

Е. С. Семенкин, Р. Б. Сергиенко

КОЭВОЛЮЦИОННЫЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ РЕШЕНИЯ СЛОЖНЫХ ЗАДАЧ УСЛОВНОЙ ОПТИМИЗАЦИИ

Рассматривается коэволюционный алгоритм условной оптимизации, отличающийся автоматической настройкой большинства параметров. Приводятся этапы разработки данного алгоритма, показаны соответствующие исследования. Алгоритм апробирован на практической задаче составления расписания посадок самолетов.

Ключевые слова: генетический алгоритм, условная оптимизация, коэволюция.

На сегодняшний день особое значение приобретают системы поддержки принятия решений в технических, организационных и экономических системах. Автоматизация определенных стадий процесса принятия решения позволяет резко сократить временные, трудовые и финансовые затраты, значительно повысить эффективность и качество принимаемого решения, снизить роль человеческого фактора.

В большинстве случаев задача принятия решения сводится к задаче оптимизации, т. е. к выбору наилучшего варианта из множества существующих.

Задачи оптимизации, которые требуется решать при поддержке принятия решений в реальных системах, обладают свойствами, существенно затрудняющими их решение: дискретные или смешанные переменные, алгоритмически заданные целевые функции, отсутствие удобных для оптимизации свойств или, по крайней мере, отсутствие информации о таких свойствах и т. д. Поэтому в таких задачах зачастую могут быть применены только алгоритмы прямого поиска, не требующие информации о свойствах оптимизируемой функции. В этом отношении перспективными признаны так называемые эволюционные алгоритмы [1].

Основная проблема использования эволюционных процедур – высокая сложность настройки параметров алгоритма для решения конкретной задачи оптимизации. Для наиболее распространенного класса эволюционных методов оптимизации – генетических алгоритмов – требуется выбор оптимального сочетания таких параметров алгоритма, как тип селекции, тип скрещивания, тип мутации и некоторых других настроек в зависимости от класса решаемой задачи оптимизации. Общее число таких комбинаций исчисляется по меньшей мере десятками. Случайный выбор настроек следует считать неприемлемым, так как многие комбинации параметров оказываются почти полностью неработоспособными. Осуществление полного перебора комбинаций параметров алгоритма также недопустимо ввиду очевидных временных, трудовых и материальных затрат.

Поэтому возникает задача разработки процедур, автоматизирующих выбор и настройку параметров генетических алгоритмов. Одним из наиболее перспективных в этом отношении подходов является коэволюционный алгоритм. Ранее были разработаны и исследованы коэволюционные алгоритмы решения задач безусловной оптимизации [2]. В то же время большинство практических задач поддержки принятия решения относятся к классу

условной оптимизации, т. е. имеются ограничения на целевую функцию в виде равенств или неравенств. Таким образом, возникла необходимость разработки коэволюционного генетического алгоритма решения задач условной оптимизации.

Коэволюционный генетический алгоритм – это несколько параллельно действующих стандартных генетических алгоритмов с различными настройками основных параметров (типа селекции, скрещивания, мутации). Отдельный генетический алгоритм называется подпопуляцией. Каждая из подпопуляций оптимизирует заданную функцию и обладает своей стратегией оптимизации. При этом популяции конкурируют за ресурс, который в течение работы алгоритма перераспределяется в пользу более эффективной из них. Стандартный коэволюционный алгоритм состоит из следующих этапов [2]:

- выбора индивидуальных алгоритмов;
- задания параметров коэволюционного алгоритма (размера общего ресурса, величины интервала адаптации, размера штрафа «проигравшего» алгоритма, размера «социальной карты») для всех индивидуальных алгоритмов;
- независимой работы выбранных алгоритмов в течение интервала адаптации (обычно около 5 поколений);
- оценки алгоритмов;
- перераспределения ресурсов;
- миграции лучших индивидов во все подпопуляции.

Перераспределение ресурсов может происходить различными способами. В классической схеме коэволюционного алгоритма изменение размеров ресурсов происходит путем сокращения каждой проигравшей популяции на некоторый процент (определенный заранее размер штрафа) и увеличением победившей популяции на число, равное сумме потерь проигравших. То есть после каждого интервала адаптации определяется один победивший алгоритм (подпопуляция). При этом размер каждой подпопуляции не может быть меньше определенного минимального уровня, называемого «социальной картой».

В данной работе предлагается новый метод перераспределения ресурсов между алгоритмами, получивший название «турнирный». Суть метода заключается в том, что после каждого интервала адаптации определяется не единственный победивший алгоритм, а происходит перераспределение ресурсов в каждой паре подпопуляций («турнир каждого с каждым»). Данный метод перераспределения ресурсов позволяет провести ранжирование подпопуляций по эффективности оптимизации и поощ-

ривать лучшие соответствующим объемом ресурсов, забрав их у худших. Результат такого ранжирования – перераспределение ресурсов между всеми подпопуляциями в соответствии с их эффективностью. Турнирный метод перераспределения ресурсов сравнивался со стандартным на десяти тестовых задачах безусловной оптимизации. В качестве показателя эффективности работы алгоритма использовалась надежность – отношение числа запусков, в которых с заданной точностью был найден оптимум, к общему числу запусков. По результатам исследований [3] выяснено, что турнирный метод перераспределения ресурсов не уступает по эффективности стандартному и даже может давать некоторый прирост надежности, что свидетельствует о целесообразности предложенного подхода.

Миграция индивидов (перераспределение ресурсов) осуществляется следующим образом: по окончании интервала адаптации алгоритмов происходит объединение индивидов из всех подпопуляций в единый массив, который затем сортируется по убыванию пригодности. Лучшие индивиды последовательно передаются во все алгоритмы. Миграция обеспечивает взаимодействие индивидуальных алгоритмов между собой и предположительно должна позволять одновременно использовать преимущества различных алгоритмов. То есть подразумевается, что коэволюционный подход позволяет не только определять алгоритм с наилучшими настройками, но и использовать положительный эффект взаимодействия алгоритмов между собой (свойство эмерджентности).

Таким образом, ключевыми этапами работы коэволюционного алгоритма являются перераспределение ресурсов и миграция, которые обеспечивают «конкуренцию» и «кооперацию» между индивидуальными генетическими алгоритмами соответственно.

Для адаптации генетического алгоритма к решению задач условной оптимизации необходимо ввести в процедуру специальные методы учета ограничений, используемые в ГА. В данной работе рассматривались следующие методы [4]:

- метод «смертельных» штрафов («зануление» пригодности недопустимых индивидов) в сочетании с «лечением» локальным поиском: перемещение заданного процента недопустимых индивидов в допустимую область с помощью локального поиска в пространстве бинарных переменных;
- метод динамических штрафов;
- метод адаптивных штрафов.

Метод статических штрафов, также применяемый для решения задач условной оптимизации с помощью генетических алгоритмов, не рассматривался ввиду значительного числа настраиваемых параметров данного метода, что неприемлемо для процедуры, перед которой поставлена задача максимальной автоматизации настройки параметров алгоритма.

Тестовые задачи условной оптимизации взяты из источников [5] (№№ 1–4) и [6] (№№ 5–8).

Разработка коэволюционного алгоритма условной оптимизации означает добавление к числу автоматически настраиваемых параметров метода учета ограничений. Однако «коэволюционирование» методов учета ог-

раничений требует отказа от стандартной схемы миграции в алгоритме. Дело в том, что миграция основана на группировании индивидов из всех подпопуляций, сортировке их по пригодности и «раздаче» всем алгоритмам наилучших индивидов. Использование же различных методов учета ограничений приводит к тому, что пригодность перестает быть единым критерием оценивания индивидов из различных подпопуляций. Поэтому были разработаны и исследованы три модификации механизма миграции в коэволюционном алгоритме.

Первая получила название «пропорционально-групповой». Суть заключается в объединении подпопуляций с одинаковым методом учета ограничений в группы, сортировке индивидов внутри групп и миграции лучших индивидов из каждой группы во все алгоритмы пропорционально доле группы в общем размере популяции. Последовательность шагов следующая:

- формирование групп: объединение подпопуляций (индивидуальных алгоритмов) с одинаковым методом учета ограничений;
- сортировка индивидов в каждой группе по значению пригодности;
- определение доли каждой группы в общем размере популяции;
- определение количества лучших индивидов в каждой группе, передаваемых в каждый алгоритм: $n_{ij} = \text{int}(N_j s_j)$, где i – номер группы; j – номер алгоритма; N_j – размер алгоритма; s_j – доля группы в общем размере популяции;
- миграция соответствующего числа лучших индивидов из каждой группы в каждый алгоритм.

Вторая схема («паретовская») заключается в последовательном поиске и исключении из общего массива (с передачей во все индивидуальные алгоритмы) точек множества Парето, недоминируемых по двум критериям – значению целевой функции и степени нарушения ограничений. Последовательность шагов следующая:

- объединение в общий массив индивидов из всех алгоритмов;
- поиск недоминируемых точек множества Парето;
- миграция найденных индивидов во все алгоритмы;
- исключение найденных индивидов из общего массива и новый поиск недоминируемых точек множества Парето; и т. д. до тех пор, пока не будут заполнены все алгоритмы необходимым числом индивидов.

В третьем случае (условное обозначение «миграция 3») отдельно формируется массив допустимых индивидов и сортируется по значению целевой функции, и отдельно формируется массив недопустимых индивидов и сортируется либо по степени нарушения ограничений, либо по значению целевой функции. Сортировка недопустимых индивидов по степени нарушения ограничений оказалась менее удачной, поэтому далее приведены исследования только с сортировкой по целевой функции. Последовательность шагов следующая:

- формирование массива всех допустимых индивидов из общей популяции и сортировка их по пригодности;
- формирование массива всех недопустимых индивидов из общей популяции и сортировка их по пригодности;
- заполнение алгоритмов лучшими индивидами из массива допустимых; в случае недостаточного их числа

дополнение лучшими индивидами из массива недопустимых.

Сравнительное исследование различных схем миграции проводилось при включении в коэволюцию пяти индивидуальных алгоритмов с настройками, приведенными в табл. 1.

Далее приведены результаты сравнительного исследования различных схем миграции. Значения надежности, усредненные по пятистам запускам, для всех тестовых задач указаны в табл. 2. Исследования проводились при исходном размере каждой подпопуляции в 100 индивидов, при 100 поколениях, шаге дискретизации для переменных 0,001, интервале для переменных [-10; 10].

Видно, что пропорционально-групповая схема миграции превосходит остальные на всех тестовых задачах. К аналогичному выводу приводит анализ полученных при эксперименте выборок надежности по ранговому критерию Вилкоксона [7]. С доверительной вероятностью 0,95 пропорционально-групповая схема миграции статистически значительно лучше остальных типов миграции на большинстве тестовых задач, в некоторых случаях зафиксирована неопределенность. Таким образом, следует принять пропорционально-групповую схему миграции основной в коэволюционном алгоритме решения задач условной оптимизации.

Для обоснования целесообразности применения коэволюционного алгоритма условной оптимизации необходимо апробировать алгоритм на практической задаче. Для этой цели выбрана задача составления расписания посадок самолетов.

Составление расписания посадок самолетов в аэропорту состоит в установлении очередности приземления воздушных судов (ВС) и назначении каждому из них времени приземления. При этом основными ограничительными критериями являются интервалы безопасности между посадками самолетов, обусловленные аэродинамическими свойствами. Во время полета самолета возникает разность давления между верхней и нижней поверхностями крыла, которая производит вращающуюся массу воздуха. После посадки более тяжелых самолетов

требуется больший временной интервал для следующих воздушных судов. Кроме того, время прибытия самолета должно находиться внутри предопределенного временного окна, ограниченного наиболее ранним и самым поздним временем приземления. Очевидно, что стоимость полета растет с разностью между достигнутым и целевым временем приземления [8].

Исходная математическая модель [9] сформулирована в виде задачи смешанного целочисленного линейного программирования (СЦЛП). Коэволюционный алгоритм условной оптимизации показал низкую эффективность при решении задачи в исходной модели в виде СЦЛП. Алгоритм способен отыскивать допустимые решения, но с очень плохим значением целевой функции. Это объясняется введением в модель значительного числа дополнительных переменных (как вещественных, так и булевых) и соответствующих ограничений для линеаризации модели. Особенно негативно на работе коэволюционного алгоритма сказывается наличие ограничений-равнств на вещественные переменные.

Однако для коэволюционного алгоритма условной оптимизации не является критичным наличие модели в виде линейного программирования. Следовательно, указанную математическую модель можно упростить с целью сокращения числа переменных и ограничений. В результате упрощения получаем задачу оптимизации со следующими целевой функцией и ограничениями:

$$\min \sum_{i=1}^P (g_i \alpha_i + h_i \beta_i),$$

$$E_i \leq x_i \leq L_i, \quad i = 1, \dots, P,$$

$$x_j \geq x_i + S_{ij}, \quad \forall (i, j) \in W, \delta_{ij} = 1,$$

где x_i – реальное время прибытия самолета; P – число самолетов; $[E_i; L_i]$ – допустимый временной интервал для приземления; S_{ij} – ограничительный временной интервал между посадками i -го и j -го самолетов; α_i – разность между ранним и целевым временем прибытием самолета; β_i – разность между поздним и целевым временем прибытия; g_i – штраф за раннее прибытие; h_i – штраф за позднее прибытие; W – множество пар самолетов, для

Таблица 1

Параметры индивидуальных алгоритмов

Алгоритм	Селекция	Скрещивание	Мутация	Метод учета ограничений
1	Пропорциональная	Одноточечное	Слабая	«Смертельные» штрафы + «слечение»
2	Турнирная (3)	Двухточечное	Средняя	Адаптивные штрафы
3	Ранговая	Равномерное	Средняя	Адаптивные штрафы
4	Пропорциональная	Равномерное	Сильная	Динамические штрафы
5	Ранговая	Одноточечное	Средняя	Динамические штрафы

Таблица 2

Сравнение различных схем миграции

Задача	Пропорционально-групповая	Паретовская	Миграция 3
1	0,956	0,898	0,914
2	0,97	0,918	0,96
3	0,146	0,096	0,026
4	0,086	0,052	0,072
5	0,782	0,162	0,406
6	0,164	0,02	0,01
7	0,786	0,618	0,652
8	0,224	0,042	0,034

которых временные ограничений не выполняются автоматически; d_{ij} – флаг прибытия j -го самолета после i -го.

Сформулированная задача включает P вещественных переменных x_i , переменные $\alpha_i, \beta_i, \delta_{ij}$ однозначно определяются через x_i . Число ограничений не более $[2P + P(P-1)/2]$, все из которых представлены неравенствами. Модификация математической модели при неизменных интервале для вещественных переменных и шаге дискретизации позволяет сократить длину индивида до 100 бит, сократить объем поискового пространства с $10^{90} \cdot 2^{100} \approx 10^{120}$ до 10^{30} . Далее приведены исходные числовые данные и результаты решения задачи коэволюционным алгоритмом условной оптимизации (табл. 3–5).

Параметры индивидуальных алгоритмов указаны в табл. 1. Начальное число индивидов для каждой подпопуляции – 2 500 при числе поколений 100, что соответствует $1,25 \cdot 10^6$ вычислений целевой функции с ограничениями, т. е. алгоритм просматривает $1,25 \cdot 10^{24}$ поискового пространства, время работы 5...6 мин на компьютере с процессором Intel Dual-Core с тактовой частотой 1,86 ГГц, 1 Гб оперативной памяти, операционной системой Windows XP.

Следует отметить важность выбора при больших размерах популяций метода сортировки при реализации алгоритма. В данной работе использовалась «быстрая» сортировка [10], в то время как менее эффективные методы (например, обменная сортировка) могут приводить к увеличению времени работы алгоритма в десятки раз.

Решение со значением целевой функции 700 является глобальным оптимумом задачи, другие указанные реше-

ния близки к нему. Полученные результаты свидетельствуют об эффективности использования коэволюционного алгоритма условной оптимизации для решения приведенной практической задачи.

Таким образом, разработан и исследован коэволюционный генетический алгоритм решения задач условной оптимизации, обладающий способностью автоматической настройки большинства параметров. Основное отличие от ранее разрабатываемых коэволюционных алгоритмов заключается в адаптации на класс задач условной оптимизации и автоматической настройке метода учета ограничений. Проведена модификация двух ключевых шагов коэволюционного алгоритма – миграции (для обеспечения работоспособности алгоритма в целом) и перераспределения ресурсов (с целью повышения эффективности алгоритма).

Проверка коэволюционного алгоритма условной оптимизации на тестовых задачах условной оптимизации и апробация на практической задаче планирования посадок самолетов показывает целесообразность использования алгоритма.

Дальнейшим направлением исследований станет экспериментальное сравнение эффективности коэволюционного алгоритма решения задач условной оптимизации и индивидуальных генетических алгоритмов.

Библиографический список

1. Семенкин, Е. С. Метод обобщенного адаптивного поиска для синтеза систем управления сложными объек-

Таблица 3

Числовые данные задачи составления расписания самолетов

Самолеты	1	2	3	4	5	6	7	8	9	10
Самое раннее время приземления (ед. вр.)	129	195	89	96	110	120	124	126	135	160
Целевое время	155	258	98	106	123	135	138	140	150	180
Самое позднее время приземления	559	744	510	521	555	576	577	573	591	657
Штраф в денежных единицах, если прилетел раньше	10	10	30	30	30	30	30	30	30	30
Штраф в денежных единицах, если прилетел позже	10	10	30	30	30	30	30	30	30	30

Таблица 4

Интервалы безопасности между посадками самолетов (в единицах времени)

Самолеты	1	2	3	4	5	6	7	8	9	10
1	–	3	15	15	15	15	15	15	15	15
2	3	–	15	15	15	15	15	15	15	15
3	15	15	–	8	8	8	8	8	8	8
4	15	15	8	–	8	8	8	8	8	8
5	15	15	8	8	–	8	8	8	8	8
6	15	15	8	8	8	–	8	8	8	8
7	15	15	8	8	8	8	–	8	8	8
8	15	15	8	8	8	8	8	–	8	8
9	15	15	8	8	8	8	8	8	–	8
10	15	15	8	8	8	8	8	8	8	–

Таблица 5

Наилучшие решения задачи составления расписания посадок самолетов

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	Целевая функция	Номер поколения
167	258	98	106	119	127	135	143	152	182	780	87
166	258	98	106	119	127	135	143	151	181	710	77
165	258	98	106	118	126	134	142	150	180	700	93

тами / Е. С. Семенкин, В. А. Лебедев. М. : МАКС-Пресс, 2002.

2. Семенкин, Е. С. Об эволюционных алгоритмах решения сложных задач оптимизации / Е. С. Семенкин, А. В. Гуменникова, М. Н. Емельянова, Е. А. Сопов // Вестн. Сиб. гос. аэрокосмич. ун-та им. акад. М. Ф. Решетнева : сб. науч. тр. / под ред. проф. Г. П. Белякова ; Сиб. гос. аэрокосмич. ун-т. Вып. 5. Красноярск, 2003. С. 14–23.

3. Сергиенко, Р. Б. Разработка турнирного метода перераспределения ресурсов между подпопуляциями в коэволюционном алгоритме / Р. Б. Сергиенко // Инновационные недра Кузбасса. IT-технологии : сб. науч. тр. Кемерово : ИНТ, 2007. С. 401–404.

4. Michalewicz, Z. Evolutionary algorithms for constrained parameter optimization problems / Z. Michalewicz, M. Schoenauer // Evolutionary Computation, 1996. Vol. 34, №1. P. 1–32.

5. Рубан, А. И. Методы оптимизации : учеб. пособие / А. И. Рубан. Красноярск : НИИ ИПУ, 2001.

6. Ворожейкин, А. Ю. Адаптивные эволюционные алгоритмы решения сложных задач оптимизации : дисс. ... канд. техн. наук / А. Ю. Ворожейкин. Красноярск, 2008.

7. Гмурман, В. Е. Теория вероятностей и математическая статистика : учеб. пособие / В. Е. Гмурман. М. : Высш. образование, 2006.

8. Carr, G. C. Airline arrival prioritization in sequencing and scheduling [Electronic resours] / G. C. Carr, H. Erzberger, F. Neuman // Air Traffic Management R&D Seminar. Orlando, 1998. Access mode: <http://www.ctas.arc.nasa.gov/publications>. Title from a display.

9. Scheduling aircraft landings – the static case / J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, D. Abramson // Transportation Science. 2000. Vol. 34. P. 180–197.

10. Шилдт, Г. Теория и практика C++ / Г. Шилдт. СПб. : BHV-Санкт-Петербург, 1996

E. S. Semenkin, R. B. Sergienko

COEVOLUTIONARY GENETIC ALGORITHM FOR SOLVING COMPLEX PROBLEMS OF CONSTRAINED OPTIMIZATION

Coevolutionary algorithm of constrained optimization with automatic tuning of parameters is considered. The process of development and results of effectiveness investigation are described. The algorithm effectiveness is illustrated on the task of aircrafts' landing scheduling.

Keywords: genetic algorithm, constrained optimization, coevolution.

УДК 519.682

О. И. Егорушкин, Д. А. Калугин-Балашов, К. В. Сафонов

О РАЗРЕШИМОСТИ СИСТЕМ НЕКОММУТАТИВНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ, ПОРОЖДАЮЩИХ КОНТЕКСТНО-СВОБОДНЫЕ ЯЗЫКИ

Рассматриваются системы алгебраических (полиномиальных) уравнений над некоммутативным относительно умножения кольцом. Получено условие разрешимости таких систем в виде формальных степенных рядов, изучаются системы линейных алгебраических уравнений, для которых исследована возможность понижения порядка систем. Рассмотренные системы обобщают свойства систем уравнений, определяющих контекстно-свободные и линейные языки.

Ключевые слова: контекстно-свободные языки, системы алгебраических уравнений, некоммутативное кольцо, коммутативный образ, граф инцидентности

Формальным языком L называют множество цепочек в алфавите $\{x_1, \dots, x_n\}$ выделенных с помощью конечного набора правил. Выделенные цепочки, принадлежащие свободной полугруппе $\{x_1, \dots, x_n\}^*$, называются при этом либо словами (над алфавитом), либо правильно построенными предложениями (над словарем), либо грамматически правильными предложениями. Конечное множество правил, с помощью которых выделяются цепочки, называют грамматикой. Таким образом, формальный язык определяется совокупностью соответствующих правил и способом выделения цепочек с помощью этих правил.

Практически важный класс формальных языков образуют контекстно-свободные языки (кс-языки), поскольку они являются адекватным средством моделирования естественных языков, а также языков программирования [1–3]. Обозначим $W = \{x_1, \dots, x_n\} \cup \{z_1, \dots, z_n\}$ свободную полугруппу с операцией умножения; ее элементами являются произвольные цепочки, составленные из элементов «расширенного» алфавита $\{x_1, \dots, x_n, z_1, \dots, z_n\}$.

Рассмотрим играющую роль словаря конечное множество $X = \{x_1, \dots, x_n\}$, состоящее из слов x_i языка и называемое терминальным множеством, а также $Z = \{z_1, \dots, z_n\}$ –