

Распределенная программная система для построения множества Парето в задаче многокритериальной оптимизации динамических систем с использованием параллельного генетического алгоритма

А.П.Карпенко, В.А.Овчинников, А.С.Семенихин

Рассматривается распределенная программная система PRADIS//FRONT для приближенного построения множества Парето в задаче многокритериальной оптимизации динамических систем с использованием параллельных генетических алгоритмов. В качестве среды моделирования динамических систем используется программный комплекс PRADIS.

1. Введение

Проектирование сложной технической системы обычно включает в себя этап параметрической оптимизации [1]. В настоящее время, чаще всего используется многокритериальная оптимизация (МКО), при которой качество функционирования объекта проектирования определяется некоторым набором критериев оптимальности.

Для пользователя наиболее удобны методы, основанные на непосредственном использовании множества Парето. В этом случае лицо, принимающее решение (ЛПР), выбирает компромиссное решение на фронте Парето неформальными методами, исходя только из своих предпочтений. Основным недостатком методов этого класса, сдерживающим их широкое применение, является высокая вычислительная сложность.

Известно значительное количество методов и алгоритмов приближенного построения множества Парето [3, 4]. Относительно новым и высокоэффективным классом таких методов являются методы на основе генетических алгоритмов (ГА). ГА обладают рядом характеристик, делающих их более предпочтительными, чем классические методы оптимизации [5]. В программной системе PRADIS//FRONT реализован параллельный генетический алгоритм (ПГА) GPGA (Global Parallel Genetic Algorithm) типа master-slave [6], построенный на основе модифицированного последовательного метода NPGA (Niche Pareto Genetic Algorithm).

Построение множества Парето в задаче МКО динамических систем требует многократного моделирования исследуемой динамической системы при различных значениях варьируемых параметров. Моделирование динамических систем является самостоятельной проблемой, для решения которой разработано значительное количество программных комплексов [7-10]. В программной системе PRADIS//FRONT моделирование выполняется средствами программного комплекса PRADIS, который функционирует под управлением операционной системы Windows NT и предназначен для анализа динамических процессов в объектах, описываемых системами обыкновенных дифференциальных уравнений (СОДУ) [10].

Многократное моделирование сложной динамической системы требует больших вычислительных ресурсов, поэтому в приемлемое для исследователя время может быть выполнено только на параллельных вычислительных машинах. Система PRADIS//FRONT ориентирована на использование в качестве таких машин распределенных ЭВМ, например, вычислительных кластеров [11]. Существует большое количество программных средств, обеспечивающих эффективное использования ресурсов распределенной ЭВМ. Программная система PRADIS//FRONT построена на основе коммуникационной библиотеки MPI (реализация – MPICH) [11].

Работоспособность системы проверена на стандартных тестовых задачах многокритериальной оптимизации [12]. Практическое использование системы рассмотрено на примере приближенного построения множества Парето для задачи МКО автомобильной коробки передач.

2. Постановка задачи

Совокупность частных критериев оптимальности $\Phi(X) = (\phi_1(X), \phi_2(X), \dots, \phi_m(X))$ называется векторным критерием оптимальности. Ставится задача максимизации каждого из указанных критериев в одной и той же области допустимых значений $D_X \in P \cap D$, где $P = \{X \mid x_i^- \leq x_i \leq x_i^+, i \in [1, n]\}$ - «технологический» параллелепипед, $D = \{X \mid g_1(X) \geq 0, g_2(X) \geq 0, \dots\}$. Здесь $X \in R^n$, $g_1(X), g_2(X), \dots$ - ограничивающие функции. Задача МКО записывается в виде

$$\max_{X \in D_X} \Phi(X) = \Phi(X^*) \quad (1)$$

Векторный критерий оптимальности $\Phi(X)$ выполняет отображение множества D_X в некоторое множество D_Φ пространства критериев, которое называется множеством достижимости. Введем на множестве D_Φ отношение предпочтения. Будем говорить, что вектор $\Phi^1 \in D_\Phi$ предпочтительнее вектора $\Phi^2 \in D_\Phi$ или вектор Φ^1 доминирует вектор Φ^2 , и писать $\Phi^1 \succ \Phi^2$, если среди равенств и неравенств $\phi_k(X^1) \geq \phi_k(X^2), k \in [1 : m]$ имеется хотя бы одно строгое неравенство. Выделим из множества D_Φ подмножество точек D_Φ^* (фронт Парето), для которых нет более предпочтительных точек. Множество $D_X^* \in D_X$, соответствующее множеству D_Φ^* , называется множеством Парето. Таким образом, если $X \in D_X^*$, то $\Phi(X) \in D_\Phi^*$.

Ставится задача приближенного построения множества Парето в задаче МКО (1) на вычислительных системах класса MIMD (Multiple Instruction Multiple Data).

3. Последовательные методы приближенного построения множества Парето на основе генетических алгоритмов

В настоящее время в вычислительной практике наиболее часто используются четыре метода приближенного построения множества Парето на основе ГА [5]:

- метод VEGA (Vector Evaluated Genetic Algorithm);
- метод FFGA (Fonseca and Fleming's Multiobjective Genetic Algorithm);
- метод NPGA (Niche Pareto Genetic Algorithm);
- метод SPEA (Strength Pareto Evolutionary Algorithm).

В методе VEGA селекция производится по переключающимся частным критериям оптимальности. Для каждого из m частных критериев создается подпопуляция, содержащая N_p / m индивидов, где N_p - размер всей популяции. В j -ю подпопуляцию индивиды отбираются с помощью пропорциональной селекции по критерию $\phi_j(X)$. Далее подпопуляции смешиваются для получения популяции размера N_p , а затем по общей схеме осуществляются скрещивание и мутация.

Метод FFGA использует процедуру ранжирования индивидов, основанную на Парето-доминировании. При этом ранг каждого из индивидов определяется количеством доминирующих его других индивидов данной популяции (чем ниже ранг, тем индивид ближе к множеству Парето). Пригодность индивида вычисляется на основе величины, обратной его рангу. Для отбора в следующее поколение используется процедура турнирной селекции.

В методе NPGA, в отличие от методов VEGA, FFGA, существует механизм поддержания разнообразия популяции. Метод основан на формировании популяционных ниш.

Метод SPEA является самым сложным из числа рассматриваемых и, так же, как метод FFGA, использует селекцию, основанную на Парето-доминировании. Для предотвращения преждевременной сходимости, метод использует популяционные ниши. Очень важным свойством SPEA является возможность априорного задания количества итоговых точек в искомой аппроксимации множества Парето.

4. Параллельные методы приближенного построения множества Парето на основе генетических алгоритмов

Существует большое количество ПГА, ориентированных на разные классы параллельных вычислительных систем и на различные критерии качества алгоритмов. Один из известных вариантов классификации ПГА (PGA) приведен на Рис. 1 [6].

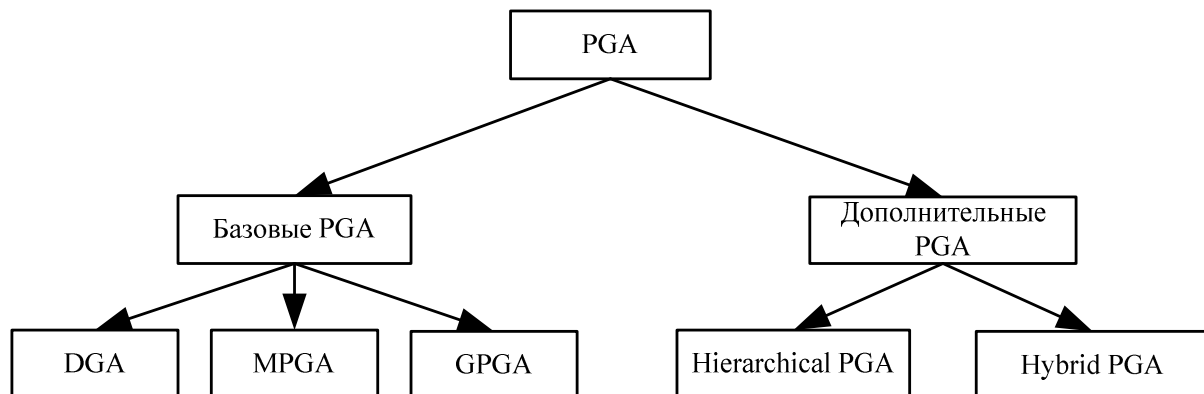


Рис. 1. Классификация ПГА

GPGA (Global Parallel Genetic Algorithm) представляет собой однопопуляционный алгоритм. Host-процессор содержит всё поколение в своей памяти и выполняет над ним операции селекции, кроссовера и мутации. Функции пригодности индивидов вычисляются на slave-процессорах. При высокой вычислительной сложности функций пригодности основной проблемой при реализации данного алгоритма является проблема балансировки загрузки slave-процессоров. Данный алгоритм ориентирован на MIMD-вычислительные системы [11].

DGA (Distributed Genetic Algorithm) – это многопопуляционный алгоритм, также ориентированный на MIMD-вычислительные системы. Каждый процессор использует свой собственный генетический алгоритм на выделенной ему части всей популяции (подпопуляции). Различают два класса распределенных генетических алгоритмов – с наличием механизма обмена индивидами между подпопуляциями (миграции) и без. Процессор начинает вычисления с разными параметрами генетического алгоритма и с разными наборами генетических операторов. Как следствие, в процессе вычислений подпопуляции будут развиваться в различных направлениях, что гарантирует большее генетическое разнообразие. Отметим, что миграционные процессы между подпопуляциями требуют очень аккуратной настройки: при большом количестве индивидов, подлежащих обмену, и высокой частоте обменов генетическое разнообразие подпопуляций нивелируется; наоборот, при малом объеме обменов и низкой их частоте может иметь место преждевременная сходимость внутри подпопуляций.

MPGA (Massively Parallel Genetic Algorithm) или Cellular Algorithm (клеточный алгоритм) ориентирован на вычислительные машины класса SIMD (Single Instruction Multiple Data) [11]. В этом случае каждым процессорным элементом в один момент времени обрабатывается один индивид. Индивиды выбирают пару и рекомбинируют со своими непосредственными соседями.

Hierarchical PGA представляют собой иерархические алгоритмы, в которых на разных уровнях иерархии реализованы разные ПГА (рассмотренные выше и другие). Обычно используются 2-х уровневые иерархии: DGA-модель - на верхнем уровне; GPGA- или MPGA-модель – на нижнем уровне.

Hybrid PGA – класс алгоритмов, использующих сочетание параллельных генетических алгоритмов и классических оптимизационных методов.

5. Используемые методы и алгоритмы приближенного построения множества Парето

В качестве базового метода используется последовательный метод NPGA. Метод NPGA более прост в реализации, чем метод SPEA, и, в то же время, обеспечивает достаточно высокую

эффективность [5]. Использована модификация метода, предложенная в работе [12] и заключающаяся в следующем:

- использование аутбридинга (дальнеродственного скрещивания) вместо турнирной селекции [13];
- использование алгоритма ранжирования индивидов из метода FFGA;
- использование операции клонирования.

Пусть P_t - текущая популяция, P' - промежуточная популяция, $P_{dom} \subseteq P_t$ - сравнительное множество, σ_{share} - заданный радиус популяционной ниши, N_{dom} - заданное количество индивидов в сравнительном множестве (давление доминирования), $d(I_1, I_2)$ - расстояние между индивидами I_1, I_2 популяции P_t в некоторой метрике.

Схему вычисления пригодности индивида и селекции в методе NPGA можно представить в следующем виде [5].

Шаг 1. Положим $s = 1$, $P' = \emptyset$, $P_{dom} = \emptyset$.

Шаг 2. Из t_{dom} случайно выбранных индивидов популяции P_t формируем сравнительное множество P_{dom} .

Шаг 3. Из оставшейся части популяции P_t случайно выбираем два индивида I_1, I_2 .

Шаг 4. Если $X_1 = M(I_1)$ недоминируем ни одним из векторов $X_k = M(I_k)$, $I_k \in P_{dom}$, а $X_2 = M(I_2)$ доминируется хотя бы одним вектором сравнительного множества P_{dom} , то победителем турнира считаем индивида I_1 : $P' = P' + I_1$.

Шаг 5. Если $X_2 = M(I_2)$ недоминируем ни одним из векторов $X_k = M(I_k)$, $I_k \in P_{dom}$, а $X_1 = M(I_1)$ доминируется хотя бы одним вектором сравнительного множества P_{dom} , то победителем турнира считаем индивида I_2 : $P' = P' + I_2$.

Шаг 6. Если победитель турнира не определен, то выполняем следующие действия.

а) Вычисляем количество индивидов в промежуточной популяции P' , которые находятся от индивида I_1 на расстоянии, не превышающем радиус ниши σ_{share} :

$$n(I_1) = \{I_k \mid I_k \in P' \wedge d(I_1, I_k) < \sigma_{share}\}. \quad (2)$$

б) Аналогично вычисляем количество индивидов $n(I_2)$ в промежуточной популяции P' , которые находятся от индивида I_2 на расстоянии, не превышающем радиус ниши σ_{share} .

в) Если $n(I_1) < n(I_2)$, то полагаем $P' = P' + I_1$. Иначе - $P' = P' + I_2$.

Шаг 7. Полагаем $s = s + 1$. Если $s \leq N_p$, то переходим к шагу 3, иначе – заканчиваем вычисления.

Далее по общей схеме ГА выполняется скрещивание индивидов промежуточной популяции P' и формируется популяция P'' . Затем реализуется мутация индивидов популяции P'' , на основе которой получается популяция $P''' = P_{t+1}$.

Вместо турнирной селекции мы используем ранжирование индивидов на основе Парето-доминирования [12]. Ранг 1 присваивается индивиду, для которого не существует другого индивида, лучшего по всем частным критериям оптимальности в текущей популяции. Ранг остальных определяется по формуле

$$rank(I_k) = 1 + a_k,$$

где a_k - количество представителей текущей популяции, лучших по всем частным критериям оптимальности.

Используемые механизмы формирования индивидов обеспечивают выполнение условия $X = M(I) \in \Pi$, но не обеспечивают, в общем случае, выполнение условия

$$X = M(I) \in D. \quad (3)$$

Ранг индивидов, которые нарушают ограничение (3) назначается в зависимости от того, в какой мере эти ограничения нарушены. Ранг индивида, для которого ограничение (3) нарушено больше ранга любого индивида, для которого оно выполняется.

Функция пригодности строится на основе выражения

$$\varphi(I) = 1 + \sum_{k=1}^{\text{rank}(I)-1} \eta(k), \quad (4)$$

где $\eta(k)$ - число индивидов ранга k . Функция пригодности имеет вид:

$$f(I_1) = \mu(I_1)\varphi(I_1), \quad (5)$$

где $\mu(I_1)$ - нишевое число индивида I_1 , вычисляемое по формуле

$$\mu(I_1) = \sum_{I_k \in P_i} Sh(d(I_1, I_k)). \quad (6)$$

Здесь $Sh(d)$ - функция разделения:

$$Sh(d) = \begin{cases} 1 - d / \sigma_{share}, & d < \sigma_{share}, \\ 0, & d \geq \sigma_{share}. \end{cases} \quad (7)$$

В системе PRADIS//FRONT индивиды для скрещивания (индивиды-родители) отбираются по следующему правилу: первый выбирается случайно, вне зависимости от значения его функции пригодности; второй выбирается из той же популяции на некоторой дистанции от первого (исходя из принципов поддержания разнообразия популяции). В случае если нельзя найти второго индивида на указанном расстоянии, то выбирается случайный индивид.

Далее происходит скрещивание. Существует множество разновидностей оператора скрещивания: одноточечный, двухточечный, равномерный и пр. [13]. В системе PRADIS//FRONT скрещивание выполняется по следующей схеме: выбирается ведущий родитель; к значениям его генов прибавляется разность между соответствующими генами родителей, умноженная на коэффициент скрещивания (CrossoverRate); потомок сохраняется во множестве P'' .

Известно множество операторов мутации, отличающихся количеством индивидов, подверженных мутации, алгоритмами мутации, количеством генов, участвующих в мутации, и пр. [13]. Отличительная особенность оператора мутации в системе PRADIS//FRONT – мутации подвергается все произведенное потомство.

Также, нами была сделана попытка использовать классическую схему оператора мутации – инвертирование заданного количества битов в хромосоме. Тестирование показало – такая схема дает существенно худшие результаты, чем рассмотренная выше.

Распараллеливание вычислений в работе реализовано по схеме алгоритма GPGA. Основания выбора следующие:

1) Ориентация работы на распределенные вычислительные системы. В результате из рассмотрения исключаются алгоритмы, ориентированные на SIMD-системы, а также алгоритмы, требующие большого количества коммуникаций между процессорами.

2) Высокая вычислительная сложность функции пригодности индивидов, обусловленная тем, что каждое вычисление такой функции требует интегрирования соответствующей СОДУ и вычисления значений всех частных критериев оптимальности.

3) Простота реализации модели параллельных вычислений master-slave.

Для балансировки загрузки процессоров используется равномерная статическая балансировка [14], при которой каждый slave-процессор обрабатывает одинаковое количество индивидов, назначаемых ему host-процессором.

6. Особенности реализации программной системы PRADIS//FRONT

Программная система PRADIS//FRONT состоит из серверной и клиентской частей. Серверная часть системы реализует следующие основные функции:

- инициализация приложения;
- поддержка файлов, с помощью которых осуществляется обмен данными с клиентской частью;

- поддержка очереди заданий для клиентов;
- синхронизация работы;
- выдача клиентам заданий для расчета;
- прием от клиентов результатов расчета;
- реализация генетического алгоритма;
- обработка результатов.

Клиентская часть системы реализует следующие функции:

- поддержка работы программного комплекса PRADIS;
- запуск PRADIS для расчета;
- получение от PRADIS результатов расчета;
- прием от серверной части заданий на расчет;
- передача серверной части результатов расчета.

В системе PRADIS//FRONT определено два типа сообщений. Сообщения, посылаемые сервером и принимаемые клиентами, имеют формат

<Номер задачи для расчета>,

а сообщения, посылаемые клиентами и принимаемые сервером, в виде

<Номер рассчитанной задачи> + <ID процессора> + <Результаты расчета>.

Перед началом обмена сообщениями для серверной и клиентских частей происходит барьерная синхронизация.

Серверная часть системы. Сервер назначает задания клиентам по одному. Поэтому при сбое в работе одного из клиентов пропадает лишь то задание, которое данный клиент обрабатывал в момент сбоя.

Формирование заданий в серверной части системы выполняется с помощью функции, которая создает необходимое количество заданий, сохраняя каждое из них в отдельном файле.

После формирования файлов с заданиями, сервер начинает выполнять только диспетчерские функции: раздает номера заданий клиентам, получает от них результаты расчетов и сохраняет их в промежуточном буфере.

После завершения обработки клиентами всех заданий и приема сервером всех результатов сервер реализует одну итерацию генетического алгоритма.

Клиентская часть системы. Основными функциями клиентской части системы являются обмен данными с серверной частью и взаимодействие с программным комплексом PRADIS. Задания для расчета, записанные на одном из внутренних языков решателя, готовит серверная часть системы, а клиентской части сообщается только номер задания.

Программный комплекс PRADIS дает возможность использовать задания, написанные на языках PSL, PPL (Python), а также задания в формате схем предпроцессора Qucs. Для каждого из способов требуется запускать свою часть решателя. В настоящей версии программной системы PRADIS//FRONT реализован запуск заданий только на языке PSL.

7. Тестирование алгоритма приближенного построения множества Парето

Работоспособность рассматриваемого алгоритма проверялась на тестовых задачах ZDT3, ZDT6, DTLZ4 из стандартного набора тестов для непрерывных многокритериальных задач [12]. При решении всех тестовых задач использовались следующие значения параметров алгоритма:

- параметр разделения $\sigma_{share}=0.01$;
- доля отбираемых для скрещивания индивидов $T_r=0.3$;
- параметр рекомбинации $CrossoverRate=0.7$;
- мутационный параметр $MutationRate=1.0$;
- число индивидов $N_p = 1000$;
- максимальное число поколений $N_{gen} = 1000$.

Рис. 2 иллюстрирует результаты тестирования. На рисунке представлены точные фронты Парето (пунктир для 2а и 2б, сплошная линия для 2в) и их аппроксимации, полученные с помощью программной системы PRADIS//FRONT (линии и точки серого цвета). Количество поколений, при которых получены приведенные на рисунках результаты, изменяется от 300 до 1000.

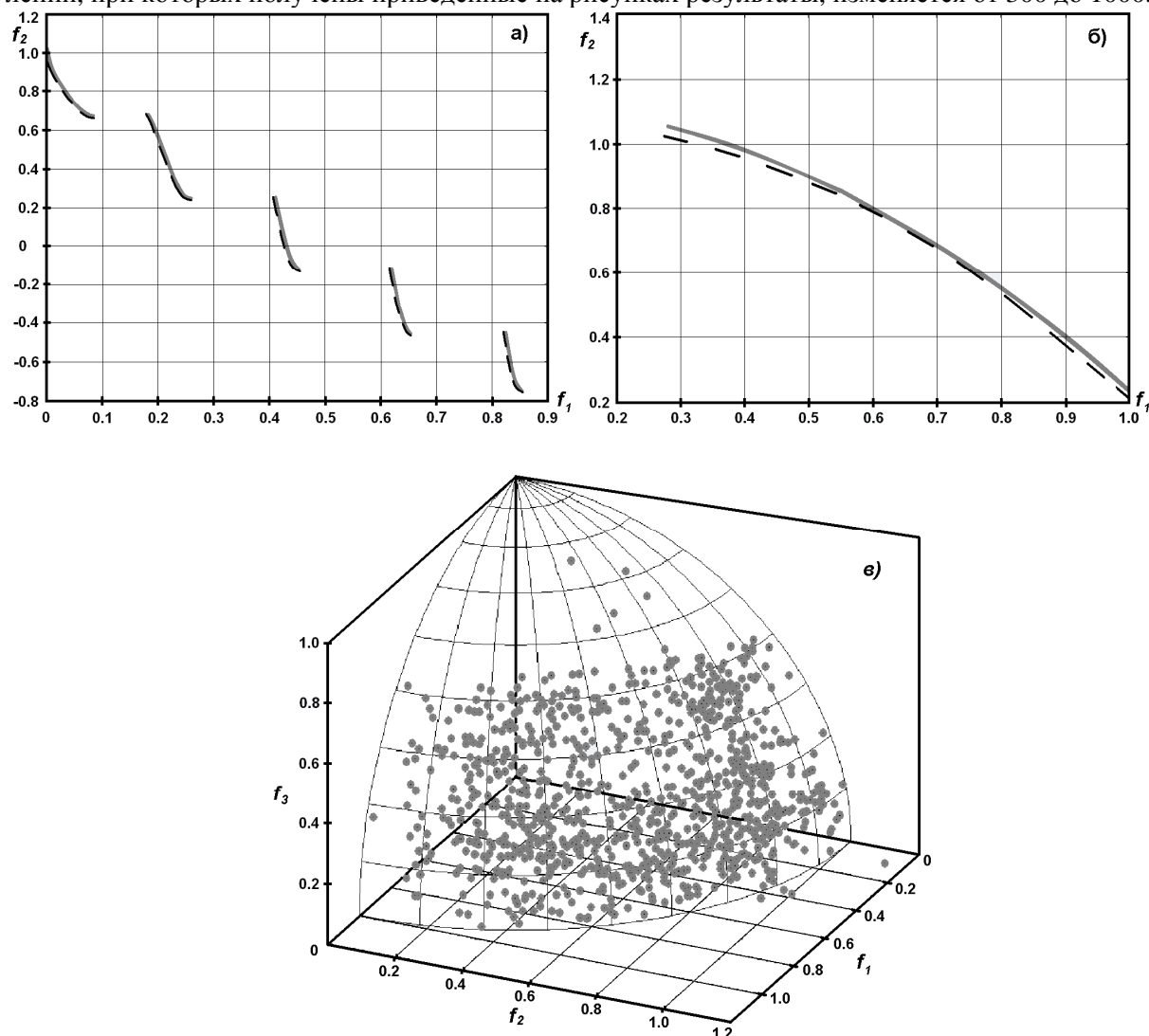


Рис. 2. Результаты тестирования для задач ZDT3 (а), ZDT6 (б), DTLZ4 (в)

Результаты тестирования показывают, что используемый в системе PRADIS//FRONT алгоритм позволяет получить достаточно точную аппроксимацию множества Парето для тестовых задач ZDT3, ZDT6 и близкую для задачи DTLZ4.

8. Приближенное построение множества Парето для задачи многокритериальной оптимизации автомобильной коробки передач

Программная система PRADIS//FRONT была использована для построения фронта Парето в задаче МКО автомобильной коробки передач. Рассматривается участок разгона заднеприводного автомобиля массой 1500 кг с пятиступенчатой коробкой передач при заданной характеристике двигателя. Разгон осуществляется из состояния покоя в течение 40 секунд. В начальный момент времени, полагается, включена первая передача. При движении автомобиля учитывается сопротивление воздуха, трение качения, инерционные и упругие характеристики трансмиссии. Общая структура модели автомобиля представлена на Рис. 3.



Рис. 3. Структурная схема модели

Варьируемыми параметрами модели являются следующие 10 параметров ($n = 10$):

- N_1, N_2, N_3, N_4, N_5 - передаточные отношения для всех 5 передач;
- T_1 – время сброса сцепления;
- T_2, T_3, T_4, T_5 – времена включения 2-й, 3-й и т.д. передач.

Параллелепипед D допустимых значений вектора варьируемых параметров определяется неравенствами

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

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

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

где R^1 - пространство вещественных чисел. Множество допустимых значений D формируется неравенствами $T_2 < T_3 < T_4 < T_5$. Выходные переменные модели:

- 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 вале двигателя;
- V_Engine – частота вращения вала двигателя;
- Out_Gear – частота вращения выходного вала коробки передач;
- Efficiency – КПД трансмиссии.

На основе указанных выходных параметров модели сформировано 4 следующих критерия оптимальности:

- максимальный момент в трансмиссии – MaxTorque (полежит минимизации);
- максимальная скорость автомобиля - VelocityMax (полежит максимизации);
- время разгона до 100 км/ч - Time100 (подлежит минимизации);
- эластичность автомобиля (время разгона с 60 до 100 км/ч) - Time60_Time100 (подлежит минимизации).

С помощью программной системы PRADIS//FRONT получены аппроксимации множества Парето для 3 следующих задач многокритериальной оптимизации:

- 1) двухкритериальная задача ($\phi_1 = \text{MaxTorque}$, $\phi_2 = \text{VelocityMax}$);
- 2) трехкритериальная задача ($\phi_1 = \text{MaxTorque}$, $\phi_2 = \text{VelocityMax}$, $\phi_3 = \text{Time100}$);
- 3) четырехкритериальная задача ($\phi_1 = \text{MaxTorque}$, $\phi_2 = \text{VelocityMax}$, $\phi_3 = \text{Time100}$, $\phi_4 = \text{Time60_Time100}$).

Результаты расчетов для двухкритериальной и трехкритериальной задач приведены на рис. 4а, 4б, соответственно. Часть результатов расчета для 4-х критерийной задачи приведена в таблице 1. Отметим, что критерий ϕ_2 в задачах 2-х и 4-х критерийной оптимизации имеет

отрицательные значения потому, что предварительно был сделан переход от задачи максимизации к задаче минимизации: $\phi_2 = -\text{VelocityMax}$

Таблица 1

$\phi_1 [H.M]$	$\phi_2 [KM \setminus Ч]$	$\phi_3 [c]$	$\phi_4 [c]$
11756.109375	-166.209230	7.596370	3.637967
16719.480469	-170.377962	7.957443	3.661078
13848.692383	-171.506975	7.986857	3.878675
18383.888672	-167.778259	7.095534	3.644868
12064.768555	-170.253357	8.701928	3.679001
15854.546875	-171.405472	7.921792	3.730051
16389.140625	-165.536545	7.094394	3.436554
12209.157227	-170.881882	8.666712	3.780350

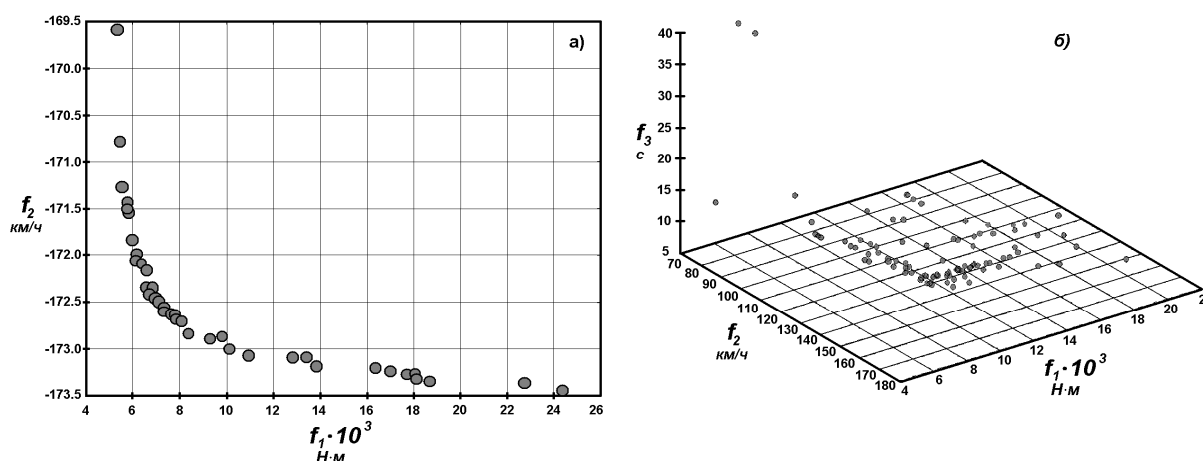


Рис. 4. Аппроксимация фронта Парето для 2-х (а) и 3-х (б) критериальной задач

9. Заключение

Работа впервые позволила использовать программный комплекс PRADIS в параллельном режиме. Результаты работы показывают, что программная система PRADIS//FRONT является удобным и надежным средством решения задач многокритериальной оптимизации сложных динамических систем на основе построения соответствующих множеств Парето.

В рамках дальнейшего развития работы будет поддерживаться запуск заданий на распределенной вычислительной системе не только на языке PSL, но также на языке PPL и в формате схем предпроцессора PRADIS – Qucs. Планируется реализация многопоточной серверной части приложения для осуществления обмена сообщениями с разными клиентами одновременно. Также планируется использовать другие методы балансировки загрузки узлов распределенной вычислительной системы [15].

Литература

1. Норенков И.П. Основы автоматизированного проектирования: Учеб. для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2006. – 448 с.
2. Lotov A.V., Bushenkov V.A., and Kamenev G.K. (2004) Interactive Decision Maps. Approximation and Visualization of Pareto Frontier. Kluwer Academic Publishers.
3. Соболев И.М., Статников Р.Б. Выбор оптимальных параметров в задачах со многими критериями.- М.: Дрофа, 2006. -175 с.
4. Подиновский В.В., Ногин В.Д. Парето-оптимальные решения многокритериальных задач. –М.: ФИЗМАТЛИТ, 2007. -256 с.
5. Гуменникова А.П. Адаптивные поисковые алгоритмы для решения сложных задач многокритериальной оптимизации: Дис. ... канд. техн. наук. - Красноярск, 2006. – 129 с.

6. Cantú-Paz, E., A Survey of Parallel Genetic Algorithms, *Calculateurs Paralleles*, Vol. 10, No. 2. Paris: Hermes, 1998., available via ftp from: <ftp://ftpillgal.ge.uiuc.edu/pub/papers/Publications/cantupaz/survey.ps.Z>.
7. MSC.ADAMS - виртуальное моделирование машин и механизмов. - <http://www.mssoftware.ru/>
8. Трудоношин В.А., Трудоношин И.В. Моделирование электромеханических систем с помощью программно-методического комплекса «ПА9»// Информационные технологии, 2006, № 4, с.
9. Погорелов Д.Ю. Компьютерное моделирование динамики технических систем с использованием программного комплекса "Универсальный механизм".- <http://www.umlabor.ru>
10. PRADIS – Руководство к программе. - <http://www.laduga.ru>
11. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. –Спб.: БХВ-Петербург, 2004. -608 с.
12. Лобарева И.Ф., Черный С.Г., Чирков Д.В., Скороспелов В.А., Турук П.А. Многоцелевая оптимизация формы лопасти гидротурбины. –Вычислительные технологии, Том 11, №5, 2006. С.63 – 76.
13. Филиппов С.Ж. Параметрическая идентификация систем поддержки принятия решений на основе параллельных генетических алгоритмов: Дис. ... канд. техн. наук. -Санкт-Петербург, 2003. - 152 с.
14. Карпенко А.П., Пупков К.А. Моделирование динамических систем на транспьютерных сетях. -М.: Биоинформ, 1995.-73 с.
15. Карпенко А.П., Федорук В.Г., Федорук Е.В. Исследование эффективности балансировки загрузки многопроцессорной системы при распараллеливании одного класса вычислительных задач // "Наука и образование: электронное научное издание. Инженерное образование", № гос. регистрации 0420700025 www.technomag.edu.ru, август, 2007, №0420700025/0034.