



Международный журнал информационных технологий и энергоэффективности

Сайт журнала: <http://www.openaccessscience.ru/index.php/ijcse/>



УДК 519.688

## ДИНАМИЧЕСКИЙ ВЫБОР АЛГОРИТМОВ НАХОЖДЕНИЯ ПУТИ

**Стальмаков И.А., Зернов М.М.**

*ФГБОУ ВО «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ» ФИЛИАЛ в г. Смоленске, Смоленск, Россия (214013, г. Смоленск, Энергетический проезд, 1), e-mail: keid.votvakson@yandex.ru*

В работе рассмотрены проблемы поиска пути в динамических (генерируемых) виртуальных средах на основе единого базового алгоритма. Предложены принципы построения комбинированного алгоритма, использующего набор базовых алгоритмов, выбор среди которых осуществляется для отдельных участков среды. Рассмотрены показатели, которые можно использовать в качестве исходной информации, исходя из которой будет осуществляться выбор алгоритма.

Ключевые слова: Динамическая среда, комбинированный алгоритм, поиск пути.

## DYNAMIC SELECTION OF PATH FINDING ALGORITHMS

**Stalmakov I.A .Zernov. M.M**

*FEDERAL STATE-FUNDED EDUCATIONAL INSTITUTION OF HIGHER PROFESSIONAL EDUCATION SMOLENSK STATE UNIVERSITY NATIONAL RESEARCH UNIVERSITY MPEI BRANCH IN SMOLENSK, Smolensk Russia (214013, Smolensk, Energetic Passage, 1), e-mail: keid.votvakson@yandex.ru*

The paper examines the problems of finding a path in dynamic (generated) virtual environments based on a single basic algorithm. The principles of constructing a combined algorithm using a set of basic algorithms are proposed, the choice among which is carried out for individual sections of the environment. Indicators that can be used as initial information on the basis of which the algorithm will be selected are considered.

Keywords: Dynamic environment, combined algorithm, pathfinding.

### Введение

В условиях быстрого развития технологий и автоматизации возникает все большая потребность в эффективных алгоритмах планирования пути для различных типов сред. Генерируемые среды, такие как пространства виртуальных миров, моделей имитации робототехнических систем и компьютерных игровых миров, представляют собой особый вызов для разработчиков, поскольку они часто динамичны и разнообразны.

Использование фиксированного алгоритма поиска пути, ориентирующего на наиболее сложные условия, может привести к недостаточной производительности алгоритмов в среднем и завышенному потреблению вычислительных ресурсов.

Для решения этой проблемы предлагается подход, основанный на динамическом выборе алгоритма в зависимости от характеристик среды. Этот подход позволяет оптимизировать процесс планирования пути, выбирая наиболее подходящий алгоритм в каждом конкретном случае

### Актуальность

Для того, чтобы оптимизировать перемещение в виртуальных средах разработчики идут на разного рода ухищрения. Одним из направлений оптимизации является выбор представления карты. Здесь и далее будем ориентироваться на задачу поиска маршрута в игровой среде, как наиболее универсальную.

Основными типами представления игровой карты местности для дальнейшего поиска пути, являются:

- условно-непрерывное представление (собственно, координатное пространство со списком размещённых на ней объектов-препятствий);
- условно-непрерывное представление (собственно, координатное пространство со списком размещённых на ней объектов-препятствий);
- дискретное представление карты с отметками проходимых и не проходимых участков — часто выступает в качестве промежуточного представления в процессе обработки карты с целью формирования более сложных представлений;
- граф соседства проходимых участков на основе регулярной сетки;
- маршрутный граф, часто соответствующий скелету проходимых областей, графу дорог, типовых маршрутов и т. д.;
- навигационный граф — граф соседства проходимых областей.

Варианты 3-5 представлены на Рисунке 1.



Рисунок 1 – Графовые представления информации о карте

Графовые представления позволяют использовать эффективные алгоритмы поиска пути. Например, в соответствии с представлением в виде навигационного графа, можно разбить большую карту на несколько маленьких участков[1]. Затем определить зазоры, рассчитать расстояния между всеми точками и применить иерархический аннотированный алгоритм A\*. Пример перехода от дискретного представления к представлению в виде навигационного графа представлен на Рисунке 2. Такой подход даёт почти оптимальные решения и на практике требует небольших затрат памяти.

По сути, переход к одному из вариантов графового представления обеспечивает сжатие информации. В идеале, подобрать такое представление карты, которое бы обеспечивало наименьшее количество узлов графового представления (наилучшее сжатие) при сохранении достаточной точности решения и «правдоподобности» пути с точки зрения поведения игрового персонажа[2]. Заметим, что построение представлений карты для применения высокоэффективных алгоритмов на графах, как правило, сопряжено с трудоёмким этапом первоначаль-

чального анализа карты, на который переносится значительная часть вычислений.

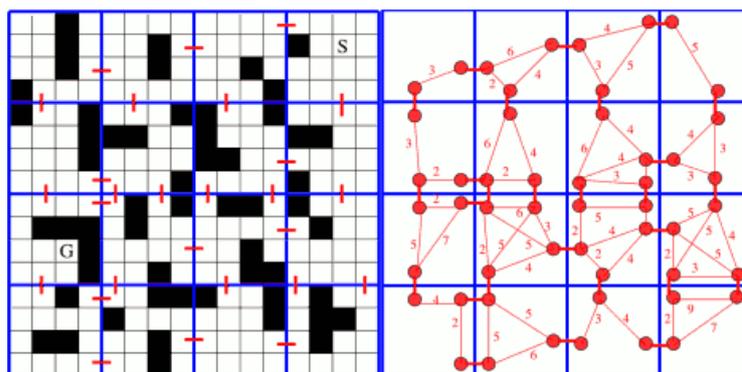


Рисунок 2 – Переход от дискретного представления карты к навигационному графу

Другим направлением повышения эффективности поиска пути является построение асинхронного алгоритма. Т.е. можно разбить процесс поиска на несколько кадров и за счёт этого снизить требования по быстродействию. Например, раз в кадр можно выполнять проверку соседних участков (Рисунок 3) [2]

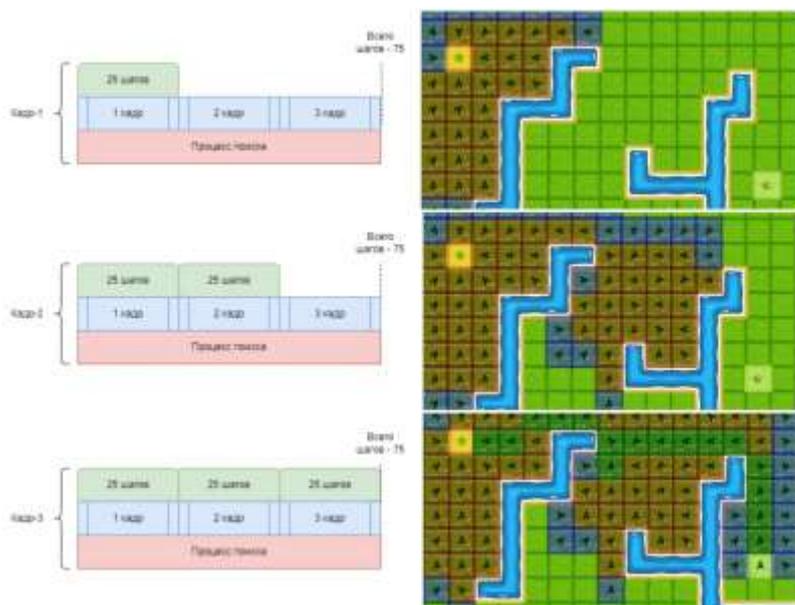


Рисунок 3 – Разбиение алгоритма на кадры

Любое такое решение влечёт за собой некоторые проблемы. Большинство оптимизаций строго привязаны к конкретным алгоритмам. И не всегда есть возможность подобрать более легковесное решение для более простых условий, что влечёт повторную разработку решения. Отметим, что эффективность алгоритмов, использующих сжатие информации на основе графов перекладывает затраты вычислительных ресурсов и времени на этап анализа карты. А это не всегда приемлемо при процедурной генерации. К тому же в местах, где, к примеру, нет препятствий, сложный анализ для обеспечения работы соответствующего алгоритма не требуется.

Это приводит к различного рода «оптимизациям» программного кода, «отключающим» анализ на основе процедурных эвристик. В условиях процедурно-генерируемых миров будет

требоваться постоянное вмешательство разработчика. Это делает систему на основе «базового универсального алгоритма» в целом менее поддерживаемой, так как для разных генерируемых миров будет требоваться разный подход к формированию эвристик, что в конечном итоге сильно усложнит разработку программных средств.

Наиболее привлекательным вариантом здесь выступает алгоритм, который мог бы динамически выбирать разные алгоритмы пути в зависимости от наличия и типа препятствий на участке местности. Данный подход имеет следующие преимущества.

1. **Уменьшение времени анализа карты и оптимизация используемых при этом ресурсов** - поскольку выбор алгоритма определяется в зависимости от конкретных условий среды, причём для отдельных участков, это может привести к улучшению производительности системы в целом за счет применения «менее требовательных» алгоритмов там, где их достаточно. Таким образом, для достаточно свободных участков, можно упростить этап анализа, в т.ч. не выделять ресурс памяти под дополнительные представления карты.

2. **Уменьшение времени нахождения пути** - использование более простых алгоритмов в условиях, где они достаточно эффективны, может значительно сократить время нахождения пути, особенно в сценариях с небольшим количеством препятствий или пустой средой.

3. **Повышение гибкости программных средств.** С одной стороны, за счёт адаптивных свойств комбинированного алгоритма, программные средства на его основе смогут эффективно работать в системах с широким диапазоном сценариев генерации карты. С другой — модульность алгоритма позволит варьировать перечень базовых алгоритмов поиска пути, добавлять новые и централизованно определять условия их применения.

### Принципы построения комбинированного алгоритма поиска пути

Введём понятие агент. Под агентом подразумевается автономная виртуальная сущность модельной (игровой) среды. Цель агента добраться из одной точки пространства в другую. Пусть в нашем случае конечной точкой будет автобус (Рисунок 4). На карте нет каких либо объектов, с которыми наш агент мог бы столкнуться. Соответственно нет необходимости в формировании дополнительных представлений для рассматриваемого участка карты. В этом случае достаточно использовать алгоритм MoveTo [3]

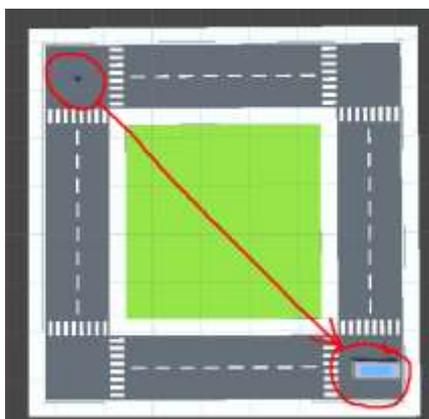


Рисунок 4 – Карта без препятствий





Рисунок 7 – Более сложная карта с препятствиями

Всё это мысленно подводит нас к тому, чтобы сконцентрироваться на выборе наилучшего алгоритма исходя из данных о препятствиях. Какие данные могут быть для нас полезны в данном случае?

1. Соотношение свободных участков карты и препятствующих. Эта информация может помочь нам не только для того, чтобы сделать выбор между алгоритмом потенциального поля и алгоритмом  $A^*$ , но также определить бюджет и тип (синхронный/асинхронный) для того, чтобы оптимизировать работу  $A^*$ .

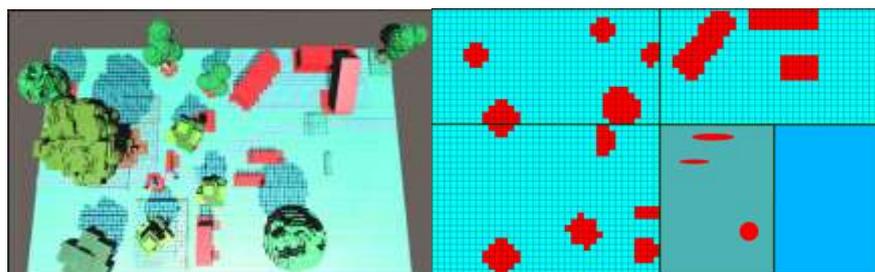


Рисунок 6 – карта препятствий и её анализ

2. Каждое препятствие также можно представить как некоторый граф [5, 6], у которого есть топологические характеристики, такие как структурная избыточность, компактность, диаметр и индекс центральности. По этим характеристикам можно выдвинуть некоторые предположения о характере препятствий.

В данном случае имеет смысл разбивать игровую карту на отдельные участки и анализировать препятствия в них. Рассмотрим, какую информацию несут топологические характеристики препятствий.

Структурная связность [7] может сказать нам о том, является ли граф замкнутым. Эта информация может быть очень важной как для выбора алгоритма, так и для выбора разрешения. Например, на Рисунке 10, разрешение для анализа участка внутри замкнутого препятствия избыточно.

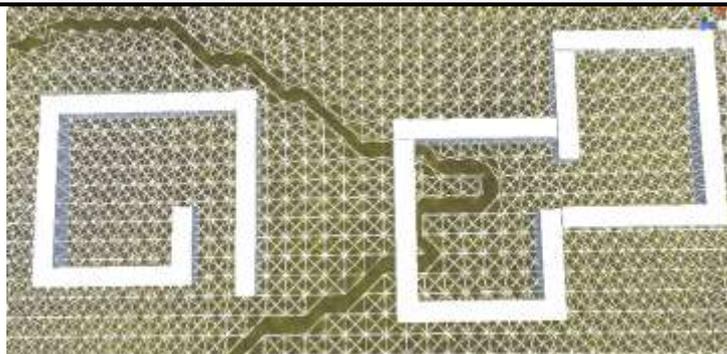


Рисунок 10 - Замкнутое и незамкнутое препятствие

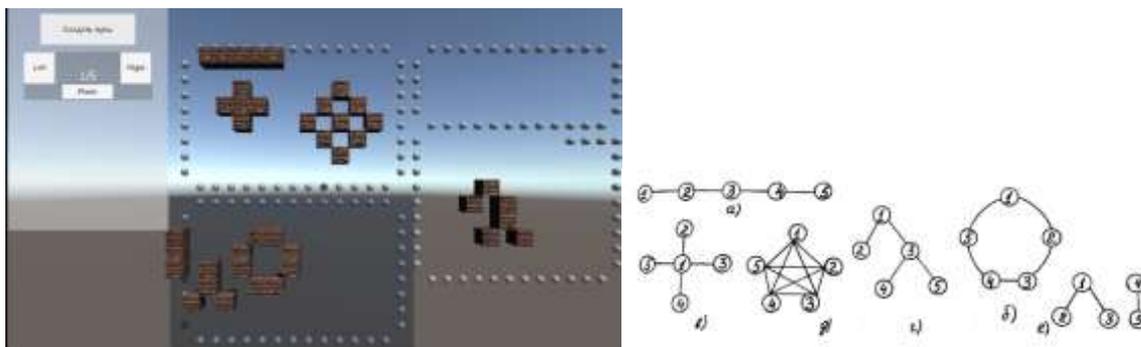


Рисунок 11 – Набор нескольких графов

Таблица 1 – Структурно-топологические характеристики, по которым можно определить вид графа

Вид структуры	Структурная избыточность		Структурная компактность Qотно	Диаметр структуры d	Индекс центральности $\alpha$
	R	$\xi^2$			
А	0	1.2	1	4	0.7
Б	0.25	0	0.5	2	0
В	0	7.2	0.6	2	1
Г	0	3.2	0.8	3	0.8
Д	1.5	0	0	1	0
Е	-0.25	-	-	-	-

Совокупность показателей также можно использовать для классификации препятствий. На рисунке 11 представлен набор препятствий и соответствующих им графов. В Таблице 1 им сопоставлены значения структурно-топологических характеристик. Наличие препятствий определённого вида может ограничивать выбор алгоритма. Например, наличие препятствия, близкого к полному графу приводит к неэффективности алгоритма потенциального поля.

### Заключение

Проблемы применения подхода на основе единого базового алгоритма для поиска пути в генерируемых средах приводят к необходимости искать более гибкие и производительные (в динамике) решения. Анализ всей карты, как при статическом подходе, оказывается уже непозволительно долгим. Кроме того, в условиях отдельных участков местности, часто ока-

зываются эффективны простые алгоритмы, ориентированные на пространства с малым количеством препятствий.

В результате предложены основные принципы построения комбинированного алгоритма — выбор своего алгоритма поиска пути из заранее определённого набора для отдельных участков карты, анализ участков карты с точки зрения количества, плотности размещения и топологии препятствий.

### Список литературы

1. Wand, Haifeng & Zhou, Jiawei & Zheng, Guifeng & Liang, Yun(2014). HAS: Hierarchical A-Star Algorithm for Big Map Navigation in Special [https://www.researchgate.net/publication/301410542\\_HAS\\_Hierarchical\\_A-Star\\_Algorithm\\_for\\_Big\\_Map\\_Navigation\\_in\\_Special\\_Areas](https://www.researchgate.net/publication/301410542_HAS_Hierarchical_A-Star_Algorithm_for_Big_Map_Navigation_in_Special_Areas) Дата публикации: 14.11.2014. Режим доступа: По запросу (дата обращения: 7.5.2024)
2. Game AI Pro – Collected Wistom of Game AI Professionals /CRC Press Publishing AG 2014 – 612 с.
3. Статья на хабре: «Чему игровой ИИ может научить робототехнику» - Режим доступа: <https://habr.com/ru/articles/349044/> Дата публикации: 16.2.2018. Режим доступа: Свободный (дата обращения: 12.12.2023)
4. Исследование алгоритма планирования пути, объединяющего оптимизационный алгоритм A-Star и метод искусственного потенциального поля: MDPI [Электронный ресурс]. – URL: <https://www.mdpi.com/2079-9292/11/22/3660> Дата публикации: 30.10.2022. Режим доступа: Свободный (дата обращения: 15.4.2024)
5. Введение в теорию графов / Уилсон Р.: Пер. с англ. - СПб. : ООО "Диалектика", 2019. - 240 с.
6. Алгоритм создания карт топологии для поиска оптимального пути для карт на основе изображений: MDPI [Электронный ресурс]. – URL: <https://www.mdpi.com/2076-3417/12/23/12436> Дата публикации: 11.10.2022. Режим доступа: Свободный (дата обращения: 15.4.2024)
7. Программирование на C, C# и Java: Структурно – топологические характеристики систем – Режим доступа: <https://vscode.ru/articles/struct-topolog-charact-system.html> Дата публикации: 19.5.2015. Режим доступа: По запросу (дата обращения: 7.5.2024)

### References

1. Wand, Haifeng & Zhou, Jiawei & Zheng, Guifeng & Liang, Yun(2014). HAS: Hierarchical A-Star Algorithm for Big Map Navigation in Special [https://www.researchgate.net/publication/301410542\\_HAS\\_Hierarchical\\_A-Star\\_Algorithm\\_for\\_Big\\_Map\\_Navigation\\_in\\_Special\\_Areas](https://www.researchgate.net/publication/301410542_HAS_Hierarchical_A-Star_Algorithm_for_Big_Map_Navigation_in_Special_Areas) Publication date: 11/14/2014. Access mode: On request (date of access: 7.5.2024)
2. Game AI Pro – Collected Wistom of Game AI Professionals /CRC Press Publishing AG 2014 – 612 с.
3. Article on Habré: “What gaming AI can teach robotics” - Access mode: <https://habr.com/ru/articles/349044/> Date of publication: 2/16/2018. Access mode: Free (access date: 12/12/2023)
4. Study of a path planning algorithm combining the A-Star optimization algorithm and the arti-

ficial potential field method: MDPI [Electronic resource]. – URL: <https://www.mdpi.com/2079-9292/11/22/3660> Date of publication: 10/30/2022. Access mode: Free (date of access: 15.4.2024)

5. Introduction to graph theory / Wilson R.: Per. from English - St. Petersburg. : Dialectics LLC, 2019. - 240 p.
  6. Algorithm for creating topology maps to find the optimal path for maps based on images: MDPI [Electronic resource]. – URL: <https://www.mdpi.com/2076-3417/12/23/12436> Date of publication: 10/11/2022. Access mode: Free (date of access: 15.4.2024)
  7. Programming in C, C# and Java: Structural - topological characteristics of systems - Access mode: <https://vscode.ru/articles/struct-topolog-charact-system.html> Date of publication: 19.5.2015. Access mode: On request (date of access: 7.5.2024)
-