

АДАПТИВНОЕ ПЛАНИРОВАНИЕ МАРШРУТОВ ГРУППЫ БПЛА В СЛОЖНОЙ СРЕДЕ

Акимов А.А., к.ф.-м.н.

МИРЭА - Российский технологический университет,
г. Москва, Россия,

Гнатенко Ю.А., к.ф.м.-н.

Стерлитамакский филиал Уфимского университета науки и технологий,
г. Стерлитамак, Россия

Аннотация. Группы БПЛА эффективнее одиночных, но планирование их маршрутов в насыщенной препятствиями среде остаётся сложной задачей. Предлагается алгоритм на базе непрерывной муравьиной оптимизации с управлением через обучение с подкреплением, который адаптивно выбирает одну из нескольких стратегий построения пути и объединяет преимущества обоих методов. На примере показана построение согласованных траекторий с предотвращением столкновений. Высокая адаптивность и надёжность делают подход перспективным для задач планирования маршрутов нескольких БПЛА.

Ключевые слова: БПЛА, обучение с подкреплением, алгоритм муравьиной колонии, задача оптимизации

Введение. БПЛА всё шире применяются в различных отраслях народного хозяйства от экомониторинга и сельского хозяйства до логистики и поисково-спасательных операций [4], [1]. Группы дронов имеют преимущество перед одиночными аппаратами в грузоподъёмности и зоне охвата и позволяют выполнять несколько задач одновременно. Ключевой момент в координации движений— планирование безопасных траекторий в среде с множеством препятствий и предотвращением взаимных столкновений. Учитываются ограничения на длину пути, радиус поворота,

высоту полёта и обход препятствий. Формально задача сводится к оптимизации траекторий нескольких агентов с ограничениями. Например, в качестве целевой функции минимизации, может выступать суммарная длина маршрутов всех M дронов:

$$J = \sum_{i=1}^M L_i ,$$

где L_i – длина пути i -го дрона. Длину отдельного маршрута можно вычислить по координатам последовательности его путевых точек $(x_{i,0}, y_{i,0}) = S_i, (x_{i,1}, y_{i,1}), \dots, (x_{i,K_i}, y_{i,K_i}) = G_i$ (старт S_i и финиш G_i):

$$L_i = \sum_{k=0}^{K_i-1} \sqrt{(x_{i,k+1} - x_{i,k})^2 + (y_{i,k+1} - y_{i,k})^2} .$$

Оптимизируя длину, необходимо соблюдать ограничения: ни один участок траектории не должен проходить через препятствия, а расстояние между любыми двумя дронами должно быть не менее допустимого $D_{\min}$. Например, для любой пары дронов i и j в любой момент времени должно выполняться неравенство:

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq D_{\min} .$$

Аналогично, если препятствия аппроксимировать цилиндрами (или кругами на плоскости) с радиусом R_k , то каждая путевая точка должна находиться вне препятствия k :

$$(x_{i,k} - X_k^{(obs)})^2 + (y_{i,k} - Y_k^{(obs)})^2 \geq R_k^2 ,$$

где $X_k^{(obs)}, Y_k^{(obs)}$ – координаты центра k -го препятствия.

За последние годы сложился ряд подходов к планированию траекторий дронов — от классической геометрии до эволюционных вычислений. К традиционным относятся сеточные методы, где пространство дискретизируется и путь ищется с использованием классических алгоритмов (Дейкстра, A*) Эти подходы просты и надёжны, но при высокой

дискретизации пространства становятся вычислительно затратными, особенно для роя агентов. Дорожные карты (в т.ч. диаграммы Вороного) строят граф допустимых ходов и хорошо держат дистанцию от препятствий, однако в сложных сценариях также страдают от комбинаторного роста числа узлов и ребер. Метод потенциальных полей вычислительно быстрый, но имеет склонность застревать в локальных минимумах.

Эти ограничения стимулировали переход к эволюционным и роевым методам, напоминающим биологическую эволюцию или поведение животных, генетическим алгоритмам, PSO, пчелиной колонии и др., которые естественно распараллеливаются при вычислениях и находят приемлемые маршруты даже в сложных ландшафтах. Среди них особенно выделяется алгоритм муравьиной колонии (ACO).

Основная часть. Алгоритм ACO [05], разработанный на основе поведения реальных муравьёв, накапливающих «феромоны» на удачных маршрутах, доказал свою эффективность для глобального поиска маршрутов, близких к оптимальным. Но классический алгоритм ACO ориентирован на дискретные задачи [02] и плохо переносится на непрерывные пространства, так как при высокой дискретизации резко возрастает вычислительная сложность этого алгоритма.

Для адаптации идей ACO к непрерывным областям был разработан алгоритм ACOR [03]. Здесь решение представлено набором непрерывных параметров (координатами маршрутных точек), а новые траектории генерируются на основе вероятностных моделей, построенных по лучшим найденным примерам, без явной дискретизации пространства.

Кроме традиционных алгоритмов хорошо зарекомендовали себя гибридные модификации, в частности гибридный алгоритм роевой оптимизации с обучением с подкреплением (RL). В этом случае алгоритм учится выбирать стратегию, подходящую текущей обстановке. В предложенной в статье схеме несколько альтернативных стратегий комбинируются и управляются

RL. Рассмотрим одну из стратегий — гибридный усреднённый конструктор с гауссовым шагом.

Пусть получены три маршрута-кандидата: $X^{(tour)}$, $X^{(rou)}$ и $X^{(rand)}$ выбранные соответственно турнирным, рулеточным и случайным методами. Каждый маршрут X можно представить в виде набора координат его маршрутных точек (вектор большой размерности). Гибридный конструктор вычисляет направляющий вектор μ как среднее значение этих трёх решений по координатам:

$$\mu = \frac{X^{(tour)} + X^{(rou)} + X^{(rand)}}{3}.$$

Вектор μ является усреднённой траекторией, комбинирующей информацию от нескольких различных решений.

Получив направляющий маршрут μ алгоритм генерирует на его основе новое решение при помощи небольшого случайного возмущения:

$$X_{\text{new}} = \mu + \delta, \delta \sim \mathcal{N}(0, \sigma^2),$$

где δ — случайный вектор той же размерности, что и μ , компоненты которого распределены по закону $\mathcal{N}(0, \sigma^2)$ (нормальный закон с нулевым средним и небольшим среднеквадратичным отклонением σ).

В алгоритме предусмотрены разные типы шага. Помимо малых гауссовских отклонений используется «прыжок Леви» — редкие крупные смещения, помогающие выбираться из локальных минимумов. Комбинация частых малых шагов и редких прыжков известна своей результативностью.

После генерации нового набора маршрутов оцениваются их качество и допустимость: траектории не должны пересекать препятствия, а минимальная дистанция между любыми двумя дронами по всему пути должна быть не меньше D_{min} .

Недопустимые маршруты не выбрасываются: для наиболее перспективных выполняется целевое исправление. Проблемные точки — те, что попали в препятствие или слишком сблизилась с соседями — сдвигаются к ближайшей разрешённой позиции (по осям до выхода из запретной зоны),

затем траектория заново проверяется и сглаживается, например сплайном. Так «сырые» решения превращаются в валидные.

Выбор стратегии делается через Q -обучение. Состояние описывает ход поиска (улучшение качества, доля допустимых решений и т.п.), действие — выбор одной из доступных стратегий, вознаграждение — улучшение маршрутов: сокращение длины лучшего пути и снижение средних значений по популяции.

Обучение происходит итерационно. В начале поиска Q -значения неопределенны, и алгоритм пробует разные стратегии (исследует пространство действий случайно). После каждой итерации происходит оценка полученного результата и обновление соответствующего Q -значения по формуле:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)],$$

где $Q(s, a)$ — ценность действий a в различных состояниях s , a — шаг обучения, γ — дисконтирующий коэффициент, r — полученное вознаграждение, s' — новое состояние, в которое перешёл алгоритм, а максимум берётся по всем действиям a' , доступным из состояния s' .

После этапа первичного поиска алгоритм начинает опираться на накопленный опыт и на каждом шаге выбирать стратегию с наибольшим Q -значением для текущего состояния. Обучение не прекращается — при смене обстановки предпочтения пересматриваются. Безопасность не страдает, какой бы способ генерации не был выбран, все траектории проходят проверку и корректировку, поэтому «удачная» по Q стратегия не приведёт к столкновениям, а Q -обучение лишь ускоряет поиск годных решений.

По трудоёмкости метод сопоставим с классическим муравьиным, так как основное время уходит на генерацию маршрутов и их оценку. Блок Q -обучения добавляет лишь небольшие операции выбора и обновления.

Короткий пример. Два дрона летят по плоскости с непроходимыми зонами. Для первого чаще выбираются «аккуратные» малые гауссовы шаги, для второго — временами крупный скачок, чтобы перепрыгнуть через

сложный участок. Новые пути проверяются; недопустимые точки сдвигаются к ближайшим разрешённым позициям и траектории сглаживаются. Награда присуждается за укорочение лучших маршрутов и общее улучшение по популяции; в результате алгоритм всё реже действует наугад, но изредка возвращается к альтернативным маршрутам, чтобы не упустить лучший маршрут. В конце оба пути безопасны и близки к кратчайшему маршруту с учётом рельефа.

Выводы. Планирование для группы дронов требует баланса между оптимальностью и жёсткими ограничениями. Классические методы либо вычислительно эффективны, но застревают в локальных экстремумах, либо плохо масштабируются. Предложенный в статье подход сочетает непрерывную муравьиную оптимизацию, обучение с подкреплением для выбора стратегии и модуль исправления траекторий. Это повышает адаптивность и устойчивость, что доказывают эксперименты, где средняя длина пути сократилась на 5–30% по сравнению с известными, а доля успешных прокладок маршрута не опускалась ниже ~43%, тогда как базовые методы в тех же сценах часто полностью «падали».

Литература

1. Горшков И. Ф., Акимов А. А. Динамическая маршрутизация дронов для доставки заказов // Научно-технический вестник Поволжья. – 2025. – № 6. – С. 202–205.
2. Мифтахов Э. Н., Акимов А. А., Гнатенко Ю. А. Методы роевой оптимизации частиц и локальных эвристик для решения мультиагентной задачи коммивояжёра // Научно-технический вестник информационных технологий, механики и оптики. – 2025. – Т. 25, № 5. – С. 856–865.
3. Niu B., Wang Y., Liu J., Yue G. Path planning for unmanned aerial vehicles in complex environment based on an improved continuous ant colony optimisation // Comput. Electr. Eng. – 2025. – Vol. 123. – Art. 110034.

4. Shakhathreh H. et al. Unmanned aerial vehicles (UAVs): a survey on civil applications and key research challenges // IEEE Access. – 2019. – Vol. 7. – P. 48572–48634.
5. Socha K., Dorigo M. Ant colony optimization for continuous domains // Eur. J. Oper. Res. – 2008. – Vol. 185, No. 3. – P. 1155–1173.