

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

Том 9, №3 (2017) <http://naukovedenie.ru/vol9-3.php>

URL статьи: <http://naukovedenie.ru/PDF/102TVN317.pdf>

Статья опубликована 06.07.2017

Ссылка для цитирования этой статьи:

Моисеев Д.В., Чинь Ван Минь Вычислительные аспекты и прикладное программное обеспечение оптимальной маршрутизации полета легкого беспилотного летательного аппарата в поле постоянного ветра // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 9, №3 (2017) <http://naukovedenie.ru/PDF/102TVN317.pdf> (доступ свободный). Загл. с экрана. Яз. рус., англ.

УДК 629.7.076:519.854.33

Моисеев Дмитрий Викторович

ФГБОУ ВО «Московский авиационный институт (национальный исследовательский университет)», Россия, Москва
Доцент кафедры «Системный анализ и управление»
Кандидат технических наук
E-mail: moiseev801@mail.ru
РИНЦ: https://elibrary.ru/author_items.asp?id=741520

Чинь Ван Минь¹

ФГБОУ ВО «Московский авиационный институт (национальный исследовательский университет)», Россия, Москва
Аспирант
E-mail: trinhminh6666@gmail.com

Вычислительные аспекты и прикладное программное обеспечение оптимальной маршрутизации полета легкого беспилотного летательного аппарата в поле постоянного ветра

Аннотация. В статье проведен сравнительный анализ нескольких вариантов разработанного авторами программно-алгоритмического обеспечения решения задач маршрутизации полета легкого беспилотного летательного аппарата в поле постоянного ветра. На основе этого анализа рекомендован наиболее эффективный вариант такого обеспечения. Процедуры построения наискорейшего маршрута облета точек, заданных своим местоположением, сведены к решению определенным образом составленных задач булева линейного программирования. Варианты программного обеспечения, разработанные и исследуемые авторами, включают в свой состав различные стандартные программы из пакетов MATLAB и CPLEX, предназначенные для решения задач целочисленного линейного программирования, а также различные варианты специального программного обеспечения, дополняющего стандартное программное обеспечение. При сравнении вариантов учитывались полученные оценки быстродействия и используемого объема оперативной памяти. Определены предельные размеры задач, устойчиво решаемых с помощью разработанного программно-алгоритмического обеспечения. Приведено описание принципов построения программного комплекса, решающего задачи маршрутизации. Особенностью разработанного прикладного программного обеспечения является его открытая архитектура, а также наличие быстродействующего программного ядра, использующего функцию `cxplexbilp`

¹ 125480, Москва, улица Вилиса Лациса, дом 14, комната 1302

пакета CPLEX, и специализированного периферийного программного обеспечения, обеспечивающего работу ядра.

Работоспособность программно-алгоритмического обеспечения продемонстрирована на примере расчета замкнутого наискорейшего маршрута облета 165 точек, который был выполнен за 132 с.

Ключевые слова: маршрутизация полета; беспилотный летательный аппарат; задача булева линейного программирования; последовательное исключение подциклов; bintprog; cplexbipr; программно-алгоритмическое обеспечение

Введение

В [1-4] достаточно подробно рассмотрено техническое содержание задач построения оптимальных маршрутов облета беспилотным летательным аппаратом (БПЛА) набора точек, заданных своими координатами. Показана целесообразность математической трактовки этой проблемы как задачи линейного программирования с булевыми переменными. Такой подход к математической формализации задач маршрутизации открывает хорошие перспективы для рассмотрения широкого спектра постановок. Например, для составления оптимальных маршрутов с учетом ограничения на время полета [5]. Однако в указанных работах вопросы алгоритмической и программной реализации в деталях не обсуждались. В тоже время эти вопросы весьма актуальны, поскольку вычислительные проблемы, возникающие при решении задач линейного программирования с булевыми переменными, хорошо известны [6]. В литературе [7] имеются ориентированные исключительно на математическую сторону дела исследования эффективности различных алгоритмов решения таких задач и реализующих их вычислительных процедур. Отдельно обсуждались вопросы программной реализации различных алгоритмов и вычислительных процедур решения задач указанного типа, исследовалась их эффективность [8, 9]. Естественно перечисленные вопросы рассматривались в отрыве от специфики конкретных технических задач. В практических ситуациях алгоритмы, вычислительные процедуры и их программная реализация представляют собой единый комплекс. Поэтому в данной статье акцент сделан на вопросах рационального комплексирования алгоритмов и вычислительных процедур с их последующей программной реализацией при учете специфики задач маршрутизации полета БПЛА.

Маршрутизация полета и ее программно-алгоритмическое обеспечение

В качестве исходных данных для составления маршрута полета используется информация о местоположении точек, которые надо связать маршрутом, воздушной скорости БПЛА, характеристиках ветра в зоне полета и при необходимости иная дополнительная информация, обусловленная спецификой рассматриваемой задачи маршрутизации. Результат решения задачи маршрутизации представляет собой последовательность облета заданных точек. Учет ветра при составлении маршрута полета является для легких аппаратов, имеющих относительно невысокую воздушную скорость, принципиальной. В наиболее простой постановке задача маршрутизации формализуется математически как задача коммивояжера. Критерием оптимальности, подлежащим минимизации, является время полета по маршруту. Особенностью задачи маршрутизации полета с учетом ветра является то, что элементы матрицы коммивояжера представляют собой минимально возможное время перелета между парами точек маршрута. Таким образом, матрица коммивояжера составляется в результате решения соответствующей задачи оптимизации. В поле постоянного ветра, когда его скорость и направление в зоне полета остаются неизменными в течение всего полета, решение такой задачи получено в [4]. В более сложных постановках задач маршрутизации ее математическая

формализация также усложняется. Например, такая ситуация возникает, когда необходимо обеспечить учет дополнительных ограничений. В частности, в [5] оптимальный маршрут находится с учетом ограничения на допустимую продолжительность полета БПЛА. Или может потребоваться учет неравноценности точек, включаемых в маршрут [10]. Такие задачи уже не могут быть сведены к классической постановке коммивояжера. Поэтому предложено математически формализовать их как задачи линейного программирования с булевыми переменными.

Из сказанного следует, что программно-алгоритмическое обеспечение маршрутизации полета должно обеспечить выполнение двух основных групп операций. Во-первых, на основе исходных данных задачи маршрутизации надо сформировать соответствующую запись задачи булева линейного программирования. А, во-вторых, обеспечить решение этой задачи. Соответственно в программное обеспечение маршрутизации входят две группы программ. Одна из них отвечает за формирование задачи булева линейного программирования, а другая отвечает за ее решение. Рациональной архитектурой прикладного программного обеспечения в такой ситуации будет программное ядро, обеспечивающее собственно решение задачи булева линейного программирования и периферийное программное обеспечение обработки исходных данных для формирования записи указанной задачи. Для реализации ядра целесообразно использовать стандартное программное обеспечение. Это позволяет существенно экономить время на написание программ, упрощает их верификацию и обеспечивает высокую вычислительную производительность. При выборе конкретного стандартного обеспечения необходимо кроме его функциональных возможностей учесть его фактическую доступность. С учетом сказанного весьма перспективными представляются такие программные продукты как MATLAB и CPLEX. Однако следует отметить, что процедуры `bintprog` и `cplexbipr`, входящие в эти продукты, обеспечивают решение задачи о назначениях. Для решения с их помощью задач маршрутизации необходимо обеспечить исключение из возможных решений задачи о назначениях решений, соответствующих так называемым подциклам, то есть маршрутам короче гамильтонова контура. А при рассмотрении более сложных постановок задач маршрутизации с дополнительными ограничениями необходимо не просто исключить подциклы, а предусмотреть их специальную обработку. Задачи, решаемые периферийным программным обеспечением, являются в основном высоко специализированными, поэтому использовать для его реализации стандартные программы затруднительно. Однако и требования, предъявляемые к нему, например, в части быстродействия, объективно являются менее жесткими.

Процедура и результаты сравнительного тестирования вариантов программно-алгоритмического обеспечения маршрутизации полета

Сравнительное тестирование вариантов программно-алгоритмического обеспечения проводилось путем решения задачи поиска замкнутого маршрута наискорейшего облета всех заданных своим положением точек с учетом постоянного ветра в зоне полета. При тестировании оценивалось время решения задачи и необходимый объем оперативной памяти в зависимости от размера задачи, определяемого количеством точек, которые связывались маршрутом. Оценки времени для задачи каждого размера получались в результате осреднения по нескольким вариантам расположения точек в зоне полета. Кроме того, определялись предельные размеры задач, которые устойчиво решались тестируемым программно-алгоритмическим обеспечением.

В результате тестирования были построены тренды изменения вышеупомянутых показателей в зависимости от размера тестовой задачи маршрутизации. В процессе

тестирования было установлено, что для выявления стабильного тренда было достаточно осреднять результаты по пятнадцати вариантам расположения точек.

Все расчеты были проведены на компьютере с процессором Intel(R) i3-4160 CPU@3.60GHz и оперативной памятью 16,00 Гб.

Отличительной особенностью первого варианта программно-алгоритмического обеспечения является то, что он предусматривает одномоментное внесение в запись задачи булева линейного программирования множества ограничений, исключающих все возможные подциклы, и использование функции `bintprog` пакета MATLAB. Указанное множество ограничений

$$\sum_{i \in J} \sum_{j \in I \setminus J} x_{ij} \geq 1, \quad J \subset I, \quad 2 \leq |J| \leq |I| - 2 \quad (1)$$

требует для любого подмножества J множества номеров точек I элемента маршрута, ведущего в точку, не принадлежащую этому подмножеству [11]. Очевидным недостатком этого варианта является то, что ограничения, запрещающие подциклы, существенно увеличивают размер задачи булева линейного программирования, кроме того количество этих ограничений и, соответственно размер задачи заметно растут с увеличением количества точек, которые надо связать маршрутом.

Результаты компьютерного тестирования первого варианта программно-алгоритмического обеспечения приведены на рис. 1 и рис. 2.

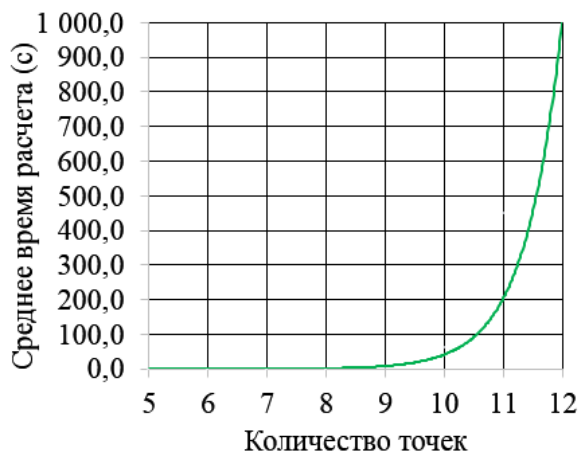


Рисунок 1. Зависимость среднего времени расчета от количества точек в маршруте – вариант 1 (разработан авторами)

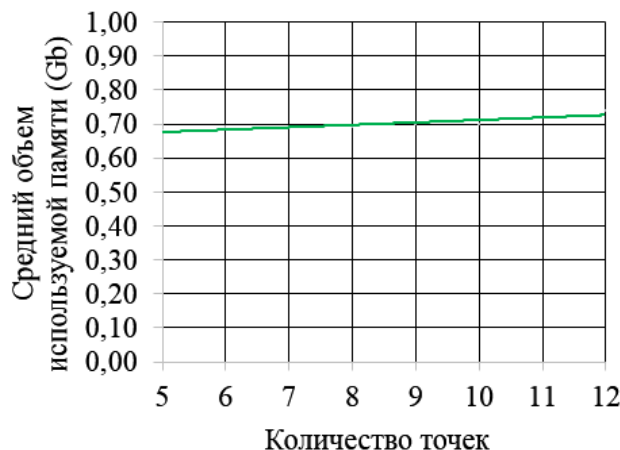


Рисунок 2. Зависимость среднего объема используемой памяти от количества точек в маршруте – вариант 1 (разработан авторами)

Как следует из графика, приведенного на рис. 1, начиная с 11 точек наблюдается резкий рост времени решения задачи маршрутизации. Фактически для случая 11 точек решение задачи маршрутизации с помощью первого варианта программно-алгоритмического обеспечения становится проблематичным. Объем памяти, используемой при расчетах, практически не зависит от количества точек в маршруте и лежит в диапазоне от 0,68 до 0,74 Гб.

Особенностью второго варианта программно-алгоритмического обеспечения по сравнению с первым является использование процедуры последовательного исключения подциклов [12]. Идея этой итеративной процедуры заключается в том, что сначала задача булева линейного программирования решается вообще без учета условий, исключающих

появление в решении подциклов. При появлении таковых, исходная запись задачи дополняется ограничением, исключающим как минимум один из появившихся подциклов. Пусть, например, при общем количестве точек n в решении появится подцикл длиной k , ($2 \leq k \leq n - 2$), который связывает точки с номерами i_1, i_2, \dots, i_k в указанном порядке. Пусть S – множество номеров точек этого подцикла. Для его исключения, согласно [12], следует добавить ограничение

$$\sum_{\substack{i \in S \\ j \in S \\ i \neq j}} x_{ij} \leq k - 1. \tag{2}$$

Ограничение (2) означает, что из k дуг, образующих рассматриваемый подцикл, в следующем решении могут присутствовать не более $k - 1$ дуг. В [13] доказано, что процедура является сходящейся, то есть в итоге будет получено решение, не содержащее подциклов.

Преимущество этого варианта в сравнении с предыдущим неочевидно. С одной стороны размер задачи булева линейного программирования, решаемой на каждой итерации, не увеличивается за счет ограничений, запрещающих все возможные подциклы. С другой стороны, задачу булева линейного программирования приходится решать многократно. Кроме того, появляется необходимость в проверке наличия подциклов в решениях, получаемых на каждой итерации.

Как следует из графика, приведенного на рис. 3, резкий рост времени, необходимого для решения задачи маршрутизации, наблюдается в районе 45 точек. Таким образом, второй из исследуемых вариантов демонстрирует заметные преимущества по сравнению с первым. Объем памяти, используемой при расчетах, как видно из рис. 4, существенно не отличается от первого варианта и практически не меняется с ростом количества точек в маршруте.

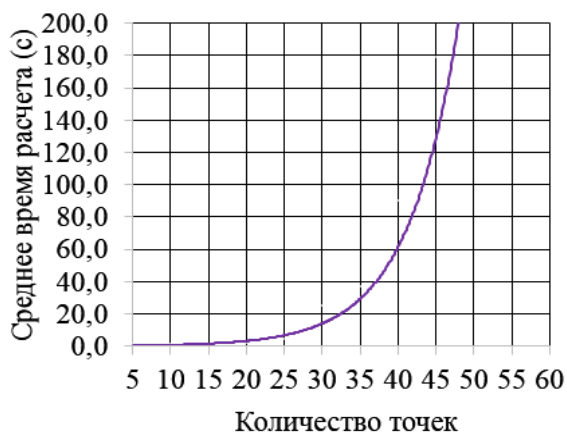


Рисунок 3. Зависимость среднего времени расчета от количества точек в маршруте – вариант 2 (разработан авторами)

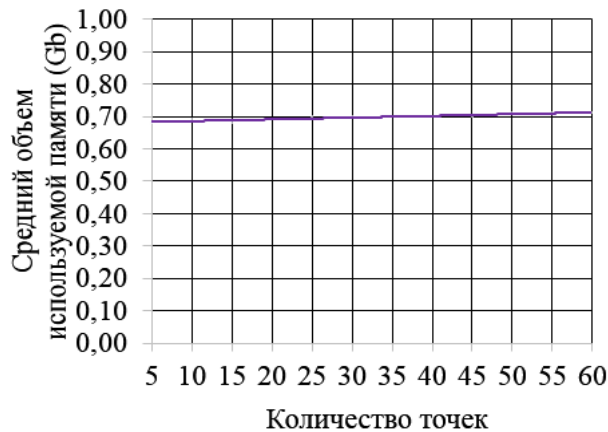


Рисунок 4. Зависимость среднего объема используемой памяти от количества точек – вариант 2 (разработан авторами)

Третий вариант программно-алгоритмического обеспечения также использует процедуру последовательного исключения подциклов, но в отличие от первых двух для решения задачи булева линейного программирования в нем применяется функция `splxlp` пакета CPLEX.

Как следует из графика, приведенного на рис. 5, резкий рост времени, необходимого для решения задачи маршрутизации, начинается в районе 150-160 точек. Таким образом, третий из исследуемых вариантов по этому показателю оказывается наиболее эффективным. Объем памяти, используемой при расчетах, постепенно растет с увеличением количества

точек в маршруте. При решении задачи с одинаковым количеством точек в маршруте третий вариант использует примерно на 15% больший объем памяти. Однако это обстоятельство не представляется существенным, поскольку для современных компьютеров ограничения по ресурсу оперативной памяти не является критичным.



Рисунок 5. Зависимость среднего времени расчета от количества точек – вариант 3 (разработан авторами)

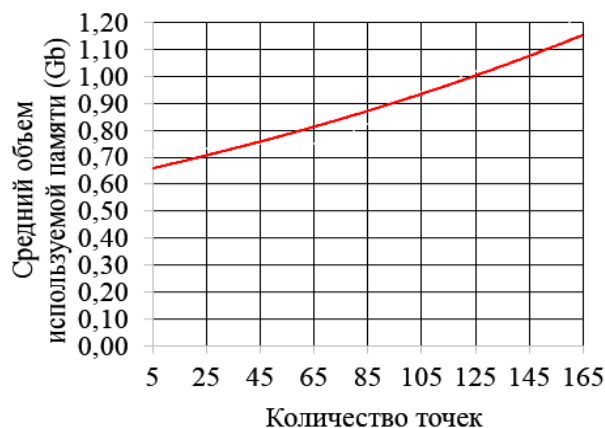


Рисунок 6. Зависимость среднего объема используемой памяти от количества точек – вариант 3 (разработан авторами)

Для удобства сравнения графики, приведенные на рис. 1, 3 и 5, изобразим совместно на рис. 7, а графики, приведенные на рис. 2, 4 и 6, совместим на рис. 8. Из графиков, приведенных на рис. 7, хорошо видно, что характер увеличения времени расчета с ростом количества точек, связываемых маршрутом, для всех рассмотренных вариантов аналогичен. Когда количество точек достигает определенного значения, начинается резкий рост времени решения задачи. Это сопровождается неустойчивой работой программного обеспечения, когда решение при некоторых исходных данных получить уже не удастся. Для первого варианта программно-алгоритмического обеспечения такой эффект встречается уже начиная с 11 точек, для второго с 45, а для третьего со 165 точек.

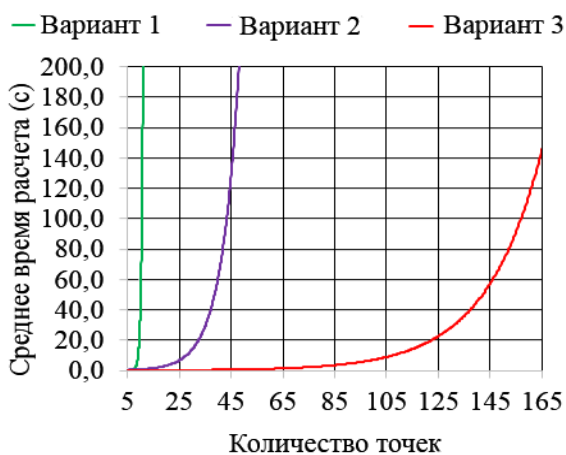


Рисунок 7. Зависимость среднего времени расчета от количества точек (разработан авторами)

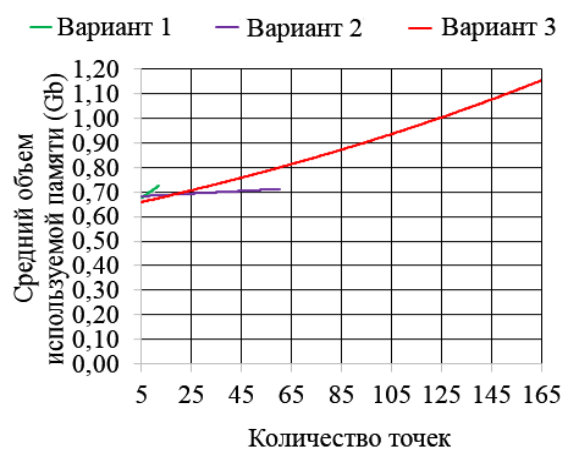


Рисунок 8. Зависимость среднего объема используемой памяти от количества точек (разработан авторами)

Прикладное программное обеспечение оптимальной маршрутизации полета легкого беспилотного летательного аппарата

Архитектура программного обеспечения, разрабатываемого авторами, является открытой и предполагает дальнейшее расширение функциональных возможностей. В качестве примера рассмотрим подробнее реализацию третьего варианта программно-алгоритмического обеспечения маршрутизации облета равноценных точек в поле постоянного ветра без учета дополнительных ограничений, которая реализована в среде MATLAB версии 2012 с использованием функции `splexbilp` пакета CPLEX.

Укрупненная блок-схема алгоритма работы программного обеспечения маршрутизации полета с использованием процедуры последовательного запрета подциклов приведена на рис. 9.

Программное обеспечение состоит из блоков, выполняющих следующие основные операции:

- Вычисления минимального времени перелета между точками;
- Формирование записи задачи булева линейного программирования;
- Решение задачи булева линейного программирования с использованием функции `splexbilp`;
- Проверка наличия подциклов в полученном решении;
- Формирование дополнительного ограничения для исключения одного из подциклов;
- Визуализация решения задачи маршрутизации.

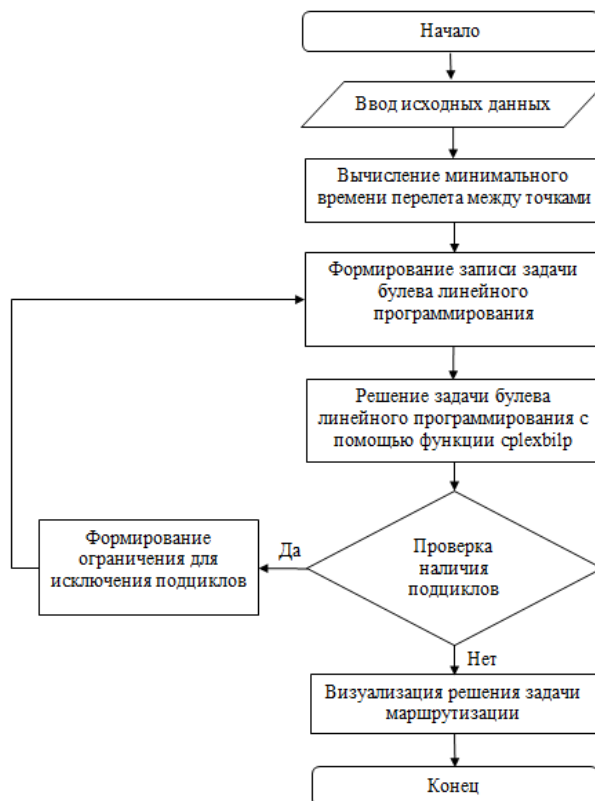


Рисунок 9. Блок-схема алгоритма работы программного обеспечения маршрутизации полета БПЛА с использованием процедуры последовательного запрета подциклов (разработан авторами)

Пример нахождения замкнутого маршрута наискорейшего облета заданных точек

Для иллюстрации работоспособности программно-алгоритмического обеспечения рассмотрим пример расчета замкнутого маршрута облета 165 точек, координаты которых случайным образом задаются в зоне полета размером 50x50 километров. Воздушная скорость БПЛА равна 16,67 м/с, а скорость юго-западного ветра равна 5,5 м/с. Расположение точек и найденный оптимальный маршрут приведены на рис. 10. Время расчета составило 132 с.

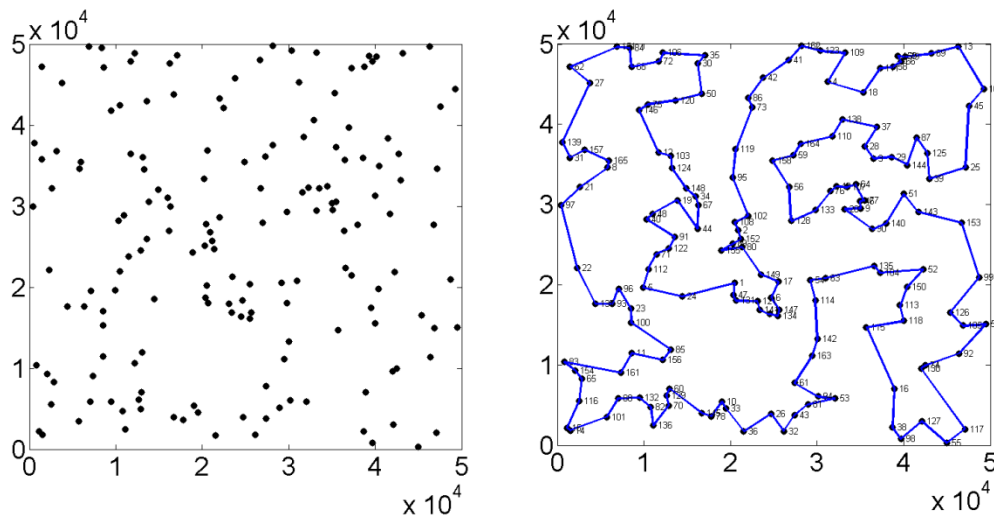


Рисунок 10. Оптимальный маршрут облета при $n = 165$ точек (разработан авторами)

Заключение

1. Проведен анализ нескольких разработанных авторами вариантов программно-алгоритмического обеспечения оптимальной маршрутизации полета легкого беспилотного летательного аппарата в поле постоянного ветра. На основе анализа полученных оценок быстродействия, требующегося объема оперативной памяти и предельных размеров устойчиво решаемых задач выявлено преимущество программно-алгоритмического обеспечения, в котором используются процедура последовательного исключения подциклов и функция `srlexb1lp` пакета CPLEX.

2. Оценка возможностей наиболее эффективного программно-алгоритмического обеспечения показала, что с его помощью устойчиво находятся точные решения задачи маршрутизации, содержащие в маршруте до 165 точек. При этом объем используемой оперативной памяти не превышает 1,2 Гб, а время вычислений меняется в зависимости от того как расположены точки в зоне полета, но в 85% случаев не превышает 300 с.

3. Особенностью разработанного прикладного программного обеспечения является его открытая архитектура, а также наличие быстродействующего программного ядра на основе функции `srlexb1lp` пакета CPLEX и специализированного периферийного программного обеспечения, обеспечивающего работу ядра. Указанные особенности позволяют наращивать возможности разработанного прикладного обеспечения как в части быстродействия за счет совершенствования ядра, так и в части решения новых видов задач маршрутизации за счет развития периферийного программного обеспечения.

ЛИТЕРАТУРА

1. Моисеев В.С., Абсалямов М.Н., Хакимуллина А.Р. Оптимизация траекторий летательного аппарата, выполняющего контроль наземной обстановки. Изв. Вузов. Авиационная техника, 2001. №1. с. 16-23.
2. Гимадеев Р.Г., Моисеев В.С. Выбор оптимального маршрута облета беспилотным летательным аппаратом заданной совокупности районов выполнения полетных заданий // Вестник КГТУ им. А.Н.Туполева. 2014. № 3. с. 208-212.
3. Nicola Ceccarelli, John J. Enright, Emilio Frazzoli, Steven J. Rasmussen and Corey J. Schumacher. Micro UAV Path Planning for Reconnaissance in Wind. Proceedings of the 2007 American Control Conference. New York City, USA, July 11-13, 2007.
4. Таргамадзе Р.Ч., Моисеев Д.В., Фам С.К. О рациональном выборе замкнутого маршрута полета легкого летательного аппарата с учетом прогноза ветра. Вестник ФГУП НПО им. С.А. Лавочкина. 2012. № 3. С. 76-83.
5. Моисеев Д.В., Чинь В.М. Маршрутизация полета легкого беспилотного летательного аппарата в поле постоянного ветра с учетом ограничения на продолжительность полета // Журнал «Мехатроника, Автоматизация, Управление», № 3, Том 17, 2016. с. 206-210.
6. Гэри В., Джонсон Д. Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982, с. 416.
7. Сигал И.Х., Иванова А.П. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы. – М.: Физматлит, 2007. – 304 с.
8. Лысенко И.В. О решении задач дискретной оптимизации в системе компьютерной математики MATLAB // Журнал «Радиоэлектронные и компьютерные системы, 2013, № 2 (61)». с. 86-93.
9. Рыкин О.Р. Линейное программирование в Матлабе. Универсальные линпрогооптимизаторы: производительность и табличный формат результата. Задания и задачи. – СПб.: Изд-во Политехн. ун-та, 2016. 208 с.
10. Моисеев Д.В., Чинь В.М., Моисеева С.Г., Фам С.К. Маршрутизация облета легким беспилотным летательным аппаратом неравноценных объектов в поле постоянного ветра с учетом ограничения на время полета по маршруту. Тезисы докладов 14-ой Международной конференции «Авиация и космонавтика», с 16 по 20 ноября 2015 г., РФ, г. Москва, Московский авиационный институт.
11. Алексеева Е.В. Построение математических моделей целочисленного линейного программирования. Примеры и задачи: Уч. Пособ. / Новосиб. гос. ун-т. Новосибирск, 2012. 131 с.
12. Козлов М.В., Костюк Ф.В., Сорокин С.В., Тюленев А.В. Решение задачи коммивояжера методом целочисленного линейного программирования с последовательным исключением подциклов: описание и алгоритмическая реализация // AdvancedScience. – 2012. №2. с. 124-141.
13. Козлов М.В., Костюк Ф.В., Сорокин С.В., Тюленев А.В. Решение задачи коммивояжера методом целочисленного линейного программирования с последовательным исключением подциклов: обоснование, тестовые испытания, применение // AdvancedScience. – 2012. №2. с. 142-159.

Moiseev Dmitriy Viktorovich

Moscow aviation institute (national research university), Russia, Moscow
E-mail: moiseev801@mail.ru

Trinh Van Minh

Moscow aviation institute (national research university), Russia, Moscow
E-mail: trinhminh6666@gmail.com

Computational aspects and application software for optimal flight routing of light unmanned aerial vehicle in constant wind field

Abstract. In the article compares several variants of the software algorithm to solve the flight routing problems of light unmanned aerial vehicle in the constant wind field. Based on this analysis, the most effective software algorithm is recommended. The procedures for constructing the fastest route are reduced to the problems of Boolean linear programming. The Software includes various standard functions of the MATLAB and CPLEX packages that are designed to solve the integer linear programming problems, as well as various special functions, which complement the standard function. Comparison of the variants was made on basis the received speed and memory usage estimates. The limiting sizes of the problems solved by the software are determined. The description of the principles of building a software package that solves the routing problem is given. The software algorithm has an open architecture, as well as presence of function cplexbilp of the CPLEX package and specialized peripheral functions. The efficiency of the software algorithm is demonstrated by the example of calculating the closed route of 165 points, which was performed in 132 s.

Keywords: flight routing; unmanned aerial vehicle; problem of Boolean linear programming; cumulative subtour elimination; bintprog; cplexbilp; software algorithm