

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра социальной информатики

**ИСПОЛЬЗОВАНИЕ ТЕОРИИ ЭВОЛЮЦИОННЫХ
ВЫЧИСЛЕНИЙ ДЛЯ РЕШЕНИЯ НЕКОТОРЫХ ЗАДАЧ
ОПТИМИЗАЦИИ**

(автореферат бакалаврской работы)

студента 4 курса 451 группы
направления 09.03.03 - Прикладная информатика
профиль Прикладная информатика в социологии
Социологического факультета
Теселкина Данила Игоревича

Научный руководитель

Кандидат физ.-математических наук, доцент _____ М.Г. Плешаков
подпись, дата

Зав. кафедрой

кандидат социологических наук, доцент _____ И.Г. Малинский
подпись, дата

Саратов 2021

ВВЕДЕНИЕ

Целью моей работы является, написание рабочей программы оптимизации для поиска оптимального решения конкретной задачи о нахождении локального оптимума при помощи эволюционных вычислений.

Любой человек, который занимался или занимается программированием знаком с понятием алгоритма. Если перед нами стоит задача, то мы вводим чёткий набор инструкций для решения этой самой задачи. Но что делать, если перед нами слишком сложная задача, или её решение не определено или не понятно?

В таком случае как правило используется искусственный интеллект. Наиболее простая из известных техник, которые можно использовать для решения подобного вида задач — эволюционные вычисления. Данная техника, основывается на идее применения дарвиновской теории эволюции к современной компьютерной программе.

Существует такое понятие как, дарвиновская концепция выживания наиболее приспособленных. Основная идея максимально проста и соответствует названию: выживание — сложная задача, и выжившие существа имеют более высокий шанс на размножение. Окружающая среда естественным образом отбирает существ, которые наиболее к ней приспособлены, избавляясь от тех, кто не справится с этой задачей.

Такой же способ можно применить к решению формальных задач оптимизаций, которые реализуются с помощью компьютерных программ, если есть приближённое решение задачи, то мы можем, модифицируя его случайным образом, в соответствии с нашими целями, циклически проверять улучшает ли оно указанное решение. Если мы целенаправленно повторяем этот процесс, то в результате можем получать лучшее или такое же решение. В то время как естественный отбор не зависит от поставленной задачи, направленный отбор обусловлен именно поставленной задачей.

Задачей оптимизации в математике называется задача о нахождении экстремума (минимума или максимума) вещественной функции в некоторой

области. Как правило, рассматриваются области, принадлежащие R^a и заданные набором равенств и неравенств. Функция, связывающая цель с управляемыми переменными в задаче оптимизации. В широком смысле целевая функция есть математическое выражение некоторого критерия качества одного объекта в сравнении с другими.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Механизмы эволюции.

Мутация. Мутационная изменчивость – это процесс изменения генетической структуры организма, его генотипа. При этом изменяется число хромосом, или их строение. Как и комбинативная изменчивость, мутационная – процесс ненаправленный, признаки могут изменяться случайным образом.

Сама мутация – это генетическое изменение, приводящее к качественно новому проявлению основных свойств генетического материала. Считается, что при мутации происходят внезапные скачкообразные изменения в структуре генотипа. В естественных системах роль мутаций заключается в том, что именно они генерируют новые функции, затем происходит дупликация, закрепляющая обе функции, а далее начинается отдельная эволюция исходной и новой функции.

Отбор. Выделяют два основных процесса отбора: искусственный отбор и естественный отбор.

Искусственный отбор или селекция – представляет собой форму отбора, где в отличие от естественного отбора эволюция направляется факторами внешней среды. Селекция, как наука, создана Ч. Дарвином, который выделял три формы отбора:

- Естественный отбор, вызывающий изменения, связанные с приспособлением к новым условиям;
- Бессознательный отбор, при котором в процессе эволюции сохраняют лучшие экземпляры;
- Методический отбор, при котором проводится целенаправленное изменение популяций в сторону установленного идеала.

Естественный отбор – это процесс, направленный к повышению или понижению вероятности оставления потомства одной формой организмов по сравнению с другими. Отбор, прежде всего, действует в пределах каждой популяции, отбирая те или иные входящие в ее состав генотипы.

При относительной стабильности внешних условий преобладающие генотипы все время будут сохранять свое доминирующее положение. Напротив, все отклонения от этой группы будут уничтожаться. Такая форма отбора названа центростремительным или стабилизирующим отбором. Однако при изменении условий существования может произойти отбор, ведущий к замене одних количественно преобладающих генотипов другими. Эта форма отбора названа движущей, или ведущей.

Естественный отбор – единственный направляющий эволюцию элементарный фактор. Его действие всегда направляется складывающимися условиями существования

Изменчивость – это способность организма приобретать новые признаки в процессе жизни. Изменчивость – одно из самых загадочных свойств жизни. В результате изменчивости в процессе эволюции возникают разные формы организмов, которые могут сосуществовать на одном и том же пространстве.

Модификации характеризуются массовым характером, не затрагивают генотип, не передаются по наследству, адекватны вызывающему фактору и кратковременны (нестойкие). Групповой изменчивости подвержены такие признаки, как рост животных и растений, их масса, окраска, жирность молока и другие признаки, т.е. количественные признаки с широкой нормой реакции. Возникновение модификационных изменений связано с тем, что условия среды воздействуют на ферментативные реакции, протекающие в развивающемся организме, и изменяют их течение.

Методы оптимизации

Под оптимизационной задачей понимается задача, в которой необходимо найти решение, в некотором смысле наилучшее или оптимальное. Для того чтобы найти оптимальное решение некоторой задачи часто используется

разбиение множества решений на подмножества с помощью дерева решений, на основе матрицы, графа или списка. В качестве исходной вершины дерева решений принимается множество Z . После этого вычисляем нижнюю границу $\Theta(z)$ множества Z . Если найдём такое решение t , что $f(t) = \Theta(z)$, то t – искомое оптимальное решение. Если найти его не удалось, то множество Z разбиваем на ряд подмножеств.

Основной способ применения теории эволюции в программировании назвали – генетическим алгоритмом. Последовательность управляющих действий и операций, моделирующая эволюционные процессы на основе аналогов механизмов генетического наследования и естественного отбора.

Генетический алгоритм представляет собой адаптивный поисковый метод, который основан на селекции лучших элементов в популяции, подобно эволюционной теории Дарвина. Состоит он из нескольких этапов:

- Создание начальной популяции. На этом шаге мы вводим набор параметров нашей задачи.
- Скрещивание/мутация. Происходит скрещивание наших особей.
- Селекция. Отбор самых оптимальных особей.
- Формирование нового поколения. Если ответ нас не устроил, то мы добавляем новые особи, либо программа сама добавляет их.
- Окончательная особь (наш ответ). Оптимальный ответ не всегда является окончательным, но он является самым приближённым к эталону.

Генетическое программирование. Исходная популяция хромосом в генетическом программировании образуется стохастически. Структуры генетического программирования, как правило, имеют древовидную форму.

Основными этапами генетического программирования являются:

1. **Инициализация.** На этом этапе стохастически генерируется популяция P , состоящая из N_p древовидных программ, причем корневой вершиной

дерева всегда является функция, аргументы которой выбираются случайно из множества функций.

2. **Оценка решений.** На этом этапе оценивается значение целевой функции каждой программы. Так как программы выбраны случайно, то для большинства из них значение целевой функции будут отличаться от лучшего решения, то есть для оценки можно взять разницу между лучшим и худшим значением целевой функции в популяции.
3. **Генерация новой популяции.** Этот этап принято разделять на следующие подэтапы.
 - 3.1. Выбор операторов генетического программирования (репродукция или кроссинговер).
 - 3.2. Селекция и рекомбинация.
 - 3.3. Образование новой популяции.
4. **Проверка критерия остановки.** Процедура генетического программирования является итерационной, и критерии её остановки аналогичны критериям для обычных генетических алгоритмов.

Решение конкретной задачи оптимизации

Практическое задание заключалось в написании рабочей программы, которая использует эволюционные вычисления. Мною была поставлена цель, написать такой программный код, который позволит определить наиболее подходящее место для построения опреснительной станции на полуострове – Крым. Карта Крыма была представлена в виде дискретного рабочего поля с динамичным размером.

Начальной популяцией в нашем случае являются:

- Населённые пункты. Сёла, города и т.д. у которых нет подачи питьевой воды.
- Препятствия. Неровность ландшафта, заповедники, исторически важные объекты и т.д.

- Водоём. Моря из которых будет браться солёная вода и опресняться.
- Опреснительная станция. Необходимая нам точка (объект).
- Координаты. Точки, которые прошли цикл, но уступили в оптимальности итоговой координате.

В ходе своей работы, программа должна выполнять следующие действия:

- Находить наиболее отдалённую точку от препятствия.
- Находить наиболее приближённую точку к населённому пункту.
- Выводить итоговый результат в виде матрицы в консоли, если размер дискретного рабочего поля меньше или равно 30.
- Выводить итоговой результат в виде объёмной модели, если размер дискретного рабочего поля больше 30.
- Отвечать на запросы пользователя.
- Прекращать работу если пользователь ввёл неверные данные.
- Производить подсчёт времени работы программного кода.

Для создания программы был выбран язык программирования высокого уровня C++ и среда разработки Microsoft Visual Studio 2019. Язык C++ — это компилируемый статически типизированный язык программирования, обладающей высокой функциональностью. Кроме того, он обладает высоким быстродействием, что особенно важно. Программы, написанные на C++ в несколько раз быстрее точно таких же программ, написанных на других языках. Это объясняется тем, что код C++ не должен интерпретироваться во время исполнения, в отличие от большинства других языков. Это возможно за счет компиляции кода.

После того как написан наш код, происходит его компиляция. Если программа запущена успешно, то на экран выводится консоль. На этой консоли пользователю предлагается выбор способа работы программы.

Всего есть два способа работы программы. Отличаются они тем, как будут внесены начальные данные.

При выборе первого способа, генерация объектов происходит случайным образом, пользователь лишь выбирает количество популяции и размер рабочего поля. Этот вариант нужен для того чтобы, проверить работоспособность программы, когда условия работы не определены конкретно. В отличии от 2 пункта, размер плоскости может достигать 1000 клеток. Если введён размер поля > 1000 , то для обработки запроса программе потребуется гораздо больше времени.

Важно заметить, размер дискретного рабочего поля равен N^2 . Количество городов и препятствий не может быть больше рабочего поля. Например, пользователь вводит размер дискретного рабочего поля N^2 , а количество городов и препятствий $(N+1)^2$, в таком случае программа заикнется, не выдаст никакого ответа и в конечном итоге прекратит свою работу. Также работа прекратится если пользователь ввёл не «1» и не «2» на этапе выбора режима.

Если все данные введённые пользователем корректны, то на экран выводится модель с обозначениями популяций. Каждой популяции присвоен свой цвет. Белые точки – это координаты которые не участвовали в ходе работы программы, зелёные точки – это населённые пункты, красные точки – это препятствия, синие точки – это координаты которые участвовали в отборе, но не прошли его, голубые точки – это водоём из которого наша будущая станция будет брать воду на опреснение, розовая точка – это наш конечный результат, удовлетворяющая нашим условиям координата.

Помимо макета, ответ также выводится в консоли, но в виде матрицы. В этой консоли мы видим координаты оптимальной точки, что является результатом работы программы. Помимо нашей точки, можно увидеть расстояния от точки до города и до препятствия. А также мы видим время, за которое была выполнена работа программы. Изначально все фигуры, обозначающие наши точки, закодированы символами. Каждому символу мы

сопоставляем определённое число, это нужно так как не все функции могут работать с символами, а с числами работать могут все.

Когда пользователь выбирает 2 способ, нажатием клавиши «2», не предоставляется выбор рабочего поля и количества объектов на нём, так как все данные для этого способа хранятся в отдельном файле. Для работоспособности этого файла, его, как и любую другую библиотеку, необходимо подключить к основному коду. В этом файле хранится вся необходимая информация, которая нужна для работы программы, в нём мы сами прописываем все условия. Такой способ используется для конкретно поставленной задачи, с конкретными условиями. В этот файл мы вносим информацию о размерности рабочего поля и количестве объектов, участвующих в ходе работы программы. Самое главное отличие, что мы сами расставляем все координаты на поверхности.

Если предварительные работы с первым способом прошли успешно, то можно начинать работать вторым способом. Данный способ более приближён к реальным условиям, из-за того, что мы сами определяем входные данные. Поэтому этот способ применим к любому объекту земли.

Недостаток этого способа заключается в том, что все координаты необходимо вносить вручную, и это может занять много времени. Изучение поверхности, вбивать точное расположение каждого населённого пункта, каждого препятствия, будь то неровный рельеф или исторически важный объект. Но благодаря этому способу, можно с точностью определять какая из всех возможных координат является наилучшей, является эталоном.

Сам файл представляет собой текстовый документ. В нём находятся только цифры, которые являются нашими начальной популяцией. Изучим этот документ подробнее:

- Первая строчка отвечает за размер дискретного рабочего поля. В нашем случае размер поля будет 12^2 .
- Следующая строчка – это количество препятствий. Мы внимательно изучаем местность, которую представили в виде плоскости, и расставляем на ней все встречающиеся препятствия. Их может быть

неограниченное количество, главное, чтобы количество точек было не больше чем размер самого рабочего поля. У нас всего 4 препятствия, и следующие 4 строчки – это их координаты.

- Строчка с цифрой 3 – отвечает за количество городов. Города имеют такие же параметры, как и препятствия.
- Последняя строчка отвечает за количество координат, которые будут участвовать в итерации. В отличие от предыдущих объектов, количество этих точек должно быть меньше чем сумма городов и препятствий. Это условие необходимо нам потому что, опреснительная станция, которую мы хотим построить, не может ставиться на населённых пунктах и красных зонах.
- Координаты задаются сначала по y, а потом по x. Отсчёт системы координат начинается с нуля, из левого верхнего угла. Система координат такая, из-за того, что аппаратная реализация именно такой СК гораздо легче привычной.

Работает второй способ точно так же, как и первый. Когда все значения введены, выводится плоскость со всеми координатами. Выводится консоль с расстоянием от точки до объектов и время работы программы.

ЗАКЛЮЧЕНИЕ

Человек нередко сталкивается с настолько сложными задачами, что решить их общеизвестными путями оказывается практически нереально. А существуют ли вообще способы решения такого рода проблем?

В ходе своей работы я изучил литературу, связанную с эволюционными вычислениями. Исходя из прочитанных мною источников, я сделал вывод, что выбранная мною тема является очень актуальной и прогрессирующей.

Эволюционные вычисления широко применяются в программировании, именно они позволяют найти решения, которые неочевидны для нашего мышления. Большинство задач разрешимости – трудные. С точки зрения современного состояния науки - это означает, что оптимальное решение в

общем случае, невозможно найти за разумное время никакими методами. Эволюционные алгоритмы позволяют приблизиться к оптимальному решению.

Сфера применения эволюционных вычислений очень обширна – они полезны везде, где нужно что-нибудь оптимизировать, и используются как в промышленности, например, в промышленном дизайне, логистике и управлении, так и в других областях науки: в биоинформатике, робототехнике, машинном обучении. Активным направлением в рамках эволюционных вычислений является так называемая поисковая инженерия программного обеспечения, в которой решаются вопросы автоматической генерации, изменения и тестирования программного кода.

Исходя из приведённого обзора на эволюционные вычисления можно сделать заключение, что генетические алгоритмы являются мощной стратегией выхода из локальных оптимумов. Эта стратегия заключается в параллельной обработке множества альтернативных решений, концентрируя поиск на наиболее перспективных из них.

Генетические алгоритмы и генетическое программирование являются основными формами эволюционного поиска. Каждая из этих форм имеет отличительные черты и особенности. С практической точки зрения необходимо установить, для каких конкретно инженерных задач лучше подходит та или иная форма моделирования эволюции.