

LATVIJAS UNIVERSITĀTE
DATORZINĀTŅU FAKULTĀTE

PAMATMODELIS MODUĻU-BĀZĒTAI MĀKSLĪGAI DZĪVĪBAI

KURSA DARBS

Autors: **Jānis Knets**
Studenta apliecības Nr.: jk07108
Darba vadītājs: M. Dat. Andrejs Hanins

RĪGA 2012

Anotācija

Raksts ir tēvēts lai aprakstītu attīstoša, adoptējoša un reproducējoša koda pamatmodeli, kā arī ģenētisko algoritmu izmantošanas nepieciešamību (Moduļu-bāzētas Mākslīgās Dzīvības ideja).

Kursa darbs apskata procesus, kuri notiek kodam reaģējot uz apkārtējās pasaules (kura ir kontroles kods) dotiem parametriem un kā kods pielāgojās pievienojot jaunus moduļus no pamatmoduļiem, kuros ietilps zemā līmeņa loģikas elementi. Ir divi principi pēc kuriem tiek definēts koda attīstība – efektivitāte un izdzīvošana (jeb veselība). Tas tiek darīts, lai noteiktu vai evolūcija\mutācija bija progresīva vai regresīva. Attīstību virzīs koda mēģinājumi atrisināt vienkāršas problēmas, kā matemātiskie vienādojumi vai apakšvirkņu meklēšana.

Abstract

This paper is targeted to describe the basic model of evolving, adopting and reproducing code, and its requirements using the idea of genetic algorithms (Modular-based Artificial Life concept).

Corse work will overview the process of code reacting to outer world (which is the controll code) given parameters and how it adopts by adding new modules from a preset of basic logic elements. There are two principles how to judge the codes progress - efficiency and survivability, to determine was the evolution/mutation successful or not. Evloving code atempts to solve basic problems as mathematical equations or search of substrings.

Atslēgvārdi

Mākslīgā dzīvība, pašreprodukcijas kods, mutācijas kods, moduļu bāzēta sistēma

Saturs

Anotācija.....	2
Abstract.....	3
Apzīmējumi.....	6
Ievads.....	7
Idejas saknes.....	7
Galvenā doma īsumā.....	7
Mērķi.....	7
1. Pasaulē atrodamie piemēri.....	8
1.1. Zinātniskie darbi.....	8
1.2. Pielietojums datorspēlēs.....	8
2. Moduļu-bāzēta sistēma.....	9
2.1. Pamatstruktūra.....	9
2.2. Organisms.....	9
2.2.1. Pirmkoda modulis.....	9
2.2.2. Reprodukcijas modulis.....	11
2.2.3. Mutācijas modulis.....	11
2.2.4. Dabīgā atlase.....	12
2.3. Pasaule.....	13
2.3.1 Struktūra.....	13
2.3.2 Fiziskā struktūra.....	15
2.3.3 Statistikas modulis.....	16
2.3.4 Novērtējuma Modulis.....	16
2.3.5 Atlases Modulis.....	17
Nobeigums.....	19
Izmantotie materiāli.....	20
Pielikums.....	21
Dokumentārā lapa.....	22

Apzīmējumi

Organisms – programmas sastāvdaļas, kuras attīstīsās iekš Pasaules. Šis kods attīstīsās un veido jaunas paaudzes. Jebkuram organismam ir noteikts mūžs.

Pasaule – vadības kods, šī moduļu saime seko līdz visiem organismiem, ievāc datus par tiem, veic globālās darbības, kuras ietekmē visus organismus.

Ievads

Idejas saknes.

Jau kādu laiku autoram galvā bija ideja par to ka var izmantot dabā redzamos processus, lai izveidotu kaut kādu “organismu”, kurš reaģētu uz ārējo pasauli un modificētu sevi, lai pielāgoties pasaulei. Tas, ko mēs uzskatam par dzīvjiem organismiem attīstās pēc tiem pašiem vispārīgiem likumiem. Šos likumus varētu pārveidot par algoritmiem un realizēt datorprogrammā.

Galvenā doma īsumā.

Ar pietiekoši sarežģītiem algortimiem var aprakstīt jebkuru procesu, kurš notiek visumā. Viena no leilākām problēmām ir sākotnēji izprast līdz pēdējam sīkumam to procesu, par kuru tiks veidots algoritms. Dotajā darbā tiks aprakstītas vienības, kurām vajdzētu būt mākslīgā dzīvībā. Šis darbs apraksta pamata elementus, kuru attiecības savā starpā veidos vienu organismu. Pamatā katram organismam eksistē trīs elementi – reprodukcijas kopums, mutācijas kopums un dotās vienības pirmkods, kurš tiek pārveidots ar katru jaunāku mutāciju.

Mērķi

Šī darba mērķis ir izstrādāt modeli uz kura bāzes turpmāk varētu rakstīt programmu, kura pati izveidotu sev uzlabojumus. Balstoties uz iegūto statistiku no pirmās paaudzes izveidotu jaunu paaudzi no pieejamiem pamatmoduļiem un ģenerētiem moduļiem.

1. Pasaulē atrodamie piemēri.

Apkopojums par autora atrastiem materiāliem globālajā tīmeklī, kā arī autora domas .

1.1. Zinātniskie darbi.

Izmantojot tikai internetu un ikdienas rīkus ir grūti atrast kaut ko taustāmu un viegli uztveramu iesācējam. Viss ko var atrast ir pietiekoši novecojis un tam vajadzētu būt sen atjaunotam. Latviešu vārdnīcā pat nav iztulkots un paskaidrots tāds termins kā “Artificial life”, kas tiešā tulkojumā nozīmē “mākslīgā dzīvība”. Līdz ar to šeit ir autora personīgs tulkojums no angļu valodas definīcijai, kura ir atrodama Wikipedia.org mājaslapā [2].

Mākslīgā Dzīvība: ir zinātnes lauks, kurš pēta sistēmas, kuras ir saistītas ar dzīvību, tās procesiem, un tās evolūciju izmantojot datoru modeļus, robotus un bioķīmiju.

Tomēr tika atrasta viena mājaslapa ar pietiekoši detalizētu aprakstu. Aprakstītais darbs bija tikai radniecīgs, bet ne pilnībā saistīts ar šeit arpakstīto moduļu bāzētu pieeju. Raksts tīmeklī saucas “Evolucionāri algoritmi”[1].

1.2. Pielietojums datospēlēs.

Praktiski jebkura datospēle izmanto Mākslīgo Intelektu. Kaut arī tas nav tieši tas temats par kuru ir šis darbs, tomēr tie ir radniecīgi temati. Mākslīgais Intelekt datospēlēs attīstās jau vairākus gadu desmitus. Ar katru gadu pretinieki kļūst gudrāki, spēj rast lielāku pretestību, diemžēl ar laiku cilvēks atrod nepilnības spēļu izstrādātāju Mākslīgā Intelektā un tās izmanto. Tā ir viena no lielākām problēmām – attīstības trūkums.

Mūsdienu pasaulē lielākā daļa spēļu izmanto internetu. Iedomājoties tādu iespēju ka visu spēlētāju ikdienas pavadītājs laiks īstenībā attīsta viņu virtuālos pretiniekus, padarot tos gudrākus un gudrākus. Vienīgā problēma šādā scenārijā ir iemācīt mākslīgo intelektu iegūt jaunas zināšanas no visiem pretiniekiem, likt tam saprast kas ir labi, kas ir slikti.

Var pieminēt šahu, kurā mākslīgais intelekts jebkurā gadījumā uzvarēs, jo tas pazīst visus iespējamus variantus kā var izspēlēt jebkuru kombināciju. Šī zināšanu glabātuve neizsakāmi palīdz. Cilvēka dzīves pieredze, kā arī ģenētiski iedzimtā informācija ir nesamērojama datu krātuve. Katrs lēmums kurš tiek izdarīts ir balstīts uz apzinātām zināšanām un tas ko mēs saucam par intuīciju.

2. Moduļu-bāzēta sistēma.

Dotajā nodaļā tiks aprakstīts autora piedāvāts risinājums un pamatmodelis, pēc kura varētu izveidot mākslīgo dzīvību imitējošu programmu.

2.1. Pamatstruktūra

Pamatā ir doma veidot visu programmatūras kodu ar moduļiem. Šādi moduļi sarunāsies savā starpā ar noteiktiem protokoliem. Tie būs sadalīti vairākos līmeņos – pamatmoduļi, programmu ģenerētie moduļi, lietotāja veidotie moduļi, un tiks uzturēti izvēles moduļu sarakstā.

Pasaules daļa veido ievaddatus priekš mutējošām programmām. Faktiski tās daļas kuras būs “dzīvās” ietilpst pasaules pirmkoda. Pasaule varētu būt arī lietojumsaskarne ar cilvēku, kurš varētu mainīt dažādus parametrus, kurs savukārt ietekmētu mutējošās daļas.

Mutējošais un attīstošais pirmkods ietilpst Pasaulē. Tādu bloku, jeb organismu ir diezgan daudz. Katrs atsevišķs cīnās par saviem resursiem no pasaules (processora laiks, atmiņa u.c.).

Pasaulei ir globāls cikls ar noteiktu (vai bezgalīgu ar pārtraukuma iespēju) iterāciju skaitu. Katras iterācijas sākumā tiks padoti ievaddati katram organismam. Tālāk Pasaule sagaida katra organisma apstrādes beigu paziņojumu. Nākošais solis ir statistikas savākšana, uz kuras pamata tiks aprēķināts katra atsevišķa organisma efektivitāte.

2.2. Organisms

Organismi ir tās vienības kuras pašas reproducējas, pārveidojas un mēģina risināt tām iedoto problēmu. Organismi vienmēr pieder pie kāda dzimtas koka.

2.2.1. Pirmkoda modulis

Sākotnēji eksistē triviāli moduļi, kurus var izmantot jebkurš organisms kā pirmkoda būvblokus. Tos var iedomāties kā dažādu veidu klucīšus, kurus var salikt kopā dažādos veidos. Katram tādām būvblokam eksistēs noteikti likumi kādā veidā tos var izmantot. Likumus var vērot kā sintaksi organisma pirmkoda programmēšanas valodai vai drīzāk otrādi – likumi tiks veidoti balstoties uz sintakses atļaujām konkrētajā programmēšanas valodā, kurā tiks rakstīta šī programma kopumā.

Moduļu galvenā ideja ir tāda, ka jebkura lietderīga kombinācija var kļūt par moduli papildus

pamata moduļu sarakstam. Moduļa lietderīgums tiek noteikts ar diviem kritērijiem – tā efektivitāte un parādīšanās neatkarīgos dzimtas zaros. Priekš tam, lai nesekotu katram ģenerētam koda gabalam noteikti sekoša kādam ģenerētam blokam sāksies balstoties uz tā efektivitāti un ar katru Pasaules iterāciju tiks samazināts skaitītājs, līdz ko tas sasniedz 0 šis modulis tiek izņemts no izvēles moduļu saraksta. Ja šis bloks ir sastaps citos zaros, viņa skaitītājs tiek palielināts. Tajos gadījumos, kad šis bloks tiek ielikts no pamatbloku satura viņš tiek speciāli marķēts, lai dotajam skaitītājam nebūtu eksplanacionāla palielināšanās iespēja.

Sākotnēji definētie pamatmoduļi:

1. Loģiskie pamatmoduļi
 - 1.1. UN Logical AND
 - 1.2. VAI Logical OR
 - 1.3. NĒ Logical NOT
2. Vadības pamatmoduļi
 - 2.2. JA/VAI IF/ ELSE
 - 2.3. LECIENS GOTO
3. Aritmētiskie pamatmoduļi
 - 3.1. SUMMA SUM
 - 3.2. ATNEMT SUB

Lietotājs var izveidot savus un pievienot tos pie kopējā saraksta. Tādā veidā panākot kādu specifisku attīstības paveidu, vai paātrināt mutāciju un uzlabotu rezultātu sasniegšanu.

Lai mutāciju modulis varētu atpazīt izmantotos moduļus un kādā secībā tie tiek izmantoti tiks veidots speciāls saraksts kuru varētu saukt par organisma DNS. Šie saraksti tiks sapāroti ar citiem zariem, lai viedotu atvasi. Kā arī jaunizveidotais DNS tiks mutēts (šie procesi sīkāk tiks aprakstīti pie mutācijas moduļa apraksta).

Moduļu arhitektūras mērķis ir izveidot pietiekoši elastīgu un veiklu vidi, kura varētu piesaistīt sev jebkādu jaunizveidotu moduli savām vajdzībām. Konkrēts organisms varētu izmantot citu organismu sasniegumus savai pilnevidošanai. Būtu interesanti novērot kā dažādi organismu zari apgūst kādu jaunu moduli un kā mēģina pielietot.

Ilglaicīgās attīstības processā varētu uzskatīt par moduļiem arī datora arējās iekārtas, ja pati programma spētu izveidot sakaru ar tādo objektu un sākt viņu izmantot. Tāds variants ir diezgan fantastisks mērķis šī dokumenta rakstīšanas brīdī, bet tādai iespējai vajadzētu būt nodrošinātai, lai organisms nebūtu ierobežots tikai ar vienu datoru.

Iespējams būs grūti izveidot zaru dažādību, vai tie tik raiti apmainīsies ar saviem moduļiem, ka beigās izskatīsies diezgan vienādi. Tās varētu būt galvenās problēmas šāda veida pieejai.

2.2.2. Reprodukcijas modulis

Kā jau minēts, katram organismam ir reprodukcijas modulis. Šis modulis atbildēs par jauna organisma izveidi balstoties uz paša uzkrāto pieredzi, kā arī sekos savam “mūža” garumam. Mūža garums var būt dažāds, un tas tiek izteikts Pasaules iterācijās.

Reprodukcijas galvenais uzdevums ir veidot jaunu organismu. Tas tiek darīts ar uzkrātās informācijas par efektivitāti palīdzību, kā arī balstoties uz sarakstu ar paša organisma pirmkoda moduļu sastāvu.

Ja tika atrasts otrs organisms ar kuru sapārot vienādās koda daļas paliek tādas pašas (līdzība ar vecākiem) Konfliktējošās daļas tiek izvērtētas sākumā pēc sintakses principiem – pirmkoda moduļi tiek pielāgoti sintakses likumiem, pārvietoti tuvāk vai tālāk, vai izmesti pilnībā. Šī darbība ir īstenība pirmais mutācijas solis, kurš ir sastāvdaļa no mutācijas procedūras. Mutācija tiek aprakstīta nākošajā apakšnodaļā.

Kad mutācija ir pabeigta ir gatavs jauns moduļu saraksts, kurš veidos pirmkodu priekš jauna organisma. Ar šī saraksta palīdzību tiek izveidots jauns organisms, kuram tiek pievienots reprodukcijas modulis un mutācijas modulis. Tiek izveidots jauns teksta fails ar visām nepieciešamām sastāvdaļām pareizā sintaktiskā pierakstā. Jaunais fails tiek nokompilēts un nosaukts ar specifisku kodu.

Organisma nosaukuma koda saturs:

- * Izveides cikls ar nullēm priekšā, ja tas ir nepieciešams;
- * Sava atsaucē nummurs;
- * Sava zara vecāka atsaucē nummurs;
- * Cita zara vecāka atsaucē nummurs.

Jaunizveidotas atvases zara noteikšana notiek ar salīdzinājumu ar abiem vecāku zariem. Attiecīgi kuram no vecākiem jaunizveidotais organisms ir līdzīgāks – tajā zarā organisms tiek arī ievietots.

2.2.3. Mutācijas modulis

Mutācijas moduļa mērķis ir sakrustot divus savienojamus pirmkodu sarakstus jaunā pirmkoda un ievietot iespējamu mutāciju. Iespēja ka papildus sakrustojumam notiks arī mutācija ir modificējams lielums, kuru var mainīt Pasaules darbības laikā, vai uzstādīt dinamisko parametru ar triviālu

funkciju, piemēram $y = 2x - 5$, kur x ir ciklu skaits. Tiks dotas arī sabalansētas dinamisko mutāciju iespējas funkcijas, kuras nedos bezgalīgi lielus rezultātus Pasaules ciklu skaitu robežās.

Pirmkoda paplašināšana notiks mutācijas laikā, kad programma mainīsies, tad viena bloka vietā būs iespējams ievietot vairākus jaunus blokus no pilna saraksta, līdz ar to paplašināsies pirmkods, un konkrēta zara jaunā paudze attīstīsies, organismi kļūs sarežģītāki un lielāki.

Aizvietojamu rindu varētu noteikt ar dažādiem paņēmieniem, gan balstoties uz statistiskiem datiem par pirmkoda efektivitāti (profilēšanu), gan ar vienkāršu varbūtisku skaitli. Ļoti iespējams, ka dažādi izvēles metodes paveidi tiks pārbaudīti vienlaikus. Visticamāk ar laiku neefektīvie zari vienkārši izmirs, jo nespēs pietiekoši ātri attīstīties.

Nākošais solis ir izvēlēties moduli ar kuru aizvietot eksistējošo. Pamatbloki var tikt aizvietoti ar loģiski pretējiem pamatblokiem, piemēram, UN tiks nomainīts ar VAI, vai pretēji. Vadības bloki varētu tikt paplašināti ar jauniem atzariem. Vadības rīki nevarēs būt aizvietoti ar citiem pamatmoduļi sintaktisku iemeslu dēļ, loģiskais bloku kopums var kļūt par vienu veselu bloku kopu, kurš rezultātā varētu būt aizvietots. Ģenerētie moduļi var būt aizvietoti ar citiem, praktiski jebkuriem, moduļiem. Izvēle ar ko tieši aizveidot kādu moduli tiks balstīta uz moduļa iekšējo uzbūvi, tātad konkrēts modulis varētu būt aizvietots ar citu – līdzīgu, vai tā iekšējās sastāvdaļas var būt aizvietotas ar pamatmoduļu aizvietošanas principiem. Jebkurš veidots modulis var saturēt citus veidotus moduļus.

Katrā Pasaules iterācija tiks atlasīti organismi nākošās paaudzes veidošanai. Lai organisms varētu būt atlasīts, tam vajadzētu izpildīt noteiktas prasības – salīdzinot ar cilvēkiem, tam ir jābūt pietiekoši veselīgam un sasniegušam reprodukcijas vecumu. Tomēr mutācija ar mazu iespēju var notikt jebkurā no organismiem ārpus atlasē un reprodukcijas processiem. Tas notiek pirms Pasaules globālā cikla primā soļa.

2.2.4. Dabīgā atlase

Dabīga atlase ir ikdienas lieta jebkurā evolūcijas kontekstā. Pasaule izvēlas tos kuri ir labāk pielāgojušies pēdējām izmaiņām, tos kuri ir efektīvāki savā resursu patēriņā un spēj sasniegt rezultātu ātrāk.

Dabīgā atlase notiks caur reprodukcijas izvēles procedūrām. Tas fakts, ka tiek paņemti līdzīgi organismi no dažādiem zariem veicina attīstību, līdz ar to jaunizveidotam organismam balstoties uz vecāku savietojamību tiks pagarināts mūžs attiecībā pret abu vecāku vidējo.

Runājot par izmiršanu – jebkurš zars cenšas veidot jaunu paudzi, jebkurš organisms zarā to

cenšas sasniegt. Tomēr jo ilgāk konkrēts zars un tā organismi būtiski nemainīsies, jo mazāka būs iespēja to sapārot ar kāda cita zara organismu, līdz ar to nākošās paudzes mūža garums arī tiks samazināts arvien vairāk līdz tam brīdim, kad tas vienkārši būs palicis par 0 pēdējai paudzei. Tādā gadījumā zars būs izmiris un vairs netiks uzskatīts par aktīvu. Tomēr tā attīstības vēsture joprojām saglabāsies, kā arī efektīvie bloki, kuri tika paņemti no šī zara (ja tie tiek izmantoti citos zaros un to organismos).

Šāda pieeja garantēs labāk sagatavotu organismu izdzīvošanu, bet tomēr ļaus pagaidu ne tik efektīviem organismiem attīstīties un ar laiku, iespējams, tas kas bija ne tik efektīvs tiks pārveidots par ļoti lietderīgu bloku, kurā brīdī pārējie organismi sāks to izmantot. Rezultātā ne tik veiksmīgajam zaram pagātnē izdevās izrauties uz augšu caur, uz pirmo acs skatienu, ne tik veiksmīgu pirmkodu.

Atlase notiks Pasaules cikla ietvaros – katrs organisms, kurš izpilda minimālās prasības mēģinās tiks ievietots potenciālo vecāku sarakstā, kurā tiks meklēti savietojamie. Sākumā visi atrastie potenciālie vecāki tiek sakārtoti dislošā secībā pēc vērtējuma rezultātiem. Tālāk katram organismam ar varbūtisku skaitli tiks noteikts vai tas piedalīsies jaunas paaudzes veidošanā. Sākot ar vislabāk novērtēto organismu un dilstošā secībā, tiek meklēts otro organisms starp potenciāliem vecākiem. Labākais pāris tiks izņemts no saraksta un apstrādāts ar reprodukcijas un mutācijas procedūrām. Pāru meklēšana notiek tik ilgi kamēr tika izmēģināts atrast pāri katram organismam ar visiem pārējiem.

Katrs organisms var tikt izmantots priekš jaunas paaudzes radīšanas līdz savam maksimumam reizes, kuras tiek uzstādīts balstoties uz primkodu novērtējumu. Kā arī var būt specifiski organismi kuri varētu reproducēties bez otrās daļas ar vienkāršu mutāciju, un tādi kuri spēj reproducēties vairāk reizes viena Pasaules cikla laikā. Kaut gan abi pēdējie varianti būs daudz retāki nekā parastā reprodukcija balstoties uz diviem elementiem no dažādiem zariem. Vienkārši vairāki iespējamie reprodukcijas varianti var veidot dažādus rezultātus, kurus salīdzināt būtu diezgan interesanti.

2.3. Pasaule

Pasaule ir lielā programma, kura izveido saskarni ar lietotāju, ļauj tam veikt izmaiņas un parāda statistiku par zariem un to novērtējumus.

2.3.1 Struktūra

Pasaulei ir noteiktas darbības, kuras tā veic katru ciklu. Pasaules lielais cikls varētu būt salīdzinārs

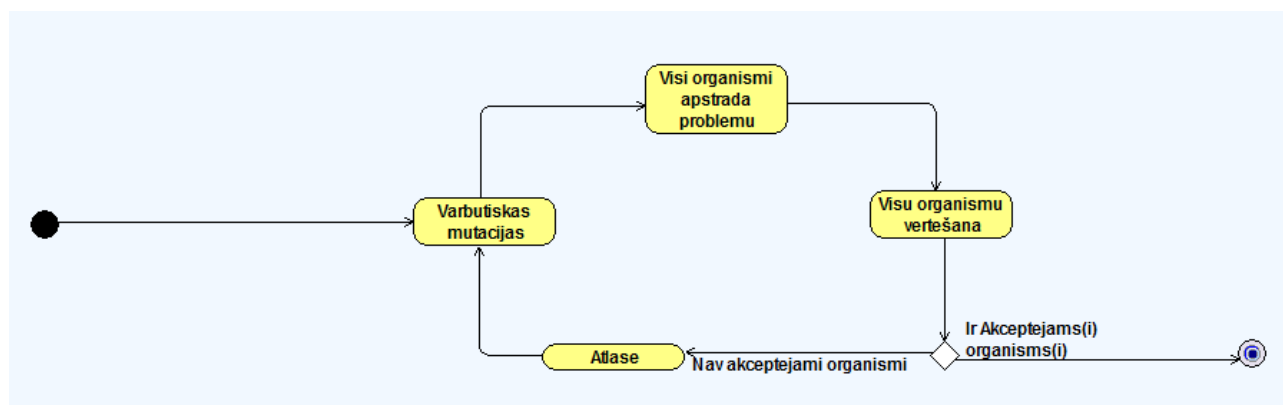
Zemes vienu kalendāro gadu (vai kādu smalkāku laika vienību, ja jums tas labpatīk) Cikla laikā pirmā darbība ir veikt varbūtiskās mutācijas visos organismos. Pasaule izvēlas 1% (vismaz vienu) no visiem organismiem un izsauc tam mutācijas procedūru.

Tālāk notiek pieprasījums no katra organisma apstrādāt iedoto problēmu. Kamēr katrs organisms apstrādā problēmu tiek ievākta satītstika un izpildes profildati.

Pēc šī processa beigām tiek veikts novērtējums. Novērtējums vienmēr tiks balstīts uz organisma doto rezultātu. Tomēr organismi tiek vērtēti vairākos parametros papildus problēmas risinājuma rezultātam. Tādi kā maksimālais mūža garums, pielāgošanās vērtējums, attīstības ātrums, resursu patēriņš rezultāta iegūšanai. (Šis saraksts varētu mainīties pēc izstrādātāja ieskatiem)

Kad tiek pabeigts novērtējums tiek sākts atlases process. Kurš jau tika nedaudz aprakstīts pie Dabīgās Atlases tēmas. Atlases beigās tiek konstatēti visi organismi kuru mūžs ir beidzies un tie tiek izņemti no Pasaules uzskaites.

Viss processu kopums ir uzskatāmāk parādīts 1. attēlā



1. attēls

Tātad Pasaulei arī vairāki moduļi:

- * Statistikas modulis;
- * Novērtējuma modulis;
- * Atlases modulis;

Tā kā Pasaule, kā arī organismi ir būvēta no moduļiem, kuri tiek izsaukti viens pēc otra, tad lietotājs arī šeit spēj pievienot savus moduļus kuri tiks izsaukti secībā, kā tas tiks norādīts. Tā rezultātā pasaule arī ir pietiekoši veikla un pielāgojama lietotāja vajdzībām, tik ilgi kamēr lietotājs pareizi sagatavo moduļus.

Pimēram, varētu ievietot moduli kurš simulētu dabas katastrofas. Tā rezultāti būtu organismu centieni pielāgoties pēkšņām izmaiņām. Spēcīgi zari varētu palikt daudz vājāki, jo viņi bija pārāk labi pielāgojušies konkrētam Pasaules stāvoklim, citi bija daudz veiklāki un ātri pielāgojās jauniem likumiem, kā rezultātā palika par daudz veselīgāku un stirpāku zaru.

Cits variants varētu būt iedvesmots no datorspēļu stratēģijas spēlēm un simulētu cīņu starp zariem, kā arī veidotu apvienības un pieteiktu karus citiem zariem. Tāds modulis gan prasītu arī specifisku kaujas moduli organismiem. Tomēr kopējais modelis atļauj to darīt.

Ir vērts pieminēt to ka visu organismu izmantota atmiņa atradīsies kopīgā izdalītā blokā, līdz ar to vienu organismu darbības varēs ietekmēt citu organismu darbību. Tā tiks simulēta cīņa par resursiem, dotajā vietā – atmiņu. Sākotnēji katram zaram tiks iedots vienlīdzīgs atmiņas daudzums, bet ar laiku balstoties uz organisma veselības rādītājiem zars kopumā var iegūt vairāk atmiņas no vājiem organismiem.

2.3.2 Fiziskā struktūra

Fiziski eksistēs viena izpildāma programma – Pasaule un vairāki pirmās paaudzes organismi kuri būs savu zaru pamatlicēji. Organismi arī būs nokompilētas programmas. Pasaule ar līdzīgu C++ valodas funkciju `system()` izsauks katru organismu un saņems tā rezultātu. Rezultāti var būt saņemti gan kā teksta faila ierakstītas atbildes, kuras tiks parsētas uz saistīto sarakstu, vai arī uzreiz saņemts `return value` no izpildāmās programmas un apstrādāts tālāk neizmantojot failu priekš starprezultātiem.

Organismiem nepieciešamo atmiņu tie pieprasīs no Pasaules izdalītā kopējā atmiņas bloka. Tā ir atmiņa par kuru cīnīsies visi organismi. Jo vairāk atmiņas ir pieejams kādam zaram, jo vairāk tie spēj reproducēties zara ietvaros, līdz ar to dotā zara organismi paliek stiprāki attiecībā pret pārējiem.

Atmiņas daudzumi tiks palielināti vai samazināti balstoties uz statistikas datiem par veselības kritērijiem.

Visi ģenerēti objekti tiks saglabāti specifiskā mapē, katram zaram eksistēs sava mape. Tas tiek darīts lai vēlāk varētu novērot attīstības gaitu katram atsevišķam zaram. Kā arī tiks glabāts žurnāls par izdarītajiem lēmumiem attiecībā pret katru zaru.

Organismi paši tiks salikti no jau minētajiem un pirms tam sagatavotiem moduļiem, izņemot pašu pirmkodu organisma pirmkodu.

2.3.3 Statistikas modulis

Statistika tiek vākta kā izpildes profildati ar specifiskām funkcijām kuras tiek ieliktas ap pirmkoda moduļiem. Ar šo statistiku tiek noteikts to pielāgojums un resursu nepieciešamība. Tāda profilēšana notiek ar jebkuru ģenerētu moduli, vēlāk šie dati parādās pie moduļu saraksta un ar ievākto datu palīdzību tiek izvēlēti moduļi mutācijas laikā

Pasaule arī seko līdzi katra organismam vajadzībām un produktivitātei. Tas varētu līdzināties Windows operētājsistēmas uzdevumu pārvaldniekam (task manager). Šis ir otrs darbs kuru veic statistikas modulis – pasniedz informāciju lietotājam veigli lasāmā un saprotamā veidā. Iespējams šī funkcionalitāte tiks veidota atsevišķā saskaranes modulī, ja radīsies parāk sarežģītas saskarnes attēlojuma prasības.

2.3.4 Novērtējuma Modulis

Novērtējuma moduļa mērķis ir atšķirt veiksmīgus organismus no neveiksmīgiem un mazāk efektīviem. Novērtēšana notiks ar ievāktās statistikas palīdzību. Visi organismi centīsies risināt vienu problēmu. Kopā ar problēmu tiek ievietota funkcija, kura nosaka organisma problēmas apstrādes rezultāta attālumu no vēlamā rezultāta, kas ir galvenais kritērijs organisma efektivitātei. Jo tuvāk vēlamajam mērķim – jo lielāka iespēja ka šis kods būs spējīgs reproducēties.

Novērtēšanas gala rezultāts ir saraksts ar visiem organismiem kuri ir sakārtoti pēc savas efektivitātes. Kā jau minēts novērtēšana notiek pēc šādiem kritērijiem:

- * problēmas risinājuma rezultātam (galvenais kritērijs)
- * maksimālais mūža garums (otrās pakāpes kritērijs)
- * attīstības ātrums (otrās pakāpes kritērijs)
- * efektīvs resursu patēriņš rezultāta iegūšanai (otrās pakāpes kritērijs)
- * reprodukcijas skatītājs (otrās pakāpes kritērijs)

Protams problēmas risinājuma rezultāts ir galvenais kritērijs, jo ja programma izpilda doto uzdevumu, tad tas ir daudz labāk nekā ļoti labs algoritms, bet kurš strādā tikai periodiski vai pie noteiktiem nosacījumiem.

Otrās pakāpes kritēriji visi kopā veido tik pat spēcīgu atlases vērtējumu, kā rezultāta tuvums vēlāmajām. Kopā visi šie kritēriji parāda cik veselīgs ir organisms, vai tas neizšķērdē pārāk daudz resursu – resursu daudzuma attiecība pret iegūtā rezultāta attālumu. Ja mums vajag ļoti daudz procesora laika un/vai atmiņas bet rezultāts ir pilnīgi garām, tad tādi organismi nespēj iegūt pietiekoši labu veselības novērtējumu.

Vai tas pedējā laikā nav iestindzis stagnācijā - atšķirībā sava zara vecāku un vecvecāku ir diezgan niecīga, līdz 5%. Tieši šo iemeslu dēļ tiek saglabātas atsauces uz vecākiem organisma faila nosaukumā. Mūža garums tiek noteikts pēc vecāku savietojamības. Kad abu vecāku organismu pirmkodi spēji būt savienoti bez daudzām problēmām, tad attiecīgi vecums tiek palielināts, izņemot tos gadījumus, kad abi vecāki ir no viena zara.

Reprodukcijas skaitītājs nosaka cik atvases var būt dotajam organismam viņa mūža garumā. Tomēr šis skaitītājs negarantē to ka visas atvases tiks izveidotas. Tas arī ļauj eksistēt tādiem organismiem kuri cenšas saglabāt savu zaru balstoties tikai uz reprodukcijas ātrumu un daudzumu, nevis efektivitāti vai mūžu. Liels reprodukcijas skaits palielina mutācijas iespējas zara ietvaros. Lielais reprodukcijas skaitītājs ir līdzīgs piemēram prusakiem, kuri ļoti ātri pielāgojās pie dažādām indēm.

Tātad visi otrās pakāpes kritēriji summā izveido veselības kritēriju, kurš pēc svara ir vienāds ar problēmas risinājuma rezultāta attāluma kritēriju. Tāda pieeja ļaus attīstīties ātri augošiem zariem, jeb stipriem zariem. Šie zari arī varētu iegūt atmiņu no tādiem organismiem kuri spēj iedod daudz labāku gaidāmu rezultātu, bet tai pašā laikā izšķērdē tam resursus un ir iestinguši uz vienas vietas. Tāds zars vienkārši tiks izkonkurēts un ar laiku tam nepaliks atmiņas.

2.3.5 Atlases Modulis

Par dabīgo atlasī jau minēju pie Dabīgās Atlases apakšnodaļas. Šis modulis balstoties uz ievāktu statistiku sagatavos sarakstu ar potenciāliem vecākiem un vēlāk atfiltrēs 4% efektīvāko un 1% varbūtiski paņemtu organismu no iespējamo vecāku saraksta. Nākošais solis ir mēģināt šos organismus sapārot.

Pirmais solis ir atrast otro organismu uz kuru pamata tiks veidots jauns organisms. Savienojamo kodu sākumā meklē citos attīstības zaros. Ja otra līdzīga organisma nevar atrast citos zaros, tad tiek

veikta meklēšana savā zarā, bet tādu organismu atvases tiks uzskatītas par vājiām un tiām būs īsāks mūžs.

Lai divi organismi būtu savienojami, to pirmkodu sarakstiem ir jābūt līdzīgiem vismaz pusē no saraksta apmērā un līdz trim ceturtdaļām. Pieņemu ka šie cipari būs daudz precīzāki, kad tiks veidota pati programma.

Sapārošanas process notiek no statistiski labākā organisma un virzoties pa sarakstu uz leju. Visticamāk efektīvākie organismi tiks sapāroti savā starpā, kas veicinās to stiprumu nākotnē. Ja netiek atrasts neviens organisms kurš būtu savietojamības kritēriju robežās tad dotais organisms nebūs spējīgs šajā ciklā izveidot jaunu organismu un tiks izņemts no potenciālo vecāku saraksta. Šīs darbības tiek pielietotas pret visiem potenciāliem vecākiem. Kad abi vecāki ir atrasti notiek process kurš tika aprakstīts Reprodukcijas moduļa nodaļā. Pibildīšu ka vecāki ar bērniem nevar būt sakrustoti, kaut gan tas ir diskutējams jautājums.

Atlases modulis arī novērtē jauna organisma savietojamību ar abiem vecāku zariem un ievieto to vajadzīgā zarā. Ja abu vecāku savietojamība ar jaunu organismu ir zem 50%, tad tiek izveidots jaun zars, kura pamatlicējis ir jaunizveidotais organisms.

Nobeigums

Galvenais mērķis bija izveidot pamatu nākotnes darbam. Šis mērķis arī sasniegts, jo dotajā rakstā ir kaut cik aprakstītas un strukturētas autora domas. Ir pateikts kādām īpašībām jābūt galvenām programmā. Ir pieņemti aptuveni skaitļu balstoties uz autora pieredzi un zināšanām, tomēr tie varētu būt neprecīzi un izstrādes laika tiks mainīti.

Pozitīvi ir tas, ka sistēma izveidojusies pietiekoši pielāgojama dažādiem papildus mērķiem, jo tā kopumā ir balstīta uz savietojamiem moduļiem. Lietotājs varētu veidot saviem mērķiem vajadzīgos moduļus balstoties uz piemēriem no jau uzrakstītiem moduļiem un pievienot tos programmai.

Negatīvi ir tas, ka viss šis teks ir virspusējs. Nav pietiekoši liela padziļināšanās visā processā, nav izstāstīti algoritmi. Tomēr šis darbs nebija tam mērķēts. Sākotnēji netika izvirzīts mērķis sagatavot gatavus algoritmus.

Vai izdosies izveidot pilnvērtīgu sistēmu, kura pildīs visas procedūras, kuras ir aprakstītas dotajā darbā varēs uzzināt nākošā gada laikā, jo autors plāno turpināt darbu saistībā ar šo tēmu.

Izmantotie materiāli

[1] Evolutionary Algorithms [tiešsaistē]

<http://www.geatbx.com/docu/algindex.html> [atsauce 18.06.2012]

[2] Wikipedia.org [tiešsaistē] http://en.wikipedia.org/wiki/Artificial_life [atsauce 18.06.2012]

Diena [tiešsaiste]. Rīga: a/s. Diena, 2000- [atsauce 10.01.2004]. Pieejams Internetā:
<http://www.diena.lv>. ISSN 1407-7833.

Pielikums

1. **pielikums. Darba angliskā versija**

Dokumentārā lapa

Kursa darbs “Pamatmodelis Moduļu-bāzētai Mākslīgai Dzīvībai” izstrādāts LU Datorikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti.

Autors: Jānis Knets

Rekomendēju/nerekomendēju darbu aizstāvēšanai

Vadītājs: M. Dat. Andrejs Hanins

Darbs iesniegts Datorikas fakultātē

Dekānā pilnvarotā persona:

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

Komisija: