МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра социальной информатики

МАТЕМАТИЧЕСКИЕ МЕТОДЫ РЕШЕНИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ ЭКОНОМИКИ И СОЦИОЛОГИИ ТРУДА

(автореферат бакалаврской работы)

студента 5 курса 531 группы
направления 09.03.03 - Прикладная информатика
профиль Прикладная информатика в социологии
Социологического факультета
Елжова Даниила Александровича

Научный руководитель
доцент, кандидат физико-математических наук, доцент

_______________ М.Г.Плешаков
подпись, дата

Зав. кафедрой
кандидат социологических наук, доцент _______________ И.Г. Малинский
подпись, дата

Саратов 2019

ВВЕДЕНИЕ

Актуальность работы. Российский математик 19 века П. Л. Чебышев говорил, что «особую важность имеют те методы науки, которые позволяют решать задачу, общую для всей практической деятельности человека – как располагать своими средствами для достижения наибольшей выгоды».

Большую часть своих сил каждый из нас тратит на поиск наилучшего или другими словами оптимального решения поставленной задачи. Мы ежедневно отвечаем на непростые вопросы, как при наименьших затратах, достичь наилучших результатов – высокого жизненного уровня, максимальной прибыли, минимальных затрат времени.

Обширный класс задач экономики и социологии труда сводится при их экономико-математическом моделировании к линейным и нелинейным оптимизационным моделям.

Усовершенствование современных средств и методов решения задач оптимизации помогают все более широко использовать их для анализа разнообразного типа данных и решения задач, связанных с оптимизацией в различных областях деятельности человека. Не стала исключением и область экономики и социологии труда.

Многие задачи, с которыми приходится встречаться в повседневной практике, являются многовариантными. Среди большого количества возможных вариантов в условиях рыночных отношений имеет смысл отыскивать наилучшие решения при ограничениях, накладываемых на природные, экономические и технологические ресурсы. Поэтому появляется необходимость использовать для анализа задач экономики и социологии труда математические методы и современную вычислительную технику.

Одним из привлекательных методов являются генетические алгоритмы, которые в отличие от классических методов позволяют даже при условиях неопределенности находить близкие к оптимальным решения. Многие оптимизационные задачи на практике имеют большую размерность, что обеспечивает долгое решение этих задач. Однако не все из них требуют максимальную точность, в этом случае генетические алгоритмы показывают себя на высоком уровне.

Цель выпускной квалификационной работы является — показать возможность генетических алгоритмов решать оптимизационные задачи экономики и социологии труда, а также их преимущество в скорости, если методы линейного программирования и перебора решают задачу недостаточно быстро.

Задачами поставлены: решение транспортной задачи вручную методом потенциалов, симплекс-методом в программе Excel, а также при помощи генетического алгоритма с и без использования штрафных функций, чтобы убедиться в правильности решения задачи генетическим алгоритмом; решение задачи коммивояжёра методом перебора и с помощью генетического алгоритма, убедиться, что на больших объемах задачи генетический алгоритм показывает себя гораздо эффективнее.

Данная работа раскрывает основные особенности генетических алгоритмов, в ней рассказывается об операторах генетических алгоритмов и их видах. Рассматривается постановка транспортной задачи, а также её решение разными способами. После чего речь заходит и о задаче коммивояжера, где объясняется, почему на больших размерностях при допустимости не самого точного решения генетические алгоритмы показывают свою актуальность. В конце работы приведён соответствующий решаемым задачам код в программе Matlab.

Структура выпускной квалификационной работы. Она содержит введение, 3 главы, заключение, список использованных источников и приложения.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

В первой главе «Генетические алгоритмы» описываются основные моменты связанные с генетическими алгоритмами. Основополагающим трудом в области исследования генетических алгоритмов является книга «Адаптация в естественных и искусственных системах», написанная в 1975 году Джоном Холландом, который считается основателем генетических алгоритмов. Их следует отнести к эвристическим алгоритмам поиска, которые комбинируя и подбирая случайным образом различные вариации искомых параметров, используются для решения задач оптимизации. В процессе работы задействуются механизмы подобные биологической эволюции. Генетические алгоритмы относят к разновидности эволюционных вычислений. Особенностью, отличающей генетический алгоритм, является акцент на оператор, выполняющий роль скрещивания, аналогичного скрещиванию в живой природе, проводящего рекомбинацию решений-кандидатов. В качестве основного назначения генетических алгоритмов можно отметить поиск решений в многомерных пространствах.

Основные принципы работы простого ГА (генетического алгоритма):

1. Генерируется начальная популяция из n хромосом.

2. Вычисляется для каждой хромосомы ее пригодность (оптимальность).

3. Выбирается пара хромосом-родителей с помощью некоторого способа отбора.

4. Проводится кроссинговер двух родителей с вероятностью p_c, производя из них двух потомков.

5. Проводится мутация потомков с вероятностью p_m.

6. Повторяются шаги 3–5, пока не будет сгенерировано новое поколение популяции, состоящее из n хромосом.

7. Повторяются шаги 2–6, пока не будет достигнут критерий окончания процесса.

Критерием окончания алгоритма чаще всего служит заданное количество поколений или схождение популяции.

Под схождением подразумевается такое состояние популяции, когда все особи популяции почти одинаковы и находятся в области определенного экстремума.

Основные типы операторов. *Выбор родителей* – выбирает пары из общей популяции, которые будут подвержены скрещиванию для создания следующей популяции. Оператор *воспроизведения или рекомбинация (также кроссинговер)* — создание этих потомков, которые наследуют часть генов от одного родителя, и часть — от другого. Оператор *мутации* служит для того, чтобы выбить очередную популяцию с локального минимума или максимума, а также он препятствует слишком ранней сходимости. Так получается посредством того, что случайно выбранный в хромосоме ген изменяется в какую-либо сторону. Оператор *отбора* выбирает, какие из особей попадут в новую популяцию.

Важнейшие операторы — рекомбинации и мутации. Оператор *рекомбинации* используют сразу же после оператора выбора родителей для создания новых особей-потомков. Смысл рекомбинации заключается в том, что сформированные потомки должны наследовать генную информацию от обоих особей-родителей. Различают дискретное воспроизведение и кроссинговер.

Дискретная рекомбинация соответствует обмену генами между членами популяции. Дискретная рекомбинация может быть применена для любого типа генов (двоичные, вещественные и символьные).

Промежуточная рекомбинация применима только к вещественным данным, но не к бинарным. В данном методе заранее определяется числовой интервал значений генов потомков, который должен включать значения генов родителей. Потомки создаются по следующему принципу:

$$\text{Потомок} = \text{Родитель } 1 + \psi \cdot (\text{Родитель } 2 - \text{Родитель } 1),$$

где множитель ψ — случайное число на отрезке $[-d, 1 + d]$, $d > 0$. Как заметили сторонники этого метода, наилучшее воспроизведение получается при $d = 0{,}25$. Для каждого гена формирующегося потомка выбирается свой множитель ψ.

Слияние (рекомбинация) двоичных данных часто называется кроссинговером, реже кроссовером, а также скрещиванием.

Перечислим несколько основных видов кроссинговера:

Одноточечный кроссинговер. У нас будут существовать две особи родителя с хромосомами, которые имеют одинаковую длину. В случайном месте в каждой хромосоме выбирается точка разрыва, которая делит родительские хромосомы на две части, которые меняются друг с другом между родителями. Одноточечным такой способ называется по той причине, что в нем выбирается только одна случайно выбранная точка.

Существует *двухточечный кроссинговер*. Как в нем, так и в многоточечном кроссинговере в целом, хромосомы можно рассматривать как циклы, формирующиеся при соединении краев линейной хромосомы. Для того, чтобы заменить сегмент одного цикла на сегмент другого цикла понадобятся две разбивающие точки. В таком виде можно представить и одноточечный кроссинговер, если вторую точку принять за начало хромосомы. Таким образом, видно, что двухточечный кроссинговер служит той же самой цели, что и одноточечный, но делает это более полно.

На практике используется и *многоточечный кроссинговер*, когда хромосомы делятся на несколько отрезков.

После того, как выполнены операторы воспроизводства, приходит время применения *мутации*. Этот оператор служит для того, чтобы выбить очередную популяцию с локального минимума или максимума, а также он препятствует слишком ранней сходимости. Так получается посредством того, что случайно выбранный в хромосоме ген изменяется в какую-либо сторону.

Мутации, точно так же как и скрещивание могут проходить по нескольким точкам. Помимо этого, выбор количества таких точек также может быть случайным. Мутации могут использоваться, изменяя сразу определенную группу идущих подряд точек.

Обычно задается некоторая вероятность мутации, которая может быть случайным числом на отрезке от нуля до единицы, либо же функцией от некоторой характеристики задачи, поставленной к решению. К примеру, вероятность мутации генов можно задать обратно пропорциональной для размерности особи (общего числа её генов).

Рассмотрим мутации с *вещественными числами* и для чисел *в двоичной* нотации.

В первом случае нужно определить специальный параметр — величину шага мутации, то есть число, которое будет прибавлено или отнято от значения гена во время мутации.

А для особей содержащих двоичный код или код Грея, мутация будет подразумевать случайную инверсию гена (единица меняется на ноль и наоборот).

Стоит отметить также *плотность мутации*. В данном случае стратегия мутации с использованием данного понятия заключается в том, что каждый ген потомка может с заданной вероятностью мутировать. То есть, помимо вероятности использования мутации к особи-потомку, используется также вероятность мутации каждого его гена. Обычно выбирают значение этой вероятности с тем планированием, чтобы мутации подверглось в среднем от одного до десяти процентов генов.

Во второй главе «Транспортная задача» дается постановка задачи её решение несколькими спрособами. Саму по себе транспортную задачу не относят к задачам оптимизации в экономике и социологии труда. Тем не менее, модификации этой задачи или потенциально приводимые через преобразование путем определенных действий к этой задаче (к примеру таких задач можно отнести задачу о назначениях или же задачу оптимизации использования фонда рабочего времени) — составляют довольно широкий ряд в области экономики и социологии труда. Потому следует кратко рассмотреть постановку транспортной задачи, её модель и методы её решения.

Метод потенциалов является модификацией симплекс-метода решения задачи линейного программирования применительно к транспортной задаче. Он позволяет, отправляясь от некоторого допустимого решения, получить оптимальное решение за конечное число итераций. Сначала находится первый опорный план, который затем улучшается, пока не станет оптимальным.

Поставленная задача была решена этим методом. Последний опорный план явился оптимальным, так все оценки свободных клеток удовлетворяют условию $u_i + v_j \le c_{ij}$. Минимальные затраты составили: F(x) = 78. Для контроля правильности решения воспользовались встроенными функциями excel. В результате получаем оптимальное решение поставленной транспортной задачи показанной, которое совпало с полученным ранее.

Во время решения задачи генетическими алгоритмами значение целевой функции совпало со значением полученным при решении классическим методом. Однако оптимальный план отличается, это объясняется тем, что поставленная задача оптимизации имеет несколько решений и генетический алгоритм приблизился к одному из них.

То, что в задаче присутствуют ограничения неравенств (функциональные ограничения) в сильной степени усложняет нахождение оптимального решения. Можно преобразовать задачу с ограничениями в задачу без ограничений, это идея представляется привлекательной, с учетом наличия надежных и эффективных методов безусловной оптимизации.

Попробуем использовать метод штрафных функций. Так называют метод решения задач (преимущественно нелинейного программирования, но в нашем случае также можно применить его), который при использовании сводит задачу к задаче минимизации определенной специальной функции. Она представляет собой сумму основной целевой функции и определенных штрафных функций, которые формируются на основе ограничений.

Основная идея этого метода заключается в замене фитнесс функции задачи на определенную обобщенную функцию, у которой значения равны значениям первоначальной функции внутри области допустимых значений, а при выходе из нее, за счет штрафных функций (созданных на основе ограничений задачи), начинает резко и сильно увеличиваться. Таким образом, штрафные функции построены таким образом, что обеспечивает невозможность выхода из области допустимых решений, либо быстрое в неё возвращение.

В нашем примере первая штрафная функция представляет собой перемножение матрицы выбора переменных (перед участвующими в ограничениях переменными стоит коэффициент 1, а перед не участвующими — 0) на матрицу найденного решения, и от этого произведения (левых частей ограничений) отнимается матрица правых частей ограничений. Далее мы оставляем элементы полученной матрицы, которые больше нуля, а остальным присваиваем нули (это сделано потому, что нам надо отказаться только от превышения отправки со склада, но не от понижения). После чего каждый элемент умножается на коэффициент, предназначенный для большего значения штрафа, который получается из сложения все полученных элементов. Штраф прибавляется к целевой функции, приводя её значение к числам, значительно бóльшим, нежели при оптимальном решении.

Вторая штрафная функция работает по тому же принципу, однако там нельзя допустить как занижение доставки в магазин, так и превышение, поэтому и отрицательные и положительные знания разности возводятся в модуль и участвуют в создании штрафа.

В третьей главе «Задача коммивояжера» описывается что, одной из самых известных задач комбинаторной оптимизации является ***задача коммивояжёра*** (Travelling salesman problem - TSP). Суть ее в нахождении наиболее выгодного маршрута с учетом того, что он проходит через все заданные точки хотя бы по одному разу и возвращается в исходную точку. В условиях данной задачи можно вводить критерии для выбора выгодности маршрута, например, по кратчайшему расстоянию, по себестоимости или по некоторой совокупности ограничений и критериев. В классической постановке маршрут должен проходить через каждую точку один раз и, следовательно, выбор в этом случае происходит среди гамильтоновых циклов, но есть и частные случаи данной задачи. Среди частных случаев можно выделить геометрическую задачу коммивояжёра. Она еще называется планарной или евклидовой. Для геометрической задачи матрица расстояний отражает расстояния между точками на плоскости. Другой разновидностью задачи коммивояжера является метрическая задача. Для нее на матрице стоимостей должно выполняться неравенство треугольника. И наконец, имеются задачи коммивояжёра как симметричная, так и асимметричная. Наибольшую популярность получила обобщённая задача коммивояжёра.

Развитие современных методов и вычислительной техники дало возможность решать более сложные задачи, на основе классической задачи коммивояжера.

Для решения задачи коммивояжера разработано большое количество методов, которые отличаются скоростью решения и точностью. Рассмотрим кратко некоторые из них: «жадный»; полный перебор; генетические алгоритмы.

Жадный алгоритм представляет собой алгоритм, который ищет наименьшее расстояние за счет выбора на каждом шаге наиболее короткого и до сих пор еще не выбранного ребра, с условием, что это ребро не будет образовывать цикл с выбранными ранее ребрами. Называется же он так потому, что за его «жадность» на последних шагах ему приходится жестоко расплачиваться, из-за оставшихся больших расстояний. Этот алгоритм довольно быстрый, но имеет малую точность.

Полный перебор, называемый также «перебор животной силой» может быть практически применён только в задачах с малой размерности. Важно отметить, что n-размерная задача коммивояжёра (то есть задача, содержащая n городов) тратит на полный перебор всех вариантов в симметричной задаче (n-

1)!/2 туров и в несимметричной — (n-1)! туров. Как известно, факториал растет очень быстро.

Для того, чтобы провести полный перебор задачи коммивояжера, следует без повторений, сгенерировать все возможные сочетания последовательности заданного количества элементов. Для этого можно использовать несколько различных способов, однако, один из самых популярных и приложимых для алгоритмов перебора — метод перебора в лексикографическом порядке. Можно для примера рассмотреть перестановку из пяти элементов имеющих обозначение цифрами от 1 до 5. По этому принципу первой перестановкой будет являться 1-2-3-4-5, второй — 1-2-3-5-4, …, последней 5-4-3-2-1. Следует усвоить общий алгоритм преобразования любой из перестановок в следующую.

Необходимо придерживаться следующего правила: допустим, дана перестановка 1-3-5-4-2. Начинать следует с движения справа налево, двигаясь до тех пор, пока не достигнем числа меньшего, чем предыдущее (в данном случае 3 после 5), данную последовательность можно назвать суффиксом (5-4-2). Данное число P_{i-1} (3) надо увеличить таким образом, чтобы оно поменялось местами с числом стоящим правее него (числом из суффикса). Такое число обязательно найдется, потому что $P_{i-1} < P_i$. В случае, если имеется несколько больших чисел, то необходимо ставить наименьшее из них (у нас надо поменять 3 и 4). Затем следует упорядочить суффикс (который изменится и в нашем случае станет равен 5-3-2) по возрастанию (2-3-5). Таким образом, мы получили следующую перестановку (в примере — 1-4-2-3-5). После неё будет идти 1-4-2-5-3 (работает все тот же алгоритм, но в более простом случае), и т.д. После генерации каждой перестановки, в главной программе происходит расчет суммы, которая записывалась в файл. После подсчета всех перестановок, из всех сумм выбирается наименьшая, которая и является решением задачи. Этот метод является точным, но очень медленным. При подсчете задач с большой размерностью требуется крайне много памяти. Полный перебор задачи коммивояжёра практически не осуществим в условиях большой размерности задачи, поэтому для её решения практикуется использование приближенных методов, которые позволяют найти приемлемое, но не обязательно оптимальных решений.

Метод полного перебора может быть применён исключительно в задачах с малой размерностью. Решим задачу коммивояжёра встроенным в Excel симплекс-методом. На машине с четырёх-ядерным процессором 2,67 ГГц 10 точек (городов) обсчитывается в среднем за 5 миллисекунд, 20 узлов уже за 15 минут. Начиная с размерности матрицы 7x7, время начинает экспоненциально расти. Следовательно, классические методы, например перебором, непригодны для решения задач с большим количеством точек (городов).

Генетический алгоритм решает задачу коммивояжёра с 15 городами за 6,26 секунд, что не является сильно выдающимся результатом, однако задачу со 100 городами приведенный в работе код решает за 13 секунд, что не сравнимо с миллиардами лет, которые пришлось бы потратить на перебор.

ЗАКЛЮЧЕНИЕ

Использование таких алгоритмов как генетический алгоритм требует формирования исходных данных, в том числе, и ограничений в виде векторов и матриц. Для более стабильной работы данных алгоритмов требуется более тонкая настройка целевой функции. Например, с использованием ведения штрафных функций или определение отклонения целевой функции в среднем. Но, несмотря на это, использование данных алгоритмов целесообразно в условиях частичной неопределенности и позволяет найти решения, которые обычными метода найти не представляется возможным. Использование данных методов возможно и на больших объемах данных и в условиях существования нескольких решений, что весьма затруднительно для простых методов оптимизации. Видим, что нахождение оптимального пути для ста городов в 14 секунд, быстрее перебора, который решал бы данную задачу в масштабах миллиардов лет.

В данной работе было показано, что генетические алгоритмы могут успешно решать оптимизационные задачи экономики и социологии труда, а так же их преимущество в скорости, если методы линейного программирования и перебора решают задачу недостаточно быстро.

На практике было выполнено решение транспортной задачи вручную методом потенциалов, симплекс-методом в программе Excel, а также при помощи генетического алгоритма с и без использования штрафных функций, благодаря чему убедились в правильности решения задачи генетическим алгоритмом. А также решили задачу коммивояжёра методом перебора и с помощью генетического алгоритма, убедившись, на больших объемах задачи генетических алгоритм показывает себя гораздо эффективнее.