.
- odabrati $U \sim Uniform[0,1]$ i izračunati

$$x^{(t+1)} = \begin{cases} \mathrm{y}, & \mathrm{U} \leq \mathrm{r}(x^{(t)}, \mathrm{y}) \\ x^{(t)} & \text{inače} \end{cases}$$

Hesting za funkciju r predlaže:

$$r(x,y) = min\left\{1, \frac{\pi(y)T(y,x)}{\pi(x)T(x,y)}\right\}.$$

Ukoliko je T simetrična funkcija tada je ovaj algoritam identičan Metropolis algoritmu. Još neke od predloga za funkciju r su dali Barker [18]:

$$r_B = \frac{\pi(y)T(y,x)}{\pi(y)T(y,x) + \pi(x)T(x,y)}$$

i Čarls Stejn (eng. *Charls Stein*):

$$r(x,y) = \frac{\delta(x,y)}{\pi(x)T(x,y)}, \qquad (2.1)$$

gde je $\delta(x,y)$ bilo koja simetrična funkcija za koju važi $r(x,y) \leq 1$ za svako x i y. Ukoliko je funkcija odbacivanja oblika (2.1) i $x \neq y$ tada je verovatnoća prelaska iz x u y jednaka:

$$A(x,y) = T(x,y)\, r(x,y) = T(x,y)\frac{\delta(x,y)}{\pi(x)T(x,y)} = \pi(x)^{-1}\delta(x,y)$$

Kako je funkcija δ simetrična odatle sledi da je

$$\pi(x)A(x,y) = \pi(y)A(y,x), \qquad (2.2)$$

tj. da je Markovljev lanac reverzibilan, a π invarijantna raspodela. Jednakost 2.2 se naziva i detaljna uravnoteženost (end. *detailed balance*), pomoću koje možemo dokazati da važi:

$$\int \pi(x)A(x,y)dx = \pi(y).$$

Dokaz:

$$\int \pi(x)A(x,y)dx = \int \pi(y)A(y,x)dx = \pi(y)\int A(y,x)dx = \pi(y).$$

Zbog ovoga detaljna uravnoteženost obezbeđuje invarijantrnost, a svaki Markovljev lanac koji zadovoljava ovaj uslov je reverzibilan.

Pokažimo sada da uslov detaljne uravnoteženosti važi i kod Metropolis-Hestingovog algoritma. U slučaju da je $x \neq y$ verovatnoća prelaska iz x u y jednaka je proizvodu predložene verovatnoće $\mathrm{T}(x,y)$ i verovatnoće prihvatanja koraka tj:

$$A(x,y) = T(x,y)min\left\{1, \frac{\pi(y)T(y,x)}{\pi(x)T(x,y)}\right\},$$

pa je stoga,

$$\pi(x)A(x,y) = \pi(x)T(x,y)min\left\{1, \frac{\pi(y)T(y,x)}{\pi(x)T(x,y)}\right\} = min\{\pi(x)T(x,y), \pi(y)T(y,x)\},$$

što predstavlja simetričnu funkciju po x i y, zbog čega je uslov detaljne uravnoteženosti zadovoljen.

Uopštenije, ako je funkcija prelaska $A(x,y)$ oblika

$$A(x,y) = \pi(y)\delta(x,y),$$

gde je $\delta(x,y)$ simetrična funkcija po x i y, takva da važi $\int A(x,y)dy = 1$, onda se lako može ustanoviti da je uslov detaljne uravnoteženosti zadovoljen i da se Metropolis funkcija prelaska može zapisati kao:

$$A(x,y) = \pi(y)min\left\{\frac{T(x,y)}{\pi(y)}, \frac{T(y,x)}{\pi(x)}\right\}, \qquad x \neq y.$$

Ukoliko se koristi Barkerovo pravilo prihvatanja tada je funkcija prelaska sledećeg oblika:

$$A(x,y) = \pi(y)\frac{T(x,y)T(y,x)}{\pi(y)T(y,x) + \pi(x)T(x,y)}$$

Prema standardnoj teoriji Markovljevih lanaca [7], ukoliko je lanac nesvodljiv [3], aperiodičan [4] i ima invarijantnu raspodelu nakon n_0 koraka, lanac će konvergirati ciljanoj raspodeli π.

[3] Lanac je nesvodljiv ukoliko je verovatnoća prelaska iz jedne pozicije sistema u bilo koju drugu u konačnom broju koraka različita od nule.

[4] Markovljev lanac je aperiodičan ukoliko je najveći zajednički delilac broja koraka potrebnih da se lanac vrati u početno stanje jednak 1.

3.4 SPECIJALNI METROPOLIS ALGORITMI

Kako bismo ilustrovali praktičnost Metropolis - Hestingovog algoritma opisaćemo nekoliko njegovih specijalnih slučajeva koji se najčešće pronalaze u literaturi.

3.4.1 SLUČAJNO KRETANJE

Slučajno kretanje je jedan od MKML algoritama koji je danas najčešće u upotrebi, a koji je prvi put opisan u Metropolisovom radu 1953. godine [5]. Pretpostavićemo da je ciljana raspodela $\pi(x)$ definisana na d-dimenzionalnom Euklidovom prostoru $\mathbb{R}^d$. Metod slučajnog kretanja se zasniva na lokalnom istraživanju okoline trenutne vrednosti Markovljevog lanca. Ideja ove impelemntacije je da se odabere uzorak y takav da je:

$$y = x^{(t)} + \epsilon_t$$

gde je ϵ_t slučajna promena (perturbacija) sa raspodelom g koja ne zavisi od $x^{(t)}$, npr. uniformna $y \sim U\left(x^{(t)} - \delta, x^{(t)} + \delta\right)$ ili normalna raspodela $y \sim N\left(x^{(t)}, \tau^2\right)$. g predstavlja predloženu raspodelu i oblika je $g(y - x)$, a Markovljev lanac izveden pomoću ove raspodele se naziva slučajno kretanje ukoliko je g simetrična.

Metropolis algoritam slučajnog kretanja se sastoji iz iteracije kroz sledeće korake [11]:

- Odabrati $\epsilon_t \sim g(y - x^{(t)})$ i izračunati $y = x^{(t)} + \epsilon_t$
- Odabrati $u \sim Uniform[0,1]$ i izračunati

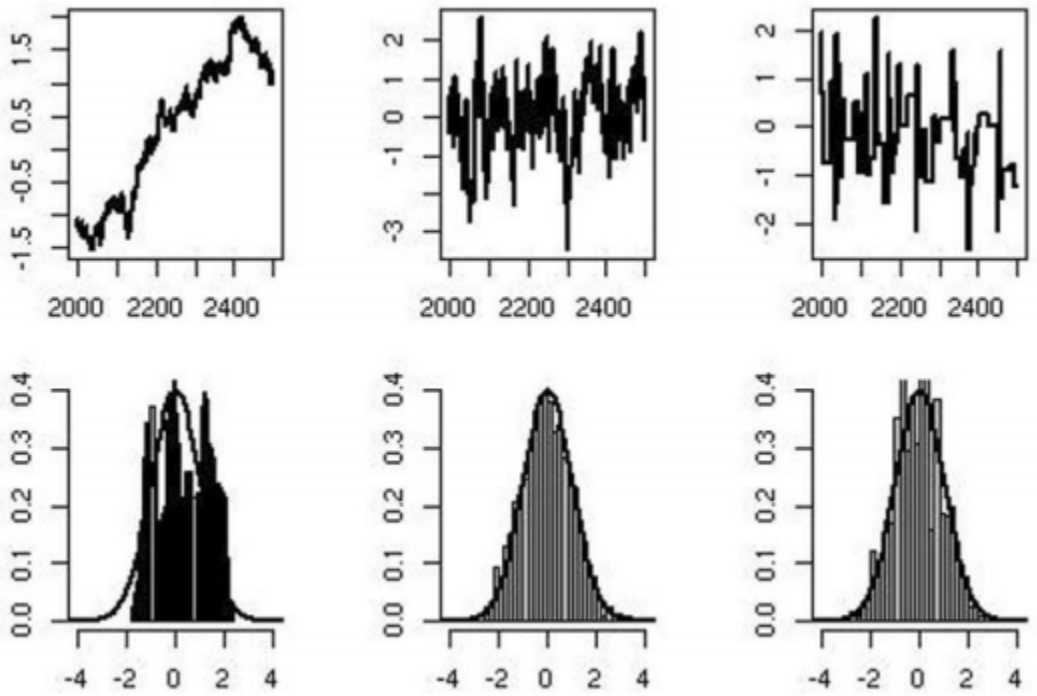
$$x^{(t+1)} = \begin{cases} y & ako\ je\ u \leq \dfrac{\pi(y)}{\pi(x^{(t)})} \\ x^{(t)} & inače \end{cases}$$

Kao što se vidi iz algoritma verovatnoća prihvatanja koraka ne zavisi od g, ali menjanje ove funcije se odražava na opseg vrednosti y i stopu prihvatanja. Često se ovaj metod koristi u ograničenim domenima, pa kada je y van tog domena, tada je verovatnoća prihvatanja jednaka 0, što znači da se ovaj uzorak odbacuje, a trenutna vrednost lanca se duplira. Zbog ovoga, ukoliko funkcija g često generiše uzorke van domena dolazi do stagniranja lanca.

Primer slučajnog kretanja je dao Hesting u pokušaju generisanja normalne raspodele $N(0,1)$ sa predloženom uniformnom raspodelom $[-\delta, \delta]$. Verovatnoća prihvatanja je tada jednaka

$$p\left(x^{(t)}, y\right) = exp\left\{\left(x^{(t)2} - \frac{y^2}{2}\right)\right\} \wedge 1.$$

Na *Slici 3* su prikazani rezultati ovog problema sa uzorkom od 5000 tačaka i $\delta = 0.1, 1, 10$ gde se tačno vidi razlika proizvedenih lanaca: previše zbijeni ili previše rašireni kandidati u zavisnosti od δ, kao i spora konvergencija.



Slika 3: Rezultat Hestingovog problema korišćenjem metode slučajnog kretanja. Na levoj strani su predstavljeni rezultati korišćenjem U(-0.1,0.1), u sredini U(-1,1) i desno U(-10,10). Gornji grafikon predstavlja poslednjih 500 iteracija, dok je na donjem prikazano kako histogram dostiže ciljanu raspodelu.

3.4.2 METROPOLIZOVAN NEZAVISNI UZORKIVAČ

Veoma specifičan izbor za predloženu funkciju prelaska $T(x, y)$ predstavlja gustina raspodele $g(y)$, gde se predloženi korak y generiše iz funkcije $g(\cdot)$ nezavisno od prethodnog stanja $x^{(t)}$. Ovaj metod (*eng. Metropolized Independence Sampler, skraćeno MIS*) predstavlja alternativu za uzorkovanje odbacivanjem i uzorkovanje prema značaju, a sastoji se iz sledećih koraka:

- Odabrati $y \sim g(y)$
- Simulirati $u \sim Uniform[0,1]$ i izračunati

$$x^{(t+1)} = \begin{cases} y & ako\ je\ u \leq min\left\{1, \frac{w(y)}{w(x^{(t)})}\right\} \\ x^{(t)} & inače \end{cases}$$

gde $w(x) = \frac{\pi(x)}{g(x)}$ predstavlja težinu pri uzorkovanju po značajnosti.

Kao i kod metode odbacivanja, efikasnost ove metode zavisi od toga koliko je probna gustina $g(y)$ blizu ciljanoj $\pi(y)$. Kako bi se obezbedile dobre performanse, preporučuje se da gustina $g(y)$ bude relativno dugorepa[5] (*eng. long-tailed*) raspodela. Galman, Rubin i Tirni (eng. Tierney) [19] predlažu ubacivanje nekoliko koraka MIS metode u Gibs iteraciju kada je ispravno uzorkovanje iz uslovne raspodele otežano. Ova metoda je veoma korisna kod različitih Bajesovih izračunavanja kod kojih svaka uslovna gustina može biti aproksimirana veoma dobro Gausovom raspodelom.

[5] Dugi rep se odnosi na statističko svojstvo prema kojem se na repu raspodele verovatnoće nalazi veći deo populacije nego što je to slučaj kod normalne odnosno Gausove distribucije.

4. GIBSOV UZORKIVAČ

4.1 UVOD

Gibsov algoritam je naziv dobio po fizičaru Josiah Willard Gibbs-u, a predstavljen je 1984. godine, osam decenija nakon njegove smrti, od strane braće Stjuart i Donald Geman (*eng. Stuart and Donald Geman*) [22]. U svojoj osnovnoj verziji ovaj algoritam je predstavljao specijalan slučaj Metropolis-Hestingovog algoritma, ali se danas smatra za generalnu osnovu uzorkovanja iz velikog skupa promenljivih, a Metropolis algoritam (kao i drugi pogodni algoritami) se može koristiti za implementaciju određenih koraka uzorkovanja.

Gibsov uzorkivač se koristi ukoliko je nepoznata zajednička raspodela ili je teško vršiti direktno uzorkovanje, dok je uslovna raspodela poznata i jednostavna za uzorkovanje. Kao i ostali MKML algoritmi, i Gibsov uzorkivač generiše od uzoraka Markovljev lanac gde je svaki sledeći u korelaciji sa predhodnim uzorkom. Kako bi se dobili nezavisni uzorci potrebno je "prorediti" dobijeni rezultat, kao i odbaciti početne uzorke koji ne odgovaraju željenoj raspodeli. Prednost ovog algoritma u odnosu na ostale je to što prati lokalnu dinamiku ciljane raspodele, a i u mnogim slučajevima predstavlja povoljniji način uzorkvanja od ostalih algoritama. Implementacija Gibsovog algoritma uzorkovanja biće predstavljena u narednom poglavlju gde će biti predstavljeni i neki primeri upotrebe ove metode.

4.2 GIBSOV ALGORITAM UZORKOVANJA

Gibsov algoritam predstavlja specijalnu MKML šemu, čija je glavna karakteristika konstruisanje Markovljevog lanca spajanjem sekvenci uslovnih raspodela duž različitih pravaca (obično duž koordinatnih osa).

Pretpostavimo da možemo slučajnu promenljivu raščlaniti na d komponenti (npr. x = $(x_1, \dots x_d)$), a zatim odaberemo jednu od koordinata, recimo x_1, i "popravimo" je sa novim uzorkom x'_1, uzetim iz uslovne raspodele $\pi(\cdot\,|\mathbf{x}_{[-1]})$, gde je $\mathbf{x}_{[-i]} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d)$. Na osnovu algoritma, Gibsovu strategiju uzorkovanja možemo podeliti na dva tipa koja ćemo sada opisati.

Gibsov uzorkivač slučajnim skeniranjem. Neka je nakon *t* iteracija $\mathbf{x}^{(t)} = (x_1^{(t)}, \dots . x_d^{(t)})$, *t+1*. iteracija sastoji iz sledećih koraka:

- nasumično izaberemo koordinatu $i \in \{1 \dots d\}$ prema zadatom vektoru verovatnoća $\{\alpha_1, \dots . , \alpha_d\}$ (npr. $(\frac{1}{d}, \dots, \frac{1}{d})$)
- odredimo $x_i^{(t+1)}$ iz uslovne raspodele $\pi(\cdot\,|x_{[-i]}^{(t)})$ i ostavimo ostale komponente nepromenjene tj. da važi:

$$\mathbf{x}_{[-i]}^{(t)} = \mathbf{x}_{[-i]}^{(t+1)}$$

Gibsov uzorkivač sa sistematskim skeniranjem. Neka je $\mathbf{x}^{(t)} = (x_1^{(t)}, \dots . x_d^{(t)})$, u *t+1*.iteraciji:

- odredimo $x_i^{(t+1)}$ iz uslovne raspodele

$$\pi(x_i|x_1^{(t+1)}, \dots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \dots x_d^{(t)})$$

za $i \in \{1 \dots d\}$.

Lako se može proveriti da nakon svakog koraka bilo kojeg algoritma raspodela π ostaje invarijantna. Da bismo ovo dokazali neka je $\mathbf{x}^{(t)} \sim \pi$, tada $\mathbf{x}_{[-i]}^{(t)}$ prati marginalnu raspodelu i zato važi:

$$\pi\left(x_i^{(t+1)}\middle|\mathbf{x}_{[-i]}^{(t)}\right) \times \pi\left(\mathbf{x}_{[-i]}^{(t)}\right) = \pi\left(\mathbf{x}_{[-i]}^{(t)}, x_i^{(t+1)}\right),$$

što znači da nova konfiguracija i dalje prati ciljanu raspodelu π. Takođe se može pokazati da lanac formiran Gibsovim uzorkivačem geometrijski konvergira i da je stopa konvergencije povezana sa korelacijom promenljivih. Ova stopa konvergencije se kontroliše maksimalnom korelacijom

između dve uzastopne iteracije, a Liu, Wong i Kong su istraživali kako grupisanje ovih usko povezanih komponenti može poboljšati efikasnost Gibsovog algoritma [23].

Jednostavna promena uslovnog ažuriranja konfiguracije može poboljšati Gibsov uzorkivač: svaki korak Gibsovog algoritma možemo posmatrati kao slučajno premeštanje trenutnog stanja **x** duž odabranog pravca (kod Metropolis algoritma koristili smo termin perturbacija stanja x). Na primer, ukoliko je odabran pravac prve koordinate, tada se slučajno premeštanje predstavlja na sledeći način:

$$(x_1, \dots x_d) \rightarrow (x_1 + \gamma, \dots x_d),$$

gde je γ slučajno izabran broj iz odgovarajuće raspodele. Ukoliko se γ izabere tako da važi $p(\gamma) \propto \pi(x_1 + \gamma, \mathbf{x}_{[-1]})$ tada je π invarijantna.

Popularnost Gibsovog algoritma u statistici proizilazi iz upotrebe uslovne raspodele u svakoj iteraciji. Njenu osnovnu teoriju je predstavio Liu, a Gelfand i Smith su demonstrirali upotrebu uslovne raspodele u mnogim Bajesovim izračunavanjima i izračunavanjima verovatnoće [24].

4.3 PRIMERI GIBSOVOG UZORKIVAČA

Za početak opisaćemo algoritam Gibsovog uzorkivača na uopštenom slučaju pridružene funkcije raspodele $f(x_1, x_2)$.

1. Generisati početni vektor $X_0 = (X_{0,1}, X_{0,2})$, t = 0
2. Generisati $X_{t+1,1}$ iz raspodele $f(X_{t+1,1}|X_{t,2} = x_{t,2})$
3. Generisati $X_{t+1,2}$ iz raspodele $f(X_{t+1,2}|X_{t+1,1} = x_{t+1,1})$
4. t = t + 1, vratiti se na korak 2.

Sada ćemo primeniti ovaj algoritam na konkretnom slučaju gde je $f(x_1, x_2)$ bivarijantna Gausova raspodela. Neka je x = (x_1, x_2) i ciljana raspodela:

$$N\left\{\begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}, \begin{pmatrix}1 & \rho\\ \rho & 1\end{pmatrix}\right\} = N\left\{\begin{pmatrix}1\\ 2\end{pmatrix}, \begin{pmatrix}1 & 0.9\\ 0.9 & 1\end{pmatrix}\right\}.$$

Markovljev lanac $\mathbf{x}^{(t)} = (x_1^{(t)}, x_2^{(t)})$ se pomoću Gibsovog uzorkivača sa sistematskim skeniranjem generiše na sledeći način:

$x_2^{(t+1)}| \, x_1^{(t+1)} \sim N\left\{\mu_2 + (\rho x_1^{(t+1)} - \mu_1), (1 - \rho^2)\right\}$,

$x_1^{(t+1)}| \, x_2^{(t)} \sim N\left\{\mu_1 + \rho(x_2^{(t)} - \mu_2), (1 - \rho^2)\right\}$.

MATLAB kod ovog algoritma je [16]:

```
// Inicijalizujmo konstante i niz
n = 6000;  // broj iteracija
xgibbs = zeros(n, 2);
rho = 0.9;
y = [1;2]; // očekivanje
sig = sqrt(1 - rho^2);
xgibbs(1,:) = [10,10]; // Početna tačka
for i = 2:n
     mu = y(1) + rho(xgibbs(i-1,2) - y(2));
     xgibbs(i,1) = mu + sig*randn(1);
     mu = y(2) + rho(xgibbs(i,1) - y(1));
     xgibbs(i,2) = mu + sig*randn(1);
end;
```

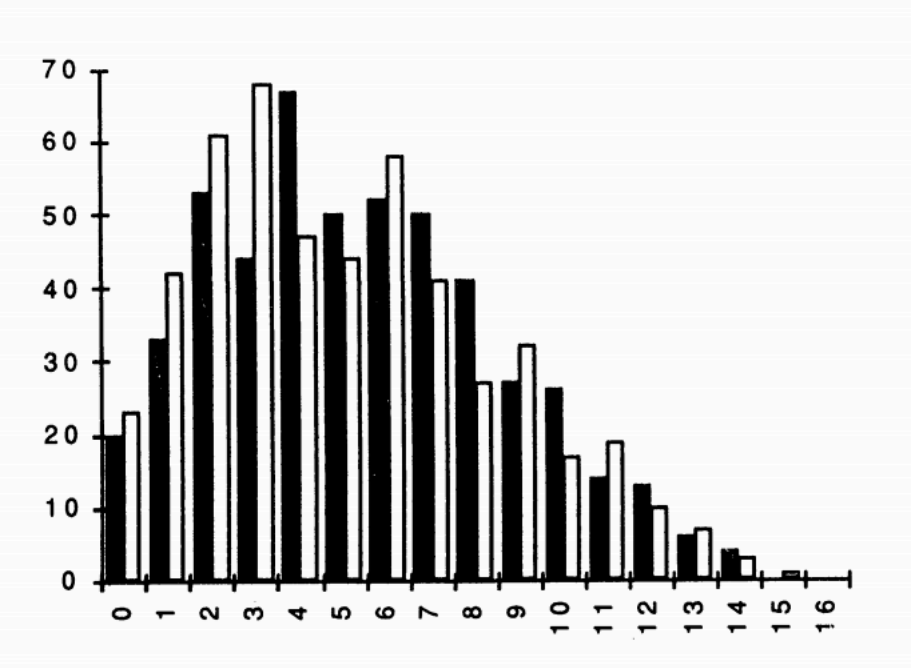
Još jedan primer su dali Casella i George [25], gde je ciljana raspodela data kao:

$$\pi(x, y) \propto \binom{n}{x} y^{x+\alpha-1}(1-y)^{n-x+\beta-1}$$

za $x = 0, 1, \ldots, n$ i $0 \leq y \leq 1$, a uslovne raspodele su:

$$x|y \sim Binom(n, y) \text{ i } y|x \sim Beta(x + \alpha, n - x + \beta).$$

Rezultati njihove simulacije su predstavljeni na *Slici 4* za sledeće vrednosti konstanti, n = 16, $\alpha = 2, \beta = 4$.



Slika 4: Primer upotrebe Gibsovog uzorkivača sa Beta-Binomnom raspodelom za n = 16 i $\alpha = 2, \beta = 4$. Crnom bojom su predstavljeni uzorci dobijeni Gibsovim algoritmom, dok su prave vrednosti Beta-Binomne raspodele predstavljene belom bojom.

4.4 SPECIJALNI PRIMERI GIBSOVOG UZORKIVAČA

4.4.1 UZORKOVANJE PO NIVOIMA

Pretpostavimo da je posmatrana funkcija gustine $\pi(x)$ i $x \in \mathbb{R}^d$. Tada je određivanje x$\sim \pi(x)$ ekvivalentno generisanju z = $(z_1, \dots, z_{d+1})$ koja ima uniformnu rasporedelu na segmentu S , gde je S jednako:

$$S = \{z \in \mathbb{R}^{d+1}: z_{d+1} \leq \pi(z_1, \dots, z_d)\}.$$

Međutim, generisanje slučajne promenljive sa uniformnom raspodelom može biti podjednako teško kao i originalni problem Monte Karlo simulacije. Da bi se odredio uzorak, koristi se sledeća Gibsova iteracija:

- odrediti $y^{(t+1)} \sim Uniform\left[0, \pi(x^{(t)})\right]$.
- odrediti $x^{(t+1)}$ uniformno na segmentu $S^{(t+1)} = \left\{x: \pi(x) \geq y^{(t+1)}\right\}$

Neka je π moguće zapisati kao proizvod *k* funkcija ($\pi(x) = f_1(x) \times \dots \times f_k(x)$) i neka su pomoćne promenljive $y_1, \dots, y_k$, tada se Gibsov uzorkivač za uzorkovanje (x, $y_1, \dots, y_k$) uniformno na segmetu $0 < y_i < f_i \; i = 1, \dots, k$ može predstaviti kao iteracija sledećih koraka:

- odrediti $y_i^{(t+1)} \sim Uniform\left[0, f_i\ (x^{(t)})\right], i = 1, \dots, k.$
- odrediti $x^{(t+1)}$ uniformno na segmentu

$$S^{(t+1)} = \bigcap_{i=1}^{k} \left\{x: f_i\ (x) \geq y_i^{(t+1)}\right\}$$

Damien, Vejkfild i Voker (*eng. Damien, Wakefield and Walker*) [26] su pokazali da se u mnogim slučajevima može pronaći dekompozicija funkcije π takva da je lako odrediti skup $S^{(t+1)}$, a samim tim i implementirati uzorkivač. Međutim, stopa konvergencije i dalje može biti niska zbog upotrebe pomoćnih promenljivih.

4.4.2 METROPOLIZOVAN GIBSOV UZORKIVAČ

Kada je prostor koji posmatramo diskretan, tada se koristi sledeća strategija kako bi se poboljšao Gibsov algoritam uzorkovanja. Neka je $\mathbf{x} = (x_1, \dots x_d)$, gde svako x_i može uzeti m_i različitih vrednosti, i neka je $\pi(\mathbf{x})$ posmatrana funkcija raspodele. Kod Gibsovog uzorkivača sa slučajnim skeniranjem prva koordinata koju biramo je *i,* a trenutnu vrednost koordinate

x_i menjamo sa izabranom vrednošću y_i prema odgovarajućoj uslovnoj raspodeli. U ovom algoritmu vrednost y_i se odabira sa verovatnoćom:

$$\frac{\pi(y_i \,|\mathbf{x}_{[-i]})}{1-\pi(x_i \,|\mathbf{x}_{[-i]})},$$

a zatim se x_i zamenjuje sa y_i sa Metropolis-Hestingovom verovatnoćom prihvatanja:

$$min\left\{1, \frac{1-\pi(x_i \,|\mathbf{x}_{[-i]})}{1-\pi(y_i \,|\mathbf{x}_{[-i]})}\right\},$$

dok u suprotnom ostaje nepromenjena vrednost. Ovaj algoritam se pokazao znatno efikasnijim od Gibsovog algoritma slučajnog skeniranja.

4.4.3 HIT AND RUN ALGORITAM

Hit and run je MKML algoritam u kome u svakoj iteraciji *t* slučajno odabiramo pravac u $\mathbb{R}^k$. Pretpostavimo da je $\mathbf{x}^{(t)}$ trenutno stanje, algoritam hit and run (skraćeno HR) se sastoji iz sledećih koraka:

- uniformno odaberemo pravac $\mathbf{e}^{(t)}$
- odaberemo skalar $r^{(t)}$ iz funkcije gustine $f(r) \propto \pi(\mathbf{x}^{(t)} + r\mathbf{e}^{(t)})$
- izračunamo stanje $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + r\mathbf{e}^{(t)}$

Ovaj algoritam je veoma koristan kada ciljana raspodela ima više modusa. Glavna poteškoća ovog algoritma je što je u praksi teško odrediti $f(r)$.

5. MONTE KARLO METOD RAZMENE KOPIJA - PRIMENA NA PROBLEM UVIJANJA PROTEINA

5.1 UVIJANJE PROTEINA

Protein je najvažniji molekul u klasi bioloških makromolekula. Oni učestvuju u svim ćelijskim i među-ćelijskim procesima i obavljaju širok spektar funkcija unutar živih organizama. Proteini su u suštini polimeri amino-kiselina koje su poređane u linearne lance i spojene međusobno peptidnim vezama. Struktura proteina je određena redosledom amino-kiselina u polipeptidnom lancu i od nje direktno zavisi funkcija proteina. Sekvenca amino-kiselina u proteinu definisana je u genima i sadržana u genetskom kodu. Genetski kod određuju dvadeset "osnovnih" amino-kiselina.

Zbog specifičnog vezivanja lanaca amino-kiselina, proteini imaju četiri strukturna nivoa koji određuju njihov izgled u prostoru (konformaciju). U zavisnosti od količine unakrsnih povezivanja struktura proteina može biti: primarna, sekundarna, tercijarna i kvaternarna [27].

- Primarna struktura

  Ovo je najjednostavnija struktura sa minimalnim brojem unakrsnih povezivanja. Primarna struktura predstavlja redosled amino-kiselina u polipeptidnom lancu i određena je genom koji odgovara proteinu. Specifičan niz nukleotida u DNK je prepisan na iRNK, koji „čitaju“ ribozomi u procesu translacije - biosinteze proteina. Niz amino-kiselina je jedinstven za taj protein i definiše njegovu strukturu i funkciju.

- Sekundarna struktura

  Sekundarne strukture predstavljaju izgled polipeptidnih lanaca u prostoru. Sastoje se od dva ili više polipeptidnih lanaca. Obrazuje se uglavnom na bazi stvaranja vodoničnih veza između atoma koji se nalaze u sastavu peptidne veze i relativno blizu u polipeptidnom lancu. Najčešće sekundarne strukture su:

  - α-heliks: tip sekundarne strukture u kojem se deo polipeptidnog lanca uvija u spiralu, najčešće u desnu zavojnicu.
  - β-ravan: tip sekundarne strukture u kojem se dva polipeptidna lanca ili delovi istog polipeptidnog lanca svrstavaju paralelno jedan prema drugom, lanci mogu biti paralelni ili antiparalelni.

- Tercijarna struktura

  Polipeptidni lanac, koji u sebi već sadrži delove sa sekundarnom strukturom, sposoban je da se u celini izuvija u prostoru i zauzme položaj koji odgovara najstabilnijoj konformaciji, pri datim uslovima, koji se naziva tercijarna struktura. Ova struktura se formira kada se sekundarne strukture uvežu i formiraju trodimenzionalnu kompleksnu strukturu prvenstveno kroz hidrofobne interakcije, ali i vodonične veze, jonske interakcije i disulfidne veze. Tercijarna struktura zaokružuje sve nekovalentne interakcije koje ne razmatra sekundarna struktura, definiše sveukupno savijanje u proteinu i utiče na funkciju koju će protein obavljati.

- Kvaternarna struktura

  Kvaternarna struktura je nivo organizacije proteina koji je sačinjen od više polipeptida, tako dobijene strukture se nazivaju kompleksi ili agregati. Svaki polipeptid (sa uređenom tercijarnom strukturom) predstavlja jednu podjedinicu. Kvaternarna struktura je prostorni raspored podjedinica u složenoj celini.

Proces kojim se formiraju više strukture se zove uvijanje proteina i predstavlja jedan od najznačajnijih problema savremene biologije. Korektna trodimenziona struktura je esencijalna za funkciju, mada neki delovi funkcionalnih proteina mogu da ostanu neuvijeni. Prirodno stanje (*eng. native state*) proteina predstavlja oblik proteina koji poseduje najmanje slobodne energije [6] [36]. Posledica neuspešnog uvijanja u prirodnu strukturu su neaktivni proteini koji su obično toksični. Iako svaki jedinstveni polipeptid može imati više od jedne stabilne uvijene konformacije, svaka konformacija ima svoju biološku aktivnost i samo jedna konformacija se smatra aktivnom. Funkcija je povezana sa prirodnim stanjem proteina i kada se ova struktura prekine protein nije u stanju da izvršava svoju specifičnu funkciju. Međutim, postoje proteini koji nisu uvijeni u prirodno stanje (tzv. neuređeni proteini) koji imaju značajnu ulogu u ćelijskim procesima, skladištenju i obradi informacija, ali dovode i do nekih ozbiljnih kardiovaskularnih oboljenja, karcinoma i neurodegenerativnih bolesti.

Mehanizam uvijanja proteina nije u potpunosti razjašnjen. Uspešan metod za predviđanje strukture proteina bi imao dalekosežnu primenu i u drugim naučnim oblastima uključujući i

[6] Slobodna energija je energija oslobođena ili apsorbovana u reverzibilnom procesu (proces koji se može odvijati u oba smera preko istih međustanja) pri konstantnoj temperaturi i pritisku. Definisana je jednačinom $G = H - TS$, gde je H entalpija, S entropija, a T termodinamička temperatura.

genetiku i medicinu. Trenutne metode koje se koriste u labaratorijama su uglavnom skupe i zahtevaju dosta vremena kako bi se dobili rezultati. U današnjoj eri savremenih tehnologija nije prihvatljivo oslanjati se samo na tehnike predviđanja strukture, kao što su rendgenska kristalografija i nuklearna magnetna rezonanca, već se javlja potreba za otkrivanjem novog efektivnog i efikasnog algoritma za predviđanje strukture proteina. Međutim, čak i za pojednostavljen model proteina, koji će biti objašnjen u nastavku, ovaj problem se pokazao kao NP kompletan problem za čije rešavanje ne postoji algoritam u polinomijalnom vremenu [45]. U ovom radu će biti objašnjena Monte Karlo metoda razmene kopija, koja se veoma uspešno primenjuje na složene proteinske modele i druge probleme optimizacije, pomoću koje ćemo predvideti strukturu proteina pronalaženjem one sa najmanjom energijom koja predstavlja prirodno stanje.

5.2 MONTE KARLO METODA RAZMENE KOPIJA

5.2.1. UVOD

Monte Karlo razmena kopija, skraćeno MKRK (*eng. replica exchange Monte Carlo*), poznata još i kao metoda paralelnog kaljenja (*eng. Parallel tempering*), predstavlja metodu simulacije čiji je glavni cilj poboljšanje dinamičkih osobina Monte Karlo metode uzorkovanja u fizičkim sistemima, a koja se zasniva na posmatranju M odvojenih sistema (kopija) na različitim temperaturama. Ova Monte Karlo simulacija, koja koristi Metropolis-Hestingov algoritam za ažuriranje stohastičkih procesa, ocenjuje energiju sistema i prihvata ili odbacuje posmatranu konformaciju u zavisnosti od temperature sistema T. Kod sistema u kojima su uzorci u velikoj korelaciji posmatrane konformacije se uglavnom odbacuju, pa se za njih kaže da se kritično usporavaju [28].

Skup konformacija za datu sekvencu predstavlja skup svih mogućih strukturnih stanja bez raskidanja kovalentnih veza, pa je moguće odrediti finalnu konformaciju korišćenjem grube sile, tj. računanjem svake konformacije i pronalaženjem one sa najmanjom energijom. Kako broj konformacija eksponencijalno raste sa brojem amino-kiselina u sekvenci, ovu metodu je moguće koristiti samo na kraćim sekvencama. Stoga je potrebno uprostiti prostor pretraživanja na uštrb tačnosti, kako bi se uštedelo na vremenu potrebnom za izračunavanje. Pažljivim odabirom temperatura i broja kopija sistema mogu se poboljšati svojstva Monte Karlo simulacije.

Razumevanje mehanizma uvijanja proteina predstavlja jedna od najizazovnijih problema molekularne biologije. U ovom odeljku će biti predstavljena metoda Monte Karlo razmene kopija koja za generisanje konformacije sistema koristi metodu molekulske dinamike (skraćeno MD), kao i njenu primenu na problem uvijanja penta-peptida[7] Met-enkefalina u gasnom stanju.

[7] Polipeptid koji sadrži pet amino-kiselina

5.2.2 MOLEKULSKA DINAMIKA

Molekulska dinamika (MD) je oblik računarske simulacije, gde atomi i molekuli mogu da interaguju u određenom vremenskom intervalu, pokoravajući se poznatim zakonima fizike. Ovaj metod su prvi predstavili Alder i Vajnvrajt (*eng. Alder and Wainwright*) [29], nakon čega on postaje jedan od korišćenijih alata za istraživanje kompleksnih fizičkih sistema. Molekulska dinamika predstavlja determinističu proceduru za integrisanje jednačina kretanja (Hamiltonove jednačine) na osnovu klasičnih principa mehanike. Prvi korak ove simulacije je podešavanje kvantitativnog sistema (modela) pod datim uslovima (npr. tačan broj čestica ili ukupna energija sistema), nakon čega se na osnovu Njutnovih zakona kretanja generiše konfiguracija sistema, kao funkcija vremena. Podaci dobijeni ovom simulacijom predstavljaju snimke pozicija i brzine čestica na osnovu kojih se može odrediti "tipična karakteristika" sistema.

Glavni zadatak molekulske dinamike je integracija jednačina kretanja u datom vremenskom periodu, kao i proučavanje fizičkih i hemijskih osobina sistema u tom vremenskom periodu (npr. uticaj vode pri uvijanju proteina). Neka je $x(t)$ d-dimenzioni vektor pozicije čestice u vremenu t, $m = (m_1, m_2 \dots m_d)$ vektor mase, a $v(t) \equiv \frac{\partial x}{\partial t}$ brzina čestice, tada možemo definisati impuls p i kinetičku energiju $k(p)$ sistema kao:

$$p = mv = (m_1 v_1, \dots m_d v_d)$$

$$k(p) = \frac{1}{2}\sum_{i=1}^{d} m_i v_i^2 = \frac{1}{2}\sum_{i=1}^{d} \frac{p_i^2}{m_i} = \frac{1}{2}\left\| \frac{p}{\sqrt{m}} \right\|^2 .$$

Neka je $U(x)$ potencijalna energija sistema, tada je ukupna energija čestica sistema:

$$H(x,p) = U(x) + k(p),$$

a Hamiltonove jednačine su sledećeg oblika:

$$x(t)' = \frac{\partial H(x,p)}{\partial p}$$

$$p(t)' = -\frac{\partial H(x,p)}{\partial x}.$$

Kako kapacitet kompjutera omogućava samo diskretne operacije, što znači da nije moguće neprekidno izračunavanje jednačina kretanja čestica, u praksi se koristi aproksimacija Hamiltonovih jednačina u vidu Tejlorovog razvoja:

$$x(t+dt) = x(t) + \frac{p(t)}{m}dt + \frac{p(t)'}{2m}dt^2 + \cdots,$$

$$p(t+dt) = p(t) + p(t)'dt + \frac{p(t)''}{2}dt^2 + \cdots.$$

Jedan od najjednostavnijih i najčešće korišćenih algoritama za integraciju jednačina kretanja je Verletov algoritam [30], koji se zasniva na zapažanju da je:

$$x(t+dt) + x(t-dt) = 2x(t) + \frac{p(t)'}{m}dt^2 + O(dt^4),$$

$$x(t+dt) + x(t-dt) = 2\frac{p(t)}{m}dt + O(dt^3).$$

Za izabrani vremenski korak Δt, pozicija i impuls čestice imaju sledeće vrednosti:

$$x(t+\Delta t) = 2x(t) - x(t-\Delta t) - \frac{1}{m}\frac{\partial H}{\partial x}\Big|_t (\Delta t)^2$$

$$p(t+\Delta t) = m\frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t}.$$

Još jedna MD metoda koja je često u upotrebi je metoda preskakanja (*eng. leap frog*) [31]. Glavna karakteristika ove metode je to da se impuls izračunava na polovini vremenskog intervala, pa su jednačine sledećeg oblika:

$$x(t+\Delta t) = x(t) + \Delta t\frac{p(t+\frac{1}{2}\Delta t)}{m}$$

$$p\left(t+\frac{1}{2}\Delta t\right) = p\left(t-\frac{1}{2}\Delta t\right) + \frac{\partial H}{\partial x}\Big|_t \Delta t.$$

Molekulski sistemi se generalno sastoje od velikog broja čestica, stoga je nemoguće analitički pronaći osobine tako složenih sistema, ali molekulska dinamika prevazilazi ovaj problem koristeći numeričke metode. Dizajn MD simulacija određen je mogućim kapacitetom kompjutera. Veličine simulacije (N = broj čestica), vremenski korak i ukupno vreme trajanja simulacije treba odabrati tako da se proračun završi u nekom razumnom vremenskom periodu. Ipak, simulacija treba da bude dovoljno duga da bi verno predstavila prirodni proces koji proučavamo. Većina naučnih radova na temu dinamike proteina i DNK koriste simulaciju za procese koji u prirodi traju od nekoliko nanosekundi do mikrosekundi, a da bi se ove simulacije posmatranog procesa izvršile potreban je vremenski period od nekoliko dana do nekoliko godina. Glavna prednost MD simulacije u fizičkim sistemima je njena zasnovanost na osnovnim fizičkim principima (npr. Njutnovim zakonima), ali jedan od glavnih nedostataka je to što ona zahteva veoma mali

vremenski korak koji se određuje na osnovu dužine trajanja najbržeg pokreta tokom posmatranog procesa. Prema tome vremenski korak mora biti adekvatno izabran kako bi se posmatrani proces pravilno simulirao, jer previše veliki vremenski korak dovodi do veće greške koja se javlja pri izračunavanju jednačina kretanja, a suviše mali vremenski korak utiče na efikasnost simulacije jer zahteva veći broj izračunavanja. Tipičan vremenski korak u klasičnoj MD je reda veličine femtosekunde (*fs*), pa bi na primer, MD simulacija uvijanja proteina koja u prirodi traje oko 1 milisekunde, trajala oko 10^6 dana.

U standardnim Monte Karlo simulacijama Metropolis-tipa predložena raspodela se ne može jednostavno uklopiti u lokalnu dinamiku ciljane raspodele. Na primer, ukoliko se posmatrani sistem sastoji od gusto zbijenih čestica, tada će se izračunati pomeraj čestice uglavnom odbacivati jer će nova pozicija biti delimično zauzeta od strane drugih čestica. Kako bi se prevazišao ovaj problem Čarls Gajer (*eng. Charls Geyer*) [33] je predstavio metod MKRK koja kombinuje osnovnu ideju MD sa Metropolis pravilima prihvatanja radi određivanja uzoraka željene raspodele.

5.2.3 SOFTVERSKI PAKETI ZA SIMULACIJU MOLEKULSKE DINAMIKE

Molekulska dinamika nam omogućava izučavanje dinamike velikih makromolekula, uključujući i biološke sisteme kao što su protein, DNK, RNK, membrane i dr. Danas je u širokoj upotrebi u farmaceutskoj industriji pri izradi lekova radi testiranja osobina molekula bez potrebe njihove sinteze koja je jako skupa. Dinamički događaji mogu imati ključnu ulogu u kontrolnim procesima koji utiču na funkciju biomolekula. Postoji nekoliko softverskih paketa koji se koriste za simulaciju molekulske dinamike. Svaki od njih ima različite karakteristike i pravila upotrebe, a ovde ćemo predstaviti tri najpopularnija paketa: AMBER, CHARm i Gromacs [39].

AMBER (*eng. Assisted Model Building and Energy Refinement*) predstavlja uopšten naziv za skup programa koji korisnicima omogućava da sprovedu i analiziraju MD simulaciju, posebno za proteine, amino kiseline i karbohidrate. Ovi programi se mogu podeliti u tri grupe: programi za pripremu, programi za simulaciju i programi za analizu. Osnovni programi za pripremu su Antechamber i LEaP. Antechamber automatizuje proces razvoja deskriptora polja sile[8] za većinu organskih molekula, koji počinje sa određenom strukturom (obično u PDB[9] formatu) i generiše fajl koji se kasnije može koristiti u LEaP radi modeliranja molekula. Deskriptor polja sile je dizajniran tako da bude kompatibilan sa standardnim AMBER poljima sile za protein i amino kiseline. LEaP je program koji obezbeđuje osnovnu izgradnju modela i kreiranje AMBER koordinata i parametara ulaznog fajla. On u sebi sadrži editor koji omogućava izgradnju ostataka i manipulaciju molekula. Glavni program za simulaciju molekulske dinamike predstavlja Sander koji se takođe koristi i kod metode razmene kopija, termodinamičke integracije i dr. Ptraj je program koji pripada grupi programa za analizu i koristi se za analiziranje MD trajektorija, hidrogenih veza i sl. U *Tabeli 1* su predstavljene prednosti i mane AMBER programa za simulaciju [39].

[8] Polje sile se odnosi na parametre i jednačine koje se koriste za izračunavanje potencijalne energije sistema prilikom simulacije molekulske dinamike.
[9] Proteinska banka podataka, (eng. *Protein Data Bank*, PDB), je kolekcija 3D strukturnih podataka velikih bioloških molekula, kao što su proteini i nukleinske kiseline

Prednosti	Nedostaci
Obezbeđuje podršku za simuliranje karbohidrata, proteina i manjih organskih molekula.	Nije moguće simulirati samo jedan deo sistema, npr. samo aktivna strana enzima.
Računa slobodnu energiju koristeći termodinamičku integraciju ili kišobran tehniku uzorkovanja (*eng. umbrella sampling*)	Programske komponente nemaju korisnički interfejs.
Radi ubrzanja konvergencije moguće je koristiti metodu zamene kopija.	Nedostatak Monte Karlo uzorkovanja, dinamike torzionih uglova, izračunavanje "dualne topologije" slobodne energije.
Omogućava analizu trajektorija i fleksibilna ograničenja koja mogu biti zasnovana na podacima spektroskopije nuklearne magnetne resonance.	Kod je pisan od strane različitih autora tokom razvoja pa sadrži delove koji su teški za razumevanje i modifikaciju.
Ima veliku i aktivnu zajednicu korisnika, kao i tutorijale i uputstva za nove korisnike. Kod je prenosiv i moguće su dopune i modifikacije.	Korisnici moraju sami da kompaliraju program.

Tabela 1: Prednosti i nedostaci AMBER programa.

CHARMM je ime grupe široko korišćenih polja sila za molekulsku dinamiku, a isto tako i ime softverskog paketa za molekulsko dinamičke simulacije i analizu. CHARMM istraživački projekat uvrstava mrežu programera širom sveta koji rade na razvoju i održavanju programa. Licence za ovaj softver su dostupne besplatno individuama i grupama koji se bave akademskim istraživanjem. Accelrys distribuira komercijalnu CHARMM verziju, koja se zove CHARMm. CHARMM program omogućava izvođenje i analizu širokog kruga molekularnih simulacija. Najosnovnije vrste simulacija su minimizacija date strukture i računanje trajektorije molekulske dinamike. CHARMM je jedan od najstarijih programa molekulske dinamike i kao što je slučaj sa AMBER programa i ovaj program je pisan od strane različitih pojedinaca i grupa pa je kod težak za razumevanje i modifikaciju [40].

GROMACS (skraćeno od Groningen mašina za hemijske simulacije (*eng. GROningen MAchine for Chemical Simulations*)) je molekulsko dinamički simulacioni paket originalno razvijen na Groningenskom univerzitetu. On se u današnje vreme održava i proširuje i na drugim mestima, uključujući Upsalski univerzitet, Stokholmski univerzitet i Maks Plank institut za istraživanje polimera. GROMACS je softver otvorenog koda pod GNU generalnom javnom licencom. GROMACS projekat je originalno započet da bi se konstruisao namenski paralelni računarski sistem za molekularne simulacije, koji je baziran na prsten strukturi. Izvorni kod specifičan za molekulsku dinamiku je prerađen u C programskom jeziku iz Fortran77-baziranog programa GROMOS [41].

Program je napisan za Unix operativne sisteme, ali se on može koristiti na Microsoft Windows mašinama koristeći Cigvin (engl. *Cygwin*) Unix sloj. GROMACS sadrži skript za konvertovanje molekulskih koordinata iz PDB fajla u formate koje program interno koristi. Nakon što je konfiguracioni fajl za simulaciju nekoliko molekula (po mogućnosti uključujući rastvarač) kreiran, izvršavanje simulacija proizvodi fajl sa trajektorijama koji opisuje kretanje atoma u toku vremena. Taj trajektorijski fajl se može analizirati ili prikazati brojnim alatima.

Mnogi specifični elementi su dodati u toku tranzicije GROMOS-a u GROMACS. Najznačajniji među njima su:

- Generička reprezentacija svih mogućih tipova periodičnih kutija
- Optimizovano rukovanje listom suseda putem smeštanja translacionih vektora ka najbližim susedima u periodičnom sistemu
- Specijalizovane rutine za računanje inverznog kvadratnog korena
- Korišćenje kubne splajn interpolacije iz tabeliranih vrednosti za evoluiranje sile/energije
- Brza na-rešetki-zasnovana pretraga suseda

Visoko optimizovani kod čini GROMACS jednim of najbržih programa za molekulske simulacije. Dodatno, podrška za različita polja sila daje GROMACS-u veliku fleksibilnost.

5.2.4 ALGORITAM MONTE KARLO RAZMENA KOPIJA

Posmatrajmo sistem od N atoma sa zadatom masom m_k $(k = 1, \dots, N)$ i pridruženim koordinatnim vektorom $q = (q_1, \dots, q_N)$ i impulsom $p = (p_1, \dots, p_N)$. Ukupna energija sistema tj. Hamiltonijan je tada jednak:

$$H(q,p) = U(q) + k(p),$$

gde je kinetička energija:

$$k(p) = \sum_{k=1}^{N} \frac{p_k^2}{2m_k}.$$

U kanonskom ansamblu[10] na temperaturi T, svako stanje $x = (q, p)$ sa Hamiltonijanom H(q, p) meri se Bolcmanovim faktorom:

$$W_b(x;T) = e^{-\beta H(q,p)},$$

gde je $\beta = \frac{1}{k_B T}$, a k_B Bolcmanova konstanta. Prosečna kinetička energija na temperaturi T je tada data kao:

$$\langle k(p) \rangle_T = \langle \sum_{k=1}^{N} \frac{p_k^2}{2m_k} \rangle_T = \frac{3}{2} N k_B T.$$

Algoritam Monte Karlo metoda razmena kopija pretražuje opšti ansambl[11] od M neinteragujućih[12] kopija ili replika koje predstavljaju potencijalno rešenje problema. Svakoj od ovih kopija se pridružuje jedinstvena temperaturna vrednost T_m $(m = 1, \dots, M)$. Na taj način imamo 1-1 korespodenciju između kopija i temperature. Oznaka $i (i = 1, \dots, M)$ za kopije je permutacija oznake $m (m = 1, \dots, M)$ za temperaturu i obrnuto:

$$\begin{cases} i = i(m) & \equiv f(m) \\ m = m(i) & \equiv f^{-1}(i), \end{cases}$$

gde je $f(m)$ funkcija permutacije od m, a $f^{-1}(i)$ njen inverz.

Neka X = $\left(x_1^{[i(1)]}, \dots, x_M^{[i(M)]}\right) = \left(x_{m(1)}^{[1]}, \dots, x_{m(M)}^{[M]}\right)$ predstavlja stanje sistema, gde je $x_m^{[i]} = (q^{[i]}, p^{[i]})_m$, stanje i-te kopije na temperaturi T_m. Pošto su replike neinteragujuće,

[10] Kanonski ansambl u statističkoj fizici predstavlja skup mogućih stanja sistema koja se nalaze u termodinamičkoj ravnoteži sa okolinom. U kanonskom ansamblu količina supstance N, zapremina V i temperatura T su konstantni.
[11] Opšti ansambl predstavlja kombinaciju statističkih ansambla, a u ovom slučaju je to kombinacija kanonskih ansambla
[12] Kod neinteragujućih sistema ukupna energija sistema je jednaka sumi energija njenih komponenti.

težinski faktor za stanje X u ovom sistemu predstavlja proizvod Bolcmanovih faktora za svaku repliku (ili na svakoj temperaturi):

$$W_{MKRK} = exp\left\{-\sum_{i=1}^{M} \beta_{m(i)} H(q^{[i]}, p^{[i]})\right\} = exp\left\{-\sum_{m=1}^{M} \beta_m H(q^{[i(m)]}, p^{[i(m)]})\right\}.$$

Pretpostavimo da razmenjujemo par kopija *i* i *j* na temperaturama T_m i T_n, tim redom:

$$X = \left(\dots, x_m^{[i]}, \dots, x_n^{[j]}, \dots\right) \rightarrow X' = (\dots, x_m^{[j]'}, \dots, x_n^{[i]'}, \dots).$$

U ovom zapisu je korišćena još jedna funkcija permutacije:

$$\begin{cases} i = f(m) \rightarrow j = f'(m) \\ j = f(n) \rightarrow i = f'(n), \end{cases}$$

pa se proces razmeme kopija može zapisati kao:

$$\begin{cases} x_m^{[i]} = (q^{[i]}, p^{[i]})_m \rightarrow x_m^{[j]'} = (q^{[j]}, p^{[j]'})_m \\ x_n^{[j]} = (q^{[j]}, p^{[j]})_n \rightarrow x_n^{[i]'} = (q^{[i]}, p^{[i]'})_n \end{cases},$$

gde se za $p^{[j]'}$ i $p^{[i]'}$ mogu uzeti sledeće vrednosti, za koje se smatra da predstavljaju najjednostavniji i najprirodniji izbor:

$$p^{[i]'} = \sqrt{\frac{T_n}{T_m}} p^{[i]},$$

$$p^{[j]'} = \sqrt{\frac{T_m}{T_n}} p^{[j]}.$$

Uočavamo da je ovaj proces ekvivalentan zameni para temperatura T_n i T_m za odgovarajuće replike *i* i *j*, kao što sledi:

$$\begin{cases} x_m^{[i]} = (q^{[i]}, p^{[i]})_m \rightarrow x_n^{[i]'} = (q^{[i]}, p^{[i]'})_n \\ x_n^{[j]} = (q^{[j]}, p^{[j]})_n \rightarrow x_m^{[j]'} = (q^{[j]}, p^{[j]'})_m \end{cases}.$$

Da bi proces zamene kopija konvergirao uravnoteženoj raspodeli dovoljno je nametnuti uslov detaljne uravnoteženosti verovatnoće prelaza $w(X \rightarrow X')$:

$$W_{MKRK}(X) w(X \rightarrow X') = W_{MKRK}(X') w(X' \rightarrow X).$$

Predložena razmena kopija se prihvata ili odbacuje u zavisnosti od Metropolisovog kriterijuma:

$$w(X \rightarrow X') = w\left(x_m^{[i]} \middle| x_n^{[j]}\right) = \begin{cases} 1, & \Delta \leq 0, \\ \exp(-\Delta) & \Delta > 0 \end{cases}$$

gde je:

$$\Delta = [\,\beta_n - \beta_m]\left(U\left(q^{[i]}\right) + U\left(q^{[j]}\right)\right).$$

Simulacija MKRK se tada realizuje naizmeničnim izvođenjem sledeća dva koraka:

(1) Svaka replika u kanonskom ansamblu na fiksnoj temperaturi se simulira simultano i nezavisno za određeni MD korak.

(2) Par replika na susednim temperaturama, recimo $x_m^{[i]}$ i $x_{m+1}^{[j]}$, se zamenjuje sa verovatnoćom $w(x_m^{[i]}|x_{m+1}^{[j]})$ u jednačini Metropolisovog kriterijuma, prema tome algoritam MKRK je oblika:

Function: Replica-Exchange(cycles c, replicas n, steps m)
for *c cycles* **do**
  for $a = 0,\ldots, n$ **do**
    perform m steps of MD
  end
  for *neighboring pairs of replicas i, i+1* **do**
    choose random z $\in (0, 1)$
    $P_{acc} = min[1, e^{-\Delta}]$
    if $z < P_{acc}$ **then**
      exchange replicas i and i+1
    end
  end
end

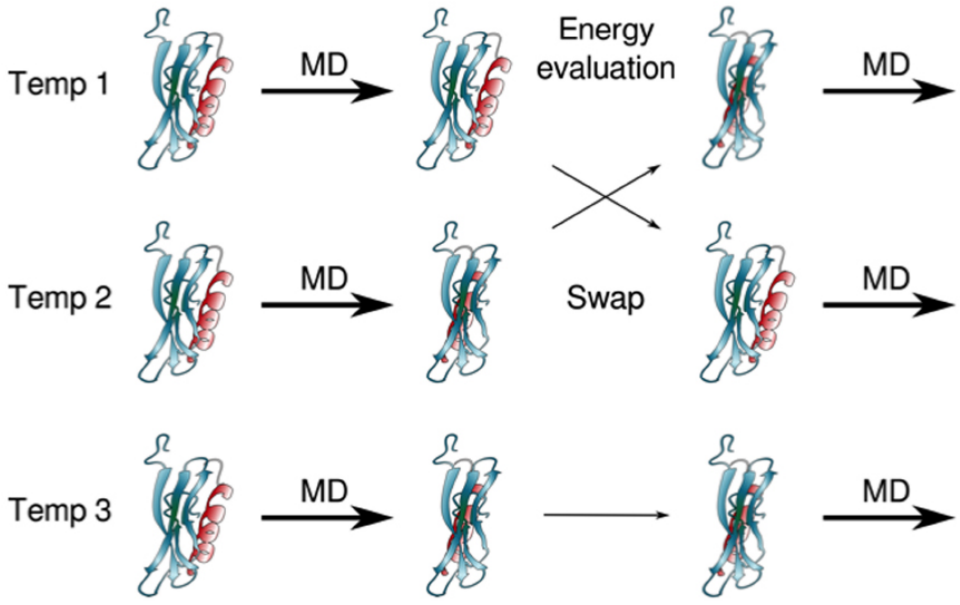
U ovom pristupu koristimo MD algoritam u prvom koraku, dok se u drugom koraku zamenjuju samo replike koje odgovaraju susednim temperaturama, zato što se opseg prihvatljivosti zamene (*eng. acceptance ratio*) umanjuje eksponencijalno sa razlikom dveju β.

Glavna prednost MKRK u odnosu na druge metode jeste to da je težinski faktor apriori poznat, dok u drugim algoritmima određivanje težinskih faktora može biti prilično monotono i dugotrajno. Ipak, za optimalne performanse MKRK potrebno je izabrati prikladnu temperaturnu raspodelu.

5.3 REZULTATI PRIMENE MKRK NA PROBLEM UVIJANJE PROTEINA

Efikasnost algoritma predstavljenog u prethodnom odeljku testirana je na sistemu pentapeptida Met-enkefalina u gasnom stanju, odnosno na problem njegovog uvijanja [38]. Ovaj peptid ima aminokiselinsku sekvencu Tyr-Gly-Gly-Phe-Met. Izvršena je MD simulacija od $N_{sim} = 10^5$ vremenskih koraka za svaku repliku (kopiju peptida), gde je jedinica vremenskog koraka $10fs$. Pri simulaciji korišćene su sledeće temperature (M = 8): 700, 585, 489, 409, 342, 286, 239 i 200K [37].

Iz opisa MKRK algoritma u prethodnom odeljku očigledno je da je simulacija razmene kopija posebno pogodna za paralelno izvršavanje na M računara. Pošto se može smanjiti količina razmenjenih informacija među čvorovima, najbolje je dodeliti jednu repliku jednom čvoru (zamenjivanje parova temperaturnih vrednosti među čvorovima je mnogo brže od zamenjivanja koordinata i impulsa). Posle svakih $10fs$ simulacije MD koraka, iteriramo kroz sve susedne replike pokušavajući da ih zamenimo ukoliko je zadovoljen Metropolisov kriterijum.



Slika 5: Šematski prikaz razmene kopija na susednim temperaturama. Između svake zamene se izvršava određen broj MD koraka

Da bismo ispitali da li se zamene replika izvršavaju na pravi način potrebno je da proverimo tri stavke. (a) Da li su temperature optimalno raspoređene? (b) Da li je broj replika (temperatura) dovoljan? (c) Da li je najviša temperatura dovoljno visoka da bi se izbeglo zaglavljivanje u stanju

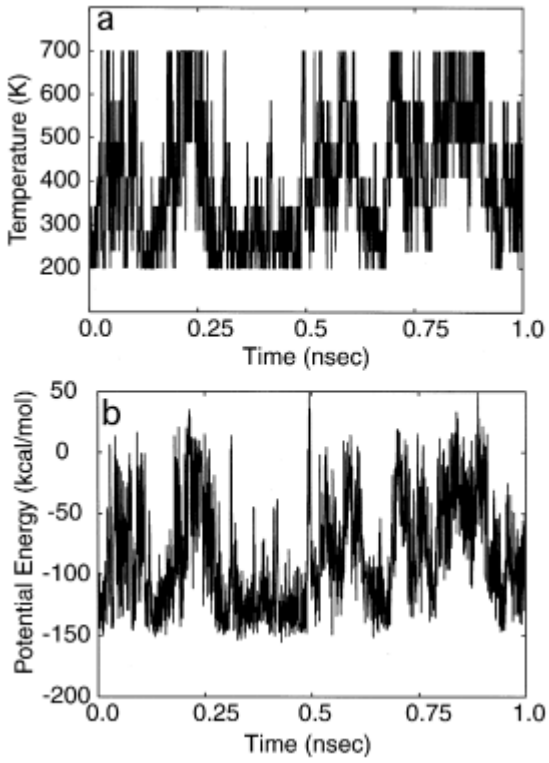
lokalnog minimuma energije? Prve dve stavke se mogu proveriti ispitivanjem opsega prihvatljivosti zamene replika koje odgovaraju susednim parovima temperatura. Što se tiče prve stavke, optimalne raspodele temperature impliciraju da su svi opsezi prihvatljivosti isti, što rezultira slobodnim prelazom u prostoru replike (temperature). Što se druge stavke tiče, broj replika (temperatura) je dovoljan ukoliko opsezi prihvatljivosti nisu isuviše mali (recimo veći od 0.1). U *Tabeli 1* su predstavljeni opsezi prihvatljivosti zamena replika, čije vrednosti su jedinstvene (oko 15% verovatnoće prihvatanja) i dovoljno veliki (>10%). Stoga, ispunjena su dva od tri gore opisana kriterijuma ((a) i (b)) za optimalni učinak.

Treću stavku ipak nije tako jednostavno proveriti kao što je to slučaj sa prve dve. Posmatrani nasumični prelazi među replikama (i temperaturama) nisu dovoljni za treću stavku. To je zato što ne možemo isključiti sledeću mogućnost. Ako se dogodi da su sve replike u istom stanju lokalnog minimuma energije i da je granica za izlazak iz ovog stanja veoma visoka (uzimajući u obzir najvišu temperaturu), onda ćemo i dalje posmatrati nasumične prelaze među svim replikama (i temperaturama) ali će oni ostati u istom lokalnom minimumu. Potkrepljujući dokaz za ispunjenost treće stavke se može dobiti upoređivanjem rezultata sa onima dobijenim iz regularne kanonske simulacije, što se razmatra u nastavku.

Parovi temperatura	Opseg prihvatljivosti
200 ↔ 239 K	0.160
239 ↔ 286 K	0.149
286 ↔ 342 K	0.143
342 ↔ 409 K	0.139
409 ↔ 489 K	0.142
489 ↔ 585 K	0.146
585 ↔ 700 K	0.146

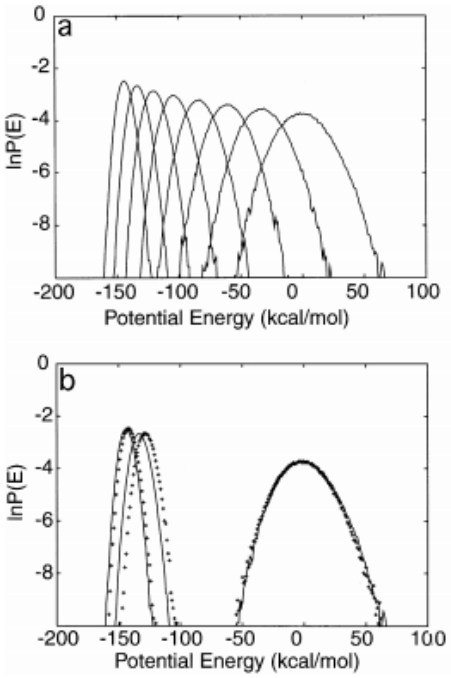
Table 1: Opsezi prihvatljivosti zamene replika koji odgovaraju parovima susednih temperatura

Rezultati iz *Tabele 1* impliciraju da za svaku repliku postoji mogućnost nasumičnog prelaz na niže ili više temperature. Rezultati razmene temperatura i ukupna potencijalna energija za jednu od replika (replika 6) su prikazani na *Slici 6*. Na *Slici 6b* prikazan je grafikon ukupne potencijalne energije tokom posmatranog vremenskog perioda od 1 nanosekunde, pa mozemo primetiti da je ostavaren nasumični prelaz u prostoru potencijalne energije između najnižih i najviših energija.



Slika 6: Grafičiki prikaz zamene temperature (a) i ukupne potencijalne energije (b) za jednu od replika.

Na *Slici 7* su prikazane kanonske raspodele verovatnoće dobijene na osam izabranih temperatura iz simulacije zamene replika. Primetimo da postoji dovoljno preklapanja između svih susednih parova raspodela, što ukazuje na to da će biti dovoljan broj zamene replika između parova. Na *Slici 7b* upoređujemo kanonske raspodele verovatnoće na tri temperature (T=200, 239 i 700k), dobijene MD simulacijom zamene replika, sa onima dobijenim regularnim kanonskim MD simulacijama, izvedenim odvojeno na odgovarajućim temperaturama. Kanonske simulacije su izvedene sa istim početnim uslovima i simulacionim vremenom kao i simulacija zamene replika. Uočavamo očekivano ponašanje da se raspodele slažu na višim temperaturama, a da su sklone odstupanju na nižim temperaturama. Činjenica da su raspodele dobijene regularnim kanonskim simulacijama na niskim temperaturama sklone prebacivanju na desno u odnosu na one dobijene zamenom replika ukazuje na to da su se kanonske simulacije zaglavile u stanjima lokalnog minimuma energije na tim temperaturama.

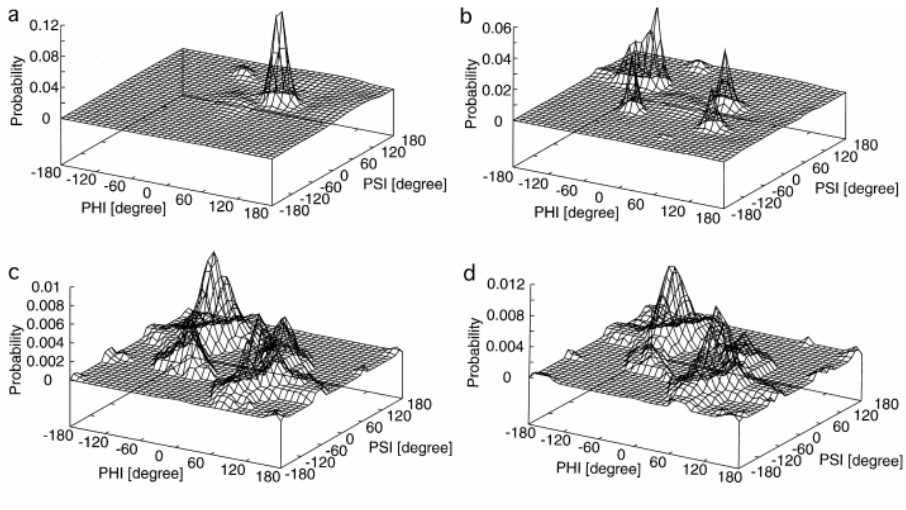


Slika 7: Kanonske raspodele verovatnoće ukupne potencijalne energije Met-enkefalina dobijene iz MD simulacije zamene replika na 8 temperatura (a) i poređenje kanonskih raspodela verovatnoće dobijenih iz MD simulacije zamene replika (pune linije) i konvencionalne kanonske MD simulacije (ukrštene) na tri temperatura (b). Raspodele u (a) odgovaraju sledećim temperaturama (s leva na desno): 200, 239, 286, 342, 409, 489, 585 i 700 K. Parovi raspodela u (b) odgovaraju sledećim temperaturama (s leva na desno): 200, 239 i 700 K.

U nastavku ćemo uporediti rezultate simulacije zamene replika sa onim dobijenim iz jedne kanonske MD simulacije na odgovarajućim temperaturama. Na *Slici 8* upoređujemo raspodele para sa diedarskim uglovima (φ,ψ)[13] Gly-2 (Gly-Gly) segmenta na dve ekstremne temperature (T = 200 i 700K). Dok se rezultati na T = 200K iz regularne kanonske simulacije lokalizuju sa samo jednim dominantnim vrhom, oni iz simulacije zamene replika imaju nekoliko vrhova (što se i vidi na slikama 8a i 8b). Stoga, simulacija zamene replika uzorkuje daleko širi konfiguracioni prostor u odnosu na konvencionalnu kanonsku simulaciju na niskim temperaturama. Uočimo da set vrhova posmatranih u raspodeli iz simulacije zamene replika uključuju one iz kanonske simulacije kao podgrupu. Međutim, vrh iz kanonske simulacije nije najviši u simulaciji zamene replika što ukazuje na to da se kanonska simulacija nije završila u prirodnom stanju, već je zapela u jednom od stanja lokalnog minimuma energije. Prosečna potencijalna energija na 200K konformacije koja

[13] Diedarski uglovi su torzioni uglovi polipeptidnog lanca koji opisuju rotaciju kičme polipeptida oko veze N-Cα (φ) i Cα-C (ψ)

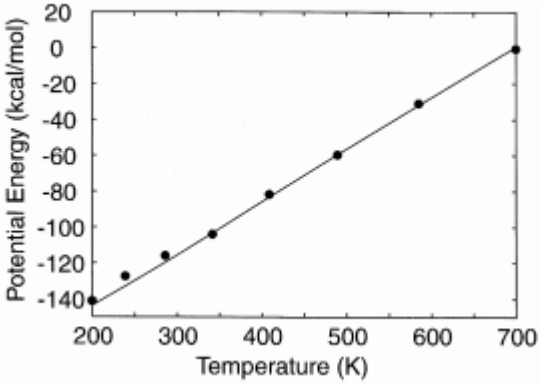
odgovara najvišem vrhu u raspodeli kanonske simulacije *(Slika 8a)* je za oko 2 kcal/mol viša od ove iz simulacije zamene replika (*Slika 8b)*(-141 u odnosu na –143 kcal/mol). Sa druge strane, rezultati na T = 700K su slični, što ukazuje na to da regularna kanonska simulacija može dati precizne termodinamičke veličine na visokim temperaturama.



Slika 8: Raspodele para diedarskih uglova (φ,ψ) Gly-2 za: (a) T=200K iz regularne kanonske MD simulacije, (b) T = 200K iz MD simulacije zamene replika, (c) T = 700K iz regularne kanonske MD simulacije, (d) T = 700 K iz MD simulacije zamene replika.

Činjenica da raspodela dobijena iz simulacije zamene replika ima nekoliko vrhova čak i na niskim temperaturama daje delimičnu potporu da je gore pomenuti treći kriterijum za optimalni učinak simulacije zamene replika ispunjen. Naime, najviša temperatura je dovoljno visoka da se uzorkuje široki konformacioni prostor i da raspodela nije prinuđena da konvergira ka samoj konformaciji čak i na niskim temperaturama, gde regularne kanonske simulacije zapadaju u stanje lokalnog minimuma energije.

Na *Slici 9* prikazana je prosečna ukupna potencijalna energija kao funkcija temperature. Kao što je i očekivano na osnovu rezultata sa *Slike 7* i *8*, uočavamo da su se kanonske simulacije na niskim temperaturama zaglavile u stanjima energije lokalnog minimuma, što je rezultovalo odstupanjima u prosečnim vrednostima između rezultata iz kanonskih simulacija i onih iz simulacije zamene replika. Moze se uočiti da kanonske simulacije počinju da zapinju već negde oko 300 K (i niže), što je eksperimentalno relevantna temperatura.



Slika 9: Prosečna ukupna potencijalna energija u funkciji temperature. Puna linija je rezultat iz MD simulacije zamene replika a tačke su one iz regularnih kanonskih MD simulacija na osam temperatura.

Ovo ukazuje na to da su potrebne dosta duže simulacije da bi se dobili termodinamički proseci na ovim temperaturama koristeći konvencionalne MD metode zasnovane na kanonskom ansamblu. Kao što je i očekivano, imamo totalno slaganje na višim temperaturama između rezultata iz kanonskih simulacija i onih iz simulacije zamene replika.

GROMACS softver u sebi sadrži podršku za MKRK simulaciju. Predstavićemo sada korake koji su potrebni pri izvršavanju ove simulacije [43]:

- definisati sistem, npr. peptid + rastvor
- u zavisnosti od broja procesora i opsega iz kojeg se mogu odabrati temperaturne vrednosti, odrediti raspodelu temperatura. Može se koristiti eksponencijalna raspodela: $T_i = T_0 * k$, gde se T_0 i k mogu podesiti tako da se dobiju razumni temperaturni intervali. Eksponencijalnost omogućava povećanje temperaturnih intervala sa povećanjem temperature, što je neophodno zbog raspodele ukupne energije, koja se povećava zajedno sa temperaturnom kao i stopa razmene (*eng. exchange rate*). Stopu razmene treba održavati konstantnom kroz sve temperaturne vrednosti. (Za odabir temperatura na osnovu T_{min}, T_{max} i broja replika se može koristiti i http://folding.bmc.uu.se/remd/)
- uravnotežiti sisteme na N odabranih temperatura odvojeno koristeći .mdp[14] fajlove i generisati N ulaznih tpr. fajlova.

[14] .mdp fajl (*eng. Molecular Dynamics Parameter file*) sadrži sve informacije potrebne za simulaciju MD kao npr. broj kopija, vremenski korak, ukupan broj koraka i sl.

- pokrenuti kratku MKRK simulaciju kako bi se estimirala stopa razmene i modifikovale temperaturne vrednosti ukoliko nije postignuta zadovoljavajuća stopa (obično se prihvatljiva stopa razmene kreće između 0,2 i 0,3). Takođe treba obratiti pažnju i na vreme koje jedna replika provede na određenoj temperaturi kao i način na koji se vrši razmena, da li vršiti razmenu svih parova ili samo jednog nasumično odabranog para.
- odrediti početne konfiguracije za svaku temperaturu. Ove konfiguracije mogu biti iste ili različite, što zavisi od razloga zašto izvršavamo MKRK simulaciju.

Primer: Posmatrajmo 8 kopija (konformacija nekog proteina) na temperaturama u opsegu od 200K do 700K i odgovarajućim .mdp fajlovima. Svi potrebni parametri su smešteni u .mdp fajlovima i svi su jednaki za svaku od kopija. Ulazni .tpr fajlovi se prave pokretanjem **gmx grompp** komande nad svakim od .mdp fajlova, nakon čega je simulacija spremna za izvršavanje pokretanjem komande:

~$ mpiexec -x -np 8 gmx mdrun -s sd_.tpr -multi 8 -replex 1000 -reseed 175320, gde je:

np – broj procesora

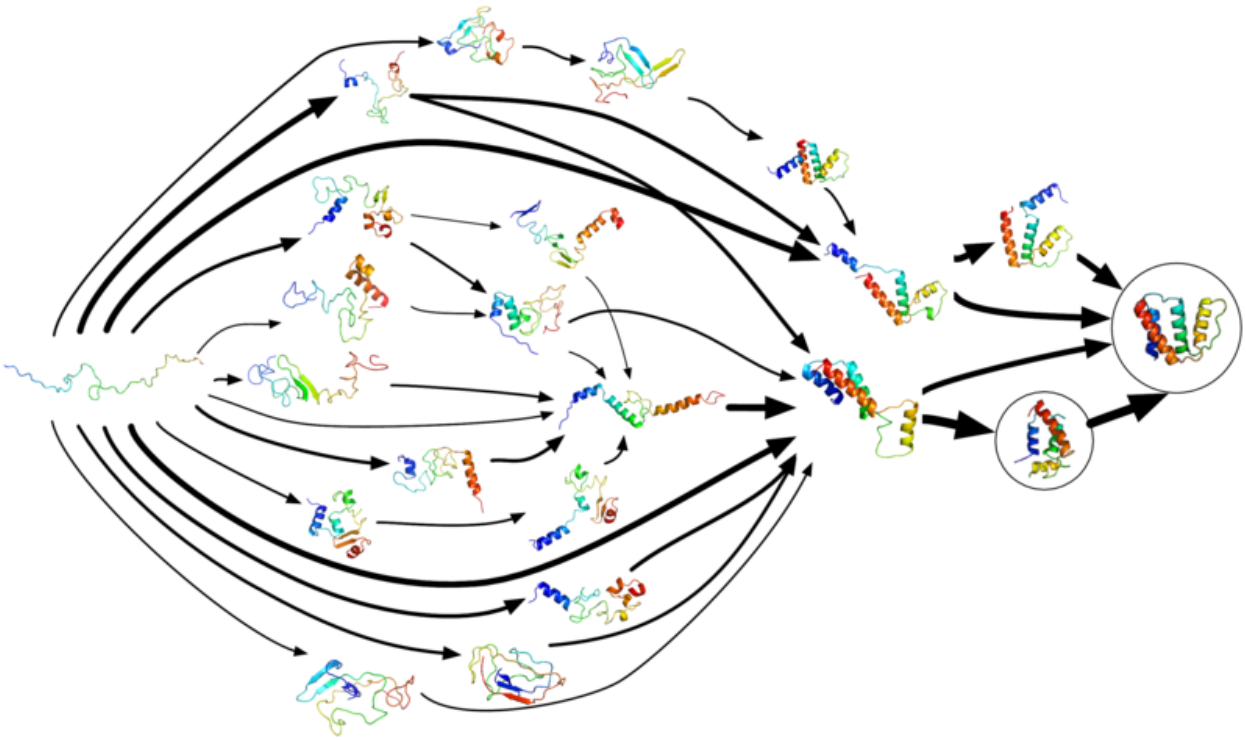
multi – instrukcija programu da se izvrši 8 puta

replex – instrukcija da sistem pokuša razmenu posle svakih 1000 koraka tj. Svake 2*ps* (pikosekunda) ukoliko je vremenski korak jednak 2*fs*

reseed – broj za inicijalizaciju generatora slučajnih brojeva.

Log fajl dobijen nakon izvršavanja gore navedene komande sadrži statistiku simulacije, kao na primer verovatnoću razmene i replike koje su učestvovale u razmeni nakon 1000 koraka, na osnovu kojeg je moguće iscrtati odgovarajuće trajektorije.

GROMACS predstavlja jedan od glavnih softverskih paketa koji se koristi u jednom od najvećih istraživačkih projekta za simulaciju uvijanja proteina – Folding@home. Ideja ovog distribuiranog računarskog projekta je da se obrada ogromnog broja podataka podeli na relativno male delove, koji se distribuiraju putem Interneta do personalnih računara širom sveta, gde se vrši obradi i potom rezultati vraćaju pokretaču projekta. Folding@home koristi skriveni Markovljev model za modeliranje mogućih oblika i putanja uvijanja proteina, kao što je predstavljeno na *Slici 10* [44].



Slika 10: Dijagram skrivenog Markovljevog modela

Simulacija uvijanja proteina koje ja izvršena pomoću Folding@home-a se može pogledati na sledećem linku: https://www.youtube.com/watch?v=gFcp2Xpd29I dok se primer simulacije razmene kopija može videti na linku https://www.youtube.com/watch?v=LPmt9nRtJ5Y&t=52s. Ova simulacija se sastoji od 8 kopija na temperaturama u intervalu od 100K do 500K.

6. ZAKLJUČAK

Monte Karlo metode su veoma široka oblast matematike. One nam putem simulacija i korišćenjem slučajnih brojeva daju dosta dobre aproksimacije nekih veoma teških problema. U ovom radu smo predstavili neke od najpoznatijih i najkorišćenijih Monte Karlo metoda, kao i primenu na problem uvijanja proteina, mada treba naglasiti da ove metode imaju veoma značajnu ulogu u rešavanju problema i u drugim oblastima i naukama. Prednost Monte Karlo metode, za razliku od molekulske dinamike, je to što ona nije ograničena Njutnovim jednačinama kretanja, pa poseduje veću slobodu pri predlaganju narednih pokreta radi generisanja novih konformacija. Različiti pokreti se mogu kombinovati radi postizanja veće fleksibilnosti simulacija koje se mogu na jednostavan način paralelno izvršavati na više računara. Monte Karlo simulacije ne pokazuju samo šta će se dogoditi već i koliko je verovatan svaki od tih ishoda, a pored toga obezbeđuju i grafički prikaz radi lakse analize i tumačenja dobijenih rezultata.

Nedostatak Monte Karlo metode je to što ona zahteva generisanje velikog broja uzoraka, što iziskuje dosta vremena i resursa. Takođe, je potrebno generisati sve uslove i ograničenja relevantna za rešavanje posmatranog problema, a pošto se proces zasniva na pokušajima i pogreškama simulacija ne predstavlja uvek optimalno rešenje. Pošto Monte Karlo metode ne rešavaju Njutnove jednačine kretanja one zato ne obezbeđuju nikakve dinamičke informacije. Jedna od glavnih problema Monte Karlo simulacije proteina u eksplicitnom rastvaraču jesu veliki koraci koji značajno menjaju unutrašnje koordinate proteina bez pomeranja rastvarača, što u velikom broju slučajeva dovodi do preklapanja atoma, a samim tim i odbacivanja posmatrane konformacije. Simulacija proteina u implicitnom rastvaraču[15] nema ovaj problem, pa je za nju pogodnije koristiti Monte Karlo metode. Takođe, ne postoji opšti program koji se koristi za MK simulaciju protenina zbog toga što odabir pokreta koji će se koristiti i njihova stopa prihvatanja varira u zavisnosti od problema koji rešavamo. Nedavno je Monte Karlo modul dodat u CHARMM softver za simulaciju [42].

Monte Karlo metode nastavljaju da budu jedne od najkorisnijih prisupa za naučna istraživanja zbog svoje jednostavnosti i opšte primenljivosti. Zbog konstantnog razvoja, sledeća generacija MK tehnika će obezbediti značajne alate za rešavanje sve složenijih problema procene

[15] Implicitni rastvarač predstavlja neprekidni medijum (suprtno eksplicitnom rastvaraču koji predstavlja skup pojedinačnih molekula rastvarača) koji se najčešće koristi u simulaciji molekulske dinamike

očekivanja i optimizacije u različitim naučnim oblastima kao što su: finansije, statistika, matematika, biologija i dr.

LITERATURA

[1] Dörrie, H., *100 Great Problems of Elementary Mathematics;Their History and Solution,* Dover Publications, New York, 1965

[2] Eckhardt, Roger, *Stan Ulam, John von Neumann, and the Monte Carlo method*, Los Alamos Science*,* 1987

[3] Radford M. Neal, *Probabilistic Infetence Using Markov Chain Monte Carlo Methods,* University of Toronto, 1993

[4] Marković J. *Lanci Markova sa diskretnim vremenom i promenom,* Beograd, 2014

[5] N.Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, *Equations of State Calculations by Fast Computing Machines,* Journal of Chemical Physics*, 1953*

[6] W. L. Winston, *Operations research, Applications and algorithms*, Thomson learning Brooks/Cole, 2004.

[7] S. Lipschutz, *Theory and problems of Probability,* Schaum's Outline series, McGraw-Hill Book CompanyNew York ,1968.

[8] Jun S. Liu, *Monte Carlo Strategie In Scientific Computing,* Harvard University, 2001

[9] D.J.C Mackay, *Introduction to Monte Carlo methods,* Cambridge University

[10] W.R. Gilks, S. Richardson, D.J. Spiegelhalter, *Marcov Chain Monte Carlo In Practice,* Chapman & Hall/CRC, 1996

[11] C. Robert, G. Cassella, *Introducing Monte Carlo Methods with R,* Softcover, 2010

[12] C. Andrieu, N. De Freitas, A. Doucet, M.I. Jordan, *An Introduction to MCMC for Machine Learning,* Kluwer Academic Publishers, 2003

[13] Charles J. Geyer, *Markov Chain Monte Carlo Lecture Notes,* 2005Bakhtiyar Uddin,*Gibbs sampling*

[14] David P. Landau, Kurt Binder, *Monte-Carlo SImulation in Statistical Physics,* Cambridge University,2009

[15] *https://theclevermachine.wordpress.com/*

[16] Wendy L. Martinez, Angel R. Martinez, *Computational Statistics Handbook with MATLAB*, 2005

[17] W. K. Hastings, *Monte Carlo sampling methods using Markov chains and their applications,* Bioinformatika, 1970

[18] A. A. Barker, *Monte Carlo calculations of radial distribution functions for a proton-electron plasma,* Australian Journal of Physics, 1965

[19] L. Tierney, *Markov chains for exploring posterior distributions,* Annals of Statistics, 1994

[20] J. I. Siepmann, D. Frenkel, *Configurational bias Monte Carlo: A new sampling sheme for flexible chains,* Molecular Physics, 1992

[21] M. D. Ceperley, *Path integrals in the theory of condensed helium,* Review of Modern Physics, 1995

[22] S. German, D. German, *Stochastic relaxations, Gibbs distribution and the Bayesian restoration of images,* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1984

[23] J. S. Liu, W. H. Wong, A. Kong, *Convariance structure of the Gibbs sampler with applications to the comparisons of estimations and augmentation shemes,* Biometrika, 1994

[24] A. E. Gelfand, A. F. M. Smith, *Sampling-based approaches to calculating marginal densities,* Journal ot the American Statistical Association, 1990.

[25] G. Cassella, E. I. George, *Explaining the Gibbs sampler,* American Statistician, 1992

[26] P. Damien, J. Wakefield, S. Walker, *Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables,* Journal of the Royal Statistical Society, Series B, 1999

[27] http://www.wikiwand.com/Proteinska_struktura

[28] R. H. Swendsen, J. S. Wang, *Replica Monte Carlo simulation of spin glasses,* Physical Review Letters, 1986

[29] B. Alder, T. Wainwright, *Studies in molecular dynamics I. General method*, Journal of Chemical Physics, 1959

[30] L. Verlet, *Computer "experiments" on classical fluids. I. Thermodynamical properties of lennard-jones molecules*, Physical Review, 1967

[31] R. W. Hockey, *The potential calculation and some applications,* Methods in Computational Physics, 1970

[32] V. I. Arnold, *Mathematical Methods of Classical Mechanics*, Springer-Verlag, New York, 1989

[33] C. J. Geyer, *Markov chain Monte Carlo maximum likelihood,* Computing Science and Statistics: The 23rd symposium on the interface, Interface Foundation, Fairfax, 1991

[34] S. Kumar, D. Bouzida, R. H. Swendsen, P. A. Kollman, J. M. Rosenberg, *The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method,* Journal of Computational Chemistry 13, 1992

[35] S. J. Weiner, P.A. Kollman, D. T. Nguyen, D. A. Case, *An all atom force field for simulations of proteins and nucleic acids,* Journal of Computational 7, 1986

[36] C. B. Anfinsen, *Principles that govern the folding of protein chains*, Science, New Series, 1973

[37] Y. Okamoto, M. Fukugita, T. Nakazawa, H. Kawai, *α-Helix folding by Monte Carlo simulated annealing in isolated C-peptide of ribonuclease A*, Protein Engineering, Design and Selection, 1991

[38] Y. Okamoto, Y. Sugita, *Replica-exchange molecular dynamics method for protein folding,* Chemical Physics Letters, 1999

[39] Zhang Xinhuai, *Three Leading Molecular Dynamics Simulation Packages,* National university of Singapore

[40] B. R. Brooks, R. E. Bruccoler, B. D. Olafson, D. J. States, S. Swaminathan, M. Karplus, *CHARMM: A program for macromolecular energy, minimization, and dynamics calculations*, Journal of Computational Chemistry, 1983

[41] D. Van der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, H. J. Berendsen, *GROMACS: fast, flexible, and free* Journal of Computational Chemistry, 2005

[42] David J. Earl and Michael W. Deem, *Monte Carlo Simulations*, Humana Press, 2008

[43] *http://www.gromacs.org/Documentation/How-tos/REMD*

[44] *https://en.wikipedia.org/wiki/Folding@home*

[45] Aviezri S. Frankel, *Complexity of protein folding,* Pergamon Press Ltd, 1993