



**AUTOMATSKO RASPOREĐIVANJE ELEMENATA DIJAGRAMA UPOTREBOM SUGIYAMA ALGORITMA, LSTM NEURONSKIH MREŽA I ELK BIBLIOTEKE**

**AUTOMATIC GRAPH LAYOUT USING SUGIYAMA ALGORITHM, LSTM NEURAL NETWORKS AND ELK LIBRARY**

Nenad Mišić, *Fakultet tehničkih nauka, Novi Sad*

**Oblast - SOFTVERSKO INŽENJERSTVO I INFOMACIONE TEHNOLOGIJE**

**Kratak sadržaj** – Ovaj rad predstavlja specifikaciju i implementaciju sistema za automatsko raspoređivanje elemenata dijagrama. Čvorovi predstavljaju komponente za obradu signala (signal processing) i energetske elektroniku (power electronics), pa je jedan od posebnih zahteva projekta da se raspoređivanje prilagodi tom slučaju korišćenja i na smislen način prikaže date dijagrame.

**Ključne reči:** vizualizacija, grafovi, automatsko raspoređivanje, sugiyama algoritam, LSTM, ELK

**Abstract** – This paper presents the specification and implementation of a system for automatic diagram layouting. Nodes are power electronics and digital signal processing components, so one of the special requirements of the project is to adjust the layouting to that use case and display the given diagrams in a meaningful way.

**Keywords:** visualization, graphs, automatic layout, sugiyama algorithm, LSTM, ELK

## 1 UVOD

Kompanija *Typhoon HIL*, lider u domenu energetske elektronike, proizvodi *HIL (Hardware-in-the-loop)* uređaje za testiranje energetskih postrojenja simulacijom u realnom vremenu. Deo ovog hardversko/softverskog proizvoda predstavlja interaktivni grafički editor (*Schematic editor*), čija je namena modelovanje kompleksnih energetskih sistema, vizuelno predstavljenih u obliku grafa komponenti.

Procenjeno je da oko 25% vremena pri kreiranju grafova biva potrošeno na manuelno raspoređivanje. Korišćenjem algoritama za automatsko raspoređivanje može se uštedeti vreme, umanjiti napor korisnika, preusmeriti pažnja sa raspoređivanja na modelovanje i znatno povećati čitljivost i razumljivost grafa.

U ovom radu predstavljeno je nekoliko pristupa za automatsko raspoređivanje grafova sa ciljem da se raspored što bolje uklopi u domen obrade signala i energetske elektronike i time rezultuje čitljivom, razumljivom i smislenom vizualizacijom. Prvi pristup, tzv. „naivni“, zasniva se na *layer-based* algoritmu razvijenom od strane Sugiyama et al. [1]. Drugi pristup koristi *LSTM (Long-short term memory)* [2] rekurentne neuronske mreže.

### NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Igor Dejanović, vanr. prof.

Poslednji pristup, koji daje najbolje rezultate, oslanja se na *ELK (Eclipse Layout Kernel)* softverski alat [3].

## 2 TEORIJSKE OSNOVE

### 2.1 Grafovi

Graf  $(V, E)$  je uređeni par čvorova i grana, gde je grana predstavljena kao par  $(u, v)$  čvorova. U zavisnosti od toga da li je par  $(u, v)$  uređen ili ne, graf je usmeren ili neusmeren.

Crtež grafa  $\Gamma$  mapira svaki čvor  $v$  na jedinstvenu tačku  $\Gamma(v)$ , a svaku granu  $(u, v)$  na krivu  $\Gamma(u, v)$  sa početkom u tački  $\Gamma(u)$  i krajem u tački  $\Gamma(v)$ . Svaki graf može biti predstavljen sa više različitih crteža. Shodno tome, način na koji se graf iscrtava utiče na čitljivost, lakoću tumačenja i razumevanja, pa samim tim i na upotrebljivost.

### 2.2 Kriterijumi dobre vizualizacije grafa

Kako isti graf može imati više različitih crteža, postavlja se pitanje kako odrediti koji crtež je „dobar“ a koji ne.. Neke od glavnih smernica za iscrtavanje estetski ugodnih i čitljivih grafova su [5]:

- Minimizacija broja preklapljenih grana
- Uniformni tok
- Prave linije
- Optimalne dužine grana
- Uniformna gustina čvorova
- Uniformne dužine grana

### 2.3 Layer-based algoritam raspoređivanja

Algoritmi za automatsko raspoređivanje elemenata dijagrama se, na osnovu načina raspoređivanja čvorova, mogu podeliti u više grupa, kao što su stabla, kružni grafovi, *force-directed*, *layer-based* itd.

Raspoređivanje zasnovano na slojevima (*layer-based*) jedan je od najpopularnijih načina za raspoređivanje dijagrama koji su zasnovani na toku, kao što su dijagrami energetske elektronike ili digitalne obrade signala. Prvi rad koji se bavio ovom temom predstavljen je od strane Sugiyama et al. [1], a takođe, i većina novijih radova na temu *layer-based* pristupa pri raspoređivanju grafova zasniva se baš na ovom algoritmu. Ono što se srž ovog pristupa jeste da se svi čvorovi rasporede po diskretnim slojevima. Kom sloju će čvor pripadati diktiraju grane koje u njemu počinju ili se završavaju, a cilj je optimizovati raspored tako da se ispune estetski

kriterijumi kao što su uniformni tok podataka, uniformna gustina čvorova i uniformna dužina grana. Kada se svaki čvor smesti, računaju se njihove pozicije unutar sloja i na taj način se minimizuju preklapanja grana [1].

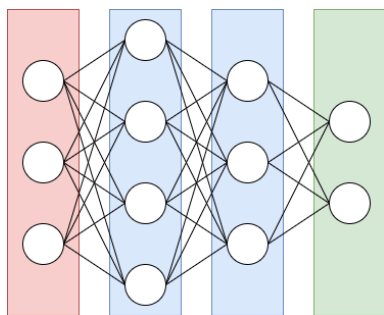
## 2.4 Neuronske mreže

Svrha neuronskih mreža je da aproksimiraju neku funkciju, koja je najčešće nepoznata ili nepogodna za rad (npr. nema izvod). Neuronske mreže se sastoje iz veštačkih neurona (perceptrona), modelovanih po uzoru na biološke neurone. Mnoštvo neurona, organizovanih po slojevima, funkcionise zajedno i na taj način formira neuronsku mrežu koja može verodostojno da aproksimira matematičku funkciju.

Perceptron ima  $n$  ulaza preko kojih podaci dolaze do tela. Tu se dešava potencijalna aktivacija, tako što se svi ulazni impulsi množe sa korespondentnim težinama i međusobno sumiraju. Dobijena suma provlači se kroz aktivacionu funkciju, čiji je zadatak određivanje pobuđenosti neurona za date ulaze, i rezultat šalje na izlaz.

Neuronske mreže sastoje se iz slojeva veštačkih neurona. Svi slojevi su potpuno povezani (*fully connected*), odnosno ulazne vrednosti neurona sloja  $k$  su skup izlaznih vrednosti svih neurona sloja  $k - 1$ , sa izuzetkom prvog sloja čiji je ulaz ujedno i ulaz čitave neuronske mreže. Izlaz iz poslednjeg sloja predstavlja izlaz cele neuronske mreže. Ova osnovna arhitektura mreže u literaturi se često naziva *višeslojni perceptron*.

Na slici 2.2 prikazan je model višeslojnog perceptrona sa dva skrivena sloja. Neuronska mreža ima tri ulazna i dva izlazna perceptrona.



Slika 2.2. Prikaz modela višeslojnog perceptrona sa dva skrivena sloja.

### 2.4.1 Rekurentne neuronske mreže

Za razliku od klasičnih neuronskih mreža, odnosno višeslojnih perceptrona, koji obrađuju jedan po jedan ulaz, rekurentne neuronske mreže (Recurrent Neural Network, RNN) obrađuju sekvencu ulaza, usput zadržavajući kontekstne informacije. Ovo ih čini idealnim sredstvom za obradu sekvence podataka varijabilne dužine, kao što su video i audio snimci ili pak grafovi sa varijabilnim brojem čvorova. Rekurentne mreže rade slično poput klasičnih neuronskih mreža, sa razlikom da izlaz obrade elementa sekvence predstavlja jedan od ulaza za obradu sledećeg elementa. Na taj način je omogućeno čuvanje informacija u toku obrade. Glavni problem rekurentnih neuronskih mreža jesu takozvani nestajući i eksplozivni gradijenti (*vanishing and exploding gradient*). Razlog pojave ovog fenomena je to što računanje greške na izlazu uključuje puno množenja težina sa aktivacionom funkcijom (posebno u slučaju

dugačkih sekvenci), pa se u praksi vrlo često koriste varijante koje rešavaju ovaj problem, kao što je *LSTM*.

*Long short-term memory (LSTM)* [2] neuronske mreže su specijalne rekurentne neuronske mreže, čija specifičnost leži u tome što, umesto jednog potpuno povezanog sloja, uvodi četiri nove komponente (gates) koje rešavaju pomenuti problem i utiču na izlaz mreže, na sledeći način:

- *Forget gate* – da li da se obriše trenutno stanje
- *Input gate* – da li da se prihvati ulaz
- *Gate gate* – koliko i na koji način treba prihvatiti ulaz
- *Output gate* – da li da se prosledi izlaz sledećoj iteraciji

## 2.5 Eclipse Layout Kernel

*KIELER* je istraživački projekat koji se bavi unapređivanjem dizajna složenih sistema zasnovanih na grafičkom modelu. Između ostalog, veliki deo istraživanja posvećen je automatskom raspoređivanju elemenata dijagrama. *Eclipse Layout Kernel (ELK)* predstavlja istraživački projekat baziran na *KIELER* projektu. Ukratko, cilj istraživanja predstavlja proširenje postojećeg layer-based algoritma za raspoređivanje da podržava port ograničenja, hijerarhijske portove i petlje (*self-loops*). *ELK* pruža relevantne implementacije za *Java* i *Javascript* tehnologije, a integracija u postojeći sistem je brza i jednostavna.

## 3 IMPLEMENTACIJA REŠENJA

Mrežni dijagrami, poput električnih kola, najčešće se iscrstavaju korišćenjem pristupa baziranog na slojevima (*layer-based*) za raspoređivanje čvorova i ortogonalnog pristupa za iscrtavanje grana [4]. Prema tome, prvi pristup u rešavanju problema automatskog raspoređivanja zasniva se na najpoznatijem *layer-based* algoritmu osmišljenom od strane *Sugiyama et al.* [1]. Drugi pristup koristi model *LSTM* neuronske mreže, dok poslednji, i pristup sa najboljim rezultatima, koristi funkcionalnosti *Eclipse Layout Kernel* biblioteke.

### 3.1 Pristup korišćenjem Sugiyama algoritma

Dok je originalni *Sugiyama* algoritam podeljen u pet faza, implementirani algoritam rešava problem automatskog raspoređivanja grafova u četiri uzastopne faze, a izostavlja petu koja se tiče iscrtavanja grana.

**Uklanjanje ciklusa**, kao prva faza algoritma, vrši se privremenim uklanjanjem jednog od čvorova iz ciklusa, za svaki ciklus koji postoji u grafu. Uklonjeni čvor biće iscrtan se u prvom mogućem sloju tako da indeks tog sloja bude veći od indeksa slojeva svih ostalih učesnika u ciklusu.

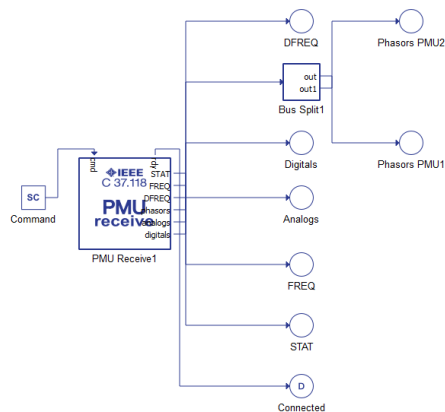
**Dodela čvorova slojevima** vrši se iterativno, tako što se u svakoj iteraciji pronalaze čvorovi čiji roditelji su već dodeljeni. U prvoj iteraciji, dodeljuju se korenski čvorovi, odnosno čvorovi koji nemaju roditelje, a s obzirom da graf ne sadrži cikluse, takvi čvorovi moraju postojati. Nakon toga, iteracije se nastavljaju sve dok se svim čvorovima ne dodele slojevi.

**Minimizacija ukrštanja** ispostavlja se kao najteži deo ovog problema, zbog ograničenja portova čiji su pozicija i redosled unapred određeni, zavisno od tipa komponente. Ovo je jedan od uočenih problema *Sugiyama* algoritma, i rešenje tog problema predloženo je u radu od strane

Schulze et al. [3]. Ovo rešenje iskorišćeno je u sklopu ELK alata, gde je jedan od glavnih ciljeva bio upravo rešavanje ovog problema.

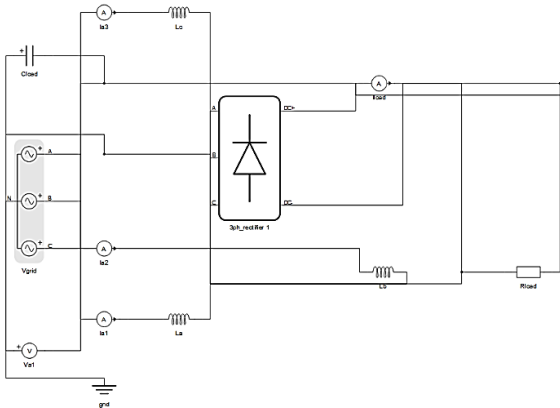
**Određivanje pozicije čvorova** je faza sa ciljem postizanja što većeg broja pravih linija podešavanjem vertikalnih pozicija čvorova. Algoritam implementiran u ovom pristupu inicijalno postavlja sve čvorove na vertikalnu poziciju koja je jednaka aritmetičkoj sredini vertikalnih pozicija roditeljskih čvorova, a potom iterativno vrši minimalna pomeranja čvorova sve dok postoje preklapanja između čvorova.

Jedan od najjednostavnijih primera za evaluaciju je model digitalne obrade signala prikazan na slici 3.1.



Slika 3.1. Prikaz jednostavnog modela digitalne obrade signala, raspoređenog korišćenjem Sugiyama algoritma

Još jedan od jednostavan primer za evaluaciju je model trofaznog ispravljača, prikazan na slici 3.2.



Slika 3.2. Prikaz modela trofaznog ispravljača, raspoređenog korišćenjem Sugiyama algoritma

Sa slika se može primetiti da su, zbog odsustva treće faze algoritma, čvorovi poređani tako da postoji dosta nepotrebnih preklapanja grana. Sa druge strane, vidi se da je generalni pristup dobar, da se *layer-based* raspoređivanje uklapa u domen problema i da postoji potencijal za kvalitetnu vizualizaciju.

### 3.2 Pristup korišćenjem LSTM neuronske mreže

Motivacija za ovaj pristup je postojanje skupa ručno raspoređenih grafova koji bi mogle da se iskoriste u svrhu treniranja adekvatnog modela. Taj skup sastoji se od 908 datoteka ručno raspoređenih grafova.

Format razmene podataka sa neuronskim mrežama mora biti u vektorskom, odnosno matricnom obliku, pa je za

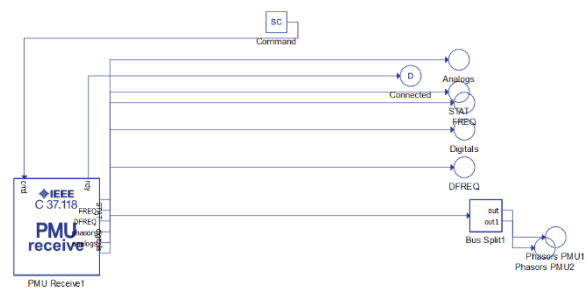
predstavu grafova iskorišćen format *matrice susedstva*. Naime, matrica susedstva grafa koji se sastoji od  $n$  čvorova je matrica dimenzije  $n \times n$ , gde je svaki čvor predstavljen jednim redom matrice. Polja tog reda mogu imati vrednosti 0 i 1, zavisno od toga da li postoji grana koja povezuje čvor na indeksu reda sa čvorom na indeksu kolone.

Pored ovih informacija, redovi matrice prošireni su dodatnim poljima kao što su broj terminala komponente, dimenzije, naziv komponente itd. Na taj način, broj čvorova grafa ograničen je na 100, a svaki red matrice je vektor dužine 135. Na osnovu ovog formata, model neuronske mreže u ulaznom sloju ima 135 neurona. Ulazi se dalje prosleđuju LSTM sloju dimenzionalnosti 512. Izlaz iz LSTM sloja ide u potpuno povezani (*dense*) sloj sa 2 neurona, kako bi se formirao izlaz iz cele mreže. Funkcija greške koja se koristi je prosečna kvadratna greška (*mean squared error, MSE*), a funkcija aktivacije koja se koristi je *Leaky ReLU*, jedna od standardnih varijanti za probleme regresije.

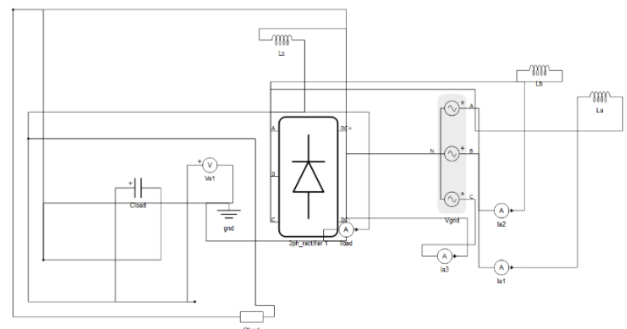
Nakon treniranja modela, jasno se vidi da rezultati nisu zadovoljavajući. Čvorovi grafova su razbacani bez puno smisla i uglavnom postoji ogroman broj preklopljenih grana. Ipak, model uspeva da razvoji komponente tako da nisu sve na sredini, i ta činjenica ulaze nadu da je problem manjak podataka za treniranje na osnovu kojih je mreža mogla da nauči težine.

Drugi potencijalni razlog neuspeha ove metode može biti prevelika dimenzionalnost za LSTM model. Ulaz u model je sekvenca vektora, svaki dimenzionalnosti preko 100, što zahteva dubok model kako bi se naučile kompleksne veze koje se kriju u podacima.

Na slikama 3.3 i 3.4 prikazani su isti apstraktni grafovi kao na slikama 3.1 i 3.2, raspoređeni istreniranim LSTM modelom.



Slika 3.3. Prikaz jednostavnog modela digitalne obrade signala, raspoređenog korišćenjem LSTM modela



Slika 3.4. Prikaz modela trofaznog ispravljača, raspoređenog korišćenjem LSTM modela

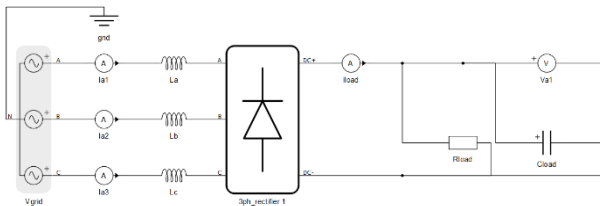
### 3.3 Pristup korišćenjem *ELK* biblioteke

*ELK* je alat za automatsko raspoređivanje grafova u sferi ugrađenog softvera kao i dijagrama električnih kola i ostalih grafova sa port ograničenjima [3]. Pored toga što se slučaj primene poklapa sa našim projektom, *ELK* biblioteka oslanja se na *Sugiyama* algoritam, koji, videli smo, daje solidne rezultate. Kako je ranije objašnjeno, odsustvo port ograničenja u *Sugiyama* algoritmu u velikoj meri je negativno uticalo na rezultate prvog pristupa. Očekivano je da ovaj pristup pruži čitljivije i razumljivije rezultate u odnosu na prvi pristup, sa manje preklapanja grana i više pravih linija.

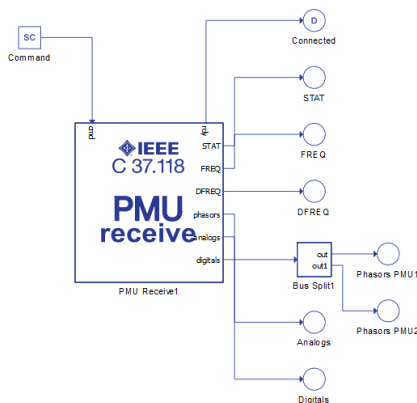
S obzirom da *ELK* nema relevantnu *Python* implementaciju, napravljena je jednostavna *Node.js* aplikacija čiji je zadatak da iščita datoteku u kojoj se nalazi tekstualni zapis grafa u *ElkGraph* formatu. Dobijeni objekat raspoređuje korišćenjem *ELKjs* biblioteke, a potom ga upisuje u izlaznu datoteku, u istom formatu. Odgovornost transformacije iz *Typhoon* formata u *ElkGraph* format prebačena je na *Typhoon HIL* softver, koji pokreće *ELK* servis kao podproces prosleđujući mu potrebne parametre korišćenjem *subprocess* modula. Ovakvom arhitekturom, postignuto je jednostavno upravljanje ali i željena portabilnost (imajući u vidu portabilnost *Javascript* aplikacija). Funkcionalnost automatskog raspoređivanja se vrlo lako može integrisati i u grafički korisnički interfejs.

Očekivano, rezultati su se ispostavili kao najbolji među ova tri pristupa. Raspored u velikoj meri podseća na raspored korišćenjem *Sugiyama* algoritma, ali zbog dodatnih podešavanja i ograničenja algoritam rezultuje kvalitetnijim rasporedom. Kod većine modela teško je uočiti razliku između automatski i ručno raspoređenog grafa.

Na slikama 3.5 i 3.6 prikazani su isti apstraktni grafovi kao na slikama 3.2 i 3.1, raspoređeni uz oslonac na *ELK*.



Slika 3.5. Prikaz modela trofaznog ispravljača, raspoređenog korišćenjem *ELK* biblioteke



Slika 3.6. Prikaz jednostavnog modela digitalne obrade signala, raspoređenog korišćenjem *ELK* biblioteke

### 4. ZAKLJUČAK

U ovom radu predstavljeno je nekoliko pristupa za automatsko raspoređivanje grafova sa ciljem da se raspored što bolje uklopi u domen obrade signala i energetske elektronike i time rezultuje čitljivom, razumljivom i smislenom vizualizacijom. Najbolje rezultate daje pristup koji se oslanja na *Eclipse Layout Kernel* biblioteku, a rezultujuće vizualizacije u velikoj meri podsećaju na ručno raspoređene grafove, što je i bio cilj istraživanja.

Dalje istraživanje moglo bi se proširiti u pravcu automatskog raspoređivanja čvorova koji predstavljaju komentare, s obzirom da ti čvorovi nisu direktno povezani sa ostatkom grafa. Takođe, dalje istraživanje moglo bi da ide u pravcu neuronskih mreža sa ciljem pronalaženja modela koji daju bolje rezultate u odnosu na *LSTM*.

### 5. LITERATURA

- [1] Sugiyama, Kozo, Shojiro Tagawa, and Mitsuhiko Toda. "Methods for visual understanding of hierarchical system structures." *IEEE Transactions on Systems, Man, and Cybernetics* 11.2 (1981): 109-125.
- [2] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.
- [3] Schulze, Christoph Daniel, Miro Spönemann, and Reinhard Von Hanxleden. "Drawing layered graphs with port constraints." *Journal of Visual Languages & Computing* 25.2 (2014): 89-106
- [4] Blackwell, Alan, Kim Marriott, and Atsushi Shimojima. "Diagrammatic Representation and Inference." (2004).
- [5] Purchase, Helen C. "Metrics for graph drawing aesthetics." *Journal of Visual Languages & Computing* 13.5 (2002): 501-516.

### Kratka biografija:

**Nenad Mišić** rođen je u Kruševcu 1997. god. Master rad na Fakultetu tehničkih nauka iz oblasti Softverskog inženjerstva i informacionih tehnologija – Inteligentni sistemi odbranio je 2021.god.

kontakt: [nenad.misic@uns.ac.rs](mailto:nenad.misic@uns.ac.rs)