

Генетический алгоритм решения задачи коммивояжера для планирования маршрута беспилотного летательного аппарата

Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ»

Описано решение задачи коммивояжера генетическим алгоритмом в сфере планирования маршрута беспилотного летательного аппарата в целях минимизации длины этого маршрута при уменьшении времени поиска. Задача коммивояжера решена для точек, расположенных в трёхмерном пространстве. Подробно описана работа непосредственно самого генетического алгоритма, а также уделено большое внимание описанию процесса скрещивания двух особей. В выводах указаны преимущества данного алгоритма перед работой метода полного перебора и «жадного» алгоритма.

Ключевые слова: беспилотный летательный аппарат, генетический алгоритм, задача коммивояжера, планировщик маршрута, программное обеспечение, популяция, мутация.

Введение

В настоящее время во всём мире люди активно начинают использовать беспилотные летательные аппараты (БПЛА). Они находят своё применение в различных областях, например, в сельском хозяйстве, в военном деле, занимаются задачами МЧС и многое другое. Однако в сфере использования БПЛА существует множество задач различного рода, которые требуют своего решения. Одна из таких задач связана с планированием маршрута.

Планировщик маршрута является ключевым элементом автономного управления БПЛА [1]. Он позволяет БПЛА автономно рассчитывать наилучший маршрут между начальной и конечной точками полёта. В то время как представители коммерческих авиакомпаний летают по неизменяемому предписанному маршруту, беспилотным летательным аппаратам, летающим в зонах оперативного реагирования, приходится постоянно менять свои траектории полёта, зависящие от текущей местности и погодных условий, прямо во время полета. В прошлом лучший маршрут ассоциировался с кратчайшим путём, который вычислялся детерминированными алгоритмами поиска. Однако с тех пор понятие лучшего маршрута изменилось, и теперь оно связано с минимизацией длины пути, средней высоты, расхода топлива, воздействия радара и т.д. Это всего лишь несколько примеров факторов, которые следует учитывать и которые ясно показывают, что сложность проблематики возросла. Для преодоления этой сложности исследователи долго переходили от детерминированных алгоритмов к недетерминированным. Например, авторы [2] предложили использовать вибрационный генетический алгоритм (ГА) для улучшения исследования и лучшего избегания локальных минимумов при поиске оптимального пути. Авторы [3] также использовали ГА, но с добавлением концепции искусственной иммунной системы, для поддержания лучшего разнообразия популяции на протяжении всего процесса эволюции.

Эти работы подтверждают актуальность рассмотренной тематики по планированию маршрута БПЛА с использованием ГА и моделирование полета с

учетом характеристик БПЛА. Следовательно, целью работы будет минимизация длины маршрута БПЛА при уменьшении времени поиска такого маршрута.

Описание генетического алгоритма решения задачи коммивояжера

Рассмотрим способ алгоритмического решения задачи коммивояжера для БПЛА с использованием генетического алгоритма. Маршрут будет состояться в системе трёх координат. Затем моделируется облёт БПЛА опорных точек заданного маршрута, в ходе которого определяется, насколько увеличивается суммарное расстояние, пройденное БПЛА, по сравнению с изначально запланированным.

Итак, в качестве входных данных имеется массив опорных точек, через которые должен пролететь БПЛА, а также указана точка, из которой он будет запущен. Каждая точка определена в пространстве тремя координатами – x , y и z .

К входным данным также относятся параметры алгоритма, такие, как размер популяции и процент новых особей. Под особью понимается любой маршрут коммивояжера. Он представляет собой массив последовательности опорных точек, причём самая первая и самая последняя точка – это всегда будет точка вылета БПЛА. Таким образом, особь имеет размер, равный количеству опорных точек + 1. Под популяцией понимается совокупность всех особей на каждой итерации алгоритма. Таким образом, количество особей всегда будет равно размеру популяции. Популяция представлена в виде массива.

Алгоритм начинается с того, что создаётся квадратная матрица расстояний, представленная двумерным массивом размера n на n , где n – количество опорных точек в маршруте. В пересечения i -й строки и j -го столбца данного массива записываются вычисленные по координатам расстояния между i -й и j -й точками. Далее формируется первая популяция, состоящая из случайно сгенерированных особей. Далее вычисляется длина пути каждой особи. Для этого суммируются расстояния между опорными точками, содержащимися в особи. Расстояния берутся из матрицы расстояний. Теперь, когда для каждой сгенерированной особи известна длина пути, массив популяции сортируется от особи с самой меньшей длиной до особи с самой большей. Таким образом, самая первая особь в массиве популяции будет иметь самый короткий путь. Однако это будет не самый оптимальный маршрут, а всего лишь лучший из первых, случайно сгенерированных.

Для поиска лучшего маршрута запускается цикл, который будет работать до тех пор, пока пользователь его не остановит. В этом цикле будет генерироваться новая популяция на основе предшествующей ей, затем опять будет проводиться сортировка, и т.д.

Для генерации новой популяции в цикле проводится скрещивание двух рядом стоящих особей, взятых из популяции, сформированной на предыдущей итерации, затем следующих двух, и так до тех пор, пока количество новых особей остаётся меньше заданного во входных параметрах процента новых особей. При скрещивании двух особей получаются также две особи. Все новые особи записываются в конец нового массива популяции, который является копией текущего. Таким образом, особи с самой большей длиной пути заменяются новыми особями.

Алгоритм, по которому скрещиваются особи, называется «частично отображаемое скрещивание». Суть этого алгоритма заключается в следующем:

пусть даны две родительские хромосомы $ch_{1,1} = \{1,2,3,4,5,6,7,8,9,10,1\}$ и $ch_{1,2} = \{1,4,6,2,9,8,7,3,10,5,1\}$. Случайным образом находим две точки разрыва: $p_1 = 4$ и $p_2 = 8$ ($1 \leq p_1 \leq p_2 \leq n$). Считаем, что разрыв произошел следующим образом: $ch_{1,1} = \{1,2,3,4 | 5,6,7,8 | 9,10,1\}$ и $ch_{1,2} = \{1,4,6,2 | 9,8,7,3 | 10,5,1\}$. Напомним, что позиции 1 и 11 в операции не участвуют, как будто реальная хромосома расположена между этими позициями. При формировании потомков $ch_{2,1} = \{1, x, x, x | x, x, x, x | x, x, 1\}$ и $ch_{2,2} = \{1, x, x, x | x, x, x, x | x, x, 1\}$ вначале проводим обмен частей, находящихся между точками разрыва: $ch_{2,1} = \{1, x, x, x | 9,8,7,3 | x, x, 1\}$ и $ch_{2,2} = \{1, x, x, x | 5,6,7,8 | x, x, 1\}$ (т.е. для потомка $ch_{2,1}$ берем среднюю часть из родителя $ch_{1,2}$, а для потомка $ch_{2,2}$ берем среднюю часть из родителя $ch_{1,1}$). Далее расставляем оставшиеся позиции от соответствующих хромосом (в $ch_{2,1}$ из $ch_{1,1}$, а для $ch_{2,2}$ из $ch_{1,2}$) слева направо до возникновения конфликта (номера вершин повторяются в уже сформированной части хромосомы). Если произошел конфликт, то записываем не конфликтующий номер, а номер из соседней хромосомы в той позиции, в которой был найден конфликт в текущей хромосоме. Так продолжается до полного разрешения конфликтов. Получаем: $ch_{2,1} = \{1,2,6,4 | 9,8,7,3 | 5,10,1\}$ и $ch_{2,2} = \{1,4,3,2 | 5,6,7,8 | 10,9,1\}$. Однако если проводить скрещивание двух идентичных особей, то окажется, что их потомки окажутся идентичными своим родителям. В итоге в популяции будет уже четыре идентичные особи, а так как следующая популяция формируется на основании предыдущей, то рано или поздно вся новая популяция будет состоять из идентичных особей. Чтобы этого избежать, после создания двух новых особей проводится проверка на их идентичность друг другу, и если оказывается, что они идентичны, то выполняется мутация каждой особи. Мутация представляет собой перестановку двух случайно выбранных позиций в каждой особи.

Когда пользователь останавливает работу алгоритма, то уже имеется определённый маршрут облёта всех опорных точек, а также известна длина пути данного маршрута.

Выводы

В работе было рассмотрено применение ГА решения задачи коммивояжера для планирования маршрута БПЛА в целях минимизации длины планируемого маршрута при уменьшении времени поиска такого маршрута. Если сравнивать с другими алгоритмами, то в целом эффективность доказана. Было разработано программное обеспечение, которое позволяет оценить достоинства алгоритма по сравнению с работой метода полного перебора и «жадного» алгоритма. На компьютере с четырёхядерным процессором 2,67 ГГц оптимальный маршрут для 20 опорных точек методом полного перебора был найден за 15 минут, ГА же справлялся с этой задачей в среднем за 3 секунды. «Жадный» алгоритм с этой задачей справляется даже быстрее, чем ГА, однако при определённом расположении опорных точек случается так, что найденный маршрут далеко не самый оптимальный. Поэтому для решения данной задачи ГА является наиболее подходящим.

Программное обеспечение, использующее предложенный алгоритм, может использоваться операторами (внешними пилотами) БПЛА в целях планирования маршрута БПЛА. Однако если запустить реальный БПЛА по данному маршруту, то

окажется, что расстояние, которое он пролетит, будет больше того расстояния, которое было посчитано алгоритмом. Это связано с тем, что БПЛА самолётного типа не летает по ломаной траектории, а разворачивается плавно. Для того, чтобы сменить курс на новую опорную точку, ему необходимо повернуться в соответствии с допустимым радиусом разворота. Программно смоделировав полёт БПЛА по предложенным маршрутам, можно сделать вывод о том, что реально длина пути может увеличиваться до 60% в зависимости от радиуса разворота БПЛА. Решение данной проблемы следует выполнять с учетом параметров БПЛА, что и будет выполнено в дальнейших исследованиях.

Список литературы

1. Chen H., A survey of autonomous control for UAV / H. Chen, X. M. Wang, Y. Li // in Proc. 2009 Int. Conf. Artif. Intell. Comput. Intell. – 2009. – Vol.2. – P. 267–271.
2. Volkan P. Y., A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV / P. Y. Volkan ; Aerosp. Sci. Technol. – Jan. 2012. – Vol. 16. – № 1. – P. 47–55.
3. Cheng Z., Path planning based on immune genetic algorithm for UAV / Z. Cheng, Y. Sun, Y. Liu // in Proc. Int. Conf. Electric Inform. Control Eng. – 2011. – P. 590–593.

Поступила в редакцию 17.03.2016

Генетичний алгоритм розв'язання задачі комівояжера для планування маршруту безпілотного літального апарату

Описано вирішення задачі комівояжера генетичним алгоритмом у сфері планування маршруту безпілотного літального апарату з метою мінімізації довжини цього маршруту при зменшенні часу пошуку. Задачу комівояжера вирішено для точок, розташованих у тривимірному просторі. Детально описано роботу безпосередньо самого генетичного алгоритму, а також приділено велику увагу опису процесу схрещування двох особин. У висновках вказано переваги даного алгоритму перед роботою методу повного перебору і «жадібного» алгоритму.

Ключові слова: безпілотний літальний апарат, генетичний алгоритм, завдання комівояжера, планувальник маршруту, програмне забезпечення, популяція, мутація.

Genetic Algorithm for Solving the Traveling Salesman Problem of Route Planning UAV

Described solution traveling salesman problem by genetic algorithm in the field of route planning UAV in order to minimize the length of the route with a decrease in search time. Traveling salesman problem is solved for points located in three-dimensional space. The work of genetic algorithm is described in detail itself, and also paid great attention to the description of the process of crossing two individuals. The findings indicated the advantages of this algorithm to work brute force and "greedy" algorithm.

Keywords: drone, genetic algorithm, traveling salesman problem, a route planner, software, population, mutation.