Лекция 6. Генетични алгоритми

6.1. Мотивация

Генетичните алгоритми (ГА) предоставят един метод за самообучение, мотивиран от аналогия с биологичната еволюция. Вместо да търсят от по-общи към по-специфични хипотези или от по-прости към по-сложни, ГА генерират последователни хипотези чрез повтарящо се мутиране и рекомбинация на части от най-добрите хипотези, известни в текущия момент. На всяка стъпка колекцията от хипотези, наречена текуща *популация*, се обновява чрез заместване на някоя част от популацията с потомство на най-пригодени в момента текущи хипотези. Този процес представлява "лъчево" търсене на хипотези от типа "породи и провери", в който вариантите на най-добрите текущи хипотези са най-вероятни кандидати за по-нататъшното разглеждане. Популярността на ГА е мотивирана от няколко фактора, из между които са:

- Известно е, че еволюцията е успешен, устойчив метод за адаптация на биологичните системи.
- ГА могат да претърсват пространства на хипотези, съдържащи комплексни взаимодействащи се части, където влиянието на всяка част върху общата годност на хипотези трудно се поддава на моделиране.
- ГА лесно се поддават на разпаралеляване и могат да извлекат изгода от намаляване на цените на мощен компютърен хардуер.

6.2. Прототипичен генетичен алгоритъм

Задачата, адресирана от ГА, е претърсване на пространството на хипотези, за да намери най-добрата хипотеза. В ГА "най-добрата хипотеза" се дефинира като тази, която оптимизира определена, предварително дефинирана за конкретна задача числова мярка, наречена *годност* (fitness) на хипотези. Например, ако задачата за обучение се състои в апроксимиране на неизвестна функция по зададени обучаващи примери, описващи нейния вход и изход, то годността може да бъде дефинирана като точността на хипотезата върху тези обучаващи данни. Ако задачата е да се научи стратегията за игра на шах, годността може да бъде дефинирана като брой игри, спечелени от някой индивид, когато той играе срещу други индивиди от текущата популация.

Макар различните имплементации на генетични алгоритми варират в своите детайли, те са обикновено имат следната обща структура: алгоритмът работи чрез итеративно обновяване на едно множество от хипотези, наречено популация. На всяка итерация всички членове на популацията се оценяват в съответствие с функцията за годност. След това новата популация се генерира чрез вероятностен избор на най-пригодните индивиди от текущата популация. Някои от тези избрани индивиди преминават в следващата генерация на популацията без изменения. Други се използват като базис за създаване на ново

поколение на индивиди чрез прилагане на такива генетични операции като обмен на гени (crossover) и мутация.

Псевдокодът на един прототипичен генетичен алгоритъм е показан в Таблица 6-1. Входовете на алгоритъма са функцията на годност, подреждаща кандидат-хипотези, прагът, определящ приемливото ниво на годност за приключване на работата на алгоритъма, размерът на популация, с който ще работи алгоритмът, и параметрите, определящи как последващите популации трябва да бъдат генерирани: частта от популацията, която трябва да бъде заменена при всяка генерация, и скоростта на мутацията.

ГА(Годност, Праг_на_годност, p, r, m)

Годност: Оценъчната функция, която изчислява оценка на дадена хипотеза

Праг_на_годност: Прагът, определящ критерий за завършване на работата на алгоритъма.

p: Броят на хипотези, които да бъдат включени в популацията

r: Частта от популация, която на всяка стъпка трябва да бъде заместена чрез генетичен обмен

m: Скоростта на мутация

- *Инициализирай популация*: $P \leftarrow$ Генерирай по случаен начин *p* хипотези
- *Оцени:* За всяка *h* от *P* изчисли *Годност(h)*
- Докато $\max_h Годност(h) < Праг\_на\_годност$ направи

  Създай новата популация P_s:

  1. *Избери:* Избери по случаен начин *(1 - r)p* члена на *P* и ги добави към P_s. Вероятността $Pr(h_i)$ за избор на хипотеза h_i от *P* се задава от формула:

$$\Pr(h_i) = \frac{Годност(h_i)}{\sum_{i=1}^{p} Годност(h_i)} \quad (6.1)$$

  2. *Генетичен обмен:* Избери по случаен начин $\frac{r \cdot p}{2}$ двойки хипотези от *P* съгласно изчислената вероятност $Pr(h_i)$. За всяка двойка $<h_1, h_2>$ генерирай два потомка чрез прилагане на оператор за генетичния обмен. Добави всички потомци към P_s.
  3. *Мутация:* Избери по случаен начин с еднаква вероятност *m* процента от членове на P_s. За всеки един обърни избран по случаен начин един бит от неговото представяне.
  4. *Обновяване:* $P \leftarrow P_s$.
  5. *Оценка*: за всеки *h* от *P* изчисли *Годност(h)*

- Върни хипотеза от *P*, която има най-висока стойност на *Годност.*

Таблица 6-1. Прототипичен генетичен алгоритъм

В този алгоритъм всяка итерация от основния цикъл произвежда новата популация от хипотези, базирани на текущата популация. Първо, определено количество хипотези от текущата популация се избират за включване в следващата популация. Те са избират *вероятностно,* като вероятността за избор

на хипотезата се определя по ф-ла (6.1). По този начин вероятността, че една хипотеза ще бъде избрана, е пропорционална на нейната собствена годност и обратно пропорционална на годността на всички други конкуриращи се хипотези от текущата популация.

След като тези членове на текущата популация са били избрани за включване в следващата популация, допълнителните членове се генерират чрез прилагане на оператора за генетичния обмен. Генетичният обмен, описан подробно в следващия раздел, взима двойка от хипотези – родители от текущата популация и създава две хипотези – потомки чрез рекомбинация на части и от двата родителя. Родителските хипотези се избират вероятностно от текущата популация отново чрез използване на вероятностната функция, зададена от ф-ла (6.1). След като новите членове бяха създадени чрез операцията на генетичен обмен, новата популация вече съдържа желания брой членове. В този момент определена част m от тези членове се избират по случаен начин и се подлагат на случайни мутации, променящи избраните хипотези.

И така, описаният генетичен алгоритъм извършва случайно паралелно "лъчево" търсене на хипотези, които имат добро поведение съгласно функцията за годност. В следващите подраздели ще опишем подробно начин на представяне на хипотези и използваните в алгоритъма генетични оператори.

6.3. Представяне на хипотези

В ГА хипотезите често се представят чрез битови низове, позволяващи лесна обработка от страна на генетичните оператори, като мутацията и генетичния обмен. Например, множествата от АКО-ТО правила могат лесно да бъдат представени по този начин чрез избор на такова кодиране на правила, което назначава специфични под-низове за всяко пред- и пост-условие на правилото. Примерите на този начин на представяне на правилата в ГА-системата са описани в Grefenstette (1988) и DeJong (1993).

Преди да покажем, как АКО-ТО правила могат да се кодират чрез битови низове, да разгледаме, как такива низове могат да кодират ограничения върху значения на един единствен атрибут. За пример, да разгледаме атрибут *Небе*, който може да приеме три възможни значения: *Слънце, Облаци* и *Дъжд*. Един очевиден начин за представяне на ограничения върху атрибута е да използваме битов низ с дължина 3, в който всяка позиция на бит съответства на една от тези три възможни значения. Поставяне на 1 в някоя позиция указва, че атрибутът може да приеме съответната стойност. Например, низът 010 представя ограничението, означаващо че *Небе* трябва да приеме втората от тези стойности, т.е. *Небе* = *Облаци*. Аналогично, низът 011 представлява по-общо ограничение, позволяващо две възможни стойности, т.е. $(Небе = Облаци \vee Дъжд)$. Очевидно е, че 111 представлява възможно най-общо ограничение, указващо, че не ни интересува, коя от възможни стойности приема атрибутът.

Използвайки този метод за представяне на ограничения върху един единствен атрибут, конюнкцията от ограничения върху няколко атрибута може лесно да се представи чрез конкатенация (обединение) на съответните битови низове.

Например, да разгледаме вторият атрибут, *Вятър*, който може да приеме стойностите *Силен* и *Слаб*. Предусловието на правило от типа на

$$(Небе = Облаци \vee Дъжд) \wedge (Вятър = Силен)$$

може да бъде представено чрез следния битов низ с дължина 5:

Небе	*Вятър*
011	10

Пост-условието на правилото (като, например *Игра_на_тенис = да*) може да бъде представено по същия начин. Следователно, едно пълно правило може да бъде представено чрез конкатенацията на битови низове, описващи пред-условията на правило, заедно с битовия низ, описващ неговото пост-условие. Например, правилото: АКО *Вятър = Силен* ТО *Игра_на_тенис = да* ще се представи чрез следния низ:

Небе	*Вятър*	*Игра_на_тенис*
111	10	10

където първите три бита описват ограничението "не ни интересува" върху *Небе,* следващите два бита описват ограничението върху *Вятър*, а последните два бита – пост-условието на правилото (предполагаме, че *Игра_на_тенис* може да приеме стойностите *да* и *не*). Използването на ограничението "не ни интересува" позволява представяне на всяко правило чрез низ с фиксирана дължина, в който под-низът, разположен в определена позиция, описва ограничение върху определен атрибут. Използвайки този начин на представяне на едно правило, лесно можем да представим множеството от правила чрез проста конкатенацията на съответните битови низове, представящи отделните правила.

При създаване на кодиране чрез битовите низове за някое пространство от хипотези, трябва да се обръща внимание на факта, че всеки синтактически коректен битов низ трябва да описва една правилно определена хипотеза. Например, обърнете внимание, че избраната по-рано схема за кодиране позволява създаване на коректен низ 111 10 11, който означава, че няма ограничение на атрибута *Игра_на_тенис* в пост-условието на правилото, т.е. създава некоректна хипотеза. Ако искаме да избегнем разглеждането на подобни хипотези, можем да използваме друг начин на кодиране (например, да назначим само един бит за пост-условието *Игра_на_тенис*, за да указваме дали неговата стойност е *да* или *не)*, да променим генетичните оператори по такъв начин, че те явно да избягват създаване на подобни битови низове, или просто на присвоим на тях една много низка стойност на функцията на годност.

6.4. Генетични оператори

Генерацията на наследниците в ГА се определя от генетичните оператори, които рекомбинират и мутират избраните членове на текущата популация. Типичните ГА оператори за манипулацията на хипотези, представени чрез битови низове, отговарят на идеализираната версия на генетичните операции, съществуващи в биологичната еволюция. Два най-често срещани оператори са *генетичен обмен* и *мутация*.

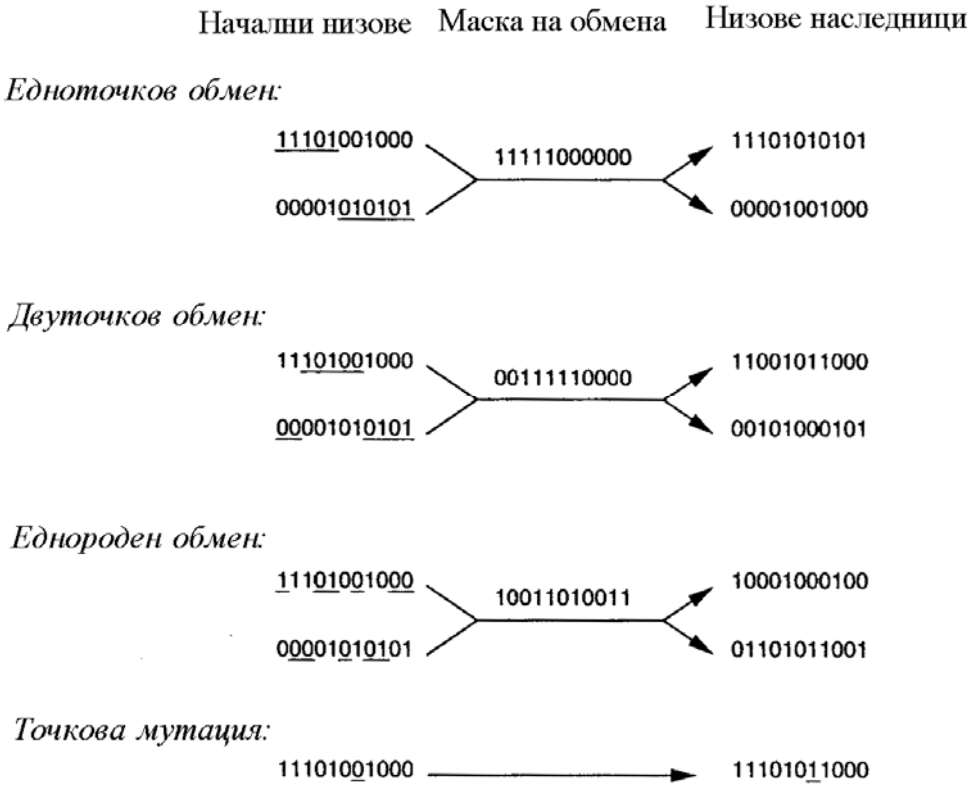

Таблица 6-2. Често използвани генетични оператори

Операторът за генетичния обмен поражда два нови наследника от два родителски низа чрез копиране на избраните битове от всеки родител. Битът в позицията *i* на всеки наследник е копието на бита в позицията *i* на един от двамата родителя. Изборът, кой родител ще даде своя бит от позицията *i*, се определя от един допълнителен низ, наречен *маска на генетичния обмен.* За илюстрация, да разгледаме *едноточков оператор за генетичния обмен*, показан в горната част на Таблица 6-2. Да разгледаме най-горния от двама наследника. Този наследник взима първите пет бита от първия си родител, а останалите шест бита – от втория си родител, тъй като маската на генетичния обмен е 11111000000 описва тези избори за всяка битова позиция. Вторият наследник използва същата маска за генетичния обмен, обаче превключва ролята на двамата си родители. По този начин той съдържа битове, които не са били използвани от първия наследник. В едноточковия оператор за генетичния обмен маската на генетичния обмен винаги се конструира по такъв начин, че се започва с низа, съдържащ *n* непрекъснати 1, след които следват необходимия за да довърши низ брой нули. Това води до създаването на наследници, в които първите *n* бита са принос на един от родители, а останалите битове – на другия. При всяко прилагане на едноточковия оператор за генетичния обмен точката на обмена *n* се избира по случайния начин, след което се създава и се прилага маската на генетичния обмен.

При *двуточков оператор за генетичния обмен* наследниците се създават чрез замяна на междинните сегменти на един от родителите със средните сегменти от другия родител. С други думи, маската на генетичния обмен представлява низ, който започва с n_0 нули, след които следват n_1 единици и се завършва

отново с необходимия за завършване на низа брой нули. При всяко прилагане на двуточковия оператор за генетичния обмен маската се генерира чрез случаен избор на целите числа n_0 и n_1. Например, в примера, показан в Таблицата 6.2, наследниците се създават чрез използване на маската, в която $n_0 = 2$, а $n_1 = 5$. Отново, двата наследника се създават чрез превключване на ролите, които играят и двата родителя.

Операторът за еднороден генетичен обмен комбинира битове, разпределени еднородно и от двата родителя, как това показано в Таблица 6.2. В този случай маската на генетичния обмен се генерира като случаен битов низ, като всеки бит се избира по случаен начин не зависимо от други битове.

В отличие от рекомбинативни оператори, които създават новото поколение чрез комбиниране на части от двама родителя, вторият тип генетични оператори произвежда наследник само от един родител. Операторът за *мутация* създава малки случайни промени в битовия низ чрез избиране по случаен начин на един единствен бит и промяна на неговата стойност. Мутацията често се изпълнява след прилагането на генетичния обмен, както е показано в прототипичния ГА алгоритъм от Таблица 6-1.

Някои ГА системи използват допълнителни оператори, особено оператори, които са приспособени към определен начин на представяне на хипотези, използван в системата. Например, Grefenstette et al. (1991) описва системата, която научава множество от правила за управление на робота. Тя използва мутацията и генетичния обмен заедно с оператор за специализацията на правила. Janikow (1993) описва системата, която научава правила чрез използване на оператори за специализация и обобщение на правила чрез няколко управлявани начина (например, чрез явната замяна на ограничение върху атрибута с “не ни интересува”).

6.5. Функция за годност и избор

Функцията за годност определя критерий за подредба (ранжиране) на потенциални хипотези и техен вероятностен избор за включване в следващата популация. Ако задачата е да се научат класификационни правила, то функцията за годност обикновено съдържа компонент, който изчислява класификационната точност на правилото върху множество от зададени обучаващи примери. Често се включват и други критерии, като, например, сложността или общността на правилото. В общия случай, когато битовото кодирана хипотеза се интерпретира като някоя сложна процедура (например, когато битовият низ представлява колекцията от АКО-ТО правила, които ще се променят заедно, за да управляват роботизираното устройство) функцията за годност може да измерва не поведението на отделни правила, а цялостното поведение на резултиращата процедура.

В прототипичния ГА алгоритъм, показан в Таблица 6.1, вероятността, че някоя хипотеза ще бъде избрана, се задава като отношение на нейната годност към годността на други членове на текущата популация, както се вижда от ф-ла (6.1). Този метод понякога се нарича *избор, пропорционален на годността*, или

избор чрез колелото на рулетката. Предложени са и други методи за използване на годност за избор на хипотези. Например, при така наречения *турнирен избор* отначало се избират по случаен начин две хипотези от текущата популация. След това, с определена предварително зададена вероятност p от тях се избира тази с по-голяма годност, а с вероятност $(1 - p)$ – тази с по-малка годност. Турнирният избор често дава по-разнообразна популация от колко при избора, пропорционален на годността. При друг метод, наречен *ранжирана селекция*, всички хипотези от текущата популация отначало се сортират съгласно своята годност. След това вероятността за избор на една хипотеза е пропорционална не на нейната годност, а на нейния ранг (позиция) в този сортиран списък.

6.6. Пример

Генетичният алгоритъм може да се разглежда като един общ метод за оптимизация, който претърсва голямото пространство на кандидат-обекти, опитвайки се да намери този, който има най-доброто поведение съгласно зададената функция на годност. Макар че не гарантират намирането на оптималния обект, генетичните алгоритми често успяват да намерят обект с висока степен на годност. ГА са били използвани за различни оптимизационни задачи извън обсега на машинното самообучение, като, например, проектиране на печатни платки и създаване на разписания. В МС те са били прилагани към задачите за апроксимацията на функции за избор на топологията на невронни мрежи, както и за избор на атрибути и прототипи за алгоритми за самообучение чрез запомняне.

За илюстрацията на използването на ГА за научаване на понятия, ще разгледаме накратко системата GABIL, описана от DeJong et al. (1993). GABIL използва един ГА за научаване на булевите понятия, представени чрез дизюнктивното множество от пропозиционални правила. GABIL използва точно този алгоритъм, който е описан в Таблица 6.1. В описаните експерименти параметър r, който определя част от родителската популация, която ще бъде заместена чрез генетичния обмен, е бил избран равен на 0.6. Параметърът m, определящ скоростта на мутацията, е бил избран равен на 0.001. Това са типични стойности на тези параметри. Размерът на популацията p варира от 100 до 1000 в зависимост от конкретна задача.

И така, ГА, използван в GABIL, се характеризира със следното:

Представяне. Всяка хипотеза съответства на едно дизюнктивно множество от пропозиционални правила, кодирани по начина, описан в раздела 6.3. В частност, пространството от хипотези се състои от правила, предусловията на които са представени чрез конюнкцията от ограничения върху фиксираното множество от атрибути, както това е описано в предишните раздели. За представяне на множеството от правила се използва конкатенацията на битови низове, представящи отделните правила. Например, да разгледаме пространството на хипотези, в което предусловията на правила са конюнкцията от ограничения върху два булеви атрибута a_1 и a_2. Пост-условието на правило се

описва чрез един бит, който показва предсказваната стойност на целевия атрибут *c*. По този начин, хипотезата, състояща от два правила:

$$АКО\ a_1 = ИСТИНА \wedge a_2 = ЛЪЖА\ \ ТО\ c = ИСТИНА;\ \text{АКО}\ a_2 = ИСТИНА\ \ ТО\ c = ЛЪЖА$$

ще се представи чрез следващия низ:

a_1	a_2	c	a_1	a_2	c
10	*01*	*1*	*11*	*10*	*0*

Обърнете внимание, че дължината на битовия низ нараства с броя на правила в хипотезата. Тази променлива дължина на низа изисква малка модификация на оператора за генетичния обмен, която е описана по-долу.

Генетични оператори. GABIL използва стандартен оператор за мутация, описан в Таблицата 6.2, който избира по случаен начин един единствен бит и го заменя с неговата противоположност. Операторът за генетичния обмен е едно разширение на стандартен двуточков оператор за обмен, описан в Таблица 6-2. В частност, за да може да работи с низове с променлива дължина, кодиращи множества от правила, както и за да ограничи системата по такъв начин, че генетичният обмен да се извършва само между подобни по характер части от низове, кодиращи правила, е бил използван следният подход. За да изпълни операцията на генетичния обмен върху двама родители, двете точки на обмена отначало се избират по случаен начин върху низа, представящ първия родител. Нека d_1 (d_2) е разстоянието от най-лявата (най-дясната) от тези точки на обмена до границата на правило, намираща се веднага вляво. Точките на обмена във втория родител пак се избират по случаен начин, но върху тях се налага ограничение, че те трябва да имат същите стойности на d_1 и d_2. Например, ако родителските низове са:

	a_1	a_2	c	a_1	a_2	c
h_1:	*10*	*01*	*1*	*11*	*10*	*0*

и

	a_1	a_2	c	a_1	a_2	c
h_2:	*01*	*11*	*0*	*10*	*01*	*0*

а точките на обмена, избрани за първия родител, са точките, следващи след битовите позиции 1 и 8:

	a_1	a_2	c	a_1	a_2	c
h_1:	*1[0*	*01*	*1*	*11*	*1]0*	*0*

където "[" и "]" означават точките на обмена, то $d_1 = 1$ и $d_2 = 3$. Следователно, разрешените двойки от точките на обмена за втория родител включват двойките от битови позиции <1, 3>, <1, 8> и <6, 8>. Ако така се случи, че бъде избрана двойка <1, 3>

	a_1	a_2	c	a_1	a_2	c
h_2:	*0[1*	*1]1*	*0*	*10*	*01*	*0*

то новите наследници ще бъдат:

	a_1	a_2	c
h_3:	11	10	0

и

	a_1	a_2	c	a_1	a_2	c	a_1	a_2	c
h_4:	00	01	1	11	11	0	10	01	0

Както показва примерът, операторът за генетичния обмен позволява наследниците да съдържат брой на правила, различен от този на техните родители, осигурявайки обаче, че всички битови низове, генерирани по този начин, представляват добре дефинирани множества от правила.

Функция за годност. Годността на всяко хипотетично множество от правила се базира на неговата класификационна точност върху обучаващите данни. В частност, функцията, използвана за измерване на годност, е

$$Годност(h) = (точност(h))^2$$

където *точност(h)* е процентът на всички обучаващи примери, коректно класифицирани от хипотеза *h*.

В експериментите върху различни задачи за научаване на понятия GABIL показа, че е сравнима по класификационната точност с такива МС алгоритми, като класификационни дървета, C4.5 и системата за научаване на покриващи правила AQ14. Върху множество от 12 бази от синтетични данни GABIL показа средната точност от 92.1%, докато поведението на останалите системи варира от 91.2% до 96.6%.

6.6.1. Разширения

Базовият алгоритъм на GABIL е бил използван и с различни разширения. При едно от тях са били добавени два нови генетични оператора, мотивирани от често използвани в МС оператори за обобщение. Първият от тях *ДобавиАлтернатива*, обобщава ограничението върху указания атрибут чрез замяна на 0 с 1 в под-низа, съответстващ на този атрибут. Например, ако ограничението на атрибута е представено от низа 10010, то този оператор може да го промени на 10110. Операторът е прилаган с вероятност 0.01 към избраните членове на популацията за всяка генерация. Вторият оператор *ПропусниУсловие* изпълнява по-радикална обобщаваща стъпка чрез заместване на всички битове на някой конкретен атрибут с 1. Този оператор съответства на обобщаване на правилото чрез пълно отстраняване на ограничение върху указания атрибут. Той е бил прилаган към всяка генерация с вероятност 0.60. Авторите съобщават, че така ревизирана система постига средната класификационна точност върху 12 синтетични бази от данни, равна на 95.2% срещу 92.1% на базовия ГА алгоритъм.

6.7. Търсене в пространство на хипотези

ГА използва рандомизиран (случаен) метод за ”лъчево” търсене, за да намери хипотеза с максимална годност. Това търсене е доста по-различно от други методи за търсене, разгледани в този курс. Ако сравняваме начина на претърсването на пространството от хипотези при ГА и при невронни мрежи с BACKPROPAGATION алгоритъм, например, то методът на градиентното спускане, използван при невронни мрежи, гладко се предвижва от една хипотеза към друга, която е доста сходна. При ГА търсенето може да се предвижва много по-рязко, заместваики някоя родителска хипотеза с наследници, които могат да бъдат радикално различни от родителите. По тази причина ГА по-малко вероятно да попаднат в локален минимум, характерен за методи на градиентното търсене.

Едно практическо затруднение при прилагането на ГА е проблемът с *натрупването (crowding)*. Натрупването е феномен, при който някой индивид, който има по-голяма годност от другите индивиди в популацията, бързо се репродуцира, така че копията на този индивид, както и много сходни с него индивиди, заемат една голяма част от популацията. Отрицателното влияние на натрупването се състои в това, че то намалява разнообразието на популацията, забавяйки по този начин прогреса на ГА. За намаляването на натрупването се използват няколко стратегии. Един от подходите е да бъде променена функцията за избор, използвайки критерий от типа на турнирния избор или на ранжираната селекция вместо метода за избор, пропорционален на годността. Другата стратегия е “споделената годност”, при който измерваната годност на индивида се намалява от присъствие на други, подобни индивиди в популацията. Третият подход е да се ограничи видове индивиди, на които е позволено да рекомбинират, за да направят новото поколение. Например, чрез разрешение да рекомбинират само най-сходните индивиди, можем да насърчим формирането на клъстери (групи) от сходни индивиди или “подвиди” в една популация. Подобният подход се състои в пространственото разпределяне на индивиди и разрешение да рекомбинират само на индивиди, намиращи се наблизо. Повечето от тези техники са провокирани от аналогията с биологичната еволюция.

6.8. Модели на еволюция и самообучение

В множество естествени системи в процеса на своето съществуване отделните организми се учат доста интензивно да се адаптират към окръжаваща ги среда. В същото време биологичните и социалните процеси позволяват на своите видове да се адаптират в периода на няколко поколения. Един от въпроси, свързани с еволюционните системи, е “Каква е връзката между самообучение по време на живота на отделен индивид и самообучение в течение на значително по-голям период от време на ниво видове, осъществявани в процеса на еволюцията?”

6.8.1. Еволюцията по Ламарк

В края на 19 в. френски естествоизпитател Жан-Батист дьо Моне, Шевалие дьо ла Марк (често наричан просто Ламарк) е предложил теория, че еволюцията в

течение на няколко поколения директно се влияе от опита на отделни организми, натрупан в течение на техния живот. В частност, той е предположил, че опитът на един отделен организъм директно влияе върху генетичната природа на неговото поколение. Например, ако някой индивид се е научил по време на своето съществуване да избягва отровната храна, то той може генетично да предаде тази черта на своето поколение, което по тази причина няма да има нужда да научава тази черта от собствения опит. Това е доста атрактивно предположение, тъй като потенциално би позволило един по-ефективен еволюционен процес, отколко баналния "породи и провери" (като този, изпълняван в ГА), който игнорира опита, натрупан през живота на отделния индивид. Обаче, съществуващите в момента научните свидетелства съкрушително разбиват модела на Ламарк. Актуалното сега мнение е, че генетичната природа на индивида фактически не се влияе от опита на неговите биологични родители. Не зависимо от този биологичен факт, последните компютърни изследвания показват, че моделът на Ламарк може понякога да подобри ефективността на компютърните генетични алгоритми.

6.8.2. Ефектът на Болдуин

Макар че моделът на Ламарк не се приема като модел на биологичната еволюция, са били предложени други механизми, в които индивидуалното самообучение може да променя пътя на еволюцията. Един такъв механизъм е наречен "ефектът на Болдуин" в чест на американския психолог Джеймс Марк Болдуин (1896), който пръв предложил тази идея. Ефектът на Болдуин се базира на следните наблюдения:

- Ако някой биологичен вид се развива в променяща се среда, съществува еволюционният натиск, фаворизиращ индивиди, способни да се самообучават по време на техния живот. Например, ако в средата се появява някой нов хищник, индивидите, които могат да се научат да избягват този хищник, ще бъдат по-успешни от тези, които не могат да се научат на това. Като резултат, способността да се учи позволява на индивида да изпълнява малко локално търсене в течение на неговия живот, за да максимизира неговата годност. От другата страна, индивидите, които не са способни на самообучение и чиято годност напълно се определя от тяхната генетична природа, ще се намират в относително неизгодно положение.
- Тези индивиди, които са способни да научат много полезни характеристики, ще се осланят по-малко на своят генетичен код, за да закрепят тези характеристики. Като резултат, такива индивиди могат да поддържат по-разнообразен генетичен набор, разчитайки на способността на индивида към самообучение, за да преодолеят "липсващи" или "не достатъчно оптимални" характеристики в своя генетичен код. Този по-разнообразен запас от гени ще поддържа по-бърза еволюционна адаптация. Следователно, способността на индивида да се самообучава може да има непряк ускоряващ ефект върху скоростта на еволюционната адаптация на цялата популация.

И така, ефектът на Болдуин предоставя един непряк механизъм на това, как самообучението на индивида позитивно влияе на скоростта на еволюционния прогрес. Чрез увеличаване на способността към оцеляване и на генетичното разнообразие на вида, самообучението на индивида поддържа по-бърз

еволюционен прогрес, увеличавайки по този начин шанса, че видът ще развие генетични, не придобити чрез самообучение характеристики (например инстинктивен страх от хищника – в горния пример), които ще му позволят по-добре да се нагласи към окръжаваща го среда.

Има няколко опита да разработят изчислителните модели за изучаване на ефекта на Болдуин. Например, Hinton и Nowlan (1987) експериментирали с развитие на една популация от прости невронни мрежи, в които някои тегла са били фиксирани при обучение на отделни мрежи ("живот на мрежа-индивид"), докато другите бяха настройвани чрез обучение. Генетичната природа на индивида определяше, кои тегла могат да се научат, а кои – да бъдат фиксирани. В случаите, когато не беше разрешено никакво индивидуално самообучение, популацията не успяваше да подобри своята годност с течение на времето. Обаче, когато индивидуалното самообучение е било разрешено, популацията бързо подобрявала своята годност. При по-ранни поколения на еволюцията, популацията съдържаше по-голямата пропорция от индивиди с множество научаеми тегла. Обаче, с напредването на еволюцията броят на фиксирани, коректни тегла на мрежите започна да се увеличава, така че популацията се развиваше в посока увеличаване на генетично придобити тегла и по-малка зависимост от индивидуално научени тегла. Един хубав обзор на изследванията в това направление може да се намери в (Mitchell 1996).