сложность предложенного генетического алгоритма о кратчайшем пути оценивается как 0 (n^2), что позволяет сократить время решения задачи о назначении.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Вагнер Г.* Основы исследования операций. − М.: Мир, 1972, т. 1. − 335 с.
2. *Гольштейн Е.Г., Юдин Д.Б.* Задачи линейного программирования транспортного типа. – М.: Наука, 1969. − 382 с.
3. *Кофман А.* Введение в прикладную комбинаторику. − М.: Наука, 1975. − 479 с.
4. *Чернышев Ю.О.* Электронное моделирование задачи о назначении // Однородные цифровые и интегрирующие структуры. − Таганрог: Изд-во ТРТИ, 1977, вып. 8. − С. 99-103.
5. *Чернышев Ю.О., Насекин В.А.*Сведение задачи выбора максимальных интервалов булевой функции к нахождению кратчайшего пути // Известия ВУЗов. Электротехника. − 1974, №3. − С. 235-238.
6. *Чернышев Ю.О., Басова А.В., Полуян А.Ю.* Решение задач транспортного типа генетическими алгоритмами. − Ростов-на-Дону: Изд-во ЮФУГОУ, 2008. − 87 с.

УДК 004.896

А.Н. Берёза, А.С. Стороженко

КОМБИНИРОВАННЫЙ МНОГОПОПУЛЯЦИОННЫЙ МУРАВЬИННЫЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ*

Введение. Комплексная интеллектуализация систем автоматизированного проектирования в электронике (САПР-Э или ECAD – Electronic Computer Aided Design) предполагает разработку интеллектуальных процедур для всех этапов маршрута проектирования. Одной из таких процедур является оптимизация схемотехнических решений на транзисторном уровне (transistor level). Цель оптимизации заключается в нахождении такой точки пространства допустимых решений, в которой критерии оптимальности принимают экстремальные значения или имеют желаемые характеристики. Создание интеллектуальных процедур параметрической оптимизации позволит повысить качество схемотехнических решений и значительно сократить время их проектирования.

В настоящее время созданы алгоритмы и описаны методы для решения различных оптимизационных задач, как для конкретных случаев, так и в общем виде. Эффективным методом решения оптимизационных задач являются генетические алгоритмы (ГА), основоположником которых считается Джон Холланд (John Holland), описавший ключевые принципы ГА в своей книге «Адаптация в естественных и искусственных системах» (Adaptation in Natural and Artificial Systems).

ГА моделируют процесс эволюции живой природы и на основе эволюционных принципов осуществляют поиск лучших решений [1, 2]. Основным недостатком этих алгоритмов является быстрая сходимость к субоптимальному решению и слабая эффективность поиска при небольшом разнообразии генетического материала. Для обеспечения выхода из локального оптимума применяются различные эвристики [1, 3, 4], одной из которых является *многопопуляционные ГА* (МГА).

Принцип функционирования МГА заключается в создании определенного количества популяций, развивающихся до определенного времени самостоятельно.

* Работа выполнена при поддержке РФФИ (гранты № 07-01-00174, № 08-01-00473).

При наступлении этого времени, называемого временем миграции, популяции обмениваются между собой решениями. Процесс обмена решениями называют механизмом миграции, на сегодняшний день существует несколько реализаций этого механизма [4, 5].

МГА за счет развития нескольких популяций и механизма обмена хромосомами обеспечивают разнообразие генетического материала, чем способствуют выходу из локальных оптимумов. Основным недостатком МГА является значительное время проведения поиска, это связано с большим количеством популяций и большими затратами времени на проведения поиска отдельной популяции. Устранить этот недостаток позволяет гибридизация МГА и алгоритма поиска при помощи муравьиной колонии.

Гибридный многопопуляционный муравьиный генетический алгоритм (ММГА). Муравьиные алгоритмы были разработаны Марко Дориго в 90-х годах XX века, они моделируют поведение группы муравьев, ищущих пищу [2, 6, 7]. Главная идея алгоритма состоит в том, что по ходу своего передвижения муравей на земле оставляет след феромона, и муравей, идущий за ним, будет выбирать дорогу, основываясь на том, где больше феромона. Также на выбор пути влияет внешняя среда: под воздействием внешней среды на каждой итерации алгоритма количество феромона уменьшается.

Муравьиные алгоритмы как и ГА работают с некоторым набором решений, но, в отличие от них, не отбрасывают худшие решения сразу, а позволяют, постепенно уменьшая значения феромона, свести к нулю вероятность выбора конкретного пути, что эквивалентно сведению к нулю вероятности появления конкретного решения при проведении дальнейшего поиска.

В ММГА задача, решаемая при помощи муравьиных алгоритмов, представляется в виде полного ориентированного графа, в вершинах которого располагаются популяции и могут находиться муравьи, а дугами этого графа являются альтернативными переходы, которые может осуществлять муравей в процессе своего передвижения. Весом дуги является значение следа феромона, оставленного муравьем, прошедшим по этой дуге. Для дуг, входящих в один и тот же узел, вес дуги считается одинаковым.

Движение муравьев заключается в перемещении муравья из одной вершины в другую по соответствующей дуге, также перемещение муравья в некоторую вершину сопровождается запуском ГА для популяции, находящейся в этой вершине.

Выбор направления перехода основывается на одном простом вероятностном правиле. Если муравей еще не закончил путь, то есть количество проб выбора популяций не равно их числу, то выбор следующей вершины выполняется при помощи алгоритма «колеса рулетки» на основе вероятностей выбора всех вершин, связанных с той, в которой он находится. Если муравей находится в i-ой вершине, то вероятность выбора j-ой вершины рассчитывается по следующей формуле:

$$P_{i,j}(t) = \begin{cases} \dfrac{[\tau_{i,j}(t)]^{\alpha}}{\sum\limits_{l\in J_{i,k}} [\tau_{i,l}(t)]^{\alpha} \cdot \sum\limits_{m=0}^{C_p} P_m(t)}, j \in J_{i,k}, m \neq k \\ 0, j \notin J_{i,k} \end{cases},$$

где α – параметр, задающий степень веса следа феромона, $\tau_j(t)$ – количество феромона j-ой популяции, C_p – количество популяций. Правило расчета вероятности выбора следующей популяции для развития во время проведения поиска не изменяется, но выбор следующей популяции является вероятностным, т.к. у каждого муравья разный пройденный им путь.

Пройдя в точку *j*, муравей откладывает на ребра $(i, j), i \in [0, C_p]$, где C_p – количество популяций, некоторое количества феромона, которое должно быть связано с оптимальностью сделанного выбора. Пусть $MaxCF_j^t$ – максимальное значение целевой функции *j*-ой популяции, ΔCF_j^t – изменение ЦФ *j*-ой популяции, тогда откладываемое количество феромона может быть задано в следующем виде:

$$\Delta\tau_{i,j,k}(t) = \begin{cases} a_{CF} \dfrac{MaxCF_j^t}{\sum\limits_{m=0}^{c_p} MaxCF_m^t} + (1 - a_{CF}) \dfrac{MaxCF_j^{t-1} - MaxCF_j^t}{\sum\limits_{m=0}^{c_p} \Delta MaxCF_m^t}, (i, j) \in J_k(t); \\ 0, (i, j) \notin T_k(t), \end{cases}$$

где $a_{CF} \in [0,1]$ – коэффициент, учитывающий процент влияния на откладываемый след феромона, значение ЦФ и развитие популяции, C_p – количество популяций.

Воздействие внешней среды определяется, в первую очередь, испарением феромона. Пусть $p \in [0,1]$ является коэффициентом испарения, тогда уравнение испарения имеет вид:

$$\tau_{i,j}(t+1) = (1-p) \cdot \tau_{i,j}(t) + \Delta\tau_{i,j}(t), \text{ где } \Delta\tau_{i,j} = \sum_{k=1}^{Cp} \tau_{i,j,k}(t).$$

Принцип работы ММГА заключается в следующем: последовательно помещают муравьев во все вершины графа, сопровождая при этом запуском ГА соответствующей популяции. После помещения муравья в последнюю вершину графа, процесс продолжается.

Муравей должен сделать некоторое количество перемещений, совпадающее с количеством вершин графа (количеством популяций). Алгоритм продолжает работу, пока не истечет выделенное количество итераций либо, пока среднее количество следов феромона не будет меньше определенного числа.

Описанный гибридный алгоритм позволяет существенно сократить время поиска, но, как и ГА, он слабо эффективен при небольшом разнообразии генетического материала. Небольшое разнообразие генетического материала обычно наблюдается при сходимости алгоритма к субоптимальному решению. Слабая эффективность поиска объясняется тем, что при выполнении оператора кроссинговера полученные потомки будут с большой долей вероятности совпадать с родителями или другими хромосомами популяции. Следовательно, генерация нового решения полностью обуславливается оператором мутации, который работает с определенной довольно маленькой вероятностью. Поэтому новые решения будут появляться с вероятностью, соответствующей значению вероятности оператора мутации, а все остальные решения будут одинаковые. Устранить этот недостаток можно, используя на финальных этапах поиска решений другой алгоритм. Такой способ использования различных алгоритмов называется комбинацией этих алгоритмов.

Выбирая алгоритм, который будет комбинироваться с ММГА, необходимо, чтобы он обладал следующими характеристиками:

- обладал более быстрой сходимостью алгоритма к субоптимальному решению по сравнению с ГА;
- обеспечивал возможность нахождения других оптимальных решений.

Основываясь на описанных требованиях, предлагается комбинировать ММГА и алгоритм *имитации отжига* (ИО).

Комбинированный ММГА. Алгоритм имитации отжига был описан следующими учеными S. Kirkpatrick, C. D. Gelatt и M. P. Vecchi в 1983 году [8]. Свое экзотическое название алгоритм получил в связи с методами имитационного моделирования в статистической физике [9], основанными на технике Монте-Карло. Исследование кристаллической решетки и поведения атомов при медленном остывании тела привело к появлению на свет вероятностных алгоритмов, которые оказались чрезвычайно эффективными в комбинаторной оптимизации. Алгоритм ИО относится к классу пороговых алгоритмов, общая схема которых может быть представлена следующим образом:

1^0. Выбор начального решения $i_0 \in I$ и положим $f^* = f(i_0)$, $k = 0$.

2^0. Если выполнен критерий останова, то выход из алгоритма, иначе переходим на шаг 3^0.

3^0. В окрестности точки i_k выбираем некоторое решение j.

4^0. Если $f(j) - f(i_k) < t_k$, то $i_{k+1} = j$.

5^0. Если $f^* > f(i_k)$, то $f^* = f(i_k)$.

6^0. Положить $k = k + 1$.

Различают различные алгоритмы этого класса, опишем некоторые из них:

- *Алгоритмы последовательного улучшения.* В алгоритмах такого типа принимают $t_k = 0$ на всех итерациях, этот вариант соответствует классическому локальному спуску с монотонным улучшением по *целевой функции* (ЦФ).
- *Алгоритмы порогового улучшения.* У алгоритмов этого типа допускается ухудшение по ЦФ до некоторого заданного порога, и этот порог последовательно снижается до нуля.
- *Алгоритм имитации отжига.* В этом алгоритме допускается произвольное ухудшение по ЦФ, но вероятность принятия худшего решения обратно пропорциональна величине ухудшения и может быть вычислена по следующей формуле:

$$P_{ij} = \begin{cases} 1, f(j) \le f(i); \\ e^{-\frac{f(i)-f(j)}{C_k}}, \textit{где } k = 0, 1, 2, \ldots . \end{cases}$$

Последовательность C_k играет важную роль при анализе сходимости и выбирается так, чтобы $C_k \to 0$ при $k \to \infty$. Иногда параметр C_k называют температурой [10], по аналогии с названием параметров физического процесса кристаллизации вещества из жидкого в твердое состояние, в том числе при отжиге металла.

Существует множество способов выбора последовательности C_k с целью повышения вероятности обнаружить глобальный оптимум. В некоторых работах, для которых удается полностью исследовать характеристики цепей Маркова, применяют небольшие параметры [11, 12]. В большинстве работ следуют рекомендациям, приведенным в [8], и используют схему геометрической прогрессии.

При использовании схемы геометрической последовательности для задания последовательности C_k вначале принимают $C_0 = \max \Delta f$. Понижение порога рассчитывают по следующей формуле:

$$C_{k+1} = \alpha_C \times C_k,\ k = 0, 1, 2, \ldots,$$

где α_C – положительная константа, достаточно близкая к 1, например, $\alpha_C \in [0{,}8;0{,}99]$. Конечное значение $C_k > 0$ определяется либо по числу сделанных изменений, либо как максимальное C_k, при котором алгоритм не меняет текущее решение в течение заданного числа шагов. При каждом значении C_k алгоритм, выполняет некоторое количество шагов, не меняя значение порога.

Экспериментальные исследования. Для оценки эффективности предлагаемого алгоритма было проведено сравнение следующих алгоритмов: ГА, МГА, ММГА и комбинированный алгоритм на основе алгоритмов ММГА и имитации отжига (КММГА). Для тестирования указанных алгоритмов в качестве ЦФ использовались пять функций от многих переменных, имеющие множество локальных экстремумов.

1. Функция Растригина $F_{\min} = F(0,0,\ldots,0) = 0$:

$F(x) = 10n + \sum_{i=1}^{n} | x_i^2 10Cos(2\pi x_i)$, при $-6{,}12 < x_i < 6{,}12$, $i \in 1,2,3,\ldots,n$ с шагом 0,001, где n – размерность пространства решений.

2. Функция “сфера” $F_{\min} = (F(0,0,\ldots,0) = 0$:

$F(x) = \sum_{i=0}^{n-1} \left\{ \sum_{j=0}^{j<=i} x_j \right\}^2$, при $-100 \le x_i \le 100$, $i \in 1,2,3,\ldots,n$ с шагом 0,001, где n – размерность пространства решений.

3. Функция Аклей $F_{\min} = (F(0,0,\ldots,0) = 0$:

$F(x) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=0}^{n-1} x_i^2}} - e^{\frac{1}{n}\sum_{i=0}^{n-1}\cos(2\pi x_i)} + 20 + e$ при $-30 \le x_i \le 30$, $i \in 1,2,3,\ldots,n$ с шагом 0,001, где n – размерность пространства решений.

4. Функция Гриванка (Griwank) $F_{\min} = (F(0,0,\ldots,0) = 0$:

$F(x) = \frac{1}{4000}\sum_{i=0}^{n-1}(x_i - 100)^2 - \prod_{i=0}^{n-1}\cos\left(\frac{x_i - 100}{\sqrt{i+1}}\right) + 1$, при $-600 \le x_i \le 600$, $i \in 1,2,3,\ldots,n$ с шагом 0.001, где n – размерность пространства решений.

5. Шаговая функция $F_{\min} = (F(0,0,\ldots,0) = 0$:

$\mathrm{F}(x) = 6n + \sum_{i=0}^{n-1} \lfloor x_i \rfloor$ при $-100 \le x_i \le 100$, $i \in 1,2,3,\ldots,n$ с шагом 0.001, где n – размерность пространства решений.

В экспериментах размерность пространства решений принималась равной 10, 20, 30, 40, 50, 60, 70, 80, 90 и 100 вершинам. Кодирование генов в хромосоме осуществлялось в виде целого положительного числа. Результаты проведенных экспериментов приведены в таблице.

Таблица

Результаты выполнения алгоритмов на тестовых функциях

Размерность задачи	МГА		КММГА		ММГА		ГА	
	Полученное решение	Время, с	Полученное решение	Время, с	Полученное решение	Время, с	Полученное решение	Время, с
Функция Аклей								
10	1,41	9184	1,91	6276	1,93	6586	8,76	75
20	2,04	9448	2,23	6566	2,40	6785	10,38	78
30	2,47	9715	2,99	6796	3,16	6980	11,24	81
40	2,91	9987	3,34	7055	3,73	7169	12,82	84
50	3,48	10262	3,84	7313	4,38	7371	14,11	87
60	3,76	10533	4,06	7619	4,56	7559	14,67	91
70	4,20	10809	4,38	7899	5,11	7761	15,49	94
80	4,58	11078	4,88	8177	5,39	7966	16,42	97
90	4,96	11351	4,93	8441	5,84	8165	16,74	100
100	5,14	11626	5,33	8753	6,29	8379	17,14	103
Функция Гриванка								
10	0,70	9182	0,84	6732	0,83	6238	12,37	76
20	0,56	9349	0,69	6810	0,78	6336	4,50	78
30	0,78	9522	0,87	6958	0,96	6431	6,99	80
40	0,88	9695	0,92	7114	1,02	6597	11,53	82
50	0,97	9871	1,02	7244	1,08	6754	15,98	84
60	1,03	10048	1,07	7392	1,12	6861	19,39	87
70	1,06	10229	1,12	7533	1,18	6983	23,12	88
80	1,07	10400	1,13	7683	1,23	7111	28,87	90
90	1,11	10577	1,18	7818	1,29	7215	34,63	92
100	1,11	10753	1,23	7950	1,34	7371	35,15	94
Функция Растригина								
10	123,13	9160	118,54	6341	122,52	6230	290,7	76
20	438,02	9465	389,91	6670	447,92	6451	898,827	79
30	930,62	9792	824,78	6976	986,22	6673	2018,59	83
40	1634,04	10118	1498,66	7287	1635,81	6893	2921,09	86
50	2428,26	10442	2097,12	7634	2386,17	7116	4281,84	90
60	3260,34	10758	2869,96	7936	3360,78	7336	5481,66	94
70	4198,56	11087	3649,75	8261	4336,33	7554	9872,1	97
80	5184,97	11398	4621,52	8579	5163,86	7771	8603,78	100
90	6399,37	11704	5553,14	8910	6374,20	7986	10160,1	104
100	7392,12	12027	6466,30	9195	7529,15	8217	11907,9	107

Функция Сферы								
10	21,51	9026	24,79	6101	33,59	6139	990,77	77
20	54,16	9187	34,44	6255	78,22	6255	2510,86	76
30	66,55	9359	63,71	6411	78,26	6372	4274,87	78
40	100,84	9535	143,61	6553	179,77	6499	7962,64	80
50	171,21	9715	176,61	6702	314,67	6624	11157,60	82
60	252,41	9894	295,10	6863	530,14	6752	19068,70	85
70	470,58	10075	432,12	7035	679,65	6869	25400,00	86
80	611,65	10255	428,47	7217	880,81	7000	34311,90	88
90	703,23	10433	517,63	7376	1178,28	7132	48528,30	90
100	1163,49	10617	769,82	7546	1775,55	7257	60358,30	92
Шаговая функция								
10	21,82	9045	29,80	6358	25,98	6195	899,88	74
20	44,56	9271	54,68	6524	51,50	6354	1966,28	77
30	74,74	9501	113,12	6708	103,66	6508	4161,28	79
40	129,58	9736	145,34	6911	175,94	6673	7049,60	82
50	235,80	9974	271,78	7128	403,06	6834	11521,00	85
60	345,68	10214	375,62	7349	517,08	6990	17214,80	89
70	454,50	10447	450,40	7561	666,48	7168	26890,90	90
80	511,80	10685	721,60	7795	1022,64	7329	36866,50	93
90	792,70	10920	838,08	8036	1267,56	7495	49336,00	95
100	1000,96	11158	952,26	8250	1846,06	7676	60647,60	98

В таблице серым цветом выделены лучшие результаты оптимизации тестовых функций. Анализируя данные таблицы видно, что лучшие решения были получены при помощи алгоритма МГА, но время поиска этого алгоритма является худшим. Алгоритм КММГА по качеству получаемых решений отличается от МГА в среднем на 14%, а время выполнения этого алгоритма – в среднем на 28% меньше. При сравнении эффективности поиска при помощи алгоритмов КММГА и ММГА следует отметить, что КММГА, в среднем, на 30% более эффективен, чем ММГА, а время вычислений, в среднем, у КММГА больше, чем ММГА на 5%.

Заключение. Полученные результаты показывают, что при решении задачи параметрической оптимизации более эффективным является применение КММГА, результаты которого незначительно отличаются от МГА, а время выполнения меньше. МГА также доказал свою эффективность и по качеству получаемых решений значительно превзошел ГА. Следует отметить, что результативность КММГА можно повысить за счет оптимального подбора параметров генетических операторов и параметров части алгоритма реализующей механизм имитации отжига.

Комбинирование ММГА и алгоритма имитации отжига для улучшения решения задачи параметрической оптимизации является эффективным способом повышения качества получаемых решений. При этом время поиска увеличивается всего на 5%.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Редько В.Г.* Эволюционная кибернетика. – М.: Наука 2001.
2. *МакКоннелл Дж.* Основы современных алгоритмов. – М.: Техносфера, 2004.
3. *Goldberg D.E.* Genetic Algorithms in Search, Optimization and Machine Learning / USA: Addison-Wesley Publishing Company, Inc., 1989.

4. *Potts C.I., Giddens T.D., Yadav S.B.* The Development and Evaluation of an Improved Genetic Algorithm Based on Migration and Artificial selection. IEEE Trans. on Systems, Man and Cybernetics, vol.24, No.1, Sammary 1994.
5. *Гладков Л.А.* Методы генетического поиска: Монография / Л.А. Гладков, Л.А. Зинченко, В.В. Курейчик. – Таганрог: Изд-во ТРТУ, 2002.
6. *Букатова И.Л.* Эволюционное моделирование и его приложения./И.Л. Букатова – М.: Наука, 1994.
7. *Штовба С.Д.* Муравьиные алгоритмы. Exponenta Pro. Математика в приложениях, 2003, №4. − С. 70-75.
8. *Kirkpatrick S., Gelatt C. D., Vecchi M. P.* Optimization by simulated annealing. Science. v220 (1983), pp. 671-680.
9. *Binder K.* Monte Carlo methods in statistical physics. Berlin: Springer, 1978.
10. *Aarts E. H. L., Korst J. H. M., Laarhoven van P. J. M.* Simulated annealing. in Aarts E., LenstraJ.K. (Eds) Local search in combinatorial optimization. Chichester: Wiley, 1997. pp. 91-120.
11. *Aarts E. H. L., Korst J. H. M.* Simulated annealing and Boltzmann machines. Chichester: Wiley, 1989.
12. *Hajek B., Sasaki G.* Simulated annealing: to cool it or not. Sys. Contr. Lett. v12 (1989), pp. 443-447.

УДК 519.712.2

Л.А. Гладков, Н.В. Гладкова

РЕШЕНИЕ ЗАДАЧИ КОММИВОЯЖЕРА НА ОСНОВЕ НЕЧЕТКОГО ГЕНЕТИЧЕСКОГО АЛГОРИТМА*

Нечеткий генетический алгоритм можно определить как алгоритм, сочетающий поисковые возможности генетических алгоритмов и возможности математического аппарата нечеткой логики. Математический аппарат теории нечетких систем используется в данном случае для кодирования, подбора оптимальных параметров генетических алгоритмов, значений вероятности генетических операторов, выбора функции пригодности и критерия останова, создания нечетких генетических операторов.

Одной из особенностей нечеткого генетического алгоритма является наличие нечеткого логического контроллера (НЛК). НЛК преобразует заданные параметры к нечеткому виду, затем с помощью Базы Правил (БП) определяет управляющее воздействие и возвращает скорректированные значения контрольных параметров.

В составе НЛК можно выделить следующие блоки [1]:

- база знаний, включающая в себя базу правил и базу данных;
- блок фаззификации;
- блок дефаззификации;
- система вывода решения;
- система контроля.

База правил, иногда называемая лингвистической моделью, представляет собой множество нечетких правил $R^{(k)}$, $k = 1, \dots, N$. НЛК оперирует нечеткими множествами. Поэтому конкретное значение входного параметра НЛК подлежит операции *фаззификации*, в результате которой ему будет сопоставлено нечеткое множество.

На выходе блока выработки решения формируется одно или несколько нечетких множеств с соответствующими функциями принадлежности. Встает задача

* Работа выполнена при поддержке РФФИ (грант № 08-01-00473).