

Peter Kasan

Použitie Genetického Algoritmu na neurónovú sieť
Evolučné algoritmy

Zadanie

Pre toroidálnu mriežku $N \times N$ vytvorte zberača potravy (táto sa nachádza náhodne rozmiestnená (x kúskov potravy) v tejto mriežke), pričom tento zberač je riadený neurónovou sieťou. Navrhните túto sieť a vylepšujte zberača pomocou genetického algoritmu (váhy siete tvoria chromozóm). Zberač má obmedzenú oblasť vnímania. Skúmajte vývoj zberačov v mriežke vzhľadom na schopnosť získať čo najviac kúskov potravy pre nejaký maximálny počet krokov v mriežke. Skúmajte schopnosť zlepšovania pre rovnakú mriežku pre všetky generácie a rôznu mriežku pre každú novú generáciu. Príklad: mriežka veľkosti 20×20 , počet kúskov potravy 50, maximálny počet krokov 50, oblasť vnímania zberača 5×5 kde zberač je v centre. Zadanie vychádza z problému popísanom v knihe: Kvasnička, Pospíchal: Evolučné algoritmy (str. 124-125).

Úvod

Táto práca sa zaoberá skúmaním akým spôsobom sa agent správa v prostredí, v ktorom hľadá potrebné zdroje. Skúma sa porovnanie medzi agentom realizovaným klasickým chromozómom, ktorého jednotlivé gény tvoria možné akcie v prostredí a agentom, ktorý je riadený viacvrstvou doprednou sieťou. Skúmajú sa aspekty „rýchlosť“ a „množstvo“ získavania zdrojov. Rýchlosť získavania určuje koľko generácií je potrebných na to aby sa agent vyvinul na jedinca, ktorý dokáže zozbierať X zdrojov. Množstvo zdrojov určuje aký počet zdrojov je agent schopný získať po Y generáciách vývoja. Prostredie je mriežka $N \times N$, v ktorej sa nachádza náhodne (určitým spôsobom) rozmiestnených X zdrojov (kúskov potravy). Agent má v prostredí stanovený určitý počet krokov, ktoré môže využívať na získavanie zdrojov. Pre porovnanie agenta zapísaného klasickým chromozómom a agenta zapísaného chromozómom kódujúcim neurónovú sieť sa realizujú prípady stabilného prostredia pre všetky generácie. Pre agenta, ktorého chromozóm predstavuje zapísanú neurónovú sieť sa skúma aj schopnosť získavania zdrojov pre meniace sa prostredie pre každú novú generáciu. Takisto sa tento agent porovnáva s agentom riadeným neurónovou sieťou, ktorá bola trénovaná klasickým spôsobom (pomocou tréningových dát metódou *backprop*).

Popis použitej metódy

Všeobecný popis použitého Genetického algoritmu

Ako už bolo spomenuté v úvode budú v tejto štúdií vystupovať tri typy agentov: agent s klasickým chromozómom, agent s chromozómom kódujúcim váhy neurónovej siete a agent s klasicky trénovanou neurónovou sieťou. V prípade prvých dvoch agentov je chromozóm nasledujúci:

$$P = \{\alpha_1, \alpha_2, \dots, \alpha_p\} \subseteq \{a, b, \dots\}^k$$

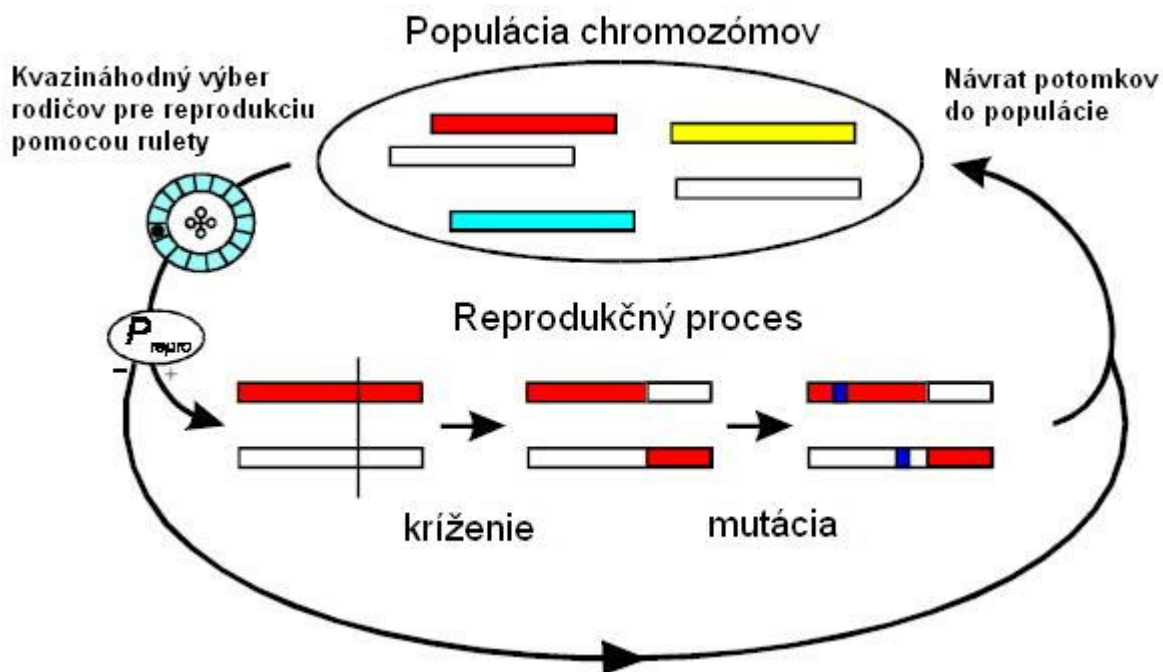
kde P je populácia chromozómov.

Chromozóm predstavuje genotyp jedinca, pričom jeho realizácia (ohodnotenie) v prostredí (mriežke) predstavuje fenotyp jedinca.

V prípade agenta s klasickým chromozómom je $k = \text{maximálny počet krokov v mriežke}$. Množina $\{a, b, \dots\}$ je tvorená štyrmi pohybovými akciami (pohyb hore, pohyb dole, pohyb doprava, pohyb doľava).

V prípade agenta s chromozómom kódujúcim váhy neurónovej siete je $k = \text{všetky váhové prepojenia v sieti}$ a pre množinu $\{a, b, \dots\}$ platí že a, b, \dots sú reálne čísla.

Realizáciu algoritmu možno vidieť na Obr. 1. Pseudokód tohto algoritmu je na Obr. 2.



Obr. 1 Grafické znázornenie použitého genetického algoritmu. Obrázok prevzatý z [1].

Vstupom algoritmu je maximálny počet generácií koľko sa môže populácia vyvíjať. Výstupom je najlepší jedinec poslednej populácie. Algoritmus sa začína nastavením čísla aktuálnej generácie na 0 a zastavovacieho kritéria na false (ak takéto existuje). Pri experimentoch v tejto práci nebolo uvedené špecifické zastavovacie kritérium. V ďalšom kroku sa vygeneruje začiatková populácia (v pseudokóde (a), bude popísané ďalej). Následne prebieha cyklus vývoja populácií až kým sa nedosiahne maximálny počet generácií definovaný ako vstup alebo nebudú splnené zastavovacie kritéria (nie sú použité). Jeden cyklus vývoja (jedna generácia) pre populáciu prebieha nasledovne: každý jedinec v aktuálnej populácii P sa ohodnotí fitness funkciou (v pseudokóde (b), bude popísané ďalej). Pre takto ohodnotenú populáciu sa realizuje pseudonáhodný výber dvoch jedincov z populácie (v pseudokóde (c), bude popísané ďalej) ich následná reprodukcia (v pseudokóde (d), bude popísané ďalej) a vloženie ich potomkov do novej populácie Q . Tento výber sa realizuje dovtedy kým veľkosť novej populácie nieje rovná veľkosti aktuálnej populácie ($|P| = |Q|$). Následne sa nová populácia stáva aktuálnou populáciou. Tu môže prísť k vyhodnocovaniu zastavovacích kritérií ak existujú.

```

procedure Genetic_Algorithm(input  $t_{\max}$ ; output  $\alpha_{\text{opt}}$ );
begin  $t:=0$ ; stop_criterion:=false;
       $P:=\{\text{randomly generated population of chromosomes}\}$ ; (a)
      while ( $t < t_{\max}$ ) and (not stop_criterion) do
        begin  $t:=t+1$ ;  $Q:=\emptyset$ ;
          each chromosome is evaluated by fitness; (b)
          while  $|Q| < |P|$  do
            begin select by roulette wheel two (c)
              chromosomes  $\alpha_1, \alpha_2 \in P$ ;
              if random  $< P_{\text{repro}}$  then
                Reproduction( $\alpha_1, \alpha_2, \alpha'_1, \alpha'_2$ ) else (d)
                  begin  $\alpha'_1 := \alpha_1$ ;  $\alpha'_2 := \alpha_2$  end;
                   $Q := Q \cup \{\alpha'_1, \alpha'_2\}$ ;
            end;
           $P := Q$ ;
          if convergence criteria are fulfilled
            then stop_criterion:=true;
        end;
       $\alpha_{\text{opt}} := \text{best chromosome of } P$ ;
end;

```

Obr. 2 Pseudoalgorithmus použitého GA (prevzatý z [1]).

Počiatočná inicializácia populácie (a)

V prípade agenta popísaného štandardným chromozómom spočíva iniciálne vygenerovanie populácie vo vytvorení X chromozómov, kde X je veľkosť populácie, pričom jednotlivé gény chromozómu sa náhodne inicializujú niektorou akciou agenta (pohyb hore, pohyb dole, pohyb doprava, pohyb doľava).

V prípade agenta, ktorého chromozóm kóduje váhové prepojenia siete spočíva náhodná inicializácia vo vytvorení X chromozómov, kde X je veľkosť populácie, pričom jednotlivé gény chromozómu sa náhodne inicializujú z intervalu $(-a, a)$ to znamená zo symetrického intervalu. Toto vychádza z teórie neurónových sietí, kde sa pri tréňovaní sieť inicializuje váhami zo symetrického intervalu.

Ohodnotenie chromozómov funkciou fitness (b)

Funkcia fitness predstavuje pre všetky typy agentov množstvo zozbieranej potravy na mriežke (mriežkach). V prípade vývoja na mriežke rovnakej pre všetky generácie (platí pre agenta s klasickým chromozómom aj pre agenta s chromozómom kódujúcim váhové koeficienty siete) sa nechajú agenti pohybovať po tejto mriežke (každý z nich má X počet krokov – akcií) a po minútí všetkých krokov, určuje počet zozbieranej potravy, daným agentom, jeho fitness. V prípade premenlivého prostredia (platí len pre agenta s chromozómom kódujúcim váhové koeficienty siete, nakoľko pri klasickom chromozóme v ktorom sú len akcie to nemá zmysel) sa každý člen populácie umiestni do Y rôznych (náhodne generovaných ale rovnakých pre

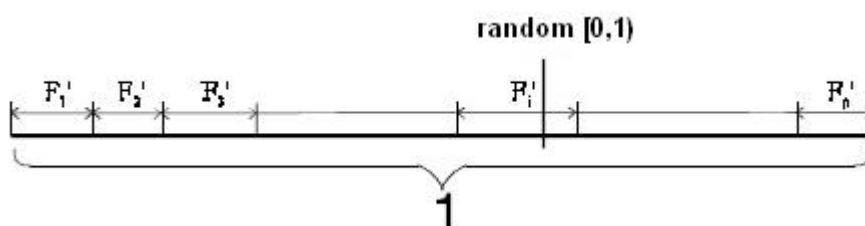
všetkých členov populácie) mriežok, pričom v každej má X krokov. Fitness takéhoto člena populácie je súčet zozbieranej potravy vo všetkých Y mriežkach. Z toho vyplýva, že budú preferovaní tí jedinci, ktorí majú schopnosť zozbierať čo najviac potravy v rôznych prostrediach (pohybujú sa systematicky). V prípade, že by sa každá generácia vyhodnocovala iba na jednej mriežke mohol by byť preferovaný jedinec, ktorý je výborný na jednej mriežke (náhodný pohyb taký, že zozbiera veľký počet potravy) pred jedincom, ktorý na danej mriežke zozbiera menej potravy ale pohybuje sa systematicky. Preto sa každý jedinec aplikuje na Y prostredí (napríklad Y = 5).

Výber jedincov z populácie (ruleta) (c)

Jednotlivý jedinci sa do procesu reprodukcie vyberajú kvázináhodne, to znamená, že sa berie do úvahy ich fitness (jedinci s väčším fitness majú väčšiu pravdepodobnosť výberu). Používa sa pritom algoritmus rulety kedy sa každému jedincovi priradí interval na jednotkovej úsečke úmerný jeho fitness (fitness jednotlivých jedincov sa normalizujú s ohľadom na jednotkovú úsečku):

$$F'_i = \frac{F_i}{\sum_i F_i}$$

a následne sa generuje náhodné číslo z intervalu [0,1), ktoré podľa toho kam padne, určí, ktorý jedinec sa vyberie. Grafické znázornenie tejto operácie možno vidieť na Obr. 3.

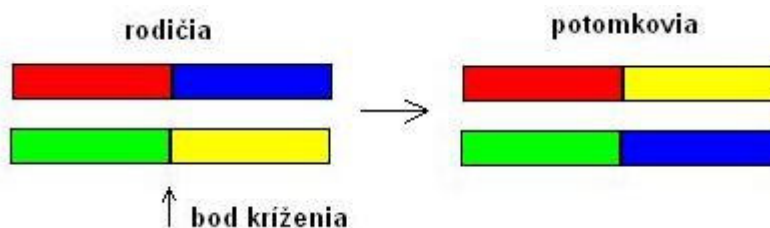


Obr. 3 Grafické znázornenie normalizovania fitness funkcie a ruletový výber

Reprodukcia (d)

Ako vidno z algoritmu na Obr. 2 reprodukcia prebehne len s pravdepodobnosťou P_{repro} . Samotná reprodukcia v sebe zahŕňa dve operácie: **kríženie** a **mutáciu**.

Do procesu kríženia vstupujú dvaja rodičia a vystupujú z neho dvaja potomkovia. Realizuje sa jednobodové kríženie pri ktorom sa určí bod kríženia kde všetky gény za týmto bodom si rodičia vymenia (potomkovia budú mať gény po bod kríženia ako rodičia a po bode kríženia budú mať gény vždy z druhého rodiča). Grafické znázornenie tohto procesu je na Obr. 4.



Obr. 4 Operácia kríženia

Operácia kríženia sa realizuje s pravdepodobnosťou P_{cross} . V opačnom prípade nepríde ku kríženiu jedincov a len sa vytvoria potomkovia identický s rodičmi.

Operácia mutácie sa realizuje po krížení (či už prišlo ku kríženiu alebo nie). Stochasticky transformuje vektor hodnôt α (chromozóm) na nový vektor hodnôt α' , pričom stochastičnosť tohto procesu je určená pravdepodobnosťou P_{mut}

$$\alpha' = O_{mut}(\alpha)$$

kde α a α' sú vektory rovnakej dĺžky

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n) \text{ a } \alpha' = (\alpha'_1, \alpha'_2, \dots, \alpha'_n)$$

kde jednotlivé komponenty α' sú určené takto

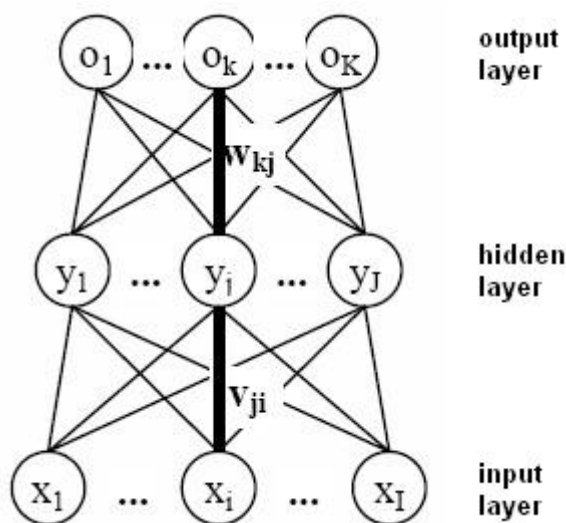
$$\alpha'_i = \alpha_i + \text{normal}(x, \sigma) \quad (\text{pre } \text{random} < P_{mut})$$

$$\alpha'_i = \alpha_i \quad (\text{ostatné prípady})$$

kde *random* je náhodné číslo z intervalu [0,1) generované s rovnomernou distribúciou a *normal*(*x*,*σ*) je číslo generované z normálnou distribúciou so strednou hodnotou *x* a odchýlkou *σ*. Operácia mutácie sa realizuje na oboch potomkoch. Takýto spôsob prebieha pri mutácií kedy chromozóm kóduje neurónovú sieť. Pri štandardnom chromozóme, ktorý obsahuje akcie agenta sa len akcia zamení za niektorú inú náhodne vybranú akciu.

Popis neurónovej siete použitej ako riadiaci prvok agenta

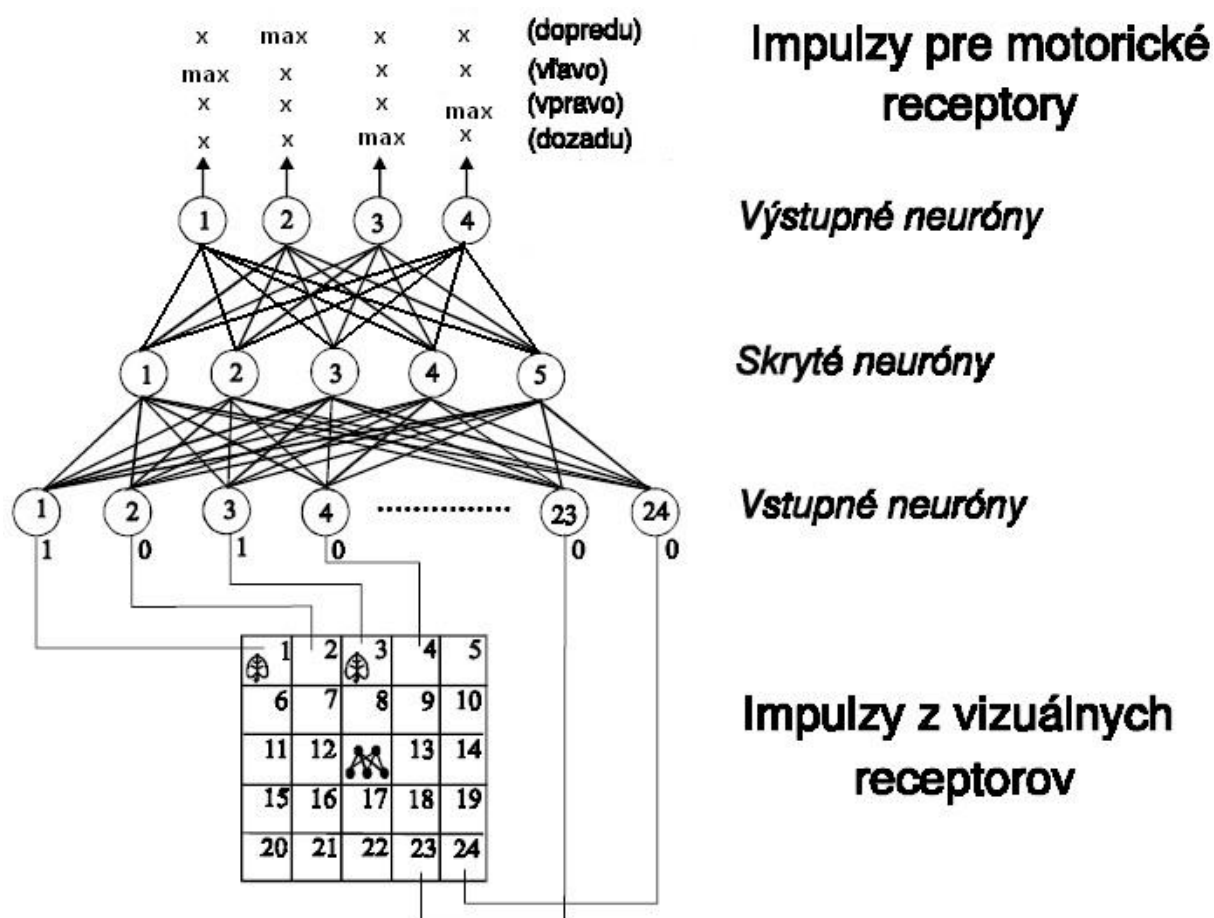
Riadiaci prvok agenta je realizovaný trojvrstvovou doprednou neurónovou sieťou. Príklad takejto siete možno vidieť na Obr. 5.



Obr. 5 Príklad trojvrstvovej doprednej siete

S ohľadom na špecifickú úlohu treba vhodným spôsobom zvoliť počet neurónov vo vstupnej vrstve (input layer), v skrytej vrstve (hidden layer) a vo výstupnej vrstve (output layer). V tejto práci je počet vstupných neurónov závislý na viditeľnej oblasti nakoľko táto sa mapuje tak ako to vidno na Obr. 6. Počet výstupných neurónov je konštantný a rovný počtu akcií, ktoré môže agent realizovať. Jeden neurón je pre pohyb hore, jeden pre pohyb dole, jeden pre pohyb doprava a jeden pre pohyb doľava (takéto kódovanie sa nazýva one-hot kódovanie). Pohyb agenta bude v každom ťahu závisieť na aktivačných hodnotách týchto neurónov (ktoré sú závislé na vstupných neurónoch v danom ťahu) a to takým spôsobom, že sa zrealizuje tá

akcia, ktorej prislúchajúci výstupný neurón má najväčšiu hodnotu. Graficky je táto situácia znázornená na Obr. 6.



Obr. 6 Grafické znázornenie riadiaceho orgánu agenta. Obrázok prevzatý z [1]

Sieť realizuje výpočty aktivačných hodnôt výstupných neurónov na základe hodnôt vstupných neurónov, ktoré sa inicializujú hodnotami okolitých viditeľných políčok mriežky. Výpočty používajú metódu dopredného šírenia, ktorá sa realizuje nasledovne (s ohľadom na sieť na Obr. 5):

$$\bar{x} = (x_1, \dots, x_I) \text{ vektor vstupných hodnôt}$$

$$\bar{o} = (o_1, \dots, o_K) \text{ vypočítaný výstupný vektor}$$

Pre skrytý neurón j sa vypočíta jeho aktivačná hodnota ako:

$$y_j = f\left(\sum_{i=1}^I v_{ji} x_i\right)$$

Pre výstupný neurón sa vypočíta jeho aktivačná hodnota ako:

$$o_k = f\left(\sum_{j=1}^J w_{kj} y_j\right)$$

kde v_{ji} je váhové prepojenie spájajúce skrytý neurón j so vstupom i , w_{kj} je váhové prepojenie spájajúce výstupný neurón k so skrytým neurónom j . Aktivačná funkcia f je sigmoidálna funkcia na intervale (0,1).

Za účelom porovnania s agentom, ktorého sieť sa vyvinula na základe Genetického algoritmu bol zavedený agent, ktorého sieť je trénovaná metódou *backpropagation*. Táto metóda je realizovaná nasledovne:

$\bar{d} = (d_1, \dots, d_K)$ očakávaný výstupný vektor

V tomto algoritme sa realizuje metóda najstrmšieho spádu kde je snaha minimalizovať chybu:

$$E = \frac{1}{2} \sum_{k=1}^K (d_k - o_k)^2$$

kde zmeny výstupných váh sa určia podľa:

$$\Delta w_{kj} = -\alpha \frac{\partial E}{\partial w_{kj}} = \alpha \delta_k y_j$$

a kde chybový signál výstupného neurónu k je definovaný ako:

$$\delta_k = f'(net_k)(d_k - o_k) = o_k(1 - o_k)(d_k - o_k)$$

Zmeny váh medzi skrytými neurónmi a vstupnými neurónmi sa vypočítajú:

$$\Delta v_{ji} = -\alpha \frac{\partial E}{\partial v_{ji}} = \alpha \delta_j x_i$$

pričom chybový signál skrytého neurónu j je definovaný ako:

$$\delta_j = f'(net_j) \sum_{k=1}^K w_{kj} \delta_k = y_j(1 - y_j) \sum_{k=1}^K w_{kj} \delta_k$$

Úprava váhových koeficientov siete sa potom realizuje podľa nasledujúcich rovníc:

$$v_{ji} = v_{ji} + \Delta v_{ji} \text{ a } w_{kj} = w_{kj} + \Delta w_{kj}$$

Symbol α predstavuje rýchlosť učenia algoritmu.

Kombinácia Genetického algoritmu a doprednej neurónovej siete

Kombinácia týchto dvoch prístupov sa snaží vylepšovať riadiaci orgán agenta reprodukciou (krížením a mutáciou) takých jedincov populácie, ktorý sa čo najlepšie vedú prispôbiť prostrediu (zobierajú čo najviac potravy). To znamená, že sa vyberajú takí jedinci, ktorý sa čo najvhodnejšie naučili vnímať svoje viditeľné okolie (Obr. 6). Táto reprodukcia sa realizuje nad chromozómami, ktoré kódujú váhy danej siete. Grafické znázornenie chromozómu, ktorý kóduje sieť z Obr. 5 je na Obr. 7.

w_{II}	...	w_{Kj}	...	w_{KJ}	v_{II}	...	v_{ji}	...	v_{JI}
----------	-----	----------	-----	----------	----------	-----	----------	-----	----------

Obr. 7 Chromozóm kódujúci váhy siete

Celá populácia agentov má jednu spoločnú sieť (štrukturálne), pričom každý agent je reprezentovaný vyššie uvedeným chromozómom. Keď sa ide vyhodnocovať daný agent jeho chromozóm sa načíta do štruktúry siete, čím vznikne plnohodnotná sieť, ktorá realizuje jeho riadenie.

Výsledky experimentov

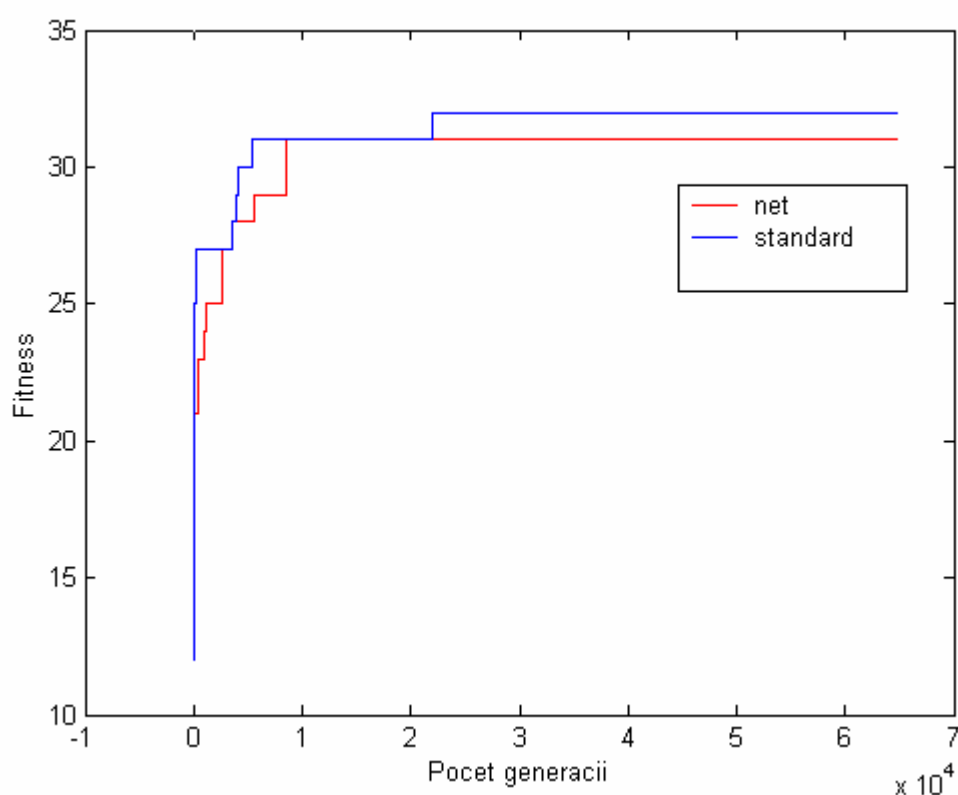
Táto časť prípadovej štúdie prezentuje výsledky realizovaných experimentov. Je rozdelená na tri časti. Prvá časť analyzuje experimenty realizované na jednej potravinovej mriežke. Porovnáva sa úspešnosť získavania potravy pre agenta:

1. riadeného len na základe akcii v svojom chromozóme získanom genetickým algoritmom
2. riadeného neurónovou sieťou získanou genetickým algoritmom
3. riadeného neurónovou sieťou získanou štandardnou metódou backpropagation.

V druhej časti tejto kapitoly sú analyzované experimenty z agentmi riadenými neurónovou sieťou získanou genetickým algoritmom a agentmi riadenými neurónovou sieťou získanou štandardnou metódou backpropagation. Ich úspešnosť sa vyhodnocuje na náhodne generovaných mriežkach.

V tretej časti sa analyzuje vplyv parametrov genetického algoritmu a parametrov siete na vývoj agentov.

Analýza správania na jednej mriežke



Obr. 8 Priebeh fitness (najlepšia fitness v danej generácii) funkcie. Všeobecné parametre: maximálny počet krokov = 50, viditeľná oblasť = 3×3 , veľkosť mriežky = 20×20 , počet potravy = 100, počiatočná pozícia $x = 10$, počiatočná pozícia $y = 10$. Parametre pre agenta riadeného sieťou: počet vstupných neurónov = 8, počet skrytých neurónov = 3, počet výstupných neurónov = 4, veľkosť populácie = 22, pravdepodobnosť reprodukcie = 0.8, pravdepodobnosť kríženia 0.5, pravdepodobnosť mutácie = 0.1, mutačná zmena = 0.05, parameter počiatočného intervalu váh = 1.0. Parametre pre agenta, ktorého akcie sú zakódované v chromozóme: veľkosť populácie = 22, pravdepodobnosť reprodukcie = 0.8, pravdepodobnosť kríženia 0.5, pravdepodobnosť mutácie = 0.02.

Ako vidno na grafe (Obr. 1) je rýchlosť vývoja agenta približne rovnaká v oboch prípadoch (aj pre agentov riadených neurónovou sieťou aj pre agentov riadených na základe akcií zakódovaných v chromozóme agenta). Po ukončení vývoja (dosiahnutie 65000 generácií) bol agent, riadený na základe akcií zakódovaných v chromozóme, schopný zozbierať 32 kúskov potravy. Pre agenta riadeného neurónovou sieťou to bolo 31 kúskov potravy. Záver plynúci z tohto experimentu je ten, že pre stabilné prostredie sú uvedený agenti približne rovnako

schopný. Pre meniace sa prostredie (rôzne mriežky pre každú novú populáciu) prestáva mať agent riadený na základe akcií zakódovaných v chromozóme zmysel. Takýto agent je použiteľný len pre nemeniace sa problém.

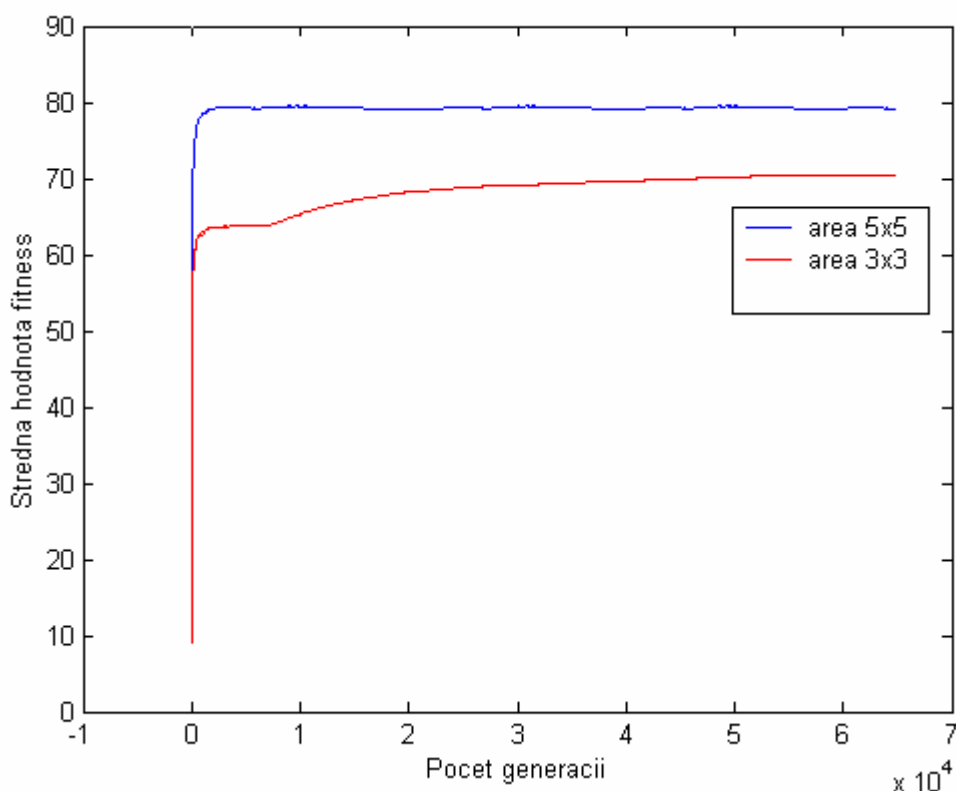
Na testovaciu mriežku bol umiestnený aj agent riadený neurónovou sieťou tréňovanou pomocou metódy backpropagation. Tomuto sa podarilo získať 26 kúskov potravy. Ako vidno je to horší výsledok ako pre vyššie spomenutých agentov ale na rozdiel od nich je tento agent univerzálny, pričom vyššie spomenutí agenti sú tréňovaní na špecifickú úlohu.

Pre porovnanie boli všetci vyššie popísaní agenti umiestnení na novú mriežku s rovnakými parametrami ako predchádzajúca a výsledky sú nasledovne:

1. riadeného len na základe akcií v svojom chromozóme = 15 kúskov potravy
2. riadeného neurónovou sieťou získanou genetickým algoritmom = 20 kúskov potravy
3. riadeného neurónovou sieťou získanou štandardnou metódou backpropagation = 25 kúskov potravy

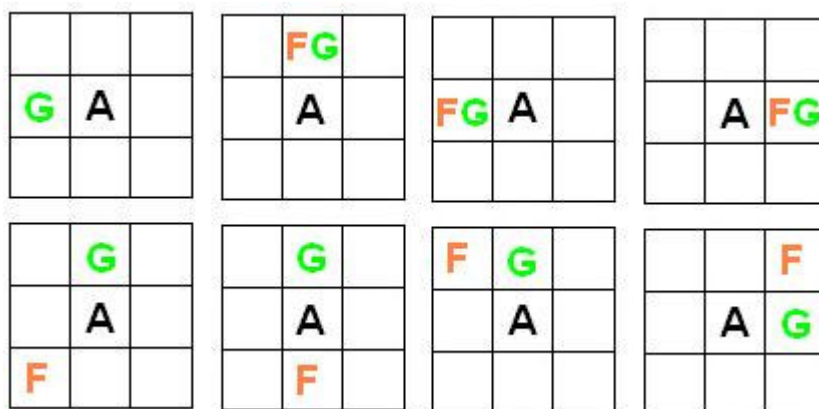
Ako vidno agenti riadení sieťou si zachovali určitú porovnateľnú schopnosť hľadať potravu aj keď v prípade špecializovaného agenta tam bol dosť významný pokles. Schopnosť agenta riadeného len na základe akcií v svojom chromozóme poklesla v novom prostredí až na polovicu.

Analýza správania v premenlivom prostredí (mriežke)

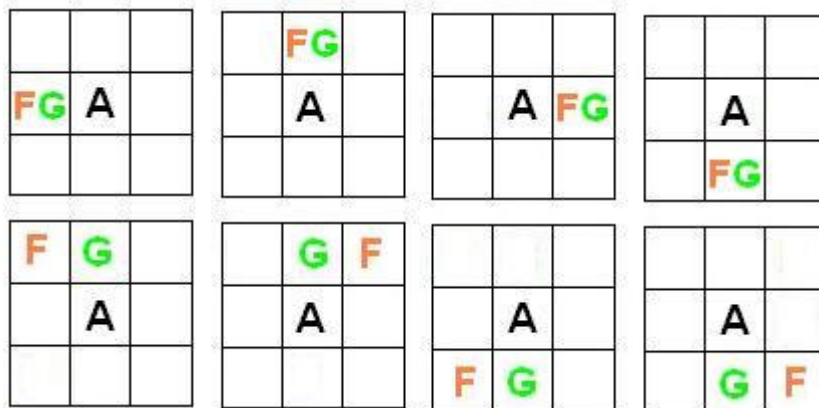


Obr. 9 Stredná hodnota fitness. Všeobecné parametre: maximálny počet krokov = 50, viditeľná oblasť = 3x3 (5x5), veľkosť mriežky = 20x20, počet potravy = 50, počiatočná pozícia x = 10, počiatočná pozícia y = 10. Parametre: počet vstupných neurónov = 8(24), počet skrytých neurónov = 3(6), počet výstupných neurónov = 4, veľkosť populácie = 42, pravdepodobnosť reprodukcie = 0.8, pravdepodobnosť kríženia 0.5, pravdepodobnosť mutácie = 0.1, mutačná zmena = 0.01, parameter počiatočného intervalu váh = 0.1.

Nakoľko pri premenlivom prostredí kolísali hodnoty fitness najlepšieho jedinca v danej generácii, bola miesto klasickej fitness (algoritmus používa túto) použitá na zobrazovanie stredná hodnota fitness, ktorá vhodne zobrazuje trend vývoja fitness. Fitness bola v prípade týchto experimentov daná počtom získaných kúskov potravy z piatich náhodne vygenerovaných mriežok (všetci členovia populácie danej generácie sa postupne umiestnili do piatich náhodne vygenerovaných mriežok, pričom pre každého člena populácie sú tieto mriežky rovnaké). Priemerná hodnota fitness na jednu mriežku sa získa vydelením danej hodnoty piatimi. Ako vidno z Obr. 9 agent s väčšou viditeľnou oblasťou má väčšiu schopnosť získavať potravu z premenlivého prostredia ako agent s menšou viditeľnou oblasťou. To je dané tým, že agent s väčšou viditeľnou oblasťou dokáže získať viac informácií zo svojho prostredia.



Obr. 10 Príklad správania, ktoré sa u agenta vyvinulo pomocou GA. A – agent, F – food, G – goal



Obr. 11 Príklad správania, ktoré bolo trébované metódou backprop. A – agent, F – food, G – goal

Na Obr. 10 je možné vidieť správanie, ktoré sa vyvinulo u agenta. Vo väčšine prípadov sa agent snaží priblížiť k potrave, či už priamo alebo medzi-krokom. Výnimku tvorí potrava v spodnom riadku agentovej viditeľnej oblasti (za agentom). V prípade takto umiestnenej potravy sa snaží agent posunúť o jeden riadok vyššie (dopredu). Takéto chovanie sa vyvinulo ako reakcia na pohyb doprava, v prípade, že sa vo viditeľnej oblasti agenta žiadna potrava nevyskytuje. Takouto kombináciou sa agent snaží traverzovať mapu, čím zmenšil pravdepodobnosť, že si skonzumuje všetku potravu na prstenci toroida (v pásovej oblasti mapy).

Na Obr. 11 je zobrazené správanie, ktoré bolo u agenta získané metódou backprop a kde sa vychádzalo z trenovacích dát explicitne navrhnutých. Boli navrhnuté tak aby sa agent snažil dostať smerom k potrave vo svojom okolí buď priamo alebo medzi-krokom. Pri takomto správaní môže ale prísť k situácii kedy si agent spotrebuje všetku potravu na prstenci toroidu. Nakoľko sa agent v prípade absencie potravy, vo viditeľnej oblasti, pohybuje vždy len jedným smerom a nakoľko je viditeľná oblasť obmedzená, môže prísť k neefektívnemu obiehaniu po prstenci toroida.

Obidva spomenuté typy agentov boli umiestnené do 100 náhodne generovaných potravinových mriežok, pričom pri každej bol uchovaný počet získaných kúskov potravy. Pre každého agenta bola následne vypočítaná stredná hodnota, ktorá vyšla:

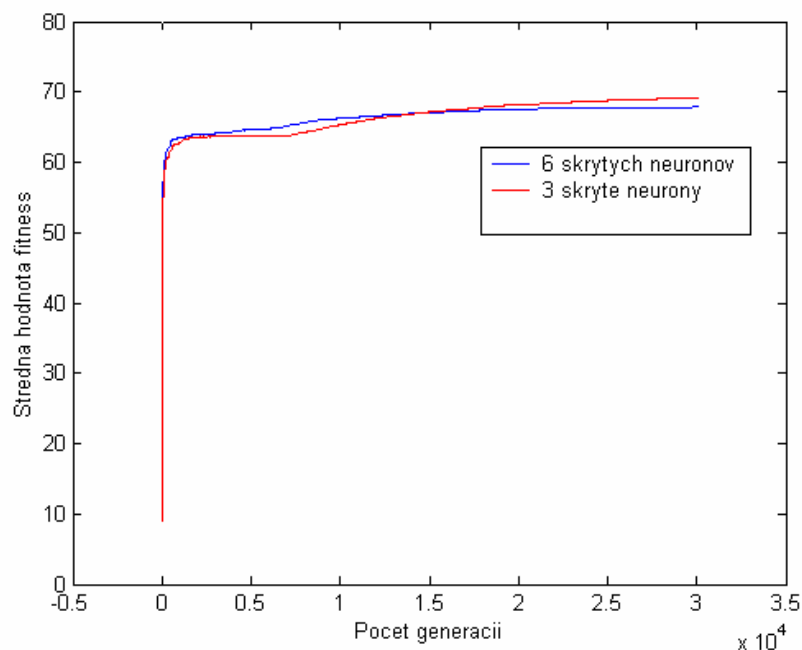
1. agent riadený neurónovou sieťou získanou genetickým algoritmom = 13.5 kúskov potravy
2. agent riadený neurónovou sieťou získanou štandardnou metódou backpropagation = 11.88 kúskov potravy

Záver plynúci z týchto experimentov je ten, že evolučný algoritmus môže v niektorých prípadoch nájsť riešenie, ktoré môže byť vhodnejšie ako na prvý pohľad ideálne riešenie navrhnuté explicitne.

Vplyv parametrov na vývoj agentov

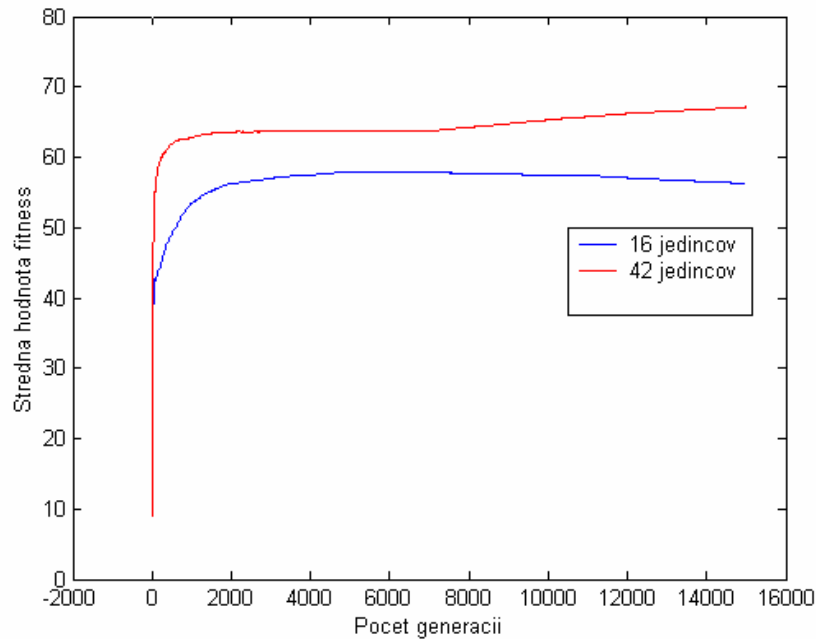
V predchádzajúcej podkapitole boli realizované experimenty s určitými parametrami siete a genetického algoritmu (počet skrytých neurónov, pravdepodobnosť kríženia, atď.), ktoré boli po väčšom počte experimentov nastavené tak, aby čo najviac maximalizovali rýchlosť vývoja a schopnosť nájsť čo najlepšie riešenie. Táto podkapitola analyzuje vplyv týchto parametrov na vývoj (rýchlosť vývoja, stabilita populácie). Analýza tohto typu je nutná najmä preto, lebo s ohľadom na dynamicky sa meniace prostredie (generovanie vždy nových potravinových mriežok) prichádza v mnohých prípadoch (pri určitých hodnotách parametrov a ich kombináciách) k destabilizácii populácie. Destabilizácia populácie je trend novej generácie zhoršovať fitness najlepšieho jedinca v populácii. Vizualne sa to prejaví tým, že v určitom mieste vývoja začne krivka strednej hodnoty fitness postupne klesať (ako bude možné vidieť na niektorých grafoch).

Prvý z grafov (Obr. 12) znázorňuje priebeh vývoja pre rôzne počty skrytých neurónov v sieti riadiacej agentov. Ako vidno, zvyšovaním počtu skrytých neurónov sa nedosiahne výraznejšia zmena vo vývoji (Ak sa zvyšuje počet z hodnoty, pre ktorú sa sieť dokáže naučiť daný problém. V prípade viditeľnej oblasti 3x3 to boli tri neuróny, čo sa podarilo ukázať aj pri učení pomocou metódy backpropagation). Zvyšovaním počtu neurónov je možné dokonca spomaliť vývoj ako vidno z obrázka kde pri hodnote okolo 15000 generácie začala sieť s troma skrytými neurónmi dávať lepšie výsledky. Je to dané počtom parametrov funkcie, ktorú predstavuje neurónová sieť. Pri troch skrytých neurónoch je počet parametrov (váh siete) rovný 43. Pri šiestich skrytých neurónoch to je 82. Takže počet parametrov, ktoré sa musia dimenzovať stúpol takmer na dvojnásobok. Toto môže značne predĺžiť čas potrebný na nájdenie ekvivalentného riešenia, ktoré možno získať so sieťou s menším počtom skrytých neurónov.

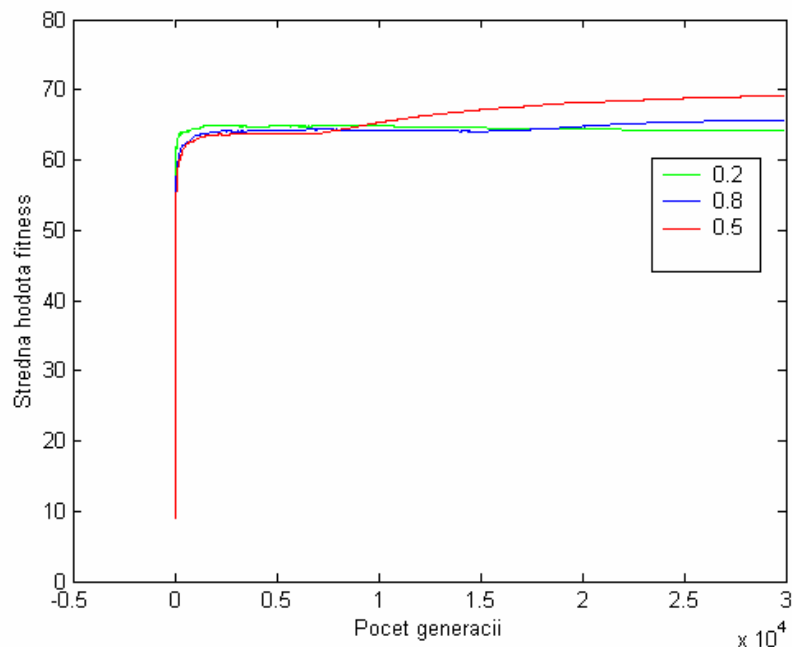


Obr. 12 Stredná hodnota fitness pre agentov s rôznym počtom skrytých neurónov. V oboch prípadoch sú nasledujúce hodnoty rovnaké: maximálny počet krokov = 50, viditeľná oblasť = 3x3, veľkosť mriežky = 20x20, počet potravy = 50, počiatočná pozícia x = 10, počiatočná pozícia y = 10, počet vstupných neurónov = 9, počet výstupných neurónov = 4, veľkosť populácie = 42, pravdepodobnosť reprodukcie = 0.8, pravdepodobnosť kríženia 0.5, pravdepodobnosť mutácie = 0.1, mutačná zmena = 0.01, parameter počiatočného intervalu váh = 0.1.

Ďalším parametrom je veľkosť populácie. Tento parameter patrí ku genetickému algoritmu. Veľkosť populácie má veľmi významnú úlohu pri vývoji. Môže nielen značne spomaliť samotný vývoj ale taktiež môže spôsobiť destabilizáciu populácie, kedy sa začne znižovať najlepší fitness v jednotlivých generáciách. Oba tu spomenuté efekty možno vidieť na Obr. 13. Malý počet jedincov spôsobuje spomalenie vývoja (na grafe vidno, že pre rovnakú generáciu je stredná hodnota fitness algoritmu s menšou populáciou značne menšia ako pre prípad väčšej populácie). Taktiež si možno všimnúť, že okolo 8000 generácie začína dochádzať k destabilizácii populácie čo sa prejavuje klesaním strednej hodnoty fitness. Tento efekt je daný tým, že pri malom počte členov v populácii je väčšia pravdepodobnosť, že do ďalšej generácie nemusí byť vybraný ten najlepší jedinec (dá sa odstrániť elitizmom). Horšia možnosť je taká, že sa v populácii môže objaviť lokálne lepší jedinec (je lepší na daných piatich vygenerovaných mriežkach ale dlhodobo je horší) a práve tento vytlačí dlhodobo úspešného jedinca (nakolko je populácia obmedzená nedostane sa tento do novej populácie). Ďalším dôležitým parametrom pri genetickom algoritme je pravdepodobnosť kríženia. Priebeh vývoja pre GA s rôznymi hodnotami pravdepodobnosti kríženia je na Obr. 14. Ako vidno príliš malé hodnoty aj príliš veľké hodnoty mierne spomaľovali rýchlosť vývoja. Zo začiatku síce vykazovali lepšie správanie ale postupne sa ukázal najrýchlejší (najväčšie hodnoty strednej hodnoty fitness v tej istej generácii) vývoj kde nebola pravdepodobnosť kríženia ani príliš malá ani príliš veľká. Takýto priebeh je daný pravdepodobne tým, že v počiatočnej fáze vývoja je dôležité prehľadať čo najväčší stavový priestor, kedy sa osvedčí väčšia pravdepodobnosť kríženia alebo naopak nechať vývoj prebiehať len na základe mutácií. V neskorších fázach vývoja už je skôr potrebné len mierne doladovanie parametrov, ktoré môže byť dané výmenou vhodných génov (nie príliš časté výmeny) a mutáciami. Preto sa v neskorších fázach ukáže ako najvhodnejší algoritmus s priemernou hodnotou pravdepodobnosti mutácie.

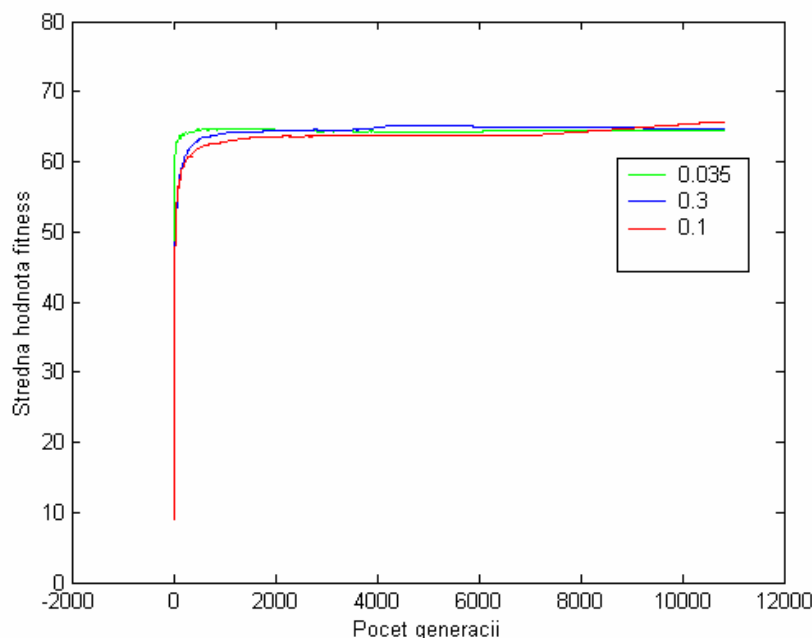


Obr. 13 Stredná hodnota fitness. Rôzny počet jedincov v populácii pri genetickom algoritme. V oboch prípadoch sú nasledujúce hodnoty rovnaké: maximálny počet krokov = 50, viditeľná oblasť = 3x3, veľkosť mriežky = 20x20, počet potravy = 50, počiatočná pozícia x = 10, počiatočná pozícia y = 10, počet vstupných neurónov = 9, počet skrytých neurónov = 3, počet výstupných neurónov = 4, pravdepodobnosť reprodukcie = 0.8, pravdepodobnosť kríženia 0.5, pravdepodobnosť mutácie = 0.1, mutačná zmena 0.01, parameter počiatočného intervalu váh = 0.1.



Obr. 14 Stredná hodnota fitness pre rôzne hodnoty pravdepodobnosti kríženia. V oboch prípadoch sú nasledujúce hodnoty rovnaké: maximálny počet krokov = 50, viditeľná oblasť = 3x3, veľkosť mriežky = 20x20, počet potravy = 50, počiatočná pozícia x = 10, počiatočná pozícia y = 10, počet vstupných neurónov = 9, počet skrytých neurónov = 3, počet výstupných neurónov = 4, veľkosť populácie = 10, pravdepodobnosť reprodukcie = 0.8, pravdepodobnosť mutácie = 0.1, mutačná zmena = 0.01, parameter počiatočného intervalu váh = 0.1.

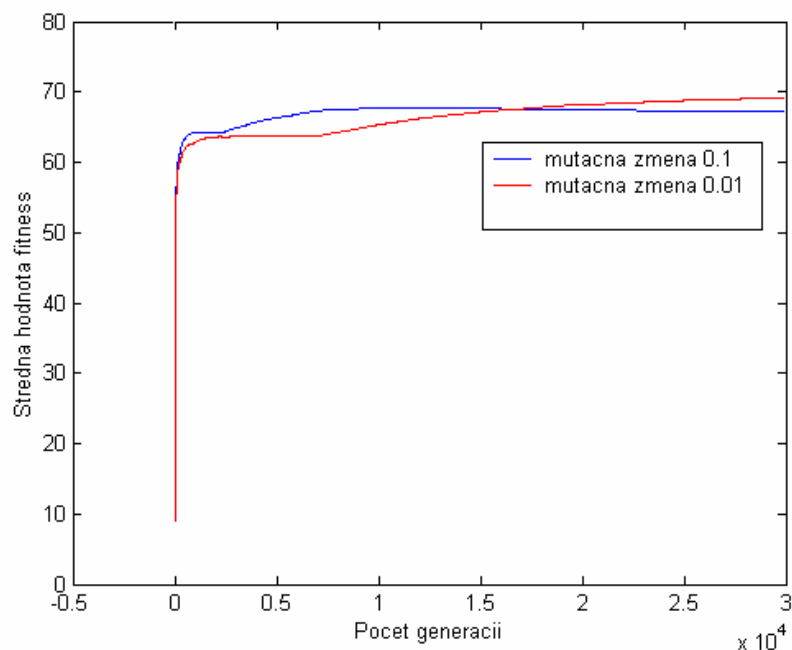
Ďalším parametrom genetického algoritmu je pravdepodobnosť mutácie. Táto má význam najmä v neskorších fázach doladovania parametrov funkcie (váh siete). Ako vidno z Obr. 15 zo začiatku je vhodnejšia väčšia pravdepodobnosť mutácie alebo malá pravdepodobnosť mutácie ale neskôr sa presadí algoritmus s pravdepodobnosťou mutácie, ktorá je kompromisom týchto hodnôt.



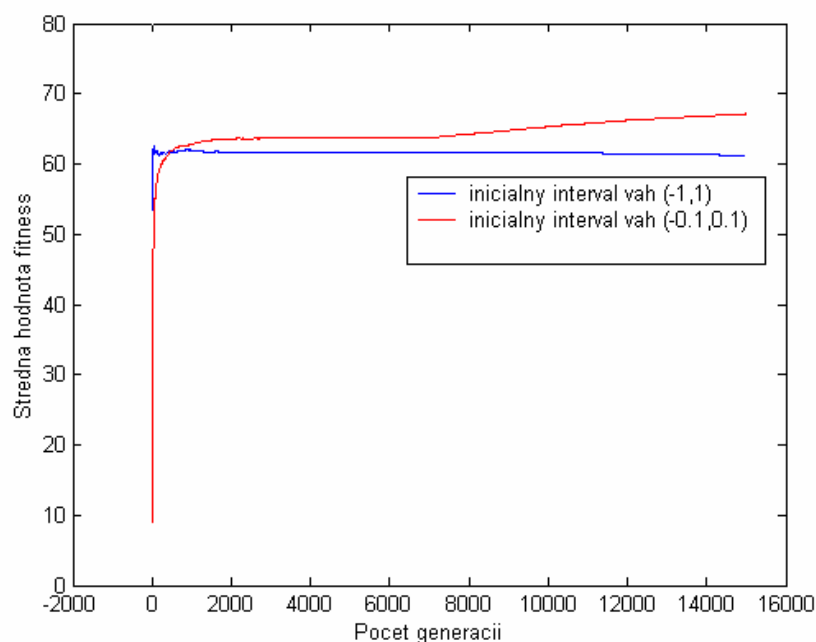
Obr. 15 Stredná hodnota fitness pre rôzne hodnoty pravdepodobnosti mutácie. V oboch prípadoch sú nasledujúce hodnoty rovnaké: maximálny počet krokov = 50, viditeľná oblasť = 3x3, veľkosť mriežky = 20x20, počet potraviny = 50, počiatočná pozícia x = 10, počiatočná pozícia y = 10, počet vstupných neurónov = 9, počet skrytých neurónov = 3, počet výstupných neurónov = 4, veľkosť populácie = 10, pravdepodobnosť reprodukcie = 0.8, pravdepodobnosť kríženia = 0.5, mutačná zmena = 0.01, parameter počiatočného intervalu váh = 0.1.

K mutácií sa vzťahuje aj ďalší parameter, ktorým je veľkosť mutácie. Priebeh evolúcie pre dve rôzne veľkosti mutačnej zmeny sú na Obr. 16. Ako vidno zo začiatku je lepší vývoj kde je mutačná zmena väčšia, nakoľko sa takto prehľadá väčšia časť stavového priestoru. V neskorších fázach, ktoré vyžadujú len mierne doladovanie váh je príliš veľká zmena nevhodná nakoľko sa môžu preskočiť potenciálne vhodné riešenia. V tejto fáze sa stávajú výhodnejšími menšie mutačné zmeny.

Ďalším parametrom, ktorý sa ale skôr vzťahuje k vlastnosti siete je veľkosť intervalu, z ktorého sa generujú počiatočné váhy pre jedincov v inicialnej populácii. Z experimentov vychádza, že príliš veľký počiatočný interval nieje príliš vhodný. Veľký interval môže spôsobiť nielen spomalenie evolúcie ale dokonca stagnáciu až destabilizáciu populácie. Je to dané tým, že pri veľkom intervale môže byť rozloženie váh príliš vzdialené od želaného rozloženia a systematickým prehľadávaním (malá pravdepodobnosť mutácie, malá veľkosť mutácie) môže trvať veľmi dlho dostať sa do optimálneho rozloženia. Priebeh vývoja pre rôzne počiatočné intervaly je na Obr. 17. Ako vidno pre interval (-1,1) prišlo najskôr k stagnácii populácie a neskôr dokonca k destabilizácii, kedy začala stredná hodnota fitness klesať.



Obr. 16 Stredná hodnota fitness pre rôzne hodnoty veľkosti mutácie. V oboch prípadoch sú nasledujúce hodnoty rovnaké: maximálny počet krokov = 50, viditeľná oblasť = 3x3, veľkosť mriežky = 20x20, počet potravy = 50, počiatočná pozícia x = 10, počiatočná pozícia y = 10, počet vstupných neurónov = 9, počet skrytých neurónov = 3, počet výstupných neurónov = 4, veľkosť populácie = 10, pravdepodobnosť reprodukcie = 0.8, pravdepodobnosť kríženia = 0.5, pravdepodobnosť mutácie = 0.01, parameter počiatočného intervalu váh = 0.1.



Obr. 17 Stredná hodnota fitness pre rôzne intervaly, z ktorých sa inicializujú počiatočné váhy siete. V oboch prípadoch sú nasledujúce hodnoty rovnaké: maximálny počet krokov = 50, viditeľná oblasť = 3x3, veľkosť mriežky = 20x20, počet potravy = 50, počiatočná pozícia x = 10, počiatočná pozícia y = 10, počet vstupných neurónov = 9, počet skrytých neurónov = 3, počet výstupných neurónov = 4, veľkosť populácie = 10, pravdepodobnosť reprodukcie = 0.8, pravdepodobnosť kríženia = 0.5, pravdepodobnosť mutácie = 0.01, mutačná zmena = 0.01

Zhodnotenie prípadovej štúdie

Získavania zdrojov je kľúčovou funkciou v mnohých oblastiach. Táto prípadová štúdia ukazuje schopnosť agentov riadených neurónovou sieťou naučiť sa a úspešne plniť takúto úlohu nielen pre stabilné prostredie ale aj dynamicky sa meniace prostredie. Použitie neurónovej siete ako kognitívneho orgánu počíta so získavaním informácií zo svojho okolia a reagovaním na tieto informácie. Takéto riešenie získava náskok pred riešením využívajúcim čisto kombináciu možných akcií v prostredí, ktoré je často nepružné pri dynamicky sa meniacom prostredí. Takisto sa ukázalo, že čím viac informácií je agent schopný získať z prostredia, tým úspešnejšie je schopný riešiť úlohy v danom prostredí. Kľúčovou úlohou pri vývoji agenta riadeného neurónovou sieťou je nájdenie vhodných parametrov algoritmu. Tieto parametre ovplyvňujú nielen dĺžku vývoja ale aj schopnosť nájsť kvalitné riešenie zadaného problému.

Použitá literatúra

- [1] V. Kvasnička, J. Pospíchal, P. Tiňo: Evolučné algoritmy, STU, Bratislava, 2000.