

Примена алгоритма неуроеволуције са променљивим топологијама (NEAT) на примеру симулације кретања робота

Ментор:
др Ђорђе Обрадовић

Аутор:
Нина Марјановић
RA138/2012



ЗАДАТАК

Опис задатка који пројекат треба да реши

ЗАДАТАК

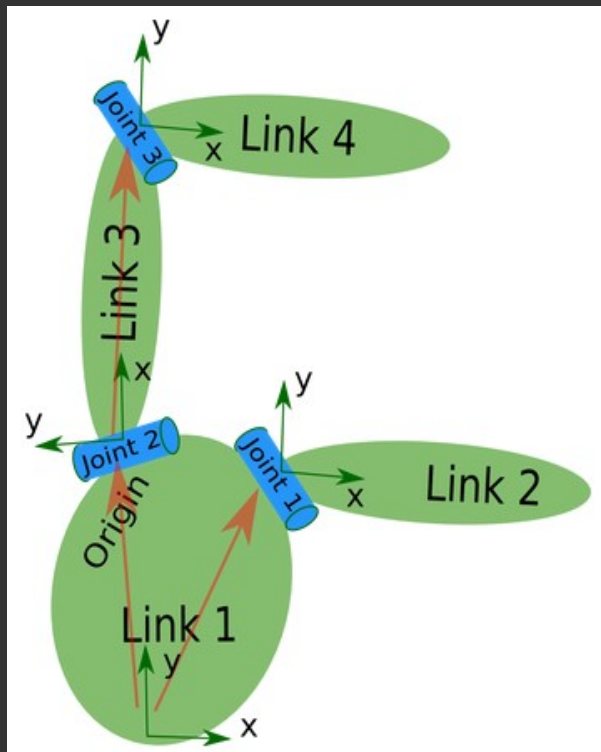
- Креирати софтвер који је у стању да симулира кретање робота на основу учитаних параметара.
- Коришћењем NEAT алгоритма пронаћи праве тежине и топологије како би вештачка неуронска мрежа била у стању да мапира улазне податке (нпр. подаци са сензора) на излаз (нпр. параметри актуатора).

МОДУЛИ



URDF

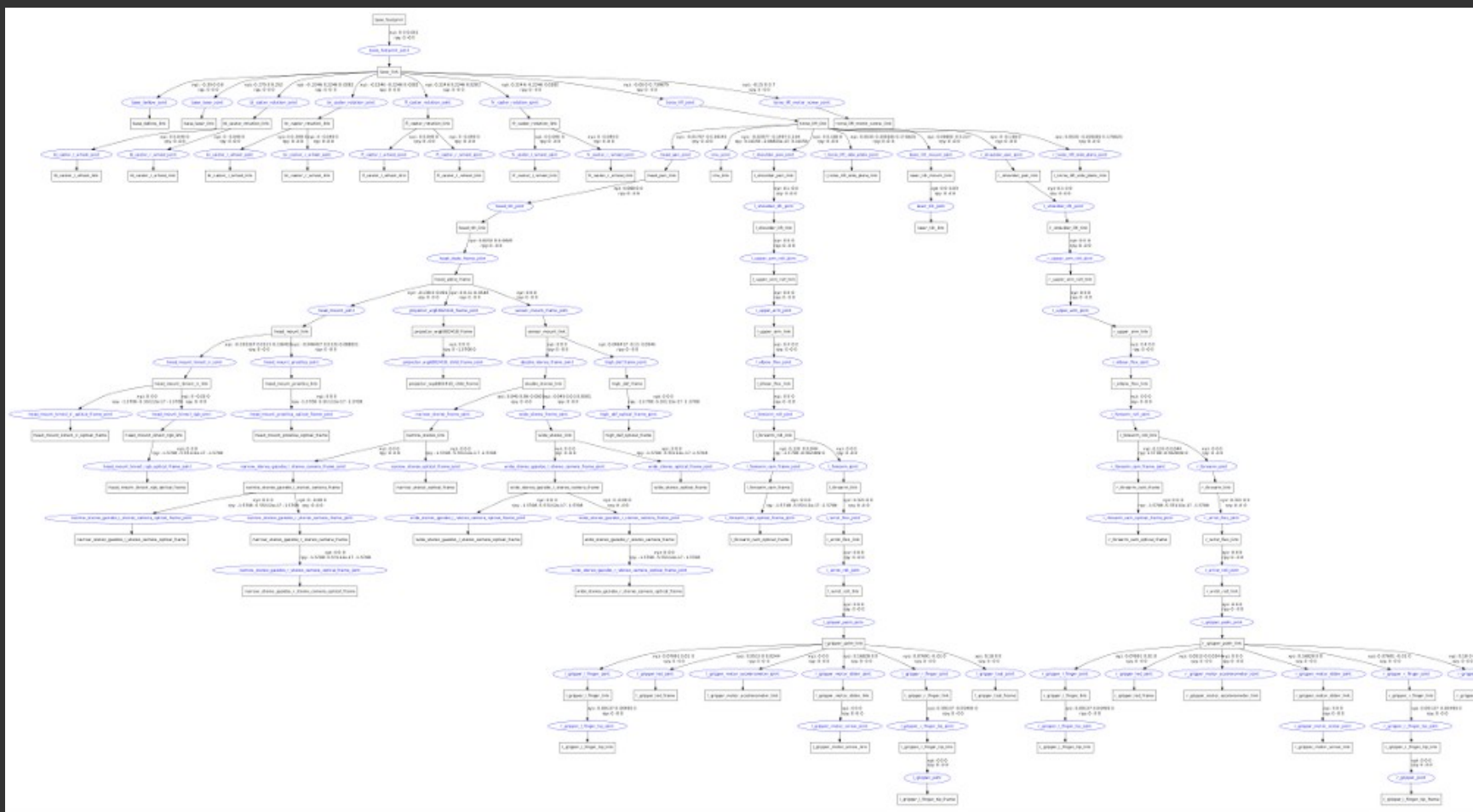
- Unified Robot Description Format (URDF)
- Стандардна ROS (Robot Operating System) XML репрезентација модела робота

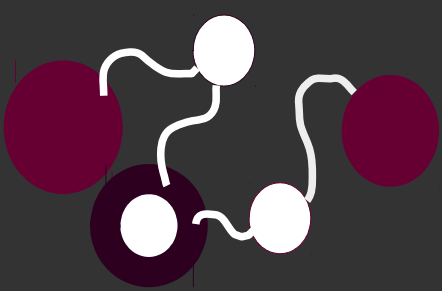


```
1 <robot name="test_robot">
2   <link name="link1" />
3   <link name="link2" />
4   <link name="link3" />
5   <link name="link4" />
6
7   <joint name="joint1" type="continuous">
8     <parent link="link1"/>
9     <child link="link2"/>
10  </joint>
11
12  <joint name="joint2" type="continuous">
13    <parent link="link2"/>
14    <child link="link3"/>
15  </joint>
16
17  <joint name="joint3" type="continuous">
18    <parent link="link3"/>
19    <child link="link4"/>
20  </joint>
21 </robot>
```

Парсирање URDF датотеке

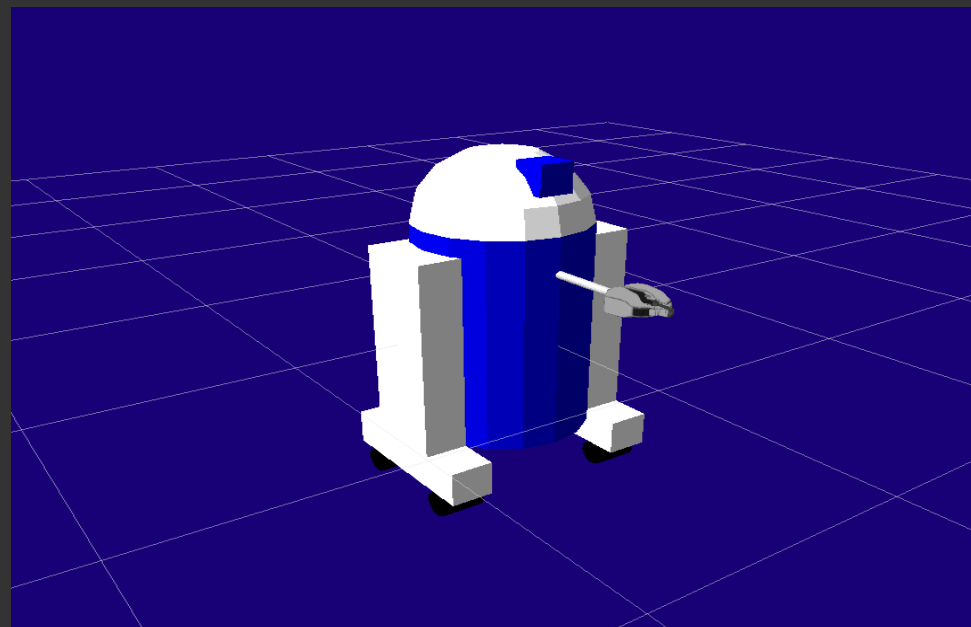
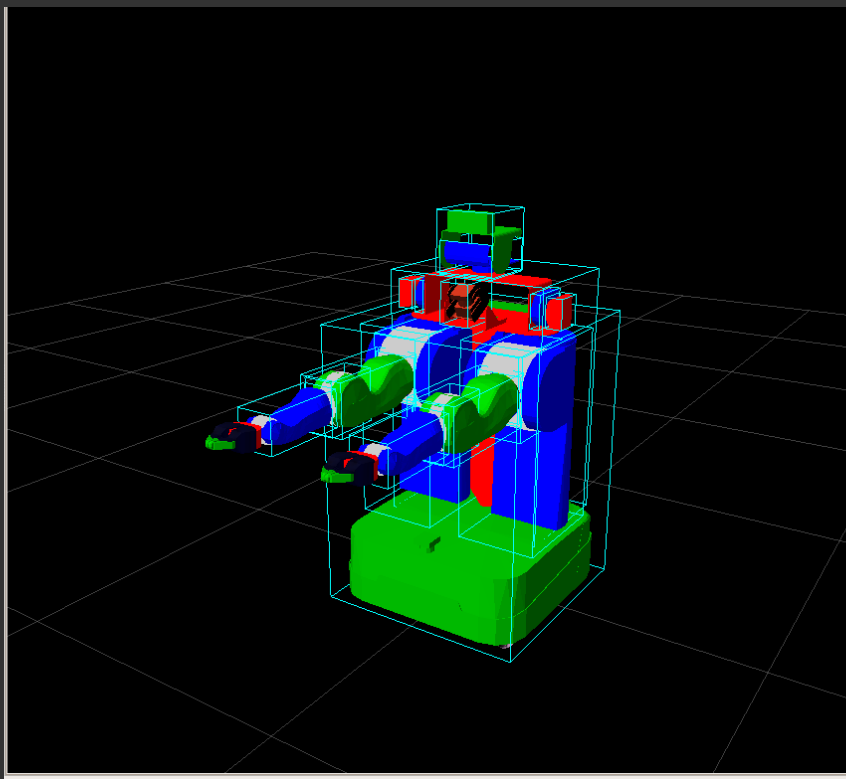
- Резултат парсирања је URDF модел
- Парсирање коришћењем класе `urdf::Model`

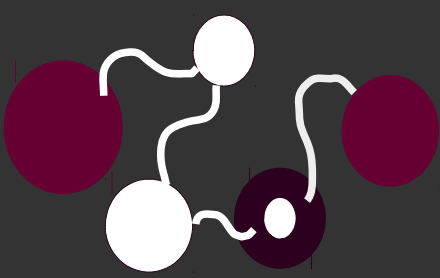




Презентовање података добијених из URDF датотеке

- Визуелна репрезентација URDF модела (RViz)





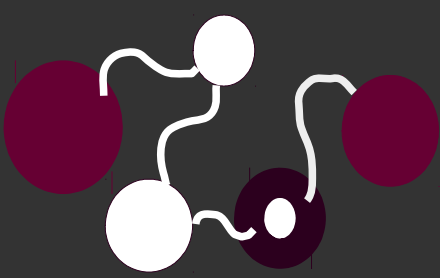
Обучавање неуронске мреже

Мотивација

- Дизајнирање кретања робота је тежак и временски захтеван посао за инжењере
- Дизајнирање се понавља сваки пут када се робот креира или модификује
- Аутоматско креирање начина кретања робота често даје боље резултате
- Коришћење еволутивних алгоритама је спорије али робустније од класичних метода (BP)

Еволутивни
алгоритми

NEAT



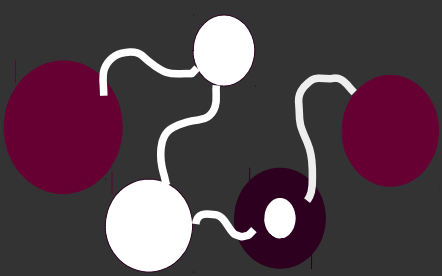
Обучавање неуронске мреже

- Инспирисани биолошком еволуцијом – мутација, селекција, укрштање
- Еволутивни алгоритми
 - Еволутивно програмирање
 - Фиксна структура програма
 - Нумерички параметри могу да еволуирају
 - Генетско програмирање
 - Решења у форми компјутерских програма
 - Фитнес се одређује на основу степена до ког је проблем решен
 - Фитнес (енг. Fitness) – искоришћеност хромозома
 - Генетски алгоритми
 - Представа решења проблема као низ цифара
 - Коришћење оператора рекомбинације, мутације
 - Често се користи за оптимизацију
 - Неуроеволуција
 - Слична генетском програмирању, с тим што геноми представљају вештачку неуронску мрежу описујући структуру и тежине.

Мотивација

Еволутивни
алгоритми

NEAT



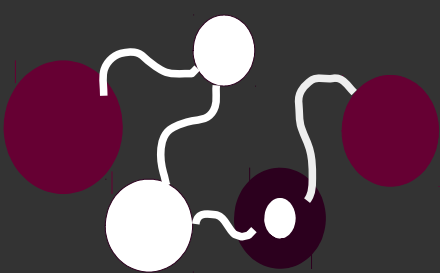
Обучавање неуронске мреже

Мотивација

Еволутивни
алгоритми

NEAT

- НЕУРОЕВОЛУЦИЈА
 - Облик машинског учења који користи еволутивне алгоритме за обучавање вештачке неуронске мреже
 - Имају ширу примену од алгоритама надгледаног учења
 - Задатак је еволуирати неуронску мрежу
 - Тежине и топологија се могу оптимизовати еволуцијом
 - Примена: рачунарске игре, еволутивна роботика
- ЗАШТО ЕВОЛУИРАТИ ТОПОЛОГИЈЕ?
 - Уштеда времена проналажењем тачног броја слојева/неурона
 - Добро дизајниран алгоритам може пронаћи глобално оптималну топологију



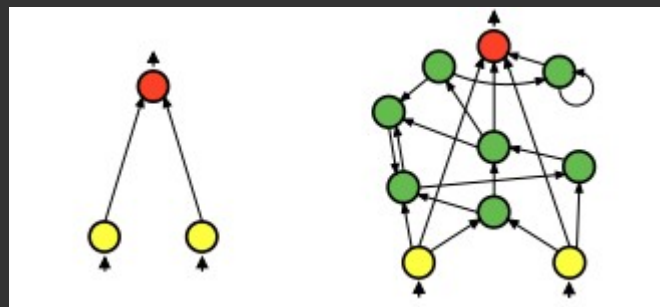
Обучавање неуронске мреже

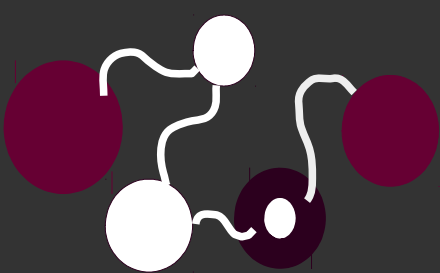
- NEAT – NeuroEvolution of Augmenting Topologies: Kenneth O. Stanley, 2001, The University of Texas at Austin
- Заснован на три основне идеје:
 - Праћење гена са историјским ознакама како би се омогућило укрштање само одговарајућих гена
 - Примена специјације (разврставања) ради очувања нових јединки. Омогућава преживљавање већих, структурно иновативних мрежа. Даје им времена да оптимизују своје тежине и покажу да је структурална иновација била ефективна
 - Инкрементални развој топологија од једноставних иницијалних структура ка сложенијим

Мотивација

Еволутивни
алгоритми

NEAT





Обучавање неуронске мреже

- Генетско кодирање
 - Геном – линеарна репрезентација веза у мрежи. Сваки геном укључује листу гена, а сваки се референцира на по два чвора које повезује.
 - Ген чворова даје информацију о улазима, скривеним чворовима и излазима који могу бити повезани.

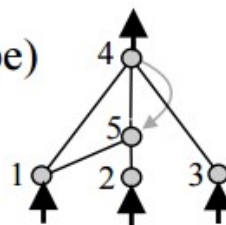
Мотивација

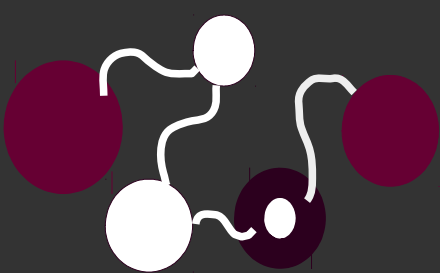
Еволутивни
алгоритми

NEAT

| Genome (Genotype) | | | | | | | |
|-------------------|------------|-----------------|------------|------------|------------|------------|------------|
| Node | Node 1 | Node 2 | Node 3 | Node 4 | Node 5 | | |
| Genes | Sensor | Sensor | Sensor | Output | Hidden | | |
| Connect. | In 1 | In 2 | In 3 | In 2 | In 5 | In 1 | In 4 |
| Genes | Out 4 | Out 4 | Out 4 | Out 5 | Out 4 | Out 5 | Out 5 |
| | Weight 0.7 | Weight -0.5 | Weight 0.5 | Weight 0.2 | Weight 0.4 | Weight 0.6 | Weight 0.6 |
| | Enabled | DISABLED | Enabled | Enabled | Enabled | Enabled | Enabled |
| | Innov 1 | Innov 2 | Innov 3 | Innov 4 | Innov 5 | Innov 6 | Innov 11 |

Network (Phenotype)





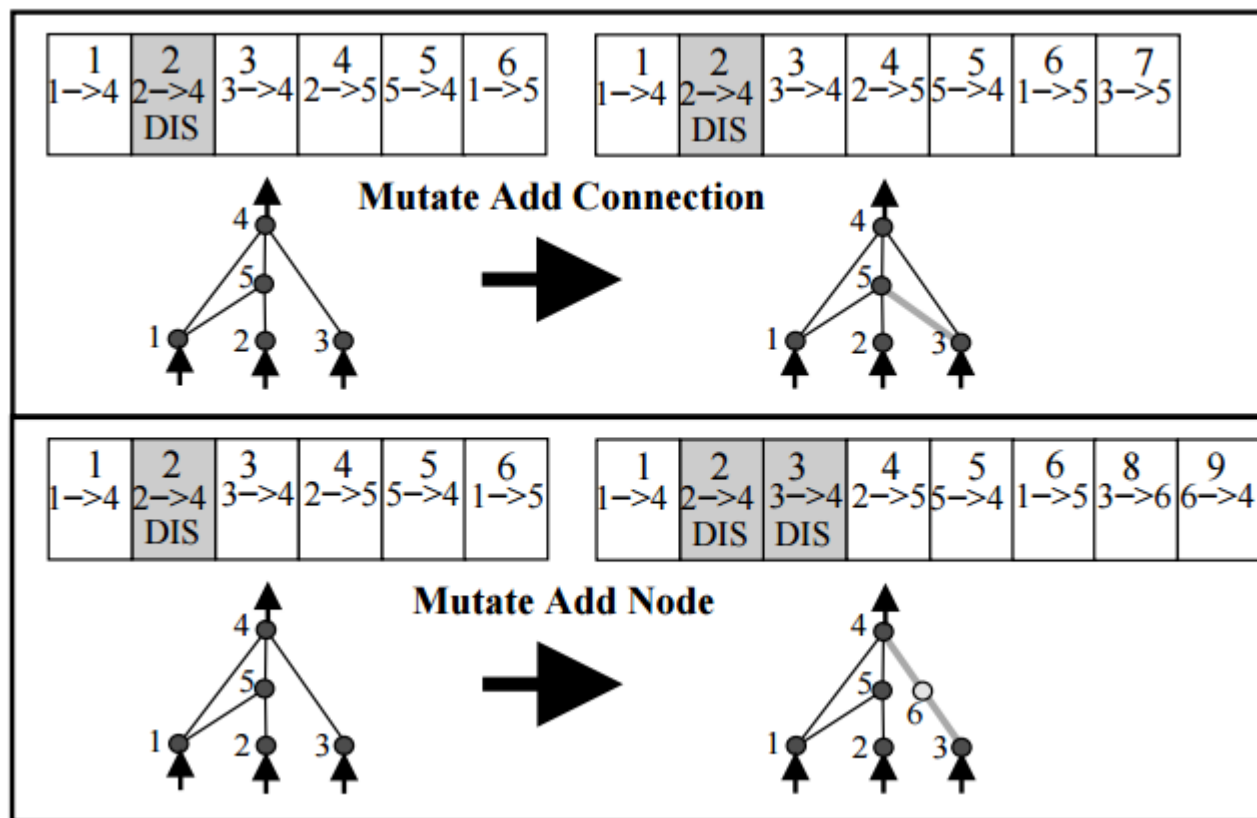
Обучавање неуронске мреже

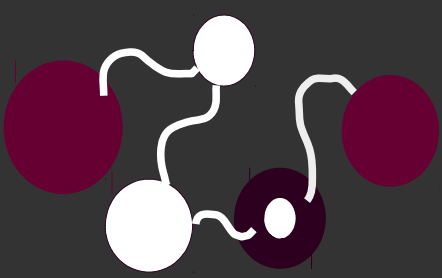
- Генетско кодирање
 - Мутација може да измени како структуру мреже (топологију) тако и тежине веза.

Мотивација

Еволутивни алгоритми

NEAT





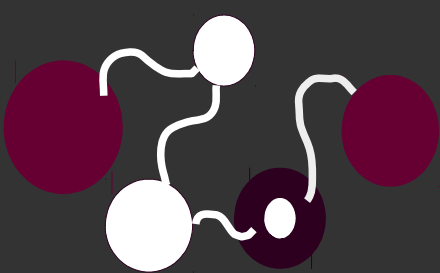
Обучавање неуронске мреже

- Праћење гена преко историјских ознака
 - Захтева мало рачуна
 - Глобални иновациони број се инкрементира са додавањем новог гена
 - Начин на који систем зна који гени се поклапају с којим
 - Приликом укрштања, гени у геномима са истим иновационим бројевима се упоређују
 - Ген се наслеђује случајно од неког родитеља
 - Спаривањем генома који представљају различите структуре, систем може да формира популацију различитих топологија
 - Нове топологије се штите специјацијом

Мотивација

Еволутивни
алгоритми

NEAT

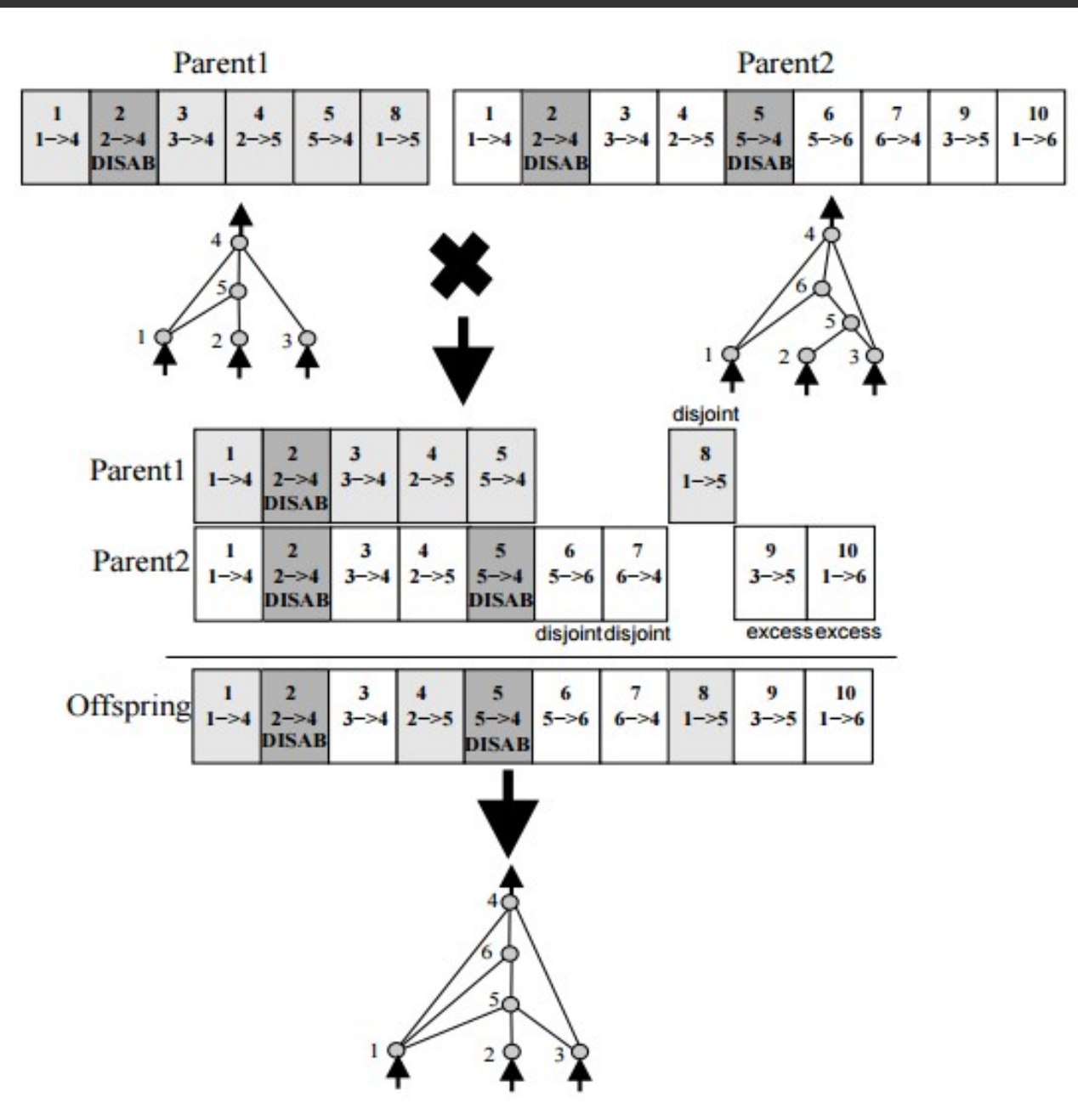


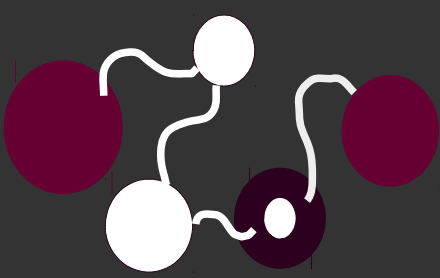
Обучавање неуронске мреже

Мотивација

Еволутивни алгоритми

NEAT





Обучавање неуронске мреже

- Заштита иновације кроз специјацију
 - Специјација популација омогућава организмима да се такмиче пре свега у својој околини, а не у оквиру целе популације
 - У околини, топологија има времена да оптимизује своју структуру
 - Број excess и disjoint гена између парова генома је мера њихове компатибилности. Већи број = већа некомпатибилност

$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \cdot \overline{W}$$

Делта – компатибилност

E – број excess гена

D – број disjoint гена

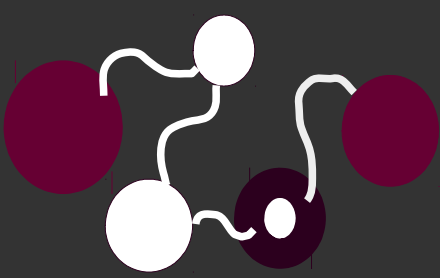
W – просечна разлика у тежинама између гена

N – број гена у геному (може бити сетован на 1 за мање геноме)

Мотивација

Еволутивни
алгоритми

NEAT



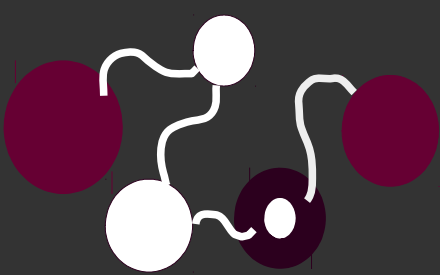
Обучавање неуронске мреже

- Заштита иновације кроз специјацију
 - Разврставање на основу делта
 - Геноми се у свакој генерацији разврставају
 - Свака врста је репрезентована случајним геномом из претходне генерације
 - Геном у текућој генерацији се смешта у ону врсту где је компатибилност са репрезентативним геномом задовољена
 - Спречава се преклапање врсте
 - Уколико није компатибилан ни са једним, креира се нова врста, а тај геном постаје репрезент

Мотивација

Еволутивни
алгоритми

NEAT



Обучавање неуронске мреже

- Заштита иновације кроз специјацију

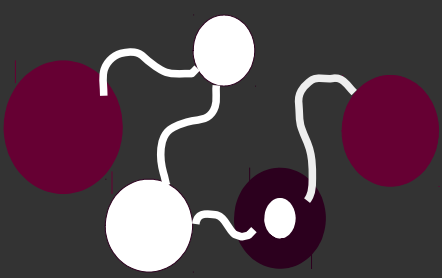
$$f'_i = \frac{f_i}{\sum_{j=1}^n \text{sh}(\delta(i, j))}$$

Мотивација

Еволутивни
алгоритми

NEAT

- Fitness за организам i се рачуна на основу делта растојања од сваког организма j у популацији.
- Sh – sharing
 - Поставља се на 0 када је растојање изнад неког threshold-а,
 - у супротном на 1
- Свакој врсти се додељује различит број потомака, пропорционално суми f_i организма који припадају тој врсти.
- Елиминишу се најлошији чланови популације
- Врши се укрштање



Обучавање неуронске мреже

- Кораци алгоритма:
 - Иницијализација популације
 - Израчунавање fitness за све јединке
 - Разврставање јединки
 - Подешавање fitness $f' = f / \text{species_size}$
 - Одређивање броја потомака за све врсте, пропорционално f'
 - Елиминисање најлошијих јединки тренутне генерације
 - Укрштање – замена тренутне генерације потомцима
 - Понављање од корака 2 док критеријум заустављања није задовољен

Мотивација

Еволутивни
алгоритми

NEAT



ПОСТОЈЕЋА РЕШЕЊА

Софтверска решења која обухватају наведени проблем

ПОСТОЈЕЋА РЕШЕЊА

- Webots – комерцијално решење. Омогућава моделовање, програмирање и симулирање робота. Имплементација је на кориснику.
- Rviz – 3D алат за визуелизацију за ROS
- Actin – учитавање CAD модела и аутоматско креирање контролног система чиме омогућава корисницима интерактивно експериментисање са симулираним роботом
- Gazebo – openSource. Симулација робота, генерисање података са сензора, рад са постојећим моделима робота или креирање нових модела.

ЛИТЕРАТУРА

1. <http://wiki.ros.org/ROS/Tutorials>
2. <http://wiki.ros.org/urdf/Tutorials>
3. <http://nn.cs.utexas.edu/?neuroevolution>
4. Competitive Coevolution through Evolutionary Complexification - Kenneth O. Stanley, Risto Miikkulainen
5. Evolving Neural Networks through Augmenting Topologies - Kenneth O. Stanley, Risto Miikkulainen
6. <http://nn.cs.utexas.edu/?neat>
7. <http://www.cs.ucf.edu/~kstanley/neat.html>