

Bc. Stanislava Leitmanová

**Aplikovanie doprednej neurónovej siete na riešenie
problému „The Artificial Ant“**

1. Zadanie projektu

Problém „The Artificial Ant“ (mravec - zberač potravy) spočíva v nájdení najlepšej stratégie pre pohyb agenta - mravca po toroidálnej mriežke $N \times N$, na ktorej sa nachádza rozmiestnených x kúskov potravy. Cieľom je za stanovený čas (dĺžka života agenta) pozbiarať čo najväčší počet kúskov potravy. Agent má obmedzenú oblasť vnímania, napr. okolie o veľkosti 3×3 .

Pohyb agenta po mriežke bude riadený pomocou doprednej neurónovej siete. Na vstupoch siete budú zakódované informácie o okolí agenta (oblasti vnímania) a výstup bude informácia, ktorým smerom sa má agent pohnúť (1 krok). Neurónová sieť bude trénovaná pomocou genetického algoritmu.

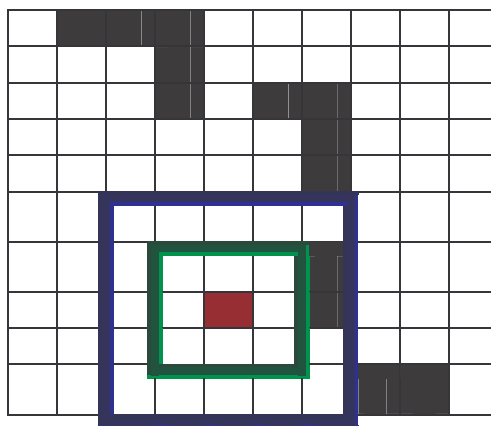
Cieľom projektu je implementovať túto neurónovú sieť a úspešnosť riadenia agenta pomocou siete overiť na konkrétnych príkladoch, napríklad na známom príklade „The Santa Fe Trail“. Skúmať môžeme napríklad schopnosť zlepšovania sa agenta pre rovnakú mriežku pre všetky generácie a rôznu mriežku pre každú novú generáciu. Môžeme použiť rôzne veľké okolia vnímania agenta a sledovať závislosť medzi veľkosťou okolia a úspešnosťou agenta.

2. Opis problému

Úlohou je ovládať pohyb mravca po mriežke tak, aby pozbieral čo možno najviac potravy za svoj život. Prostredie, v ktorom sa mravec pohybuje, je definované nasledovne [2]:

- Je rozdelené do toroidnej mriežky $N \times N$ zostavenej zo štvorcových buniek
- V prostredí existuje práve jeden mravec
- V každom časovom okamihu sa mravec nachádza vnútri práve jednej bunky
- Každá bunka buď obsahuje kúsok potravy alebo nie

Mravec vníma svoje bezprostredné okolie. Okolie vnímania môže byť dané vzdialenosťou od aktuálnej pozície mravca – d . Ak napríklad d je rovné jednej, agent vníma jedno najbližšie políčko okolo seba v každom smere, to znamená mriežku 3×3 , pričom mravec je v strede. Ak je d rovné 2, okolie vnímania tvorí mriežku 5×5 . Na Obr. 1 je zelenou farbou znázornené okolie vnímania s $d = 1$ a modrou farbou okolie s $d = 2$. Čierne políčka predstavujú bunky s potravou a červené políčko je mravec.



Obr. 1: Oblasť vnímania mravca

Mravec sa môže pohybovať štyrmi rôznymi smermi. Ak sa na políčku, na ktoré sa mravec premiestni, nachádza potrava, mravec ju zoberie.

3. Opis riešenia

3.1 Repräsentácia problému

Pozícia poľa v mriežke je daná usporiadanou dvojicou (x,y) , kde x je číslo riadku a y číslo stĺpca, pričom číslovanie sa začína v ľavom hornom rohu.

	1	2	3	4	5	6	7	8	9	10
1		■	■	■						
2				■						
3				■		■	■			
4							■			
5							■			
6		U								
7	L	■	R				■			
8		D					■			
9										
10								■	■	

Obr. 2: Repräsentácia mriežky

Celá mriežka je reprezentovaná jedným zoznamom dĺžky $N \times N$, ktorého prvky nadobúdajú hodnoty *true* alebo *false*, podľa toho, či sa na príslušnej bunke nachádza potrava alebo nie. Ak máme bunku s pozíciou (x,y) , potom pozíciu (index) v zozname vypočítame ako $(x-1) \cdot N + y$.

Pohyb agenta po mriežke je daný štyrmi operátormi: U (hore), D(dole), R (doprava), L (doľava), znázornenými na Obr. 2. Tieto operátory menia pozíciu agenta nasledovne:

$$U : (x, y) \rightarrow (x - 1, y)$$

$$D : (x, y) \rightarrow (x + 1, y)$$

$$R : (x, y) \rightarrow (x, y + 1)$$

$$L : (x, y) \rightarrow (x, y - 1)$$

Binárne operátory sčítania a odčítania musia zachovávať rozmery mriežky tak, aby sa javila ako toroidná. To znamená, že pre operátory „plus“, „minus“ a rozsah mriežky N platí:

$$x \langle \text{plus} \rangle 1 \rightarrow 1, \text{ ak } x = N$$

$$\rightarrow x + 1, \text{ inak}$$

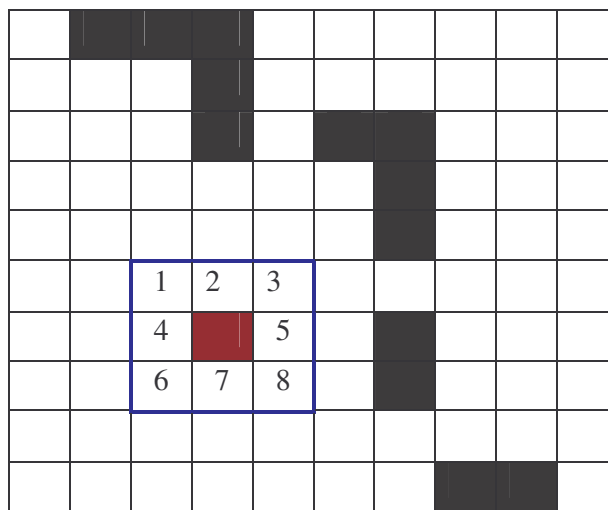
$$x \langle \text{minus} \rangle 1 \rightarrow N, \text{ ak } x = 1$$

$$\rightarrow x - 1, \text{ inak}$$

Cesta agenta po mriežke je potom daná nejakou postupnosťou operátorov U, D, R, L a jeho úspešnosť je daná počtom kúskov potravy, ktoré agent zozbiera. Pre lepšie porovnávanie úspešnosti je vhodnejšie použiť relatívnu úspešnosť, kde počet zozbieraných kúskov ešte podelíme počtom všetkých kúskov potravy v danej mriežke.

Vnímanie agenta

Agent vníma svoje okolie do vzdialenosti d. Vnem agenta sa pretransformuje do poľa tak, že prvky poľa nadobúdajú hodnoty 0 alebo 1, podľa toho, či sa na konkrétnom políčku nachádza potrava alebo nie.



Obr. 3: Reprezentácia vnímania agenta

Na Obr. 3 je znázornený spôsob očíslovania políček, ktoré vidí agent. Z toho sa potom spraví pole, v tomto prípade sú všetky hodnoty rovné 0, pretože sa v okolí nenachádza žiadna potrava.

3.2 Neurónová sieť

Neurónová sieť obsahuje vstupné, skryté a výstupné neuróny. Počet vstupných neurónov je daný počtom políček, ktoré agent vidí. Každému políčku zodpovedá jeden vstupný neurón. Neuróny na vstupe teda nadobúdajú jednu z hodnôt 0 alebo 1.

Počet neurónov skrytej vrstvy je možné meniť.

Výstup neurónovej siete predstavujú dva výstupné neuróny. Hodnoty na výstupe sú v intervale (0,1) a po zaokrúhlení dostaneme dva bity, ktoré predstavujú príkaz pre agenta, ktorý určuje, ktorým smerom sa má agent pohnúť. Tieto príkazy sú mapované nasledovne:

- (1 1) – U - move up
- (1 0) – L - move left
- (0 1) – R - move right
- (0 0) – D - move down

Riadenie agenta pomocou siete si môžeme predstaviť, ako zmenu stavov, pričom táto zmena je daná použitím niektorého príkazu pre posunutie agenta a teda zmenou súradníc agenta. V jednotlivých stavoch sa samozrejme mení aj mriežka a to tak, že z nej odbúďajú kúsky potravy, ktoré agent zozbral.

3.3 Evolučný algoritmus

Reprezentácia a fitness jedinca

Jedinca predstavuje reťazec, ktorý tvoria váhy prepojení medzi neurónmi danej neurónovej siete. Váhy sú zoradené do reťazce v nasledovnom poradí:

$$(\overline{v_1}, \dots, \overline{v_k}, \overline{w_1}, \dots, \overline{w_i}),$$

Kde v_1, \dots, v_k sú vektory váh do k-teho skrytého neurónu a w_1, \dots, w_i do i-teho výstupného neurónu.

Váhy sú dané reálnymi číslami a aj jedinec je teda reprezentovaný reťazcom s reálnymi hodnotami.

Fitness

Pri riešení boli brané do úvahy dva spôsoby vyhodnocovania úspešnosti jedincov.

Základná fitness funkcia pre jedinca predstavuje počet kúskov potravy, ktoré agent za svoj život zozbiera. Počet zozbieraných kúskov potravy jedinca určíme tak, že riadime pohyb agenta po danej mriežke pomocou neurónovej siete, ktorej váhy sú dané týmto jedincom.

Pri určení úspešnosti jedincov však môžeme brať do úvahy aj to, do akej miery sa výstupy siete líšia od hodnôt 0 a 1, to znamená, do akej miery si je sieť „istá“ svojim výstupom. Môžeme definovať premennú *chyba výstupu*, ktorá bude určovať rozdiel medzi skutočným výstupom siete a zaokrúhlenou hodnotou. Keďže sieť má dva výstupné neuróny, bude to suma takýchto chýb pre každý neurón. Je zjavné, že najväčšia hodnota chyby bude 1 a to vtedy, keď obidve hodnoty budú rovné 0.5. Priemernú chybu výstupu určíme ako priemernú hodnotu takýchto chýb pre každý krok siete (počet krokov je daný počtom jednotiek života agenta). Výslednú fitness potom vypočítame ako: $\text{food_eaten} + k \cdot \text{output_error}$, kde food_eaten je počet kúskov potravy, ktoré agent zoberie, output_error je priemerná chyba výstupu a k je koeficient, ktorý udáva mieru, akou sa výstupná chyba podieľa na hodnote fitness. Tento koeficient by mal nadobúdať nízke hodnoty v pomere k celkovému počtu potravy v mriežke, aby sa nestalo, že budú uprednostnení tí jedinci, pri ktorých sa sieť síce správa „rozhodne“, no celkovo zozbierajú menej potravy.

Mutácia a kríženie

Mutácia jedinca predstavuje zmenu prvku v reťazci. Keďže reťazec je zložený z reálnych čísel, mutácia je gaussovská. To znamená, že k hodnote prvku sa pripočíta číslo z normálneho rozdelenia $N(0,k)$, kde k je nejaký koeficient. Pri výpočtoch bolo použité rozdelenie $(0,1)$.

Kríženie jedincov je implementované ako štandardné jednobodové kríženie.

Algoritmus

Evolučný algoritmus pozostáva z týchto krokov:

2. Náhodne sa vygeneruje prvá generácia jedincov
3. Ohodnotia sa jedinci v súčasnej generácii
4. a) Ak medzi jedincami existuje taký, ktorý spĺňa ukončovaciu podmienku, alebo ak sa dosiahne maximálny počet generácií, tak algoritmus skončí a výsledkom je najlepší jedinec
b) Ak nie je splnená žiadna ukončovacia podmienka, ide sa na krok 4
5. Vytvorí sa nová generácia (podľa pravidiel popísaných vyššie) a prejde sa na krok 3.

Algoritmus tvorby novej generácie je nasledovný:

1. kvázináhodne sa vyberú dvaja jedinci na prežitie
2. náhodne sa vygeneruje číslo p z intervalu $(0,1)$

3. na základe pravdepodobnosti reprodukcie p_{repro} a vygenerovaného čísla p :
 - a) ak $p \leq p_{\text{repro}}$, nastáva reprodukcia, vzniknú dvaja noví jedinci a pridajú sa do novej generácie
 - b) ak $p > p_{\text{repro}}$, reprodukcia nenastane a títo jedinci sa bez zmeny pridajú do novej generácie
4. algoritmus končí, keď počet jedincov v novej generácii sa rovná počtu jedincov v pôvodnej generácii.

Kvázináhodný výber jedincov znamená, že jedinci sú náhodne vyberaní z populácie s pravdepodobnosťou priamo úmernou veľkosti ich fitness.

4. Výsledky

Experimenty boli vykonané s rôznymi počiatočnými nastaveniami a s dvoma rôznymi mriežkami. Prvá generácia jedincov v experimentoch je vždy vygenerovaná náhodne, pričom váhy sú generované ako čísla z intervalu $(-1,1)$. Výsledné grafy predstavujú závislosť priemernej fitness jedincov od počtu generácií.

4.1 Mapa 10x10

V týchto experimentoch bola použitá mapa z Obr. 1. Je to mapa s rozmermi 10x10 a je na nej umiestnených 13 kúskov potravy. Život agenta je 17 jednotiek času.

Počiatočná pozícia agenta bola vždy daná súradnicami (1,1).

Experiment 1

V experimente sa porovnával priebeh fitness funkcie v závislosti od rozsahu videnia agenta. Použité boli dve rôzne okolia videnia – 3x3 a 5x5. Počiatočné nastavenia parametrov sú dané nasledujúcou tabuľkou:

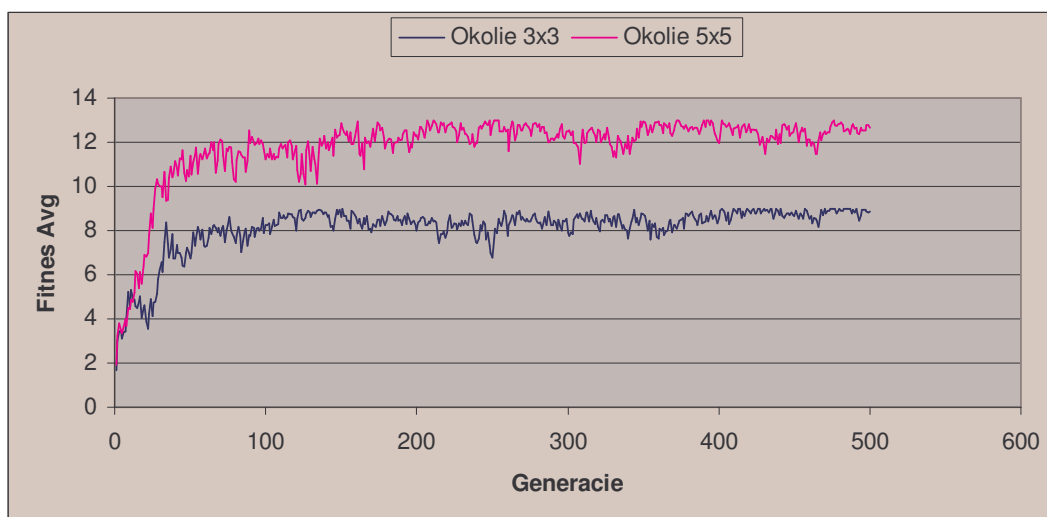
Počet jedincov v generácii	50
Počet generácií	500
Doba života agenta	17
Počet políčok, ktoré agent vidí	3x3, 5x5
Pravdepodobnosť reprodukcie	0.8
Pravdepodobnosť mutácie	0.1
Pravdepodobnosť kríženia	0.3
Počet skrytých neurónov	5

Pre obidve okolia bol algoritmus spustený 20 krát. Pri použití okolia 3x3 bol najlepší jedinec taký, čo zozbieral 9 kúskov potravy. Pre toto okolie ani nie je možné nájsť jedinca, ktorý by pozbieral všetku potravu. Je to kvôli tomu, že v mriežke existujú pozície, pri ktorých je rovnaké okolie videnia, ale je potrebné sa pohnúť rôznymi smermi. Príkladom sú bunky so súradnicami (3,4) a (5,7) z Obr. 2, kde agent vidí okolo seba rovnaké okolie (predpokladáme, že agent sa do tých pozícií dostal po vyznačenej ceste, t.j. zobral príslušné kúsky potravy), no v prvom prípade sa musí pohnúť doprava a v druhom dole.

Pri použití okolia 5x5 sa vždy dosiahol správny výsledok, t.j., agent pozbieral všetkých 13 kúskov potravy. Priemerné číslo generácie, v ktorej sa dosiahol tento jedinec (pre 20 behov evolúcie) je 9.6.

Našli sa traja takíto jedinci, ktorí pozbierajú všetkých 13 kúskov potravy. Príklad cesty jedného z nich je daný nasledovnou postupnosťami operátorov: R R R D D R R R D D D D D D R D R.

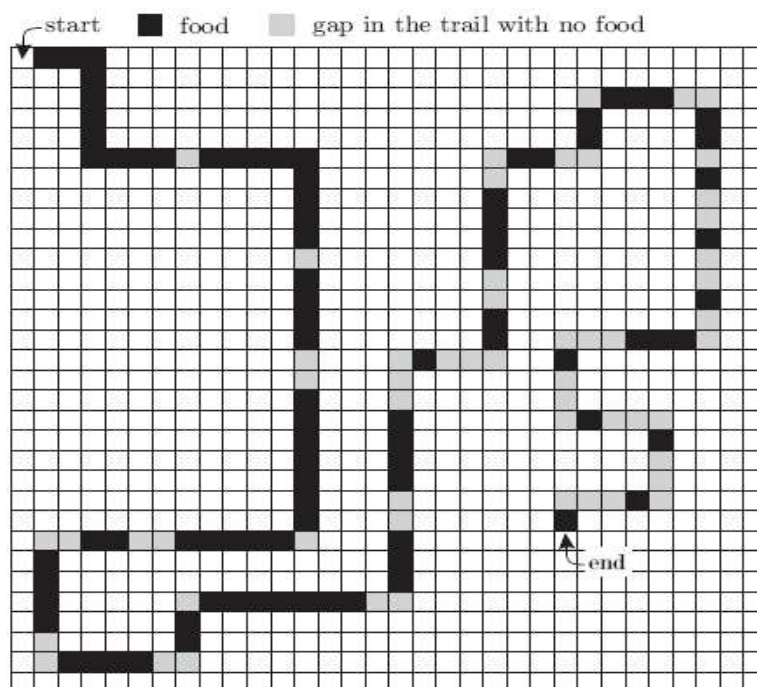
Po skontrolovaní podľa Obr. 2 môžeme povedať, že v danej mriežke je tento jedinec natrénovaný ideálne. Na Obr. 4 je znázornený graf priebehu priemernej fitness populácie v čase. Parametre merania boli tie isté ako uvedené v tabuľke vyššie, pre dve rôzne okolia videnia agenta. Z grafu je vidno, že pre okolie 3x3 priemerná fitness konverguje k hodnote 9 a pre okolie 5x5 k hodnote 13.



Obr. 4: Experiment 1

4.2 Mapa 32x32

V týchto experimentoch bola použitá mapa z Obr. 5. Je to mapa s rozmermi 32x32 a je na nej umiestnených 89 kúskov potravy. Limit pre získanie všetkých kúskov potravy je 200 jednotiek času. Na obrázku sú čiernou farbou vyznačené políčka s potravou a šedou farbou optimálna cesta mravca.



Obr. 5: Santa Fe trail

Počiatočná pozícia agenta bola vždy daná súradnicami (1,1).

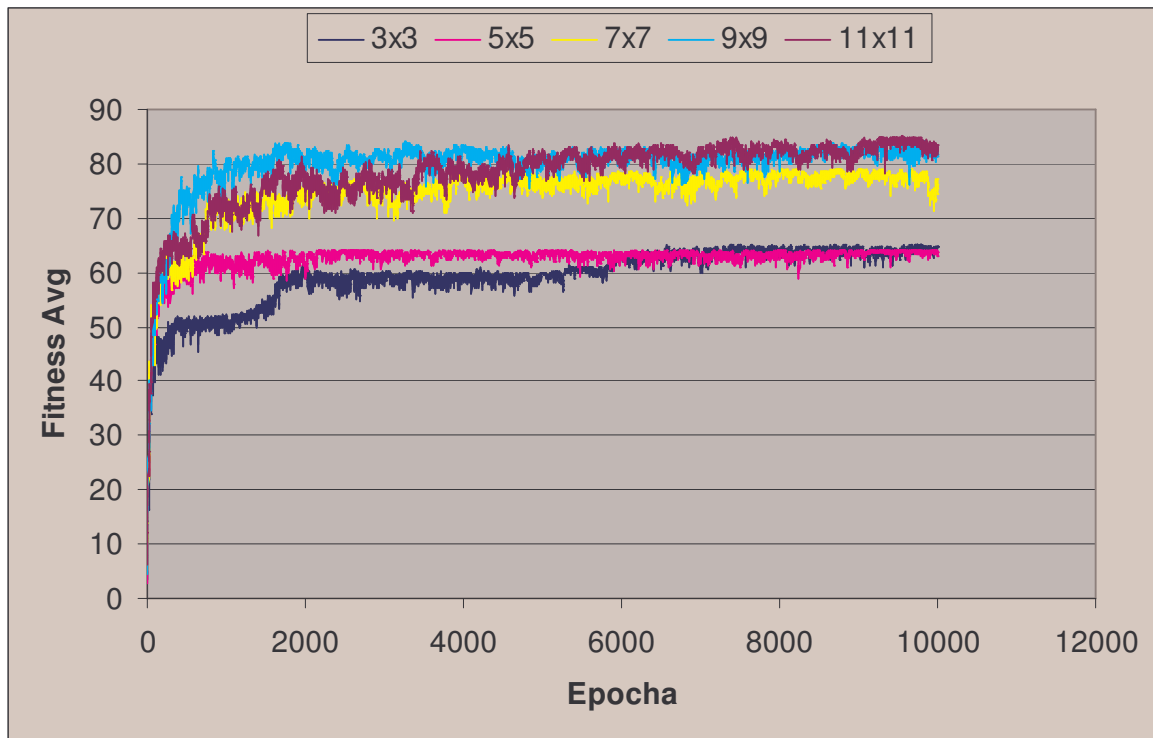
Experiment 2

V tomto experimente sa zameriavame na skúmanie priebehu fitness v závislosti od veľkosti okolia, pričom ostatné parametre sa nemenia.

Počiatočné nastavenia parametrov sú dané nasledujúcou tabuľkou:

Počet jedincov v generácii	100
Počet generácií	10000
Doba života agenta	200
Počet políčok, ktoré agent vidí	3x3, 5x5, 7x7
Pravdepodobnosť reprodukcie	0.8
Pravdepodobnosť mutácie	0.1
Pravdepodobnosť kríženia	0.3
Počet skrytých neurónov	5

Nasledujúci graf znázorňuje priebeh priemernej fitness populácie v čase pre rôzne okolia videnia agenta.



Obr. 6: Experiment 2

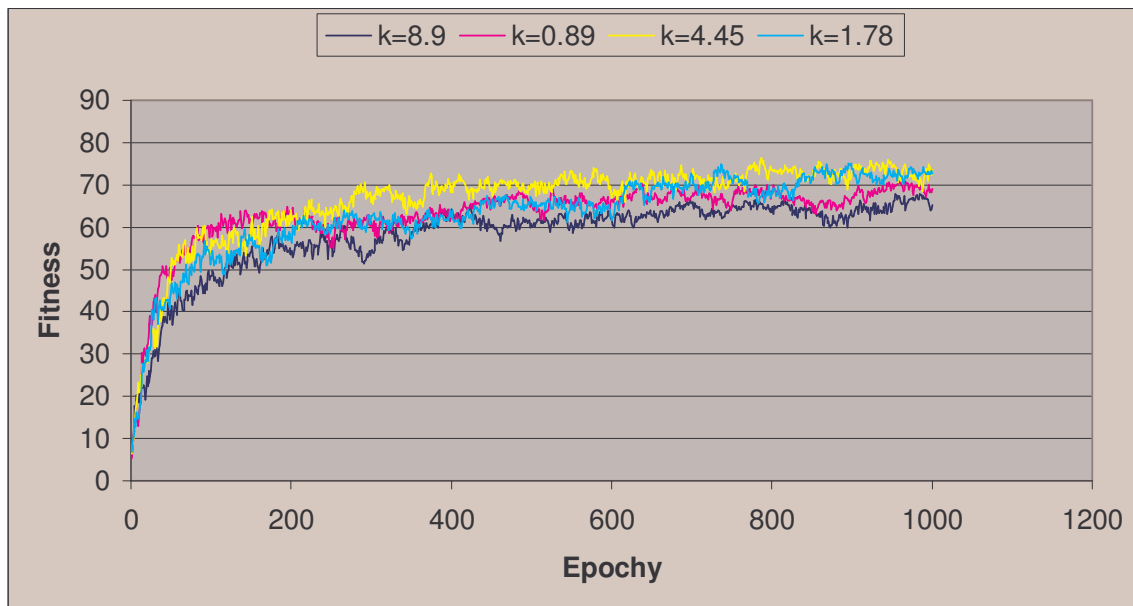
Z grafu je vidno, že pri okolí 3x3 a 5x5 fitness konverguje približne k hodnote 65, preto tieto okolia nie sú vhodné pre riešenie mapy Santa-Fe. Nasledujúca tabuľka zobrazuje priemernú fitness v poslednej generácii a taktiež fitness najlepšieho jedinca v tejto generácii

	3x3	5x5	7x7	9x9	11x11
Fitness avg	64.38	63.04	74.34	81.96	83.13
Fitness best	65	64	79	84	85

Najlepší jedinec, ktorý sa našiel evolúciou, dokázal pozbierať 85 kúskov potravy.

Experiment 3

V tomto experimente sa skúma vplyv spôsobu počítania fitness na úspešnosť jedinca. Vzorec pre výpočet fitness je $:\text{food_eaten} + k \cdot (1 - \text{error_avg})$. Veľkosť okolia pre všetky behy evolúcie bola nastavená na 7x7 a ostatné parametre boli rovnaké ako v predchádzajúcom experimente. Menil sa jedine parameter k pri výpočte fitness. Graf znázorňuje priebeh priemernej úspešnosti agentov v čase, pričom táto úspešnosť sa určí ako priemerný počet zbraných kúskov potravy.



Obr. 7: Experiment 2

Z grafu je vidieť, že nie sú vhodné ani príliš malé, ani príliš veľké hodnoty koeficientu k . Zaujímavé je, že pri koeficiente 0.89 sa našli v priebehu evolúcie aj jedinci s úspešnosťou 84 ale potom sa tieto riešenia stratili a na konci evolúcie bol najlepší jedinec s úspešnosťou 74.

5. Zhodnotenie

Úlohou tohto zadania bolo navrhnuť a implementovať neurónovú sieť optimalizovanú pomocou genetického algoritmu. Táto sieť sa má použiť pre riešenie problému mravca pohybujúceho sa po mriežke a zbierajúceho potravu.

Pri implementácii som použila štandardný genetický algoritmus s výberom jedincov pomocou rulety, s gaussovou mutáciou parametrov a s jednobodovým krížením.

Program bol implementovaný v jazyku C++ a bol otestovaný na dvoch mriežkach. Jedna s rozmermi 10x10 a druhá s rozmermi 32x32.

Pre prvú mriežku sa viackrát podarilo nájsť „ideálneho“ agenta, t.j. takého, ktorý pozbieral za daný počet krokov všetky kúsky potravy. Pri experimentoch som zistila, že nie je veľmi vhodné používať malý rozsah okolia, ktoré agent vidí. Pri okolí 3x3 sa napríklad ani raz nepodarilo nájsť správne riešenie.

Pre druhú mriežku bolo vykonaných tiež niekoľko experimentov s rôznymi parametrami. Pri porovnaní vplyvu veľkosti videneho okolia na úspešnosť jedincov je evidentné, že väčšie okolie dáva lepšie výsledky a takisto aj riešenia rýchlejšie konvergujú. Okrem toho sa skúmal aj vplyv rôznych výpočtov fitness funkcie na úspešnosť. Použitý vzorec na výpočet fitness však vo všeobecnosti nedával lepšie výsledky a vývoj úspešných jedincov v čase bol menej stabilný.

Pre problém Santa-Fe sa ani raz nepodarilo nájsť agenta, ktorý by pozbieral všetky kúsky potravy. Najlepší agent mal úspešnosť 85 z 89. Je pravdepodobné, že toto bolo spôsobené hlavne obmedzením evolúcie na maximálny počet generácií, keďže tieto výpočty sú veľmi časovo náročné. No v experimentoch bolo vidieť zvyšovanie fitness v priebehu evolúcie a najlepší nájdený jedinec mal skóre 85 kúsok potravy z 89.

6. Zoznam použitej literatúry

- [1] Kvasnička, V., Pospíchal, J., Tiňo, P.: Evolučné algoritmy, Bratislava: STU, 2000, 1. vydanie, 223 s., ISBN 80-227-1377-5.
- [2] Weise, T.: Global Optimization Algorithms – Theory and Application, 2008.
<http://www.it-weise.de/projects/book.pdf>
- [3] Návrat, P. et al.: Umelá Inteligencia. Bratislava: STU, 2002. 405s. ISBN 80-227-1645-6