DOI 10.24411/2076-1503-2020-11042
629.7.05

ШАМУНОВА Вероника Германовна,
аспирант кафедры управления качеством, математическое моделирование, численные методы и комплексы программ РФ Национального исследовательского Томского государственного университета,
e-mail: evtushenko_nika@mail.ru

ОПТИМАЛЬНЫЙ АЛГОРИТМ ПОИСКА ПУТИ В ДИНАМИЧЕСКОЙ СРЕДЕ ДЛЯ ЗАДАЧ ПОЗИЦИОНИРОВАНИЯ ПОДВИЖНОГО ОБЪЕКТА

Аннотация. *На сегодняшний день задача навигации беспилотных летательных аппаратов в сложных динамических средах весьма актуальна. Динамическая среда характеризуется наличием препятствий различного характера на пути летательного аппарата, которые необходимо преодолевать быстро, выбирая верную траекторию движения и не отклоняясь от заданного маршрута. Для обхода препятствий существует множество алгоритмов, применяемых для решения задачи обхода препятствий в различных условиях. В статье приведены результаты исследований выбора оптимального алгоритма поиска пути для условий динамической среды из наиболее простых и популярных.*

Ключевые слова: *БПЛА, системы навигации БПЛА, алгоритмы поиска пути, обход препятствий.*

SHAMUNOVA Veronika Germanovna,
graduate student department of quality management, mathematical modeling, numerical methods and software complexes Russia of National research Tomsk state university

OPTIMAL PATH SEARCH ALGORITHM IN A DYNAMIC ENVIRONMENT FOR PROBLEMS OF POSITIONING A MOVING OBJECT

Annotation. *Today, the task of navigating unmanned aerial vehicles in complex dynamic environments is very urgent. A dynamic environment is characterized by the presence of various obstacles on the path of the aircraft, which must be overcome quickly, choosing the correct trajectory and not deviating from the specified route. For obstacle avoidance, there are many algorithms used to solve the obstacle avoidance problem in various conditions. The article presents the results of research on the choice of the optimal path finding algorithm for the conditions of a dynamic environment.*

Key words: *UAV, UAV navigation systems, vision systems, obstacle avoidance.*

Задача поиска пути для БПЛА в условиях динамической среды на сегодняшний день является одной из самых актуальных и приоритетных ввиду того, что современные БПЛА находят применение все в больших и больших областях. Применение БПЛА в различных областях требует от устройств максимальной адаптивности к различным условиям, в которых необходимо выполнять полет. Выполняя полет по заданной траектории, устройство в динамической среде может столкнуться как со статическими, так и с динамическими препятствиями, которые необходимо обойти, не отклоняясь от заданного маршрута.

Ныне наиболее эффективный способ идентифицировать препятствие - использование систем технического зрения, строящихся на базе группы камер, предоставляющих системе снимки окружающего пространства, на базе которых система выполняет построение карт глубины. Карты глубины позволяют при помощи соответствующих алгоритмов идентифицировать на пути устройства препятствия. Количество камер, составляющих группу для системы технического зрения, выбирается так, чтобы при построении карт глубины иметь представление о препятствиях в трехмерной среде.

Идентифицировав на текущий момент времени препятствия, располагающиеся на пути

БПЛА, следующим этапом выполняется поиск оптимального пути для обхода обнаруженного препятствия или препятствий, не отклоняясь от заданного маршрута. Для решения обозначенной задачи применяются специализированные алгоритмы, позволяющие с наибольшей точностью определить кратчайший путь, позволяющий избежать столкновения [1].

На сегодняшний день существует ряд алгоритмов, применяемых для решения задачи поиска оптимального пути для различного рода устройств. Однако достаточно проблематично выбрать действительно эффективный и простой алгоритм для БПЛА, выполняющих полеты в сложных динамических средах. Рассмотрим наиболее популярные, простые и эффективные алгоритмы поиска оптимального пути на основе клеточной декомпозиции и интеллектуальные алгоритмы и оценим возможность и эффективность их применения для БПЛА в динамических средах [2; 3].

Алгоритм Дейкстры

Сегодня данный алгоритм считается наиболее популярным для решения задачи поиска кратчайшего пути в графе. В основе алгоритма лежит клеточная декомпозиция.

Алгоритм позволяет выполнить поиск всех кратчайших путей взвешенного графа с неотрицательными весами, двигаясь от исходной вершины до каждой вершины графа, последовательно увеличивая множество вершин, для которых известен кратчайший путь. Как только очередной вершиной оказывается целевая точка траектории и для нее найден кратчайший путь, работа алгоритма фактически считается завершенной. В результате – полученный кратчайший путь от исходной точки, которая является текущим местонахождением устройства, до целевой точки, которая может быть как заданной точкой полетного маршрута, так и некая опорная точка с некоторым процентом отклонения от заданного маршрута, ввиду наличия препятствий на заданном пути.

Вычислительная сложность алгоритма Дейкстры оценивается как квадратичная $O(1.5N^2)$, где n – количество вершин входного графа. Данный алгоритм может быть применен для решения отдельных задач поиска кратчайшего пути для БПЛА в динамической среде, однако существует ряд недостатков, которые в значительной мере влияют на эффективность использования данного алгоритма в описанных условиях [6].

1. Применим только в работе с графами, имеющими неотрицательные длины дуг.

2. Отыскивает кратчайшие пути и их вершины только для одной вершины.

3. Требует многократного повтора для всех пар вершин.

Перечисленные выше недостатки значительно снижают эффективность применения данного алгоритма для поставленной задачи ввиду того, что возрастает программное время работы данного алгоритма, а для ряда условий алгоритм оказывается несостоятельным.

В рамках исследования проведен опытный эксперимент путем простейшей реализации данного алгоритма на языке программирования C++ на ПЭВМ с процессором Intel® Core™ i5-6400 CPU, 2.70GHz и оперативной памятью 16 Гб. При подаче на вход алгоритма графа с 10 вершинами и 11 ребрами время, за которое алгоритм выполнил поиск оптимального пути, составило 0,03225 секунд. Алгоритму потребовалась 941 операция.

Алгоритм A* (A-star)

Алгоритм поиска оптимального пути A* довольно прост и находит применение во многих областях. В сравнении с алгоритмом Дейкстры требует меньше программного времени за счет применения эвристики.

Алгоритм использует эвристическую функцию:Алгоритм использует эвристическую функцию:

$$f(n) = g(n) + h(n),$$

где $f(n)$ – оценка для узла n;

$g(n)$ – наименьшая стоимость достижения узла n из начальной точки;

$h(n)$ – эвристическое приближение стоимости пути к цели от узла n.

Основными преимуществами применения данного алгоритма для задачи поиска пути БПЛА является то, что алгоритм может быть охарактеризован как

оптимальный и полный. Алгоритм A* позволяет выполнить поиск оптимального пути до единственной целевой точки.

Алгоритм продолжает свою работу до тех пор, пока значение функции $f(n)$ целевой вершины не окажется меньшим, чем любое значение в очереди, либо пока всё дерево не будет просмотрено. Из множества решений выбирается решение с наименьшей стоимостью [5].

Вычислительная сложность алгоритма A* оценивается как логарифмическая ***O((n+m)lnn)*** (за *n* принимается количество вершин, за *m* – количество ребер исходного графа).

Алгоритм применим для решения задачи поиска оптимального пути для БПЛА в описанных условиях, однако имеет ряд недостатков, которые снижают эффективность данного алгоритма в рамках поставленной задачи.

1. Ввиду необходимости дискретизации трехмерной области необходимы значительные ресурсы памяти для хранения графа.

2. Снижает скорость БПЛА, т.к. найденные пути зачастую являются ступенчатыми и в значительной мере приводят к необходимости снижения скорости устройства.

В рамках исследования проведен опытный эксперимент путем реализации данного алгоритма на языке программирования C++ на ПЭВМс процессором Intel® Core™i5-6400 CPU, 2.70GHzи оперативной памятью 16 Гб. При подаче на вход алгоритма графа с 10 вершинами и 11 ребрами время, за которое алгоритм выполнил поиск оптимального пути, составило 0,00041 секунд. Алгоритму потребовалось 158 операций.

Алгоритм муравьиной стаи

Алгоритм муравьиной стаи – относительно простой в реализации интеллектуальный алгоритм. В основе данного алгоритма лежит поведение муравьиной стаи в задачах поиска пищи. Найдя пищу и возвращаясь в муравейник, муравей помечает феромоном путь для других членов стаи. Феромон со временем испаряется и членам стаи все сложнее его почувствовать. Чем длиннее путь до пищи, тем быстрее испаряется феромон. Соответственно, чем короче путь, тем сильнее на нем будет запах феромона и тем привлекательнее данный путь будет для членов муравьиной стаи.

Алгоритм муравьиной стаи позволяет выполнить поиск оптимального пути

за относительно приемлемое время. Для данного алгоритма задача преобразуется в задачу поиска в графе. Ребро графа характеризуется объемом феромона τ_i^j (i – номер вершины, из которой берет начало ребро, j– номер вершины, в которую ребро идет). В алгоритме участвует mмуравьев, выполняющих задачу поиска пути. Для найденных путей выполняется расчет объемов феромона, приходящихся на данный путь.

Основная формула алгоритма выглядит следующим образом:

$$\rho_{i,j} = \tau_{i,j}^{\alpha} \mu_{i,j}^{\beta}$$

где $\tau_{i,j}$ – объем феромона для ребра между вершинами i и j, α – параметр контроля влияния $\tau_{i,j}$, $\mu_{i,j}$–привлекательность ребра между вершинами i и j, β – параметр контроля влияния $\mu_{i,j}$.

Для каждого ребра в очередную итерацию алгоритма рассчитывается объем феромона с учетом предыдущего полученного значения и скорости испарения феромона, $\tau_{i,j}$ после чего определяется общее количество феромона [4].

Вычислительная сложность алгоритма муравьиной стаи оценивается как квадратичная $\boldsymbol{O(t \cdot m \cdot n^2)}$, где t–количество итераций алгоритма, m–количество муравьев, n – количество вершин.

Скорость работы алгоритма муравьиной стаи напрямую зависит от количества итераций и количества вершин. Увеличение количества вершин, требуемых обхода, приводит к квадратичному увеличению времени работы алгоритма.

Также для реализации подобного алгоритма относительно алгоритмов Дейкстры и A*могут потребоваться большие вычислительные мощности, которые в условиях размещения на БПЛА не могут быть предоставлены.

В рамках исследования проведен опытный эксперимент путем реализации данного алгоритма на языке программирования C++ на ПЭВМс процессором Intel® Core™i5-6400 CPU, 2.70GHzи оперативной памятью 16 Гб. В опыте была реализована и использована только одна стая муравьев, содержащая 25 членов. Количество вершин, которые необходимо пройти равно 10. Количество итераций алгоритма 100. С указанными параметрами время, за которое алгоритм в указанных условиях находит оптимальный путь, составляет 0,44597 секунд.

Заключение

Анализ вычислительной сложности наиболее простых и популярных на сегодняшний день алгоритмов поиска оптимального пути, относящихся к алгоритмам на основе клеточной декомпозиции и интеллектуальным алгоритмам,показал, что наилучший результат по времени показывает алгоритм A* (A-star), который является эвристическим, всегда выполняющим поиск пути, если таковой существует. Алгоритм Дейкстры показывает результат по времени хуже и в значительной мере зависит от числа вершин графа, который принимает на вход для расчета. Согласно оценке сложности, при увеличении числа вершин графа время работы алгоритма увеличится квадратично. Аналогичная ситуация и с интеллектуальным алгоритмом муравьиной стаи, время работы которого квадратично зависит от числа вершин графа, а также имеет зависимость от количества итераций и числа муравьев, используемых для поиска пути. Опытный эксперимент проводился в простейших условиях: с небольшим количеством вершин. Увеличение числа вершин, соответственно, приведет и к увеличению времени, которое требуется каждому из рассматриваемых алгоритмов, для поиска оптимального пути.

Список литературы:

[1] Lee T-L., Wu C-J. Fuzzy motion planning of mobile robots in unknown environments // Journal of Intelligent and Robotic Systems. – 2003. - № 37 (2). – P. 177 - 191.

[2] Alferov G.V., Malafeyev O.A. The robot control strategy in a domain with dynamical obstacles // Lecture Notes in Computer Science. – 1996. – № 1093. – P. 211 - 217.

[3] Герасимов В.Н. Система управления движением мобильного робота в среде с динамическими препятствиями // Научно-технические ведомости СПбГПУ. - 2013. – № 5. - С. 94 - 102.

[4] Mohamad M.M., Dunnigan M.W., Taylor N.K. Ant colony robot motion planning // EUROCON 2005: Intern. conf. on computer as a tool (Belgrade, Serbia, November 21 - 24, 2005): Proc. N.Y.: IEEE. – 2005. – № 1. – P. 213-216.

[5] Кристофидес Н. Теория графов: алгоритмический подход. – 2-е изд., испр. – М.: Мир, 1978. – С. 430.

[6] Minieka E. Optimization Algoritms for Networks and Graphs. Marcel Dekker, Inc., New York and Basel. – 1978. – P. 323.

Spisok literatury:

[1] Lee T-L., Wu C-J. Fuzzy motion planning of mobile robots in unknown environments // Journal of Intelligent and Robotic Systems. – 2003. - № 37 (2). – P. 177 - 191.

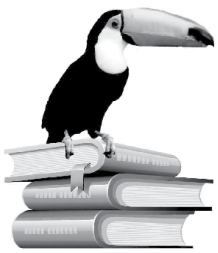
[2] Alferov G.V., Malafeyev O.A. The robot control strategy in a domain with dynamical obstacles // Lecture Notes in Computer Science. – 1996. – № 1093. – P. 211 - 217.

[3] Gerasimov V.N. Sistema upravleniya dvizheniem mobil'nogo robota v srede s dinamicheskimi prepyatstviyami // Nauchno-tekhnicheskie vedomosti SPbGPU. - 2013. – № 5. - S. 94 - 102.

[4] Mohamad M.M., Dunnigan M.W., Taylor N.K. Ant colony robot motion planning // EUROCON 2005: Intern. conf. on computer as a tool (Belgrade, Serbia, November 21 - 24, 2005): Proc. N.Y.: IEEE. – 2005. – № 1. – P. 213-216.

[5] Kristofides N. Teoriya grafov: algoritmicheskij podhod. – 2-e izd., ispr. – M.: Mir, 1978. – C. 430.

[6] Minieka E. Optimization Algoritms for Networks and Graphs. Marcel Dekker, Inc., New York and Basel. – 1978. – P. 323.



ЮРКОМПАНИ

www.law-books.ru

Юридическое издательство «**ЮРКОМПАНИ**» издает научные журналы:

- Научно-правовой журнал «Образование и право», рекомендованный ВАК Министерства науки и высшего образования России (специальности 12.00.01, 12.00.02), выходит 1 раз в месяц.
- Научно-правовой журнал «Право и жизнь», рецензируемый (РИНЦ, E-Library), выходит 1 раз в 3 месяца.