

Случайная и детерминированная компоненты в эволюционном моделировании

Е.В. Балдин

Эволюция живых организмов, как известно, определяется сочетанием случайных и неслучайных факторов, поэтому естественной оказывается попытка выявить соотношение случайных и детерминированных компонент в алгоритмах, моделирующих биологические процессы. Методы поиска в ограниченном символьном пространстве, основанные на эволюционном преобразовании объектов, получили название генетических алгоритмов (ГА).

В теории ГА используются понятия, заимствованные из генетики и теории эволюции. Потенциальные решения кодируются в виде *хромосом* (чаще всего это строки двоичных символов, причем отдельные символы или их группы в таком случае называются *генами*). Из хромосом случайным образом формируется начальная *популяция* (массив строк), над элементами которой производятся операции *селекции* (отбор строк с наилучшими значениями целевой функции), *репродукции* (копирование хромосом в соответствии со значениями целевой функции на них), *кроссинговера* (обмен подстроками) и *мутации* (случайное изменение значения одного или нескольких генов на противоположное). Примененные последовательно, эти операторы образуют *генерационный цикл*. В результате его создается новая популяция, служащая основанием для нового цикла. Этому описанию соответствует следующее формальное представление генетического алгоритма [9]:

$$\text{ГА} = (P^0, \lambda, l, s, p, f, t), \quad (1)$$

где $P^0 = (a_1^0, \dots, a_\lambda^0)$ – исходная популяция, a_i^0 – решение задачи, представленное в виде хромосомы; λ – целое число (размер популяции); l – целое число (длина каждой хромосомы популяции); s – оператор отбора; p – отображение, определяющее рекомбинацию (кроссинговер, мутация); f – функция оптимальности (целевая функция); t – критерий останова.

Подробно механизм работы ГА описан в [11, 12]. Новый подход к исследованию эффективности генетических алгоритмов предложен в [1]. Данный подход основан на выводе следствий из предположения о том, что в результате работы генетического алгоритма найдено оптимальное решение некоторой задачи. Логическая схема анализа эффективности следующая.

Сначала устанавливается, что возможность вычисления значений целевой функции f в отдельных точках, обеспечивающая работу оператора селекции s , сама по себе не достаточна для получения универсального критерия останова t . Для нахождения t требуется дополнительное специальное исследование свойств f , в результате чего исходная задача сводится к поиску координат глобального экстремума при известном его значении [1, с. 22–23].

Подобное сведение, как нетрудно заметить, возможно лишь при условии, что имеющихся свойств функции f достаточно для идентификации найденного значения как глобального экстремума *без обращения к полному перебору* пространства допустимых решений, иначе вообще нет смысла прибегать к помощи ГА. Это, выглядящее, на первый взгляд, малосущественным, условие, в действительности предъявляет довольно жесткие требования к «качеству» информации о свойствах f . В задаче о коммивояжере, например, все известные на сегодняшний день способы нахождения кратчайшего пути содержат в скрытом виде полный перебор всех возможных вариантов, что под силу только детерминированным алгоритмам. При решении задачи методом Монте-Карло происходит осознанный отказ от контроля за всем пространством допустимых решений, что исключает в результате гарантию нахождения не приближенного (аппроксимирующего в терминологии Беллмана), а именно точного решения. Но те же самые соображения применимы и к ГА, что приводит к довольно неожиданному выводу: при нынешнем состоянии знаний относитель-

но методов решения задачи коммивояжера нахождение глобального экстремума (вместе с обоснованием) на основе идей эволюционного моделирования невозможно.

Преимущество ГА над чисто вероятностными алгоритмами может проявиться, далее, только в способности за счет использования мутации и кроссинговера собрать на заключительном шаге своей работы точку искомого глобального экстремума, «склеив» подстроки ранее найденных хромосом со значениями целевой функции f , близкими к оптимальному. Более того, возможность подобной «сборки» должна систематически выполняться и на этапах, предшествующих получению глобального экстремума [1, с. 27–28]. (Этого нельзя утверждать относительно *всех* этапов работы ГА, так как случайный характер генетических операторов не позволяет утверждать ничего определенного относительно конкретных шагов алгоритма. Наличие в популяции большого числа хромосом, содержащих фрагменты оптимального решения, должно быть подготовлено многократным применением оператора селекции).

Подобное эффективное применение рекомбинации p было бы невозможно при чисто случайном выборе способа кодирования значений аргументов целевой функции двоичными символами. Именно кодирование отвечает за возможность нетривиального взаимодействия генетических операторов s и p и функции оптимальности f в процессе работы алгоритма, что, по замыслу ГА, и должно привести к качественному скачку в его эффективности.

Из приведенных соображений и вытекает основной вывод работы [1]: уже на стадии выбора формальной модели (1) ГА требуется предусмотреть возможность согласованного взаимодействия кодирования и операторов рекомбинации, а это непосредственным образом подрывает идею универсального характера применения ГА.

Отмеченные на основе теоретического анализа в работе [1] недостатки ГА с точки зрения эффективности их применения были эмпирическим путем обнаружены рядом исследователей, применяющих методы ГА в различных предметных областях. Ими были предложены модификации стандартного ГА (1) с целью совершенствования его эффективности. Анализ этих работ с позиций общих выводов работы [1] и составляет цель настоящей работы.

ГА хорошо себя зарекомендовали при решении ряда комбинаторных задач, что послужило основанием для предположения об их универсальном характере [5]. Для оценки степени правдоподобности данной гипотезы следует, однако, учитывать, что эффективное по сравнению с другими алгоритмами поиска функционирование ГА возможно лишь при одновременном выполнении сразу нескольких условий [1, 9]:

- каждая хромосома в популяции декодируется для последующей оценки, после чего для нее определяется значение целевой функции (ЦФ). При задании целевой функции должна быть предусмотрена возможность сравнения хромосом по степени их близости к оптимальному решению, иначе процесс поиска решения превращается в слепой перебор;
- так как с практической точки зрения интерес представляют только те хромосомы, которые являются допустимыми закодированными решениями, то на каждом этапе работы алгоритма должна контролироваться «легальность» хромосом на предмет удовлетворения существующих ограничений. Для этого может быть использован один из следующих методов:
 - разработка генетических операторов таким образом, чтобы производимые с их помощью новые хромосомы удовлетворяли налагаемым условиям;
 - применение к «нелегальным» хромосомам штрафных функций для наказания в соответствии со степенью нарушения ограничений;
 - удаление недопустимых хромосом из популяции;
- в целях обеспечения «обмена информацией» между хромосомами механизм их преобразования не может не быть согласован со способом кодирования решений. Способ кодирования и упорядочивания битов внутри строки может направить алгоритм по ложному пути, вызывая сходимость к локальным оптимумам. Так, например, в задаче о нахождении максимума монотонно возрастающей целочисленной функции на отрезке $[0; 8]$ любой обмен информацией (без перестановки битов) между близкими к оптимальному решениями 0110 (6) и 0111 (7) не

приведет к получению глобального решения 1000 (8). Для преодоления данного затруднения могут быть предложены два варианта рекомбинации: (1) использование предварительной информации для правильного упорядочивания битовых комбинаций внутри строки (сцепление битов); (2) использование оператора инверсии (перестановки битов подстроки в обратном порядке). Однако по своему механизму инверсия схожа с мутацией, в общем случае она является случайным оператором и для придания ей целенаправленности придется опять же использовать информацию о задаче. Например, в предыдущем примере инверсия 2-го, 3-го и 4-го битов хромосомы 0100 (6) приведет к получению хромосомы 0001 (1), то есть худшего решения.

В свете вышесказанного нельзя исключить, что попытка применения ГА в качестве универсального метода ИИ может привести к затруднениям, связанным со специфическими особенностями самого класса ГА.

За неимением общего способа кодирования хромосом и выбора целевой функции, который позволял бы надеяться на нечто более эффективное, нежели чисто случайный «перебор», для каждой конкретной задачи приходится специально строить ее формальную модель с так или иначе заданными параметрами генерационного цикла. Последнее обстоятельство ставит под сомнение обоснованность претензий генетических алгоритмов как класса на роль универсального метода решения задач (можно говорить лишь об универсальности *общего метода* поиска, основанного на идеях эволюции решений).

В ряде недавно появившихся «экспериментальных» работ [2, 3, 4, 10] содержится утверждение о том, что «классическая» модель ГА не позволяет эффективно решать большинство практических задач. Если в более ранних работах отечественных авторов [6, 7], вышедших в 1993–97 гг., подходящей считалась любая допустимая для ГА формализация естественной задачи, то в последних трудах проблеме «разумного» кодирования уделено значительно большее внимание. С известными оговорками это можно назвать «сменой парадигм» в методологии эволюционного моделирования: при анализе области применения ГА теперь используются более осторожные оценки, осо-

бо отмечается необходимость применения содержательных знаний о задаче при кодировании, задании генетических операторов (кроссинговера, мутации, репродукции), а также при определении целевой функции и критерия останова [10]. Последнее означает, что отсутствие содержательной информации о задаче затрудняет целенаправленную работу алгоритма. Возможный компромисс заключается в том, чтобы рассматривать ГА уже не как алгоритмы поиска глобального решения, а лишь как метод отыскания «оптимально-компромиссных» [4] решений. Тем не менее, даже при подобном ослаблении вопрос о практической эффективности ГА остается открытым.

Многие исследователи, на практике столкнувшиеся с подобными проблемами классических ГА, предлагают свои модификации традиционной модели (или ее отдельных компонентов) с целью преодоления выявленных ограничений. Большинство таких методов при этом в той или иной форме используются и в традиционных алгоритмах решения комбинаторных задач. Пытаясь преодолеть ограничения, связанные с неэффективностью «традиционных» алгоритмов, исследователи ГА в конечном счете зачастую пользуются их же приемами – например методом выделения клика в графе для задачи о его оптимальной раскраске или методом разбиения задачи на подзадачи. Ниже будут рассмотрены модификации классической модели ГА (на наш взгляд, наиболее показательные), разработанные в последние годы. Важно выяснить, позволяют ли предложенные модификации разрешить указанное противоречие между универсальностью и эффективностью ГА или же, наоборот, оно является принципиально неустранимым в рамках данного класса алгоритмов.

Когда в [3, 4, 5] говорится о необходимости учета специфики задачи при выборе способа кодирования, то при этом имеется в виду возможность посредством содержательной интерпретации элементов хромосом придания работе оператора кроссинговера по обмену информацией между строками целенаправленного характера. С другой стороны, как показано в [1], детальная формализованная информация о задаче делает зачастую ненужным использование идей ГА. Способность алгоритма осуществлять целенаправленный поиск означает, что особенности решаемой задачи были уже учтены на

этапе кодирования решений и, возможно, еще раньше при определении целевой функции. При использовании информации о задаче ГА зачастую вырождается в некоторый специализированный численный метод, сохраняющий внешние атрибуты ГА, однако по сути стоящий ближе к детерминированному алгоритму.

Способ использования содержательного знания о задаче является в некотором роде центральной проблемой для этого класса алгоритмов, разбиваясь, в свою очередь, на ряд подзадач со своими методами решения. Безусловно, существуют задачи, «естественная» (наиболее очевидная) формализация которых является одновременно и приемлемой для ГА. В качестве примера рассмотрим уже описанную задачу о нахождении максимума монотонно возрастающей целочисленной функции на отрезке $[0; 8]$. Если хромосомы представляют собой двоичные представления целых чисел из области определения, то два из трех перечисленных выше условий выполняются, то есть

- каждая хромосома может быть оценена по степени близости к решению;
- некорректных хромосом не существует и нет необходимости контролировать их «легальность» на каждом этапе, поскольку любая комбинация из 4 бит является допустимым решением из области определения ЦФ.

Вместе с тем, без знания свойств функции во всей области определения нельзя утверждать, что механизм преобразования хромосом согласован со способом их кодирования. В данном случае исходя из предположения о том, что кроссинговер действует целенаправленно, из факта близости хромосом 0110 (6) и 0111 (7) к решению (8) можно сделать ошибочный вывод о том, что и их общая подстрока 011 является подстрокой точки глобального максимума 1000 (8) (то есть глобальное решение представимо в виде шаблона 011*, где * – любой двоичный символ).

Следовательно, успех целенаправленного поиска ГА будет гарантирован только при условии «прозрачной» интерпретации каждого этапа решения, начиная с формализации и выбора способа кодирования задачи и вплоть до остановки алгоритма (либо же поиск реше-

ния средствами ГА на практике вынужденно опирается на неполную информацию о специфике задачи).

С этим аспектом работы ГА тесно связана проблема ограничения действия случайных операторов (мутации и кроссинговера, а также механизма формирования начальной популяции). Кроме того, существует тенденция использования специальной информации о задаче и в других, детерминированных, процедурах (кодирование хромосом, определение критерия останова алгоритма) с тем, чтобы по возможности предотвратить потенциальный разрушительный эффект от работы случайных операторов. С другой стороны, для использования в полной мере предполагаемых преимуществ ГА необходимо уметь эффективно применять эти же случайные операторы для выхода из локальных экстремумов (предотвращение так называемой преждевременной сходимости алгоритма).

Для управления работой ГА в первую очередь используются средства, ограничивающие область поиска на основании содержательных знаний о задаче и одновременно изолирующие случайные операторы ГА на тех этапах, когда они могут увести алгоритм в сторону от глобального решения. К таким методам относится и разработка подходящей системы кодирования. Например, в [10] рассматривается несколько вариантов возможного кодирования хромосом для алгоритма распределения трассируемых соединений СБИС (цепей) по слоям. Один из них – задание хромосом в виде списков натуральных чисел длиной W , где W – количество слоев, необходимых для распределения трассируемых соединений. Каждому элементу списка соответствует количество цепей, расположенных в данном слое, при этом номера этих цепей хранятся отдельно и обращение к ним осуществляется по мере надобности. При таком способе кодирования затруднены отбор «нелегальных» хромосом (для анализа их корректности придется каждый раз обращаться к массиву номеров цепей, чтобы определить их совместимость) и рекомбинация (кроссинговер не может изменить порядок цепей в рамках одного гена или их количество – то есть значение самого гена). Поэтому автор предлагает другой способ кодирования, основанный на представлении хромосом в виде списков, длина которого равна количеству распределяемых по слоям соединений (N). Значение элемента списка соответствует

номеру цепи, каждой цепи графа присваивается уникальный номер. Важным моментом здесь является то, что на основании исходных данных создается массив ограничений, который определяет, какие цепи могут располагаться в одном слое. Один из вариантов декодирования хромосом предполагает сверку каждого декодируемого гена с этим массивом. Если цепи могут находиться в одном слое, то новая цепь добавляется в имеющийся слой, если нет, то слой объявляется заполненным и соединения определяется в следующий. Таким образом, в данном случае контроль легальности не осуществляется на каждом этапе работы ГА, но при этом требует дополнительных ресурсов (память, работа алгоритма по описанию ограничений и обращение к массиву). Кроме того, такой подход зачастую делает бесполезными мутацию и кроссинговер: распределение цепей по слоям производится фактически только на основании их совместимости, значение ЦФ играет второстепенную роль.

Несколько другой подход к кодированию предлагается в [3]. В указанной работе ГА используются для решения задачи о размещении элементов СБИС. В качестве хромосом выступают конкретные закодированные решения, в которых элементы СБИС представлены генами. При этом предпочтительно было бы сохранить сильные связи между элементами СБИС (то есть связи, характеризующиеся относительно высоким количеством цепей, соединяющих некоторую пару элементов e_i, e_j). В противном случае ГА в силу случайного характера мутации и кроссинговера может разместить эти элементы в удаленных друг от друга позициях. Автор предлагает объединять гены в изначально оптимальные подстроки (кластеры, совокупности элементов) на этапе кодирования хромосом. Основанием для такой группировки является матрица связности, определяющая количество связей между элементами. Далее кластеры рассматриваются как отдельные гены вплоть до заключительного этапа работы алгоритма, когда допускается их разрушение, а также ограниченное использование случайных операторов для обмена информацией между ними.

До запуска ГА эвристический алгоритм формирует кластеры из общего пространства элементов. Как указывает автор, формирование слишком больших кластеров приведет к значительному сокращению размерности задачи, однако уменьшение степеней свободы по-

влечет ухудшение качества решения. Для качественной работы ГА целесообразно было бы, чтобы размер кластеров не превышал 2%–20% от общего числа размещенных элементов n , а сумма связностей элементов в кластерах не превышала 8%–20% от суммы связностей всех элементов.

Предлагается следующий алгоритм формирования кластеров. Сначала вычисляется общая сумма всех связей SE (на основании матрицы связностей). Затем определяется первый элемент первого кластера исходя из принципа максимальности следующей оценки:

$$\tau_{i_0 j_0} = \max_{e_i \in \bar{E}_k} \max_{e_j \in \bar{E}_k} c_{ij},$$

где $C = \|c_{ij}\|$ – матрица связностей, e_i, e_j – элементы, E_k – множество свободных, то есть не размещенных в кластеры, элементов (для первого элемента первого кластера это множество совпадает с множеством всех элементов).

Далее определяется следующий элемент кластера, максимально связанный с уже имеющимися, и так далее до тех пор, пока не будет достигнуто критическое значение размера кластера $\alpha * n$. Затем алгоритм приступает к созданию следующего кластера, завершая свою работу тогда, когда суммарная длина связей в кластерах превысит ранее определенную величину $\beta * SE$, где $\beta = 0,08 \div 0,2$.

В результате кластеризации изменяется не только кодирование хромосом, но и сама матрица связностей. Строки и столбцы, соответствующие элементам одного кластера, преобразуются в одну строку и один столбец, соответствующие значения в них суммируются. На последней стадии разработанного алгоритма (уже после завершения работы ГА) кластеры разрушаются, и полученное оптимальное решение улучшается с помощью других методов.

Легко видеть, что задача нахождения оптимального размера отдельного кластера $\alpha * n$ и общего их размера $\beta * SE$ (по сути отыскания компромисса между сокращением пространства поиска и качеством итогового размещения) сама по себе должна решаться при помощи дополнительного эвристического алгоритма (либо же значения этих величин определяются опытным путем). По утверждению же ведущих специалистов в области ГА [11, 12] основное преимуще-

ство этих алгоритмов состоит в синтезе «исследования» и «использования» пространства поиска. Под «исследованием» (*exploration*) в данном случае понимается процесс разработки новых ветвей поиска при помощи случайных операторов, а под «использованием» (*exploitation*) – направленный поиск, позволяющий находить оптимальные решения и отсеивать остальные. Сокращение пространства поиска при помощи внешних методов (эвристик формирования кластеров) по сути подрывает идею взаимодействия случайных и направленных операторов ГА, максимально усиливая детерминированную компоненту и ограничивая действие случайных факторов, которые могут сыграть и негативную роль, разрушая сильные связи между элементами. Кластеризация фактически оптимизирует один из параметров размещения, выполняя одну из задач ГА. Кроме того, на основании содержательных знаний о задаче (и в результате работы эвристического алгоритма) вполне детерминированным образом сокращается пространство поиска и размерность хромосом, что снижает трудоемкость всех операторов ГА.

Большое значение имеет и длина хромосом, задаваемая при кодировании. Как указывается в [2], увеличение длины кодировки зачастую ускоряет процесс сходимости всех членов популяции к глобальному решению, тогда как задача может состоять в исследовании области поиска или в отыскании нескольких равнозначных экстремумов. С другой стороны, длина хромосомы должна быть достаточно большой, чтобы обеспечить быстрый поиск. Для определения оптимальной длины в работе предлагается придерживаться компромиссного решения, учитывая специфику решаемой задачи.

Содержательная информация о задаче зачастую используется и при создании начальной популяции (в канонической модели ГА это осуществляется при помощи случайной процедуры). Так, в [3] указывается, что наилучшие условия для дальнейшей эволюции обеспечивает начальная популяция с достаточно высокими значениями ЦФ у хромосом и одновременно с разнообразным генофондом. Предлагается стратегия, которая реализуется случайно детерминированным способом с ориентацией на структуру решаемой задачи. Популяция формируется следующим образом. В качестве первого элемента выбирается максимально связанный со всеми, то есть имеющий макси-

мальную характеристику

$$t_i = \sum_j c_{ij},$$

где c_{ij} – элемент матрицы связностей.

Выбор очередного элемента основывается на определении «меры связности» каждого из оставшихся элементов с уже размещенными и поиске среди них такого элемента, для которого это значение максимально (под мерой связности понимают суммарную связность элемента с уже размещенными). Выбор позиции для очередного элемента производится согласно используемым критериям оптимальности размещения.

В целом эта процедура аналогична кластеризации. Заметим, что здесь, как и в предыдущем примере, для оценки связности элементов потребуется создание и обращение к матрице связности, а также задействуется дополнительный алгоритм формирования популяции со сложностью по времени $O(\lambda l)$, где l – длина хромосомы, λ – размер популяции.

Среди других приемов формирования начальной популяции используется, например, принцип максимального побитового разнообразия [11], обеспечивающий максимальное богатство генетического материала в начальной популяции.

Таким образом, можно говорить об использовании вспомогательных детерминированных алгоритмов, призванных облегчать работу генетического алгоритма на отдельных этапах решения (по сути, происходит разбиение общей задачи на подзадачи, соответствующие случайным и направленным операторам). Такая методика предлагается и в [4], но уже на более общем уровне. Используется сетка дискретизации, позволяющая сократить длину хромосом (и соответственно сузить область поиска). При этом шаг дискретизации (способ кодирования хромосом) уже не равен заданной точности поиска, как это делается в традиционных ГА. Пространство параметров разбивается на множество равных «гиперкубиков» с центрами в узлах сетки. Вычисления целевой функции (необходимой для оценки потенциальных решений) предлагается производить не в узлах сетки дискретизации, а в произвольной случайно выбранной точке «ги-

перкубика». В памяти вместе с хромосомой (генотипом) хранится и фенотип (вектор из пространства параметров, определяющий потенциальное решение из окрестности «узла»). Таким образом, разделяются детерминированная составляющая (выбор узла и перемещение в новые области поиска согласно заданным правилам) и случайная (случайный выбор элемента внутри окрестности). Очевидно, что при этом мутация сможет работать только на ограниченном интервале, что лишает ее возможности направлять алгоритм в другие области поиска в случае попадания в локальные экстремумы.

Рассмотрим сначала, повышает ли этот прием быстродействие ГА. В первую очередь следует отметить, что, разумеется, никакой экономии памяти не достигается, поскольку в рассматриваемом варианте ГА информация обо всех решениях хранится в виде векторов параметров. На основании данных о «гиперкубике» и значении целевой функции, полученном в некоторой случайно выбранной в нем точке, нельзя выяснить реальную близость «гиперкубика» к глобальному решению (то есть среднее значение целевой функции на нем, либо наличие внутри него точки экстремума). Это может достигаться только путем исчерпывающего оценивания параметров из гиперкубика (когда, например, алгоритм многократно возвращается к одному «гиперкубику», каждый раз выбирая новую случайную точку из него), что возвращает нас либо к полному перебору, либо к исходному кодированию без «узловых» решений. Существует и другой вариант – исследование целевой функции внутри гиперкубика (очевидно, что оценка поведения функции в окрестности некоторой точки по одному значению ее в этой точке возможна только при наличии у функции подходящих свойств). В данном случае ограничение пространства поиска (более грубая сетка дискретизации) не избавляет нас от необходимости исследования самих «гиперкубиков».

В той же работе предлагается и другой прием, позволяющий избежать потери «хороших» хромосом в результате действия мутации. Для этой цели создается «репродукционное множество», аккумулирующее в себе все потенциальные решения, включая как старые, так и вновь формируемые в процессе поиска. Оператор отбора применяется ко всему этому множеству, что дает возможность вернуть в популяцию потенциально выигрышные хромосомы.

Подобным образом осуществляется попытка реализовать одно из свойств естественного интеллекта, а именно – привлечение всего содержательного объема знаний на каждом этапе решения. Безусловно, для задач малого объема и при условии наличия мощных технических средств это вполне осуществимо, но с точки зрения эффективности все равно сводится к полному перебору. Для задач большой размерности, однако, данный экстенсивный метод теряет всякую привлекательность. Попытки избежать потери перспективных хромосом приводят к резкому росту объема используемой памяти и снижают быстродействие ГА по причине введения дополнительных операторов (добавление элементов в репродукционное множество, применение оператора отбора ко всему этому множеству и т.д.).

Большое внимание в последних работах уделяется «разумной» мутации (то есть мутации, основанной на знаниях и действующей относительно целенаправленно). Так, в [3] предлагается, в частности, использовать оператор мутации, основанный на методе релаксации и предназначенный для оптимизации суммарной длины соединений СБИС. Соединения между элементами уподобляются пружинам, притягивающим элементы друг к другу. Если было бы позволено перемещаться только одному элементу e , то та устойчивая позиция, которую он займет, есть целевая точка $t(e)$, в которой сумма воздействующих на него сил равно нулю. В соответствии с этим определяется силовой вектор $F_t(e)$, который имеет начальную точку в месте расположения элемента и задает направление до целевой точки с нулевым напряжением, а также расстояние до нее. Координаты целевой точки задаются следующими выражениями:

$$\bar{x}_e = \left[\frac{\sum_i c_{ei} x_i}{\sum_i c_{ei}} \right], \quad \bar{y}_e = \left[\frac{\sum_i c_{ei} y_i}{\sum_i c_{ei}} \right],$$

где $C = \|c_{ij}\|$ – матрица связностей, (x_i, y_i) – координаты мест установки тех элементов, с которыми соединен элемент e ; квадратные скобки означают округление до ближайшего целого.

Выбранный элемент e размещается в найденной позиции, если же она занята другим элементом, то два элемента меняются местами. Таким образом, в итоговом размещении общая длина связей минимизи-

зируется. Очевидно, что мутация такого рода перестает быть случайным оператором и действует вполне целенаправленно. При этом отпадает необходимость в подключении генетических операторов к решению отдельных подзадач общей задачи размещения элементов СБИС, поскольку алгоритм кластеризации, описанный в той же работе, и оператор мутации, примененные последовательно, оптимизируют два критерия размещения (вес связей и суммарную длину связей).

Одной из основных проблем, возникающих при кодировании, не опирающемся на специфику задачи, является деструктивная работа случайных операторов (в первую очередь мутации), на что указывается в [11, 12]. Когда мутация применяется в окрестности глобального решения, она способна разрушить имеющиеся в популяции перспективные хромосомы, уводя тем самым алгоритм в сторону (если, конечно, нет препятствий для выхода из этой окрестности). Кроме того, на начальной стадии решения задачи нам могут быть известны потенциально «хорошие» подстроки, которые целесообразно было бы сохранить. И все же при всех указанных недостатках мутация реализует одно из основных преимуществ ГА по сравнению с детерминированными алгоритмами: обновление генетического материала и предотвращение «застревания» алгоритма (преждевременной сходимости).

Таким образом, двойственный эффект мутации обуславливает модификации этого оператора. Как указывалось выше, предпочтительным является ограничение действия мутации на тех стадиях решения, когда она с большой вероятностью может разрушить потенциально «хорошие» хромосомы [3, 4]. Но очень часто возникают обратные ситуации, когда, например, действие алгоритма каким-либо образом ограничено окрестностью экстремума (то есть решение не может быть потеряно), а для нахождения более точного решения необходимо локальное изменение хромосом (пусть даже случайное). Для этого в алгоритм вводится управляемый оператор мутации, который применяется в специфических случаях.

Так, например, в [10] предлагается при попадании в локальный экстремум (если алгоритм выясняет, что более половины популяции состоит из одинаковых хромосом, что говорит о преждевременной

сходимости) осуществлять принудительный выход из цикла: популяция полностью уничтожается, вместо нее создается случайным образом новая, в которую из прежней переносится лучшее решение.

Подобный принцип, названный макромутацией, описан в [4]. Макромутация насильственно меняет генетический материал всех членов популяции (используя нормальное распределение), кроме нескольких лучших, после каждого 20-го поколения, не принесшего улучшения лучшей особи.

При применении подобных модифицированных операторов мутации возникает проблема, родственная проблеме выбора критерия остановки ГА: как для запуска макромутации, так и для остановки алгоритма необходимо привлекать дополнительные сведения о задаче. Макромутация либо остается случайным оператором (так как она уничтожает хромосомы, близкие к глобальному решению, если в популяции есть строки, соответствующие локальным экстремумам, но с более высокими значениями целевой функции), либо превращается в детерминированный оператор, который мы применяем осмысленно (в случае если мы можем отличить локальное решение от глобального на основании знаний о задаче). В последнем случае она фактически перестает быть генетическим оператором.

Существуют модификации и оператора кроссинговера, в первую очередь призванные придать его работе большую предсказуемость. В [3] предлагается использовать динамическую стратегию скрещивания, заключающуюся в комбинировании пяти форм образования пар для кроссинговера:

- панмиксия (случайное скрещивание);
- отрицательное ассортативное скрещивание;
- инбридинг («родственное» скрещивание);
- положительное ассортативное скрещивание;
- «лучший со всеми».

Панмиксия наиболее эффективна для придания разнообразия генотипу популяции, поэтому ее целесообразно применять на ранних этапах решения (как и отрицательное ассортативное скрещивание «лучших особей с худшими»). Инбридинг (скрещивание генетически

схожих особей) и положительное ассортативное скрещивание («лучшие с лучшими») применяются на более поздних этапах. На последнем этапе работы алгоритма предлагается производить кроссинговер между лучшей хромосомой и остальными попарно.

Очевидно, что для эффективного применения такой стратегии необходима информация о том, на каком этапе работы алгоритм находится в каждый конкретный момент (то есть о критерии остановки и поведении целевой функции).

Существует и другой подход к «дерандомизации» кроссинговера – без использования информации о решаемой задаче и стратегии работы кроссинговера на каждом этапе, но с некоторыми усредненными значениями случайных параметров, позволяющими достичь компромисса между эффективным поиском и исследованием пространства поиска. Например, в [2] предлагается не задействовать в работе алгоритма вероятность кроссинговера, а зафиксировать постоянное количество брачных пар хромосом для каждого поколения. При этом процесс работы кроссинговера становится более предсказуемым и управляемым, хотя и становится менее эффективным по сравнению с динамической стратегией скрещивания (так как более целесообразным было бы использование различного числа брачных пар в зависимости от близости к решению). Однако, следует отметить, что и в данном случае не удастся полностью отказаться от содержательной информации о задаче, так как такие фиксированные значения можно найти либо обладая информацией о поведении ЦФ, либо проведя достаточное число экспериментов.

В рассмотренных примерах ограничение действия случайных операторов производится практически на каждом этапе работы ГА. Более того, модифицируются и детерминированные операторы (выбор кодирования и другие) с тем, чтобы обеспечить максимальную предсказуемость работы случайных операторов в дальнейшем. Очевидно, что происходит это в силу того, что случайные операторы придают ГА универсальный характер, но снижают их эффективность при решении практических задач. Для придания ГА целенаправленности при решении задачи приходится использовать максимальные объемы содержательной информации о ней, при этом фактически сводя случайные операторы к детерминированным.

Таким образом, можно утверждать, что основным направлением работ при применении ГА для решения практических задач в настоящее время является усиление детерминированной компоненты (достигаемое путем локализации случайных операторов и использования формализованной информации о задаче) и последовательное разделение случайных и направленных операторов по этапам работы алгоритма, что подрывает идею их синтеза, лежащую в основе предполагаемой эффективности ГА и считающуюся их главным отличием от других классов алгоритмов поиска [11]. По сути дела, для решения отдельных вспомогательных подзадач зачастую предлагается использовать алгоритмы, внешние по отношению к ГА (причем именно те традиционные детерминированные методы [8], проблему неэффективности которых и призваны были решить ГА). В то же время усиление детерминированных операторов ГА [4] не позволяет преодолеть ограничения, связанные с этим классом алгоритмов [1]. Тем самым налицо несомненная тенденция превращения генетических алгоритмов в обычные численные детерминированные методы оптимизации.

В заключение следует отметить, что проведенный анализ не затрагивает возможностей применения ГА для улучшения неоптимальных решений. Исследование вопросов сходимости ГА обычно абстрагируется от необходимости получения глобального решения за конечное число шагов, что делает необязательным исследование проблемы останова и связанных с ней трудностей.

Список литературы

- [1] Балдин Е.В., Шашкин Л.О. Генетические алгоритмы: возможности и ограничения // Научно-техническая информация. Сер. 2: Информационные системы и процессы. 2000. №8. С. 19–33.
- [2] Батищев Д.И., Исаев С.А. Оптимизация многоэкстремальных функций при помощи генетических алгоритмов.
- [3] Ведерникова О.Г. Разработка и исследование комбинированного алгоритма генетического поиска и имитации отжига для задачи

размещения элементов СБИС. Дисс. канд. тех. наук. Ростов-на-Дону, 1999.

- [4] Исаев С.А. Разработка и исследование генетических алгоритмов для принятия решений на основе многокритериальных нелинейных моделей. Автореф. дисс. канд. тех. наук. Н. Новгород, 2000.
- [5] Курейчик В.М. Генетические алгоритмы. Обзор и состояние // Новости искусственного интеллекта. 1998. №3. С. 14–63.
- [6] Курейчик В.В. Исследование и разработка генетических алгоритмов для конструкторского синтеза элементов СБИС. Дисс. канд. тех. наук. Таганрог, 1995.
- [7] Лебедев О.Б. Исследование и разработка генетических алгоритмов формирования топологии СБИС повышенной плотности. Дисс. канд. тех. наук. Таганрог, 1997.
- [8] Лорьер Ж.Л. Системы искусственного интеллекта. М.: Мир, 1991.
- [9] Скурихин А.Н. Генетические алгоритмы // Новости искусственного интеллекта. 1995. №4. С. 6–46.
- [10] Щеглов С.Н. Разработка и исследование генетических методов расщепления топологии СБИС. Дисс. канд. тех. наук. Таганрог, 1998.
- [11] Goldberg D.E. Genetic Algorithms in Search, Optimization, and Machine Learning. McGraw-Hill, USA, 1989.
- [12] Srinivas M., Patniak L.M. Genetic Algorithms: A Survey // Computer. 1994. #6.