УДК 004.045:004.942

ИСПОЛЬЗОВАНИЕ МЕТОДОВ ПОИСКА ОПТИМАЛЬНОГО ПУТИ ПРИ ПОСТРОЕНИИ МАРШРУТА ДЛЯ ПРОВЕДЕНИЯ МОНИТОРИНГА ВОДНОЙ СРЕДЫ

Сафонов М.Д., Светличная В.А.
Донецкий национальный технический университет г.Донецк
кафедра автоматизированных систем управления
e-mail: msafonovmail@gmail.com

Аннотация

Сафонов М.Д., Светличная В.А. Использование методов поиска оптимального пути при построении маршрута для проведения мониторинга водной среды. *В данной статье описывается реализация методов для определения наилучшего пути исследовательских суден, проводящих измерения параметров водной среды, с помощью генетического алгоритма, метода ветвей и границ и муравьиного алгоритма*

Постановка задачи.

Основной задачей исследований водной среды является изучение пространственно-временной изменчивости параметров океана. Изменчивость характеристик водной среды в пространстве изучается с помощью различной зондирующей и буксируемой аппаратуры, устанавливаемой на надводных и подводных кораблях, подводных аппаратах.[1].

Средства и методы исследования океана являются одной из быстро развивающихся отраслей современной океанологии. С одной стороны, это вызвано потребностями самой океанологической науки. С другой стороны, ряд важных и проблемных прикладных народно-хозяйственных задач стимулирует разработку различной аппаратуры.

Для дальнейшего изучения и освоения океана, необходимо регулярно получать данные о его структуре и состоянии. Вместе с тем измерение параметров характеризующих состояние водной среды довольно трудная и дорогая операция. Поэтому одной из основных задач является построение наиболее эффективных алгоритмов и методов для измерения параметров водной среды.

Постановка задач исследования.

Экспедиционные измерения гидрофизических параметров - дорогостоящая операция. Одним из слабых мест систем, производящих измерения, является не оптимальный маршрут при экспедиционных исследованиях. Использование при построении маршрута корабля опорных точек, позволит уменьшить затраты на измерения.

Поэтому необходимо проводить измерения по возможности оптимальным путем. Исходя из этого, одной из основных целей информационных измерительных систем является - оптимизация маршрута экспедиционного корабля. На данный момент для экспедиционных исследований задана акватория разбивается на квадраты. Корабль проходит поочередно все квадраты акватории. Примерная схема маршрута изображена на рисунке 1.

Таким образом, экспедиционное судно получает наиболее полные данные о заданной области. Однако они обычно количество данных является избыточным.

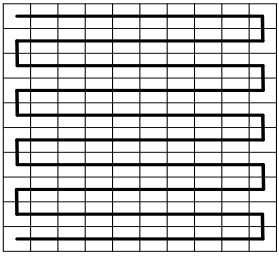

Рисунок 1. - Маршрут экспедиционного корабля

Функция изменения измеряемого параметра обладает экстремумами в заданной области. То есть для данного параметра можно найти несколько максимальных и минимальных значений.[2,3]

Набор значений измеренных между этими точками, дает представление о градиенте изменения исследуемого параметра в данной области. Из этого следует, что выполнив измерение значения параметров в точках, соответствующих экстремумам, а также между ними, возможно, сформировать полную картину о распределении заданного параметра в акватории. Исходя из этого, возможно определенным образом оптимизировать маршрут экспедиционного судна, то есть производить измерения не в каждом квадрате, а в квадратах, соответствующих расположению опорных точек.. Вопрос нахождения опорных точек, который рассматривался в [], является первой частью общей задачи. Определение наиболее рационального маршрута для обхода опорных точек представляет ее вторую часть.

Решение задачи и результаты исследований.

Задачу поиска оптимального пути по опорным точкам можно формализовать как задачу коммивояжера, которая является одной из самых известных задач комбинаторной оптимизации. Задача заключается в отыскании самого выгодного маршрута, проходящего через указанные города хотя бы по одному разу с последующим возвратом в исходный город. В условиях задачи указываются критерий выгодности маршрута (кратчайший, самый дешевый, совокупный критерий и т.п.) и соответствующие матрицы расстояний, стоимости и т.п. Как правило, указывается, что маршрут должен проходить через каждый город только один раз - в таком случае выбор осуществляется среди гамильтоновых циклов. Для рассматриваемой задачи построение наиболее оптимального пути корабля с измерительной системой выполняется по стоимости.

Существует несколько частных случаев общей постановки задачи, в частности геометрическая задача коммивояжера (также называемая планарной или евклидовой, когда матрица расстояний

отражает расстояния между точками на плоскости), треугольная задача коммивояжера (когда на матрице стоимостей выполняется неравенство треугольника), симметричная и асимметричная задачи коммивояжера. Также существует обобщение задачи, так называемая обобщенная задача коммивояжера [4].

Общая постановка задачи, как и большинство ее частных случаев, относится к классу NP-Сложных задач. Все эффективные (сокращающие полный перебор) методы решения задачи коммивояжера - методы эвристические.

В большинстве эвристических методов находится не самый эффективный маршрут, а приближенное решение. Чаще востребованы так называемые any-time алгоритмы, то есть постепенно улучшающие некоторое текущее приближенное решение.

Существует также постановки, в которых задача коммивояжера становится NP-Полной, Обычно такие постановки выглядят следующим образом: существует на заданном графе B такой обход, что его стоимость не превышает x. Часто на ней проводят обкатку новых подходов к эвристическому сокращению полного перебора. На практике используются различные модификации наиболее эффективных методов: метод ветвей и границ, метод генетических алгоритмов, а также метод муравьиных колоний.

Для поиска оптимального маршрута было рассмотрены три метода: ветвей и границ, генетический алгоритм и алгоритм муравьиной колонии. Для определения наиболее подходящего для данной задачи метода были реализованы все. Представим описание объектных моделей данных методов.

Метод ветвей и границ

При реализации метода ветвей и границ были использованы следующие классы:

1) RouteAlgorithm - интерфейс определяющий функциональность класса реализующего построение оптимального маршрута.

2) BranchBordersAlgorithm – основной класс модели реализующий интерфейс RouteAlgorithm. В данном классе заполняется матрица стоимостей переходов между узлами и производится поиск наиболее оптимального маршрута.

3) Subset – класс определяющий подмножество. В данном классе реализована редукция матрицы стимостей, поиск текущей наиболее выгодной дуги, а также создание новых матриц. Каждое подмножество ссылается на своего родителя, для возможности возврата вверх по графу и проверки на наличие более оптимального решения.

4) Route - класс определяющий маршрут. Хранит в себе последовательность подмножеств образованных в ходе выполнения алгоритма. Выполняет преобразование набора подмножеств в маршрут, а также проверку полученного маршрута на незамкнутость.

5)

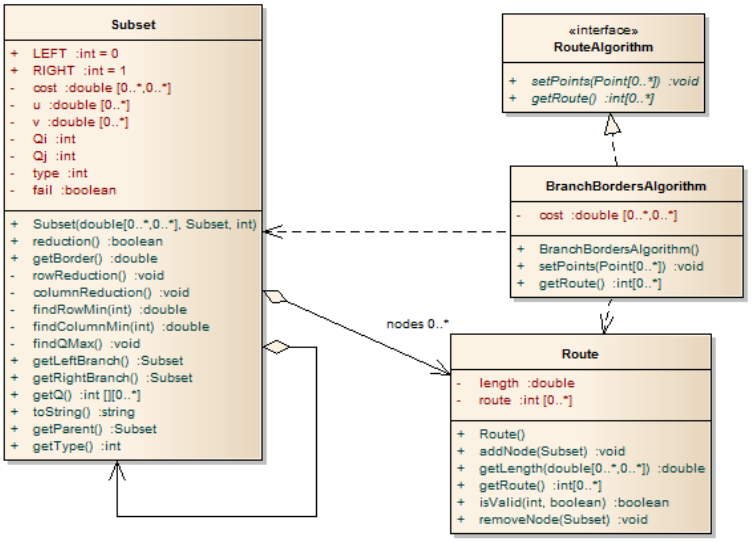

Рисунок 2 – Диаграмма классов метода ветвей и границ

Генетический алгоритм.

Для определения наиболее подходящего метода было реализовано три представления хромосомы в генетическом алгоритме: путевое, соседство и порядковое. Для каждого представления были реализованы свои варианты кроссинговера. Рассмотрим классы созданные при реализации генетического алгоритма:

1) RouteAlgorithm - интерфейс определяющий функциональность класса реализующего построение оптимального маршрута.

2) GenAlgorithm – класс реализующий интерфейс RouteAlgorithm. Производит инициализацию системы, а также пошаговое выполнение эволюционного алгоритма до выполнения условия останова. Останов алгоритма происходит , когда разница между средним значением длины маршрута и минимальным значением не превышает заданный порог.

3) Population – класс реализующий популяцию. Хранит в себе набор хромосом заданного типа. Реализует операции репродукции, скрещивания, мутации и отсева хромосом.

4) Chromosome - интерфейс определяющий функциональность хромосомы при решении задачи коммивожера.

5) NeighbourViewChromosome – абстрактный класс реализующий интерфейс Chromosome. Реализует представление соседство. Генерирует хромосому в таком представлении, проверяет на корректность, а также преобразовывает хромосому в список узлов. В зависимости от операции кроссинговера расширяется двумя классами – HeuristicChromosome, AlternatingEdgesChromosome.

6) SerialViewChromosome –класс реализующий интерфейс Chromosome. Реализует порядковое представление. Генерирует хромосому в таком представлении, проверяет на корректность, а также преобразовывает хромосому в списке узлов.

7) RouteViewChromosome – абстрактный класс реализующий интерфейс Chromosome. Реализует путевое

представление . Генерирует хромосому в таком представлении, проверяет на корректность.В зависимости от операции кроссинговера расширяется тремя классами – PartiallyMappedChromosome, CircleChromosome, OrderChromosome.

8) ChromosomeFactory – класс-фабрика. Создает объект хромосома заданного типа.

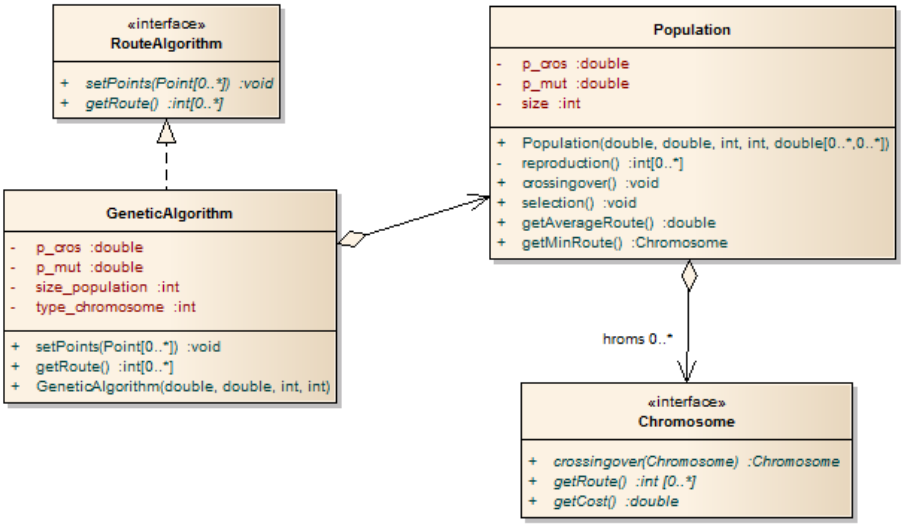

Рисунок 3 – Диаграмма классов генетического алгоритма

Муравьиный алгоритм

В ходе реализации муравьиного алгоритма для решения задачи построения оптимального маршрута были разработаны следующие классы:

1) RouteAlgorithm - интерфейс определяющий функциональность класса реализующего построение оптимального маршрута.

2) AntAlgorithm – класс реализующий интерфейс RouteAlgorithm. Класс инициализирует систему построения оптимального маршрута. Также класс пошагово выполняет

прохождение муравьями выбранных маршрутов и обновление количества феромона.

3) Ant – класс реализующий функции муравья. Хранит список пройденных узлов, а также определяет вероятностным способом следующую дугу по которой муравей продолжит маршрут. По окончании пути муравей оставляет след феромона на всех пройденных дугах пропорциональный силе феромона и обратно пропорциональный длине пройденного маршрута .

4) Arc - представляет одну из дуг соединяющих два узла. Дуга хранит свою длину, а таже количество феромона. На каждом шаге количество феромона уменьшается в соотвествии с указанным коэффициентом.

5) Node - класс реализующий узел маршрута. Содержит список дуг по которым можно выйти из города, а также список муравьев которые находятся на данном шаге в том узле.

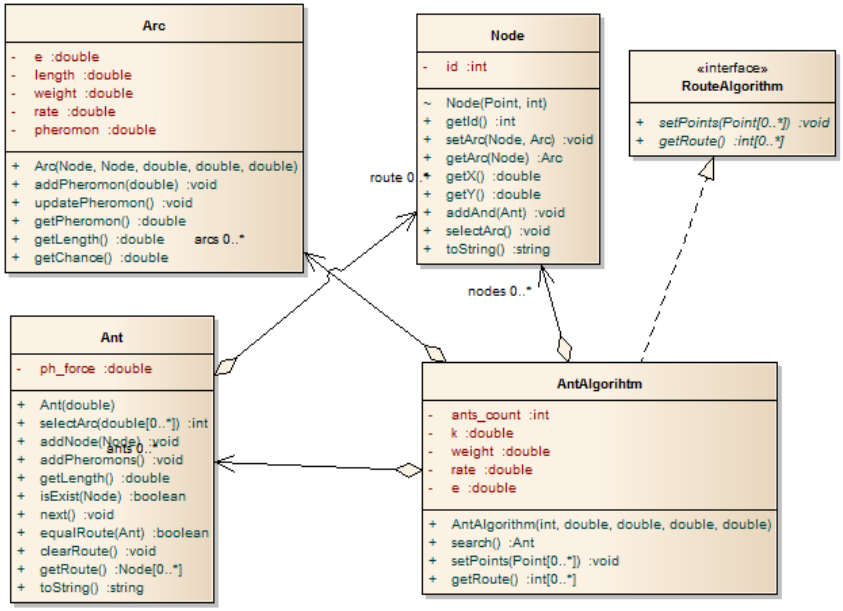

Рисунок 4 – Диаграмма классов муравьиного алгоритма

Выводы.

В результате были рассмотрены три алгоритма поиска оптимального маршрута - генетический алгоритм, метод ветвей и границ и муравьиный алгоритм. При проверке алгоритмов на разных вариантах расположения опорных точек представляется получать разные предполагаемые оптимальные маршруты.

Список литературы

1. Зори А.А., Коренев В.Д., Хламов М.Г. Методы, средства, системы измерения и контроля параметров водных сред.- Донецк: РИА ДонГТУ, 2000. – 388с.

2. Чесноков В. І., Світлична В. А., Савкова О. Й. Визначення опорних точок у процесі планування моніторингу водяного середовища. - Міжвідомчий збірник наукових праць «Відбір і обробка інформації » Випуск 33(109), 2011 – с. 61-67

3. Савкова Е.О., Светличная В.А. Объектно-эволюционная модель системы мониторинга гидрофизических полей водной среды. - Наукові праці Донецького національного університету Серія: Обчислювальна техніка та автоматизація» Випуск 171(19) – Донецьк:ДонНТУ, 2010 – с.224.

4. Новиков Ф.А Дискретная математика для программистов –С.Петербург изд. Дом. «Питер».2002 – 304с.