

# Systemy sprężynowe uczące się mechanicznych zachowań

Maja Milewska (Czoków)

Maj 2021

## 1 Motywacja

Systemy sprężynowe reprezentowane za pomocą grafów, których krawędzie zachowują się jak sprężyny harmoniczne, są powszechnie wykorzystywane w modelowaniu własności obiektów makroskopowych jak i mikroskopowych. Są one na przykład stosowane w modelowaniu materiałów o strukturze nieuporządkowanej w materiałoznawstwie [12] oraz własności elastycznych systemów fizycznych [10]. Pod wieloma względami systemy sprężynowe mogą być rozpatrywane jako metody elementu skończonego dla cząstkowych równań różniczkowych [8]. W systemach mikroskopowych, sprężyna harmoniczna symuluje oddziaływanie pomiędzy atomami/cząsteczkami znajdującymi się w stanie równowagi albo w stanie nieznacznie od niego odchylonym.

W wielu wymienionych wyżej zagadnieniach systemy sprężynowe, symulowane za pomocą aplikacji komputerowych, pomagają znaleźć odpowiedź na pytanie, jak dany obiekt zachowa się pod wpływem działających na niego czynników zewnętrznych. Na przykład jak ciało stałe odkształci się lub pęknie pod wpływem działania na niego siły zewnętrznej albo jak zmieni się stan układu atomów w wyniku podgrzania go lub schłodzenia [9].

Celem tej rozprawy jest zastosowanie odwrotnego podejścia. Mianowicie, naszym celem jest zbudowanie w sposób automatyczny systemu, który po zadziałaniu na niego siły zewnętrznej dzięki swoim własnościom elastycznym ma się odkształcić w odpowiedni sposób.

Stosowane przez nas podejście do systemów sprężynowych mocno nawiązuje do problemu automatycznego projektowania, ponieważ system sprężynowy jest poddany rzeczywistym prawom fizyki i może być wykorzystany przy tworzeniu rzeczywistych obiektów. Według naszej wiedzy jest to nowatorskie podejście w dziedzinach inżynierskich wykorzystujących systemy sprężynowe.

Systemy sprężynowe są również wykorzystywane do badania właściwości białek [1, 7] w metoda Elastic Network Model (ENM). W tej metodzie wszystkie cząsteczki poruszają się ruchem harmonicznym. W naszej metodzie ruch ten jest tłumiony i dodatkowo pozycje części wierzchołków są zewnętrznie kontrolowane. W metodzie ENM dwie cząsteczki są połączone sprężyną, gdy odległość między nimi jest mniejsza niż wartość parametru modelu. W naszej metodzie również stosujemy taką regułę, gdy wykorzystujemy systemy sprężynowe do zakodowania w ich strukturze ścieżek aktywności białek. Natomiast w naszym ogólnym modelu możemy łączyć krawędziami dowolne pary wierzchołków. Wymagamy tylko aby system sprężynowy miał strukturę grafu sztywnego [11].

Wyniki przedstawione w rozprawie zostały opublikowane w pracach [4] oraz [5]. Inne wyniki dotyczące naszego modelu sprężynowego, ale nie opisane w rozprawie, zostały opublikowane w [6].

## 2 Model

W naszym modelu system sprężyn reprezentowany jest za pomocą nieskierowanego grafu sztywnego [11]  $\mathcal{G} := (\mathcal{V}, \mathcal{E})$  ze zbiorem wierzchołków  $\mathcal{V} \subset \mathbb{R}^3$  i zbiorem krawędzi/sprężyn  $\mathcal{E}$ . Z każdą krawędzią  $e = \{u, v\} \in \mathcal{E}$ ,  $u, v \in \mathcal{V}$  kojarzymy jej spoczynkową długość  $\ell_0[e]$  i przez  $\ell[e]$  oznaczamy jej aktualną długość. Do każdej krawędzi  $e \in \mathcal{E}$  jest przypisany współczynnik sprężystości  $k[e] \geq 0$  określający jej elastyczność. Energia konfiguracji systemu sprężyn  $\bar{x}_{\mathcal{V}}$  (konfiguracja  $\bar{x}_{\mathcal{V}}$  określa współrzędne zbioru wierzchołków  $v \in \mathcal{V}$ ) jest dana przez wzór:

$$\mathcal{H}(\bar{x}_{\mathcal{V}}) := \frac{1}{2} \sum_{e \in \mathcal{E}} k[e] (\ell[e] - \ell_0[e])^2 . \quad (1)$$

W rozprawie konstruujemy algorytm, który znajduje minimum lokalne energii (1). Nasz algorytm jest oparty na dynamice spadku gradientowego opisanej następującym równaniem różniczkowym z warunkiem początkowym:

$$\begin{cases} \frac{d}{dt} \bar{x}_{\mathcal{V}} = -\nabla \mathcal{H}(\bar{x}_{\mathcal{V}}) \\ \bar{x}_{\mathcal{V}}^0 = \bar{x}_{\mathcal{V}}(t_0) \end{cases} . \quad (2)$$

Przez  $G[\bar{x}_{\mathcal{V}}^0]$  oznaczmy stan równowagi, do którego dąży układ przy warunku początkowym  $\bar{x}_{\mathcal{V}}^0$ . Proces znajdowania  $G[\bar{x}_{\mathcal{V}}^0]$  nazywamy *relaksacją*.

W praktyce stan równowagi znajdujemy przemieszczając wierzchołki zgodnie z działającymi na nie siłami. Na każdy wierzchołek  $v \in \mathcal{V}$  działa siła od sprężyn przyczepionych do niego:

$$F_v = \sum_{e=u \sim v} k[e] (\ell[e] - \ell_0[e]) \frac{\bar{x}_v - \bar{x}_u}{\ell[e]} . \quad (3)$$

W celu usunięcia niepożądanych ruchów systemu podczas relaksacji, chcemy aby stan równowagi  $G[\bar{x}_{\mathcal{V}}^0]$  był lokalnie asymptotycznie stabilny. Żeby to osiągnąć, wymagamy, aby graf  $\mathcal{G}$  był sztywny. Również wymagamy, aby w  $\mathcal{V}$  istniały co najmniej 3 wierzchołki *zamrożone*, które nie przemieszczają się pod wpływem dynamiki (2). Zbiór tych wierzchołków oznaczamy  $\mathcal{V}_{\text{frozen}}$ . Zbiór wierzchołków, które przemieszczają się pod wpływem dynamiki (2) nazywamy *ruchomymi* i oznaczamy  $\mathcal{V}_{\text{movable}}$ . Wierzchołki zamrożone  $\mathcal{V}_{\text{frozen}}$  dzielimy na:

1. Zbiór  $\mathcal{V}_{\text{con}}$  wierzchołków kontrolnych. Są to wierzchołki, których położenie może być tylko zmienione w wyniku zewnętrznej interwencji na system, poza procesem relaksacji. Celem ich przemieszczenia jest spowodowanie pożądanego przemieszczenia wierzchołków  $\mathcal{V}_{\text{obs}}$ .
2. Zbiór  $\mathcal{V}_{\text{fixed}}$  wierzchołków unieruchomionych, których pozycje nigdy nie ulegają zmianie.

Wierzchołki ruchome  $\mathcal{V}_{\text{movable}}$  dzielimy na:

1. Zbiór  $\mathcal{V}_{\text{obs}}$  wierzchołków obserwowanych, które pod wpływem przemieszczenia wierzchołków kontrolnych i zadziałania dynamiki mają przesunąć się w pożądaną lokalizację.
2. Pozostałe elementy  $\mathcal{V}$  stanowią zbiór wierzchołków pomocniczych  $\mathcal{V}_*$ .

Skoro tylko ruchome wierzchołki podlegają dynamice (2), możemy przekształcić oznaczenie  $G[\bar{x}_{\mathcal{V}}^0]$  w formę  $G[\bar{x}_{\mathcal{V}_{\text{movable}}}^0; \bar{x}_{\mathcal{V}_{\text{frozen}}}]$ , gdzie drugi człon w nawiasie oznacza ograniczenia narzucone na sieć. Dodatkowo ponieważ wierzchołki unieruchomione  $\mathcal{V}_{\text{fixed}}$  są zawsze w tej samej pozycji, ukrywamy ich lokalizacje i wyróżniamy lokalizacje wierzchołków kontrolnych. To pozwala nam przekształcić  $G[\bar{x}_{\mathcal{V}_{\text{movable}}}^0; \bar{x}_{\mathcal{V}_{\text{frozen}}}]$  w formę  $G[\bar{x}_{\mathcal{V}_{\text{movable}}}^0; \bar{x}_{\mathcal{V}_{\text{con}}}]$ .

### 3 Cel

Technicznie mówiąc, głównym celem rozprawy jest stworzenie algorytmu, który dla każdej sprężyny należącej do systemu znajdzie takie wartości parametrów (długość spoczynkową, współczynnik sprężystości), że po przemieszczeniu wierzchołków kontrolnych na z góry zdefiniowane pozycje i przemieszczeniu wierzchołków ruchomych do stanu równowagi zgodnie z dynamiką (2), system będzie miał pożądaną lokalizację wierzchołków obserwowanych. Relacje te opisujemy za pomocą *przykładu uczącego*  $E^{(i)} := (\bar{y}_{\mathcal{V}_{\text{con}}}^{(i)}, \bar{y}_{\mathcal{V}_{\text{obs}}}^{(i)})$ , gdzie  $\bar{y}_{\mathcal{V}_{\text{con}}}^{(i)}$  określa lokalizacje, które możemy narzucić wierzchołkom kontrolnym, w rezultacie wierzchołki obserwowane pod wpływem dynamiki mają się przesunąć w lokalizacje oznaczone przez konfigurację  $\bar{y}_{\mathcal{V}_{\text{obs}}}^{(i)}$ . Dopuszcza się dowolną liczbę przykładów uczących dla danego systemu sprężynowego  $(E^{(i)})_{i=1}^N$ .

W celu określenia jakości dopasowania danej sieci do zbioru przykładów uczących, zdefiniowana jest specjalna kwadratowa funkcja błędu:

$$\Phi = \Phi[(k[e], \ell_0[e])_{e \in \mathcal{E}}] = \frac{1}{N} \sum_{i=1}^N \Phi^{(i)}, \quad (4)$$

gdzie:

$$\begin{aligned} \Phi^{(i)} &= \Phi^{(i)}[(k[e], \ell_0[e])_{e \in \mathcal{E}}] \\ \Phi^{(i)} &= \frac{1}{|\mathcal{V}_{\text{obs}}|} \sum_{v \in \mathcal{V}_{\text{obs}}} (\text{dist}(\bar{y}_v^{(i)}, G[\bar{x}_{\mathcal{V}_{\text{movable}}}^0; \bar{y}_{\mathcal{V}_{\text{con}}}^{(i)}]_v))^2, \end{aligned} \quad (5)$$

gdzie *dist* jest to odległość Euklidesowa pomiędzy dwoma punktami, natomiast  $G[\bar{x}_{\mathcal{V}_{\text{movable}}}^0; \bar{y}_{\mathcal{V}_{\text{con}}}^{(i)}]_v$  jest to lokalizacja wierzchołka  $v$  osiągnięta po ustawieniu wierzchołków ruchomych w konfiguracji  $\bar{x}_{\mathcal{V}_{\text{movable}}}^0$ , wierzchołków kontrolnych w konfiguracji  $\bar{y}_{\mathcal{V}_{\text{con}}}^{(i)}$  i relaksacji systemu sprężynowego.

Aby nauczyć sieć zachowań mechanicznych, które zostały zdefiniowane przez przykłady uczące, wykorzystywany jest algorytm, który modyfikuje fizyczne parametry sprężyn za pomocą metody gradientu prostego, tak żeby zminimalizować wartość funkcji błędu  $\Phi^{(i)}$ . Na wejściu, algorytm dostaje sieć sprężynową  $\mathcal{G} := (\mathcal{V}, \mathcal{E})$  łącznie z początkową konfiguracją  $\bar{x}_{\mathcal{V}}$  oraz zbiorem przykładów uczących  $E^{(i)} := (\bar{y}_{\mathcal{V}_{\text{con}}}^{(i)}, \bar{y}_{\mathcal{V}_{\text{obs}}}^{(i)})$ . Początkowe wartości współczynnika elastyczności mogą mieć taką samą wartość dla wszystkich sprężyn. Każda sprężyna może mieć długość spoczynkową taką jak długość aktualna sprężyny  $\ell_0[e] = \ell[e]$ ,  $e \in \mathcal{E}$ .

Algorytm uczący mechanicznych zachowań zaczyna działanie od ustawienia pozycji wierzchołków kontrolnych zgodnie z ich lokalizacją opisaną w pierwszym przykładzie uczącym  $\bar{y}_{\mathcal{V}_{\text{con}}}^{(1)}$ . Następnie, znajdujemy stan równowagi systemu, zgodnie z dynamiką (2). Zapamiętujemy bieżącą konfigurację systemu, wartości parametrów sprężyn i wartość błędu  $\Phi^{(1)}$  pod postacią zmiennej  $\Phi_{\text{before}}^{(1)}$ .

Aby zminimalizować  $\Phi^{(1)}$  modyfikujemy współczynnik sprężystości  $k[e]$  danej sprężyny  $e \in \mathcal{E}$ , zwiększamy jego wartość o  $\delta$  będącą małym dodatnim parametrem algorytmu i przeprowadzamy proces relaksacji systemu. Ponownie liczymy wartość funkcji błędu i na podstawie różnicy tej wartości i zapamiętanego błędu  $\Phi_{\text{before}}^{(1)}$  przybliżamy pochodną cząstkową  $\frac{\partial \Phi^{(1)}}{\partial k[e]}$ , którą również zapamiętujemy. Pochodne cząstkowe obliczamy w analogiczny sposób dla wszystkich współczynników sprężystości i długości spoczynkowych. Każde obliczenie dokonujemy w oparciu o taki sam zapamiętany stan (współczynniki sprężystości, długości spoczynkowe) i konfigurację sieci. Mając gradient wyliczony, możemy dokonać modyfikacji parametrów dla wszystkich sprężyn za pomocą kroku:

1.  $k[e] = k[e] - \rho \cdot \frac{\partial \Phi^{(1)}}{\partial k[e]}$ ,  $e \in \mathcal{E}$ ,

$$2. \ell_0[e] = \ell_0[e] - \rho \cdot \frac{\partial \Phi^{(1)}}{\partial \ell_0[e]}, \quad e \in \mathcal{E},$$

gdzie  $\rho > 0$  jest to współczynnik długości kroku. Po modyfikacji parametrów dokonujemy relaksacji. Ponownie liczymy i zapamiętujemy  $\Phi^{(1)}$ . Konfiguracja sieci otrzymana w wyniku relaksacji jest konfiguracją początkową dla kolejnego kroku algorytmu. Analogicznych zmian począwszy od zmiany konfiguracji wierzchołków kontrolnych  $\bar{x}_{\mathcal{V}_{\text{con}}} = \bar{y}_{\mathcal{V}_{\text{con}}}^{(i)}$  dokonujemy cyklicznie dla pozostałych przykładów uczących. Jeśli błąd po wszystkich przykładach uczących  $\Phi$  ma trend spadkowy powtarzamy cały proces uczenia. Zarówno model jak i algorytm uczenia mechanicznych zachowań zostały opisane w publikacji [4].

W rozprawie przeprowadziliśmy testy numeryczne w celu sprawdzenia czy system sprężynowy dla różnych warunków początkowych  $\bar{x}_{\mathcal{V}_{\text{movable}}}^0$  i dla takich samych lokalizacji wierzchołków kontrolnych  $\bar{x}_{\mathcal{V}_{\text{con}}} = \bar{y}_{\mathcal{V}_{\text{con}}}^{(i)}$  zbiegnie do stanu równowagi, który osiągnął w trakcie treningu dla *i*tego przykładu uczącego. Nasze testy pokazały, że prawdopodobieństwo odtworzenia poprawnego stanu równowagi jest wysokie dla systemu z małą liczbą wierzchołków pomocniczych i wysokim średnim stopniem wierzchołka. Można zwiększyć odporność sieci na wpadanie w niewytrenowane minima poprzez zaburzanie lokalizacji wierzchołków kontrolnych i zrelaksowanie systemu. Taki szum należy dodać przed dokonaniem przypisania  $\bar{x}_{\mathcal{V}_{\text{con}}} = \bar{y}_{\mathcal{V}_{\text{con}}}^{(i)}$  w trakcie uczenia. Dzięki temu, gdy po dokonaniu przypisania  $\bar{x}_{\mathcal{V}_{\text{con}}} = \bar{y}_{\mathcal{V}_{\text{con}}}^{(i)}$  szukamy stanu równowagi  $G[\bar{x}_{\mathcal{V}_{\text{movable}}}^0; \bar{y}_{\mathcal{V}_{\text{con}}}^{(i)}]$ , mamy zawsze bardzo różne lokalizacje dla warunku początkowego  $\bar{x}_{\mathcal{V}_{\text{movable}}}^0$ .

Procedura dodawania szumu w trakcie uczenia i algorytm testowania odporności sieci na wpadanie w niewytrenowane minima zostały opisane w Sekcji 2.5 rozprawy. Wyniki opisanych testów

zostały opisane w Sekcji 4.6 rozprawy.

## 4 Algorytm generujący topologię grafu $\mathcal{G}$

Na potrzeby testów zaprojektowaliśmy stochastyczny algorytm automatycznie budujący sztuczny system sprężynowy o strukturze grafu sztywnego. Jest on oparty na konstrukcji Henneberga, która została opisana w Sekcji 5 w [14]. Algorytm ten zaczyna tworzenie systemu sprężynowego od wygenerowania grafu pełnego o 4 wierzchołkach. Następnie w sposób indukcyjny dodaje nowe wierzchołki. Każdy nowo dodany wierzchołek przyłącza do  $c_{edge}$  już istniejących najbliższych wierzchołków ( $c_{edge} \geq 3$  jest to parametr algorytmu). Jeśli w grafie jest mniej wierzchołków niż wynosi wartość parametru  $c_{edge}$ , to nowy wierzchołek jest przyłączany do wszystkich już istniejących wierzchołków. Lokalizacja wierzchołków unieruchomionych i pomocniczych losowana jest z rozkładu jednostajnego ograniczonego przez kulę, której punkt centralny jest parametrem algorytmu. Lokalizacja wierzchołków kontrolnych i obserwowanych określona jest przez przykłady uczące, które muszą być podane na wejściu algorytmu. W rozprawie opisaliśmy również propozycję stochastycznej procedury, która może być użyta do budowy takiego zbioru przykładów uczących.

Algorytm generujący topologię grafu  $\mathcal{G}$  został opisany w publikacji [5] oraz w Podsekcji 2.4 w rozprawie. Procedura generująca zbiór przykładów uczących opisana została w Sekcji B.6 dodatku do rozprawy.



## 5 Wyniki

W Podsekcji 5.1 opisujemy najważniejsze własności numeryczne algorytmu uczącego mechanicznych zachowań. Wskazujemy tam, że na jakość wytrenowania sieci przede wszystkim ma wpływ nasylenie jej struktury sprężynami.

W Podsekcji 5.2 pokazujemy w jaki sposób wykorzystaliśmy algorytm uczący mechanicznych zachowań do znalezienia systemów sprężynowych o strukturze i dynamice białek. Podsumowujemy tam również jakość odwzorowania rzeczywistych obiektów białkowych przez znalezione struktury. Nasze testy pokazały, że w ścieżkach, które pokonuje białko w trakcie swojej biologicznej aktywności zakodowane są wartości jego oddziaływań międzycząsteczkowych.

### 5.1 Algorytm uczący mechanicznych zachowań

Przeprowadziliśmy testy numeryczne w celu sprawdzenia jak liczba wierzchołków pomocniczych i krawędzi grafu wpływa na umiejętności przystosowywania się sieci sprężynowych do przykładów uczących. W Tablicy 1 zaprezentowaliśmy złożoność problemów użytych do testów oraz średni błąd  $\Phi$  osiągnięty przez wytrenowane sieci w zależności od liczby zasobów (wierzchołków pomocniczych, krawędzi). Sieci sprężynowe użyte do testów były budowane automatycznie za pomocą naszego algorytmu indukcyjnego opisanego w Sekcji 4. Również zbiory przykładów uczących dla każdej sieci były generowane w sposób losowy. Konstrukcja i adaptacja sieci dla każdego zestawu parametrów z Tablicy 1 została wykonana stukrotnie.

Testy pokazały, że dodawanie kolejnych wierzchołków pomocniczych do pewnego stopnia dobrze wpływa na lepsze dopasowanie

$N$	$ \mathcal{V}_{\text{con}} $	$ \mathcal{V}_{\text{obs}} $	rzęd błędu $\bar{\Phi}$ przed uczeniem	średni błąd $\bar{\Phi}$ nauczonej sieci			
				$ \mathcal{V}_* =5,$ $c_{\text{edge}}=3$	$ \mathcal{V}_* =50,$ $c_{\text{edge}}=3$	$ \mathcal{V}_* =5,$ $c_{\text{edge}}=8$	$ \mathcal{V}_* =50,$ $c_{\text{edge}}=8$
5	5	4	$8 \cdot 10^3$	2139	1618	642	241
5	5	10	$9 \cdot 10^3$	3539	3307	2144	1414
10	15	3	$3 \cdot 10^3$	1054	873	252	123

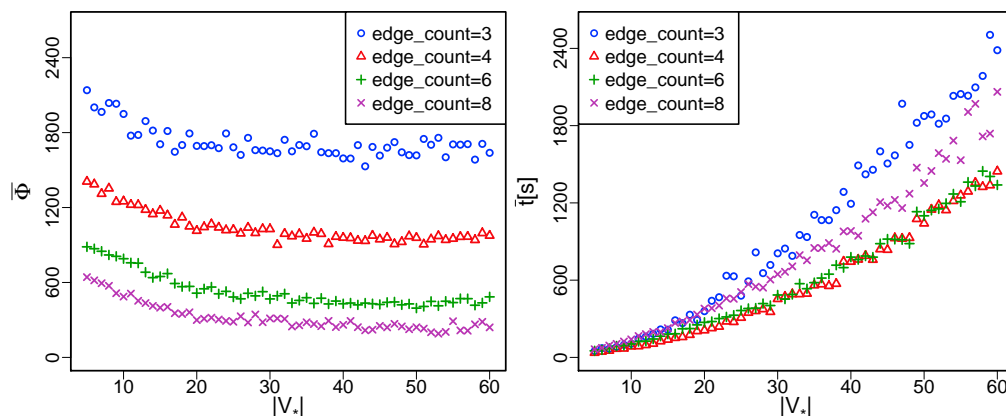
Tablica 1: Trzy pierwsze kolumny opisują złożoność zbiorów przykładów uczących użytych do testowania. Następną kolumną podaje średni rząd błędu przed uczeniem. Cztery ostatnie kolumny zawierają średnie wartości osiągniętych błędów dla wytrenowanych sieci w zależności od nasycenia sieci wierzchołkami pomocniczymi oraz krawędziami.

się sieci do przykładów uczących w procesie uczenia. Aczkolwiek kolejne wierzchołki znacząco zwiększają czas potrzebny do wytrenowania systemu. Aby system przystosował się lepiej do pożądanych zachowań, najbardziej istotne jest nasycenie jego struktury dużą liczbą krawędzi, czyli zapewnienie mu wysokiego średniego stopnia wierzchołka (zobacz Podsekcję 4.5 w rozprawie). Na Rysunku 1 można zobaczyć jak zmniejsza się średni błąd  $\bar{\Phi}$  dla wytrenowanej sieci oraz wydłuża się czas uczenia wraz ze wzrostem liczby wierzchołków pomocniczych oraz krawędzi (wraz ze wzrostem parametru  $c_{\text{edge}}$ ).

Dla naszych problemów testowych, systemy z odpowiednią liczbą wierzchołków pomocniczych oraz krawędzi ( $|\mathcal{V}_*|=50, c_{\text{edge}}=8$ ), po nauczaniu osiągały średnio 40, 9 i 30-krotnie mniejszy błąd niż przed procesem uczenia.

## 5.2 Uczenie zachowań białek

Algorytm uczący mechanicznych zachowań wykorzystaliśmy do znalezienia systemów sprężynowych o strukturze i dynamice białek. W tym celu struktury białek powiązaliśmy ze strukturami



Rysunek 1: Wykres  $\bar{\Phi}$  (z lewej) oraz średni czas symulacji (z prawej) w zależności od liczby wierzchołków pomocniczych. Każdy punkt na wykresie jest średnią wartością dla 100 symulacji. Sieć trenowana dla 4 przykładów uczących ( $N = 4$ ,  $|\mathcal{V}_{\text{con}}| = 5$ ,  $|\mathcal{V}_{\text{obs}}| = 5$ ).

sieci sprężynowych. Cząsteczki danego białka zostały zmapowane jeden do jednego na wierzchołki grafu. W naszych testach losowe 20% cząsteczek danego białka/wierzchołków zdefiniowaliśmy jako wierzchołki kontrolne a pozostałe 80% jako wierzchołki obserwowane. W tym przypadku dynamika sprężyn harmonicznnych przybliża interakcje pomiędzy parami cząsteczek znajdującymi się blisko siebie. Skrajne (pierwszy i ostatni) przykłady uczące zostały zbudowane w oparciu o rzeczywiste konformacje białek pochodzące z bazy PDB, zobacz [2]. Są to dwie konformacje brzegowe danego białka przed i po wykonaniu aktywności biologicznej. Pośrednie przykłady uczące zostały wygenerowane za pomocą metody GOdMD opisanej w [13].

By odwzorować daną konformację za pomocą wytrenowanego systemu, należy ustawić lokalizacje jej wierzchołków kontrolnych zgodnie z jej przykładem uczącym i znaleźć jej stan stabilny. Struktury białek wytrenowane przez nasz algorytm bardzo dobrze przybliżają wzorcowe przykłady uczące. W naszych testach

średni błąd  $\Phi$  dla wszystkich protein wyniósł  $0.33\text{\AA}^1$ .

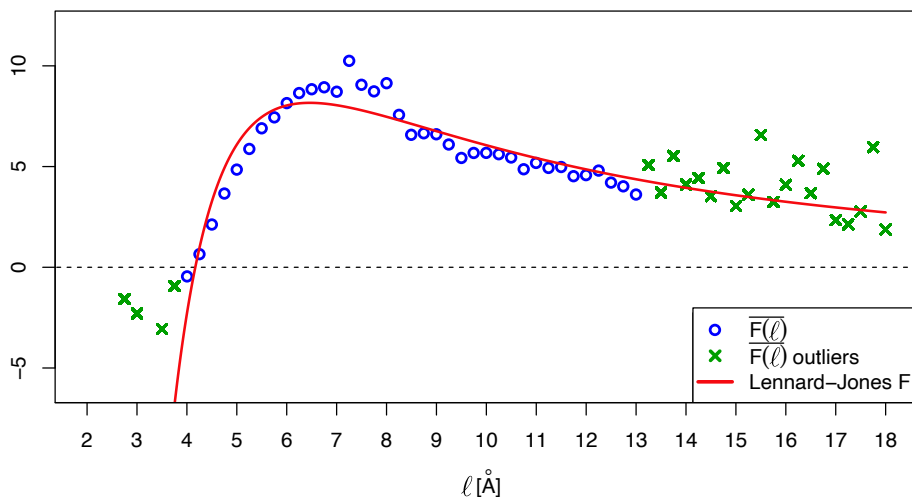
Przeanalizowaliśmy również fizyczne właściwości wytrenowanych konformacji. W tym celu zobrazowaliśmy jak zmienia się średnia wartość sił działająca między parami ich cząsteczek wraz ze wzrostem odległości pomiędzy tymi cząsteczkami. Pokazaliśmy, że otrzymana zależność jest zgodna z kształtem siły Lennarda-Jonesa, która jest uznana za pierwowzór modelu służącego do obrazowania rzeczywistych interakcji pomiędzy cząsteczkami. By porównać obie funkcje, dla siły Lennarda-Jonesa znaleźliśmy parametry, dla których w jak najdokładniejszy sposób interpoluje ona uśrednione siły istniejące w naszych wytrenowanych sieciach sprężynowych. W tym celu użyliśmy algorytmu spadku gradientowego, który minimalizował błąd średniokwadratowy pomiędzy obiema krzywymi. Błąd interpolacji wyniósł  $0.84\text{\AA}$ . Wykresy obu funkcji zaprezentowane są na Rysunku 2.

## 6 Podsumowanie

W naszej rozprawie zaproponowaliśmy nowy parametryczny algorytm uczący systemy sprężynowe mechanicznych zachowań. Dla danego problemu, dla określonej pozycji wierzchołków sterujących procedura znajduje parametry fizyczne (długości spoczynkowe, współczynniki sprężystości), dla których dynamika sieci spowoduje pożądaną reakcję wierzchołków obserwowanych. Algorytm został przez nas zaimplementowany i przetestowany na różnych zestawach przykładów uczących. Nasze testy pokazały, że znajduje on dobrze przystosowane sieci sprężynowe dla konstrukcji o wysokim średnim stopniu wierzchołka i nie za dużej liczbie wierzchołków pomocniczych. Dodawanie wierzchołków pomocniczych

---

<sup>1</sup> $\text{\AA}$  - angstrom jest jednostką długości równą  $10^{-10}$  metra



Rysunek 2: Średnie siły  $\overline{F(\ell)}$  w naszym wytrenowanym modelu są zaznaczone na niebiesko dla wartości użytych do interpolacji oraz na zielono dla pozostałych wartości, które zaklasyfikowaliśmy jako odstające. Na czerwono zaznaczyliśmy siłę Lennarda-Jonesa, która przybliża siły obliczone w naszej symulacji.

szybko przestaje zauważalnie poprawiać przystosowanie sieci do przykładów uczących i wydłuża czas uczenia. Dla naszych problemów testowych błąd przed i po uczeniu zmniejszył się odpowiednio 40, 9 i 30-krotnie. Utrzymanie wysokiego średniego stopnia wierzchołka i nie za dużej liczby wierzchołków pomocniczych ma dodatkowe zalety. Dzięki temu system jest bardziej odporny na szum, który może powodować nieprawidłowe przemieszczenie wierzchołków obserwowanych po przesunięciu wierzchołków kontrolnych.

Model zdefiniowany w rozprawie został wykorzystany do znajdowania sieci sprężynowych naśladujących dynamikę białek. Nasze systemy sprężynowe nauczyły się odtwarzać ścieżki, które po

konują białka w trakcie pełnienia swoich funkcji biologicznych, osiągając średni błąd  $\Phi$  równy  $0.33\text{\AA}$  dla białka. Zobrazowaliśmy jak zmienia się średnia wartość sił działająca między parami cząsteczek wraz ze wzrostem odległości pomiędzy tymi cząsteczkami. Pokazaliśmy, że otrzymana zależność jest zgodna z kształtem siły Lennarda-Jonesa.

## Literatura

- [1] A. R. Atilgan, S. R. Durell, R. L. Jernigan, M. C. Demirel, O. Keskin, I. Bahar, *Anisotropy of fluctuation dynamics of proteins with an elastic network model*, Biophysical journal, 80(1), 505–515, DOI: 10.1016/S0006-3495(01)76033-X
- [2] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, P. E. Bourne, *The Protein Data Bank*, Nucleic Acids Research 28, 235–242, (2000)
- [3] R. Connelly, W. Whiteley, *Second-Order Rigidity and Prestress Stability for Tensegrity Frameworks*, SIAM Journal of Discrete Mathematics 9, 453–491, (1996)  
A. L. Cauchy, *Méthode générale pour la résolution des systèmes d'équations simultanées*, Comp. Rend. Sci. Paris, 25, 46–89, (1847)
- [4] M. Czoków, T. Schreiber, *Adaptive Spring Systems for Shape Programming*, ICAISC 2010, Part II, LNAI 6114, 420–427, (2010)
- [5] M. Czoków, T. Schreiber, *Structure Searching for Adaptive Spring Networks for Shape Programming in 3D*, ICAISC 2012, PART II, LNCS 7268, 207–215, (2012)

- [6] M. Czoków, J. Miękiś, *Influence of a Topology of a Spring Network on its Ability to Learn Mechanical Behaviour*, PPAM 2013, PART I, LNCS 8384, 412–422, (2014)
- [7] G. M. Crippen, T.F. Havel, *Distance Geometry and Molecular Conformation*, Wiley, New York, (1988)
- [8] A. A. Gusev, *Finite Element Mapping for Spring Network Representations of the Mechanics of Solids*, Phys. Rev. Lett. 93, 034302, (2004)
- [9] E. Granato, S. C. Ying, *Dynamical transitions and sliding friction in the two-dimensional Frenkel-Kontorova model*, Phys. Rev. B 59, 5154, (1999)
- [10] M. Kellomäki, J. Aström, J. Timonen *Rigidity and Dynamics of Random Spring Networks*, Phys. Rev. Lett. 77, 2730, (1996)
- [11] R. Olfati-Saber, R. M. Murray, *Graph Rigidity and Distributed Formation Stabilization of Multi-Vehicle Systems*, Proc. of the 41st IEEE Conf. on Decision and Control, Las Vegas, Nevada, (2002)
- [12] M. Ostoja-Starzewski: *Lattice Models in Micromechanics*, Appl. Mech. Rev. 55, 35–60, (2002)
- [13] P. Sfriso, A. Hospital, A. Emperador, M. Orozco, *Exploration of conformational transition pathways from coarse-grained simulations*, Bioinformatics, 2013 15 Aug, 29(16), 1980–1986, (2013)
- [14] A. E. Varvitsiotis, *Algebraic and combinatorial techniques in rigidity theory*, MSc thesis, (2009), <http://users.uoa.gr/avarvits/MSc.pdf>