

Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики

Кафедра компьютерных технологий

А. П. Ковшаров

# **Оптимизация параметров стратегий поиска объектов на море**

Бакалаврская работа

Научный руководитель: А. С. Ковалев

Санкт-Петербург  
2015

# Содержание

Содержание . . . . .	2
Введение . . . . .	3
<b>Глава 1. Постановка задачи . . . . .</b>	<b>5</b>
1.1 Задача построения маршрута поиска в общем случае . . . . .	5
1.1.1 Расширения задачи коммивояжера . . . . .	5
1.1.2 Распределение частиц . . . . .	7
1.1.3 Формулировка задачи построения маршрута поиска в общем случае . . . . .	8
1.2 Задача построения маршрута поиска стратегией “Параллель- ное галсирование” . . . . .	8
1.2.1 Стратегии поиска . . . . .	8
1.2.2 Формулировка задачи построения маршрута поиска стратегией “Параллельное галсирование” . . . . .	9

# Введение

Существующие подходы строят весь маршрут исходя из статичных данных в начальный момент времени — информации о начальном распределении и модели его распределения. Таким образом главным недостатком существующих подходов является необходимость разработки нового алгоритма для всех различных моделей изменения распределения. Учитывая тот факт, что подобрать правильную модель, хорошо приближающую реальность, крайне непросто, возникает необходимость разработки алгоритма, работающего единообразно на широком классе различных моделей. Основная идея рассматриваемого подхода — использование симулятора для получения информации о распределении в любой момент времени при построении маршрута. Таким образом при планировании пользователь в первую очередь выберет модель, как можно лучше приближающую реальность в данном случае, а после запустит единственный алгоритм.

Концепт данной задачи был продемонстрирован на одной из выставок. К задаче был проявлен интерес и было решено внедрить ее в комплекс расчетных морских задач.

В главе 1 решаемая задача будет рассмотрена более подробно. Описаны классы маршрутов, получаемые при использовании стратегии “Параллельное галсирование”. Будут приведены особенности задачи, которые отличают ее от классической задачи коммивояжера и делают невозможным использование ранее разработанных методов решения TSP для решения исходной задачи в общем случае.

В главе 2 будут рассмотрены вопросы, связанные с разработкой симулятора на CUDA. Обозначены предоставляемые им сервисы.

В главе 3 будет описан алгоритм построения маршрутов согласно

стратегии “Параллельное галсирование”.

# Глава 1. Постановка задачи

## 1.1. ЗАДАЧА ПОСТРОЕНИЯ МАРШРУТА ПОИСКА В ОБЩЕМ СЛУЧАЕ

### 1.1.1. Расширения задачи коммивояжера

В классической формулировке задача коммивояжера (traveling salesman problem, TSP) звучит так: Дан взвешенный граф необходимо найти цикл, минимального веса, посещающий все его вершины. Евклидовым TSP — называется частный случай TSP, когда весами ребер являются расстояния на плоскости. Задача построения маршрута поиска объекта на первый взгляд очень похожа на задачу коммивояжера. Однако в реальности особенности задачи оказываются существенными:

- Средство поиска в один момент времени посещает несколько вершин, а именно все вершины попадающие в круг определенного радиуса с центром в его текущем положении. Существует расширение TSP под названием GTSP или обобщенная задача коммивояжера, которое решает следующую задачу: Дан взвешенный граф и разбиение его вершин на **непересекающиеся** множества, необходимо найти цикл минимального веса, посещающий хотя бы одну вершину из каждого множества. Существует сведение данной задачи к TSP не увеличивающее размерности и доказывающее ее  $NP$ -полноту. Но к сожалению данное расширение неприменимо к нашей задаче, так как множества могут пересекаться.
- Зачастую все вершины посетить физически невозможно. Соответственно выделяются вершины в которых более вероятно обнаружить объект. Сопоставим вершине  $v$  величину  $p_v$  — вероятность обнару-

жить объект в этой вершине.  $p_{path} = \sum_{v \in path} p_v$ . На практике длина путей с  $p_{path} \geq 0.99$  может превышать длину путей с  $p_{path} \geq 0.9$  в десятки раз. То есть длина пути растет экспоненциально в зависимости от  $p_{path}$ . Следовательно необходимым параметром задачи становится максимальная длина пути (или время поиска с физической точки зрения). Известно обобщение Profit Based TSP: каждой вершине сопоставляется значение  $p_v$ , при посещении вершины к сумме призов добавляется  $p_v - t_v$ , где  $t_v$ —время посещения, необходимо составить маршрут с наибольшей суммой призов. К сожалению наша задача и здесь сравнительно более общая, так как величины призов могут изменяться нелинейно.

- Распределение  $p_v$  действительно может быть не статично по времени и изменяться согласно заданной модели. Следовательно время и расстояние не взаимозаменяемы в поставленной задаче и оптимальное значение  $p_{path}$  может быть различно если мы фиксируем один из параметров. В поставленной задаче будет фиксированно время. Однако все модели изменения обладают свойством: распределение изменяется непрерывно, перераспределяясь не превышая фиксированную скорость, не появляется извне и не исчезает (будем считать что при посещении призы “собираются” и исчезают). Соответственно  $p_{v,t} = \sum_{pos(q_i)=t} q_i$  и приз  $q_i$  в вершине  $v$  мы можем собрать лишь в какие-то промежутки времени. Обобщение Time Windows TSP решают соответствующую задачу: вершину  $v$  — можно посетить лишь во время  $[t_{v,l}; t_{v,r}]$ . Проблема сведения к этому обобщению в том, что значительно увеличивается количество вершин, в частности новая вершина будет сопоставлена  $j$ -му моменту, когда приз  $q_i$  оказался в вершине  $v$ .

- Другое обобщение для случая с движущимися вершинами — Kinetic TSP или Moving Targets TSP. Это классическое евклидово TSP, в котором вершины движутся в фиксированном направлении. Однако ранее были рассмотрены только случаи где вершины движутся с фиксированной скоростью или коммивояжер должны возвращаться в стартовую вершину после посещения каждой. В исходной задаче скорость и направления перемещения призов могут изменяться. Кроме того сведение вновь значительно увеличивает число вершин.
- Фактически до текущего момента времени задача формулировалась в непрерывном случае. При разработке алгоритма для ЭВМ необходимо провести некоторую дискретизацию. Необходимое количество вершин для двумерной задачи растет квадратично и соответственно приемлемое значение может достигать миллионов вершин. Такое количество вершин велико даже для самых быстрых реализаций приближенных решений TSP.
- В реальной жизни невозможно искать согласно произвольной траектории из-за технических ограничений на передвижение поискового средства. Помимо этого недопустимо применение не общепризнанных, зарекомендовавших себя стратегий поиска.

### 1.1.2. Распределение частиц

Использование понятия распределение вероятности обнаружения противника в дальнейшем будет не совсем удобно, так как одна из основных частей задачи “сбор” вероятности — не является корректной операцией над вероятностями.

Введем альтернативное понятие — распределение “частиц”. Частица  $\pi : \Pi$  — гипотеза изначального расположения объекта и его поведения в дальнейшем. Вес частицы  $w_\pi$  — вероятность осуществления именно этой

гипотезы.  $pos(\pi, t)$  — положение частицы  $\pi$  в момент времени  $t$ . Сумма весов всех частиц изначально равна единице. Расширение понятия веса частиц на непрерывный случай аналогично расширению для вероятностного пространства. “Собрать частицу” — проверить гипотезу. После проверки частица исчезает и больше не может быть собрана. Распределение частиц  $f(dS, t, path)$  — функция, ставящая в соответствие области и времени сумму весов частиц находящихся в этой области в заданное время, с учетом частиц собранных средством поиска к данному моменту. С течением времени частицы могут перемещаться в любом направлении с ограниченной скоростью, однако не могут появляться из ниоткуда или исчезать иным способом, кроме сбора их средством поиска. Сумма собранных частиц фактически равна априорной вероятности обнаружить объект, имеющий заданное начальное распределение вероятности и закон его изменения.

### 1.1.3. Формулировка задачи построения маршрута поиска в общем случае

Задано распределение частиц  $f(dS, t, path)$ ,  $t_{search}$  — время поиска и параметры средства поиска:  $pos_0$  — начальная позиция,  $r$  — радиус обнаружения и  $v_{max}$  — максимальная скорость передвижения. Необходимо найти маршрут  $path(t)$  — определяющий позицию средства поиска в любой момент времени, максимизирующий сумму весов собранных частиц  $sum_{res}$ . Более формально  $sum_{res} = 1 - \int_S f(dS, t_{search}, path)$ ,  $path(0) = pos_0$ ,  $|path'(t)| \leq v_{max}$ , собраны те и только те частицы  $\pi$  для которых  $\exists \tau \leq t_{search} ||pos(\pi, \tau) - path(\tau)|| \leq r$ .

## 1.2. ЗАДАЧА ПОСТРОЕНИЯ МАРШРУТА ПОИСКА СТРАТЕГИЕЙ “ПАРАЛЛЕЛЬНОЕ ГАЛСИРОВАНИЕ”

### 1.2.1. Стратегии поиска



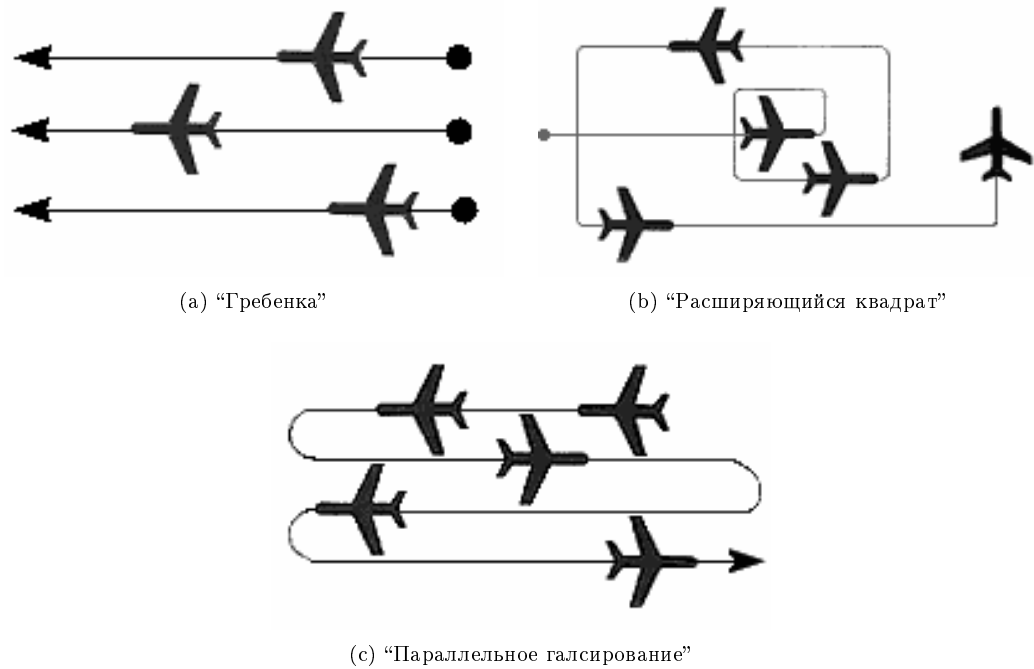


Рис. 1.1: Стратегии поиска

### 1.2.2. Формулировка задачи построения маршрута поиска стратегией "Параллельное галсирование"