

Интернет-журнал «Наукоедение» ISSN 2223-5167 <http://naukovedenie.ru/>

Том 8, №4 (2016) <http://naukovedenie.ru/index.php?p=vol8-4>

URL статьи: <http://naukovedenie.ru/PDF/67EVN416.pdf>

DOI: 10.15862/67EVN416 (<http://dx.doi.org/10.15862/67EVN416>)

Статья опубликована 26.08.2016.

Ссылка для цитирования этой статьи:

Грошев С.В., Карпенко А.П., Мартынюк В.А. Эффективность популяционных алгоритмов Парето-аппроксимации. Экспериментальное сравнение // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 8, №4 (2016) <http://naukovedenie.ru/PDF/67EVN416.pdf> (доступ свободный). Загл. с экрана. Яз. рус., англ. DOI: 10.15862/67EVN416

Работа поддержана РФФИ (проекты № 16-07-00287, 15-07-01764)

УДК 519.6

Грошев Сергей Владимирович

ФГБОУ «Московский Государственный технический университет им. Н.Э. Баумана», Россия, Москва¹
Старший преподаватель кафедры «Системы автоматизированного проектирования»
E-mail: groshev_sergey@mail.ru

Карпенко Анатолий Павлович

ФГБОУ «Московский Государственный технический университет им. Н.Э. Баумана», Россия, Москва
Зав. кафедрой «Системы автоматизированного проектирования»
Доктор физико-математических наук, профессор
E-mail: akarpenko@bmstu.ru

Мартынюк Владимир Алексеевич

ФГБОУ «Московский Государственный технический университет им. Н.Э. Баумана», Россия, Москва
Доцент кафедры «Системы автоматизированного проектирования»
Кандидат технических наук
E-mail: Martrk6@gmail.com

Эффективность популяционных алгоритмов Парето-аппроксимации. Экспериментальное сравнение

Аннотация. Задача многокритериальной оптимизации возникает в различных областях науки и техники, и предложено несколько различных способов ее решения. Классические алгоритмы решения задачи многокритериальной оптимизации (МКО-задачи) основаны на сведении ее к совокупности задач глобальной однокритериальной оптимизации. Относительно новый и быстро развивающийся класс алгоритмов многокритериальной оптимизации образуют алгоритмы на основе Парето-аппроксимации, предполагающие предварительное построение некоторой конечномерной аппроксимации множества, а тем самым, и фронта Парето рассматриваемой МКО-задачи. Обычно алгоритмы Парето-аппроксимации строят на основе эволюционных и, чаще всего, генетических алгоритмов. Принципиальными в эволюционных алгоритмах Парето-аппроксимации являются правила формирования фитнес-функции, обеспечивающей перемещение индивидов популяции, в конечном счете, в направлении множества Парето. Работа носит обзорный характер и включает в себя результаты экспериментальных исследований эффективности ряда

¹ 105005, г. Москва, ул. 2-я Бауманская, д. 5

популяционных алгоритмов Парето-аппроксимации, полученные с участием авторов в период с примерно 2010 года по настоящее время.

Работа имеет целью показать на ряде известных тестовых и практически значимых МКО-задач сравнительную эффективность Парето-аппроксимации с помощью генетического алгоритма, алгоритмов роя частиц, колонии муравьев и колонии пчел в комбинации с различными методами формирования фитнес-функции. Для оценки эффективности указанных алгоритмов Парето-аппроксимации в работе использован, индикатор, называемый *разреженностью покрытия* (*scarcity*), имеющий смысл минимального расстояния между решениями, принадлежащими Парето-аппроксимации. Кроме этого в качестве индикатора используется число недоминируемых решений, найденных алгоритмом Парето-аппроксимации за заданное число итераций.

Результаты исследования показывают, что для обеспечения высокого качества аппроксимации множества Парето по каждому из рассматриваемых индикаторов качества необходимо использовать специальные алгоритмические решения, например, нишевание, для обеспечения равномерности покрытия. Важно, что эти решения, вообще говоря, различны для различных индикаторов качества. Потому актуальной является задача разработки алгоритмов, «заточенных» под каждый из индикаторов качества Парето-аппроксимации. С другой стороны, наличие значительного числа индикаторов качества Парето-аппроксимации означает многокритериальность задачи оценки этого качества. Таким образом, актуальной является проблема разработки алгоритмов Парето-аппроксимации как многокритериальных алгоритмов.

Ключевые слова: многокритериальная оптимизация; множество Парето; парето-аппроксимация; популяционные алгоритмы; генетические алгоритмы; фронт Парето; качество решения

Введение

В настоящее время интенсивно развивается класс алгоритмов многокритериальной оптимизации (МКО), которые основаны на предварительной конечномерной аппроксимации множества (фронта) Парето МКО-задачи. Называем эту аппроксимацию *Парето-аппроксимацией*. Алгоритмы Парето-аппроксимации строят, преимущественно, с использованием популяционных алгоритмов [1, 2]. Принципиальными для популяционных алгоритмов Парето-аппроксимации (ПП-алгоритмов) является способ конструирования фитнес-функции, которая обеспечивала бы перемещение агентов популяции в направлении множества Парето. При этом в качестве популяционных могут использоваться эволюционные и мета-эвристические алгоритмы, например, алгоритм роя частиц [3, 4], муравьиный алгоритм [5, 6], пчелиный и другие алгоритмы [7, 8].

Работа носит обзорный характер и включает в себя результаты исследования эффективности некоторых ПП-алгоритмов, полученные с участием авторов в период с примерно 2010 года по настоящее время. Работа имеет целью показать на ряде известных тестовых и практически значимых МКО-задач эффективность Парето-аппроксимации с помощью генетического алгоритма, алгоритмов роя частиц, колонии муравьев и колонии пчел в комбинации с различными методами формирования фитнес-функции.

Для оценки качества Парето-аппроксимации могут быть использованы различные индикаторы [9, 10]. В работе в качестве таких индикаторов используется величина, называемая *разреженностью покрытия* (*scarcity*), а также число недоминируемых решений, найденных ПП-алгоритмом за заданное число итераций.

В первом параграфе приведена постановка МКО-задачи. Обзор ПП-алгоритмов представлен во втором параграфе. При этом основное внимание сконцентрировано на правилах конструирования фитнес-функций. Третий параграф представляет примеры решения тестовых и практически значимых задач Парето-аппроксимации с помощью генетического алгоритма, а также с помощью алгоритмов роя частиц, колонии муравьев и колонии пчел.

Если A – некоторый вектор, то принято, что $|A|$ есть размерность этого вектора.

1. Постановка задачи многокритериальной оптимизации (МКО-задачи)

Полагаем, что множеством допустимых значений вектора варьируемых параметров X является ограниченное и замкнутое множество $D_X \subset \{X\} = R^{|X|}$. Вектор-функция критериев $F(X) = (f_1(X), f_2(X), \dots, f_{|F|}(X))$ имеет значения в пространстве критериев $\{F\} = R^{|F|}$. Лицо, принимающее решения (ЛПР), имеет целью минимизировать во множестве D_X каждую из функций $f_1(X), f_2(X), \dots, f_{|F|}(X)$:

$$F(X) \rightarrow \min_{X \in D_X} \quad (1)$$

Множество достижимости задачи (1) обозначаем D_F , фронт Парето - $D_F^* \in D_F$, соответствующее множество Парето - $D_X^* \in D_X$. Полагаем, что решением задачи (1) является множество Парето D_X^* .

2. Обзор популяционных алгоритмов Парето-аппроксимации

ПП-алгоритмы отличаются от не популяционных алгоритмов тем, что реализуем одновременный поиск большого числа недоминируемых точек, близких к паретовским. *Архивными* называем множества Θ^F, Θ^X , содержащие точки $F_j^\Theta \in D_F, X_j^\Theta \in D_X$; $j \in [1:|\Theta|]$. Популяцию агентов обозначаем $S = \{s_i, i \in [1:|S|]\}$. Текущие координаты агента s_i в пространствах $\{X\}, \{F\}$ полагаем равными X_i и $F_i = F(X_i)$ соответственно. Фитнес-функция ПП-алгоритмов должна обеспечивать перемещение агентов популяции S в направлении множества Парето D_X^* (при этом соответствующие точки $\{F_i\}$ перемещаются, очевидно, в направлении фронта Парето D_F^*).

Выделяем классы ПП-алгоритмов, основанные на: 1) лексикографической селекции, 2) чередующихся критериальных функциях, 3) непосредственно использующие доминирование по Парето.

Алгоритм лексикографической турнирной селекции (*lexicographic tournament selection*) требует, чтобы список функций $f_k(X), k \in [1:|F|]$ был упорядочен по важности.

Для простоты записи примем, что этот список имеет вид $f_1(X), f_2(X), \dots$ [10]. Лучшего из агентов $s_i, s_j, i, j \in [1:|S|], i \neq j$ определяем последовательным сравнением величин $(f_k(X_i), f_k(X_j))$ до тех пор, пока не будет установлено, например, неравенство $f_{k_i}(X_i) < f_{k_i}(X_j)$; $k, k_i \in [1:|F|]$. При этом полагаем, что агент s_i имеет лучшую приспособленность, чем агент s_j .

Известно большое число модификаций алгоритма лексикографической турнирной селекции, которые предполагают при определении лучшего агента использование случайно выбранной функции $f_k(X), k \in [1:|F|]$, использование правила голосования, многоуровневой турнирной селекции и т.д.

Алгоритмы чередующихся критериальных функций. К этому классу алгоритмов относят алгоритм переключающихся функций, алгоритм динамического соседства, а также алгоритм «хищник-жертва».

В алгоритме переключающихся функций (*altering objective functions*) выбор лучшего агента производится по схеме, аналогичной схеме алгоритма лексикографической турнирной селекции. Широко известным алгоритмом данного класса является генетический алгоритм *VEGA (Vector Evaluated Genetic Algorithm)* [11].

Пусть величины $|S|, |F|$ кратны и $|S| > |F|$. Приспособленность агентов в алгоритме *VEGA* вычисляют по представленной ниже схеме.

- 1) Разбиваем популяцию S на $|F|$ субпопуляций $S_1, S_2, \dots, S_{|F|}$ случайным образом.
- 2) В качестве фитнес-функции агентов популяции S_j принимаем функцию $f_j(X), j \in [1:|F|]$.

Алгоритм *VEGA* использует в популяции S_j пропорциональную селекцию на основе фитнес-функции $f_j(X), j \in [1:|F|]$. Затем по правилам применяемого генетического алгоритма к отобранными агентами применяются операторы скрещивания и мутации.

Алгоритм динамического соседства (dynamic neighborhood) [11]. Схема выбора лучшего для агента $s_i, i \in [1:|S|]$ решения имеет в данном случае следующий вид.

- 1) Полагаем функцию $f_1(X)$ динамической.
- 2) Вычисляем расстояния (в смысле используемой метрики) от точки F_i до всех точек $F_j^\ominus, j \in [1:|\Theta|]$.
- 3) На основе вычисленных расстояний выбираем m ближайших соседей точки F_i . Здесь m - свободный параметр алгоритма.

- 4) Из числа выбранных точек находим точку F^{j_b} , которая является лучшей с точки зрения данной динамической функции.
- 5) Переходим к следующей итерации, принимая в качестве динамической функцию $f_2(X)$. И так далее.

Алгоритм хищник-жертва (*predator-prey*) [13] исходит из предположения, что множество D_x представляет собой параллелепипед Π . Каждому из векторов решений ставим в соответствие одну из жертв, а каждому хищнику – одну из функций $f_1(X)$, $f_2(X)$,... Популяция хищников и жертв эволюционирует на тороидальной сетке.

Идея алгоритма заключается в следующем. Каждый хищник рассматривает все жертвы в своей окрестности и «съедает» жертву с худшим значением соответствующей функции $f_1(X)$, $f_2(X)$,... Далее в рассматриваемой окрестности выбираем и некоторым образом модифицируем случайную жертву. Результат модификации помещаем на место «съеденной» жертвы. Затем хищник случайным образом перемещается в один из соседних с ним узлов сетки. Указанная последовательность действий итерационно повторяется до выполнения условий окончания.

Недостаток представленного алгоритма хищник-жертва заключается в возможности потери лучших решений. Известен ряд модификаций алгоритма, которые обеспечивают его более высокую эффективность [14].

Алгоритмы на основе ранжирования агентов. Алгоритмы этого класса используют понятие *ранга* агента популяции. Основные из известных правил вычисления рангов представлены ниже.

Алгоритм недоминируемой сортировки (Non-Dominated Sorting, NDS). Данный алгоритм использован в широко известном генетическом алгоритме *NSGA (Non-Dominated Sorting Genetic Algorithm)* [11]. Алгоритм использует следующее правило вычисления рангов.

- 1) Недоминируемым агентам популяции присваиваем ранг r , равный единице, и исключаем из рассмотрения.
- 2) Оставшимся недоминируемым агентам присваиваем ранг r , равный двум, и исключаем из рассмотрения. Процесс повторяем до исчерпания популяции.

Приспособленность индивида s_i , например, вычисляем по формуле вида

$$\varphi(X_i) = \frac{1}{1 + r_i}, \quad i \in [1 : |S|].$$

Здесь r_i - ранг индивида s_i .

Известным алгоритмом рассматриваемого класса является *алгоритм MOGA (Multi-Objective Genetic Algorithm)*, который в некоторых публикациях называют алгоритмом *FFGA (Fonseca and Fleming's Multiobjective Genetic Algorithm)* [11]. Ранг r_i агента s_i в этом алгоритме равен числу агентов в популяции, которые его доминирует, плюс единица:

$$r_i = 1 + \left| \{s_j \mid X_j \succ X_i, s_j \in S\} \right|, \quad i \in [1 : |S|].$$

Высоко эффективным алгоритмом на основе ранжирования агентов является *генетический алгоритм Парето-аппроксимации с нишеванием (Niche-Pareto Genetic Algorithm, NPGA)* [15]. Для поддержания разнообразия популяции алгоритм использует механизм нишевания [15].

Для обеспечения высокого Парето-аппроксимации качества при формировании архивов Θ^X , Θ^F можно исключать из рассмотрения агентов, расстояние которых до архивных элементов меньше заданной величины. Такой прием использован в генетическом алгоритме Парето-аппроксимации *NGSA-II (Non-Dominated Sorting Genetic Algorithm II)* [16].

Алгоритм Парето-силы (Pareto strength) [17] использует понятия *силы* агента и его *хилости (wimpiness)*. Первая величина равна числу агентов, которых доминирует по Парето данный агент. Хилость агента w_i равна суммарной силе всех агентов, доминирующих данный. Приспособленность агентов при этом вычисляем по формуле вида

$$\varphi_i = \frac{1}{(1 + w_i)}, \quad i \in [1 : |S|].$$

Алгоритм Парето силы реализует известный ПП-алгоритм *SPEA (Strength Pareto Evolutionary Algorithm)* [17], развитием которого является алгоритм *SPEA-2* [18]. Упрощенным вариантом последнего алгоритма можно считать алгоритм *PAES (Pareto Achieved Evolution Strategy)* [19].

Известно значительное число других алгоритмов, основанных на идее ранжирования: алгоритм средневзвешенного ранжирования (*Weighted Average Ranking, WAR*) [20]; алгоритм суммы взвешенных оценок (*Sum of Weighted Ratios, SWR*) [26]; алгоритм суммы взвешенных глобальных оценок (*Sum of Weighted Global Ratios, SWGR*) [26]; алгоритм максимального взвешенного ранжирования (*Weighted Maximum Ranking, WMR*) [26].

Достаточно большой класс ПП-алгоритмов образуют алгоритмы, не использующие ранжирование агентов. Наиболее известными представителями этого класса алгоритмов являются сигма-алгоритм, алгоритм композитных точек, алгоритм гиперкубов [1].

3. Экспериментальная оценка эффективности некоторых методов Парето-аппроксимации

3.1. Генетический алгоритм NSGA II. Тестовые задачи ZDT1 – ZDT4

Эффективность Парето-аппроксимации на основе генетического алгоритма иллюстрируем на примере алгоритма *NSGA II* [16]. Рассматриваем задачи двухкритериальной оптимизации *ZDT1 – ZDT4* [21]. Полученные Парето-аппроксимации, представлены на рисунках 1 – 4 соответственно. Всюду на рисунках сплошная линия показывает точный фронт Парето, а крестики – полученное приближения к нему. Если не оговорено иное, полагается, что число итераций алгоритма равно $\hat{t} = 500$.

Задача ZDT1. Множество D^X представляет собой параллелепипед

$$P = \{X \mid x_1 \in [0; 1], x_i \in [-1; 1], i \in [2 : |X|]\}, \quad (3)$$

где $|X| = 30$. Рисунок 1 показывает, что алгоритм *NSGA II* обеспечивает для этой задачи высокое качество Парето-аппроксимации.

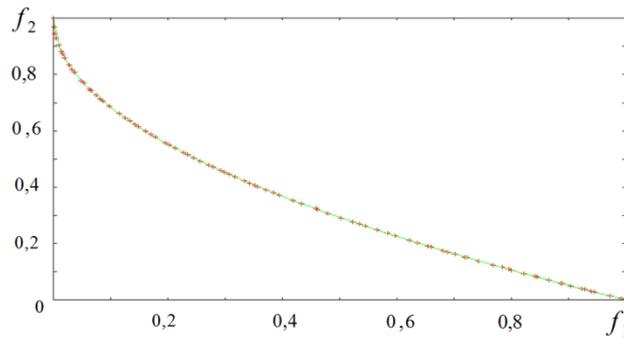


Рисунок 1. Парето-аппроксимация: задача ZDT1; алгоритм NSGA II [2]

Задача ZDT2. Множество D_x в данном случае также имеет вид параллелепипеда (4); $|X| = 30$. Задача является достаточно трудной с точки зрения получения Парето-аппроксимации ее фронта Парето в силу невыпуклости этого фронта. Рисунок 2 показывает высокую эффективность алгоритма NSGA II и в данном случае.

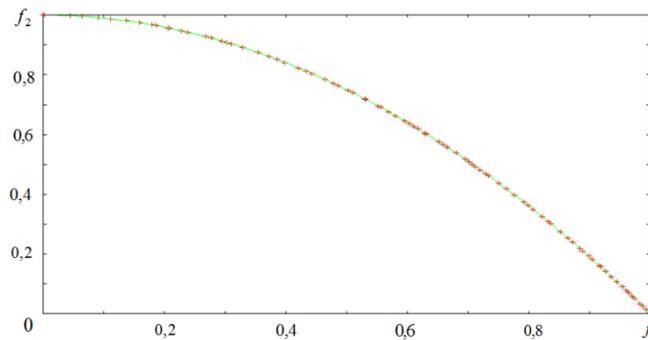


Рисунок 2. Парето-аппроксимация: задача ZDT2; алгоритм NSGA II [2]

Задача ZDT3 (множество D_x есть параллелепипед (4); $|X| = 30$). Результаты решения задачи представлены на рисунке 3. Задача имеет разрывный, хотя и выпуклый, фронт Парето и поэтому является сложной для многих алгоритмов Парето-аппроксимации. Из рисунка 3 вытекает, что алгоритм NSGA II и в данном случае показывает достаточно высокое качество аппроксимации.

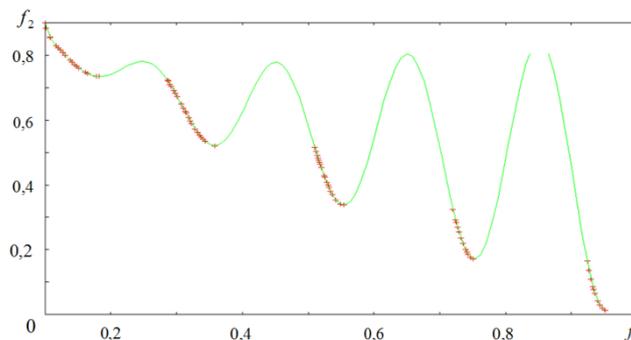


Рисунок 3. Парето-аппроксимация: задача ZDT3; алгоритм NSGA II [2]

Задача ZDT4. Множество D_x в этой задаче есть параллелепипед

$$D = \{X \mid x_1 \in [0; 1], x_i \in [-5; 5], i \in [2 : |X|]\}$$

при $|X| = 10$. Особенностью задачи является наличие локальных субоптимальных фронтов, которые могут остановить Парето-аппроксимацию вдали от истинного фронта (рисунок 4).

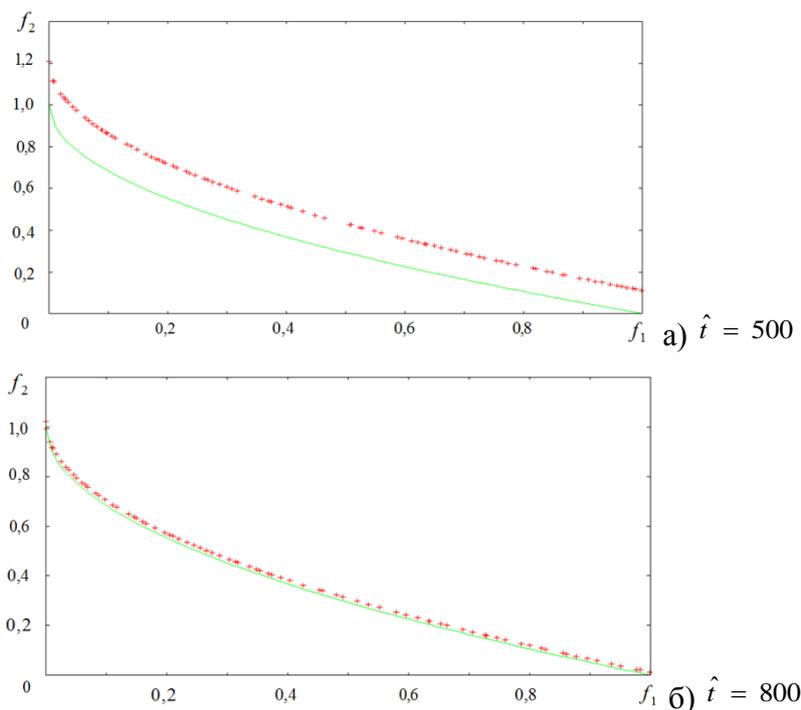


Рисунок 4. Парето-аппроксимация: задача ZDT4; алгоритм NSGA II; [2]

Рисунок 4,а показывает, что 500 итераций алгоритма NSGA II позволяют получить аппроксимацию, в которой нет решений, находящихся вблизи истинного фронта Парето. При увеличении числа итераций до 800, получается аппроксимация, близкая к истинному фронту Парето (рисунок 4,б), хотя, как и при $\hat{t} = 500$, не найдено решений, точно лежащих на фронте.

В целом, представленные результаты показывают, что алгоритм NSGA – II позволяет даже для сложных МКО-задач обеспечить приемлемую точность, а также относительно невысокие вычислительные затраты.

3.2. Генетический алгоритма NSGA II. Задача об управляемом спуске космического аппарата

В соглашениях работы [22], математическая модель управляемого космического аппарата (КА) имеет вид следующей системы обыкновенных дифференциальных уравнений (ОДУ)

$$\begin{aligned} \frac{dx_1}{d\tau} &= x_2, \\ \frac{dx_2}{d\tau} &= -\frac{\mu}{r^3} x_1 - (1-u) q x_2, \\ \frac{dx_3}{d\tau} &= x_4, \\ \frac{dx_4}{d\tau} &= -\frac{\mu}{r^3} x_3 - (1-u) q x_4 \end{aligned} \quad (4)$$

при начальных условиях $x_1(0) = 6471$ км, $x_3(0) = 0$, $x_2(0) = 0,1$ км/с, $x_4(0) = 7$ км/с. Здесь x_1, x_3 - координаты центра масс КА, x_2, x_4 - компоненты скорости КА; u - скалярное управление; $r = \sqrt{x_1^2 + x_3^2}$ - радиус-вектор КА; μ - гравитационная постоянная Земли; $q = c e^{\frac{h-x_1}{\eta}} v$ - компонента аэродинамической силы; $v = \sqrt{x_2^2 + x_4^2}$ - скорость КА; c - аэродинамическая характеристика КА; h - высота атмосферы; η - известный коэффициент.

В качестве функционалов качества управления рассматриваем функционалы

$$f_1(u) = \max_{\tau \in [0; \hat{\tau}]} \frac{(x_1^2(\tau) + x_3^2(\tau) \sqrt{x_2^2(\tau) + x_4^2(\tau)})}{g_0 r_3^2} \rightarrow \min_{u \in D_u}, \quad (5)$$

$$f_2(u) = x_3(\hat{\tau}) - l \rightarrow \min_{u \in D_u}, \quad (6)$$

первый из которых имеет смысл максимальной перегрузки КА, второй - его отклонения от заданной точки посадки. В формулах (5), (6) приняты обозначения: $r_3 = 6371$ км - радиус Земли; $g_0 = 9,81$ м/с² - ускорение свободного падения; $\hat{\tau}$ - длительность полета; l - координата заданной точки посадки; $D_u = \{u(\tau) \mid |u(\tau)| \leq u^+\}$ - совокупность допустимых управлений; $u^+ > 0$ - заданная константа.

Используем метод сведения двухкритериальной задачи оптимального управления (4) - (6) к задаче нелинейного программирования. С этой целью покрываем интервал $[0; \hat{\tau}]$ сеткой с узлами $\tau_i, i \in [0; |U|]$. Оптимальное управление $u^*(\tau)$ отыскиваем в классе кусочно-постоянных функций. Пусть $U = (u_1, u_2, \dots, u_{|U|})$ - $(|U| \times 1)$ -вектор, где $u_i = u(\tau_i)$. В этих соглашениях и обозначениях задача (4) - (6) представляет собой двухкритериальную

задачу оптимизации с критериями $f_1(U)$, $f_2(U)$, где вектор варьируемых параметров $U \in D_U$; $D_U = \{u_i \mid |u_i| \leq u^+, i \in [1:|U|]\}$.

Для решения полученной МКО-задачи применяем вещественный генетический алгоритм, в котором хромосома есть совокупность $|U|$ генов, где ген H_i кодирует компоненту u_i вектора U ; $i \in [1:|U|]$. Алгоритм использует оператор случайной мутации, вероятность которой равна $\xi_m = 0,95$; оператор скрещивания реализует расширенный линейный кроссинговер; в операторе селекции применяется метод отбора на основе элитизма.

Избранные результаты вычислений представлены на рисунке 5. Рисунок показывает, что рассматриваемый алгоритм не обеспечивает высокую плотность покрытия фронта Парето и высокую скорость сходимости. При $\hat{t} = 100$ данный алгоритм смог найти 12 недоминируемых решений (рисунок 5,а), а при $\hat{t} = 1000$ – всего 14 таких решений (рисунок 5,б). Подчеркнем, что найденные решения достаточно равномерно распределены по фронту Парето.

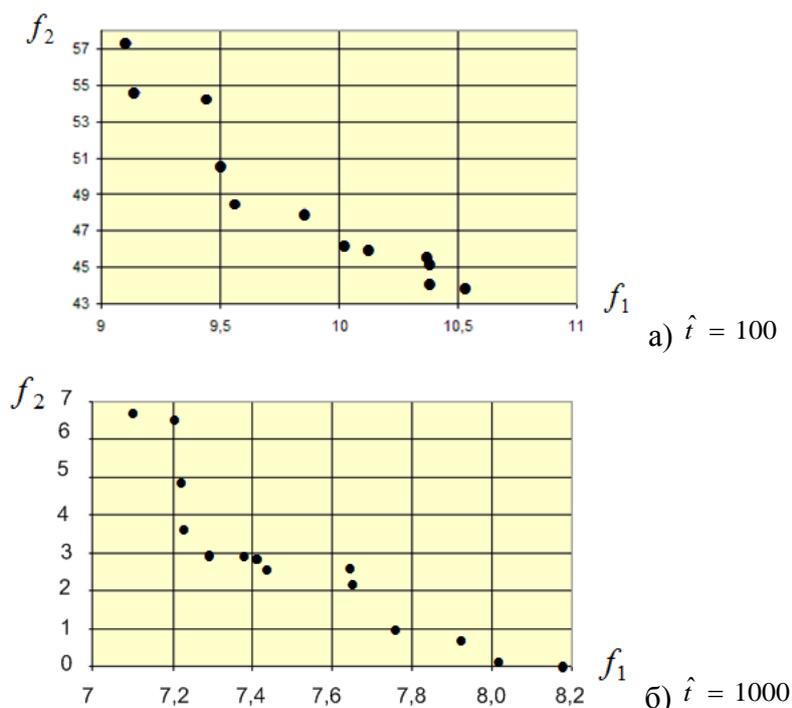


Рисунок 5. Аппроксимация фронта Парето задачи об управляемом спуске КА [2]

3.3. Генетический алгоритма NPGA. Тестовые задачи многокритериальной оптимизации

Рассматриваем модификацию алгоритма NPGA, которая предложена в работе [23]. Модифицированный алгоритм использует ранжирование индивидов на основе Парето-доминирования, аутбридинг вместо турнирной селекции [11], операцию клонирования индивидов.

Эффективность алгоритма исследована на тестовых задачах ZDT3, ZDT6, DTLZ4 (п.3.1). При решении всех этих задач использованы следующие значения свободных

параметров алгоритма [24]: параметр разделения $\sigma_{share} = 0,01$; доля отбираемых для скрещивания индивидов $T_r = 0,3$; параметр рекомбинации $CrossoverRate=0,7$; мутационный параметр $MutationRate=1,0$; $|S| = 1000$; $\hat{t} = 1000$.

Результаты исследования иллюстрируют рисунки 6 – 8, на которых представлены точные фронты Парето (сплошные линии) и их полученные аппроксимации.

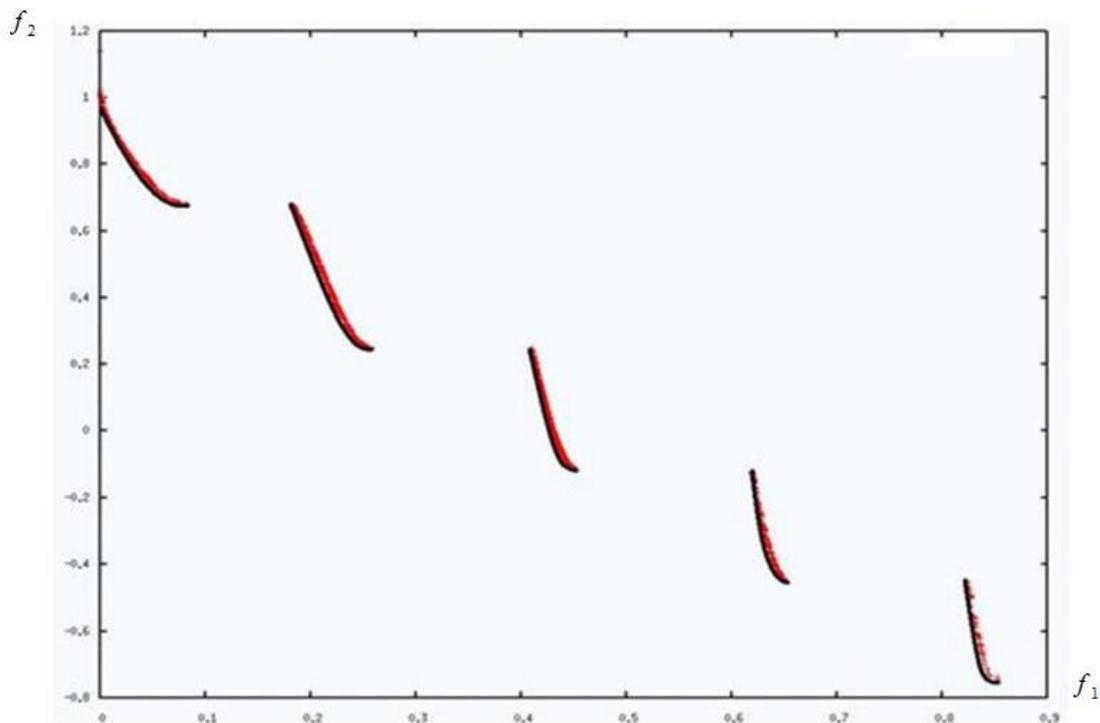


Рисунок 6. Аппроксимация фронта Парето алгоритмом NPGA: тестовая задача ZDT3 [24]

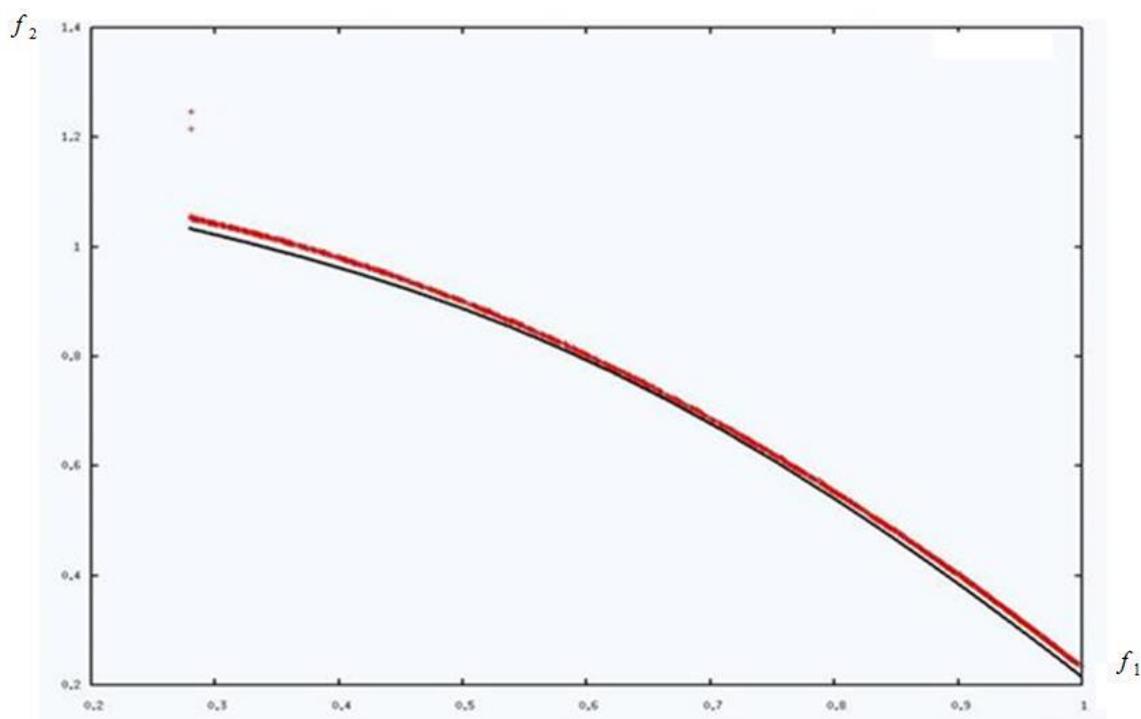


Рисунок 7. Аппроксимация фронта Парето алгоритмом NPGA: тестовая задача ZDT6 [24]

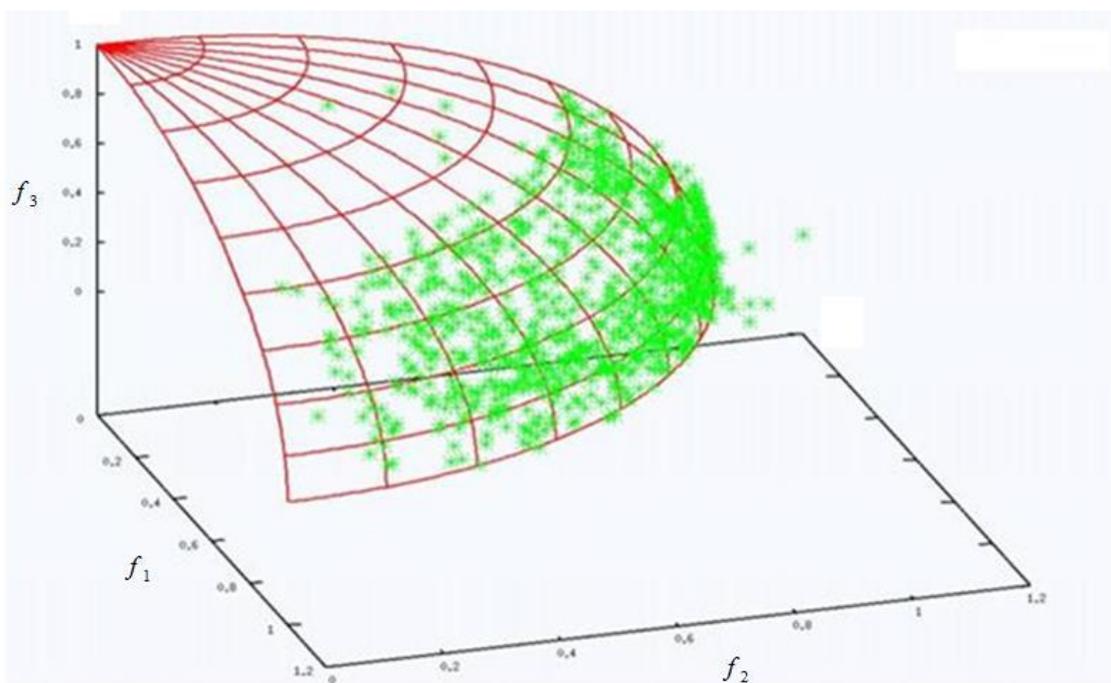


Рисунок 8. Аппроксимация фронта Парето алгоритмом NPGA: тестовая задача DTLZ4 [24]

Представленные результаты вычислительного эксперимента показывают, что модифицированный алгоритм NPGA позволяет получить хорошие решения для тестовых задач ZDT3, ZDT6 и удовлетворительные результаты – для задачи DTLZ 4 .

3.4. Генетический алгоритма NPGA. Задача многокритериальной оптимизации автомобильной коробки переключения передач

Рассматриваем указанную в предыдущем пункте модификацию алгоритма NPGA. Используемая математическая модель коробки переключения передач автомобиля в виде схемы редактора Qucs системы PRADIS [24] приведена на рисунке 9, где DVTBLUI – двигатель, MUFTA1 – муфта сцепления, MGEAR11 - коробка переключения передач, DIFMC1 - дифференциал, M1 и M2 - колеса.

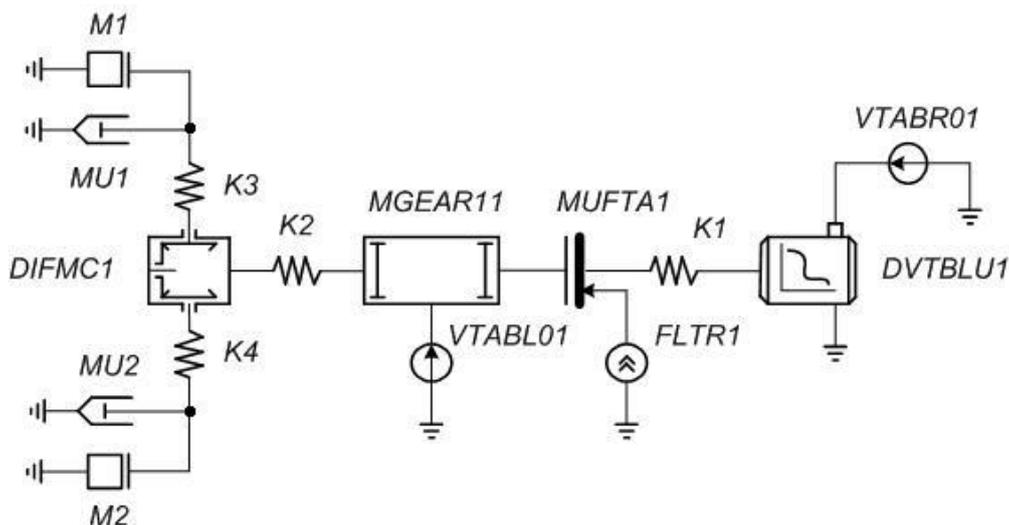


Рисунок 9. Схема математической модели коробки переключения передач (редактор Qucs) [24]

В представленной модели варьируемыми являются следующие параметры: N1, N2, N3, N4, N5 - передаточные отношения соответствующих передач; T1 – время сброса сцепления; T2, T3, T4, T5 – времена включения соответствующих передач.

Множество D_x представляет собой параллелепипед, определяемый ограничениями

$$1 \leq N_i \leq 20, N_i \in Z, i \in [1 : 5],$$

$$0 \leq T_1 \leq 1, T_1 \in R^1,$$

$$1 \leq T_i \leq 20, T_i \in R^1, i \in [2 : 5],$$

и стесненный неравенствами $T_2 < T_3 < T_4 < T_5$.

Выходными являются следующие переменные: Time60 – время разгона до 60 км/ч; Time100 – время разгона до 100 км/ч; Velocity – скорость автомобиля; MaxTorque – максимальный момент в трансмиссии; Torque – момент в трансмиссии; Torque_Engine – момент на валу двигателя; V_Engine – частота вращения вала; Out_Gear – частота вращения выходного вала коробки переключения передач; Efficiency – КПД трансмиссии.

Рассматриваем следующие критерии оптимальности:

- максимальный момент в трансмиссии MaxTorque $\rightarrow \max$;
- максимальная скорость автомобиля VelocityMax $\rightarrow \max$;
- время разгона до 100 км/ч Time100 $\rightarrow \min$;
- время разгона с 60 до 100 км/ч Time60_Time100 $\rightarrow \min$.

С использованием этих критериев сформированы три следующие МКО-задачи:

- 1) $f_1 = -\text{MaxTorque}$, $f_2 = \text{VelocityMax}$; 2) $f_1 = -\text{MaxTorque}$, $f_2 = \text{VelocityMax}$, $f_3 = -\text{Time100}$;
- 3) $f_1 = -\text{MaxTorque}$, $f_2 = \text{VelocityMax}$, $f_3 = -\text{Time100}$, $f_4 = -\text{Time60_Time100}$.

Парето-аппроксимации этих задач получены с использованием программной системы PRADIS//FRONT [24]. Результаты решения первой задачи представлены на рисунке 10, а второй задачи – на рисунке 11. Из рисунков следует, что для этих (достаточно сложных задач) алгоритм NPGA позволяет получить удовлетворительное качество аппроксимации фронта Парето, особенно, в его средней части.

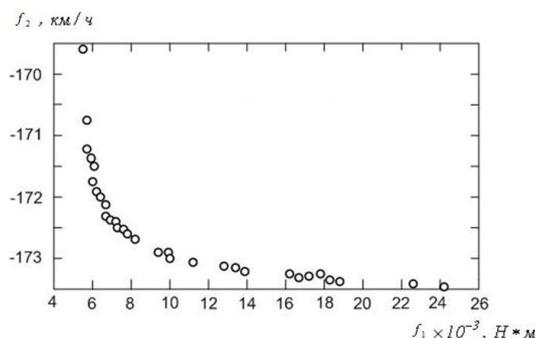


Рисунок 10. Парето-аппроксимация двухкритериальной задачи: алгоритм NPGA

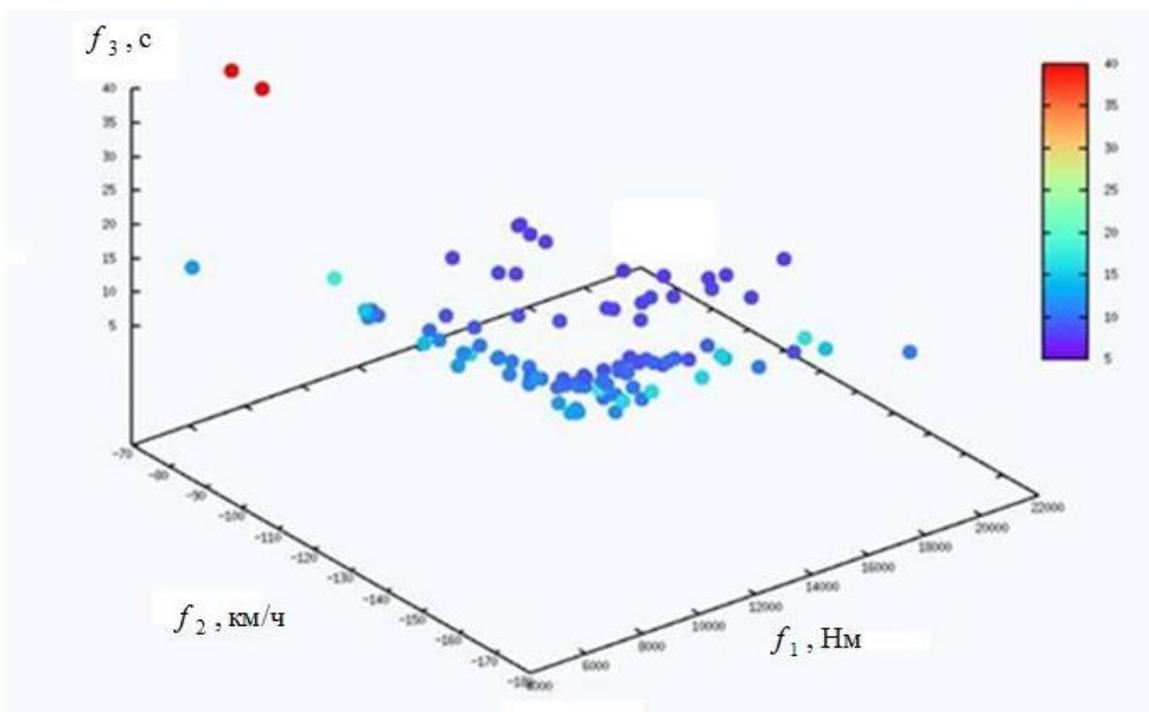


Рисунок 11. Парето-аппроксимация трехкритериальной задачи: алгоритм NPGA [24]

3.5. Когенетический алгоритм. Тестовые задачи многокритериальной оптимизации

Рассматриваем *генетический коэволюционный алгоритм (Generic Co Evolutionary algorithm, GCEA)* [25], использующий недоминируемую сортировку. Оригинальность алгоритма состоит в том, что его субалгоритмы имеют различные наборы эволюционных

операторов. Основные параметры алгоритма: $|\mathbf{S}|$ - число субпопуляций; n_f - максимально допустимое число вычислений значений рассматриваемых критериев оптимальности; t_a - интервал адаптации; n_p - величина штрафа, который назначается проигравшим субпопуляциям; $|S|_{\min} = \min_i |S_i|, i \in [1 : |\mathbf{S}|]$ - минимально допустимый размер субпопуляции.

Субалгоритмы могут использовать следующие наборы эволюционных операторов: оператор рекомбинации - инбридинг и аутбридинг; оператор скрещивания - расширенный линейный оператор, BLX-оператор SBX-оператор, эвристический оператор; оператор мутации - оператор случайной мутации; оператор селекции – операторы турнирной и элитарной селекции [26].

Число субпопуляций в исследовании принято равным $|\mathbf{S}| = 4$, величина интервала адаптации - $t_a = 5$, величина штрафа - $n_p = 0,1$, минимально допустимый размер субпопуляции - $|S|_{\min} = 0,12 |S|$, вероятность мутации - $\xi_m = 0,05$. Для оценки качества аппроксимации используем следующие индикаторы [27]: равномерность распределения решений I_s ; число найденных паретовских решений N_{Pareto} ; общее число испытаний N_{fit} . Исследование выполнено для двухкритериальных задач ZDT1 - ZDT4 (п.3.1). Для задач

ZDT1 - ZDT3 использованы размеры субпопуляций, равные $|S| = 20, 50, 100$, и числа поколений, равные $\hat{t} = 10, 25, 50$. Поскольку задача ZDT4 является мультимодальной, при ее исследовании использованы также размеры субпопуляций $|S| = 200, 400$ и $\hat{t} = 50, 100, 1000$.

Задача ZDT1. Исследование выполнено для алгоритма CGA, включающего в себя четыре субалгоритма (рисунок 12). Результаты исследования показывают, что CGA-алгоритм по индикатору I_s до ~13%, а по индикатору N_{Pareto} до ~100% эффективнее генетического алгоритма. С другой стороны, с точки зрения индикатора N_{fit} , алгоритм CGA почти в четыре раза проигрывает генетическому алгоритму (данный вывод относится и ко всем тестовым задачам, рассмотренным ниже в данном пункте).

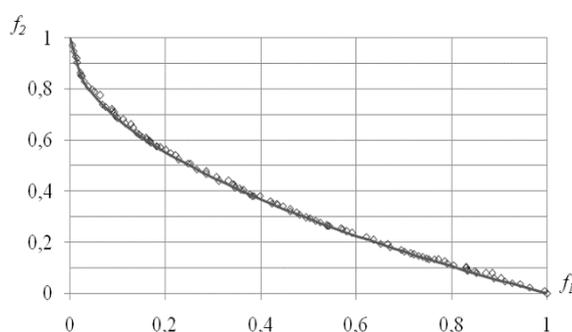


Рисунок 12. Парето-аппроксимация задачи ZDT1; $|S| = 100$; $\hat{t} = 50$ [25]

Задача ZDT2 (рисунок 13). Исследование показывает некоторое преимущество алгоритма CGA над генетическим алгоритмом по индикатору I_s . По индикатору N_{Pareto} алгоритм CGA до примерно 95% эффективнее генетического алгоритма.

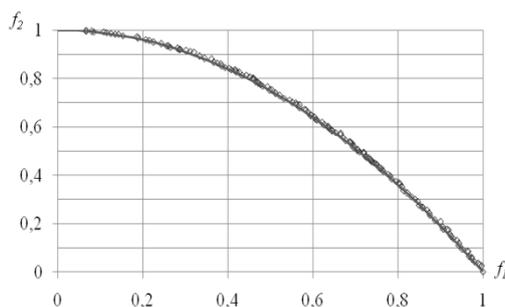


Рисунок 13. Парето-аппроксимация: задача ZDT2; $|S| = 100$; $\hat{t} = 50$ [25]

Задача ZDT3 (рисунок 14). Для данной задачи алгоритм CGA во всех случаях показывает по индикаторам I_s , N_{Pareto} более высокую эффективность, чем генетический алгоритм.

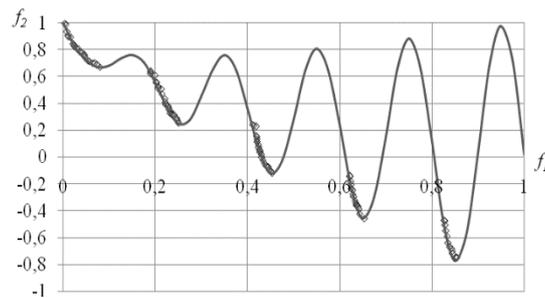


Рисунок 14. Парето-аппроксимация: задача ZDT3; $|S| = 100$; $\hat{t} = 50$ [25]

Задача ZDT4. Рисунок 15 показывает, что с увеличением численности популяции, Парето-аппроксимации, полученные алгоритмом CGA, становятся более равномерным и оказываются способными преодолевать локальные фронты Парето. Генетический алгоритм, напротив, не смог преодолеть эти фронты и приблизиться к точному фронту Парето. Как и для задач ZDT1 - ZDT3, алгоритм CGA показал превосходство над генетическим алгоритмом по индикаторам I_s , N_{Pareto} , но уступил этому алгоритму по индикатору N_{fit} .

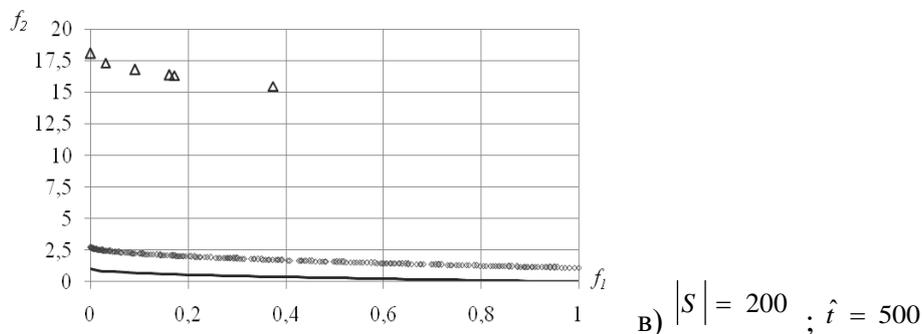
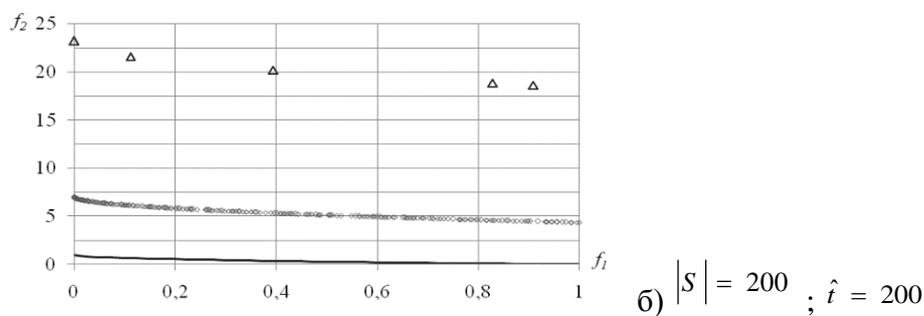
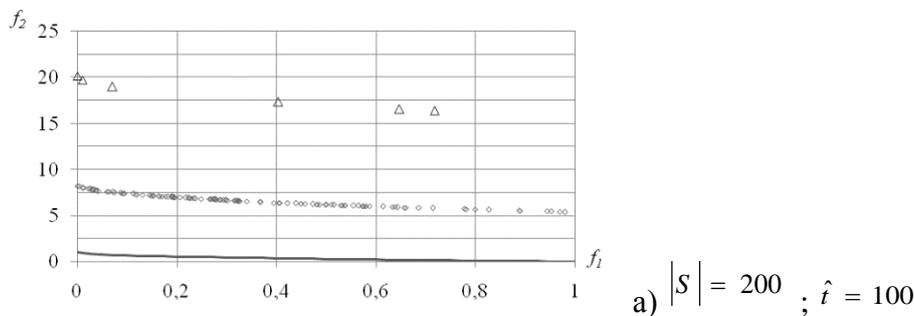


Рисунок 15. Парето-аппроксимация: задача ZDT4;
 \diamond - коалгоритм; Δ - генетический алгоритм [25]

3.6. Когенетический алгоритм. Двухкритериальная задача управления спуском космического аппарата в атмосфере Земли

Постановка данной задачи, а также схеме сведения ее к двухкритериальной задаче оптимизации представлены в п. 3.2.

Размерность вектора варьируемых параметров $|U|$ принята равной 100, величину $|S|$ варьируем в диапазоне от 100 до 200, величину \hat{t} - от 5 до 100; $|S| = 4$, $t_a = 5$, $n_p = 0,1$, $|S|_{\min} = 0,12 \cdot |S|$, $\xi_m = 0,05$ (см. п. 3.5).

Результаты решения задачи иллюстрирует рисунок 16, показывающий значительное улучшение качества Парето-аппроксимации с ростом числа поколений, хотя число найденных паретовских точек остается небольшим.

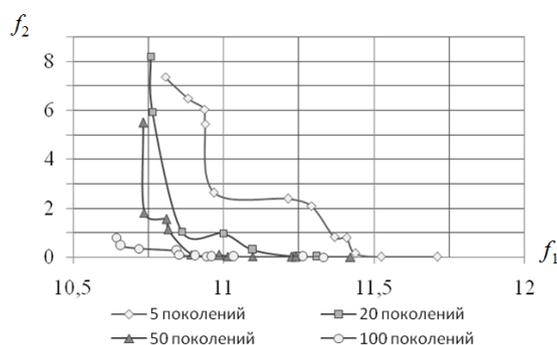


Рисунок 16. Парето-аппроксимация алгоритмом CGA: задача о спуске КА [25]

3.7. Парето-аппроксимация с помощью алгоритма роя частиц. Простейшая тестовая задача двухкритериальной оптимизации

Рассматриваем эффективность алгоритмов Парето-аппроксимации *MOPSO* и *MS MOPSO* [28] при решении простейшей двумерной ($|X| = 2$) двухкритериальной ($|F| = 2$) задачи оптимизации

$$f_1(X) = x_1^2 + x_2^2 \rightarrow \min_{X \in D_X}, \quad f_2(X) = (x_1 - 1)^2 + (x_2 - 1)^2 \rightarrow \min_{X \in D_X}, \quad (7)$$

$$D_X = \{X \mid x_1, x_2 \in [-1; 1]\}. \quad (8)$$

Алгоритм *MOPSO* использует канонический алгоритм роя частиц [3] и алгоритм динамического соседства [11].

В многороевом алгоритме *MOPSO* (*MS MOPSO*) параллельно (может быть, только на логическом уровне) функционирует $|S|$ роев частиц S_j . Каждый из этих роев имеет свои локальные архивы Θ_j^X, Θ_j^F . Алгоритм использует также глобальные архивы Θ^X, Θ^F . Суть алгоритма заключается в использовании декомпозиции множества D_X на $|S|$ подобластей D_X^j , $j \in [1 : |S|]$. Чтобы агенты популяции S_j не покидали свою подобласть, применяем метод штрафных функций.

Алгоритм MOPSO. Для решения указанной задачи использован рой из 10 частиц; $\hat{t} = 100$. После завершения указанного числа итераций, в архивах Θ^X, Θ^F оказалось по 315 решений (рисунок 17).

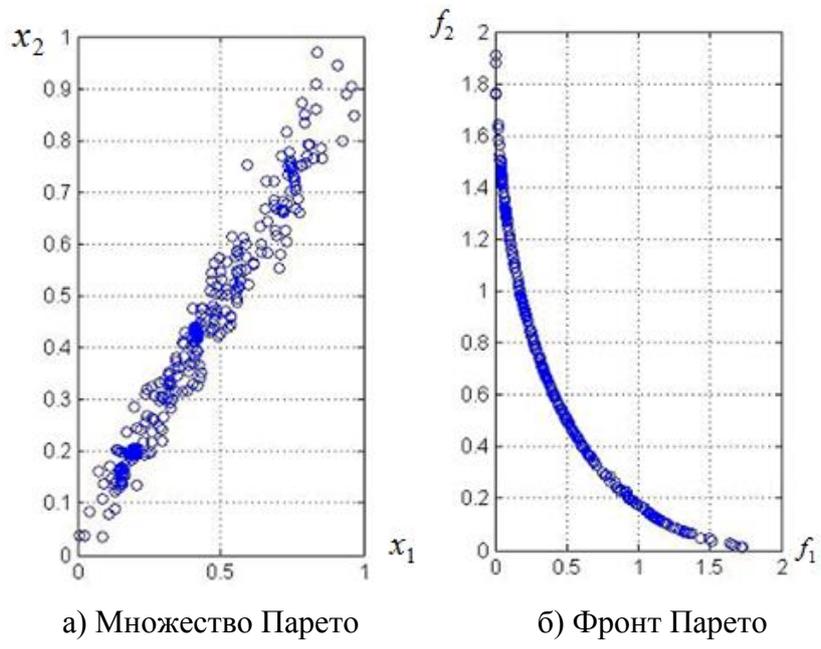


Рисунок 17. Результаты вычислительного эксперимента: задача (7), (8); алгоритм *MOPSO* [2]

Рисунок 17 показывает следующее: даже для простейшей МКО-задачи (7), (8) алгоритм *MOPSO* позволяет получить необходимые плотности покрытия множества Парето и фронта Парето только в их средних частях.

Алгоритм MS MOPSO (рисунок 18). Использовано 100 роев из 10 частиц в каждом; $\hat{t} = 100$; период миграции $t_{mig} = 1$. После завершения итераций в архивах Θ^X, Θ^F оказалось по 456 решений, то есть алгоритм *MS MOPSO* обеспечил высокую плотность и равномерность покрытия всех частей множества и фронта Парето. Ценой этого явились высокие вычислительные затраты.

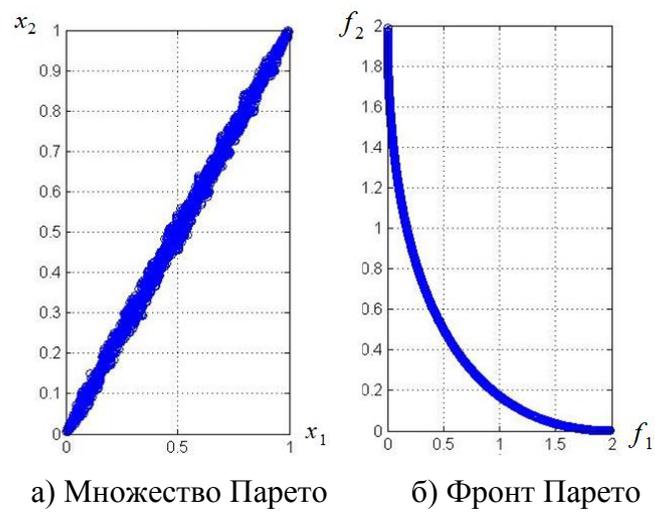


Рисунок 18. Результаты вычислительного эксперимента: задача (7), (8); алгоритм *MS MOPSO* [2]

Алгоритмы MOPSO, MS MOPSO с нишеванием. Использована простейшая схема нишевания: недоминируемое решение помещается в архивы Θ^X, Θ^F (алгоритм MOPSO) или в архивы Θ_j^X, Θ_j^F (алгоритм MS MOPSO), только если в окрестности данного решения заданного радиуса r_x в архиве не имеется других решений.

Рисунки 19, 20 иллюстрируют эффективность модифицированных алгоритмов MOPSO и MS MOPSO соответственно для случая $r_x = 0,001$. Значения остальных параметров приведены выше. После $\hat{t} = 100$ итераций мощность архивного множества для алгоритма MOPSO оказалось равной 62, а для алгоритма MS MOPSO – 172. Представленные результаты вычислительного эксперимента показывают, что нишевание для обоих алгоритмов позволяет получить высокую равномерность покрытия множества и фронта Парето. Важно, что при этом суммарное число вычислений значений критериальных функций уменьшается в несколько раз. С другой стороны, значительно понижается точность аппроксимации, особенно множества Парето.

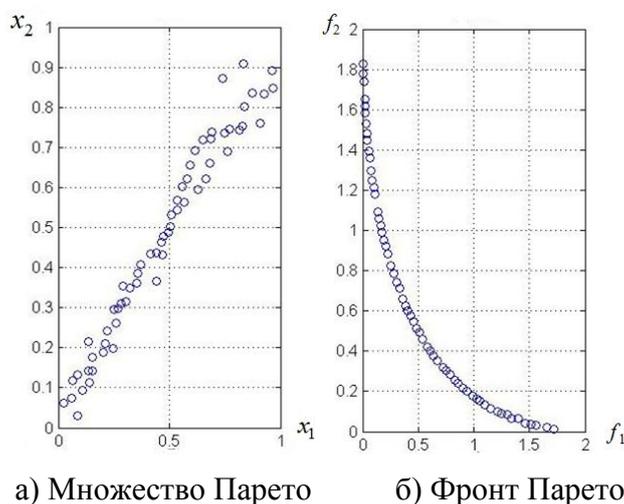


Рисунок 19. Результаты вычислительного эксперимента: задача (7), (8); алгоритм MOPSO с нишеванием [2]

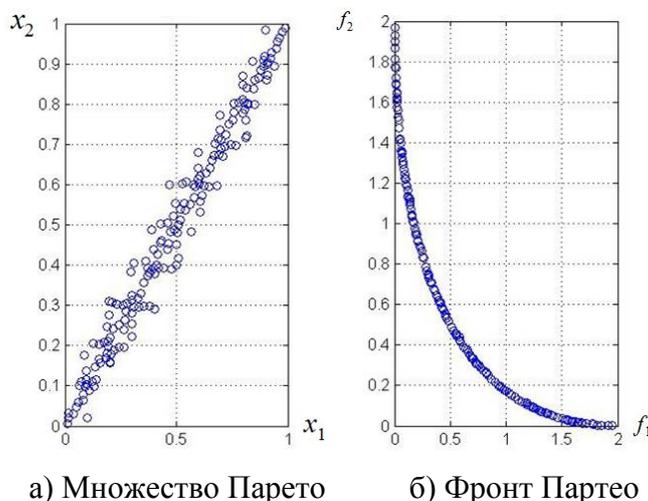


Рисунок 20. Результаты вычислительного эксперимента: задача (7), (8); алгоритм MS MOPSO с нишеванием [2]

3.8. Парето-аппроксимация с помощью алгоритма роя частиц. Двухкритериальная задача оптимального управления

Рассматриваем двумерную динамическую систему, эволюцию которой во времени определяет система ОДУ

$$\begin{cases} \dot{x}_1(\tau) = x_2(\tau), & x_1(0) = 0, \\ \dot{x}_2(\tau) = -\frac{\kappa}{m} x_2(\tau) + \frac{1}{m} u(\tau), & x_2(0) = 0. \end{cases} \quad (9)$$

Здесь κ , m – заданные константы; $u(\tau)$ – скалярное управление. Множество допустимых управлений имеет вид

$$D_u = \{u(\tau) \mid \text{abs}(u(\tau)) \leq a\} \quad (10)$$

Заданы два следующих критерия оптимальности функционирования системы:

$$f_1(u) = \int_0^{\hat{t}} (x_1(\tau) - \phi(\tau))^2 d\tau \rightarrow \min_{u(\tau) \in D_u}; \quad (11)$$

$$f_2(u) = \gamma \int_0^{\hat{t}} u^2(\tau) d\tau \rightarrow \min_{u(\tau) \in D_u}. \quad (12)$$

Подобно п. 3.2, используем метод сведения двухкритериальной задачи оптимального управления (9) – (13) к двухкритериальной $|U|$ -мерной задаче оптимизации

$$\min_{U \in D_U} F(U) = F(U^*) \quad (13)$$

где $F(U) = (f_1(U), f_2(U))$ и

$$D_U = \{u_i \mid \text{abs}(u_i) \leq a, i \in [1 : |U|]\} \quad (14)$$

Парето-аппроксимация задачи (13), (14) получена с помощью алгоритма *MOPSO* (без нишевания). Использованы значения параметров задачи $m = 1$; $\kappa = 1$; $\gamma = 1$; $a = 20$; $\hat{t} = 10$; $\phi(\tau) = \tau^2$; $|U| = 10$ и рой из 40 частиц.

Результаты вычислительного эксперимента иллюстрирует рисунок 21, который показывает удовлетворительную равномерность покрытия.

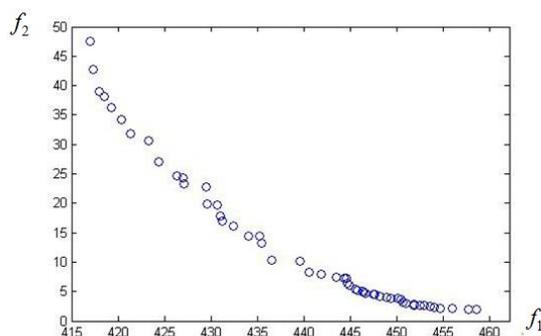


Рисунок 21. Парето-аппроксимации задачи (13), (14): $\hat{t} = 800$ [2]

3.9. Аппроксимация на основе алгоритма колонии муравьев

В данном пункте представляем Парето-аппроксимацию двухмерной двухкритериальной задачи (п. 3.7) комбинированным алгоритмом *CIAC* [5] и динамического соседства [11]. На рисунке 22 представлены результаты вычислений для случая использования алгоритмом *CIAC* прямого канала обмена данными и канала стигмертии. Рисунок 23 иллюстрирует случай использования только канала стигмертии.

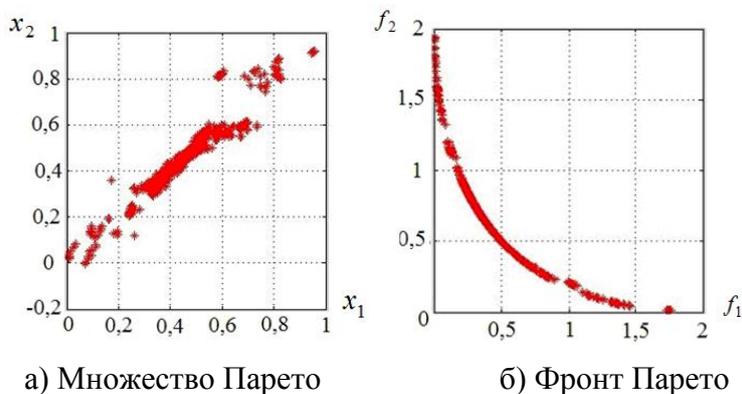


Рисунок 22. Результаты вычислительного эксперимента: задача (7), (8); комбинация алгоритмов *CIAC* (оба канала) и динамического соседства [6]

Исследование выполнено при следующих значениях параметров алгоритма *CIAC*: число агентов в популяции 186; максимальное число сообщений в памяти муравья $m_{\max} = 5$; знаменатель размерного параметра $b_r = 50$; число итераций $\hat{t} = 200$ [29].

Из рисунков 22, 23 следует, что без нишевания комбинация алгоритмов *CIAC* и динамического соседства позволяет обеспечить удовлетворительную равномерность покрытия лишь при использовании популяции большого размера и значительного числа итераций.

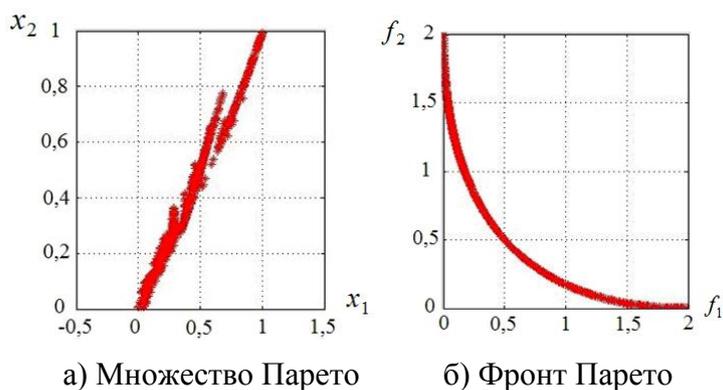


Рисунок 23. Результаты вычислительного эксперимента: задача (7), (8); комбинация алгоритмов *CIAC* (прямой канал) и динамического соседства [6]

3.10. Аппроксимация на основе пчелиного алгоритма

Простейшая двухкритериальная задача (п. 3.7). Рассматриваем комбинацию алгоритмов пчелиного роя [5] и недоминируемой сортировки, предназначенную для решения задачи Парето-аппроксимации [30]. Вычислительный эксперимент выполнен при следующих значениях параметров комбинированного алгоритма: числа пчел-разведчиков, элитных

участков и рекрутированных пчел равны $|S^o| = 200$, $|A^b| = 40$, $n^b = 60$ соответственно [30]. Из рисунка 24 следует, что алгоритм обладает достаточно высокой скоростью сходимости итераций: за $\hat{t} = 10$ итераций получено 73 недоминируемых решений, причем эти решения достаточно точно приближают аналитический фронт Парето задачи.

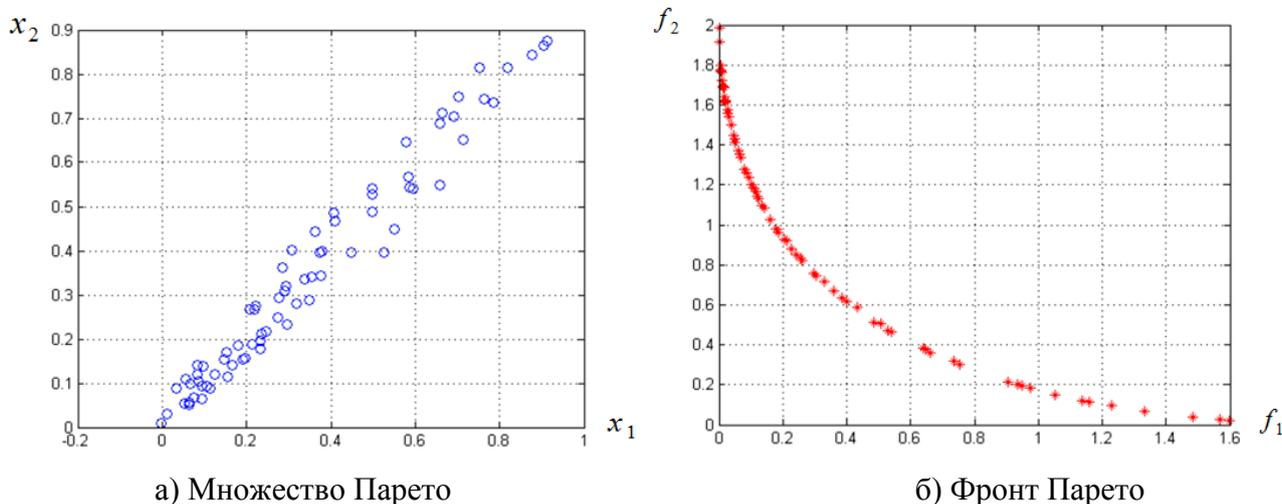


Рисунок 24. Результаты вычислительного эксперимента: задача (7), (8); модификация пчелиного алгоритма; $\hat{t} = 10$ [8]

Задача с невыпуклым фронтом Парето [21]. Рассматриваем задачу минимизации критериев

$$f_1(X) = (x_1^2 + x_2^2)^{1/8}, \quad f_2(X) = ((x_1 - 0,5)^2 + (x_2 - 0,5)^2)^{1/4}, \quad (15)$$

$$D_X = \{ X \mid x_i \in [-5; 10], \quad i = 1, 2 \}. \quad (16)$$

Решение задачи выполнено при значениях свободных параметров, равных $|S^o| = 100$, $|A^b| = 20$, $n^b = 30$, $\hat{t} = 200$. Получено 48 недоминируемых решений, которые с приемлемой точностью аппроксимируют фронт Парето (рисунок 25).

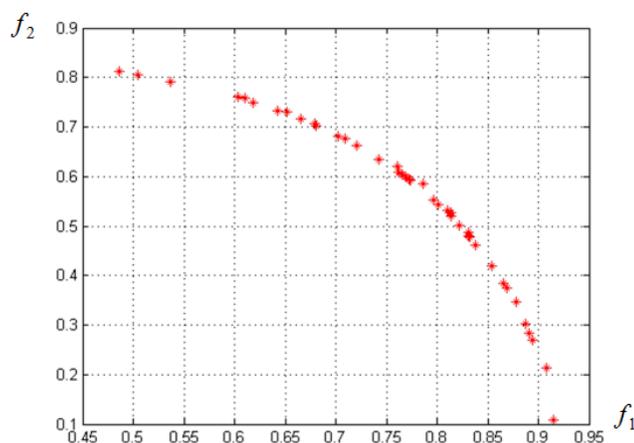


Рисунок 25. Парето-аппроксимация: задача (15), (16); модификация пчелиного алгоритма; $\hat{t} = 200$ [8]

Следует отметить, что алгоритм *MOPSO* в аналогичной ситуации смог отыскать для этой задачи 94 недоминируемых решения.

Задача с разрывным фронтом Парето [21]:

$$f_1(x) = \begin{cases} -x, & x \leq 1, \\ -2 + x, & x \in (1; 3], \\ 4 - x, & x \in (3; 4], \\ -4 + x, & x > 4, \end{cases} \quad f_2(X) = (x - 5)^2, \quad (17)$$

$$D_X = \{x \mid x \in [-100; 100]\} \quad (18)$$

Критерии $f_1(X)$, $f_2(X)$ полагаем, подлежит минимизации. Задача является одномерной двухкритериальной.

В вычислительном эксперименте использованы указанные выше значения свободных параметров алгоритма. Эксперимент позволил получить 155 недоминируемых решений (рисунок 26). Отметим, что алгоритм *MOPSO* для рассматриваемой задачи отыскал всего 96 решений, то есть пчелиный алгоритм обеспечил, по сравнению с алгоритмом *MOPSO*, значительно более высокую эффективность.

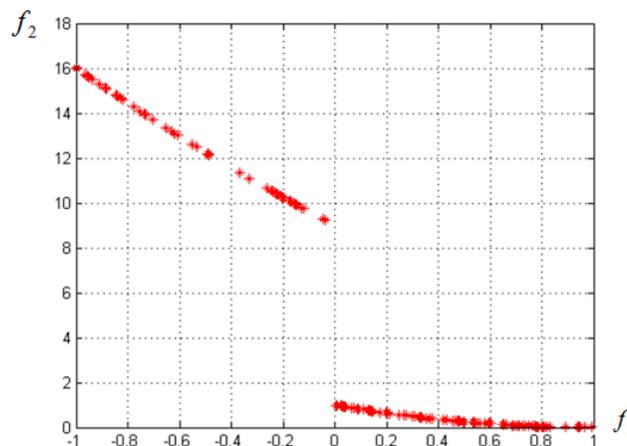


Рисунок 26. Парето-аппроксимация: задача (17), (18);
 модификация пчелиного алгоритма: $\hat{t} = 200$ [8]

Задача с множественными разрывами фронта Парето [20]:

$$f_1(X) = x_1, \quad f_2(X) = a(x_2) b(X), \quad (19)$$

$$D_X = \{X \mid x_i \in [0; 1], \quad i = 1, 2\}, \quad (20)$$

$$a(x_2) = 1 + 9 x_2,$$

$$b(X) = 1 - \left(\frac{f_1(x_1)}{a(x_2)} \right)^{1/2} - \left(\frac{f_1(x_1)}{a(x_2)} \right) \sin(10 \pi f_1(x_1)).$$

Оба критерия подлежат минимизации.

Задачу решаем и использованием значений свободных параметров алгоритма, равных $|S^o| = 200$; $|A^b| = 40$; $n^b = 60$; $\hat{t} = 100$. В вычислительном эксперименте получено 55 недоминируемых решений (рисунок 27).

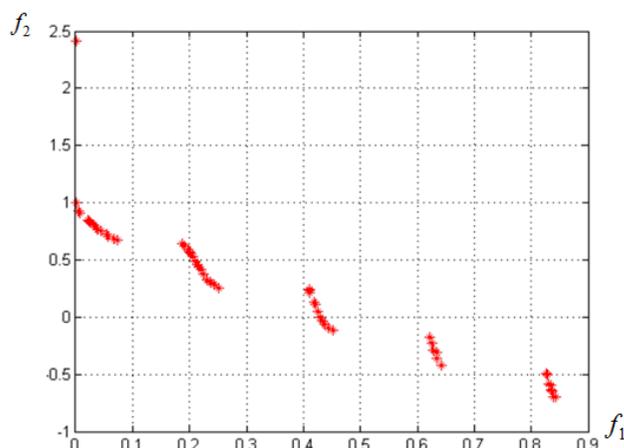


Рисунок 27. Партео-аппроксимация: задача (19), (20);
модификация пчелиного алгоритма: $\hat{t} = 100$ [8]

В тех же условиях алгоритм *MOPSO* за 100 итераций смог найти лишь 48 недоминируемых решений, то есть, для данной (сложной) задачи модифицированный пчелиный алгоритм в сравнении с алгоритмом *MOPSO*, обеспечивает значительно более высокую эффективность.

Заключение

В работе представлен обзор следующих способов оценки приспособленности агентов популяции: лексикографическая турнирная селекция; алгоритмы чередующихся критериальных функций; алгоритмы на основе ранжирования агентов; алгоритмы, не использующие ранжирование агентов.

Рассмотрены следующие популяционные алгоритмы Парето-аппроксимации, использующие эти способы оценки приспособленности: генетический алгоритм NSGA II; генетический алгоритм NPGA; когенетический эволюционный алгоритм CGEA; алгоритмы MOPSO, MS MOPSO на основе алгоритма роя частиц; алгоритм колонии муравьев CIAC; модифицированный алгоритм пчелиного роя.

Оценка эффективности указанных алгоритмов Парето-аппроксимации выполнена с использованием следующих индикаторов качества аппроксимации: разреженностью покрытия (*scarcity*); число недоминируемых решений, найденных ПП-алгоритмом за заданное число итераций.

Основная часть исследования выполнена с привлечением известных тестовых МКО-задач ZDT1 – ZDT4, ZDT6, DLTZ4, а также практически значимых многокритериальных задач оптимального управления спуском космического аппарата и оптимизации автомобильной коробки переключения передач.

Результаты исследования показывают, что для обеспечения высокого качества аппроксимации множества Парето по каждому из рассматриваемых индикаторов качества необходимо использовать специальные алгоритмические решения, например, нишевание - для обеспечения равномерности покрытия. Важно, что эти алгоритмические решения, вообще

говоря, различны для различных индикаторов качества. Потому актуальной является задача разработки алгоритмов, «заточенных» под каждый из индикаторов качества Парето-аппроксимации.

С другой стороны, наличие значительного числа индикаторов качества Парето-аппроксимации означает многокритериальность задачи оценки этого качества. Таким образом, актуальной является проблема разработки алгоритмов аппроксимации, как многокритериальных алгоритмов.

В случае высокой вычислительной сложности критериальных функций, что имеет место, например, при многокритериальной оптимизации динамических систем (пп. 3.2, 3.4, 3.6, 3.8), решение МКО-задачи может требовать высоких вычислительных затрат. Поэтому актуальным является исследование параллельных алгоритмов решения этой задачи, ориентированных на различные классы параллельных ЭВМ. Опыт авторов данной работы в этой области, предполагается, станет предметом самостоятельной публикации

ЛИТЕРАТУРА

1. Карпенко А.П., Митина Е.В., Семенихин А.С. Популяционные методы аппроксимации множества Парето в задаче многокритериальной оптимизации / Обзор // Наука и образование: электронное научно-техническое издание. – 2012. - №4 (<http://www.technomag.edu.ru/doc/363023.html>).
2. А.П. Карпенко Современные алгоритмы поисковой оптимизации. Алгоритмы вдохновленные природой / А.П. Карпенко. – Москва: Издательство МГТУ им. Н.Э. Баумана, 2014. – 446 с.
3. Kennedy J., Eberhart R. Particle swarm optimization / Proc. of the IEEE Int. Conf. On Neural Networks. -1995. - pp. 1942–1948.
4. Карпенко А.П., Селиверстов Е.Ю. Глобальная оптимизация методом роя частиц. Обзор // Информационные технологии. - 2010. - №2. - С. 25-34.
5. Dorigo M., Maniezzo M., Colorni A. Ant System: Optimization by Colony of Cooperating Agent's // IEEE Transaction Systems, Man and Cybernetics. – Part B. – 1996. – Vol. 26. – P. 29–41.
6. Карпенко А.П., Чернобривченко К.А. Эффективность оптимизации методом непрерывно взаимодействующей колонии муравьев (CIAC) // Наука и образование: электронное научно-техническое издание. – 2011. - №2 (<http://technomag.edu.ru/doc/165551.html>).
7. Karaboga D. An idea based on honey bee swarm for numerical optimization / Technical report - TR06, Erciyes University, Engineering Faculty, Computer Engineering Department. Kayseri/Türkiye. - 2005. – 10 pp.
8. Гришин А.А., Карпенко А.П. Исследование эффективности метода пчелиного роя в задаче глобальной оптимизации // Наука и образование: электронное научно-техническое издание. – 2010. - №8 (<http://technomag.edu.ru/doc/154050.html>).
9. Zitzler E., Deb K., Thiele L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results // Evolutionary Computation. – 2000. - Vol. 8(2). - pp. 173-195.

10. Белоус В.В., Грошев С.В., Карпенко А.П., Шибитов И.А. Оценка качества Парето-аппроксимации в задаче многокритериальной оптимизации. Обзор программных систем // Наука и образование: электронное научно-техническое издание. – 2014. – №4 (<http://technomag.bmstu.ru/doc/709198.html>).
11. Deb K. Multi-Objective Optimization Using Evolutionary Algorithms. - Chichester, U.K.: Wiley. – 2001. - 497 pp.
12. Goldberg D.E. Genetic Algorithms in Search, Optimization, and Machine Learning. - Addison-Wesley. - 1989. - 432 pp.
13. Laumanns M., Rudolph G., Schwefel H.P. A spatial predator-prey approach to multi-objective optimization: a preliminary study / Proceedings of the Parallel Problem Solving from Nature. – 1998. - Vol. 5. - P. 241–249.
14. Chowdhury S., Dulikravich G.S., Moral R.J. (2009) ‘Modified predator-prey algorithm for constrained and unconstrained multi-objective optimization // Int. J. Mathematical Modeling and Numerical Optimization. -2009. - Vol. 1. - Nos. 1/2. - pp. 1–38.
15. Horn J, Nafpliotis N., Goldberg D.E. A niched Pareto genetic algorithm for multiobjective optimization / Proceedings of the First IEEE Conference on Evolutionary Computation. - NJ: IEEE Press. – 1994. - pp. 82–87.
16. Deb K., Pratap A., Agarwal S., Meyarivan T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II // IEEE Transactions on Evolutionary Computation. - 2002. -№6 (2). - pp. 182–197.
17. Zitzler E., Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach / Evolutionary Computation, IEEE Transactions on. -1999. - vol. 3. - pp. 257-271.
18. Zitzler E., Laumanns M., Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm / Proc. EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. – 2001.
19. Knowles J., Corne D. The Pareto achieved evolution strategy: A new baseline algorithm for Pareto multiobjective optimization / Proceedings of the Congress on Evolutionary Computation (CEC99). – 1999. - Vol. 1. - pp. 98–105.
20. Bentley P.J., Wakefield J.P. An Analysis of Multiobjective Optimization within Genetic Algorithms / Technical Report ENGPJB96, University of Huddersfield, UK. – 1996. – 19 pp.
21. Deb K., Thiele L., Laumanns M., Zitzler E. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Evolutionary Multiobjective Optimization / Theoretical Advances and Applications. - Springer. - 2005. - P. 105-145.
22. Дивеев А.И., Северцев Н.А. Метод сетевого оператора для синтеза системы управления спуском космического аппарата при неопределенных начальных условиях // Проблемы машиностроения и надежности машин. Машиноведение. – 2009. - №3. - С. 85 – 91.
23. Лобарева И.Ф., Черный С.Г., Чирков Д.В., Скороспелов В.А., Турук П.А. Многоцелевая оптимизация формы лопасти гидротурбины // Вычислительные технологии. – 2006. - Том 11. - С. 63 – 76.

24. Карпенко А.П., Овчинников В.А., Семенихин А.С. Программная система PRADIS // FRONT для построения множества Парето в задаче многокритериальной оптимизации динамических систем с использованием параллельного генетического алгоритма // Информационные технологии. – 2009. - №8. – С. 27 – 33.
25. Карпенко А.П., Митина Е.В., Семенихин А.С. Когенетический алгоритм Парето-аппроксимации в задаче многокритериальной оптимизации // Информационные технологии. – 2013. - №1. - С. 22-32.
26. Семенкин Е.С. Эволюционные методы моделирования и оптимизации сложных систем / Е.С. Семенкин и др. // Электронные публикации НГУ (http://library.krasu.ru/ft/ft/_umkd/22/u_lectures.pdf).
27. Zitzler E., Deb K., Thiele L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results // Evolutionary Computation. – 2000. - Vol. 8(2). - P. 173-195.
28. Hu X., Eberhart R. Multiobjective optimization using dynamic neighborhood particle swarm optimization / World Congress on Computational Intelligence. Proceeding. - 2002. P. - 1677–1681.
29. Карпенко А.П., Чернобривченко К.А. Эффективность оптимизации методом непрерывно взаимодействующей колонии муравьев (CIAC) // Наука и образование: электронное научно-техническое издание. – 2011. – №2 (<http://technomag.edu.ru/doc/165551.html>).
30. Pham D.T., Ghanbarzadeh A., Кос Е., Otri S., Rahim S., Zaidi M. The Bees Algorithm - A Novel Tool for Complex Optimisation Problems / Proc 2nd Int Virtual Conf. on Intelligent Production Machines and Systems (IPROMS 2006). - Oxford: Elsevier. – 2006. - PP. 454-459.

Groshev Sergey Vladimirovich

Bauman Moscow state technical university, Russia, Moscow
E-mail: groshev_sergey@mail.ru

Karpenko Anatoly Pavlovich

Bauman Moscow state technical university, Russia, Moscow
E-mail: akarpenko@bmstu.ru

Martynyuk Vladimir Alekseevich

Bauman Moscow state technical university, Russia, Moscow
E-mail: Martrk6@gmail.com

The effectiveness of population-based Pareto-approximation algorithms. Experimental comparison

Abstract. Multicriteria optimization problems arise in various fields of science and technology, and a number of different ways to solve it is suggested. Classic algorithms for solving the problem of multi-criteria optimization (MCO-task) are based on the reduction of its tasks to the aggregate of the global one-criterion optimization. A relatively new and rapidly growing class of multi-criteria optimization algorithms, are algorithms, based on Pareto-approximation, involving preliminary construction of a finite-dimensional approximation of the set, and thus, the Pareto front and considered MCO-task. Typically Pareto-approximation algorithms are based on the evolution and, more often, genetic algorithms. The principal on the evolutionary algorithms Pareto-approximation are rules for the fitness function that ensures the movement of a population of individuals, ultimately, in the direction of the Pareto set. This work is a review, and includes the results of experimental studies on the effectiveness of a number of population-based Pareto-approximation algorithms, obtained with the participation of the authors in the period from about 2010 to the present time.

The work aims to show in a number of known test and practically important MCO-task comparative efficiency of Pareto-approximation using a genetic algorithm, swarm algorithms particles, ant colonies and colonies of bees in combination with a variety of methods of formation of the fitness function. . To evaluate the effectiveness of these Pareto-approximation algorithms we are using the indicator, called sparse coverage (scarcity), which has the meaning of the minimum distance between the solutions belonging to the Pareto-approximation. Besides used as an indicator of the number of non-dominated solutions found Pareto approximation algorithm for a predetermined number of iterations.

The research results show that to ensure the quality of approximation of Pareto set for each of the considered indicators of quality specific algorithmic solutions must be used, for example, nicheness to ensure uniform coverage. It is important that these decisions are, generally speaking, different for different quality indicators. Because the actual problem is the development of algorithms, "sharpened" under each of the Pareto approximation of quality indicators. On the other hand, the presence of a significant number of Pareto-approximation of quality indicators is multicriteriality task of evaluating this quality. Thus, the actual problem is the development of a Pareto-approximation algorithms as a multicriteria algorithms.

Keywords: multicriteria optimization; Pareto set; Pareto approximation; population-based algorithms; genetic algorithms; Pareto Front; quality of solution

REFERENCES

1. Karpenko A.P., Mitina E.V., Semenikhin A.S. Populyatsionnye metody approksimatsii mnozhestva Pareto v zadache mnogokriterial'noy optimizatsii / *Obzor // Nauka i obrazovanie: elektronnoe nauchno-tehnicheskoe izdanie.* – 2012. №4 (<http://www.technomag.edu.ru/doc/363023.html>).
2. A.P. Karpenko *Sovremennye algoritmy poiskovoy optimizatsii. Algoritmy vdokhnovlennye prirodoy / A.P. Karpenko.* – Moskva: Izdatel'stvo MGTU im. N.E. Baumana, 2014. – 446 s.
3. Kennedy J., Eberhart R. Particle swarm optimization / *Proc. of the IEEE Int. Conf. On Neural Networks.* 1995. pp. 1942–1948.
4. Karpenko A.P., Seliverstov E.Yu. Global'naya optimizatsiya metodom roya chastits. *Obzor // Informatsionnye tekhnologii.* 2010. №2. S. 25-34.
5. Dorigo M., Maniezzo M., Coloni A. Ant System: Optimization by Colony of Cooperating Agent's // *IEEE Transaction Systems, Man and Cybernetics.* – Part B. – 1996. – Vol. 26. – P. 29–41.
6. Karpenko A.P., Chernobrivchenko K.A. Effektivnost' optimizatsii metodom nepreryvno vzaimodeystviyushchey kolonii murav'ev (CIAC) // *Nauka i obrazovanie: elektronnoe nauchno-tehnicheskoe izdanie.* – 2011. №2 (<http://technomag.edu.ru/doc/165551.html>).
7. Karaboga D. An idea based on honey bee swarm for numerical optimization / *Technical report - TR06, Erciyes University, Engineering Faculty, Computer Engineering Department. Kayseri/Türkiye.* 2005. – 10 pp.
8. Grishin A.A., Karpenko A.P. Issledovanie effektivnosti metoda pchel'nogo roya v zadache global'noy optimizatsii // *Nauka i obrazovanie: elektronnoe nauchno-tehnicheskoe izdanie.* – 2010. №8 (<http://technomag.edu.ru/doc/154050.html>).
9. Zitzler E., Deb K., Thiele L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results // *Evolutionary Computation.* – 2000. Vol. 8(2). pp. 173-195.
10. Belous V.V., Groshev S.V., Karpenko A.P., Shibitov I.A. Otsenka kachestva Pareto-approksimatsii v zadache mnogokriterial'noy optimizatsii. *Obzor programmnykh sistem // Nauka i obrazovanie: elektronnoe nauchno- tekhnicheskoe izdanie.* – 2014. – №4 (<http://technomag.bmstu.ru/doc/709198.html>).
11. Deb K. *Multi-Objective Optimization Using Evolutionary Algorithms.* Chichester, U.K.: Wiley. – 2001. 497 pp.
12. Goldberg D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley. 1989. 432 pp.
13. Laumanns M., Rudolph G., Schwefel H.P. A spatial predator-prey approach to multi-objective optimization: a preliminary study / *Proceedings of the Parallel Problem Solving from Nature.* – 1998. Vol. 5. P. 241–249.
14. Chowdhury S., Dulikravich G.S., Moral R.J. (2009) 'Modified predator-prey algorithm for constrained and unconstrained multi-objective optimization // *Int. J. Mathematical Modeling and Numerical Optimization.* 2009. Vol. 1. Nos. 1/2. pp. 1–38.
15. Horn J, Nafpliotis N., Goldberg D.E. A niched Pareto genetic algorithm for multiobjective optimization / *Proceedings of the First IEEE Conference on Evolutionary Computation.* NJ: IEEE Press. – 1994. pp. 82–87.
16. Deb K., Pratap A., Agarwal S., Meyarivan T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II // *IEEE Transactions on Evolutionary Computation.* 2002. №6 (2). pp. 182–197.

17. Zitzler E., Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach / *Evolutionary Computation*, IEEE Transactions on. 1999. vol. 3. pp. 257-271.
18. Zitzler E., Laumanns M., Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm / *Proc. EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems.* – 2001.
19. Knowles J., Corne D. The Pareto achieved evolution strategy: A new baseline algorithm for Pareto multiobjective optimization / *Proceedings of the Congress on Evolutionary Computation (CEC99).* – 1999. Vol. 1. pp. 98–105.
20. Bentley P.J., Wakefield J.P. An Analysis of Multiobjective Optimization within Genetic Algorithms / *Technical Report ENGPJB96*, University of Huddersfield, UK. – 1996. – 19 pp.
21. Deb K., Thiele L., Laumanns M., Zitzler E. Scalable Test Problems for Evolutionary Multi-Objective Optimization. *Evolutionary Multiobjective Optimization / Theoretical Advances and Applications.* Springer. 2005. P. 105-145.
22. Diveev A.I., Severtsev N.A. Metod setevogo operatora dlya sinteza sisetmy upravleniya spuskom kosmicheskogo apparata pri neopredelennykh nachal'nykh usloviyakh // *Problemy mashinostroeniya i nadezhnosti mashin.* Mashinovedenie. – 2009. №3. S. 85 – 91.
23. Lobareva I.F., Chernyy S.G., Chirkov D.V., Skorospelov V.A., Turuk P.A. Mnogotselevaya optimizatsiya formy lopasti gidroturbiny // *Vychislitel'nye tekhnologii.* – 2006. Tom 11. S. 63 – 76.
24. Karpenko A.P., Ovchinnikov V.A., Semenikhin A.S. Programmaya sistema PRADIS // *FRONT dlya postroeniya mnozhestva Pareto v zadache mnogokriterial'noy optimizatsii dinamicheskikh sistem s ispol'zovaniem parallel'nogo geneticheskogo algoritma* // *Informatsionnye tekhnologii.* – 2009. №8. – S. 27 – 33.
25. Karpenko A.P., Mitina E.V., Semenikhin A.S. Kogeneticheskiy algoritm Pareto-approximatsii v zadache mnogokriterial'noy optimizatsii // *Informatsionnye tekhnologii.* – 2013. №1. S. 22-32.
26. Semenkin E.S. Evolyutsionnye metody modelirovaniya i optimizatsii slozhnykh sistem / E.S. Semenkin i dr. // *Elektronnye publikatsii NGU* (http://library.krasu.ru/ft/ft/_umkd/22/u_lectures.pdf).
27. Zitzler E., Deb K., Thiele L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results // *Evolutionary Computation.* – 2000. Vol. 8(2). P. 173-195.
28. Hu X., Eberhart R. Multiobjective optimization using dynamic neighborhood particle swarm optimization / *World Congress on Computational Intelligence. Proceeding.* 2002. P. 1677–1681.
29. Karpenko A.P., Chernobrivchenko K.A. Effektivnost' optimizatsii metodom nepreryvno vzaimodeystvuyushchey kolonii murav'ev (CIAC) // *Nauka i obrazovanie: elektronnoe nauchno-tekhnicheskoe izdanie.* – 2011. – №2 (<http://technomag.edu.ru/doc/165551.html>).
30. Pham D.T., Ghanbarzadeh A., Koc E., Otri S., Rahim S., Zaidi M. The Bees Algorithm - A Novel Tool for Complex Optimisation Problems / *Proc 2nd Int Virtual Conf. on Intelligent Production Machines and Systems (IPROMS 2006).* Oxford: Elsevier. – 2006. PP. 454-459.