Параллельные вычисления в задачах механики космического полета*

М.Г. Широбоков[1], С.П. Трофимов[1], А.А. Целоусова[1], М.Д. Коптев[1]

Институт прикладной математики им. М.В. Келдыша РАН[1]

Доклад посвящен обзору методов параллельных вычислений для решения задач механики космического полета (МКП). Описываются особенности задач МКП, а также ряд случаев, когда задачи МКП допускают параллелизм и его эффективную реализацию. Обосновывается необходимость распараллеливания таких задач, перечисляются возможности их распараллеливания и используемые при этом технологии. Коротко описывается и существующее программное обеспечения для решения типичных задач. Приводятся примеры решения реальных задач с использованием многопроцессорных вычислительных систем.

Ключевые слова: механика космического полета, астродинамика, космический аппарат, многопроцессорная вычислительная система, параллельные вычисления, суперкомпьютер.

1. Механика космического полета

Механика космического полета, также называемая в литературе *астродинамикой*, изучает движение искусственных небесных тел, т.е. космических аппаратов (КА). При этом движение обычно разделяют на угловое (вращения спутника вокруг центра масс) и орбитальное (движение центра масс). В данной работе речь пойдет только о задачах орбитального движения, так как в литературе параллельные методы для решения задач углового движения авторам не встречались.

Движение космических аппаратов описывается системами обыкновенных дифференциальных уравнений (ОДУ), для этого используются различные модели. Например, в рамках задачи двух тел (Земля и аппарат) уравнения движения аппарата в инерциальной системе координат записываются в виде

$$\ddot{\mathbf{r}} = -\frac{\mu}{|\mathbf{r}|^3}\mathbf{r},$$

где $\mathbf{r}$ – радиус-вектор аппарата относительно Земли, $\mu = 3.986 \cdot 10^5$ км^3/с^2 – гравитационный параметр Земли [1]. В приложениях часто встречается и модель круговой ограниченной задачи трех тел, различные модели задачи четырех тел и так далее. В реальных миссиях для интегрирования межпланетных траекторий перелета пользуются высокоточной моделью движения центра масс аппарата, учитывающей гравитационное притяжение тел Солнечной системы, а также силы светового давления:

$$\ddot{\mathbf{r}} = -\frac{\mu}{|\mathbf{r}|^3}\mathbf{r} - \rho P \left(\frac{æ}{|\mathbf{r}_0 - \mathbf{r}|}\right)^2 \frac{A}{m}\,\frac{\mathbf{r}_0 - \mathbf{r}}{|\mathbf{r}_0 - \mathbf{r}|} + \sum_{i=1}^{N}\left(\frac{\mu_i}{|\mathbf{r}_i - \mathbf{r}|^3}(\mathbf{r}_i - \mathbf{r}) - \frac{\mu_i}{|\mathbf{r}_i|^3}\mathbf{r}_i\right), \tag{1}$$

где N – число учитываемых тел Солнечной системы (кроме Земли), $\mathbf{r}$ – радиус-вектор КА относительно Земли, $\mathbf{r}_0$ – радиус-вектор Солнца относительно Земли, $\mathbf{r}_i$ – положение i-го тела относительно Земли, μ_i – гравитационный параметр i-го тела, A/m – отношение площади к массе аппарата, $P = 4.56 \cdot 10^{-6}$ Н/м^2 – давление солнечного излучения на расстоянии $æ = 1$ а.е. от Солнца, $\rho \in [0, 1]$ – коэффициент освещенности; $\rho = 0$ отвечает полной тени, $\rho = 1$ отвечает полному освещению, а значения $\rho \in (0, 1)$ соответствуют ситуациям в полутени.

*Работа выполнена при поддержке Российского научного фонда (проект №17-71-10242).

Решение уравнений движения выполняется обычно методами интегрирования высокого порядка: это могут быть методы Рунге–Кутты 8 порядка с постоянным шагом [2], вложенные методы Рунге–Кутты 8(7) с адаптивным шагом [3, 4] или методы Адамса переменного порядка и с адаптивным шагом [5].

Задачи МКП – это в первую очередь задачи построения управляемого движения, удовлетворяющего целям конкретной миссии. Если аппарат оснащен двигателями для управления своим орбитальным движением, то в правых частях уравнения (1) добавляется слагаемое $\mathbf{u}$ – реактивное ускорение. Наиболее общая задача МКП и состоит в том, чтобы отыскать такую функцию $\mathbf{u}$, чтобы удовлетворить каким-то требованиям, предъявляемым к миссии. С математической точки зрения такие задачи, как правило, сводятся к задачам Коши или краевым многоточечным задачам, задачам глобальной оптимизации, а также высокоточному моделированию движения аппарата на длительных интервалах времени. Некоторые из этих задач оказываются вычислительно затратными и ресурсоемкими и требуют применения многопроцессорных вычислительных систем для решения. В следующем разделе мы представим обзор таких задач.

2. Параллельные вычисления в МКП

В задачах МКП параллельные методы достаточно активно используются в следующих областях: 1) параллельный расчет траекторий, 2) параллельные методы оптимизации и 3) параллельные методы для высокоточного моделирования. Разберем по порядку эти три категории.

2.1. Параллельный расчет траекторий

Начнем с того, что в МКП типичная ситуация – это необходимость рассчитать траекторию перелета между двумя точками пространства или между орбитами для большого числа различных значений параметров (*параметрический анализ*). Например, в случае перелета между Землей и Луной такими параметрами могут служить дата и время старта, параметры стартовой околоземной орбиты, время полета и параметры целевой окололунной орбиты. В этом случае после дискретизации возможных значений выбранных параметров задача сводится к решению независимых задач Коши – идеальный для распараллеливания случай. К последовательности независимых задач Коши сводится и моделирование испытаний Монте-Карло; это – типичная задача анализа чувствительности рассчитанной траектории: после того, как номинальное решение получено, запускается моделирование всевозможных навигационных ошибок, ошибок исполнения маневров, ошибок проектирования аппарата и так далее. В этом случае к переменным, описывающим движение аппарата, добавляются шумы в виде случайных величин. Проведенный анализ помогает разработчикам миссий выяснить, насколько «живуче» найденное номинальное решение в условиях ошибок, которые неизбежно будут присутствовать в реальной миссии.

Еще одна типичная ситуация – поиск траекторий перелета между двумя заданными точками или между заданным набором точек (краевая задача). В МКП такие задачи чаще всего решаются методами простой или параллельной пристрелки [6]. В этом случае решение краевой задачи сводится к итерационной процедуре решения последовательности независимых задач Коши. Задачи Коши могут решаться на различных ядрах вычислительной системы, а расчет вектора невязок и шага метода решения возникающей системы нелинейных уравнений приходится на некоторое выделенное, материнское, ядро.

Для задач околоземной динамики потребность в распараллеливании возникает, например, в задачах прогнозирования движения околоземных аппаратов на длительные интервалы времени [7], сопровождения каталога спутников [7, 8], моделирования поведения объектов космического мусора [7, 9], а также распространения навигационной неопределенности [10]. Например, для каталогизирования спутников или каких-либо других объектов в космосе можно применять классическую схему master-slave: ядро master отвечает за процессы ввода/вывода и раздает задания (начальные условия задач Коши) рабочим, а те, в свою очередь, их интегрируют [7] и возвращают решение ядру master.

Во всех случаях, описанных выше, параллелизм возникает из-за возможности независимым или почти независимым способом насчитывать базы данных траекторий перелета, что принципиально важно на этапе предварительного анализа миссий. Вместе с тем сами по себе численные методы интегрирования обыкновенных дифференциальных уравнений иногда тоже допускают параллелизм. В первую очередь здесь стоит упомянуть о существовании параллельных методов Рунге-Кутты [4, 11, 12]. В отличии от классических непараллельных методов Рунге-Кутты, здесь некоторые стадии для расчета шага могут вычисляться независимо от некоторых других стадий. Имеются также перспективные спектральные методы интегрирования ОДУ и решения краевых задач, основанные на итерациях метода Пикара и естественно допускающие параллелизм [13]. Отметим, что эти алгоритмы могут быть эффективно распараллелены на графических процессорах. Упомянем и работу [14], в которой представлен параллельный алгоритм интегрирования орбит, основанный на рядах Ли. Этот алгоритм позволяет получать решения с точностью до 100 значащих знаков, но расчет коэффициентов такого интегратора слишком времязатратен. Авторы [14] показали, как эти коэффициенты можно эффективно вычислить с использованием многоядерных вычислений.

Параллельные методы могут применяться для ресурсозатратных расчетов инвариантных динамических структур фазового пространства, например, двумерных торов [15]. Как показано в этой работе, такая задача сводится к решению серии систем линейных алгебраических уравнений и задач Коши, что высокоэффективно распараллеливается. Вычислительные затраты растут линейно с ростом числа параметров, описывающих двумерные торы. Мы же заметим, что этот алгоритм может быть удобен для расчета квазипериодических траекторий вокруг коллинеарных точек либрации различных систем трех тел, что имеет прямое и актуальное значение для проработки реальных миссий.

2.2. Параллельные методы оптимизации

При построении номинальной траектории перелета часто решается некоторая задача оптимального управления. Ее решение сводится к поиску локального или глобального экстремума некоторой целевой функции, например, затрат топлива на перелет или времени перелета. Здесь параллелизм возможен как на уровне целевой функции и функций ограничения, так и на уровне метода оптимизации. Сосредоточим свое внимание на параллелизации методов глобальной оптимизации.

В методах глобальной оптимизации параллелизм в первую очередь может быть достигнут в подходе мультистарта (multistart methods) [16, 17]: по определенному алгоритму создаются начальные приближения к оптимальному решению, и для каждого начального приближения независимо от других запускается какой-нибудь локальный метод оптимизации. Локальная оптимизация может выполняться на отдельных ядрах, а сбор полученной информации о решениях и создание новых точек-начальных приближений может выполняться на основном ядре.

Для глобальной оптимизации можно воспользоваться так называемыми *островными* моделями [18, 19]. Суть островной модели заключается в создании на узлах вычислительного кластера множества «островов», на которых одновременно решается поставленная оптимизационная задача. На различных островах могут быть использованы различные оптимизационные методы, при этом в процессе оптимизации острова могут обмениваться своими наилучшими решениями подобно генетическим алгоритмам. Здесь большую роль играет выбранная топология островов, от которой зависят связи между ними и пути миграций лучших решений. В разделе 4.3 мы коротко описываем решение одной из наших задач с использованием островной модели.

Отметим и развивающийся класс методов оптимизации *с помощью кривых, заполняющих пространство* и вообще методов редукции размерности [20]. Суть таких методов состоит в том, чтобы свести задачу оптимизации в n-мерной области пространства к задаче одномерной оптимизации функционала на кривой, представляющей собой ε-сеть данной области пространства. Известны параллельные алгоритмы решения возникающих при этом задач одномерной оптимизации [21, 22].

Что касается непосредственных приложений по данному пункту, упомянем параллельные варианты эволюционных алгоритмов для расчета как околоземных, так и межпланетных траекторий [23–26]. Другие приложения включают в себя проектирование формаций КА [27,28]. Интересно отметить, что в работе [27] задача оптимизации спутниковой формации представляется в виде иерархической модели: есть один общий функционал, отвечающий за формацию в целом, и несколько независимых функционалов для каждого аппарата отдельно. Получается, что на каждом шаге общей оптимизационной задачи независимо друг от друга решается серии частных оптимизационных задач. Это и есть точка параллелизма в данном случае. Что же касается работы [28], то здесь был создан целый фреймворк для параллельного обсчета траекторий множества КА на декартовой сетке из процессоров (Cartesian processor grid).

Кроме того, методы глобальной оптимизации оказываются актуальными при поиске межпланетных траекторий полета с множеством гравитационных маневров у планет [29]. Этот класс задач в МКП является одним из самых ресурсозатратных, так как обычно вклчает в себя как комбинаторную часть оптимизации, так и непрерывную с большим числом переменных. В работе [29] параллелизм достигается на уровне метода глобальной оптимизации: используется островная модель, открытое программное обеспечение PaGMO и его модификация IGATO.

Целый пласт работ также посвящен поиску оптимальных перелетов с продолжительной непрерывной тягой [30–35]. Например, в работе [30] параллелизм достигается на уровне эволюционного метода оптимизации и на уровне численного метода интегрирования уравнений движения, причем все расчеты выполняются на GPU-процессорах. В работе [34] подчеркиваются преимущества GPU-вычислений по сравнению с вычислениями на CPU, а также даются алгоритмы параллельного решения задачи оптимизации непрямым методом (через решение краевой задачи).

2.3. Параллельные методы для высокоточного моделирования

В некоторых задачах требуется высокоточное моделирование движения аппарата под действием сил светового давления и атмосферного сопротивления на больших интервалах времени [36, 37]. В этих случаях аппарат моделируется не как материальная точка, а как тело конечных размеров, что особенно важно, если аппарат имеет сложную геометрию и/или оснащен солнечным парусом и большими солнечными панелями. Для вычисления сил и моментов сил здесь можно использовать GPU-процессоры.

Кроме того, параллельные методы могут применяться для моделирования неупругих колебаний конструкций [38]. Различные методы существуют для описания движения таких конструкций. В МКП пользуются преимущественно методом Лагранжа и получают систему ОДУ. Другой подход – записать уравнения в частных производных с краевыми условиями и решать их специальными методами.

К этому же пункту отнесем и параллельные методы высокоточного расчета геопотенциала [39]. Отметим, что вообще причиной появления подобных работ является то, что классические способы высокоточного моделирования гравитационного поля (метод сферических гармоник) оказываются слишком вычислительно затратными. Поэтому были изобретены другие подходы: модели точечных масс, когда притягивающее тело заменяется на конечный набор материальных точек (возникает естественный параллелизм при расчете суммарного потенциала), а также различные интерполяционные методы на основе базисных функций (что также поддается эффективному распараллеливанию).

3. Параллельное программное обеспечение

Несмотря на большое число параллельных алгоритмов для проектирования траекторий, наблюдается относительно малое разнообразие готового программного обеспечения, направленного на расчет траекторий аппаратов и при том основанного на параллельных методах. Самыми известными здесь являются пакеты PaGMO/PyGMO, PyKEP и MIDACO [18, 40, 41], разработанные Европейским космическим агентством, и тулбокс GMAT [42], разработанный в NASA. Упомянем также библиотеку NLPAROPT для оптимизации общего назначения. Другие же широко распространенные библиотеки (OTIS, EMTG, MALTO), ориентированные конкретно на оптимизацию траекторий, а также библиотеки для оптимизации общего назначения (SNOPT, IPOPT, WORHP) состоят в основном из непараллельных алгоритмов.

4. Авторский опыт применения параллельных методов в задачах МКП

Перейдем теперь к краткому описанию конкретных задач, которые были решены авторами с использованием многоядерных вычислительных систем: на кластере К-100 и К-60 в Институте прикладной математики им. М.В. Келдыша РАН.

4.1. Метод виртуальных траекторий

Метод виртуальных траекторий – это метод, разработанный авторами для проектирования межпланетных траекторий с гравитационными маневрами [43,44]. Суть метода состоит в том, чтобы представить орбиты всех планет неподвижными в пространстве эллипсами, выбрать на этих эллипсах отдельные точки (узлы) с каким-то шагом и построить, грубо говоря, всевозможные кеплеровы дуги между этими узлами (см. рис. 1). Непрерывные цепочки из таких дуг дадут всевозможные траектории перелета в Солнечной системе, базу виртуальных траекторий. Естественно, не всякая такая цепочка подойдет с физической точки зрения: переход между дугами требует затрат топлива, которые для некоторых цепочек могут быть слишком велики, но такие цепочки можно сразу отсеивать.

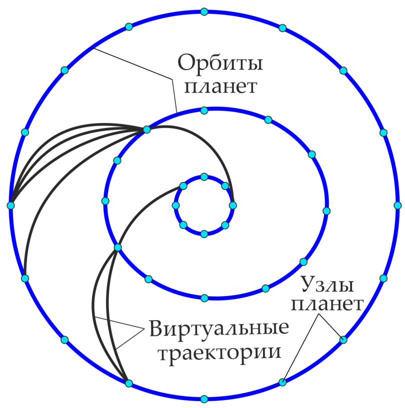


Рис. 1. Виртуальные траектории

Построение такой базы данных траекторий – это первый этап метода, его достаточно проводить для каждого маршрута (например, Земля–Венера–Юпитер или Земля–Венера–Марс–Юпитер) всего один раз. Однако в этой базе данных «все узлы соединяются со всеми узлами», и реальное положение планет не учитывается. Реальные же траектории всегда проходят через узлы на орбитах планет в то время, когда в этих узлах находятся планеты. Поэтому чтобы получить реальную траекторию, пользователь задает дополнительные параметры (например, время старта с Земли), после чего из базы выбираются виртуальные траектории, близкие к реально возможным, а потом траектории корректируются так, чтобы аппарат в моменты прохождени узлов попадал в них в точности в тот момент, когда там находится планета.

В своих исследованиях мы показали, что отсев и уточнение занимают лишь 5% от всего времени расчетов, а 95% уходит на создание базы виртуальных траекторий. Вместе с тем задача построения базы данных успешно распараллеливается [45].

Для распараллеливания задачи можно выбрать известную стратегию master-slave, когда есть главный MPI-процесс и он раздает задания рабочим MPI-процессам (см. рис. 2). Построение базы данных начинается с участка между первой и второй планетами маршрута. Рабочие процессы принимают на вход эти траектории, продляют их до третьей планеты, отправляют продленные траектории главному процессу и так далее. Траектории хранятся в столбцах массивов, массивы хранятся в файлах. Файлы раздаются рабочим, а каждый рабочий для расчетов распределяет столбцы-траектории между своими OpenMP-потоками.

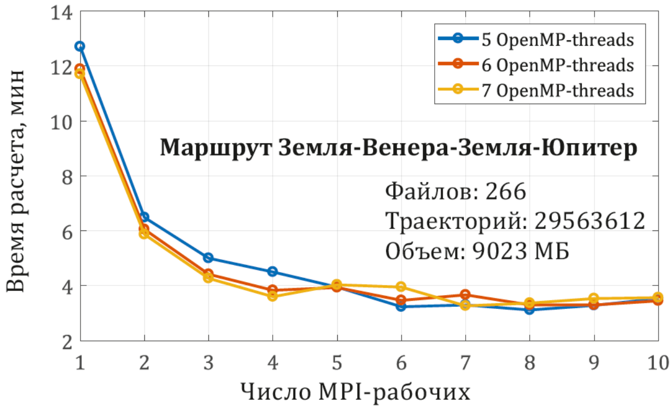
Рис. 2. Схема распараллеливания MPI+OpenMP при построении базы виртуальных траекторий

Все расчеты были проведены на вычислительном кластере К-60 в Институте прикладной математики им. М.В. Келдыша РАН [46]. Мы протестировали время расчетов в различных ситуациях (см. рис. 3). Например, расчеты выполнялись на нескольких узлах и на одном узле, исследовали различное число MPI-процессов и OpenMP-потоков, разные флаги оптимизации и опции библиотеки MKL. Мы выяснили, что наибольшее влияние на производительность оказывают число MPI-процессов и OpenMP-потоков: и тех и других должно быть около 5-7 для быстрой работы алгоритма. Кроме того, OpenMP-потоков не должно быть слишком много, так как они встают в очередь на отправку главному процессу данных. MPI-процессов тоже не должно быть слишком много, так как они будут простаивать в самом начале, когда созданных виртуальных траекторий еще мало и, соответственно, мало файлов.

4.2. Прямая оптимизация траекторий с малой тягой

На сегодня существует множество методов оптимизации траекторий перелета с малой тягой, т.е. непрерывной на продолжительных интервалах времени реактивной тягой, медленно изменяющей параметры орбиты аппарата. Эти методы принято делить на прямые и непрямые.

Непрямые методы основываются на принципе максимума Понтрягина или Беллмана. Наиболее существенным недостатком непрямых методов можно назвать высокую чувствительность поведения итерационной процедуры расчета сопряженных переменных к их на-



OpenMP-threads = 1

Число MPI-рабочих	Время расчета
25	7 мин 05 сек
30	7 мин 32 сек
35	7 мин 34 сек
40	9 мин 59 сек

OpenMP-threads = 28

Число MPI-рабочих	Время расчета
1	16 мин 47 сек
2	9 мин 42 сек

Возможный разброс времени расчета: 2-3%

Рис. 3. Результаты распараллеливания при различном числе MPI-процессов и OpenMP-потоков

чальному приближению, хотя в некоторых случаях известны регулярные методы построения оптимального движения.

Прямые методы основываются на дискретизации функции управления и сведения исходной задачи к задаче математического программирования. Практика показывает, что найти решение в этом случае удается существенно проще, чем непрямыми методами, хотя решение и получается квазиоптимальным. Более того, траектория в этих случаях состоит из независимых участков, поэтому ее интегрирование естественно распараллеливается. Прямые методы оптимизации наиболее важны на этапе предварительного анализа миссий для полного исследования возможностей перелета. Иногда решение, найденное прямым методом, используется в качестве начального приближения в итерационных процедурах поиска оптимального решения непрямыми методами, поэтому очень важно, чтобы задача была решена прямыми методами как можно быстрее.

В работе [47] мы распараллелили, исследовали и сравнили два широко распространенных прямых метода оптимизации: в одном случае управление аппроксимируется кусочно-постоянной функцией, а в другом случае – импульсами. Для того чтобы сравнить методы, необходимо было перебрать большое число начальных условий и подсчитать время расчетов и число итераций. Множество начальных условий делилось на равные доли, задача запускалась на одном узле и 28 ядрах. Как и ожидалось, из-за независимости расчетов эффективность распараллеливания была максимальной, расчеты прошли в 28 раз быстрее, чем если бы они были проведены на одном ядре той же вычислительной системы. Этот анализ позволил нам в сжатые сроки выявить сильные и слабые исследуемых алгоритмов.

4.3. Поиск оптимальных конфигураций спутниковых группировок

Еще одна задача, решенная авторами, – построение тетраэдральных формаций спутников на высокоэллиптической околоземной орбите [48]. В этой работе с помощью вычислительного кластера К-60 мы находили оптимальные орбиты спутников для обеспечения наилучшего качества геометрической конфигурации формации. Скажем, для исследования отдаленных областей земной магнитосферы требуется вывести на близкие высокоэллиптические орбиты минимум 4 аппарата так, чтобы конфигурация формации сохраняла геометрию правильного (или почти правильного) тетраэдра на протяжении порядка двух месяцев. Соответствующая оптимизационная задача имеет большую размерность – от 18 до 24 переменных – и чрезвычайно чувствительна к выбору начального приближения по причине многоэкстремальности.

Для решения данной задачи мы воспользовались островной моделью глобальной оптимизации. А именно, в отличие от стандартных схем, когда каждое отдельное ядро загружено своей оптимизационной задачей, отвечающей некоторому вектору в сканируемом пространстве параметров системы, и работает автономным образом, в новой схеме на разных ядрах решается одна и та же задача математического программирования, но разными оптимизаторами, асинхронно обменивающимися друг с другом информацией. Подобный обмен, похожий на эволюционный механизм генетических алгоритмов оптимизации, определяется топологией островной модели, т.е. структурой информационных потоков между различными островами-оптимизаторами. Сочетание методов глобальной и локальной оптимизации позволяет обнаруживать глобальные экстремумы гладких и негладких функционалов с разнообразного вида ограничениями. Островная архитектура естественным образом масштабируема: она дает возможность эффективно задействовать любые доступные вычислительные мощности.

В качестве топологии архипелага была выбрана достаточно популярная кольцевая структура с центральным островом, реализующим один из методов локальной оптимизации, и преобладанием глобальных оптимизаторов на периферии (на рис. 4 изображен пример с девятью островами). Каждый из периферийных островов обменивается информацией с двумя соседями, а также с центральным островом.

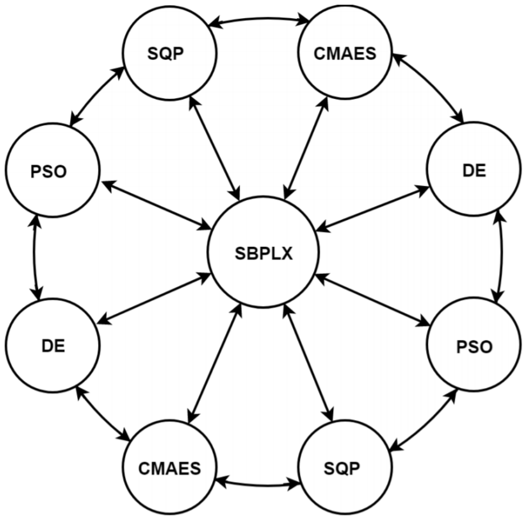


Рис. 4. Пример расположения девяти островов и используемых алгоритмов при оптимизации на суперкомпьютере

По результатам тестирования для глобальной оптимизации были выбраны алгоритмы дифференциальной эволюции (DE), эволюции ковариационной матрицы (CMAES), роя частиц (PSO) и последовательного квадратичного программирования (SQP). Начальные приближения для алгоритмов глобальной оптимизации генерировались равномерным распределением в заданных границах, что обеспечило наилучшее покрытие исследуемой области. Для центрального острова был выбран subplex-алгоритм (SBPLX), являющийся модификацией метода Нелдера-Мида. Политика мигрирования была выбрана такой, что наилучшие решения с островов кольца передавались на центральный остров, уточнялись процедурой локальной оптимизации, а затем возвращались обратно на внешние острова. Программная реализация выбранных методов была позаимствована из популярной в космическом сообществе библиотеки PaGMO (от слов Parallel Global Multiobjective Optimiser), разработанной в Европейском космическом агентстве [18]. Архипелаг состоял из 361 острова (включая один центральный), каждый из которых был приписан к своему ядру кластера. Номинальное решение задачи, не учитывающее ошибок выведения КА на орбиту, позволяет без какого бы то ни было управления сохранять требуемую геометрию тетраэдральной формации в течение почти одного года, что значительно превосходит практические потребности. Аналогичные расчеты, осуществляемые в последовательном режиме в среде MATLAB на персональном компьютере, отнимают много машинного времени и, что еще хуже, не сходятся при недостаточно хорошем начальном приближении для локального оптимизатора.

4.4. Оптимизация схемы облета объектов космического мусора

В апреле 2017 года авторы приняли участие в престижном международном соревновании Global Trajectory Optimization Competition (GTOC), в котором участовало 69 коллективов со всего мира [49, 50].

Типичные задачи GTOC – это комбинаторно-непрерывные задачи глобальной оптимизации. В этот раз задача состояла в том, чтобы определить оптимальную схему облета 123 объектов космического мусора (в рамках нескольких пусков с Земли) с учетом ограничений на время перелета между объектами, время между пусками, общего времени отводимого на все миссии, орбитальные параметры КА, затраты топлива и число импульсов для коррекции орбиты. Функционал включал в себя как финансовые затраты на все пуски, так и время на решение задачи у коллектива [51].

Задача была разделена на две части. Сначала комбинаторная часть: перебор всевозможных последовательностей облета объектов, переходы между которыми наиболее дешевы с точки зрения изменения орбитальных параметров КА; здесь использовался переборный алгоритм Кнута, Knuth's Algorithm X. Этот первый этап позволял отсеять заранее большое число неудовлетворяющих требованиям последовательностей, хотя затраты и оценивались приближенными методами. Затем следовала непрерывная часть: для каждой найденной последовательности стандартными градиентными методами «честно» численно решалась оптимизационная задача перелета, минимизирующая затраты топлива при всех данных ограничениях.

Комбинаторная часть задачи оказалась наиболее ресурсозатратной, поэтому ее расчеты проводились на кластере К-100 в Институте прикладной математики им. М.В. Келдыша РАН. Для ее решения использовалось 384 потока на 32 узлах, что составляло половину всего кластера К-100. Расчеты комбинаторной части занимали 10 часов. Непрерывная часть рассчитывалась на персональных компьютерах и не занимала много времени.

Наш коллектив авторов стал единственной командой из России, представившей решение этой оптимизационной задачи. Из 123 объектов мусора в течение 8 лет нам удалось облететь и убрать 112, что дало итоговое 26 место. Первое место заняла команда Лаборатории реактивного движения NASA.

5. Заключение

Многие задачи механики космического полета оказываются чрезвычайно ресурсоемкими. Причины этого состоят в том, что на предварительном этапе анализа миссии приходится сканировать многомерное пространство параметров с учетом всех ограничений. Часто это выражается в независимых или почти независимых циклах по различным параметрам перелета. Кроме того, каждая итерация такого цикла может представлять собой не только решение задачи Коши или краевую задачу, но и решение некоторой задачи оптимизации. Поэтому задачи механики космического полета довольно естественно допускают параллелизм; его можно выразить как на уровне постановки задачи, так и на уровне численных методов: параллельные методы интегрирования обыкновенных дифференциальных уравнений, решения краевых задач или, в некоторых специальных случаях, уравнений в частных производных.

Наиболее актуальной темой авторам кажется глобальная оптимизация траекторий перелета, в особенности – с использованием островных моделей. Это та область механики космического полета, которая даже с учетом разработанных (полу)аналитических результатов оказывается вычислительно затратной, требует применения многоядерных вычислительных систем для своего решения, но помогает решить очень важные и насущные задачи науки и техники. Еще одно важное направление, которое мы бы хотели выделить, это массивные расчеты траекторий с малой тягой на GPU-процессорах; эффективность такого распараллеливания уже была показана ранее другими исследователями.

Список литературы

1. Охоцимский Д. Е. Основы механики космического полета. Наука. Гл. ред. физ.-мат. лит., 1990.

2. Shanks E. B. Higher order approximations of runge-kutta type // NASA TN D-2920, George C. Marshall Space Flight Center, Huntsville,Ala., September. 1965.

3. Prince P. J., Dormand J. R. High order embedded Runge-Kutta formulae // Journal of Computational and Applied Mathematics. 1981. Vol. 7, no. 1. P. 67–75.

4. Hairer E., Nørsett S., Wanner G. Solving Ordinary Differential Equations I. Springer-Verlag, 1996.

5. Холл Дж., Уатт Дж. Современные численные методы решения обыкновенных дифференциальных уравнений / под ред. А. Д. Горбунова. М.: Мир, 1979.

6. Pavlak T. A. Mission Design Applications in the Earth–Moon System. MS Thesis, Purdue University, West Lafayette, Indiana, USA. 2010.

7. Coffey S., Healy L.and Neal H. Applications of parallel processing to astrodynamics // International Astronomical Union Colloquium / Cambridge University Press. Vol. 165. 1997. P. 61–70.

8. Search and determine integrated environment (SADIE) for space situational awareness: Tech. Rep.: / C. A. Sabol, A. Segerman, A. Hoskins et al.: NAVAL Research LAB Washington DC, 2012.

9. Using parallel computing for the display and simulation of the space debris environment / M. Möckel, C. Wiedemann, S. Flegel et al. // Advances in Space Research. 2011. Vol. 48, no. 1. P. 173–183.

10. Massari M., Di Lizia P., Rasotto M. Nonlinear Uncertainty Propagation in Astrodynamics: Comparing Taylor Differential Algebra with Monte-Carlo on GPUs // 26th AAS/AIAA Space Flight Mechanics Meeting. 2016. P. 14–18.

11. Van der Houwen P. J., Sommeijer B. P., huu Cong N. Parallel diagonally implicit Runge-Kutta-Nyström methods // Applied numerical mathematics. 1992. Vol. 9, no. 2. P. 111–131.

12. Aristoff J., Poore A. Implicit Runge-Kutta methods for orbit propagation // AIAA/AAS Astrodynamics Specialist Conference. 2012. P. 4880.

13. Bai X. Modified Chebyshev–Picard Iteration Methods for Solution of Initial Value and Boundary Value Problems. PhD. Thesis. 2010.

14. Mai E., Geyer R. Numerical orbit integration based on Lie series with use of parallel computing techniques // Advances in Space Research. 2014. Vol. 53, no. 1. P. 77–89.

15. Jorba À., Olmedo E. On the computation of reducible invariant tori on a parallel computer // SIAM Journal on Applied Dynamical Systems. 2009. Vol. 8, no. 4. P. 1382–1404.

16. Scatter search and local NLP solvers: A multistart framework for global optimization / Z. Ugray, L. Lasdon, J. Plummer et al. // INFORMS Journal on Computing. 2007. Vol. 19, no. 3. P. 328–340.

17. Glover F. A template for scatter search and path relinking // Lecture notes in computer science. 1998. Vol. 1363. P. 13–54.

18. Biscani F., Izzo D., Yam C. H. A global optimisation toolbox for massively parallel engineering optimisation // arXiv preprint arXiv:1004.3824. 2010.

19. Izzo D., Ruciński M., Biscani F. The generalized island model // Parallel Architectures and Bioinspired Algorithms. Springer, 2012. P. 151–169.

20. Sergeyev Y. D., Strongin R. G., Lera D. Introduction to global optimization exploiting space-filling curves. Springer Science & Business Media, 2013.

21. Strongin R. G., Sergeyev Y. D. Global optimization with non-convex constraints: Sequential and parallel algorithms. Springer Science & Business Media, 2013. Vol. 45.

22. Баркалов К. А., Лебедев И. Г., Соврасов В. В. Сравнение схем редукции размерности для параллельных алгоритмов глобальной оптимизации // Суперкомпьютерные дни в России. Москва: 2018.

23. Lee S., Russell R. P. Multi-objective parallel genetic algorithms applied to the primer vector control law // Paper AAS. 2006. P. 06–196.

24. McGrath C. B., Proulx R. J., Karpenko M. Parallel Genetic Algorithms For Optimal Control // 26th AAS/AIAA Space Flight Mechanics Meeting. Paper AAS 16-367.

25. Ferringer M. P., Clifton R. S., Thompson T. G. Efficient and accurate evolutionary multi-objective optimization paradigms for satellite constellation design // Journal of Spacecraft and Rockets. 2007. Vol. 44, no. 3. P. 682–691.

26. Evolutionary Computing for Low Thrust Navigation / S. Lee, W. Fink, R. Russell et al. // Space 2005. 2005. P. 6835.

27. Efficient control of formation flying spacecraft / M. Dellnitz, O. Junge, A. Krishnamurthy et al. // Meyer auf der Heide, F., Monien, B.(eds.) New Trends in Parallel & Distributed Computing. 2006. P. 235–247.

28. Ghosh A. Multi-CubeSat mission planning enabled through parallel computing. PhD. Thesis: University of Illinois at Urbana-Champaign. 2013.

29. Bryan J. M. Global Optimization of MGA-DSM Problems Using the Interplanetary Gravity Assist Trajectory Optimizer (IGATO). 2011.

30. Wittig A., Wase V., Izzo D. On the use of GPUs for massively parallel optimization of low-thrust trajectories. 2016. URL: `http://www.esa.int/gsp/ACT/doc/MAD/pub/ACT-RPR-MAD-2016-ICATT_GPU_preprint.pdf`.

31. Massari M., Wittig A. Optimization of multiple-rendezvous low-thrust missions on general-purpose graphics processing units // Journal of Aerospace Information Systems. 2015. P. 1–13.

32. High-speed, high-fidelity low-thrust trajectory optimization through parallel computing and collocation methods / J. F. C. Herman, J. S. Parker, B. A. Jones et al. // Advances in the Astronautical Sciences Spaceflight Mechanics. 2015. Vol. 2015.

33. An Algorithm for Generating Feasible Low Thrust Interplanetary Trajectories / P. R. Patel, A. D. Gallimore, T. H. Zurbuchen et al. // Proceedings of theInternational Electric Propulsion Conference, Michigan, USA. 2009.

34. Antony T. Rapid Indirect Trajectory Optimization On Highly Parallel Computing Architectures. PhD Thesis: Purdue University. 2014.

35. Lantoine G. A methodology for robust optimization of low-thrust trajectories in multi-body environments. PhD. Thesis: Georgia Institute of Technology. 2010.

36. Kenneally P. W., Schaub H. High Geometric Fidelity Modeling of Solar Radiation Pressure Using Graphics Processing Unit // AAS/AIAA Spaceflight Mechanics Meeting, Napa Valley, CA. 2016.

37. Tanygin S., Beatty G. M. GPU-accelerated computation of SRP forces with graphical encoding of surface normals // AAS/AIAA Astrodynamics Specialist Conference, Vail, CO. 2015.

38. Junkins J. L., Kim Y. Introduction to dynamics and control of flexible structures. American Institute of Aeronautics and Astronautics, 1993.

39. Arora N. High performance algorithms to improve the runtime computation of spacecraft trajectories. PhD. Thesis: Georgia Institute of Technology. 2013.

40. Pagmo & Pygmo. URL: `https://esa.github.io/pagmo2/index.html`.

41. Официальный сайт MIDACO. URL: `http://www.midaco-solver.com/index.php/about`.

42. Hughes S. P. General Mission Analysis Tool (GMAT) // International Conference on Astrodynamics Tools and Techniques (ICATT). Darmstadt, Germany: 2016.

43. Овчинников М.Ю., Трофимов С.П., Широбоков М.Г. Метод виртуальных траекторий для проектирования межпланетных миссий с гравитационными маневрами // Космические исследования. 2013. Т. 51, № 6. С. 484–496.

44. Широбоков М.Г., Трофимов С.П., Овчинников М.Ю. Проектирование межпланетных траекторий с пассивными гравитационными маневрами и импульсами в глубоком космосе // Космические исследования. 2018. Т. 56, № 4. С. 57–70.

45. Широбоков М.Г., Трофимов С.П., Овчинников М.Ю. Создание базы данных межпланетных траекторий на многопроцессорных вычислительных системах // Суперкомпьютерные дни в России. Москва: 2018.

46. Вычислительный комплекс К-60. URL: `http://www.kiam.ru/MVS/resourses/k60.html`.

47. Trofimov S., Tselousova A., Shirobokov M. Two Direct Low Thrust Trajectory Optimization Techniques // Journal of Computer and Systems Sciences International. 2018. Vol. 57, no. 6. P. 989–1000.

48. Коптев М.Д., Трофимов С.П. Проектирование, развертывание и поддержание тетраэдральной формации наноспутников на высокоэллиптических орбитах // Препринты ИПМ им. М.В. Келдыша. 2018. № 97. С. 1–28.

49. Новость об участии в Global Trajectory Optimization Competition. URL: `http://keldysh.ru/microsatellites/old_news/news-2017/04_05_2017.html`.

50. The global trajectory optimisation competition portal. URL: `https://sophia.estec.esa.int/gtoc_portal/?page_id=814`.

51. Постановка задачи GTOC 9. URL: `https://sophia.estec.esa.int/gtoc_portal/wp-content/uploads/2017/05/gtoc9_problem_stmt.pdf`.