

Сравнение генетических алгоритмов MOGA и NSGA-II на задаче оптимизации формы рабочего колеса гидротурбины

А. К. ГАРАГУЛОВА*, Д. О. ГОРБАЧЕВА, Д. В. ЧИРКОВ

¹Институт вычислительных технологий СО РАН, Новосибирск, Россия

*Контактный e-mail: akgaragulova@gmail.com

Проведено сравнение генетических алгоритмов MOGA и NSGA-II на тестовых задачах и задаче оптимизации формы рабочего колеса гидротурбины. Рассмотрен модифицированный алгоритм NSGA-II_m, в котором операторы рекомбинации и мутации заимствованы из MOGA. Для сравнения скорости сходимости алгоритмов использована метрика, характеризующая расстояние от приближенного фронта Парето до точного. Представлены результаты решения тестовой задачи ZDT_3 без ограничений с 12 и 30 параметрами и тестовой задачи OSY с ограничениями. Алгоритмы сравниваются на задаче оптимизации формы радиально-осевого рабочего колеса гидротурбины с 24 свободными параметрами. Получены схожие результаты скорости и сходимости.

Ключевые слова: генетический алгоритм, NSGA-II, MOGA.

Библиографическая ссылка: Гарагулова А.К., Горбачева Д.О., Чирков Д.В. Сравнение генетических алгоритмов MOGA и NSGA-II на задаче оптимизации формы рабочего колеса // Вычислительные технологии. 2018. Т. 23, № 5. С. 21–36. DOI: 10.25743/ICT.2018.23.5.003.

Введение

Проектирование сложных проточных частей гидро- и газодинамических турбомашин осуществляется с активным применением вычислительного моделирования потока и методов автоматической оптимизации [1]. В работах [2–4] предложены постановки задач и результаты многоцелевой оптимизации форм рабочих колес гидравлических турбин по критериям максимизации КПД, прочности, минимизации металлоемкости с учетом кавитационных ограничений. В указанных работах для решения многоцелевой оптимизационной задачи использовался генетический алгоритм (ГА) MOGA [2], основанный на работах [5, 6]. Как правило, форма рабочего колеса задается значениями 20–30 геометрических параметров, и решается двух-, трехцелевая оптимизационная задача. Для вычисления целевых функционалов (КПД на нескольких режимах работы, кавитационных характеристик) по каждому варианту формы рабочего колеса проводился расчет 3D-течения жидкости с применением уравнений Эйлера или Навье — Стокса. Поиск оптимального решения — фронта Парето — требует перебора нескольких тысяч вариантов геометрии, что занимает около недели процессорного времени.

Увеличение числа свободных геометрических параметров, а также учет показателей работы гидромашин в широком диапазоне режимов приводят к замедлению сходимости генетического алгоритма MOGA и, как следствие, увеличению времени проектирования. Это обстоятельство наталкивает на поиск более эффективных алгоритмов и подходов к решению оптимизационной задачи.

Одним из наиболее популярных алгоритмов многоцелевой оптимизации является NSGA-II [7]. Он используется как самостоятельно, так и в связке с метамоделями. Более чем в половине работ, посвященных многоцелевой оптимизации для решения инженерных задач, использован этот алгоритм [8]. NSGA-II отличается от MOGA определением ранга и алгоритмом отбора индивидуумов-родителей. Благодаря введению специальной функции расстояния NSGA-II гарантирует попадание индивидуумов с экстремальными значениями целевых функционалов во множество родителей. Кроме того, NSGA-II использует операторы рекомбинации и мутации, имитирующие бинарный кроссовер и мутацию.

В настоящей работе дано описание алгоритмов MOGA и NSGA-II, а также проведено их сравнение на тестовых задачах и задаче оптимизации формы рабочего колеса гидротурбины. Рассмотрен модифицированный алгоритм NSGA-II_m, в котором операторы рекомбинации и мутации заимствованы из MOGA. Для сравнения скорости сходимости алгоритмов использована метрика M_1 , характеризующая расстояние от приближенного фронта Парето до точного [9]. Представлены результаты решения тестовой задачи ZDT_3 без ограничений и задачи OSY с ограничениями.

1. Постановка задачи многоцелевой оптимизации

Математически задача оптимизации формы рабочего колеса формулируется следующим образом: найти значения геометрических параметров $\mathbf{x} = (x_1, \dots, x_n)$, обеспечивающих минимальные значения M целевых функционалов (критериев качества):

$$\min (f_1(\mathbf{x}), \dots, f_M(\mathbf{x})), \quad \mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}_n,$$

где

$$\mathbf{X} = \{\mathbf{x} : x_{L,i} \leq x_i \leq x_{R,i}, i = 1, \dots, n\}$$

при наличии ограничений

$$\varphi_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, K.$$

Фронт Парето. Так как рассматриваемая задача оптимизации многоцелевая, не существует единственного решения, дающего минимум сразу всех целевых функционалов. Решением такой задачи является семейство оптимальных точек $\mathbf{P} \subset \mathbf{X}$, называемое фронтом Парето. Каждая точка фронта Парето оптимальна в том смысле, что путем ее возмущения невозможно улучшить какой-либо целевой функционал без ухудшения остальных критериев качества.

Понятие доминирования. В большинстве многоцелевых оптимизационных алгоритмов использована концепция доминирования. В таких алгоритмах два решения $\mathbf{p}, \mathbf{q} \in \mathbf{X}$ сравниваются путем проверки доминирования одного решения над другим.

Определение. Решение \mathbf{p} называется доминирующим над решением \mathbf{q} , если одновременно выполняются условия 1 и 2:

1. Решение \mathbf{p} не хуже решения \mathbf{q} по всем функционалам, т.е. выполняется $f_l(\mathbf{p}) \leq f_l(\mathbf{q})$ для всех $l = 1, \dots, M$.

2. Решение \mathbf{p} строго лучше решения \mathbf{q} хотя бы по одному функционалу, т. е. $f_{\bar{j}}(\mathbf{p}) < f_{\bar{j}}(\mathbf{q})$ хотя бы для одного $\bar{j} \in \{1, \dots, M\}$.

Исходя из этого определения возможны три отношения доминирования между двумя решениями: (i) решение \mathbf{p} доминирует над решением \mathbf{q} , (ii) решение \mathbf{q} доминирует над решением \mathbf{p} , (iii) оба решения не доминируют друг над другом. Вообще говоря, если решение \mathbf{p} не доминирует над решением \mathbf{q} , то это не означает, что \mathbf{q} доминирует над \mathbf{p} .

Для решения задачи многоцелевой оптимизации использованы генетические алгоритмы MOGA [2], NSGA-II [7] и модифицированный NSGA-IIм. Работа ГА основана на случайном подборе, комбинировании и вариации искомым параметров с применением методов, эмулирующих процессы естественной эволюции, таких как отбор, наследование, скрещивание и мутация.

2. Алгоритм MOGA

Пусть p_0 — число индивидуумов в начальном поколении, p — во всех остальных поколениях. Работа генетического алгоритма начинается с формирования случайным образом начальной популяции, состоящей из p_0 индивидуумов $\{\mathbf{x}_1, \dots, \mathbf{x}_{p_0}\} \subset \mathbf{X}$. Затем для каждого индивидуума построенной популяции вычисляются соответствующие им значения целевых функционалов и ограничений. Для каждого индивидуума, удовлетворяющего всем ограничениям, вычисляется ранг, характеризующий его качество [10]:

$$\text{rank}(\mathbf{x}^i) = 1 + a_i.$$

Здесь a_i — число индивидуумов текущей популяции, которые доминируют над \mathbf{x}^i и удовлетворяют всем ограничениям [10]. Оставшимся индивидуумам (для которых не выполнено хотя бы одно ограничение) ранг присваивается равным

$$\text{rank}(\mathbf{x}^i) = 2R + c^i,$$

где R — максимальный ранг удовлетворяющих ограничениям индивидуумов; c^i — максимальный выход за ограничения для индивидуума \mathbf{x}^i : $c^i = \max_j(\varphi_j(\mathbf{x}^i))$.

Для каждого индивидуума популяции \mathbf{x}^i вычисляется критерий качества, основанный на ранге:

$$F_r(\mathbf{x}^i) = 1 + \sum_{k=1}^{\text{rank}(\mathbf{x}^i)-1} \eta(k),$$

где $\eta(k)$ — число индивидуумов ранга k . После этого проводится процедура, предложенная Голдбергом и Ричардсоном [11]. Цель процедуры — распределить популяцию около некоторого числа различных пиков в пространстве поиска, причем возле каждого пика сосредоточивается часть популяции, пропорциональная высоте пика [6]. Иначе говоря, процедура проводится во избежание сходимости к какому-либо подмножеству точек фронта Парето, а также для получения равномерного распределения точек вдоль фронта Парето. Для этого критерий качества индивидуума, основанный на ранге F_r , умножается на нишевое число m_i , определяемое для каждого индивидуума. Нишевое

число m_i — это оценка того, как много индивидуумов популяции располагается в окрестностях (нише) индивидуума i . Оно вычисляется по всей популяции, включая i -й индивидуум, по формуле

$$m_i = \sum_{j \in Pop} Sh(\rho(i, j)),$$

где $\rho(i, j)$ — расстояние между индивидуумами \mathbf{x}^i и \mathbf{x}^j , Pop — текущая популяция, а $Sh(\rho)$ — функция разделения, которая находится как

$$Sh(\rho) = \begin{cases} 1 - \rho/\sigma_{share}, & \rho < \sigma_{share}, \\ 0, & \text{иначе.} \end{cases}$$

Здесь σ_{share} — параметр разделения, который подбирается эмпирически. Индивидуумы, находящиеся на расстоянии σ_{share} , ухудшают (увеличивают) функции качества $F_s = F_r m$ друг друга. Следовательно, сходимость происходит в нише радиуса σ_{share} , но сходимости всей популяции удастся избежать. Поскольку целевые функционалы могут принимать значения на отрезках разной длины, они предварительно нормируются таким образом, чтобы отображать область поиска \mathbf{X} в гиперкуб с единичным ребром $[0, 1]^M$. Расстояние между индивидуумами вычисляется в пространстве функционалов по формуле

$$\rho(i, j) = \max_{k=1, \dots, M} |f_k(\mathbf{x}^i) - f_k(\mathbf{x}^j)|.$$

Отбор индивидуумов проводят исходя из значения функции качества F_s . А именно выбирают $T_r p$ индивидуумов, имеющих наименьшие значения F_s . Здесь $T_r \in (0, 1)$, $T_r = \text{const}$. Как правило, $T_r = 0.3$. Получено множество родителей. Затем осуществляют рекомбинацию, мутацию, клонирование.

На этапе рекомбинации из множества родителей случайным образом выбирают два индивидуума-родителя $\mathbf{x} = (x_1, \dots, x_n)$ и $\mathbf{y} = (y_1, \dots, y_n)$, которые производят новый индивидуум $\mathbf{w} = (w_1, \dots, w_n)$. Это реализуется следующим образом: $w_i = x_i + \alpha_i(y_i - x_i)$, где α_i — случайное число между $-d$ и $1 + d$. В расчетах принималось $d = 0.7$. После рекомбинации получилось p индивидуумов. На этапе мутации каждый индивидуум меняется следующим образом: $w_i = w_i \pm \mu(w_{L,i} - w_{R,i})\delta$, где $\mu \in [0, 1]$ — мутационный параметр, $\delta = 2^{-16\gamma}$, а γ — случайное число между 0 и 1. В результате получается новое поколение размера p , заменяющее предыдущее, и этот процесс повторяется до тех пор, пока не будет рассчитано заданное число поколений N_{Gen} .

Процесс поиска решения включает следующие шаги.

1. Формирование начальной популяции, состоящей из p_0 индивидуумов. Каждый индивидуум — набор параметров $\mathbf{x} = (x_1, \dots, x_n)$, определяющий объект оптимизации, при этом $x_{L,i} \leq x_i \leq x_{R,i}$, $i = 1, \dots, n$. Начальная популяция создается случайным образом применительно ко всем параметрам.
2. Вычисление значений целевых функционалов f_1, \dots, f_M для каждого индивидуума популяции.
3. Вычисление ранга $rank$ каждого индивидуума популяции.
4. Вычисление критерия качества F_r , основанного на ранге, для каждого индивидуума.
5. Вычисление расстояний ρ между индивидуумами.
6. Вычисление нишевых чисел m для каждого индивидуума.
7. Вычисление функции качества F_s .

8. Отбор. Выбирается $T_r p$ ($0 < T_r < 1$) наилучших индивидуумов для скрещивания. В данном случае это индивидуумы, которым соответствуют наименьшие значения F_s .
9. Построение нового поколения. На данном этапе строится новое поколение, состоящее из p индивидуумов, путем рекомбинации и последующей мутации.
10. Клонирование. Копирование без изменения всех индивидуумов ранга 1, удовлетворяющих ограничениям, из предыдущего поколения в новое поколение. После клонирования размер новой популяции будет равен $p + p_e$, где p_e — число клонированных индивидуумов.
11. Переход на шаг 2, пока не будет рассчитано заданное число поколений N_{Gen} .

3. Алгоритм NSGA-II

Алгоритм NSGA-II основан на процедуре быстрой сортировки множества решений на слои не доминирующих друг над другом решений (fast non-dominated sort — FNDS) и процедуре оценки скученности решений в пространстве функционалов (crowding-distance-assignment).

Для отбора лучших индивидуумов вводят две характеристики: ранг индивидуума r и расстояние $dist$. Значение r для каждого решения вычисляется с помощью процедуры FNDS, после чего решения с одинаковым рангом r объединяются в подмножества \mathcal{F}_r , называемые фронтами ранга r .

Для каждого решения $\mathbf{x} \in \mathcal{F}_r$ вычисляется величина расстояния $dist(\mathbf{x}) = \sum_{m=1}^M dist_m(\mathbf{x})$, где $dist_m(\mathbf{x})$ — расстояние \mathbf{x} для m -го функционала. Расчет производится следующим образом.

Пусть $l = |\mathcal{F}_r|$. Решения сортируются в \mathcal{F}_r так, чтобы $\mathcal{F}_r(1) = f_m^{\min}$, $\mathcal{F}_r(l) = f_m^{\max}$. Граничные решения (решения с f_m^{\min} и f_m^{\max}) объявляются решениями с $dist_m = 10^6$. Для остальных промежуточных решений $\mathcal{F}_r(i)$ значение $dist_m$ определяется по формуле

$$dist_m(\mathcal{F}_r(i)) = \frac{\mathcal{F}_r(i+1).m - \mathcal{F}_r(i-1).m}{f_m^{\max} - f_m^{\min}},$$

где $i \in [2, l-1]$, а $\mathcal{F}_r(i).m$ — значение m -го функционала для i -го индивидуума в \mathcal{F}_r . Данный алгоритм имеет название crowding distance assignment. Ниже представлен псевдокод алгоритма [7].

При отборе решений лучшими считаются те, которые имеют меньший ранг r . При равенстве рангов лучшими считаются решения с большим значением расстояния $dist$. Таким образом, на всем множестве решений вводится отношение \prec_n . Выражение $\mathbf{p} \prec_n \mathbf{q}$ означает, что \mathbf{p} лучше \mathbf{q} , и равносильно выполнению

$$\mathbf{p} \prec_n \mathbf{q} \Leftrightarrow \left(r(\mathbf{p}) < r(\mathbf{q}) \right) \quad \text{или} \quad \left(r(\mathbf{p}) = r(\mathbf{q}) \text{ и } dist(\mathbf{p}) > dist(\mathbf{q}) \right).$$

Процесс оптимизации в алгоритме NSGA-II состоит из следующих шагов.

1. Случайным образом формируется начальная популяция родителей \mathbf{P}_0 , состоящая из N индивидуумов. Для каждого индивидуума в \mathbf{P}_0 вычисляются соответствующие значения функционалов и ограничений.
2. Далее случайным образом формируется начальная популяция детей \mathbf{Q}_0 , также состоящая из N индивидуумов.

Алгоритм. Crowding distance assignment (\mathcal{F}_r)

$l = \mathcal{F}_r $	▷ число индивидуумов в \mathcal{F}_r
for each $p \in \mathcal{F}_r$ do	
$dist(p) = 0$	▷ инициализация массива $dist$
end for	
for each $f_m, m \in [1, M]$ do	
$\mathcal{F}_r = sort(\mathcal{F}_r, m)$	▷ сортируем \mathcal{F}_r по значению m -го функционала
$dist(\mathcal{F}_r(1)) = dist(\mathcal{F}_r(l)) = 10^6$	▷ для граничных индивидуумов $dist = 10^6$
for each $i \in [2, l - 1]$ do	▷ для всех остальных используем формулу
$dist(\mathcal{F}_r(i)) = dist(\mathcal{F}_r(i)) + (\mathcal{F}_r(i + 1).m - \mathcal{F}_r(i - 1).m) / (f_m^{\max} - f_m^{\min})$	
end for	
end for	

3. Для каждого индивидуума в популяции детей \mathbf{Q}_t , $t \geq 0$, вычисляются соответствующие значения функционалов и ограничений.
4. Популяции детей и родителей объединяются: $\mathbf{R}_t = \mathbf{P}_t \cup \mathbf{Q}_t$. С помощью использования FNDS множество \mathbf{R}_t разбивается на фронты разного ранга \mathcal{F}_r , $r = 1, \dots, r_{\max}$. Далее для каждого индивидуума $\mathbf{x} \in \mathcal{F}_r$ определяется значение расстояния $dist(\mathbf{x})$ с помощью описанной выше процедуры crowding-distance assignment.
5. Отбор. На этом этапе из \mathbf{R}_t отбираются N лучших индивидуумов (согласно отношению \prec_n), остальные отбрасываются. Отобранные индивидуумы объединяются в новую популяцию родителей \mathbf{P}_{t+1} .
6. Построение новой популяции детей \mathbf{Q}_{t+1} , состоящей из N индивидуумов, с использованием процедуры рекомбинации и мутации. В алгоритме NSGA-II применяются SBX-рекомбинация и полиномиальная мутация, описанные в подразд. 3.1.
7. Переход на шаг 3, пока не будет рассчитано N_{Gen} поколений.

3.1. Рекомбинация SBX и полиномиальная мутация

В [7] авторы алгоритма NSGA-II предлагают использовать рекомбинацию SBX (simulated binary crossover) и полиномиальную мутацию, описанные в [12].

SBX-рекомбинация реализуется следующим образом. Из P_{t+1} случайным образом отбираются две пары индивидуумов. После сравнения индивидуумов внутри каждой пары по отношению \prec_n из них отбираются два лучших индивидуума-родителя $\mathbf{p}_1 = (p_{11}, \dots, p_{1n})$ и $\mathbf{p}_2 = (p_{21}, \dots, p_{2n})$ (эта процедура называется binary tournament selection — BTS). Индивидуумы-родители \mathbf{p}_1 и \mathbf{p}_2 с вероятностью $P_c = 0.9$ производят двух новых индивидуумов-детей $\mathbf{c}_1 = (c_{11}, \dots, c_{1n})$ и $\mathbf{c}_2 = (c_{21}, \dots, c_{2n})$. Реализуется это формулами

$$c_{1i} = 0.5 \left((1 + \beta)p_{1i} + (1 - \beta)p_{2i} \right),$$

$$c_{2i} = 0.5 \left((1 - \beta)p_{1i} + (1 + \beta)p_{2i} \right),$$

где

$$\beta = \begin{cases} (2u)^{\frac{1}{\eta_c + 1}}, & \text{если } u \leq 0.5, \\ \left(\frac{1}{2(1 - u)} \right)^{\frac{1}{\eta_c + 1}}, & \text{иначе.} \end{cases}$$

Здесь u — случайное число из $(0, 1)$, свое для каждого $i = 1, \dots, n$, а η_c — параметр рекомбинации. Получили Q_{t+1} .

На этапе мутации каждый параметр c_i , $i = 1, \dots, n$, индивидуума $\mathbf{c} \in Q_{t+1}$ меняется с вероятностью $P_m = 1/n$ следующим образом:

$$c_i = c_i + (x_{R,i} - x_{L,i})\delta,$$

$$\delta = \begin{cases} (2r)^{\frac{1}{\eta_m+1}} - 1, & \text{если } r < 0.5, \\ 1 - (2(1-r))^{\frac{1}{\eta_m+1}}, & \text{иначе,} \end{cases}$$

где r — случайное число из $(0, 1)$, свое для каждого $i = 1, \dots, n$, а η_m — параметр мутации.

В данной работе наряду с алгоритмом NSGA-II рассмотрен модифицированный алгоритм NSGA-III. Модификация заключается в замене операторов рекомбинации и мутации на шаге 6, а именно в NSGA-III используются операторы из алгоритма MOGA (см. разд. 2).

4. Сравнение алгоритмов

Сравнение алгоритмов MOGA, NSGA-II и NSGA-III проводилось на тестовых задачах ZDT_3 и OSY , а также на задаче оптимизации формы рабочего колеса гидротурбины.

Пусть \bar{Y} — множество точек, задающих точный фронт Парето, Y' — множество точек полученного фронта Парето, а $\bar{\mathbf{f}} = (\bar{f}_1, \dots, \bar{f}_M)$ и $\mathbf{f}' = (f'_1, \dots, f'_M)$ — функционалы, принадлежащие точному и полученному фронтам Парето соответственно. Для оценки сходимости рассмотрим расстояние от полученного фронта Парето до точного в пространстве функционалов. Обозначим данное расстояние M_1 и определим его следующей формулой [9]:

$$M_1(Y') = \frac{1}{|Y'|} \sum_{\mathbf{f}' \in Y'} \min \left\{ \|\mathbf{f}' - \bar{\mathbf{f}}\|; \bar{\mathbf{f}} \in \bar{Y} \right\}.$$

В качестве $\|\mathbf{f}' - \bar{\mathbf{f}}\|$ рассматривается евклидово расстояние между точками \mathbf{f}' и $\bar{\mathbf{f}}$. Чем меньше значение M_1 , тем ближе расположение найденного алгоритмом фронта Парето к точному фронту Парето. Для усреднения данного расстояния проведено по 100 запусков алгоритма. Далее для усредненного расстояния использовано обозначение M_{avr} .

4.1. Задача ZDT_3

Целевые функционалы задачи ZDT_3 :

$$f_1(\mathbf{x}) = x_1, \quad f_2(\mathbf{x}) = g \left(1 - \sqrt{\frac{f_1}{g}} \right) - \left(\frac{f_1}{g} \right) \sin(10\pi f_1),$$

где $g(x_2, \dots, x_n) = 1 + 9 \sum_{i=2}^n \frac{x_i}{n-1}$ с ограничениями $0 \leq x_i \leq 1$, $i = 1, \dots, n$, n — число параметров. Точное решение задачи ZDT_3 определяется функционалами f_1 , f_2 при $g(x_2, \dots, x_n) \equiv 1$, т. е.

$$f_1(\mathbf{x}) = x_1, \quad f_2(\mathbf{x}) = \left(1 - \sqrt{f_1} \right) - f_1 \sin(10\pi f_1).$$

Рассмотрим задачу ZDT_3 с $n = 30$. На рис. 1, *a* представлены приближенные фронты Парето для алгоритмов MOGA (1), NSGA-II (2) и NSGA-IIм (3), полученные для 80 индивидуумов на поколении $N_{Gen} = 100$. С целью удобства визуализации фронты Парето для алгоритмов NSGA-II и NSGA-IIм смещены вверх и вправо. На рис. 1, *a* и 2, *a* сплошной серой линией (0) показан точный фронт Парето. Можно заметить, что фронт Парето, полученный при использовании алгоритма NSGA-II, аппроксимирует точный фронт лучше, чем MOGA и NSGA-IIм. Данное замечание подтверждает рис. 1, *б*,

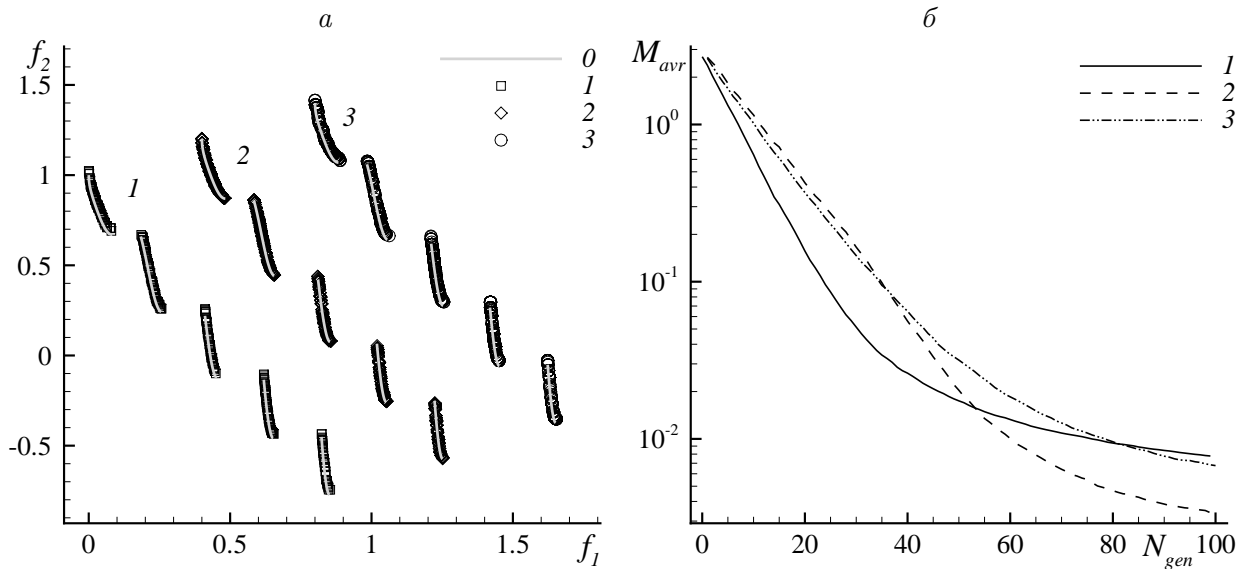


Рис. 1. Фронты Парето для 80 индивидуумов на 100-м поколении (сплошная серая кривая (0) — точный фронт Парето) (а); график среднего расстояния от приближенного фронта Парето до точного (б) для алгоритмов MOGA, NSGA-II и NSGA-IIм, кривые 1–3 соответственно

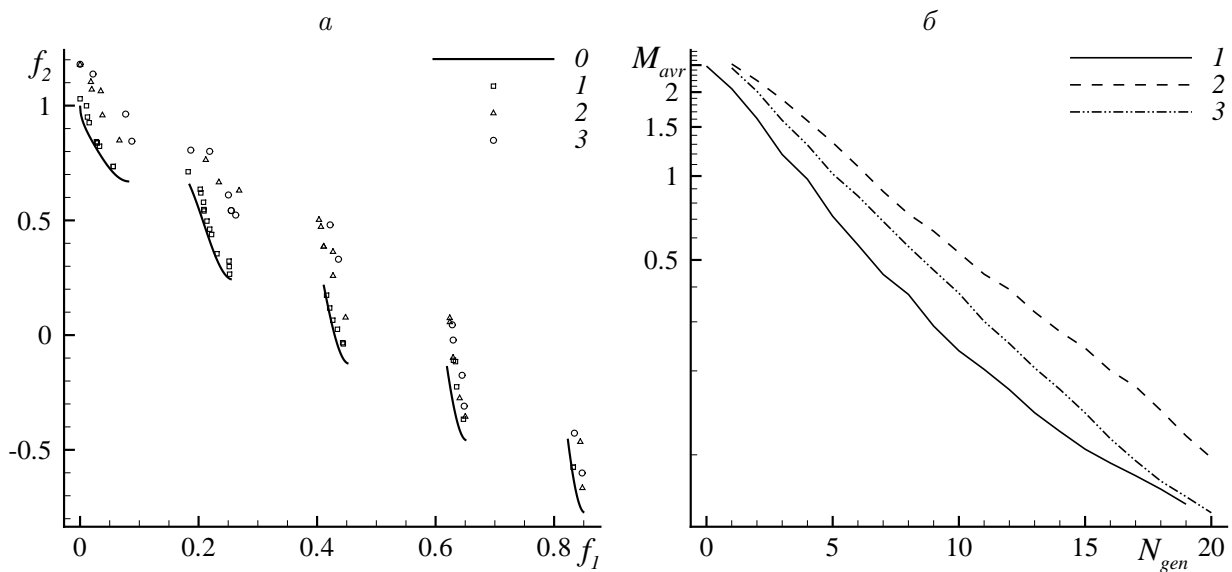


Рис. 2. Фронты Парето для 40 индивидуумов на 20-м поколении (сплошная кривая (0) — точный фронт Парето) (а); график среднего расстояния от приближенного фронта Парето до точного (б) для алгоритмов MOGA, NSGA-II и NSGA-IIм, кривые 1–3 соответственно

на котором для каждого алгоритма представлены зависимости среднего расстояния M_{avr} от номера поколения. Отметим, что фронт Парето, полученный с помощью алгоритма MOGA, сходится к точному фронту быстрее NSGA-IIм и NSGA-II, при этом последний дает бóльшую точность.

Рассмотрим задачу ZDT_3 с $n = 12$. Аналогично, на рис. 2, а представлены приближенные фронты Парето для алгоритмов 1–3, полученные для 40 индивидуумов на поколении $N_{Gen} = 20$. При таком количестве параметров видим, что алгоритм MOGA находит приближенный фронт Парето лучше и быстрее (рис. 2, б), чем алгоритмы NSGA-II и NSGA-IIм.

4.2. Задача OSY

В качестве еще одной тестовой задачи рассмотрим задачу OSY, целевые функционалы в которой задаются следующим образом:

$$f_1(\mathbf{x}) = - [25(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2], \quad f_2(\mathbf{x}) = \sum_{i=1}^6 x_i^2.$$

Ограничения:

$$0 \leq x_1, x_2, x_6 \leq 10, \quad 1 \leq x_3, x_5 \leq 5, \quad 0 \leq x_4 \leq 6,$$

$$\varphi_1(\mathbf{x}) = 2 - x_1 - x_2 \leq 0, \quad \varphi_2(\mathbf{x}) = x_1 + x_2 - 6 \leq 0, \quad \varphi_3(\mathbf{x}) = -x_1 + x_2 - 2 \leq 0,$$

$$\varphi_4(\mathbf{x}) = x_1 - 3x_2 - 2 \leq 0, \quad \varphi_5(\mathbf{x}) = (x_3 - 3)^2 + x_4 - 4 \leq 0, \quad \varphi_6(\mathbf{x}) = 4 - (x_5 - 3)^2 + x_6 \leq 0.$$

Точный фронт Парето задачи OSY состоит из пяти участков (табл. 1) и получается при $x_4^* = x_6^* = 0$. В расчетах число индивидуумов взято равным 100, число поколений $N_{Gen} = 100$. Задача является сложной, и результаты работы алгоритмов меняются от запуска к запуску.

На рис. 3, а представлены фронты Парето, полученные алгоритмами MOGA и NSGA-II. Можно заметить, что NSGA-II покрывает найденными решениями бóльшую часть точного фронта Парето, нежели решения, полученные алгоритмом MOGA. Данный эффект наблюдался в большинстве испытаний. Чтобы его наглядно продемонстрировать, вычислено среднее по ста запускам минимальное значение f_1 на приближенном фронте Парето. На рис. 3, а это значение помечено цифрами 3 и 4 для алгоритмов MOGA и NSGA-II соответственно, а именно $f_{1,\min}^{\text{MOGA}} = -247.113$ и $f_{1,\min}^{\text{NSGA-II}} = -258.315$. По-видимому, такое более распределенное расположение решений NSGA-II на точном фронте Парето связано с особенностью селекции. В алгоритме NSGA-II крайние точки,

Т а б л и ц а 1. Точный фронт Парето для задачи OSY [12]

Участок	Оптимальные параметры			
	x_1^*	x_2^*	x_3^*	x_5^*
1	5	1	1, ..., 5	5
2	5	1	1, ..., 5	1
3	4.056, ..., 5	$(x_1^* - 2)/3$	1	1
4	0	2	1, ..., 3.732	1
5	0, ..., 1	$2 - x_1^*$	1	1

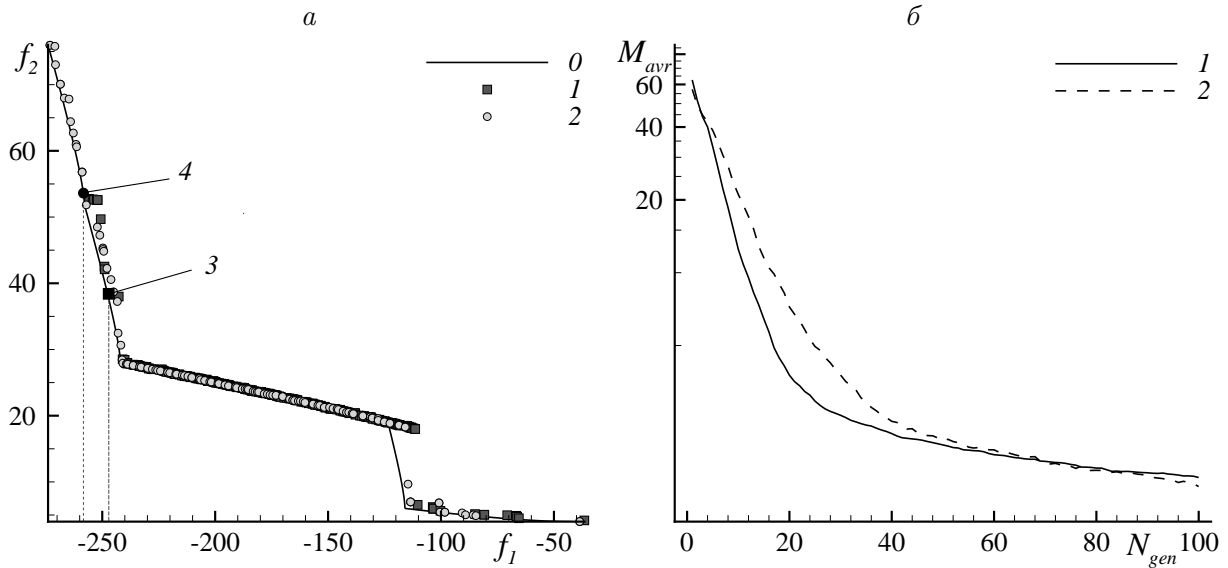


Рис. 3. Фронты Парето для ста индивидуумов на 100-м поколении (сплошная кривая 0 — точный фронт Парето) (а); график среднего расстояния от приближенного фронта Парето до точного (б) для алгоритмов MOGA, NSGA-II — кривые 1, 2 соответственно; точки 3, 4 — среднее минимальное значение f_1 для алгоритмов MOGA и NSGA-II соответственно

т. е. точки с экстремальными значениями каждого функционала, точно войдут в новое множество родителей в отличие от MOGA.

На графике зависимости среднего по ста запускам расстояния M_{avr} от номера поколения (рис. 3, б) видно, что по скорости сходимости алгоритмы близки (хотя на первых поколениях алгоритм MOGA сходится быстрее).

4.3. Двухцелевая оптимизации формы рабочего колеса гидротурбины

Параметризация формы рабочего колеса гидротурбины разработана в [2]. Форма рабочего колеса определяется значениями 24 параметров, задающих форму лопасти, ступицы и обода (рис. 4) [2, 3]. Варьирование формы рабочего колеса отталкивается от некоторого начального прототипа, который задает пользователь, и заключается в изменении кривизны срединной поверхности лопасти, контуров RZ -проекции лопасти, ступицы и обода. Как правило, оптимизация направлена на повышение КПД η в нескольких режимах работы турбины. Для вычисления целевых функционалов для каждого варианта геометрии проводится расчет трехмерного течения с помощью комплекса гидродинамических расчетов CADRUN-1 [2]. Численно решаются усредненные по Рейнольдсу уравнения Навье — Стокса в направляющем аппарате (НА) и рабочем колесе (РК). Для моделирования турбулентности используется $k - \varepsilon$ -модель [13]. Расчет одной геометрии занимает порядка 20 мин процессорного времени (процессор Intel® Core™ i7 CPU 950 @ 3.07GHz. ОЗУ 12 ГБ).

Сравнение алгоритмов проводилось на примере двухцелевой оптимизации формы радиально-осевого рабочего колеса ГЭС Каракая с целевыми функционалами: максимизация КПД в режиме 1 (неполной нагрузки) η_1 и в режиме 2 (повышенной нагрузки) η_2 . Задавались ограничения на рассчитанные КПД в первом режиме ($\eta_1 > 0.9$) и во втором ($\eta_2 > 0.89$).

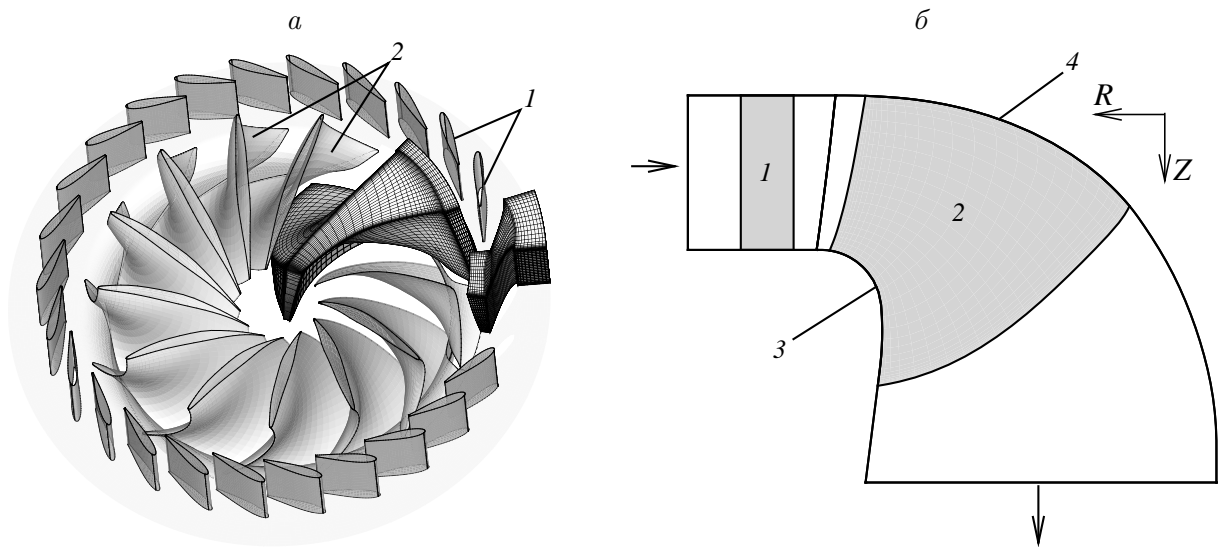


Рис. 4. Расчетная сетка в направляющем аппарате (НА) и рабочем колесе (РК) (а); RZ-проекция НА и РК: 1 — лопатки НА, 2 — лопасти РК, 3 — обод, 4 — ступица

На рис. 5, а представлены результаты двух оптимизационных расчетов: MOGA и NSGA-II. В каждом расчете выполнено по 60 поколений для 72 индивидуумов в каждом. Можно заметить, что оба алгоритма в качестве результата выдали близкие фронты Парето. Отличие наблюдается в количестве решений и равномерности их распределения. Алгоритм MOGA предоставил более расширенный и заполненный фронт Парето, нежели NSGA-II.

Для данной задачи точный фронт Парето неизвестен, поэтому для оценки скорости сходимости алгоритмов создан искусственный “точный” фронт Парето. Так как задача вычислительно затратная, выполнены только по три расчета для алгоритмов MOGA

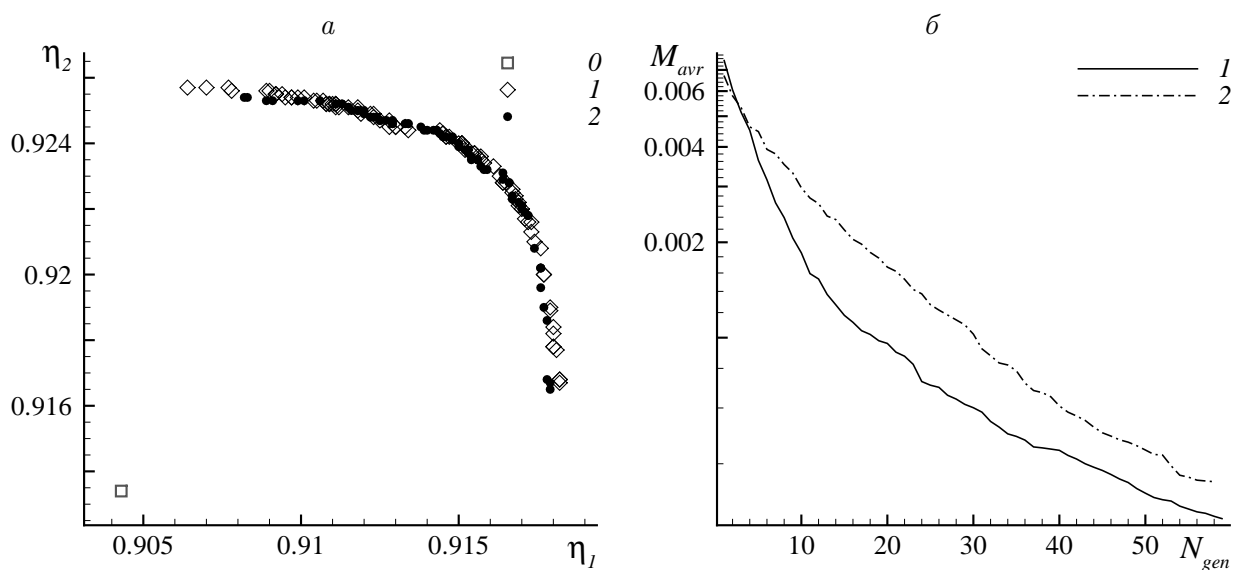


Рис. 5. Фронты Парето для 72 индивидуумов на шестидесятом поколении (0 — начальный индивидуум) (а); график среднего расстояния от оптимального фронта Парето до “точного” (б) для алгоритмов MOGA (1) и NSGA-II (2)

Т а б л и ц а 2. Результаты трех испытаний алгоритмов MOGA и NSGA-II

Показатель	MOGA			NSGA-II		
	1	2	3	1	2	3
№ запуска						
Количество решений на фронте	87	93	87	75	78	84
η_1^{\max}	0.9187	0.9182	0.9184	0.9187	0.9179	0.9178
η_2^{\min}	0.9142	0.9167	0.9160	0.9167	0.9165	0.9184
η_2^{\max}	0.9248	0.9257	0.9253	0.9247	0.9254	0.9255
η_1^{\min}	0.9115	0.9064	0.9094	0.9133	0.9082	0.9062
$M_2^*, \sigma = 10^{-5}$	0.9882	0.9890	0.9882	0.9823	0.9841	0.9863
$M_2^*, \sigma = 2.5 \cdot 10^{-4}$	0.9494	0.9544	0.9558	0.9232	0.9505	0.9618

и NSGA-II и уже для полученных фронтов была проведена общая огибающая, которая принималась за точный фронт Парето. На рис. 5, б показана зависимость расстояния M_{avr} от приближенного фронта до “точного”, от номера поколения, усредненного по трем запускам алгоритмов. По результатам трех испытаний можно сделать вывод, что на первых десяти поколениях скорость сходимости алгоритма MOGA превышает скорость сходимости NSGA, тогда как далее скорости обоих алгоритмов совпадают. Аналогичный результат наблюдался в тестовых задачах (см. рис. 1, б–3, б).

Для количественной оценки равномерности распределения индивидуумов на приближенном фронте Парето рассчитана метрика M_2 [9]:

$$M_2(Y) = \frac{1}{|Y|(|Y| - 1)} \sum_{\mathbf{f} \in Y} |\{\mathbf{g} \in Y : \|\mathbf{f} - \mathbf{g}\| > \sigma^*\}|,$$

где Y — множество точек приближенного фронта Парето в пространстве функционалов, $|Y|$ — мощность множества Y , а $\sigma^* > 0$ выбирается в зависимости от задачи. Значение функции M_2 лежит в отрезке $[0, 1]$, определяя часть точек, лежащих относительно друг друга на расстоянии больше σ^* . Чем ближе M_2 к 1, тем более равномерно распределены точки приближенного фронта Парето.

В табл. 2 представлены количество решений на приближенном фронте Парето в последнем поколении для обоих алгоритмов, экстремальные значения целевых функционалов, а также значения функции M_2 для $\sigma^* = 10^{-5}, 2.5 \cdot 10^{-4}$. Видно, что MOGA и NSGA-II дают сопоставимую равномерность распределения.

4.4. Трехцелевая оптимизация формы рабочего колеса гидротурбины

Известно, что наиболее подвержены кавитации тыльные поверхности лопастей. В случае, когда область кавитации занимает более 30 % площади всей тыльной поверхности лопасти, наблюдается резкое падение мощности и КПД турбины. Поэтому при оптимизации формы РК наряду с повышением КПД логично минимизировать относительный размер области кавитации на тыльной стороне лопасти.

Следующее сравнение алгоритмов проведено на задаче трехцелевой оптимизации формы рабочего колеса ГЭС Каракая со следующими целевыми функционалами:

- максимизация КПД η_1 и η_2 в режимах 1 и 2;
- минимизация взвешенной относительной площади кавитации (W_{cav}) на тыльной стороне лопасти на режиме 2 W_{cav2} [2].

В данной задаче рассматриваются те же ограничения на рассчитанные КПД, что и в разд. 4.3.

На рис. 6 приведены результаты трех оптимизационных расчетов: MOGA, NSGA-II и NSGA-II_m, в каждом из которых рассчитано по 60 поколений с 72 индивидуумами в каждом. Изображены проекции трехмерных приближенных фронтов Парето на плоскость η_1, η_2 . Сплошной линией представлена общая кривая, введенная для удобства сравнения результатов трех алгоритмов. В качестве такой кривой выбрана огибающая приближенного фронта Парето, полученного алгоритмом MOGA. Кругом выделена область компромиссных решений, представляющая интерес в данной задаче, поскольку в ней имеет место одновременное увеличение η_1 и η_2 . Можно заметить, что в выделенной области фронты Парето, полученные алгоритмами MOGA и NSGA-II_m, находятся близко к огибающей в отличие от фронта, полученного алгоритмом NSGA-II. Прибли-

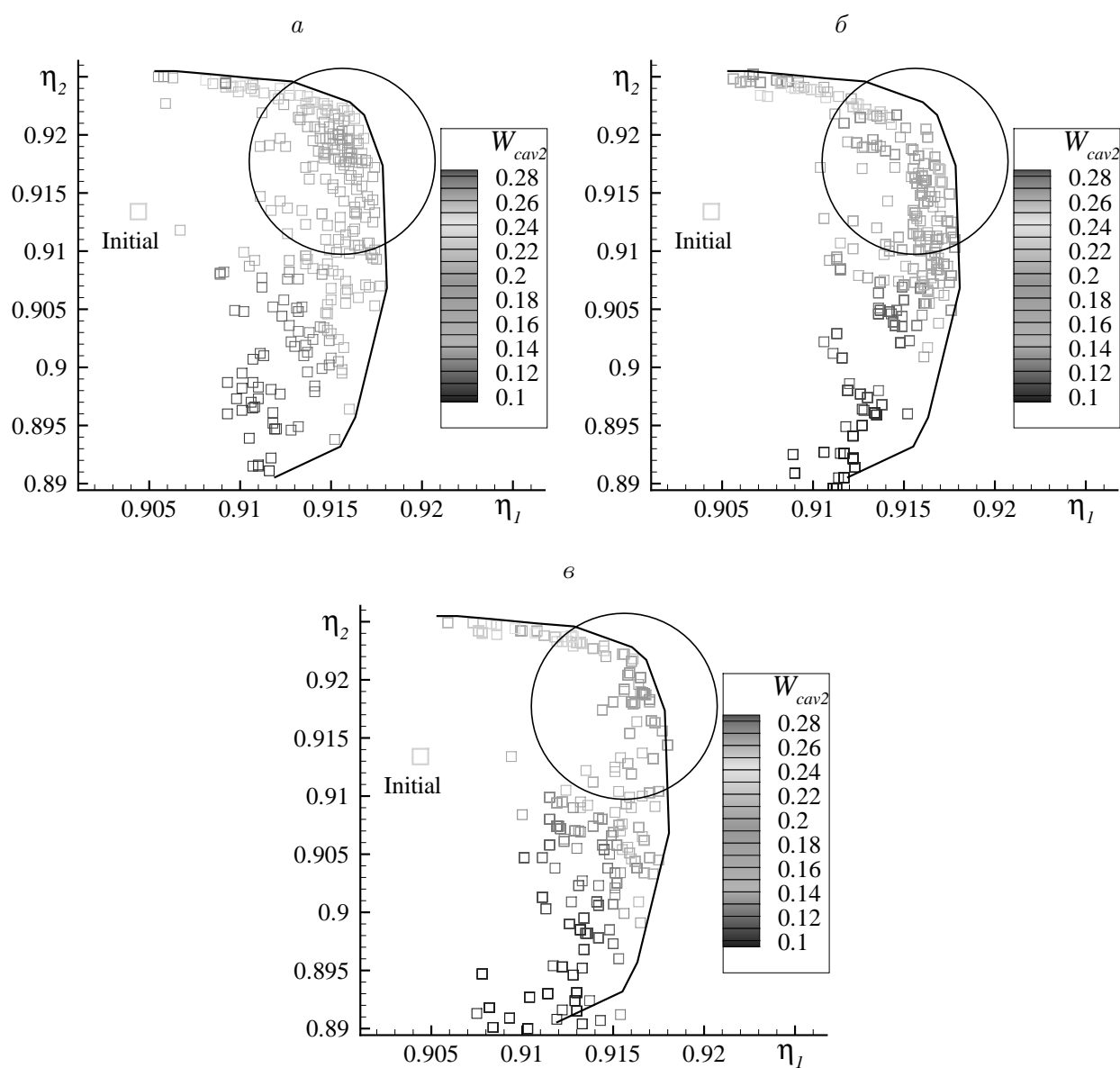


Рис. 6. Приближенные фронты Парето для 72 индивидуумов на 60-м поколении для алгоритмов: *a* — MOGA, *б* — NSGA-II, *в* — NSGA-II_m

Т а б л и ц а 3. Экстремальные значения целевых функционалов

Алгоритм	η_1^{\max}	η_2^{\max}	W_{cav2}^{\min}
MOGA	0.9177	0.9250	0.0930
NSGA-II	0.9178	0.9252	0.0327
NSGA-II _m	0.9180	0.9149	0.0178

женный фронт Парето в MOGA более плотный, чем в алгоритмах NSGA-II и NSGA-II_m. В алгоритме MOGA приближенный фронт Парето содержит 285 индивидуумов, тогда как количество решений во фронте Парето, полученных NSGA-II, ограничено самим алгоритмом. NSGA-II может найти не более $2N$ точек, где N — число индивидуумов в поколении (в данном случае $N = 72$). Фронт Парето в расчете с NSGA-II получился хуже, чем в расчете с NSGA-II_m. Достоинством NSGA-II является чуть бóльшая точность нахождения экстремальных значений целевых функционалов (табл. 3).

Заключение

Основной целью данной работы был анализ и сравнение двух популярных генетических алгоритмов для решения задач многокритериальной оптимизации: MOGA и NSGA-II. Дополнительно введен и рассмотрен модифицированный алгоритм NSGA-II_m, отличие которого от основного NSGA-II заключается в операторах рекомбинации и мутации. Работа алгоритмов сравнивалась на тестовых задачах ZDT_3 с 30 и 12 параметрами, задаче OSY с шестью параметрами и шестью ограничениями, а также на задаче двух- и трехцелевой оптимизации формы рабочего колеса гидротурбины с 24 геометрическими параметрами. Для сравнения скорости сходимости алгоритмов была введена метрика M_1 [9], характеризующая расстояние от точного фронта Парето до приближенного. В тестовых задачах данная метрика усреднялась по ста запускам алгоритма.

Замечено, что на первых 50 поколениях алгоритм MOGA сходился быстрее, однако после 50 поколений алгоритм NSGA-II показывал чуть лучшие результаты.

Задача оптимизации формы рабочего колеса гидротурбины вычислительно затратная, поэтому количество испытаний было ограничено. Так, для двухцелевой задачи оптимизации проведено по три испытания каждого алгоритма. Приближенные фронты, полученные MOGA и NSGA-II, практически совпадают. На первых поколениях алгоритм MOGA чуть быстрее сходится к искусственно созданному “точному” фронту Парето и дает приближенный фронт, содержащий большее число решений, нежели NSGA-II. На задаче трехцелевой оптимизации рабочего колеса проведено по одному запуску каждого алгоритма. Скорость сходимости не сравнивалась. Приближенный фронт Парето, полученный алгоритмом MOGA, распределен равномернее в отличие от других алгоритмов и содержит большее число решений. Достоинством алгоритмов NSGA-II и NSGA-II_m является чуть лучшее определение экстремальных значений целевых функционалов.

Из-за ограниченности числа испытаний сравнение алгоритмов на задаче оптимизации формы РК нельзя назвать строгим. Однако обнаруженные в этих расчетах отличия по скорости сходимости алгоритмов MOGA и NSGA-II совпадают с результатами статистически более аккуратного сравнения на тестовых задачах. Кроме того, сами отличия относительно невелики. Из этих соображений можно сделать вывод, что оба алгоритма сопоставимы и могут применяться для решения сложных инженерных задач.

Список литературы / References

- [1] Nakamura, K., Kurosawa, S. Design optimization of a high specific speed Francis turbine using multi-objective genetic algorithm // Intern. J. of Fluid Machinery and Systems. 2009. Vol. 2, No. 2. P. 102–109.
- [2] Численное моделирование течений в турбомашинах / Черный С.Г., Чирков Д.В., Лапин В.Н. и др. Новосибирск: Наука, 2006. 202 с.
Numerical modeling of flows in turbomachines / Cherny S.G., Chirkov D.V., Lapin V.N. et al. Novosibirsk: Nauka, 2006. 202 p. (In Russ.)
- [3] Lyutov, A.E., Chirkov, D.V., Skorospelov, V.A. et al. Coupled multipoint shape optimization of runner and draft tube of hydraulic turbines // J. Fluids Eng. 2015. Vol. 137, iss. 11. Article Number: 111302. 11 p.
- [4] Chirkov, D.V., Ankudinova, A.S., Kryukov, A.E. et al. Multi-objective shape optimization of a hydraulic turbine runner using efficiency, strength and weight criteria // Structural and Multidisciplinary Optimization. 2018. Vol. 58, iss. 2. P. 627–640.
- [5] Fonseca, C.M., Fleming, P.J. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization // Proc. of the 5th Intern. Conf. on Genetic Algorithms, Urbana-Champaign, IL, USA, 1993. San Francisco, CA, USA: Morgan Kaufmann Publ., 1993. P. 416–423.
- [6] Horn, J., Nafpliotis, N. Multiobjective optimization using the niched Pareto genetic algorithm // Proc. of the First IEEE Conf. on Evolutionary Comput., IEEE World Congress on Comput. Intelligence, Urbana Illinois, 1993. Piscataway, New Jersey: IEEE Service Center, 1994. Report № 93005. 33 p.
- [7] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. A fast and elitist multi-objective genetic algorithm: NSGA-II // IEEE Trans. on Evol. Comput. 2002. Vol. 6. P. 182–197.
- [8] Diaz-Manriquez, A., Toscano, G., Barron-Zambrano, J.H., Tello-Leal, E. A review of surrogate assisted multiobjective evolutionary algorithms // Comput. Intelligence and Neurosci. 2016. Vol. 2016. Article ID 9420460. 14 p.
- [9] Zitzler, E., Deb, K., Thiele, L. Comparison of multiobjective evolutionary algorithms: Empirical results // Evolutionary Comput. 2000. Vol. 8, No. 2. P. 173–195.
- [10] Латыпов Ф.Ф., Никуличев Ю.В. Специализированный комплекс программ оптимизации. Новосибирск, 1985. 41 с. (Препр. / АН СССР. Сиб. отд-ние. ИТПМ; № 15).
Latypov, F.F., Nikulichev, Yu.V. Specialized complex of optimization programs. Novosibirsk, 1985. 41 p. (Prepr. / AN SSSR. Sib. otd-nie. ITPM; No. 15.) (In Russ.)
- [11] Goldberg, D.E., Richardson, J.J. Genetic algorithms with sharing for multimodal function optimization // Proc. of the Second Intern. Conf. on Genetic Algorithms, Cambridge, MA, 1987. Hillsdale, New York: Lawrence Erlbaum Assoc., 1987. P. 41–49.
- [12] Deb, K. Multi-objective optimization using evolutionary algorithms. New York: John Wiley & Sons, 2001. 497 p.
- [13] Launder, B.E., Spalding, D.B. The numerical computation of turbulent flows // Comput. Methods in Appl. Mech. and Eng. 1974. Vol. 3, iss. 2. P. 269–289.

Поступила в редакцию 3 мая 2018 г.,
с доработки — 9 июля 2018 г.

Comparative analysis of MOGA and NSGA-II on the case study of optimization for the profile of the hydraulic turbine runner

GARAGULOVA, ANASTASIYA K.* , GORBACHEVA, DARIA O., CHIRKOV, DENIS V.

¹Institute of Computational Technologies SB RAS, Novosibirsk, 630090, Russia

*Corresponding author: Garagulova, Anastasiya K., e-mail: akgaragulova@gmail.com

Purpose. The main goal of this article is to compare two popular multi-objective genetic algorithms MOGA and NSGA-II by solving test problems and a practical problem of hydraulic turbine runner optimization. Modification of NSGA-II called NSGA-II_m is also considered. The major problem is to compare convergence rates of approximate solution to the exact Pareto front.

Methodology. The genetic algorithms MOGA and NSGA-II are described in detail. The modified algorithm NSGA-II_m is NSGA-II with the recombination and mutation operators taken from the MOGA algorithm. The known problems ZDT3 (2 objectives, 30 and 12 parameters, no constraints) and OSY (2 objectives, 6 parameters, 6 constraints) are taken as test problems. These algorithms are compared by solving runner optimization problem with 24 free parameters and 2 or 3 objectives.

To compare the algorithms, a metric characterizing the distance from the approximate Pareto front to the exact one is introduced. Since the algorithms are stochastic in nature, the value of the metric was averaged over 100 runs of the algorithm. In the two-objective runner optimization problem the metric value was averaged over 3 runs of the algorithm.

Findings. Solving the test problems, it was found that in the first 50 generations the MOGA algorithm converges faster than other algorithms, but after 50 generations the NSGA-II algorithm has shown the best result.

The MOGA algorithm gives an approximate front containing more solutions than NSGA-II. When solving the two-objective and the three-objective runner optimization problem the similar results were obtained. The approximate Pareto front, obtained by the MOGA algorithm, is distributed more uniformly than other algorithms and contains a larger number of solutions. The advantage of the algorithms NSGA-II and NSGA-II_m is a slightly better definition of extreme values of target functionals. However, the obtained differences are not very significant.

Originality/value. Obtained results show that both MOGA and NSGA-II are very similar in terms of convergence rate and can be applied for solving complex engineering problems.

Keywords: genetic algorithm, NSGA-II, MOGA.

Cite: Garagulova, A.K., Gorbacheva, D.O., Chirkov, D.V. Comparative analysis of MOGA and NSGA-II on the case study of optimization for the profile of the hydraulic turbine runner // Computational Technologies. 2018. Vol. 23, No. 5. P. 21–36. (In Russ.) DOI: 10.25743/ICT.2018.23.5.003.

Received 3 May 2018

Received in revised form 9 June 2018