УДК 519.68

ВЕРОЯТНОСТНЫЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ РЕШЕНИЯ СЛОЖНЫХ ЗАДАЧ МНОГОКРИТЕРИАЛЬНОЙ ОПТИМИЗАЦИИ С АДАПТИВНОЙ МУТАЦИЕЙ И ПРОГНОЗОМ МНОЖЕСТВА ПАРЕТО

© 2011 Е. А. Сопов, С. А. Сопов

Сибирский государственный аэрокосмический университет

В статье представлен новый алгоритм решения сложных задач условной многокритериальной оптимизации, построенный на базе вероятностного генетического алгоритма. Предложена эффективная модификация алгоритма с адаптивной мутацией и прогнозом множества Парето, позволяющая получать решения задач при меньшем использовании вычислительного ресурса. Представлены результаты исследования эффективности подхода в сравнении со стандартным генетическим алгоритмом.

Многокритериальная оптимизация, эволюционные алгоритмы, вероятностный генетический алгоритм, адаптивная мутация.

Введение

Необходимость в разработке моделей сложных систем возникает в различных областях науки и техники: математика, экономика, медицина, управление космическими аппаратами и др. При разработке моделей часто возникают задачи оптимизации, которые обладают такими свойствами, как многоэкстремальность, алгоритмическое задание функций, сложная конфигурация допустимой области, наличие нескольких типов переменных и т.д. Более того, в реальных задачах решение редко оценивается по единственному критерию, поэтому важно не только найти допустимое парето-оптимальное решение, но и аппроксимировать множество, чтобы предложить лицу, принимающему решение (ЛПР) объективный выбор. Такие задачи не решаются с помощью классических процедур оптимизации, что приводит к необходимости разрабатывать и применять более эффективные и универсальные методы. К таким методам относятся, в частности, эволюционные алгоритмы (ЭА), доказавшие свою эффективность при решении многих сложных задач [1, 2].

Эффективность ЭА определяется тщательной настройкой и контролем их параметров, что затрудняет применение ЭА, особенно неподготовленными пользователями. В последние годы наблюдается тенденция развития эволюционных подходов, способных к самонастройке параметров за счёт использования сложных гибридных схем, а также эффективных алгоритмов с меньшим числом параметров.

Одним из подходов, позволяющих уменьшить число настраиваемых параметров эволюционных алгоритмов, является использование вероятностных генетических алгоритмов (ВГА) [3, 4]. Их отличие от стандартного генетического алгоритма (ГА), в частности, состоит в том, что в них отсутствует оператор скрещивания, а новые решения получаются на основе статистической информации о поисковом пространстве. Таким образом, накапливая и используя эту информацию, данные алгоритмы самостоятельно могут адаптироваться к решаемой задаче. ВГА показали свою эффективность и применимость к некоторым сложным задачам оптимизации [5, 6] и являются перспективными для дальнейшего изучения и совершенствования. Данная работа посвящена исследованию эффективности ВГА в сложных задачах многокритериальной оптимизации.

1. Многокритериальная оптимизация генетическими алгоритмами

В самом общем виде задача условной многокритериальной оптимизации включает набор из N параметров (переменных), множество K целевых функций этих переменных и множество M ограничений типа равенств и неравенств. При решении многокритериальной задачи необходимо найти оптимум по совокупности K критериев, а сама задача формально записывается следующим образом:

$$y = f(x) = (f_1(x), f_2(x), ..., f_K(x)) \to opt,$$

$$\begin{cases} g_j(x) \le 0, j = \overline{1,r}, \\ h_j(x) = 0, j = \overline{r+1,M}, \end{cases}$$

где $x = (x_1, x_2, ..., x_N) \in X$ – вектор решений, $y = (f_1(x), f_2(x), ..., f_K(x)) \in Y_f$ – вектор целевых функций. При этом X называется пространством решений (или альтернатив), а Y_f – пространством целей (или критериальным пространством).

В наиболее общей постановке задачи условной многокритериальной оптимизации от функций $f_i(x)$, $g_j(x)$, $h_j(x)$ не требуется никаких дополнительных свойств, удобных с точки зрения оптимизации (выпуклость, дифференцируемость, непрерывность и т.д.). Функции и ограничения могут быть заданы алгоритмически, а переменные могут быть непрерывными, дискретными, бинарными или смешанными.

Введём следующие определения.

Допустимое множество D определяется как множество векторов $x \in X$, которые удовлетворяют ограничениям $g_j(x)$ и $h_j(x)$:

$$D = \{x \in X \mid g_j(x) \le 0,$$
$$j = \overline{1,r}; h_j(x) = 0, j = \overline{r+1,M}\}.$$

Образ допустимой области D в пространстве целей обозначается как $Y_D = \{y = f(x) \mid x \in D\}$.

Будем считать, что решается задача максимизации. Тогда понятие ***Парето-доминирования*** для двух любых векторов определяется тремя возможными вариантами:

- решение a доминирует решение b: $a \succ b$, если $f(a) > f(b)$;

- решение a слабо доминирует решение b: $a \succeq b$, если $f(a) \ge f(b)$;

- решения a и b несравнимы: $a \approx b$, если $f(a) \not\le f(b) \wedge f(a) \not\ge f(b)$.

Таким образом, $a \in D$ называется ***парето-оптимальным***, если не существует $b \in D$: $b \succ a$. При этом вектор $x \in D$, называется ***недоминируемым*** относительно неко-торого множества $A \subseteq D$, если не существует $a \in A$: $a \succ x$.

Определим оператор $p(A)$, который даёт множество недоминируемых решений в A:

$$p(A) = \{a \in A \subseteq D \mid a\} - \textit{недоминируем относительно } A\}.$$

Множество $X_p = p(D)$ называется ***множеством Парето***, а множество $Y_p = f(X_p)$ определяет ***фронт Парето***.

Для решения задач многокритериальной оптимизации генетические алгоритмы должны быть оснащены методами учета многих целевых функций. В многокритериальных ГА за основу берется стандартный ГА, однако при разработке конкретных методов решения многокритериальных задач упор делается на модификацию этапов назначения пригодности и селекции с поддержанием разнообразия популяции.

Наиболее распространенными многокритериальными схемами сегодня являются: VEGA (генетический алгоритм векторных оценок) [7], FFGA (генетический алгоритм Фонсеки и Флеминга) [8], NPGA (паретовский генетический алгоритм с нишами) [9], SPEA (усиленный по Парето эволюционный алгоритм) [10].

Метод VEGA, который впервые был предложен в 1984 году Шаффером [7], относится к категории методов селекции по переключающимся целевым функциям. Здесь селекция производится по пригодности индивидов для каждого из K критериев в отдельности. Таким образом, промежуточная популяция заполняется равными порциями индивидов, отобранных по каждому из частных критериев.

Метод FFGA (1993) использует основанную на Парето-доминировании процедуру ранжирования индивидов, где ранг каждого индивида определяется числом доминирующих его индивидов [8]. Метод FFGA реализует лишь схему назначения пригодности, а для отбора индивидов в следующее поколение используется процедура селекции ГА.

Метод NPGA принципиально отличается от двух предыдущих, так как в нём заложен механизм поддержания разнообразия [9]. Метод NPGA представляет собой комбинацию турнирной селекции и концепции доминирования по Парето. В NPGA этап назначения пригодности заменяется модифицированной схемой деления пригодности с использованием понятия ниши, которая определяется для индивидов в пространстве целевых функций и обеспечивает возможность поддержания разнообразия, позволяя получить представительное множество Парето.

Метод SPEA (Zitzler and Thiele, 1998) кардинально отличается от рассмотренных ранее методов, так как в нем [10]:

- для назначения индивидам скалярного значения пригодности используется концепция Парето-доминирования;

- индивиды, недоминируемые относительно других членов популяции, хранятся отдельно в специальном внешнем множестве;

- для уменьшения количества индивидов, хранящихся во внешнем множестве, выполняется кластеризация, что, в свою очередь, никак не влияет на приобретенные в процессе поиска свойства индивидов.

Уникальность и преимущества метода SPEA заключаются в том, что:

- он сочетает вышеперечисленные подходы в одном алгоритме;

- пригодность каждого индивида популяции определяется только относительно индивидов внешнего множества, независимо от того, доминируют ли индивиды популяции друг друга;

- несмотря на то, что «лучшие» индивиды, полученные в предыдущих поколениях, хранятся во внешнем множестве, все они принимают участие в селекции;

- для предотвращения преждевременной сходимости используется особый механизм образования ниш, где деление общей пригодности осуществляется не в смысле расстояния между индивидами, а на основе Парето-доминирования.

Одним из недостатков SPEA может быть признано то, что большая часть ресурсов и времени тратится на процедуру кластеризации, которая обеспечивает поддержание разнообразие популяции.

2. Вероятностный генетический алгоритм многокритериальной оптимизации

Как отмечалось выше, в многокритериальных схемах ГА в основе лежит стандартный ГА, а многокритериальность реализуется за счёт модификации процедур селекции и назначения пригодности. В [3, 4] был предложен оригинальный ГА, в котором накопленная информация о пространстве поиска фиксируется в виде оценки распределения единичных компонентов векторов-решений. Это распределение используется для формирования новых индивидов (заменяет оператор скрещивания) и переоценивается после отбора индивидов. ВГА содержит меньшее число настраиваемых параметров, а за счёт использования дополнительной статистической информации о пространстве поиска – в среднем превосходит стандартный ГА по надёжности и трудоёмкости.

Общая схема ВГА имеет следующий вид:

1. Инициализировать случайным образом популяцию решений.

2. С помощью оператора селекции выбрать r наиболее пригодных индивидов текущей популяции (родителей). Вычислить вектор вероятностей по формуле:

$$\overline{P} = (p_1, \ldots, p_n),$$

$$p_j = P\{x_j = 1\} = \frac{1}{r}\sum_{i=1}^{r} x_j^i,\ j = \overline{1,n},$$

где n – длина хромосомы, x_j^i – j-й бит i-го индивида.

3. В соответствии с распределением $\overline{P}$ сформировать популяцию потомков.

4. Из популяции родителей и потомков сформировать новую рабочую популяцию.

5. Новую популяцию подвергнуть мутации.

6. Повторять шаги 2 – 5 пока не выполнится условие остановки.

ВГА с адаптивной мутацией. При решении задач многокритериальной оптимизации основной упор делается на получение репрезентативного множества ответов. Это означает, что решения, полученные алгоритмом, должны быть достаточно широко распределены в поисковом пространстве и давать реальное представление об истинном множестве Парето.

В эволюционных алгоритмах оператор мутации отвечает за «широкий» поиск и захват новых областей поискового пространства. Для его настройки пользователю дополнительно приходится принимать решение о том, какое значение вероятности мутации следует использовать. При низкой мутации алгоритм будет редко захватывать новые области поискового пространства и быстрее сходиться к локальным экстремумам. При высокой мутации алгоритм будет чаще обследовать различные области поискового пространства, но при этом не будет локализовать перспективные области пространства поиска. Таким образом, выбор оператора мутации напрямую влияет на качество работы алгоритма и получаемых им решений.

В данной работе предложена модификация ВГА, в котором отсутствует необходимость настраивать оператор мутации вручную (впервые подобная схема была предложена в [11]). На каждом шаге работы алгоритма вероятность мутации рассчитывается автоматически по следующей формуле:

$$P_m = \frac{S}{n},$$

$$S = \begin{cases} \dfrac{n}{2}, (X_k = 0) \vee \left(\dfrac{X_{k-1}}{2 \cdot X_k} \geq \dfrac{n}{2} \right) \\ \dfrac{X_{k-1}}{2 \cdot X_k}, \text{иначе}, \end{cases}$$

где n – длина хромосомы; X_{k-1} – разброс точек в пространстве решений на (k-1)-м поколении; X_k – разброс точек в пространстве решений на k-м поколении; функция S определяет силу мутации.

При этом разброс точек в пространстве решений определяется по формуле

$$X = \frac{1}{\overline{N}} \sum_{n=1}^{\overline{N}} \frac{\Delta d^n}{M_n - m_n},$$

где $\Delta d^n = \frac{1}{N'} \sum_{i,j}^{N'} \left\| d_{ij}^n - d^n \right\|$ – среднее отклонение расстояний между индивидами i и j от среднего расстояния d^n в пространстве решений по каждой из n переменных; $d^n = \frac{1}{N'} \sum_{i,j}^{N'} d_{ij}^n$ – среднее расстояние между индивидами i и j по каждой из n переменных в пространстве решений; $d_{ij}^n = \left\| d_i^n - d_j^n \right\|$ – расстояние между индивидами i и j по каждой из n переменных в пространстве решений; i, j – номера индивидов в популяции; N' – количество всех возможных пар индивидов; $\overline{N}$ – количество переменных (альтернатив) в решаемой задаче; M_n – максимально возможное значение n-й переменной во всей допустимой области; m_n – минимально возможное значение n-й переменной во всей допустимой области.

Таким образом, вероятность мутации увеличивается, если решения-индивиды начинают группироваться около какого-либо локального оптимума, и уменьшается, когда решения распределяются в пространстве более широко.

ВГА с прогнозом множества Парето. Изменение значений вектора вероятностей единичных компонентов в ВГА можно представить в виде графиков [3, 4]. При этом можно наблюдать следующее свойство:

$$\begin{cases} p_i \to 1, \text{если } x_i^{opt} = 1, \\ p_i \to 0, \text{если } x_i^{opt} = 0, \end{cases}$$

где x_i^{opt} - значение i-го бита оптимального решения. Т. е. если значения вектора вероятностей стремятся к единице, то вероятнее всего соответствующее значение в векторе оптимального решения равно единице и если стремятся к нулю, то – ноль.

Последнее свойство можно использовать для прогноза оптимального значения и, в частности, для прогнозирования множества Парето в задачах многокритериальной оптимизации. В [4] предложен алгоритм прогноза для задач безусловной однокритериальной оптимизации, однако данный алгоритм требует модификации, т.к. в исходной постановке алгоритм прогнозирует единственное решение, а в многокритериальной постановке множество Парето в общем случае может состоять более чем из одного решения. Далее приведен модифицированный алгоритм прогнозирования множества Парето:

1. Определить шаг прогноза K.

2. На каждом поколении рассчитывать распределение нулей и единиц во множестве недоминируемых относительно текущей популяции решений, т.е. рассчитать вектор $\overline{P}$ по формуле:

$$\overline{P} = \{p_j\},\ p_j = P\{x_j = 1\} = \frac{1}{r}\sum_{i=1}^{r} x_j^i,\ j = \overline{1,n},$$

где n – длина хромосомы, x_j^i – j-й бит i-го индивида во множестве недоминируемых решений, r – количество недоминируемых решений в текущей популяции.

3. Каждому поколению придавать вес в зависимости от его номера:

$$\sigma_i = 2i/N_K(N_K+1),\ i = 1,\ldots,N_K.$$

4. Через каждые K поколений по набранным статистикам $\overline{P_i}$, $i = \overline{1,N_K}$, $N_K = t \cdot K$, $t \in \{1,2,\ldots\}$ рассчитать вектор вероятностей по формуле:

$$\widetilde{P} = \sum_{i=1}^{N_K} \sigma_i \cdot \overline{P_i}.$$

5. Используя вектор $\widetilde{P}$, сгенерировать заданное количество прогнозируемых решений.

6. Добавить полученные на предыдущем шаге решения в текущую рабочую популяцию и продолжить работу ВГА.

Выбор весовых коэффициентов позволяет увеличивать вклад в прогноз более поздних значений вероятностей, т. к. на более поздних поколениях накоплено больше информации о поисковом пространстве.

3. Анализ эффективности алгоритмов

В качестве исследуемых характеристик эффективности алгоритмов были выбраны: разброс точек в пространстве решений (*X*), разброс точек в пространстве критериев (*Y*), процент паретовских (*%*) и допустимых (*% доп.*) точек в итоговой популяции решений. В тех задачах, где множество Парето представляет собой отрезок прямой или части окружности, дополнительно исследовалось среднее расстояние *(Ср.р.)* от всех решений до множества Парето. Для оценки скорости сходимости алгоритмов с прогнозом дополнительно исследовался номер итерации алгоритма, на которой процент паретовских точек в текущей популяции впервые достигает определённого значения (*Скор.*). Ниже представлены способы расчёта характеристик.

Разброс точек в пространстве решений (*X*) определяется по следующей формуле (аналогично разбросу точек при расчете значения адаптивной мутации):

$$X = \frac{1}{\overline{N}}\sum_{n=1}^{\overline{N}} \frac{\Delta d^n}{M_n - m_n}.$$

Заметим, что $X \in [0,\ 1]$. Частный случай $X = 0$ означает, что все индивиды в популяции сгруппировались в одну точку. Чем больше значение данной характеристики, тем более равномерно и широко распределены решения в пространстве альтернатив. Алгоритм, у которого результирующее множество ответов имеет больший показатель разброса, будем считать более предпочтительным.

Разброс точек в пространстве критериев (*Y*) определяется по следующей формуле:

$$Y = \frac{1}{K}\sum_{k=1}^{K} \frac{\Delta f^k}{M_k - m_k},$$

где $\Delta f^k = \frac{1}{N'}\sum_{i,j}^{N'} \left\| f_{ij}^k - f^k \right\|$ – среднее отклонение расстояний между индивидами i и j от среднего расстояния f^k в пространстве критериев по каждой из K целевых функций; $f^k = \frac{1}{N'}\sum_{i,j}^{N'} f_{ij}^k$ – среднее расстояние между индивидами i и j по каждой из K целевых функций в критериальном пространстве; $f_{ij}^k = \left\| f_i^k - f_j^k \right\|$ – расстояние между индивидами i и j по каждой из K целевых функций в критериальном пространстве; i, j – номера индивидов популяции; N' – количество всех возможных пар индивидов; K – количество критериев (целевых функций) в решаемой задаче; M_k – максимально возможное значение по k-й целевой функции: $M_k = \max\limits_{x \in D}\left(f^k(x)\right)$; m_k – минимальное возможное значение по k-й целевой функции: $m_k = \min\limits_{x \in D}\left(f^k(x)\right)$.

Процент паретовских точек (%) определяется по следующей формуле:

$$Percent_{Par} = 100\frac{N_P(P_{res})}{N(P_{res})},$$

где P_{res} – результирующее множество решений, выдаваемое алгоритмом в качестве ответа, $N_P(P_{res})$ – количество паретовских точек во множестве P_{res}, $N(P_{res})$ – количество всех точек во множестве P_{res}.

Очевидно, что $Percent_{Par} \in [0,100]$. Чем выше значение данного показателя, тем больше паретовских точек в итоговом множестве решений. При сравнении двух алгоритмов предпочтение будем отдавать тому, у которого данный показатель выше.

Процент допустимых точек (*% доп.*) определяется по следующей формуле:

$$Percent_{Feas} = 100\frac{N_{Feas}(P_{res})}{N(P_{res})},$$

где P_{res} – результирующее множество решений, выдаваемое алгоритмом в качестве ответа; $N_{Feas}(P_{res})$ – количество допустимых точек во множестве P_{res}; $N(P_{res})$ – количество всех точек во множестве P_{res}.

Среднее расстояние до множества Парето (*Ср.р.*) определяется по формуле

$$\Delta d = \frac{1}{N}\sum_{i=1}^{N} d_i,$$

где N – количество индивидов (решений) в итоговом множестве решений, d_i – кратчайшее расстояние в пространстве альтернатив (X) от i-го индивида до множества Парето рассматриваемой задачи.

В тех случаях, когда множество Парето является отрезком прямой, d_i – есть кратчайшее расстояние от точки до отрезка, вычисляемое по следующей формуле:

$$d(X) = \sqrt{\sum_{i=1}^{N}\left(a_i - x_i - \lambda(b_i - a_i)\right)^2},$$

$$\lambda = \frac{1}{\sum\limits_{i=1}^{N}\left(b_i - a_i\right)^2}\sum_{i=1}^{N}\left(b_i - a_i\right)\left(x_i - a_i\right),$$

где $A=(a_1,\ldots, a_N)$, $B=(b_1,\ldots, b_N)$ – координаты концов отрезка AB; $X=(x_1,\ldots, x_N)$ – точка, от которой требуется найти расстояние до отрезка AB; N – количество переменных в решаемой задаче.

Очевидно, что чем меньше значение данного показателя, тем ближе к множеству Парето лежит результирующее множество решений. При сравнении двух алгоритмов предпочтение будем отдавать тому, у которого данный показатель меньше.

Скорость сходимости *(Скор.)* определяется следующей процедурой оценивания скорости сходимости алгоритма к множеству Парето:

1) Перед началом работы ГА положить $s = 0$.

2) На k-м поколении ГА вычислить процент паретовских точек P_{par} в текущей популяции.

3) Если $P_d < P_{\min}$ и $s = 0$, то продолжить работу ГА и перейти на шаг 2.

4) Если $P_d \ge P_{\min}$ и $s = 0$, то вычислить значение s по формуле $s = 100\frac{k}{N_P}$ и в качестве оценки скорости сходимости выдать

полученное значение s. Завершить данную процедуру оценки в рамках текущего запуска ГА и продолжить дальнейшую работу ГА.

Здесь k – номер текущего рабочего поколения, N_P – общее количество популяций в ГА, P_{min} – минимальный процент паретовских точек, который должен быть достигнут алгоритмом (задается пользователем). В данной работе использовалось значение $P_{min} = 50$.

Для задач, в которых множество Парето представляет собой отрезок прямой или части окружности, процент паретовских точек на шаге 2 может определить процент точек, лежащих на заданном расстоянии от отрезка или части окружности.

Очевидно, что значение $s \in [0,100]$. Интерпретация данного параметра следующая: требуется s процентов поколений ГА для того, чтобы текущая рабочая популяция стала содержать как минимум P_{min} процентов паретовских точек (или как минимум P_{min} точек, лежащих на заданном расстоянии от множества Парето). Значение $s = 0$ говорит о том, что заданный порог P_{min} не достигается в ходе работы ГА. Значение $s = 100$ говорит о том, что заданный порог P_{min} достигается на самом последнем поколении ГА. При сравнении двух алгоритмов предпочтение будем отдавать тому, у которого данный показатель меньше и отличен от 0.

Сравнение эффективности стандартного ГА и ВГА проводилось на тестовых задачах условной двух-, трёх- и четырёхкритериальной нелинейной оптимизации. Тестовые задачи представлены ниже:

Задача 1

$$f_1(x,y) = (x-6)^2 + (y-4)^2 \to \min,$$
$$f_2(x,y) = (x+2)^2 + (y-5)^2 \to \min,$$

$$\begin{cases} g_1(x,y) = (x-1)^2 + (y-4)^2 \le 4 \\ g_2(x,y) = (x-3)^2 + (y-4)^2 \le 6,25. \end{cases}$$

Задача 2

$$f_1(x,y) = (x-6)^2 + (y-4)^2 \to \min,$$
$$f_2(x,y) = (x+2)^2 + (y-5)^2 \to \min,$$
$$f_3(x,y) = (x-4)^2 + (y+4)^2 \to \min,$$

$$\begin{cases} g_1(x,y) = (x-2)^2 + (y-2)^2 \le 10,24 \\ g_2(x,y) = (x-5)^2 + (y-1)^2 \le 16 \\ g_3(x,y) = (x-3)^2 + (y-4)^2 \le 9. \end{cases}$$

Задача 3

$$f_1(x,y) = (x-1)^2 + (y+1)^2 \to \min,$$
$$f_2(x,y) = (x+2)^2 + (y-2)^2 \to \min,$$
$$f_3(x,y) = (x-3)^2 + (y-4)^2 \to \min,$$
$$f_4(x,y) = (x-4)^2 + (y-2)^2 \to \min,$$

$$\begin{cases} g_1(x,y) = (x+1,8)^2 + (y-2)^2 \ge 4 \\ g_2(x,y) = (x-1)^2 + (y-4,5)^2 \ge 4 \\ g_3(x,y) = (x-5,2)^2 + (y-2)^2 \ge 9 \\ g_4(x,y) = (x-1)^2 + (y+2)^2 \ge 9 \\ g_5(x,y) = (x-1)^2 + (y-2)^2 \le 6,25. \end{cases}$$

Задача 4

$$f_1(x,y) = (x-1)^2 + (y+1)^2 \to \min,$$
$$f_2(x,y) = (x+2)^2 + (y-2)^2 \to \min,$$
$$f_3(x,y) = (x-3)^2 + (y-4)^2 \to \min,$$
$$f_4(x,y) = (x-4)^2 + (y-2)^2 \to \min,$$

$$\begin{cases} g_1(x,y) = x^2 + (y-6)^2 \le 4 \\ g_2(x,y) = (x+2)^2 + (y-5)^2 \le 4. \end{cases}$$

Поскольку алгоритмы являются стохастическими, то для каждой задачи проводилась серия из 100 независимых запусков алгоритмов, и исследуемые характеристики усреднялись.

Ниже в таблицах представлены результаты экспериментального исследования алгоритмов (лучшие показатели для конкретной задачи выделены).

Из табл. 1 видно, что в среднем метод SPEA более эффективен (лучшие показатели по совокупности оценок), при этом эффективность ГА и ВГА сравнима.

Поскольку оба алгоритма являются стохастическими, для проверки того, являются ли различия между алгоритмами случайны-

Таблица 1. Сравнение эффективности ГА и ВГА при наилучших настройках

Задача	Метод	ГА				ВГА			
		X	Y	% / Ср.р.	% доп.	X	Y	% / Ср.р.	% доп.
1	**VEGA**	0,0358	0,0169	0,181	95,8005	0,0296	0,0158	0,2811	94,718
	FFGA	0,0287	0,0161	0,075	96,6498	0,0253	0,0159	0,1505	97,2794
	NPGA	0,0325	0,0185	0,1838	93,7909	0,0307	0,0166	0,2864	95,7321
	SPEA	0,0253	0,0205	0,1078	95,9333	0,0254	0,0205	0,1074	96,3287
2	**VEGA**	0,0529	0,0337	26,62	95,4466	0,0495	0,0309	26,2175	96,3083
	FFGA	0,0512	0,0327	50,1888	95,2674	0,0497	0,0314	56,8789	95,7812
	NPGA	0,0489	0,0308	92,5909	92,894	0,0476	0,0301	92,6855	92,8477
	SPEA	0,0553	0,0348	94,9333	98,5333	0,0554	0,035	93,4	98,6
3	**VEGA**	0,031	0,0181	32,1662	94,5157	0,0234	0,0131	41,7492	90,5893
	FFGA	0,0299	0,0173	28,8523	94,2752	0,0227	0,0126	44,3984	93,8237
	NPGA	0,0316	0,0181	92,7138	92,7138	0,0231	0,0132	93,5905	93,5905
	SPEA	0,0368	0,022	86,5609	86,4966	0,0364	0,0217	83,2409	84,6644
4	**VEGA**	0,003	0,0026	0,0103	30,1628	0,0014	0,0012	0,0255	33,0459
	FFGA	0,0032	0,0027	0,0189	74,8996	0,0014	0,0014	0,0458	72,016
	NPGA	0,0033	0,0033	0,0267	61,9082	0,0015	0,0015	0,0511	63,9618
	SPEA	0,0082	0,0065	0,0031	73,8352	0,0077	0,0061	0,0042	74,5916

ми, применялся статистический тест Колмогорова-Смирнова (KS-test) с 5 % уровнем значимости. Тест подтвердил, что различия случайны, а потому схема SPEA одинаково эффективна со стандартным ГА и с ВГА.

Из табл. 2 видно, что ВГА с адаптивной мутацией эффективнее, чем ВГА. В тех же случаях, когда ВГА с адаптивной мутацией уступает ВГА, средняя относительная эффективность не превосходит 5 % уровня значимости, а значит, можно сделать вывод о том, что алгоритмы сопоставимы по эффективности. При этом стоит отметить, что в случае адаптивной мутации пользователю алгоритма не нужно настраивать значение мутации (выбор которого существенно влияет на эффективность алгоритма).

Из табл. 3 видно, что при использовании прогнозирования критерий достижения множества Парето удовлетворяется на более ранней стадии работы алгоритма. Кроме того, алгоритм с прогнозированием показывает наилучший процент паретовских и допустимых точек. По степени разброса точек в пространстве альтернатив и пространстве критериев ВГА с прогнозом уступает по эффективности, однако это различие не превосходит 5 %, поэтому может быть признано несущественным.

Выводы

Использование ВГА в сочетании со схемой SPEA позволяет эффективно решать сложные задачи многокритериальной условной оптимизации, при этом эффективность

Таблица 2. Сравнение эффективности ВГА и ВГА с адаптивной мутацией

Задача	ВГА (адаптивная мутация)				ВГА			
	X	Y	% / Ср.р.	% доп.	X	Y	% / Ср.р.	% доп.
1	0,0245	0,0207	0,1047	98	0,0254	0,0205	0,1074	96,3287
2	0,0548	0,0348	93,8	98,8333	0,0554	0,035	93,4	98,6
3	0,0379	0,0219	83,5695	85,5695	0,0364	0,0217	83,2409	84,6644
4	0,0063	0,0049	0,0037	65,3628	0,0077	0,0061	0,0042	74,5916

Таблица 3. Сравнение эффективности ВГА и ВГА с прогнозом множества Парето

Задача	**ВГА (прогноз множества Парето)**				**ВГА**			
	X	Y	% / Ср.р.	% доп.	X	Y	% / Ср.р.	% доп.
1	0,0254	0,0207	0,0904	97,8833	0,0254	0,0205	0,1074	96,3287
2	0,0546	0,0348	93,8333	99	0,0554	0,035	93,4	98,6
3	0,0364	0,0216	85,1973	85,1973	0,0364	0,0217	83,2409	84,6644
4	0,0066	0,0054	0,0041	76,114	0,0077	0,0061	0,0042	74,5916

алгоритма сопоставима с эффективностью стандартного ГА. Однако использование ВГА более предпочтительно, т.к. у пользователя алгоритма нет необходимости настраивать параметры ВГА – использование распределения вероятностей замещает оператор скрещивания, а оператор мутации является адаптивным. В случае использования стандартного ГА необходим выбор наилучших (с точки зрения эффективности) параметров алгоритма, в противном случае эффективность алгоритма может существенно снижаться.

При использовании ВГА с прогнозом множества Парето, скорость нахождения парето-оптимальных решений выше, а значит требуется меньше вычислительных ресурсов для решения задачи. Более того, прогноз позволяет получать больший процент паретовских решений к концу работы алгоритма, следовательно, позволяет получить и более репрезентативную аппроксимацию множества Парето.

Библиографический список

1. Holland, J. H. Adaptation in natural and artificial systems / J. H. Holland. - MI: University of Michigan Press, 1975.

2. Goldberg, D. E. Genetic algorithms in search, optimization, and machine learning / D. E. Goldberg. - Reading, MA: Addison-Wesley, 1989.

3. Сопов, Е.А. Вероятностный генетический алгоритм и его исследование [Текст] / Сопов Е. А. // VII Королёвские чтения. - Самара: Изд-во Самарского научного центра РАН. - 2003. - Том 5. – С. 38-39.

4. Семенкин, Е. С. Вероятностные эволюционные алгоритмы оптимизации сложных систем [Текст] / Е. С. Семенкин, Е. А. Сопов // Труды Междунар. науч.-техн. конф. «Интеллектуальные системы» (AIS’05) и «Интеллектуальные САПР» (CAD-2005). В 3 т. – М.: ФИЗМАТЛИТ, 2005. - Т. 1.

5. Сопов, Е. А. О способах учета ограничений в вероятностном генетическом алгоритме [Текст] / Е. А. Сопов // Труды Междунар. научн.-техн. конф. «Интеллектуальные системы» (AIS’06) и «Интеллектуальные САПР» (CAD-2006). В 3 т. – М.: Физматлит, 2006. - Т.1. – 456 с.

6. Ворожейкин, А. Ю. Об одной модификации вероятностного генетического алгоритма для решения сложных задач условной оптимизации [Текст] / А. Ю. Ворожейкин [и др.] // Вестн. Сибир. гос. аэрокосм. ун-та. – Вып. 4 (25). - Красноярск, 2009.

7. Schaffer, J.D. Multiple objective optimization with vector evaluated genetic algorithms / J. D. Schaffer // In J. J. Grefenstette (Ed.), Proceedings of an International Conference on Genetic Algorithms and Their Applications, Pittsburgh, PA, 1985. – P. 93-100.

8. Fonseca, C. M. Multiobjective optimization and multiple constraint handling with evolutionary algorithms / C. M. Fonseca, P. J. Fleming // Part I: A unified formulation. Technical report 564, University of Sheffield, Sheffield, UK, January 1995.

9. Horn, J. A niched Pareto genetic algorithm for multiobjective optimization / J. Horn, N. Nafpliotis, D. E. Goldberg // In Proceedings of the First IEEE Conference on Evolutionary Computation, Vol. 1, Piscataway, 1994. – P. 82-87.

10. Zitzler, E. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach / E. Zitzler, L. Thiele // IEEE Transactions on Evolutionary Computation, Vol. 3, No. 4, 1999. – P. 257-271.

11. Ворожейкин, А. Ю. Разработка и исследование вероятностного генетического алгоритма многокритериальной оптимизации [Текст] / А. Ю. Ворожейкин, Е. С. Семенкин // Труды междунар. научн.-техн. конф. «Интеллектуальные системы» (AIS'07) и «Интеллектуальные САПР» (CAD-2007). – М.: Физматлит, 2007. - Т.1. – С. 10-19.

PROBABILITY-BASED GENETIC ALGORITHM FOR SOLVING COMPLEX MULTIOBJECTIVE OPTIMIZATION TASKS USING ADAPTIVE MUTATION OPERATOR AND PARETO SET PREDICTION

© 2011 Ye. A. Sopov, S. A. Sopov

Siberian State Aerospace University

The article is devoted to a new algorithm for solving complex constrained multiobjective optimization tasks on the basis of probability-based GA. An efficient modification of the algorithm using adaptive mutation and Pareto set prediction is proposed. The results of efficiency investigation are presented.

Multiobjective optimization, evolutionary algorithms, probabilitys-based GA, adaptive mutation.

Информация об авторах

Сопов Евгений Александрович, кандидат технических наук, доцент кафедры системного анализа Сибирского государственного аэрокосмического университета. E-mail: es_gt@mail.ru. Область научных интересов: генетические алгоритмы, интеллектуальные технологии анализа и моделирования сложных систем.

Сопов Сергей Александрович, студент 6-го курса Института информатики и телекоммуникаций Сибирского государственного аэрокосмического университета. E-mail: sopov_sergey@mail.ru. Область научных интересов: эволюционные алгоритмы, многокритериальная оптимизация.

Sopov Yevgeny Alexandrovitch, candidate of technical sciences, associate professor of the department of system analysis, Siberian State Aerospace University. E-mail: es_gt@mail.ru. Area of research: genetic algorithms, intelligent systems of complex system analysis and modeling.

Sopov Sergey Alexandrovitch, 6-th year student of the Institute of Informatics and Telecommunications, Siberian State Aerospace University. E-mail: sopov_sergey@mail.ru. Area of research: evolutionary algorithms, multiobjective optimization.