

УДК 519.853.6, 517.958

DOI:10.25729/ESI.2026.42.2.007

Построение быстрых маршрутов без столкновений в анизотропной среде с динамическими препятствиями

Казаков Александр Леонидович, Лемперт Анна Ананьевна, Чан Туан Вьет

Иркутский национальный исследовательский технический университет,

Россия, Иркутск, *kazakov@icc.ru*

Аннотация. В работе рассматривается задача построения маршрутов, минимальных по времени преодоления, для подвижных объектов, в качестве которых могут выступать, например, автономные необитаемые подводные аппараты в водной среде. Основные трудности здесь связаны с наличием пространственно-неоднородных течений, а также статических и движущихся препятствий. Предложен подход, основанный на оптико-геометрической аналогии и принципах Ферма–Гюйгенса. Задача формулируется в терминах обобщенного уравнения эйконала, которое описывает распространение фронта волны в среде с заданным векторным полем скоростей. Такой подход позволяет свести исходную вариационную задачу к дифференциальному уравнению в частных производных и избежать непосредственного перебора возможных траекторий. Поле времени прибытия строится, как решение краевой задачи, а оптимальная траектория восстанавливается движением в направлении антиградиента этого поля. Для численной реализации разработаны два алгоритма. Первый, основанный на методе быстрого марша, вычисляет поле минимального времени от стартовой точки до всех узлов расчетной сетки. Второй алгоритм по этому полю восстанавливает траекторию движения и позволяет получить ее фрагмент, соответствующий заданному временному интервалу. Проведена серия вычислительных экспериментов, включающих четыре сценария разной сложности: прямолинейное течение, два противоположных течения, течение с вихревой зоной и статическими препятствиями, а также сценарий с движущимися препятствиями и вихрем. Во всех случаях алгоритм успешно построил траектории, минимизирующие время движения и обеспечивающие безопасный обход препятствий. В сценарии с динамическими препятствиями продемонстрирована возможность оперативного перестроения маршрута. Проведено сравнение предложенного подхода с методом роевого интеллекта SSA (Salp Swarm Algorithm). Результаты показали, что разработанный алгоритм позволяет находить пути с меньшим временем прохождения, даже если их геометрическая длина превышает длину маршрута, найденного SSA. При этом время расчета траектории достаточно мало, что позволяет осуществлять планирование движения в режиме реального времени.

Ключевые слова: задача маршрутизации, динамическая среда, оптико-геометрический подход, уравнение эйконала, алгоритм быстрого марша

Цитирование: Казаков А.Л. Построение быстрых маршрутов без столкновений в анизотропной среде с динамическими препятствиями / А.Л. Казаков, А.А. Лемперт, Т.В. Чан // Информационные и математические технологии в науке и управлении, 2026. – № 2(42). – С. 87-101. – DOI:10.25729/ESI.2026.42.2.007.

Введение. Развитие автономных необитаемых подводных аппаратов (АНПА) открыло новые возможности для мониторинга морской среды, геологических изысканий, инспекции и обслуживания подводной инфраструктуры [1, 2]. Время выполнения миссии является одним из важнейших факторов, определяющих эффективность и энергопотребление АНПА, поэтому построение для них минимальных по времени (быстрых) маршрутов является актуальной проблемой. Отметим, что похожие задачи возникают в области инфраструктурной логистики при размещении объектов коллективного пользования [3, 4].

Кроме того, специфика подводной среды существенно усложняет решение задачи. Во-первых, необходимо учитывать наличие сложных и изменчивых океанических течений [5, 6], которые могут как способствовать движению, так и препятствовать ему, делая кратчайший по расстоянию путь далеко не самым быстрым [7]. Во-вторых, аппарат должен безопасно перемещаться в среде, где присутствуют как статические барьеры (особенности рельефа и запретные для движения зоны), так и динамические объекты [8–10].

Для решения задач навигации и планирования движения в таких условиях существует ряд подходов. Классические методы оптимизации, например, дифференциальная эволюция (DE), демонстрируют эффективность в статической среде [11]. Методы на основе искусственных потенциальных полей (APF) позволяют организовать уклонение от статических препятствий [8]. Планировщики на основе графов, такие как алгоритм D* и его модификации [12, 13], успешно применяются для перепланирования маршрутов, а RRT* [14] – для поиска пути в сложных пространствах. В последнее время активно развиваются методы глубокого обучения с подкреплением (DRL), способные вырабатывать сложные стратегии управления путем взаимодействия со средой [15].

Однако в условиях среды, сочетающей стационарные течения и движущиеся объекты, применение этих методов в чистом виде проблематично. Большинство из них либо не учитывают влияния течений на временные затраты, либо требуют слишком много вычислительных ресурсов для оперативного перепланирования маршрута.

В настоящей работе предлагается подход, основанный на оптико-геометрической аналогии [16, 17], которая позволяет свести задачу поиска кратчайшего по времени пути к решению обобщенного уравнения эйконала [18, 19]. Это дает возможность построить глобальное поле времени прибытия от стартовой точки; любая траектория, следующая в направлении, противоположном градиенту этого поля, гарантированно является оптимальной по времени при текущих условиях. Для численного решения уравнения эйконала применяется модификация метода быстрого марша, адаптированная для учета как стационарных течений, так и динамических препятствий. Предложенный алгоритм позволяет не только находить оптимальный маршрут, но и перестраивать его при изменении обстановки за счет локальной актуализации поля времени.

Статья организована следующим образом. В разделе 1 дана математическая постановка задачи. Раздел 2 описывает переход к обобщенному уравнению эйконала и постановку соответствующей краевой задачи. В разделе 3 подробно описаны разработанные вычислительные алгоритмы. Раздел 4 содержит результаты вычислительных экспериментов, иллюстрирующих поведение алгоритма в различных условиях: при наличии течений, вихревых зон и движущихся препятствий, а также сравнение предложенного подхода с методом роевого интеллекта SSA. Заключение подводит итог и намечает направления дальнейших исследований.

1. Постановка задачи. В данном разделе строятся модели среды, препятствий и объекта, а также формулируется задача минимизации времени его движения между двумя точками.

1.1. Модель среды и препятствий. Рассмотрим область $\Omega \subseteq R^2$, в которой движется объект. Выбор двумерной постановки соответствует планированию маршрута на фиксированной глубине, либо планированию маршрута на карте. Переход к трехмерному случаю возможен путем добавления третьей пространственной координаты и не меняет сути предложенного подхода.

Пусть задано стационарное векторное поле скоростей внешней среды $\vec{w} = (w_1(x, y), w_2(x, y))$, непрерывное по пространственным координатам и ограниченное $|\vec{w}| < v$, где v – собственная скорость объекта. Это условие гарантирует возможность движения против течения. Среда считается идеальной жидкостью, поскольку вязкостные эффекты пренебрежимо малы по сравнению с переносным движением на масштабах рассматриваемой задачи. Наличие течения делает среду анизотропной для движущегося объекта, поскольку время перемещения зависит от направления.

Множество запретных для движения зон в момент времени t зададим с помощью индикаторной функции $h: R^+ \times \Omega \rightarrow \{0,1\}$. Если точка (x, y) в момент времени t непроходима, то $h(t, x, y) = 0$, и $h(t, x, y) = 1$ в противном случае. Тогда множество барьеров определяется, как $H(t) = \{(x, y) \in \Omega : h(t, x, y) = 0\}$. Траектория движения не должна пересекать множество $H(t)$ ни в какой момент времени.

1.2. Модель объекта и критерий оптимальности. Объект движется в области Ω из точки A в точку B , обладая собственной постоянной по модулю скоростью $v > 0$ относительно среды. Предполагается, что объект является материальной точкой и имеет изотропную тягу – его собственная скорость не зависит от направления движения, а инерционностью разгона и торможения можно пренебречь по сравнению с характерным временем движения. Такое допущение оправдано для небольших аппаратов с маневровыми двигателями, работающими в режиме постоянной тяги. Необходимо построить маршрут с минимально возможным временем прохождения, где $A, B \in \Omega \setminus H(t), t > 0$.

1.3. Математическая формализация. Пусть Γ – произвольная гладкая кривая, соединяющая точки A и B :

$$\Gamma: [0, L] \rightarrow \Omega \setminus H(t), \Gamma(0) = A, \Gamma(L) = B,$$

где L – полная длина кривой, $s \in [0, L]$ – натуральный параметр, т.е. $\Gamma(s)$ определяет точку на кривой на расстоянии s от ее начала. В каждой точке кривой определен единичный касательный вектор:

$$\vec{\tau}(s) = \frac{1}{\sqrt{1+\Gamma'(s)^2}} \cdot \vec{\tau}_x + \frac{\Gamma'(s)}{\sqrt{1+\Gamma'(s)^2}} \cdot \vec{\tau}_y, \|\vec{\tau}(s)\| = 1.$$

Если объект движется вдоль кривой Γ с путевой скоростью $v(s)$ (скорость вдоль кривой в неподвижной системе координат), то его собственная скорость относительно среды $u(s)$ должна удовлетворять векторному уравнению

$$v(s)\vec{\tau}(s) = \vec{u}(s) + \vec{w}(\Gamma(s)), \|\vec{u}(s)\| = v.$$

Выражая $\vec{u}(s) = v(s)\vec{\tau}(s) - \vec{w}(\Gamma(s))$ и возводя в квадрат, имеем

$$\|v(s)\vec{\tau}(s) - \vec{w}(\Gamma(s))\|^2 = v^2.$$

Раскрыв скобки с учетом $\|\vec{\tau}(s)\| = 1$, получаем, что

$$v^2(s) - 2v(s)(\vec{w} \cdot \vec{\tau}) + \|\vec{w}\|^2 = v^2.$$

Это квадратное уравнение относительно $v(s)$, его положительный корень, соответствующий движению вперед (второй корень соответствует движению назад), имеет вид

$$v(s) = \vec{w}(\Gamma(s)) \cdot \vec{\tau}(s) + \sqrt{v^2 - \|\vec{w}(\Gamma(s))\|^2 + [\vec{w}(\Gamma(s)) \cdot \vec{\tau}(s)]^2}. \quad (1.1)$$

Предполагается, что дискриминант в (1.1) всюду является положительным, в противном случае попасть из точки A в точку B невозможно.

Время движения вдоль кривой Γ равно интегралу от обратной путевой скорости по длине маршрута

$$T(\Gamma) = \int_{\Gamma} dt = \int_0^L \frac{ds}{v(s)}.$$

Таким образом, минимальное время достижения точки B из точки A находится, как решение следующей вариационной задачи:

$$T(B) = \min_{\Gamma: A \rightarrow B} \int_0^L \frac{ds}{\vec{w} \cdot \vec{\tau} + \sqrt{v^2 - \|\vec{w}\|^2 + (\vec{w} \cdot \vec{\tau})^2}}. \quad (1.2)$$

2. О методе решения задач. Введем функцию $T(x, y)$, равную минимальному времени перемещения из фиксированной стартовой точки A в произвольную точку

$$T(x, y) = \min_{\Gamma: A \rightarrow (x, y)} \int_{\Gamma} \frac{ds}{\vec{w} \cdot \vec{\tau} + \sqrt{v^2 - \|\vec{w}\|^2 + (\vec{w} \cdot \vec{\tau})^2}}.$$

Множества $\{(x, y) : T(x, y) = \text{const}\}$ образуют семейство изохрон. Вектор-градиент ∇T ортогонален изохронам и указывает направление наискорейшего роста функции T .

Рассмотрим две изохроны $T(x, y) = t$ и $T(x, y) = t + dt$. Пусть расстояние между изохронами вдоль нормали \vec{n} равно dl . Тогда приращение dT при переходе с одной изохроны на другую можно выразить двумя способами: с одной стороны, по определению изохрон, это приращение равно dt , с другой – из определения градиента имеем, что $dT = \|\nabla T\| dl$ и тогда получаем следующее выражение для скорости движения фронта между изохронами:

$$v_n = \frac{dl}{dt} = \frac{1}{\|\nabla T\|}.$$

В то же время, абсолютная скорость движения элемента фронта складывается из собственной скорости $v\vec{n}$, поскольку фронт распространяется вдоль нормали относительно среды, и переносной скорости среды \vec{w} : $\vec{V} = v\vec{n} + \vec{w}$. Проекция этой скорости на нормаль есть:

$$v_n = \vec{V} \cdot \vec{n} = (v\vec{n} + \vec{w}) \cdot \vec{n} = v + \vec{w} \cdot \vec{n}.$$

Приравняем выражения для v_n . С учетом того, что единичный вектор нормали выражается через градиент как $\vec{n} = \nabla T / \|\nabla T\|$, получаем равенство:

$$\frac{1}{\|\nabla T\|} = v + \vec{w} \cdot \frac{\nabla T}{\|\nabla T\|}.$$

Умножая обе части на $\|\nabla T\|$, получаем обобщенное (анизотропное) уравнение эйконала:

$$1 = v\|\nabla T\| + \vec{w} \cdot \nabla T.$$

Таким образом, функция минимального времени $T(x, y)$ является решением следующей краевой задачи:

$$\begin{cases} v\|\nabla T\| + \vec{w} \cdot \nabla T = 1, x \in \Omega \setminus \{A\}, \\ T(A) = 0, \\ T(x, y) = +\infty, (x, y) \in H(t). \end{cases} \quad (2.1)$$

Если функция $T(x)$ найдена, то оптимальная траектория из B в A получается, как решение системы обыкновенных дифференциальных уравнений в попятном времени:

$$\frac{d}{d\tau} \begin{pmatrix} x \\ y \end{pmatrix} = - \frac{v \frac{\nabla T}{\|\nabla T\|} + \vec{w}}{\left\| v \frac{\nabla T}{\|\nabla T\|} + \vec{w} \right\|}, \quad x(0) = x_B, y(0) = y_B, \quad (2.2)$$

где τ пробегает значения от 0 до $T(x_B, y_B)$. В результате получается искомый оптимальный маршрут – кривая, соединяющая точки A и B .

3. Вычислительные алгоритмы. Для численного решения задачи (2.1) введем равномерную прямоугольную сетку $X_h \subset \Omega$ с шагом h по обеим координатам. Рассмотрим два узла сетки $q = (x_q, y_q)$ и $m = (x_m, y_m)$. Обозначим $\Delta x = x_m - x_q$, $\Delta y = y_m - y_q$.

Пусть объект движется из q в m , выбирая направление θ . Вектор собственной скорости относительно среды равен $(v \cos \theta, v \sin \theta)$. За время t объект должен пройти путь $(\Delta x, \Delta y)$ с абсолютной скоростью, равной сумме собственной скорости и скорости течения:

$$\begin{cases} \Delta x = [v \cos \theta + w_1(q)]t, \\ \Delta y = [v \sin \theta + w_2(q)]t, \end{cases} \quad (3.1)$$

где $w_1(q), w_2(q)$ – компоненты поля \vec{w} в точке q .

Выразим из (3.1) проекции собственной скорости, возведем в квадрат, и, используя универсальное тригонометрическое тождество $\sin^2 \theta + \cos^2 \theta = 1$, получаем:

$$(w_1^2 + w_2^2 - v^2)t^2 - 2(w_1 \Delta x + w_2 \Delta y)t + (\Delta x^2 + \Delta y^2) = 0. \quad (3.2)$$

Наименьший положительный корень уравнения (3.2) дает минимальное время t_{qm} , за которое объект может переместиться из q в m . Если уравнение не имеет положительных корней, переход невозможен.

Алгоритм 1: Построение скалярного поля времен

Шаг 0 (инициализация): Для каждого узла $p(x_p, y_p) \in X_h$ устанавливаются время прихода волны $T_p = +\infty$ и предшествующий узел $S_p = \emptyset$. Для начальной точки A полагается $T_A = 0$. Вводятся шаг по времени Δt , с которым проверяется положение препятствий, и счетчик шагов по времени $N = 0$. Создаются приоритетная очередь Q и список отложенных узлов $L, Q = \{A\}, L = \emptyset$.

Шаг 1: Если $Q \neq \emptyset$, для первого элемента $q \in Q$ определяется множество M , включающее $(2k + 1)^2 - 1, k \geq 1$ точек, соседних по сетке с точкой $q(x_q, y_q)$:

$$M = \{m(x_q + \delta_x, y_q + \delta_y) : \delta_x = ih, \delta_y = jh, i, j = -k, \dots, k\}.$$

Шаг 2: Для каждого $m \in M$ вычисляется время перехода t_{qm} , как наименьший положительный корень уравнения (3.2). Если положительного корня нет, то $t_{qm} = +\infty$. Подсчитывается $T'_m = T_q + t_{qm}$. Если $T_q = +\infty$ или $t_{qm} = +\infty$, то $T'_m = +\infty$.

Шаг 3: Если $m \notin H(T'_m)$ и $T'_m < T_m$, то $T_m = T'_m, S_m = \{q\}$.

Шаг 4: Если $T_m \leq t_0 + (N + 1)\Delta t$, то $Q = Q \cup \{m\}$ (волна достигла узла в текущем временном слое), иначе $L = L \cup \{m\}$.

Шаги 2 - 4 выполняются независимо для всех $m \in M$.

Шаг 5: Точка q исключается из очереди $Q = Q \setminus \{q\}$. Если $Q = \emptyset$, переход к шагу 6, иначе переход к шагу 1.

Шаг 6: Если $Q = \emptyset$, выполняется цикл проверки отложенных узлов: $Q = \emptyset$ и $L \neq \emptyset$, то увеличивается счетчик времени: $N = N + 1$, формируется множество

$U = \{p \in L : T_p \leq t_0 + (N+1)\Delta t\}$ и элементы множества U переносятся из L в $Q: Q = Q \cup U$, $L = L \setminus U$.

Шаг 7: Если $Q \neq \emptyset$, переход к шагу 1. Если $Q = \emptyset$ и $L = \emptyset$, алгоритм завершается.

После завершения алгоритма для каждого узла сетки $p \in X_n$ известно время прихода волны T_p из стартовой точки A и узел-предшественник S_p .

Алгоритм 2: Построение маршрута (RCA)

Алгоритм позволяет найти самый быстрый маршрут из точки A в точку B , $A, B \in \Omega \setminus H(t)$ и определить, какая его часть будет пройдена за заданный интервал времени Δt .

Шаг 0 (инициализация): Пусть $L(A, B)$ – список точек маршрута из A в B . Полагается $L(A, B) = \{B\}$. Вводится параметр $type$, определяющий режим построения маршрута. Если $type = 0$, то строится полный маршрут из точки A в точку B , $type = 1$ – определяется часть маршрута, проходимая за время Δt .

Шаг 1: Для точки A строится скалярное поле времени по алгоритму 1.

Шаг 2: Если $T_B = +\infty$, алгоритм завершается: маршрут не существует.

Шаг 3: Для последней точки p в списке $L(A, B)$ определяется ее предшественник S_p , который добавляется в конец списка: $L(A, B) = L(A, B) \cup \{S_p\}$.

Шаг 4: Если $S_p = A$, то переход к шагу 5. В противном случае переход к шагу 3.

Шаг 5: Элементы списка $L(A, B)$ упорядочиваются в обратном порядке, так чтобы начальной точкой стала A , а конечной – B .

Шаг 6: Если $type = 0$, то алгоритм завершается возвратом списка $L(A, B)$. В противном случае переход к шагу 7.

Шаг 7: В маршруте $L(A, B)$ находится точка $p^* \in L(A, B)$, для которой время T_{p^*} удовлетворяет условию $T_{p^*} = \max\{T_p : p \in L(A, B), T_p \leq t_0 + \Delta t\}$. Эта точка является наиболее удаленной точкой на маршруте, которую объект может достичь за время Δt .

Шаг 8: Находится подсписок $L(A, p^*) \subseteq L(A, B)$, содержащий точки от A до p^* . Алгоритм завершается.

Алгоритм возвращает список точек $L(A, p^*)$, представляющий собой маршрут движения объекта из начальной точки A в точку p^* , достижимую за время Δt . Если $p^* = A$, то объект не может сдвинуться с начальной точки за отведенное время, а если $p^* = B$, то объект успевает достичь конечной точки.

4. Вычислительный эксперимент. Все алгоритмы реализованы на языке C# в пакете Visual Studio 2022, вычисления производились на персональном компьютере с конфигурацией Intel (R) Core(TM) i5-3337U (частота 1.8 GHz, 4 CPUs, 6 GB RAM) и операционной системой Windows 10. Для демонстрации возможностей алгоритма выбраны четыре сценария с усложняющимися условиями: от однородного течения до комбинации вихря, статических и движущихся препятствий. Параметры среды (размеры области, скорости течений, расположение препятствий) и характеристики объекта (собственная скорость) подобраны так, чтобы наглядно показать особенности оптимальных траекторий, в частности, использование попутных течений, обход препятствий, перепланирование при движущихся барьерах. Численные значения не привязаны к конкретному натурному

эксперименту, но соответствуют порядкам величин, характерным для подводных аппаратов в прибрежных зонах. Для сравнения эффективности предложенного алгоритма RCA был использован алгоритм SSA (Salp Swarm Algorithm) [20], поскольку в [21, 22] показано, что он демонстрирует лучшие результаты по сравнению с другими эвристическими алгоритмами (PSO, WOA, GWO и MVO).

Пример 1. Рассматривается задача движения объекта со скоростью $v = 0.2$ (м/с) из точки $S(20;20)$ в точку $F(90;90)$ в среде размером 100×100 (м) без препятствий $H(t) = \emptyset$. Поле скоростей среды описывается следующим образом:

$$\vec{w}(w_1, w_2) = \begin{cases} \omega \vec{i} + 0 \vec{j}, & y \in [30; 70], \\ 0 \vec{i} + 0 \vec{j}, & y \notin [30; 70], \end{cases} \text{ где } \omega > 0.$$

В полосе $y \in [30; 70]$ имеет место течение, направленное вдоль оси Ox со скоростью ω (м/с).

Маршрут строится с помощью описанных выше алгоритмов со следующими параметрами: шаг сетки $h = 0.2$ (м), временной шаг $\Delta t = 1$ (с), размер окрестности $k = 3$, $type = 0$. Результаты расчетов представлены на рисунке 1.

При $\omega = 0.3$ (м/с) объект движется по прямой от S до точки входа в поток $S_1(20;30)$, затем использует попутное течение и перемещается в точку $S_2(90;70.4)$, после чего, покинув зону течения, следует прямо к цели F . Время движения составляет $t_{0.3} = 356.9$ (с), длина маршрута – $l_{0.3} = 110.42$ (м). Маршрут показан жирной линией на рисунке 1а.

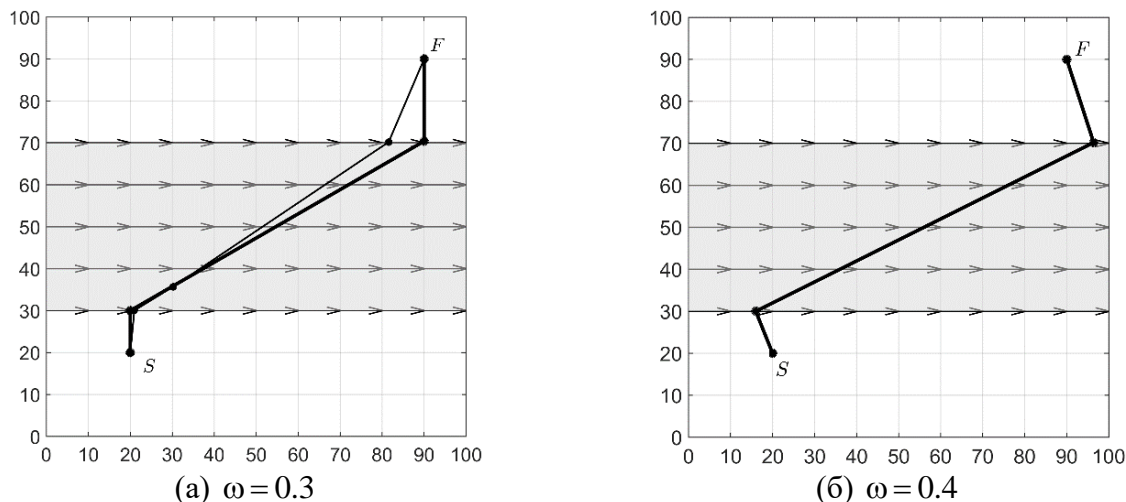


Рис. 1. Маршруты движения объекта в примере 1

При увеличении скорости потока до $\omega = 0.4$ (м/с) маршрут объекта еще сильнее отклоняется от прямой, соединяющей S и F (см. рис. 1б). Такое поведение объясняется стремлением алгоритма максимально эффективно использовать течение для минимизации общего времени в пути. Время движения – $t_{0.4} = 357.8$ (с), длина маршрута – $l_{0.4} = 120.67$ (м).

Результаты сравнения с алгоритмом SSA представлены в таблице 1. При $\omega = 0.3$ (м/с), наилучший найденный им маршрут длиной 104.32 (м) имеет вид ломаной из пяти отрезков с вершинами $P_0(20;20)$; $P_1(20.96; 30.05)$; $P_2(30.21; 35.70)$; $P_3(81.56; 70.21)$; $P_4(90; 90)$, время движения составляет 359.75 (с). Маршрут показан тонкой линией на рисунке 1а.

Таблица 1. Результаты эксперимента 1

Алгоритм	Длина маршрута	Время движения
RCA	110.42 (м)	356.9 (с)
SSA	104.32 (м)	359.75 (с)

Пример 2. Рассматривается область размером 100×100 (м) без препятствий ($H(t) = \emptyset$). Скорость течения задается следующим образом:

$$\vec{w}(w_1, w_2) = \begin{cases} 0\vec{i} + 0\vec{j}, & x \in [0; 20] \cup [40; 60] \cup [80; 100], \\ 0.3\vec{i} + 0\vec{j}, & x \in (20; 40), \\ 0\vec{i} - 0.3\vec{j}, & x \in (60; 80), \end{cases}$$

Объект движется из точки $S(10; 30)$ в точку $F(90; 30)$ с собственной скоростью $v = 0.2$ (м/с).

Как видно из описания, в области имеются два противоположно направленных течения: одно – в положительном направлении оси Ox (поток 1), другое – в отрицательном (поток 2). Полученная траектория показана на рисунке 2а. Объект движется по прямой, чтобы войти в поток 1 в ближайшей точке $S_1(20.2; 30)$, следует по течению до точки $S_2(40; 59.4)$, затем по кратчайшему пути пересекает область без течения и входит в поток 2 в точке $S_3(60; 59.4)$, движется по нему до точки $S_4(80; 30)$ и, наконец, достигает цели F по кратчайшему пути. Время движения составляет 400 (с), длина маршрута 112.2 (м).

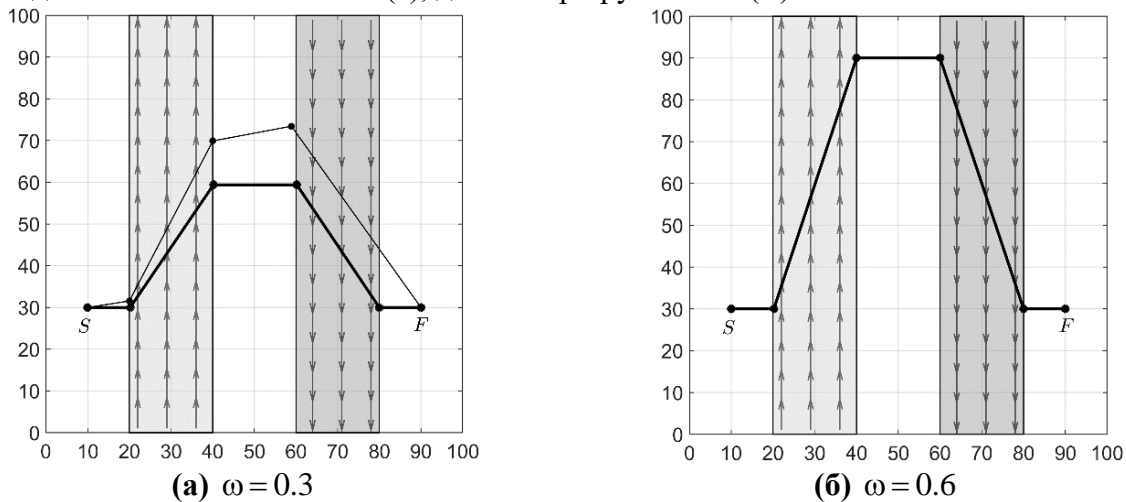


Рис. 2. Маршруты движения объекта в примере 2

Если увеличить скорость обоих потоков до 0.6 (м/с), траектория удлинится, но время движения остается равным 400 (с) (см. рис. 2б).

Результаты сравнения с алгоритмом SSA представлены в таблице 2. Можно видеть, что в этом случае RCA показал результаты лучше, чем SSA, как по длине маршрута, так и по времени движения.

Таблица 2. Результаты эксперимента 2

Алгоритм	Длина маршрута	Время движения
RCA	112.2 (м)	400 (с)
SSA	126 (м)	480 (с)

Пример 3. Рассматривается задача поиска маршрута из точки $S(10;10)$ до $F(90;90)$ в области размером 100×100 (м), содержащей неподвижные препятствия. В полосе $x \in [30, 70]$ присутствует течение, направленное вдоль оси Ox и имеющее скорость 0.3 (м/с). Кроме того, в области имеется вихревая зона, имеющая форму круга с центром в точке $O(50,50)$ и радиусом $r = 12$ (м). Внутри этой зоны вращение происходит по часовой стрелке, скорость течения направлена по касательной к окружности, постоянна по модулю и равна 0.5 (м/с). Таким образом, поле скорости задается следующим образом:

$$\vec{w}(w_1, w_2) = \begin{cases} 0\vec{i} + 0\vec{j}, & y \in [0;30) \cup (70;100] \cup (50;50), \\ 0.3\vec{i} + 0\vec{j}, & y \in [30;70], d > 12, \\ a\vec{i} + b\vec{j}, & y \in [30;70], 0 < d \leq 12, \end{cases}$$

где $d = \sqrt{(x-50)^2 + (y-50)^2}$, $a = -0.5(y-50)d^{-1}$, $b = 0.5(x-50)d^{-1}$.

В области также расположены два прямоугольных препятствия

$$H_1(t) = \{(x, y) : 0 \leq x \leq 15, 15 \leq y \leq 25\}, \quad H_2(t) = \{(x, y) : 85 \leq x \leq 100, 75 \leq y \leq 85\}.$$

Собственная скорость объекта составляет 0.2 (м/с).

Построенный маршрут показан на рисунке 3. На первом этапе (0-114.4 с) объект движется из стартовой точки $S(10;10)$ к нижней границе области течения, огибает препятствие H_1 , проходя через $S_1(15.2; 15.2)$, и достигает точки входа в поток в $S_2(20.1; 30)$. На втором этапе (114.4-277.1 с) объект входит в область горизонтального течения, движется в потоке в сторону вихревой зоны и входит в нее в $S_3(42.1; 41)$. В вихре скорость направлена по касательной к окружности, поэтому траектория искривляется. Двигаясь внутри вихря, объект достигает точки $S_4(50.6; 61.4)$, где направление вихревого течения максимально отклоняет траекторию от прямолинейной. В точке $S_5(53.8; 61.4)$ объект покидает вихревую зону и выходит в область чистого горизонтального течения, после чего продолжает движение в потоке, покидая его в точке $S_6(79.6; 70)$. На заключительном этапе (277.1-392.1 с) огибает препятствие H_2 , проходя через точку $S_7(84.9; 85.1)$, и достигает целевой точки. Общее время движения составляет 392.1 с.

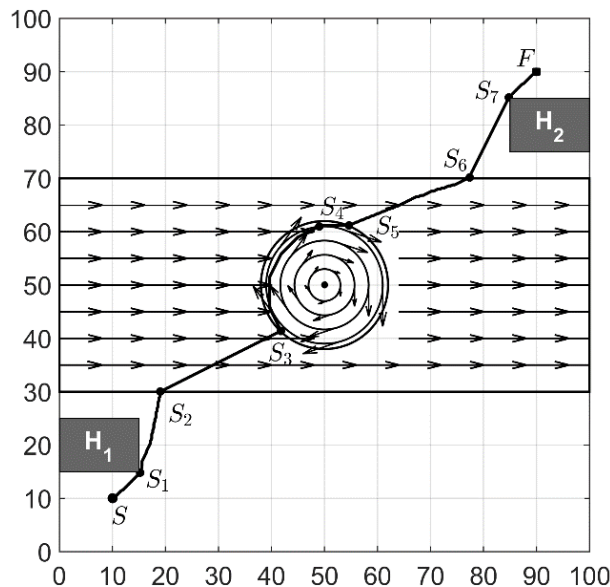


Рис. 3. Маршрут объекта в пример 3

Отметим, что прямолинейное движение в неподвижной среде без препятствий потребовало бы 565.7 (с). Выигрыш во времени достигается за счет использования течения и вихря, несмотря на необходимость обхода препятствий.

Пример 4. Рассматривается область с двумя противоположно направленными вертикальными течениями. Скорость каждого из потоков составляет 0.1 (м/с). В центре области $O(50;50)$ расположена вихревая зона – круг радиуса $r = 10$ (м). Внутри этой зоны вращение происходит по часовой стрелке, вектор скорости направлен по касательной к окружности, а модуль возрастает при приближении к центру (суперпозиция вращения и радиального сноса к центру). Таким образом, поле скорости задается следующим образом:

$$\vec{w}(w_1, w_2) = \begin{cases} 0\vec{i} + 0.1\vec{j}, & x \in [0;50], d > 10, \\ 0\vec{i} - 0.1\vec{j}, & x \in [50;100], d > 10, \\ a\vec{i} + b\vec{j}, & 0 < d \leq 10, \\ 0\vec{i} + 0\vec{j}, & d = 0, \end{cases}$$

где $d = \sqrt{(x-50)^2 + (y-50)^2}$, $a = 0.3(y-50)d^{-1} - (x-50)d^{-1}$, $b = -0.3(x-50)d^{-1} - (y-50)d^{-1}$.

В области имеются три препятствия: статический барьер H_1 , представляющий собой квадрат со стороной 30 (м), барьер H_2 – квадрат со стороной 20 (м), движущийся слева направо со скоростью 0.3 (м/с), и H_3 – прямоугольник 10×24 (м), движущийся справа налево со скоростью 0.07 (м/с). Тогда множество $H(t)$ имеет вид

$$H(t) = H_1(t) \cup H_2(t) \cup H_3(t), \quad \text{где } H_1(t) = \{(x, y) : 0 \leq x \leq 20, 30 \leq y \leq 50\}, \\ H_2(t) = \{(x, y) : 0 \leq x - 0.3t \leq 10, 10 \leq y \leq 20\}, H_3(t) = \{(x, y) : 75 \leq x + 0.07t \leq 99, 75 \leq y \leq 85\}.$$

Объект движется из точки $S(0;0)$ в точку $F(99;99)$ с собственной скоростью 0.2 (м/с).

Для решения задачи Алгоритм RCA применяется многократно: при каждом запуске в режиме $type = 1$ строится участок маршрута, который объект проходит за время $\Delta t = 1.667$ (с). После каждого запуска начальная точка сдвигается в текущее положение объекта.

На первом интервале времени $[0;16.67]$ (с) объект движется по прямой и достигает точки $S_1(3;2.67)$ (рис. 4а). На интервале $[16.67;46.67]$ (с) препятствие H_2 смещается вправо, и объект, отклоняясь от прямолинейного курса, чтобы избежать столкновения, достигает точки $S_2(6.19;8.19)$ (рис. 4б). На интервале $[46.67;145]$ (с) объект движется в области, свободной от препятствий, и направляется к точке $S_3(24;26.19)$, избегая попадания в центр вихревой зоны (рис. 4в). На интервале $[145;235]$ (с) препятствие H_3 смещается влево и блокирует прямой путь. Объект мог бы обойти его справа от вихря, но тогда он двигался бы против течения, что увеличило бы время. Поэтому выбирается траектория слева от вихря, проходящая вблизи препятствия H_1 и использующая попутное течение. Объект достигает точки $S_4(24.33;52.97)$ (рис. 4г).

На заключительном интервале $[235;721.67]$ (с) объект движется вдоль правого течения, затем перпендикулярно ему, к цели. Полное время движения $T^* = 721.67$ (с) (рис. 5).

Таким образом, во всех сценариях алгоритм RCA находит маршруты с меньшим временем движения по сравнению с SSA там, где сравнение проводилось (выигрыш до 20% в примере 2). В примере 3 рациональное использование течений позволяет сократить время на 30% по сравнению с прямолинейным движением в неподвижной среде. Геометрическая длина траектории RCA может быть больше, чем у SSA (пример 1), что является «платой» за

использование попутных течений. Поскольку собственная скорость объекта постоянна, энергозатраты пропорциональны времени движения, поэтому RCA также минимизирует энергопотребление. В примере 4 продемонстрирована возможность оперативного перепланирования маршрута при движущихся препятствиях, что подтверждает пригодность алгоритма для систем реального времени.

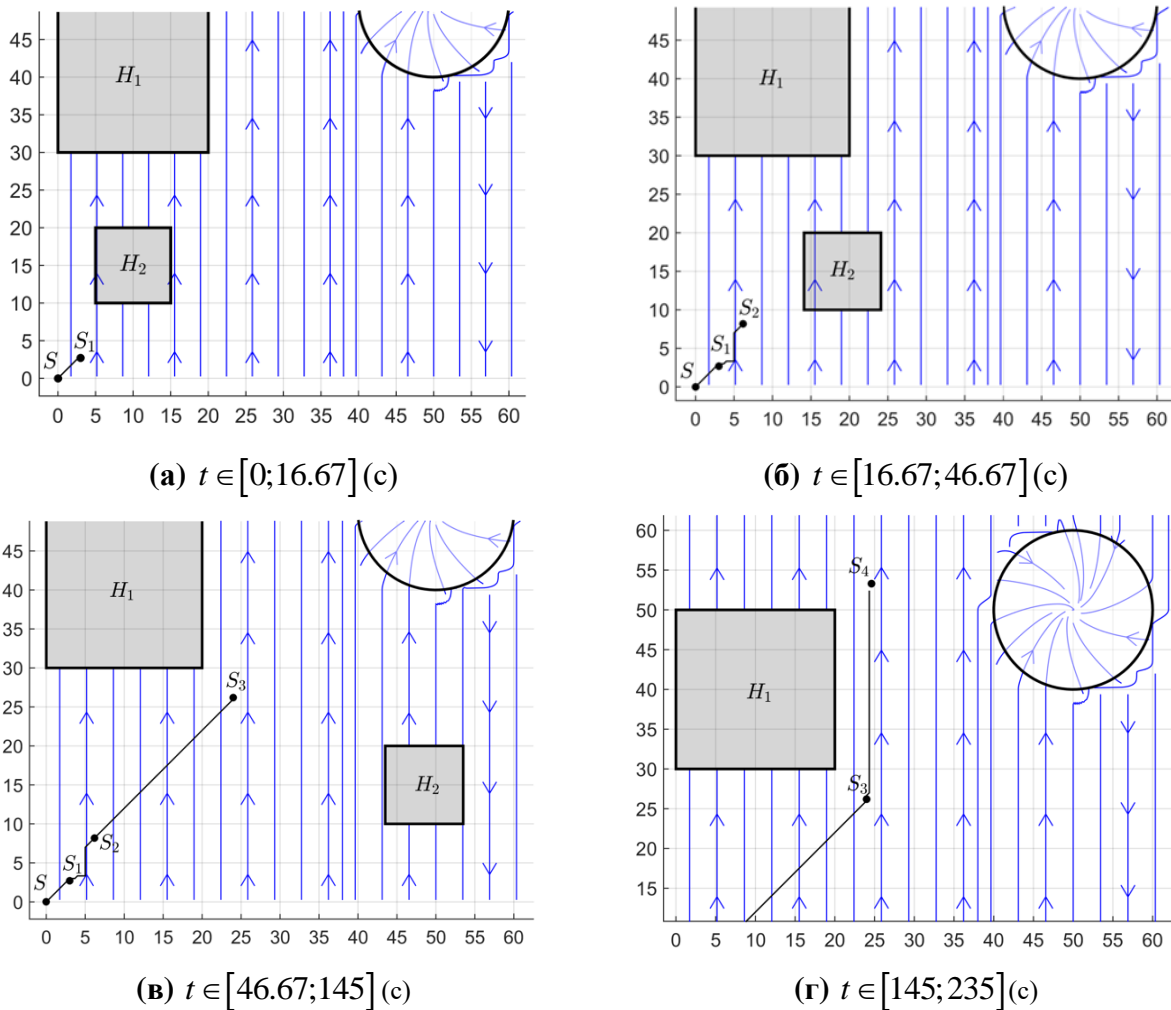


Рис. 4. Маршрут объекта на различных промежутках времени в примере 4

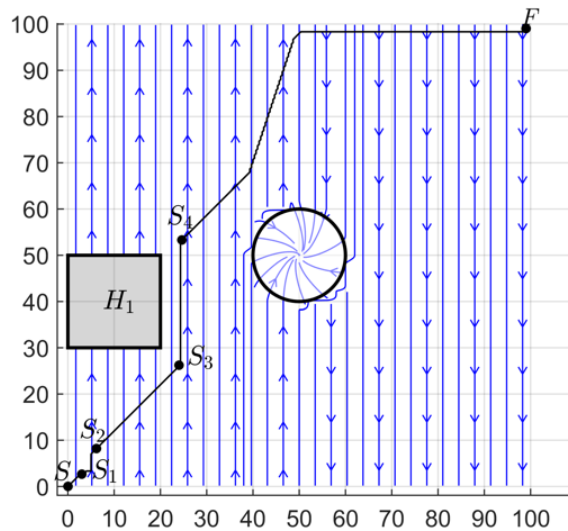


Рис. 5. Маршрут объекта в примере 4

Заключение. В работе рассмотрена задача поиска минимальных по времени маршрутов в динамической водной среде с учетом течений и движущихся препятствий. Предложен новый подход, основанный на численном решении уравнения эйконала и модифицированном методе быстрого марша. Разработаны два базовых алгоритма.

На различных сценариях показано, что разработанный алгоритм не только строит траектории, минимизирующие время движения, но и обеспечивает безопасный обход статических и динамических препятствий за счет оперативной перепланировки пути. Сравнение с методом SSA подтверждает преимущество предложенного подхода по основному критерию – времени в пути. При этом время расчета траектории достаточно мало, что позволяет осуществлять планирование движения в режиме реального времени.

В реальных условиях данные о течениях и препятствиях поступают от навигационных систем с определенной погрешностью. Предложенный алгоритм может работать в режиме повторного планирования: по мере получения уточненной информации поле времени пересчитывается локально. Анализ чувствительности к ошибкам входных данных является предметом дальнейших исследований.

Кроме того, дальнейшие исследования могут быть направлены на учет нестационарных полей течений, а также на адаптацию метода для работы с более сложными моделями движения объекта, учитывающими, в том числе, парусность – зависимость собственной скорости от ориентации.

Список источников

1. Liu L., Sun M., Zhao E. et al. Three-dimensional dynamic trajectory planning for autonomous underwater robots under the PPO-IIFDS framework. *Journal of Marine Science and Engineering*, 2025, vol. 13, no. 3, p. 445, DOI: 10.3390/jmse13030445.
2. Paull L., Saeedi S., Seto M., Li H. AUV navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, 2014, vol. 39, no. 1, pp. 131-149. DOI: 10.1109/JOE.2013.2278891.
3. Бухаров Д.С. Программная система «Виголт» для решения задач оптимизации, возникающих в транспортной логистике / Д.С. Бухаров, А.Л. Казаков // *Вычислительные методы и программирование*, 2012. – Т. 13. – № 3. – С. 65-74.
4. Lempert A.A., Kazakov A.L., Bukharov D.S. Mathematical model and program system for solving a problem of logistic objects placement. *Automation and Remote Control*, 2015, vol. 76, no. 8, pp. 1463-1470.
5. Doshi M.M., Bhabra M.S., Lermusiaux P.F.J. Energy–time optimal path planning in dynamic flows: theory and schemes. *Computer Methods in Applied Mechanics and Engineering*, 2023, vol. 405, p. 115865, DOI: 10.1016/j.cma.2022.115865.
6. Chen M., Zhu D. Optimal time-consuming path planning for autonomous underwater vehicles based on a dynamic neural network model in ocean current environments. *IEEE Transactions on Vehicular Technology*, 2020, vol. 69, pp. 14401-14412, DOI: 10.1109/TVT.2020.3034628.
7. Zhu D., Yang S.X. Path planning method for unmanned underwater vehicles eliminating effect of currents based on artificial potential field. *Journal of Navigation*, 2021, vol. 74, pp. 955-967, DOI: 10.1017/S0373463321000345.
8. Xanthidis M., Karapetyan N., Damron H. et al. Navigation in the presence of obstacles for an agile autonomous underwater vehicle. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2020, pp. 892-899, DOI:10.48550/arXiv.1903.11750.
9. Yang H., Mahmoudian N. Effective underwater glider path planning in dynamic 3D environments using multi-point potential fields. *arXiv preprint*, arXiv:2403.08163, 2024, DOI: 10.48550/arXiv.2403.08163.
10. Zhu D., Cheng C., Sun B. An integrated AUV path planning algorithm with ocean current and dynamic obstacles. *International Journal of Robotics and Automation*, 2016, vol. 31, pp. 382-389, DOI: 10.2316/Journal.206.2016.5.206-4570.
11. Mahmoud Zadeh S., Powers D.M.W., Yazdani A. et al. Efficient AUV path planning in time-variant underwater environment using differential evolution algorithm. *Journal of Marine Science and Application*, 2018, vol. 17, no. 4, pp. 585-591, DOI: 10.1007/s11804-018-0034-4.
12. Raheem F.A., Hameed U.I. Path planning algorithm using D* heuristic method based on PSO in dynamic environment. *American Scientific Research Journal for Engineering, Technology and Sciences*, 2018, vol. 490, no. 1, pp. 257-271.

13. Jin P., Li W., Zhan J., Shan C. 3D Obstacle Avoidance Path Planning Algorithm and Software Design for UUV Based on Improved D* Lite-APF. *Journal of Marine Science and Engineering*, 2026, vol. 14, no. 4, p. 373. DOI: 10.3390/jmse14040373.
14. Gao J., Geng X., Zhang Y. et al. Underwater vehicle path planning based on bidirectional path and cached random tree star algorithm. *Applied Sciences*, 2024, vol. 14, p. 947, DOI: 10.3390/app14020947.
15. Chu Z., Wang F., Lei T. et al. Path planning based on deep reinforcement learning for autonomous underwater vehicles under ocean current disturbance. *IEEE Transactions on Intelligent Vehicles*, 2022, vol. 8, no. 1, pp. 108–120, DOI: 10.1109/TIV.2022.3153352.
16. Казаков А.Л. Об одном подходе к решению задач оптимизации, возникающих в транспортной логистике / А.Л. Казаков, А.А. Лемперт // *Автоматика и телемеханика*, 2011. – № 7. – С. 50–57. – DOI: 10.1134/S0005117911070071.
17. Казаков А.Л. Об одном численном методе решения некоторых задач оптимизации, возникающих в транспортной логистике / А.Л. Казаков, А.А. Лемперт, Д.С. Бухаров // *Вестник Иркутского государственного технического университета*, 2011. – № 6 (53). – С. 6–12.
18. Казаков А.Л. О построении маршрутов в динамической среде с использованием решений уравнения эйконала / А.Л. Казаков, А.А. Лемперт // *Известия Института математики и информатики Удмуртского государственного университета*, 2021. – Т. 82. – № 6. – С. 3–13. – DOI: 10.35634/2226-3594-2021-58-04.
19. Боровских А.В. Двумерное уравнение эйконала // *Сибирский математический журнал*, 2006. – Т. 47. – № 5. – С. 993–1018. – DOI:10.1007/s11202-006-0091-9.
20. Mirjalili S., Gandomi A.H., Mirjalili S.Z. et al. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 2017, vol. 114, pp. 163–191, DOI:10.1016/j.advengsoft.2017.07.002.
21. Guo X., Zhao D., Fan T. et al., Autonomous underwater vehicle path planning based on improved salp swarm algorithm. *Journal of Marine Science and Engineering*, 2024, vol. 12, no 8, p. 1446. DOI:10.3390/jmse12081446.
22. Joshi M., Kalita K., Jangir P. et al. A conceptual comparison of dragonfly algorithm variants for CEC-2021 global optimization problems. *Arabian Journal for Science and Engineering*, 2023, vol. 48, pp. 1563–1593, DOI: 10.1007/s13369-022-06880-9.

Казаков Александр Леонидович. Докт. физ.- мат. наук, профессор Иркутского национального исследовательского технического университета. AuthorID: 14911, SPIN: 8101-9424, ORCID: 0000-0002-3047-1650. kazakov@icc.ru, 664033, Россия, Иркутск.

Лемперт Анна Ананьевна. Канд. физ.- мат. наук, доцент Иркутского национального исследовательского технического университета, AuthorID: 128334, SPIN: 2774-4751, ORCID: 0000-0001-9562-7903. lempert@icc.ru, 664033, Россия, Иркутск.

Чан Туан Вьет. Аспирант Иркутского национального исследовательского технического университета, AuthorID: 1345480, SPIN: 7043-1530, ORCID: 0009-0008-1048-1473. mr.ttviet.1986@gmail.com, 664033, Россия, Иркутск.

UDC 519.853.6, 517.958

DOI:10.25729/ESI.2026.42.2.007

Time-optimal collision-free path planning in anisotropic environments with moving obstacles

Aleksandr L. Kazakov, Anna A. Lempert, Viet T. Tran

Irkutsk National Research Technical University, Russia, Irkutsk, kazakov@icc.ru

Abstract. This paper addresses the problem of constructing time-optimal paths for a moving vehicle, specifically an autonomous underwater vehicle operating in an aquatic environment. The main difficulties stem from spatially non-uniform currents and the presence of both static and moving obstacles. We propose an approach based on the optical-geometric analogy and the Fermat–Huygens principle. The problem is formulated as a

generalized eikonal equation that describes the propagation of a wavefront in a medium with a prescribed velocity vector field. This formulation reduces the original variational problem to a partial differential equation, eliminating the need for an exhaustive search over candidate trajectories. The arrival time field is obtained as the solution to a boundary value problem, and the optimal path is recovered by moving against the gradient of this field. We develop two algorithms for numerical implementation. The first, based on the fast marching method, calculates the minimal time field from the start point to all nodes of a computational grid. The second algorithm reconstructs the optimal trajectory from this field and extracts the section corresponding to a specified time horizon. We perform a series of numerical experiments on four test scenarios of increasing complexity: a uniform flow, two opposing flows, a flow with a vortex and static obstacles, and a scenario that combines moving obstacles with a vortex. In all cases, the algorithm constructs time-optimal paths successfully while ensuring safe obstacle avoidance. The scenario with dynamic obstacles demonstrated the ability to adjust the route dynamically. We compared the proposed approach with the Salp Swarm Algorithm (SSA). The results show that our algorithm finds paths with shorter travel times, even when their geometrical length exceeds that of the routes produced by SSA. Moreover, the computation time is sufficiently low to allow real-time path planning.

Keywords: routing problem, dynamic environment, optical-geometric approach, eikonal equation, fast marching method

References

1. Liu L., Sun M., Zhao E. et al. Three-dimensional dynamic trajectory planning for autonomous underwater robots under the PPO-IIFDS framework. *Journal of Marine Science and Engineering*, 2025, vol. 13, no. 3, p. 445, DOI: 10.3390/jmse13030445.
2. Paull L., Saeedi S., Seto M., Li H. AUV navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, 2014, vol. 39, no. 1, pp. 131-149. DOI: 10.1109/JOE.2013.2278891.
3. Bukharov D.S., Kazakov A.L. Programmaya sistema "Vigolt" dlya resheniya zadach optimizatsii, vznikayushchikh v transportnoy logistike [The Vigolt software system for solving optimization problems arising in transport logistics]. *Vychislitel'nyye metody i programmirovaniye [Computational Methods and Programming]*, 2012, vol. 13, no. 3, pp. 65-74.
4. Lempert A.A., Kazakov A.L., Bukharov D.S. Mathematical model and program system for solving a problem of logistic objects placement. *Automation and Remote Control*, 2015, vol. 76, no. 8, pp. 1463-1470.
5. Doshi M.M., Bhabra M.S., Lermusiaux P.F.J. Energy-time optimal path planning in dynamic flows: theory and schemes. *Computer Methods in Applied Mechanics and Engineering*, 2023, vol. 405, p. 115865, DOI: 10.1016/j.cma.2022.115865.
6. Chen M., Zhu D. Optimal time-consuming path planning for autonomous underwater vehicles based on a dynamic neural network model in ocean current environments. *IEEE Transactions on Vehicular Technology*, 2020, vol. 69, pp. 14401-14412, DOI: 10.1109/TVT.2020.3034628.
7. Zhu D., Yang S.X. Path planning method for unmanned underwater vehicles eliminating effect of currents based on artificial potential field. *Journal of Navigation*, 2021, vol. 74, pp. 955-967, DOI: 10.1017/S0373463321000345.
8. Xanthidis M., Karapetyan N., Damron H. et al. Navigation in the presence of obstacles for an agile autonomous underwater vehicle. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2020, pp. 892-899, DOI:10.48550/arXiv.1903.11750.
9. Yang H., Mahmoudian N. Effective underwater glider path planning in dynamic 3D environments using multi-point potential fields. *arXiv preprint*, arXiv:2403.08163, 2024, DOI: 10.48550/arXiv.2403.08163.
10. Zhu D., Cheng C., Sun B. An integrated AUV path planning algorithm with ocean current and dynamic obstacles. *International Journal of Robotics and Automation*, 2016, vol. 31, pp. 382-389, DOI: 10.2316/Journal.206.2016.5.206-4570.
11. Mahmoud Zadeh S., Powers D.M.W., Yazdani A. et al. Efficient AUV path planning in time-variant underwater environment using differential evolution algorithm. *Journal of Marine Science and Application*, 2018, vol. 17, no. 4, pp. 585-591, DOI: 10.1007/s11804-018-0034-4.
12. Raheem F.A., Hameed U.I. Path planning algorithm using D* heuristic method based on PSO in dynamic environment. *American Scientific Research Journal for Engineering, Technology and Sciences*, 2018, vol. 490, no. 1, pp. 257-271.
13. Jin P., Li W., Zhan J., Shan C. 3D Obstacle Avoidance Path Planning Algorithm and Software Design for UUV Based on Improved D* Lite-APF. *Journal of Marine Science and Engineering*, 2026, vol. 14, no. 4, p. 373. DOI: 10.3390/jmse14040373.

14. Gao J., Geng X., Zhang Y. et al. Underwater vehicle path planning based on bidirectional path and cached random tree star algorithm. *Applied Sciences*, 2024, vol. 14, p. 947, DOI: 10.3390/app14020947.
15. Chu Z., Wang F., Lei T. et al. Path planning based on deep reinforcement learning for autonomous underwater vehicles under ocean current disturbance. *IEEE Transactions on Intelligent Vehicles*, 2022, vol. 8, no. 1, pp. 108-120, DOI: 10.1109/TIV.2022.3153352.
16. Kazakov A.L., Lempert A.A. Ob odnom podkhode k resheniyu zadach optimizatsii, vznikayushchikh v transportnoy logistike [On one approach to solving optimization problems arising in transport logistics]. *Avtomatika i telemekhanika [Automation and Remote Control]*, 2011, no. 7, pp. 50–57, DOI: 10.1134/S0005117911070071.
17. Kazakov A.L., Lempert A.A., Bukharov D.S. Ob odnom chislennom metode resheniya nekotorykh zadach optimizatsii, vznikayushchikh v transportnoy logistike [On one numerical method for solving some optimization problems arising in transport logistics]. *Vestnik Irkutskogo gosudarstvennogo tekhnicheskogo universiteta [Bulletin of Irkutsk State Technical University]*, 2011, no. 6 (53), pp. 6-12.
18. Kazakov A.L., Lempert A.A. O postroyenii marshrutov v dinamicheskoy srede s ispol'zovaniyem resheniy uravneniya eikonala [On path planning in a dynamic environment using solutions of the eikonal equation]. *Izvestiya Instituta matematiki i informatiki Udmurtskogo gosudarstvennogo universiteta [Proceedings of the Institute of Mathematics and Informatics of Udmurt State University]*, 2021, vol. 82, no. 6, pp. 3–13, DOI: DOI: 10.35634/2226-3594-2021-58-04.
19. Borovskikh A.V. Dvumernoye uravneniye eikonala [Two-dimensional eikonal equation]. *Sibirskiy matematicheskiy zhurnal [Siberian Mathematical Journal]*, 2006, vol. 47, no. 5, pp. 993–1018, DOI: 10.1007/s11202-006-0091-9.
20. Mirjalili S., Gandomi A.H., Mirjalili S.Z. et al. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 2017, vol. 114, pp. 163-191, DOI:10.1016/j.advengsoft.2017.07.002.
21. Guo X., Zhao D., Fan T. et al., Autonomous underwater vehicle path planning based on improved salp swarm algorithm. *Journal of Marine Science and Engineering*, 2024, vol. 12, no 8, p. 1446. DOI:10.3390/jmse12081446.
22. Joshi M., Kalita K., Jangir P. et al. A conceptual comparison of dragonfly algorithm variants for CEC-2021 global optimization problems. *Arabian Journal for Science and Engineering*, 2023, vol. 48, pp. 1563-1593, DOI: 10.1007/s13369-022-06880-9.

Kazakov Aleksandr Leonidovich. *Dr. Phys.-Math. Sci., professor at Irkutsk National Research State Technical University, AuthorID: 14911, SPIN: 8101-9424, ORCID: 0000-0002-3047-1650. kazakov@icc.ru, 664033, Russia, Irkutsk.*

Lempert Anna Ananievna. *Cand. Phys.-Math. Sci., associate professor at Irkutsk National Research Technical University, AuthorID: 128334, SPIN: 2774-4751, ORCID: 0000-0001-9562-7903. lempert@icc.ru, 664033, Russia, Irkutsk.*

Tran Tuan Viet. *PhD student at Irkutsk National Research Technical University, AuthorID: 1345480, SPIN: 7043-1530, ORCID: 0009-0008-1048-1473. mr.ttviet.1986@gmail.com, 664033, Russia, Irkutsk.*

Статья поступила в редакцию 17.03.2026; одобрена после рецензирования 13.04.2026; принята к публикации 19.05.2026.

The article was submitted 03/17/2026; approved after reviewing 04/13/2026; accepted for publication 05/19/2026.