**Teмa 10. Генетични алгоритми**

Наред с многото приложения за оптимизации, генетичните алгоритми се използват за оптимизиране на невронните мрежи. Както е известно последните са основен метод за Data Mining, прилаган в БИС. Това е причината за разглеждане на тази тема в курса.

10.1. Еволюционни изчисления.

Идеята за еволюционни изчисления е била представена за първи път през 1960 от I.**Rechenberg** (Берлински технически университет) в неговия труд "Еволюционни стратегии". Представените идеи след това са развити от други изследователи. В еволюционните алгоритми се използват трите основни принципа на естествената еволюция, описани от Darwin: репродукция, естествен подбор и разнообразие на индивидите. През 60-те години на 20 век тези характеристики на естествената еволюция вдъхновяват изследователите да създадат следните методи за популационна оптимизация с общо име **еволюционни изчисления**:

• Генетични алгоритми;

• Генетично програмиране;

• Еволюционно програмиране.

Генетичните алгоритми (ГА) представляват фамилия от изчислителни модели за оптимизация, основани на свойствата на биологичния прототип. Те са въведени от John Holland (професор в Университета в Мичиган) и D. Goldberg (Университет в Мичиган) през 1970г. През 1975 от печат излиза книгата на Holland "*Adaption in Natural and Artificial Systems*" (Адаптация в естествените и изкуствени системи), където са представени основните идеи.

Терминът ГА се използва в два аспекта. В тесен смисъл ГА се разглеждат като прилагане на модела на Holland- алгоритъм, който се опитва да получи достатъчно добри решения чрез създаване на структури от данни, които да ги представят. Оценяват се тези структури, комбинират се решенията за получаване на нови, следва ново оценяване и т.н.

В широк смисъл, ГА е всеки, базиран на популации изчислителен модел, който използва генетични оператори, за да генерира нови точки в пространството на търсене.

ГА често се описват като методи за глобално търсене в пространство от възможни решения, които не използват градиентна информация. Принадлежат към т.нар. „слаби” методи на ИИ, тъй като правят относително малко допускания за решавания проблем.

Изборът на най- добрите обекти е ключова евристика на всички еволюционни методи, позвoляващи ускореното намиране на решение. Генетичните алгоритми неколкократно превишават по бързина пълното (сляпото) търсене в множество задачи.

Традиционно използваните алгоритми за решаване на задачите имат наред с предимствата, трудноразрешими недостатъци. Някои от тях са:

- Като изходно се използва само едно решение;

- Работят с ограничен брой елементи, които за икономически задачи с големи размерности са недостатъчни;

- Голяма част от алгоритмите се установяват на локални решения, които може да бъдат далеч от оптималните;

- Чувствителни са към избора на начални условия.

При еволюционните изчисления като начални се използват няколко решения, а не едно, които може да бъдат получени като резултати от детерминирани, стохастични или хибридни алгоритми. Тези решения по- нататък се обработват чрез генетични оператори и в съответствие с принципите на еволюцията. Паралелната обработка на множество решения позволява търсене на по-перспективните от тях. За избягване на бързата сходимост на алгоритъма в локален оптимум се използват различни генетични оператори, схеми на селекция и различни видове търсене.

## 10.2. Биологична основа на генетичните алгоритми.

Генетичните алгоритми прилагат понятийния апарат от биологията, начините на представяне и предаване на генетична информация, идеята за предпочитане на най- приспособените индивиди.

Всички живи организми се състоят от клетки. Във всяка клетка има еднакво множество от хромозоми. Хромозомите представляват низове от ДНК (дезоксирибонуклеинова киселина) и служат като модел на целия организъм. Известно е, че човешката ДНК се състои от 4 нуклеидни киселини, т.е. има 4 битов код. Съществуват 48 хромозома, определящи отделния индивид, които съдържат 3 милиарда такива битове. Тогава всички възможни индивиди са 4 3 000 000 000. Уместен е въпросът: как природата решава задачата да усъвършенства човешкия вид при толкова голямо многообразие? Отговорът може би е в Дарвиновата еволюционна теория. Тя утвърждава като основна направляваща сила на еволюцията естественият подбор, т.е. оцеляването на най-силният (в широкия смисъл на понятието).

Хромозомите се състоят от гени. Това са блокове на ДНК. Всеки ген кодира конкретен протеин (белтък). Това може да е личен белег (цвят на очите, форма на ушите, вид на косата, наследствени болести и др.). Възможните комбинации на белега се наричат алели. Например, къдрaва и права коса са алелите на гена за вид на коса. Всеки ген има своя позиция в хромозомата. Пълното множество от генетичен материал (всички хромозоми) се нарича геном. Конкретно множеството гени в генома се нарича генотип.

В репродуцирането на индивидите се използват няколко основни механизма: селекция, кръстосване, мутация и инверсия.

При кръстосване се генерира нова хромозома, обединяваща генетичен материал на двама родители. Това променя белезите на индивида и му дава по – богати възможности за приспособяване.

Мутацията представлява изменение в даден ген, което води до изменения в структурата на хромозомите. Обикновено мутацията се предизвиква от грешка при копирането на гените от родителската хромозома. Така шансовете за оцеляване на индивида се променят (подобряват се или се влошават). Коя мутация е успешна и коя не се определя в процеса на естествен подбор.

Според Дарвин селекцията представлява естествен подбор, при който най – добрите индивиди от потомството оцеляват и създават следващо поколение, като по този начин предават от своите индивидуални характеристики на потомството.

Съвкупността от „индивиди” от един вид, които се размножават успешно и създават поколение се нарича **популация**. Oбикновено новата популация се получава след подбор на най – годните хромозоми от старата.

От гледна точка на изкуствения интелект, еволюцията е добър пример за механизъм на адаптация.

**10.3. Oсновни термини при работа с ГА.**

**Хромозома-** структура от данни, моделираща биологична хромозома, която представлява потенциално решение на разглеждания проблем. В нея са кодирани параметрите на задачата.

**Ген**- изграждащ елемент на хромозомата. Кодира един параметър от задачата.

**Алел**- стойност на признака, кодиран в даден ген.

**Популация-** множество от хромозоми, представляващи възможни решения на определен етап от ГА.

**Пространство на търсене-** пространство на възможните решения.

**Фитнес функция** (оценъчна функция, функция на приспособеност, целева функция)- специално създавана функция, с която се измерва оптималността на решението (хромозомите). Тя е важен компонент в ГА и се създава за всяка конкретна задача. Обикновено дава част от описанието на проблема и е близо до целта на алгоритъма. За това се казва, че **единствена съдържа знания за решаваната задача**. Фитнес функцията трябва да позволява намиране на разнообразни решения и да се изчислява бързо. Последното влияе върху скоростта на работа на ГА. Може да бъде два вида- непроменлива и мутираща.

Еволюционният процес съответства на процес на търсене в пространството от възможни решения.

**Рекомбинация-** процес, в който възникват нови комбинации от гени.

**Репродукция-** създаване на нови хромозоми в резултат на рекомбинация на гени на родителски хромозоми.

#### 10.4. Кодиране на гените в генетичните алгоритми.

Кодирането на гените е първата стъпка, която се прилага при решаване на задача с използването на генетични алгоритми. В класическия вариант, предложен от John Holland, тези елементи представляват битове, които могат да приемат стойности от 0 и 1. Използва се и шаблона “\*” за заместване както на 1, така и на 0. Например „1\*0\*\*1”.

Запредставяне на множество от хромозомисе използва т.нар. **схема. Схемата** е шаблон, образуван от единици, нули и знака „\*”. Например схемата „\*\*\*\*\*\*\*\*” представя всички двоично кодирани хромозоми с дължина 8, т.е. 28 степен броя.

Броят на фиксираните позиции- 1 или 0 в схемата се нарича „**ред** на схемата”. Например редът на схемата „1\*\*00\*11” е 5.

Използват се следните видове кодиране: двоично, по номера на гена в хромозомата, по стойности и дървовидно.

**Двоичното кодиране** е основният начин за кодиране на гените. Той се е използвал при първите разработки на генетични алгоритми. При двоичното кодиране всяка хромозома представлява последователност от битове (0 и 1). Например:

|  |  |
| --- | --- |
| Хромозома 1 | 110101001010101100010100 |
| Хромозома 2 | 111100011100110111011100 |

Изразът 6+5 \* 4/2 +1 след двоично кодиране ще изглежда така: 0110 1010 0101 1100 0100 1101 0010 1010 0001. Тези гени формират хромозомата 011010100101110001001101001010100001.

Този начин на кодиране дава много възможни хромозоми, дори и с малък брой гени. От друга страна, това кодиране, често не е естествено и понякога трябва да се правят корекции след кръстосване и/или мутация.

Главното предимство на двоичното кодиране се изразяват в използване на минимално количество памет.

Недостатъците са:

* Тъй като цялото число е в двоичен формат, точката на разделяне при кръстосване може да попадне между битовете на една цифра и обменът на части да доведе до изменение на числото.
* Има възможност да се появят забранени комбинации от гени.
* Необходимо е преобразуване на параметрите при кодиране и декодиране на решението към и от двоичен формат.

При представяне на числата чрез генетичните алгоритми е удобно да се използва двоичната бройна система. Но при нея не винаги се осигурява разликата между записванията на две съседни числа да бъде 1 **бит**. Например двоичното представяне на 7 е 0111(2), а на 8 е 1000(2). Този проблем не се отнася до всички подобни представяния. Например за числата 2=0010(2),3=0011(2) и др. той не съществува. Но възможността за по-големи разлики от един бит между две съседни двоични числа, представлява проблем при задачи с използване на генетичните алгоритми (например разпознаване на образи). За това е въведен кода на Грей. При него разликата в кодирането между две съседни числа е винаги 1 бит (таблица 1). С използването на кода на Грей прилагането на генетичните оператори води до по-естествено представяне на индивидите.

Taблица 1

|  |  |  |
| --- | --- | --- |
| Десетично число | Двоично число | Код на Грей |
| 0 | 0000 | 0000 |
| 1 | 0001 | 0001 |
| 2 | 0010 | 0011 |
| 3 | 0011 | 0010 |
| 4 | 0100 | 0110 |
| 5 | 0101 | 0111 |
| 6 | 0110 | 0101 |
| 7 | 0111 | 0100 |
| 8 | 1000 | 1100 |
| 9 | 1001 | 1101 |
| 10 | 1010 | 1111 |
| 11 | 1011 | 1110 |
| 12 | 1100 | 1010 |
| 13 | 1101 | 1011 |
| 14 | 1110 | 1001 |
| 15 | 1111 | 1000 |

При **кодирането на пермутации**, всяка хромозома е поредица от числа, които представляват номера на гена в хромозомата. Например:

|  |  |
| --- | --- |
| Хромозома 1 | 4 5 7 12 6 1 3 9 8 10 2 11 |
| Хромозома 2 | 9 12 11 6 7 2 5 3 1 4 10 8 |

**Кодирането по стойност** представлява директно задаване на определени стойности (реални числа, символни низове, потребителски типове и др.) в хромозомата. Например:

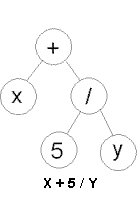
|  |  |
| --- | --- |
| Хромозома 1 | (black), (red), (white), (blue), (pink) |
| Хромозома 2 | “SLDKRJGHALWIEOTUAGALRIWJS” |
| Хромозома 3 | 8.9543 1.8653 0.1856 2.3962 5.2187 |

Предимства на кодирането по стойност са:

* задаването на параметрите непосредствено;
* при кръстосване точката на деление е винаги между числата и значението им не се изменя;
* всяко значение може да бъде изменяно на всяко друго с една стъпка на оператор мутация.

Недостатък на кодирането по стойност е излишният разход на памет, тъй като за съхранение на всеки параметър винаги се ползват по 4 байта.

При **дървовидното кодиране** всяка хромозома представлява дърво (фиг. 2) от някакъв тип обекти (функции или команди на език за програмиране). Например:



Фиг. 2. Дървовидно кодиране на хромозома

Дървовидното кодиране е удобно за прилагане при разширяващи се програми. Това кодиране често се използва при разработки на езика за програмиране LISP.

### 10.5. Основни операции в генетичните алгоритми.

**Селекция**

Това е метод, за избор на родителските хромозоми, които да участват при формиране на новото поколение. За да се изберат най – добрите, всички хромозоми се подлагат на оценяване чрез фитнес функцията до колко отговарят на изискванията на задачата. След това обикновено се селектират най – добрите (годните). За да се разшири пространството на търсене в някои случаи се избират и индивиди с по- ниски стойности на оценка. Така се очаква да се получи поколение, което по- късно да донесе разнообразие и по- добри качества.

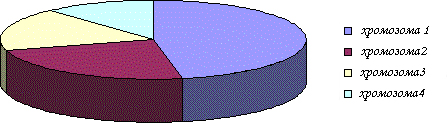
Съществуват множество подходи за селекция на хромозомите. Някои от тях са селекция по кръгова рулетка, „турнир” и др.

**-Селекция по кръгова рулетка**

При селекцията по кръгова рулетка (фиг. 73) всяка от хромозомите участва със стойността на своята фитнес функция. Колкото по – голяма е тя, толкова по – голям шанс има хромозомата да бъде избрана. За онагледяване на метода ще си представим една кръгова рулетка, където са поставени всички хромозоми от популацията. Всяка хромозома има сектор, чиято големина съответства на стойността на фитнес функцията. Изборът се онагледява с хвърляне на топчето в рулетката. Хромозомите, които са по – жизнеспособни (с по- голям сектор), имат по- висок шанс да бъдат избрани.

При всяко завъртане на рулетката се избира една хромозома за новата популация.

Основен недостатък на селекцията по кръгова рулетка е, че хромозомите с по- широк сектор, например 90%, намаляват шанса на останалите да бъдат избрани.



Фиг. 3. Селекция по кръгова рулетка

##### Когато хромозомите имат приблизително еднакви стойности на фитнес функцията, техният шанс да бъдат избрани е почти равен. Тогава се стига до случайно търсене. Едно от решенията на проблема е с всяка хромозома да се асоциират две стойности: тази на фитнес функцията и една нормализирана оценка, наречена „сила” на хромозомата. За определяне на силата се използват методите:

- линейна нормализация. Хромозомите се подреждат по низходящите стойности на фитнес функциите си. На тази с най- високата стойност се присвоява определена произволно константа. Всяка следваща хромозома получава стойност, която е намалена от тази на предходната с друга константа.

- рамкиране. Използва се най- ниската стойност на фитнес функция за разглежданите хромозоми. На всяка от тях се присвоява сила, която е раликата между стойността на фитнес функцията й и тази минимална стойност.

С тези две стойности за всяка хромозома се постига по- добро оценяване и съответно по- качествена селекция.

##### Селекция „турнир”

При нея се избират по случаен начин две хромозоми. Избира се някакъв праг, свързан със задачата, например К, и се генерира случайно число k в интервала 0- 1. Ако k < K, то се избира първата хромозома, а в противен случай- втората.

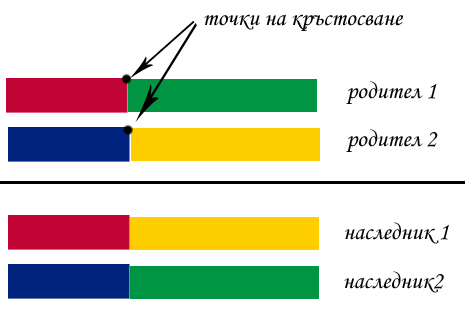
* Елитарност. За да се съхранят добри решения (хромозоми) в след ващите поколения, се въвежда параметър, с който се задава какъв процент от тях да премине в новата генерация. Така хромозоми с високи оценки се запазват за постигане на по- добро крайно решение.

#### Кръстосване

#### При кръстосване се обменят гени от родителските хромозоми. Избирането на случайни и силно отличаващи се хромозоми способства за повишаване на генетичното разнообразие.

#### Основни видове кръстосвания са следните:

##### - едноточково кръстосване. Избира се по една точка от двете родителски хромозоми, която трябва да има една и съща стойност (фиг. 74). Новото поколение се формира като се разменят частите от двете хромозоми преди и след точката на кръстосване.

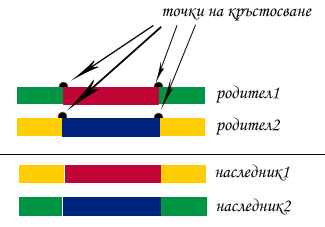


Фиг. 4. Схема на кръстосване в една точка

Обикновено не всички родителски хромозоми се кръстосват. Препоръчва се част от старата популация да оцелее т.е някои от хромозомите да не се кръстосат и да се копират в новата популация

##### - Кръстосване в две точки

Кръстосването в две точки е аналогично на кръстосването в една точка, но тук се избират две точки на кръстосване (фиг. 5).



Фиг. 5. Схема на кръстосване в две точки

Потомството се получава след като гените преди първата и след втората точка на кръстосване в родител1 се комбинират с гените след първата и преди втората точка на кръстосване в родител2, при което се формира наследник1. Когато гените преди първата и след втората точка на кръстосване от родител2 се комбинират с гените след първата и преди втората точка на кръстосване от родител1 се формира наследник2.

Използва се и n- точково кръстосване.

* **Шаблонно** кръстосване.

Техниката работи отново с две родителски хромозоми и един създаден шаблон. В последния гените приемат по случаен начин 1 или 0. Първият наследник приема от първия родител гените, за които има в шаблона 1, а от втория родител гените, за които в шаблона има 0. Вторият наследник, обратно: приема от родител1 гените, за които има в шаблона 0, а от втория родител гените, за които в шаблона има 1. Пример, фиг. 6:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Родител 1: | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| Родител 2: | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| Шаблон: | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
|  | | | | | | | |
| Наследник 1: | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| Наследник 2: | 1 | 1 | 0 | 1 | 0 | 0 | 1 |

Фиг. 6. Пример за шаблонно кръстосване

##### -Кръстосване, основано на подреждане

Селектират се две родителски хромозоми. Създава се шаблон с генерирани по случаен начин нули и единици. Наследник 1 приема от родител 1 гените, на чиито места в шаблона има 1-ци. Създава се списък от гени на родител 1, на чиито места в шаблона има 0-ли. Този списък се сортира, така че гените в него да са подредени така, както в родител 2. Свободните позиции в наследник 1 се попълват като се заместят с гените от сортирания списък. Аналогично се работи и с другия наследник.

Пример: нека разгледаме точки от транспортна задача (а, b, c, d, e ,f, g), кодирани като гени в родителските хромозоми (фиг. 7)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Родител 1: | A | b | c | d | E | f | g |
| Родител 2: | E | c | f | b | D | g | a |
| Шаблон: | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
|  | | | | | | | |
| Наследник 1: | f | b | c | d | E | a | g |

Фиг. 7. Пример за кръстосване, основано на подреждане

Използват се и други методи за кръстосване- циклично, частично съвпадащо и др.

**Вероятността за кръстосване** показва колко често ще се извършва кръстосване. Ако няма кръстосване, поколението е точно копие на родителите. Ако има кръстосване, поколението е направено от части на родителските хромозоми. Ако вероятността за кръстосване е **100%**, тогава цялото поколение ще бъде създадено чрез кръстосване. Ако е **0%**, цялото ново поколение ще бъде точно копие на хромозомите от старата популация (това не означава, че новото поколение е същото). Кръстосването се извършва с надеждата, че новите хромозоми ще имат добри части от старите хромозоми и може би новите хромозоми ще бъдат по- добри.

**Мутация**

Мутацията представлява промяна на гените в хромозомата на случаен принцип. Обикновено се прилага, за да се обогати новото поколение. Необходима е, когато родителските хромозоми имат един и същ ген в някоя от позициите си. Тогава кръстосването няма да може да предизвика промяна.

##### При двоичното кодиране мутацията се реализира с инвертиране на бит от 0 в 1 или обратно.

При кодиране на **пермутации** мутацията представлява размяна на два гена в хромозомата.

###### При числови стойности в гените на хромозомата мутацията представлява промяна на стойността, която се осъществява чрез събиране или изваждане на малко число със/от съответната стойност в гена. Числото, с което ще се извършва мутацията се задава при решаване на задачата.

При **символни** стойности в гените на хромозомата мутацията се осъществява чрез замяна в избран символ в хромозомата с друг.

###### При низови стойности в гените, мутацията представлява промяна на низа чрез замяната му с друг.

При дървовидното кодиране мутацията е два вида- във възлите и в листата на дървото.

**Вероятност за мутация** означава колко често части от хромозомата ще мутират. Ако няма мутация, поколението се взема след кръстосване (или копиране) без никакви промени. Мутацията е въведена за предпазване на ГА от попадане в локален екстремум, но не трябва да се появява твърде често, защото тогава ГА в действителност ще се промени в произволно търсене.

Използва се и параметър **скорост на мутация**, който показва какъв е шансът един ген да бъде променен. Типична стойност е 0.008.

Ако една популация e с размер **m** от хромозоми с дължина **n**, то е възможно да бъдат генерирани между 2n и m \* 2n схеми. Holland доказва, че най- малко m3 oт тях не се променят при кръстосване и мутация.

ГА обработват популация от **m** индивида, но с много повече съдържащи се в тях схеми, без допълнителна памет. Holland нарича това качество „имплицитен паралелизъм”.

Схемите, които са по-къси на дължина и са с по-нисък ред са с по- голям шанс да не се променят при кръстосване и мутация.

Установено е, че мутацията се прилага на по- късен етап от изпълнението на генетичния алгоритъм, когато пространството на търсене е намалено. В началото на ГА от по- голямо значение е кръстосването. Това означава, че с развитието на ГА трябва да се повишава скоростта на мутация, а тази на кръстосване да се намалява.

**10.6. Същност на генетичния алгоритъм.**

Генетичният алгоритъм е опростен модел на еволюцията в природата, реализиран във вид на компютърна програма с цел решаване на определена задача. В него се използват аналозите на механизма на генетичното унаследяване и на естествения подбор.

При генетичните алгоритми видът на изкуствените “индивиди” (хромозомите) е точно определен и е с **фиксирана дължина**.

Основните етапи на работа на генетичния алгоритъм могат да се изразят в следния ред:

1. Избор на кодиране на потенциалните решения като хромозоми.
2. В момент t= 0 се инициализира популация от кандидати за решения x1, x2,…,xn. Всички хромозоми се генерират случайно. За целта се използва генератор на случайни числа.

Оптимални стойности за размера на популацията се установяват според обема на обработваните данни и пространството на търсене. При големи размери на популацията алгоритъмът може да се забави твърде много и това да го направи неефективен. Параметърът силно зависи от вида на кодирането и от размера на хромозомата.

3. Изчислява се значението на фитнес функцията за всеки член от популацията. Изчислява се и средната за популацията стойност за същата функция. Мярката за класификация се установява като процентно отношение между двете величини. Хромозомите с по-висока стойност на този критерий от зададена стойност, ще участват в процеса на рекомбинация (възпроизводство) с по- голяма вероятност. В някои случаи се използва само стойността на фитнес функцията като мярка за класификация.

4. Селектира се необходимият брой хромозоми. Обикновено размерът на популацията е постоянен.

5. С помощта на генетичните оператори се създава следващо поколение. Новите индивиди се получават в резултат на кръстосване и/или мутация. Необходимо е да се следи за поява на “нежелани” свойства в популацията, например такива които променят изискванията на задачата.

6. Заменят се хромозоми от популацията с техни потомци като се отчитат стойностите на мярката за класификация. Обикновено се заменя фиксиран процент от най-слабите кандидат решения. В някои случаи се отстраняват някои от по-силните представители, като се очаква съхранените по-слаби да допринесат по-късно за получаване на по-точно решение.

7. Установява се време t= t+1 и се повтаря цикъла 2-6 до достигане на условието за край.

Накратко идеята на ГА може да се представи така:

{ Инициализиране на популация;

Оценка на популацията;

Докато се удовлетвори критерий за край:

{ Селектиране на двойки за репродукция;

Рекомбинация и мутация;

Оценка на популацията;

}

}

За ГА е характерно:

* Многопосочно търсене. Обработва се множество от възможни решения, за разлика от останалите алгоритми за търсене, които обработват отделни точки от пространството на търсене.
* Заедно с това, се обменя информация между посоките на търсене, тъй като за създаване на нова хромозома се използва кръстоска на налични хромозоми. Обменените гени са тези, които предават информацията.
* Изполването на мутация позволява да се променят някои гени и така да се въведе нова информация. Възможно е решението да се подобри.

Важен параметър на ГА е **размерът на популацията**. Той се изразява чрез броя на хромозомите, включени в популацията (в едно поколение). Често използван размер на популацията е около 20-30 хромозоми. Ако има твърде малко хромозоми, ГА ще има малки възможности за извършване на кръстосване и само малка част от пространството на търсене ще се изследва. От друга страна, ако хромозомите са твърде много, ГА ще се забави. Изследванията показват, че след определена граница (която зависи главно от кодирането и проблема) не е изгодно да се увеличава размера на популацията, защото това няма да доведе до по- бързото разрешаване на проблема.

При каноничните ГА всеки член на популацията е двоичен стринг с определена дължина, който кореспондира с кодирането на проблема. Началната популация се генерира като случайна комбинация. Тя се оценя с фитнес функцията и й се присвоява оценка за годност. Фитнес функцията (f) при каноничните ГА се дефинира като отношение между оценката (fm), асоцииранa с даден стринг m и средната оценка на всички стрингове в популацията (ḟ), т.е. f= fm/ ḟ.

При такъв подход за решаване на сложна многовариантна задачи не са необходими много знания. Необходим е само критерий за избор на най-перспективните индивиди. В него се включва и преценка на необходимостта от мутация. Този подход е от типа “проба- грешка”, при който се запазват само най-добрите решения на всеки етап. Но за намереното решение не може да се каже, че представлява глобален екстремум.

Търсенето на решение може да се сведе до намиране на екстремум на функция. Пространството на търсене може да е напълно известно по време на решаването на проблема, или да се знаят само няколко точки от него. Чрез тях се генерират други точки (техни наследници), при което процеса за намиране на решения продължава. Има много методи за откриване на **подходящо решение,** (често не най-доброто), като **изкачването на хълм**, **симулирано закаляване** и **генетичен алгоритъм**.

**Оценка на метода**

Генетичните алгоритми не отхвърлят веднага безперспективните решения, в противоположност например на евристичното търсене. Това е така, понеже липсва система за систематизация на следващите състояния. Просто се формира нова популация, всеки индивид от която, може да внесе своя принос за получаване на ново решение.

Генетичните алгоритми съдържат висока степен на паралелизъм. Всеки от перспективните клонове може да продължи да участва в търсенето на решения паралелно с останалите. По този начин се поддържат успоредно няколко решения и се реализира мощна форма за търсене на екстремум. Паралелизмът на генетичните алгоритми позволява лесна реализация на многопроцесорни компютърни архитектури, което не изисква специален хардуер както при невронните мрежи.

Генетичните алгоритми са адаптивни. Те могат да се обучават, да натрупват факти и знания, без предварително да ги имат. Те започват само с критерий за избор, който в процеса на решение може да се усъвършенства. Генетичните алгоритми могат да се ползват като обучващ се модул към една ЕС.

Генетичните алгоритми са пример как може да се преодолее детерминирания подход в програмирането чрез залагане на самообучението и адаптацията, започвайки от минимални знания. В тях се прилагат и правила за извод (фрагменти от програмния код), които еволюират в процеса на работа. Това е по-сложна форма на представяне на този метод.

**Пример за прилагане на генетичен алгоритъм**

Да се покаже прилагането на генетичния алгоритъм при намиране на максимума на функцията:

1

f(x,y)= ----------------

1 + x2 + y2

Решението ще се разглежда като хромозома с два гена- x и y. Първоначално за тях се избират произволни стойности и се създават хромозомите C0, C1, C2, C3 (Taблица 3).

Taблица 3

|  |  |  |  |
| --- | --- | --- | --- |
| **Хромозома** | **x** | **y** | **Годност** |
| **C0** | **-1** | **2** | **?** |
| **C1** | **-2** | **3** | **?** |
| **C2** | **1.5** | **0** | **?** |
| **C3** | **0.5** | **-1** | **?** |

Създадена е първоначалната популация от решения. Трябва да се оцени годността (да се изчисли фитнес функцията) за всяко от тях. За тази цел се изчислява f(x,y) за всяка хромозма и се записва в таблица 4.

Колкото по- висока стойност за годност се получи, толкова по жизнеспособна е хромозомата. Така че С3 и С2 съдържат най- добрите решения и може да служат като родителска двойка за следващи решения след прилагане на кръстосване. Като друга родителска двойка може да бъде С3 и С0, а С1 има малка вероятност да бъде избрана, поради ниската стойност на критерия за годност.

Таблица 4

|  |  |  |  |
| --- | --- | --- | --- |
| **Хромозома** | **x** | **y** | **Годност** |
| **C0** | **-1** | **2** | **0.167** |
| **C1** | **-2** | **3** | **0.007** |
| **C2** | **1.5** | **0** | **0.31** |
| **C3** | **0.5** | **-1** | **0.44** |

Кръстосването се изпълнява в точката между двата гена x и y. Новите хромозоми са показани в таблица 5. Отново се изчислява f(x,y). Сравняват се новополучените стойности с тези в таблица 4. Вижда се, че се получават по- високи стойности за f(x,y), т.е. решението се подобрява, тъй като се търси максимизиране на функцията. Средната стойност на критерия за годност в таблица 8 е 0.231, а в таблица 9 е 0.39.

Таблица 5

|  |  |  |  |
| --- | --- | --- | --- |
| **Хромозома** | **x** | **Y** | **Годност** |
| **Co(xc3,yc2)** | **0.5** | **0** | **0.8** |
| **C1(xc2,yc3)** | **1.5** | **-1** | **0.24** |
| **C2(xc3,yc0)** | **0.5** | **2** | **0.19** |
| **C3(xc0,yc3)** | **-1** | **-1** | **0.33** |

Селекцията на хромозоми и кръстосването продължава до достигане на поставено условие за край (например доказване на най-доброто решение).

Ако се приложи мутация чрез произволна промяна на гените с малки стойности (например 0.1), ще се получи разширяване на областта от решения с внасяне на нов генетичен материал. Така вероятността да се получи по- добро решение се увеличава.

**10.7. Приложения на генетичните алгоритми.**

Много оптимизационни задачи се решават чрез генетичните алгоритми (за търговския пътник, приложни физични задачи и др.). Оптимизационен процес може да се приложи, ако съществуват потенциални решения и се търси най- доброто от тях.

Генетичните алгоритми се прилагат също при комплексно изследване на функционални връзки и от тук индукция на правила, оптимизиране на входа на обучаващи системи и др.

Интересни са приложенията на генетичните алгоритми за оптимизиране на невронните мрежи. Те могат да се разглеждат в следните области:

1. За подготовка на обучаващото множество.
2. За подбор на параметрите, управляващи процеса на обучение. ГА се използват за избор на параметрите „скорост за обучение” и сила на сигнала (momentum). Адаптивната настройка на параметрите се разглежда като първи опит за еволюционна промяна на правилото за обучение.
3. За еволюционно обучение на НМ. Създава се популация от хромозоми, като всяка съдържа пълен набор от тегла в НМ. Предпочита се задаването им с действителните значения. След това се изпълнява процес на еволюция в цикъл от 4 стъпки, основан на ГА:

* «декодиране» на всяка хромозома за възстановяване на теглата и конструиране на невронни мрежи, със зададени тегла.
* Изчисляване на средноквадратичната грешка между фактическите и целеви значения на всички изходи на НМ в процеса на обучение. Тази грешка играе роля на фитнес функция за хромозомата и определя приспособеността й.

1 p

Е = --- ∑ (di- Outi )2

2 i=1

p- брой обучаващи примери

di -желан изход

Outi - фактически изход

* Репродукция на хромозомите с вероятност, съответна на тяхната приспособеност.
* Прилагане на генетични оператори за получаване на ново поколение.

Процесът продължава до постигане на зададена грешка на обучение.

При разгледаното еволюционно обучение се реализира т.нар. пакетно обучение, при което теглата се променят след като на НМ се подадат всички обучаващи примери (за разлика от backpropagation, където теглата се коригират след всеки пример).