

**С. ПАЛИС**, д-р техн. наук, проф. Магдебургский университет им. Отто фон Герике, Германия;  
**В. Е. ПЛЮГИН**, канд. техн. наук, доц. НТУ «ХПИ»;  
**А. М. МАСЛЕННИКОВ**, канд. техн. наук, доц. НТУ «ХПИ»

## ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ОПТИМИЗАЦИИ В ПРОЕКТИРОВАНИИ ЭЛЕКТРИЧЕСКИХ МАШИН

**Введение.** Решение задачи оптимизации заключается в нахождении наилучшего решения среди возможных рассчитанных вариантов. Несмотря на различные постановки проблемы поиска, структура оптимизационной задачи однотипна и содержит следующие компоненты [1]:

– целевая функция  $f(x)$   $n$ -мерного векторного аргумента  $x = (x_1, x_2, \dots, x_n)$ :

$$F(x) \in R^n, f(x) \in R^1; \quad (1)$$

– ограничения в виде неравенств;  
 – ограничения в виде равенств;  
 – область допустимых значений.

В общем виде задача оптимизации сводится к нахождению экстремума целевой функции.

В настоящее время для решения задач оптимизации применяются различные методы, которые в общем случае можно классифицировать на непрерывные, дискретные и целочисленные. В свою очередь, перечисленные задачи разделяют на целочисленные, одномерные и многомерные [2].

Основной проблемой применяемых алгоритмов оптимизации является поиск экстремума функции в случаях с нелинейной областью поиска и разработка методов по сокращению времени поиска и ресурсов вычислительных машин в сложных задачах.

В настоящей работе рассматривается применение генетического алгоритма оптимизации, который позволяет при минимальных затратах времени и вычислительных ресурсов получить экстремум целевой функции. В предыдущих работах была рассмотрена программная реализация генетического алгоритма [3]. Настоящая работа посвящена обоснованию применимости генетического алгоритма в оптимальном проектировании электрических машин и рассматривается впервые.

**Цель работы:** рассмотреть особенности применения классического генетического алгоритма оптимизации в проектировании электрических машин и выполнить анализ его эффективности.

### Основные положения генетических алгоритмов.

Генетические алгоритмы (ГА) – это одно из направлений исследований в области искусственного интеллекта, занимающееся созданием упрощенных моделей эволюции живых организмов для решения задач оптимизации [4].

При описании ГА используют определения, заимствованные из генетики. Например, говоря о популяции особей, в качестве базовых понятий применяются ген, хромосома, генотип, фенотип, аллель (рис. 1).

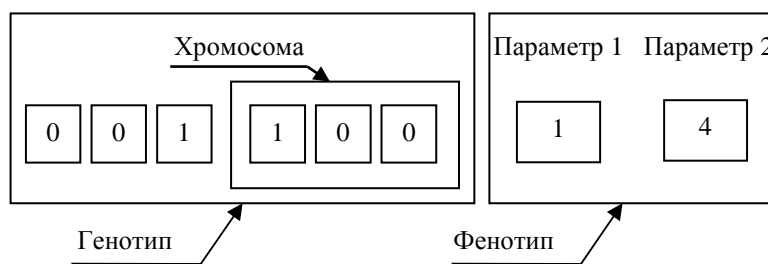


Рисунок 1 – Генотип и фенотип в ГА

В приведенном примере на рисунке число «001» первой хромосомы в двоичной системе исчисления соответствует «1» «параметра 1» в десятичной и число «100» второй хромосомы в двоичной – «4» «параметра 2» в десятичной.

Этим терминам соответствуют определения из технической лексики, в частности, «цепь», «двоичная последовательность», «структура».

Классический ГА состоит из следующих шагов:

- 1) инициализация, или выбор исходной популяции хромосом;
- 2) оценка приспособленности хромосом в популяции – расчет функции приспособленности для каждой хромосомы;

© С. Палис, В.Е. Плюгин, А.М. Масленников, 2015

- 3) проверка условия остановки алгоритма;
  - 4) селекция хромосом – выбор тех хромосом, которые будут участвовать в создании потомков для следующей популяции;
  - 5) применение генетических операторов – мутации и скрещивания;
  - 6) формирование новой популяции;
  - 7) выбор «наилучшей» хромосомы.
- Блок-схема ГА изображена на рис. 2.

Простой ГА случайным образом генерирует начальную популяцию. Работа ГА представляет собой итерационный процесс, который продолжается до тех пор, пока не выполняется заданное число поколений или какой-либо иной критерий остановки. На каждом поколении реализуется пропорциональный отбор по приспособленности, скрещивание и мутация. Хромосомы, полученные в результате применения генетических операторов к хромосомам временной родительской популяции, включаются в состав новой популяции.

Она становится так называемой текущей популяцией для данной итерации генетического алгоритма (рис. 3).

#### Практическая реализация генетического алгоритма в оптимизации ЭМ.

Порядок оптимизации ЭМ при использовании классического ГА выглядит следующим образом:

- 1) задаем диапазон варьируемых переменных;
- 2) задаем ограничения;

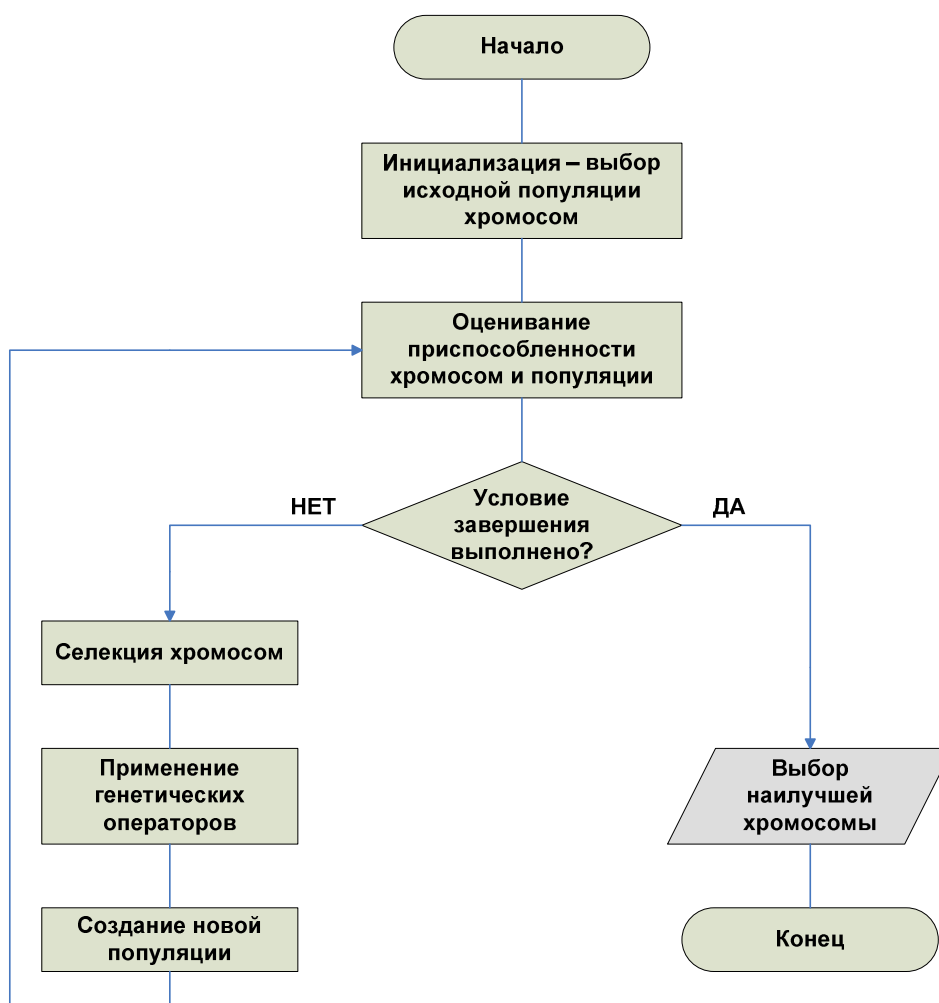


Рисунок 2 – Классический генетический алгоритм

- 3) выбираем критерий оптимальности;
- 4) вызываем функцию ГА оптимизации и получаем оптимальный набор варьируемых переменных;
- 5) для найденного набора вызываем функцию автоматического расчета электродвигателя.

На каждой очередной итерации рассчитываются значения функции приспособленности для всех хромосом этой популяции, после чего проверяется условие остановки алгоритма и либо фиксируется результат в виде хромосомы с наибольшим значением функции приспособленности, либо осуществляется переход к следующему шагу генетического алгоритма, т.е. к селекции.

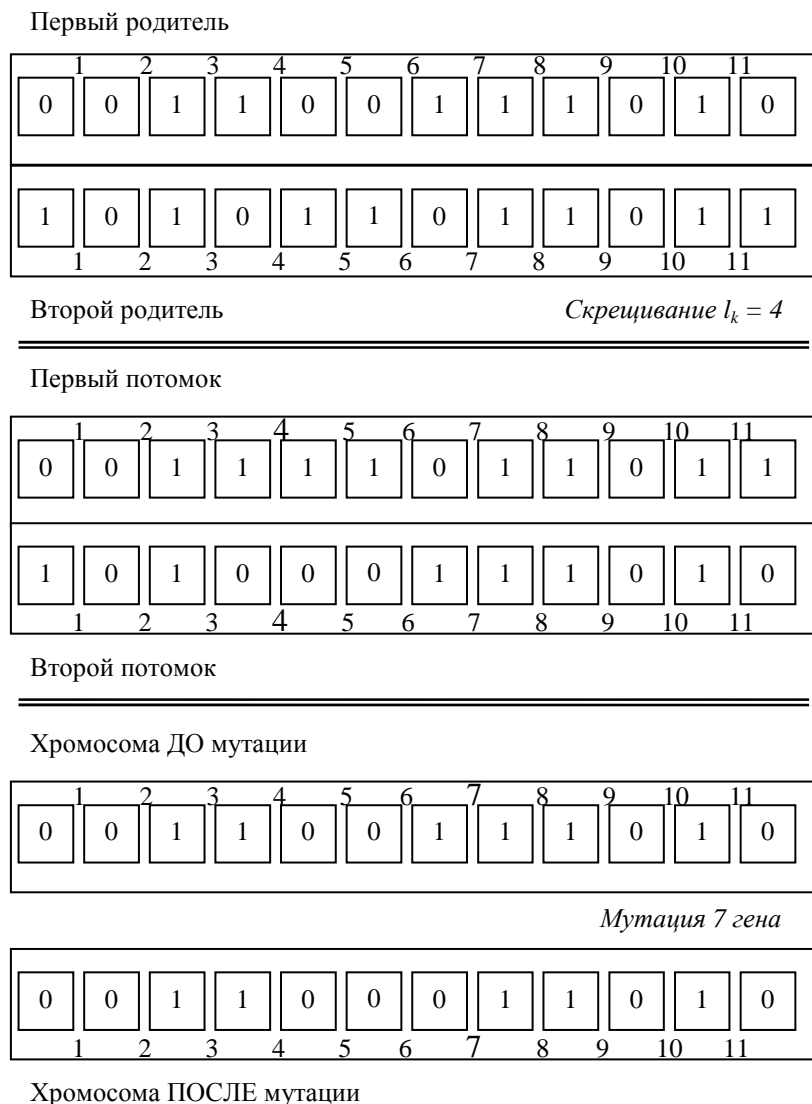


Рисунок 3 – Пример реализации скрещивания в генетическом алгоритме

Рассмотрим на примере оптимизации асинхронного двигателя программную реализацию генетического алгоритма на языке Java в среде NetBeans. Для решения поставленной задачи воспользуемся свободно распространяемой Java-библиотекой EvoJ (англ. Evolution Java) [5].

Проект EvoJ задуман как расширяемый каркас классов Java для решения всевозможных задач оптимизации с помощью эволюционных (генетических) алгоритмов. Для использования EvoJ программист должен имплементировать лишь один простой интерфейс, состоящий из одного метода. Все остальные шаги алгоритма берет на себя EvoJ.

Чтобы решить задачу при помощи EvoJ надо:

- 1) создать интерфейс с переменными;
- 2) имплементировать интерфейс фитнес-функции;
- 3) создать популяцию решений и осуществить над ними нужное количество итераций ГА, используя код, приведенный выше.

В примере будет рассмотрено две варьируемых переменных: внутренний диаметр пакета статора и длина пакета статора [6].

EvoJ позволяет изменять переменные без задания диапазона изменения переменных. Однако, если нужно имплементировать собственную стратегию мутации, то декларировать сеттеры придется - иначе не получится изменять переменные.

Если решение не устраивает, можно продолжить итерации ГА (увеличивая число популяций и итераций), пока не будет достигнуто желаемое качество решения. Результаты выполнения ГА при выборе максимального КПД в качестве критерия оптимальности показаны на рис. 4.

Как видно из рис. 4, в оптимальной машине КПД выше, чем у базовой, а остальные параметры не выходят за допустимые пределы.

Наименование	Ед.из...	Базовое знач...	Оптимальное
Индукция в возд. зазоре	Тл	0.748	0.807
Внутр. диаметр сердечника ст...	мм	185.0	194.0
Длина сердечника статора	мм	130.0	115.0
Величина $\lambda = L/d\tau$	-	0.895	0.755
Высота паза статора	мм	21.9	14.6
Высота паза ротора	мм	32.2	33.2
Ширина верха паза статора	мм	7.7	7.8
Ширина дна паза статора	мм	10.2	9.3
Верхний диаметр паза ротора	мм	7.9	7.8
Диаметр дна паза ротора	мм	3.7	3.4
КГД	-	0.885	0.891
$\cos \Phi$	-	0.893	0.9
$I_p/I_n$	-	5.84	6.52
$M_p/M_n$	-	1.4	1.62
$M_{\max}/M_n$	-	2.65	2.88
Плотность тока в обмотке стат...	А/мм....	5.912	5.912
Плотность тока в обмотке рот...	А/мм....	2.503	2.5
Превышение темп. обмотки ст...	град. С	93.25	95.69

Рисунок 4 – Таблица параметров АД с КЗР при оптимизации с применением ГА

**Выводы** Рассмотренный ранее алгоритм декартова произведения множеств (ДПМ) [7], в сравнении с ГА, позволяет выполнять многокритериальную оптимизацию, что является его несомненным плюсом. Кроме того, ДПМ всегда (!) дает однозначный единственный наилучший вариант среди существующих. Однако в ДПМ, при варьировании всего лишь двух переменных с отклонением  $\pm 20\%$  от базового значения время расчетов приближается к 48 мин.

Применение ГА при тех же варьируемых параметрах и отклонении  $\pm 100\%$  (!) относительно базового значения, время расчета составило всего лишь 40 с! Однако, ГА (в классической постановке), не позволяет заполнить многокритериальную оптимизацию.

В ГА количество варьируемых параметров и диапазон их изменения не принципиален с точки зрения производительности, т.к. набор варьируемых переменных создается динамически, а не заранее, как в методе сочетаний ДПМ. Кроме того, все комбинации с переменными и значениями целевой функции осуществляются в бинарном виде. Однако, время работы ГА весьма критично к количеству создаваемых популяций и числу итераций в популяциях.

Выбор числа популяций и итераций осуществляется опытным путем и увеличивается до тех пор, пока не будет получен приемлемый результат. Результат оптимизации с применением ГА всегда будет наилучшим по выбранному критерию, но нет гарантии, что не существует варианта еще лучше, чем полученный. Собственно, в этом и состоит логика генетического отбора – мы получаем результат, *приближающийся* к наилучшему среди порожденных случайным образом популяциях. Чем их больше, тем результат получается более достоверным. При этом степень достоверности можно оценить по тому, насколько велик разброс получаемых результатов повторных расчетов при данном количестве популяций.

Производительность ГА определяется не числом и диапазоном варьируемых переменных, а лишь размером популяций эффективных вариантов, а результат оптимизации неоднозначен и приближен к наилучшему. Если нужно произвести оценочные расчеты в максимально сжатые сроки, а качество получаемых результатов вписывается в допустимую погрешность, то применение ГА оптимизации может стать для проектировщика незаменимым инструментом.

**Список Литературы:** 1. Кочегурова Е.А. Теория и методы оптимизации / Е.А. Кочегурова. – Томск: Изд-во Томского политехнического университета, 2012. – 157 с. 2. Реклейтис Г. Оптимизация в технике / Г. Реклейтис, А. Рейвиндан, К. Рэгсдел. – М.: Мир, 1986. – 351с. 3. Zablodskij N. Induction Motor Design by Use of Genetic Optimization Algorithms / N. Zablodskij, V. Pliugin, J. Lettl, K. Buhr, S. Khomitskiy // “Transactions on electrical engineering”. – 2013. – No. 3. – P. 65 – 69. 4. Емельянов В.В. Теория и практика эволюционного моделирования / В.В. Емельянов, В.В. Курейчик, В.М. Курейчик. – М.: Физматлит, 2003. – 432 с. 5. EvoJ - Evolutionary computations framework [Электронный ресурс] / Inprise Corporation, 2002. – 12 р. – Режим доступа к книге: <http://evoj-firmw.appspot.com>. 6. Заблюдський М.М. САПР електромеханічних пристроїв: навчальний посібник / М.М. Заблюдський, В.Є. Пліугін, К. Бур. – Алчевськ, Ладос, 2013. – Ч. 2. – 320 с. 7. Zablodskij N. Induction Motor Optimal Design by Use of Cartesian Product / N. Zablodskij, V. Pliugin, J. Lettl, K. Buhr // “Transactions on electrical engineering”, 2013. – No. 2. – P. 54 - 58.

**Bibliography (transliterated)** 1. Kochegurova E.A. Teoriya i metody optimizatsii / E.A. Kochegurova. – Tomsk: Izd-vo Tomskogo politeh-nicheskogo universiteta, 2012. – 157 s. 2. Rekleytis G. Optimizatsiya v tehnike / G. Rekleytis, A. Reyvindan, K. Regsdel. – M.: Mir, 1986. – 351s. 3. Zablodskij N. Induction Motor Design by Use of Genetic Optimization Algorithms / N. Zablodskij, V. Pliugin, J. Lettl, K. Buhr, S. Khomitskiy // “Transactions on electrical engineering”. – 2013. – No. 3. – P. 65 – 69. 4. Emelyanov V.V. Teoriya i praktika evolyutsionnogo modelirovaniya / V.V. Emelyanov, V.V. Kureychik, V.M. Kureychik. – M.: Fizmatlit, 2003. – 432 s. 5. EvoJ - Evolutionary computations framework [Elektronnyiy resurs] / Inprise Corporation, 2002. – 12 p. – Rezhim dostupa k knige: <http://evoj-firmw.appspot.com>. 6. Zablodskiy M.M. SAPR elektromehanichnih pristroyiv: navchalniy posibnik / M.M. Zablodskiy, V.E. Plyugin, K. Bur. – Alchevsk, Lado, 2013. – Ch. 2. – 320 s. 7. Zablodskij N. Induction Motor Optimal Design by Use of Cartesian Product / N. Zablodskij, V. Pliugin, J. Lettl, K. Buhr // “Transactions on electrical engineering”, 2013. – No. 2. – P. 54 - 58.

Поступила (received) 26.08.2015