ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ПРИ РЕШЕНИИ ЗАДАЧ КОМБИНАТОРНОЙ ОПТИМИЗАЦИИ

Скобцов Ю.А., Хмелевой С.В., Дробинский В.Г.
Донецкий Национальный технический университет, каф. АСУ.
E–mail: skobtsov@kita.dgtu.donetsk.ua

Abstract
Skobtsov Yu., Khmelevoy S., Drobinsky V. The genetic algorithms application to combinatorial optimization. *The travelling salesman problem is taken as an benchmark. The effect of different genetic algorithm parameters is researched. The program implementation is verified for international test benchmark problems.*

Существует достаточно большой круг задач, для которых нет достаточно быстрых алгоритмов поиска оптимальных решений. К ним относится, например, построение кратчайшего пути, получение наибольшей прибыли, минимизация затрат и т.п. Большую часть подобных проблем составляют задачи комбинаторной оптимизации. Они часто встречаются в бизнесе и производственной деятельности. Очень часто комбинаторная оптимизация встречается не как отдельная задача, а как составная часть более сложной проблемы.

Известны классические методы решения таких задач, в основном, основанных на переборе вариантов возможных решений и дающих оптимальное решение. Но при увеличении сложности задачи время нахождения решений такими алгоритмами катастрофически увеличивается. С другой стороны существуют эвристические алгоритмы, которые дают субоптимальное решение. Размерность задач комбинаторной оптимизации, встречающихся на практике постоянно растет. Поэтому разрабатываются новые виды эффективных алгоритмов, обладающих все большим быстродействием. Одним из самых перспективных направлений являются генетические алгоритмы (ГА).

Генетические алгоритмы решения задачи коммивояжера.

Напомним постановку этой классической комбинаторной оптимизационной задачи. Дано n городов и стоимость проезда между произвольной парой городов. Коммивояжер должен посетить каждый город на своём участке один раз и вернуться в исходный пункт. Необходимо выбрать оптимальный маршрут с минимальной стоимостью тура (обхода всех городов).

Пространством поиска решений этой задачи является множество перестановок n городов. Любая простая (одиночная) перестановка n городов даёт решение, являющееся полным туром из n городов. Оптимальным решением является перестановка, которая даёт минимальную стоимость тура. Очевидно размерность пространства поиска пропорциональна n!. Известно, что задача коммивояжера (ЗК) является NP – полной, т.е. переборной. Она имеет многочисленные практические приложения, в которых число "городов" может быть достаточно большим. За последние десятилетия разработано достаточно много алгоритмов решения этой задачи, дающих субоптимальное решение. В последнее десятилетие эта задача является базовой для исследования ГА в области комбинаторной оптимизации [1,3].

При решении ЗК генетические алгоритмы показали себя с лучшей стороны. Хотя на задачах малой размерности они значительно уступают разработанным ранее (в основном эвристическим) алгоритмам, при увеличении размерности задачи их эффективность возрастает. Со сверхбольшими задачами за разумное время уже могут справиться только методы, разработанные на основе ГА.

Первоначально использовалась следующая классическая схема ГА.

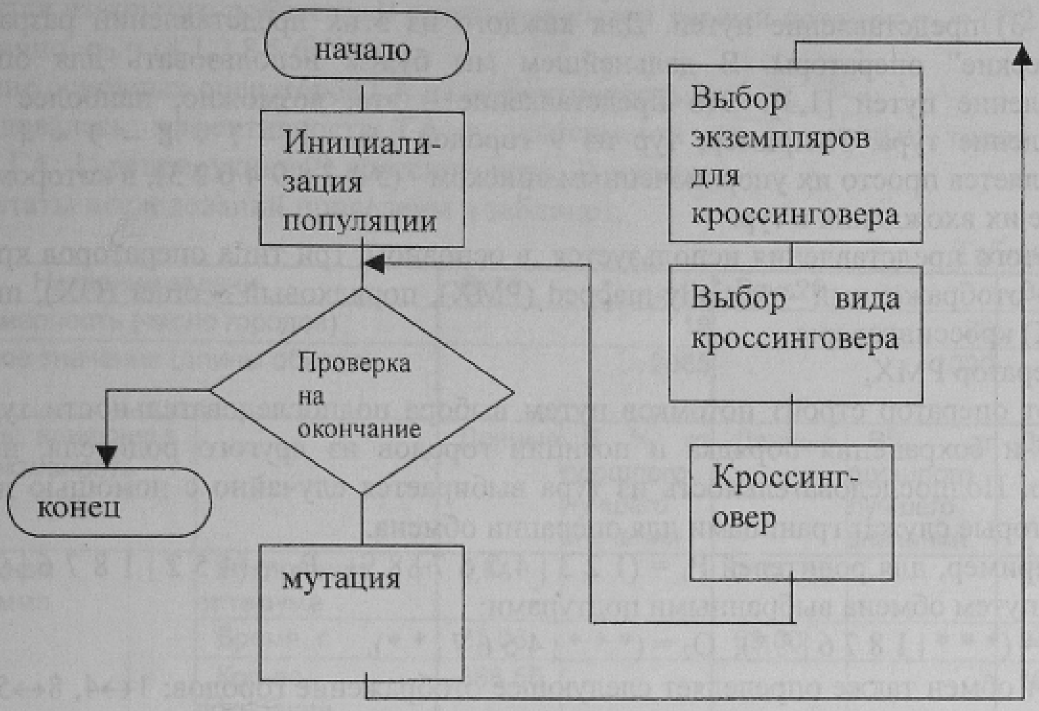

Рис.1 Схема алгоритма

Как видно и рисунка основу ГА составляют операторы репродукции (выбора для "размножения"), кроссинговера и мутации. В дальнейшем эта схема была несколько изменена. Критерием окончания являлась стабилизация оптимального результата за определенное число поколений. В первоначальном варианте число решений в популяции было постоянно; новые экземпляры добавлялись только вместо старых. Для борьбы с преждевременной сходимостью алгоритма был применен следующий прием: уничтожались все экземпляры с одинаковым значением целевой (фитнесс) функции , кроме одного (при достаточно больших турах вероятность того, что существуют два разных тура с одинаковым значением фитнесса, достаточно мала).

Машинные эксперименты проводились следующим образом: выполнялась серия опытов, находились средние значения целевой функции (длина тура) для популяции . Для каждого опыта измерялись следующие параметры: 1) отклонение среднего решения от оптимального, 2) время поиска решения и 3) количество потребовавшихся поколений. При выборе эффективности данного фактора принимались во внимание только первые два параметра (третий нужен для объяснения поведения системы при данном факторе). По международным стандартам апробация алгоритмов проводилась на задачах из международного каталога [5]. В первую очередь была исследована стандартная задача bays29 из [5].

Для решения задачи с применением ГА нужно прежде всего выбрать (разработать) представление (кодирование) решения (хромосомы) и определить на нем основные операторы ГА : репродукцию, кроссинговер и мутацию.

Очевидно, что классическое двоичное представление хромосомы (тура) при решении ЗК нецелесообразно. Действительно если мы интересуемся оптимальной перестановкой городов, т.е. $(i_1, i_2, \dots, i_n)$ и используем двоичное представление в виде одного бинарного вектора, то изменение даже в одном бите может дать двоичный вектор, не принадлежащий к области решения, т.е. не являющейся перестановкой n городов.

В последнее время , в основном используются 3 способа представления тура при решении ЗК с использованием ГА [1,2]: 1) представление соседства; 2) представление порядка; 3) представление путей. Для каждого из этих представлений разработаны свои "генетические" операторы. В дальнейшем мы будем использовать для описания тура представление путей [1,3]. Это представление – это, возможно, наиболее естественное представление тура. Например, тур из 9 городов 5 – 1 – 7 – 8 – 9 – 4 – 6 – 2 – 3 представляется просто их упорядоченным списком (5 1 7 8 9 4 6 2 3), в котором города идут в порядке их вхождения в тур.

Для этого представления используется ,в основном, три типа операторов кроссинговера: частично отображенный - partially-mapped (PMX), порядковый – order (OX), циклический - cycle (CX) кроссинговеры.

Оператор PMX.

Этот оператор строит потомков путем выбора подпоследовательности тура из одного родителя и сохранения порядка и позиции городов из другого родителя, насколько это возможно. Подпоследовательность из тура выбирается случайно с помощью двух секущих точек, которые служат границами для операции обмена.

Например, для родителей P_1 = (1 2 3 | 4 5 6 7 | 8 9), P_2 = (4 5 2 | 1 8 7 6 | 9 3) потомки строятся путем обмена выбранными подтурами:

O_1 = (* * * | 1 8 7 6 | * *), O_2 = (* * * | 4 5 6 7 | * *).

Этот обмен также определяет следующее отображение городов: 1↔4, 8↔5, 7↔6, 6↔7. Затем заполняются позиции, помеченные звездочками, городами из своих родителей, для которых нет конфликтов, то есть не образуются преждевременные циклы:

σ_1 = (* 2 3 | 1 8 7 6 | * 9), σ_2 = (* * 2 | 4 5 6 7 | 9 3).

При возникновении конфликта используется приведенное отображение. В нашем примере первую звездочку в σ1 нельзя заменить на 1, поскольку возникает преждевременный цикл, поэтому с помощью отображения * заменяется на 4, и предпоследняя * заменяется на 5: σ_1 = (4 2 3 | 1 8 7 6 | 5 9). Аналогично строится второй потомок σ_2 = (1 8 2 | 4 5 6 7 | 9 3).

Оператор OX.

Оператор OX использует то , что при данном представлении важен прежде всего порядок городов, а не исходный пункт. Например, туры 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 и 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 1 являются фактически идентичными.

Оператор строит потомка путем выбора подтура из одного родителя и сохранением относительного порядка городов из другого родителя. Например, для родителей

P_1 = (1 2 3 | 4 5 6 7 | 8 9), P_2 = (4 5 2 | 1 8 7 6 | 9 3) потомок строится так: сначала сегменты между двумя секущими точками копируются в потомки:

O_1 = (* * * | 4 5 6 7 | * *), P_2 = (* * * | 1 8 7 6 | * *).

Далее, начиная со второй секущей точки одного родителя города из другого родителя копируются в том же порядке, пропуская при этом уже присутствующие города в построенном подтуре. При достижении конца списка этот процесс продолжается с первой позиции до первой точки сечения (по кольцу). Для нашего примера после второй точки сечения во втором родителе имеем последовательность P_2 = (9-3-4-5-2-1-8-7-6). Далее удаляется из этой последовательности города 4-5-6-7, так как они уж есть в O_1. Получается P_1 = (9-3-2-1-8). Эта подпоследовательность помещается в первый потомок, начиная со второй точки сечения по кольцу: σ_1=(2 1 8 | 4 5 6 7 | 9 3), аналогично σ_2=(3 4 5 | 1 8 7 6 | 9 2).

Циклический оператор CX.

Здесь потомок строится так, что каждый город вместе со своей позицией идет от одного из родителей. Например, для родителей P_1 = (1 2 3 4 5 6 7 8 9), P_2 = (4 1 2 8 7 6 9 3 5) сначала получаем потомка путем выбора первого города из первого родителя:

$\sigma_1^1 = (1 * * * * * * * * *)$.Этот выбор определяет следующий город: (в первой позиции из второго родителя): $\sigma_1^2 = (1 * * 4 * * * * *)$. Город 4 имплицирует город 8: $\sigma_1^3 =(1 * * 4 * * * 8 *)$, аналогично $\sigma_1^5 = (1\ 2\ 3\ 4 * * * 8 *)$. Здесь процесс прерывается, так как выбор 2-1 ведет к преждевременному циклу (город 1 уже есть в построенном подтуре). Поэтому оставшиеся города берутся из другого родителя. В результате имеем первый потомок $\sigma_1 = (1\ 2\ 3\ 4\ 7\ 6\ 9\ 8\ 5)$ и аналогично $\sigma_2 = (4\ 1\ 2\ 8\ 5\ 6\ 7\ 3\ 9)$.

Влияние основных операторов ГА на эффективность.

Исследовалась эффективность ГА в зависимости от различных типов основных операторов ГА: 1) репродукции,2) кроссинговера, 3) мутации.

Результаты исследований приведены в таблице1.

Таблица1.

Название задачи			Gr18		Bays29		Berlin52	
Размерность (число городов)			18		29		52	
Оптимальное значение (длины обхода)			2085		2020		7553	
Операторы, влияющие на эффективность			Данные	В % от прошлого лучшего значения	Данные	В% от прошлого лучшего значения	Данные	В% от прошлого лучшего значения
Исходная программа	1	Отклонение от оптимума	4,54	-	2100,00	-	1148,00	-
		Время, с	25,08	-	65,00	-	346,64	-
		Кол-во поколений	388,55	-		-	1650,00	-
Кроссинговер со случайными точками разреза	2	Отклонение от оптимума	*1,33*	70,70	*30,53*	98,55	*820,60*	28,52
		Время, с	*3,08*	87,72	*11,60*	82,15	*76,56*	77,91
		Кол-во поколений	*116,27*	70,08	*214,93*	-	*460,40*	72,10
Использование мутации сдвига	3	Отклонение от оптимума	*1,00*	24,81	*21,87*	28,37	*544,16*	33,69
		Время, с	*3,67*	-19,16	*11,76*	-1,38	*86,22*	-12,62
		Кол-во поколений	*118,37*	-1,81	*223,10*	-3,80	*533,00*	-15,77
Использование комбинации двух мутаций	4	Отклонение от оптимума	*0,00*	100,00	*20,14*	7,91	717,00	-31,76
		Время, с	*4,16*	-13,35	*11,54*	1,87	76,69	11,05
		Кол-во поколений	*120,63*	-1,91	*193,10*	13,45	484,48	9,10
Стандартный оператор репродукции	5	Отклонение от оптимума	2,00	-	31,97	-58,74	675,30	-24,10
		Время, с	4,76	-14,42	13,43	-16,38	87,24	-1,18
		Кол-во поколений	117,97	2,21	211,47	-9,51	488,70	8,31
Серия из 15 запусков, поп.=125, ост.15 неуд.	6	Отклонение от оптимума	***0,00***	-	*8,87*	55,96	570,75	-4,89
		Время, с	***2,64***	36,54	*11,66*	-1,04	83,51	3,14
		Кол-во поколений	***847,27***	-602,37	*1804,67*	-834,58	4142,25	-677,16
Серия из 10 запусков, поп.=125, ост.20 неуд.	7	Отклонение от оптимума	0,00	-	***8,00***	9,81	***638,39***	-11,85
		Время, с	2,88	-9,09	***8,48***	27,27	***63,04***	24,51
		Кол-во поколений	998,97	-17,90	***1378,93***	23,59	***3035,60***	26,72

Лучшая комбинация операторов ГА по сравнению с первоначальной		Отклонение от оптимума	***0,00***	100,00	***8,00***	99,62	***638,39***	44,39
		Время, с	***2,64***	89,47	***8,48***	86,95	***63,04***	81,81
		Кол-во поколений	***847,27***	-118,06	***1378,93***	-	***3035,60***	-83,98

Таблица1. Окончание.

Название задачи			Brasil58		Eil76		Eil101	
Размерность (число городов)			58		76		101	
Оптимальное значение (длины обхода)			22086*		597		597	
Операторы, влияющие на эффективность			Данные	В% от прошлого лучшего значения	Данные	В% от прошлого лучшего значения	Данные	В% от прошлого лучшего значения
Исходная программа	1	Отклонение от оптимума	5105,80	-	105,53	-	194,00	-
		Время, с	679,71	-	698,34	-	2715,45	-
		Кол-во поколений	1144,80	-	1453,27	-	2418,80	-
Кроссинговер со случайными точками разреза	2	Отклонение от оптимума	*2083,60*	59,19	*114,40*	-8,40	*227,50*	-17,27
		Время, с	*145,74*	78,56	*468,36*	32,93	*1375,66*	49,34
		Кол-во поколений	*710,00*	37,98	*895,40*	38,39	*1520,00*	37,16
Использование мутации сдвига	3	Отклонение от оптимума	***1490,25***	28,48	*105,50*	7,78	*202,00*	11,21
		Время, с	***147,23***	-1,02	*413,66*	11,68	*1432,16*	-4,11
		Кол-во поколений	***652,60***	8,08	*995,00*	-11,12	*1619,50*	-6,55
Использование комбинации двух мутаций	4	Отклонение от оптимума	1968,44	-32,09	125,33	-18,80	261,43	-29,42
		Время, с	139,87	5,00	409,27	1,06	1197,84	16,36
		Кол-во поколений	624,64	4,28	1007,00	-1,21	1409,71	12,95
Стандартный оператор репродукции	5	Отклонение от оптимума	2262,00	-51,79	113,63	-7,71	255,00	-26,24
		Время, с	111,83	24,04	441,29	-6,68	1344,09	6,15
		Кол-во поколений	549,60	15,78	1049,00	-5,43	1543,25	4,71
Серия из 15 запусков, поп.=125, ост.15 неуд.	6	Отклонение от оптимума	5629,53	-277,76	174,67	-65,56	*305,25*	-51,11
		Время, с	76,25	48,21	212,66	48,59	*647,06*	54,82
		Кол-во поколений	2451,47	-275,65	4241,33	-326,26	*5418,75*	-234,59
Серия из 10 запусков, поп.=125, ост.20 неуд.	7	Отклонение от оптимума	5436,15	-176,17	***130,88***	-4,43	***277,63***	9,05
		Время, с	60,00	57,10	***184,30***	54,97	***525,94***	18,72
		Кол-во поколений	2547,50	-307,83	***3764,06***	-273,79	***4998,25***	7,76
Лучшая комбинация операторов ГА по сравнению с первоначальной		Отклонение от оптимума	***1490,25***	70,81	***130,88***	-24,02	***277,63***	-43,11
		Время, с	***147,23***	78,34	***184,30***	73,61	***525,94***	80,63
		Кол-во поколений	***652,60***	42,99	***3764,06***	-159,01	***4998,25***	-106,64

В табл.1 столбцы соответствуют типовым ЗК из международного каталога [5], строки – машинным экспериментам с различными операторами ГА. *Курсивом* отмечен оператор, при котором достигнуто улучшение, ***жирным подчеркнутым курсивом*** – лучшая комбинация операторов ГА. Наличие * означает, что для данного тура не представлен оптимальный маршрут и оптимумом считается лучший полученный результат измерений.

Большинство опытов были проведены при размере популяции около 1000 особей. При проведении серии опытов размер популяции брался равным 125. Критерий остановки: количество поколений, при которых не достигнуто улучшения = 60 (при серии опытов – 15 или 20).

Операторы кроссинговера, взятые из [1] и приведенные выше, использованы практически без изменения. Использовалась комбинация из всех трех методов. Было проведено только исследование разницы производительности между операциями с постоянными (1 строка табл.) и со случайными (2 строка табл.) точками разреза. Как и следовало ожидать, метод со случайными точками разреза обладает намного лучшими характеристиками. Это объясняется большей сохранностью удачных подтуров.

Оператор мутации

Исследованы следующие виды мутации (из [4]):

- случайно выбрать два города, поменять их местами (мутация перестановки), строка1 табл.1.
- случайно выбрать два города, первый город поставить на место второго, города между этими двумя точками сдвинуть на одну позицию в направлении убранного первого (мутация сдвига), строка 3 табл.1.

Как следовало ожидать, второй оператор должен был оказаться более действенным, чем первый, поскольку он с меньшей вероятностью разрушает уже правильные подтуры, комбинации городов. Но практическая реализация такой операции занимает большее время. Поэтому несмотря на то, что отклонение от оптимального решения уменьшилась на 11-33%, время поиска решения колеблется от –19 до 11% (В некоторых опытах ухудшилось). Несмотря на некоторое ухудшение времени поиска, значительное увеличение точности решения показало целесообразность использования именно этого вида мутации. В дальнейшем была внедрена смешанная система, т.е. случайным образом производился выбор типа оператора мутации для текущего поколения (строка 4 табл.). Результаты оказались несколько неоднозначными. Давая в выигрыш во времени решения (а в некоторых опытах – и в точности), в большинстве опытов отклонение от оптимума значительно увеличилось. Поэтому данная система в дальнейшем не использовалась.

Оператор репродукции.

В качестве альтернативных вариантов оператора репродукции использовались следующие варианты:

- Стандартный оператор репродукции типа «колесо рулетки» [1] (строка 5 табл.1);
- Модифицированный оператор репродукции [3], при котором особь допускается к размножению в зависимости от некоторой экспоненциальной случайной функции. В отличие от стандартного оператора, здесь вероятность допуска первых элементов к размножению гораздо выше (строка 1 табл.1);
- Оператор репродукции типа "аутбридинг". Аутбридинг [3] заключается в том, что один из двух родителей выбирается случайно, а другой – как можно дальше от первого родителя. Естественно, для этого необходимо определить понятие расстояния между двумя турами.

При исследовании этих операций лучшим оказался модифицированный оператор. Результат «колеса рулетки» скорее всего объясним тем, что лучшие экземпляры популяции имеют слишком мало шансов быть допущенными к размножению. На основе же аутбридинга были произведены опыты только с одном туром. Результат этой операции дает определенный выигрыш в точности, но реализация операции «расстояния между турами» связана с значительной сложностью, поэтому данный метод неприменим (значительное увеличение времени нахождения результата, порядка 20%).

Эксперименты на серии опытов. Суть этого подхода заключается в том, что вместо одного опыта проводилась серия из нескольких опытов с пониженными требованиями к качеству и уменьшенной популяцией. В качестве итогового результата брался лучший результат популяции. Данное изменение очень вариативно и зависит от числа опытов в серии, размера популяции и требований к качеству. Наилучший результат, которого удалось достигнуть – (60.28%, 26.52%, -614,10%) показывает, что данное изменение достаточно эффективно. Значительное же ухудшение результатов на больших задачах говорит о недостаточной мощности начальной популяции.

Выводы

В результате внесения указанных изменений в основную схему ГА , а также некоторого улучшения работы самой программы были достигнуты следующие результаты: уменьшение отклонения от оптимального решения на (до 90%), уменьшение времени поиска решения (до 90%). Также отмечена большая зависимость точности решения от размерности задачи. Избыток начальных точек на задаче GR18 привел к практически полному нахождению оптимума в каждом эксперименте. На больших же задачах точность значительно теряется. Сравнение эффективности решения задач комбинаторной оптимизации с использованием ГА и стандартных показывают, что хотя на задачах малой и средней размерности имеет ГА проигрывают в быстродействии, на больших и сверхбольших размерностях ГА имеют явное преимущество. Но генетические алгоритмы – это только общее направление. В самих алгоритмах существует множество направлений, порой друг другу противоречащих или же применимых только для определенного вида задач. Только умелое их применение, правильный их изначальный выбор и грамотная реализация способны сделать конкретную реализацию действительно производительной.

Литература

1. Zbigniew Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs. Springer, Varczava,1999.
2. Д.И. Батищев, С.А. Исаев. Оптимизация многоэкстремальных функций с помощью генетических алгоритмов
3. Курейчик В.М. Генетические алгоритмы. –Таганрог: издательство ТГТУ,1998
4. А.В. Кисляков. Генетические алгоритмы: операторы скрещивания и мутации. //Информационные технологии. 2001, №1
5. *http://www.keck.caam.rice.edu/tsp/*