

## ТЕХНИЧЕСКИЕ НАУКИ

*Лобода Максим Николаевич*

магистрант

*Березкина Галина Леонидовна*

старший преподаватель

ФГАОУ ВПО «Дальневосточный федеральный университет»

г. Владивосток, Приморский край

**АНАЛИЗ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ ЛОКАЛЬНОГО ПОИСКА  
В СРЕДЕ WEB**

*Аннотация:* в данной работе производится оценка эффективности алгоритмов оптимизации для задачи нелинейного программирования. Критериями оценки являются эффективность решения задачи, затраченное время, использованная память. В качестве эталонного, в смысле эффективности решения задачи, выбран алгоритм случайной оптимизации. Оценка алгоритмов производится в среде web, что подразумевает реализацию их на языке, используемом для программирования web-приложений.

*Ключевые слова:* целевая функция, оценка эффективности, локальный поиск, экстремум функции, имитация отжига, случайная оптимизация, генетический алгоритм, движение в сторону, жадный поиск.

В настоящее время существует тенденция повышения популярности приложений, функционирующих в среде web. Это подтверждается созданием и уже повсеместным использованием облачных технологий, увеличением вычислительных мощностей переносных устройств, а также широчайшими зонами покрытия беспроводных сетей. Всё это в совокупности открывает новые перспективы для разработки сложных приложений в данной сфере. Но даже наличие очень больших вычислительных ресурсов не может гарантировать решения некоторых задач. Такими задачами могут являться задачи нелинейного программирования. Рассмотрим решение одной такой задачи на примере.

Допустим, мы разрабатываем приложение, помогающее менеджеру по закупкам или логисту определиться, у кого из  $N$  поставщиков закупить каждый из  $M$  вида продукции. Для упрощения расчетов предположим, что у каждого из поставщиков в наличии имеется неограниченное количество единиц каждого вида товара. В таком случае, менеджеру предстоит перебрать вручную  $N^M$  количество вариантов, что соответствует количеству размещений из  $N$  поставщиков по  $M$  возможными видам товара с повторениями. Сложность такой задачи оценивается как экспоненциальная. Другими словами, если размерность задачи возрастает линейно, то время её решения возрастает экспоненциально. В нашем случае, как раз, так и происходит. Количество поставщиков может измениться так же, как и количество видов товара, необходимых для закупки. Надо отметить также, что изменения эти являются линейными.

Менеджер заинтересован в нахождении состояния, при котором закупка товара будет наиболее выгодной. Таким состоянием в нашем случае является конфигурация с наименьшей суммарной стоимостью, при которой, каждому виду товара соответствует поставщик. Оценить суммарную стоимость каждой конфигурации можно с помощью функции называемой целевой. На значение целевой функции могут влиять различные факторы. Например:

- расстояние до поставщика;
- наличие скидки за покупку одного вида товара в количестве большем, чем  $K$ ;
- наличие скидки за покупку товара двух и более видов в количестве большем, чем  $S$  каждого;
- наличие условий сложения скидок.

Не трудно догадаться, что решение такого рода задачи с помощью полного перебора всех возможных вариантов является не самым удачным выбором. К счастью в настоящее время разработано множество алгоритмов, позволяющих, если не находить лучшее решение, то получать довольно-таки приемлемые результаты. Такие алгоритмы называются локальными алгоритмами оптимизации.

Их задачей является поиск состояния наилучшего с точки зрения целевой функции.

В данной работе были рассмотрены следующие алгоритмы:

- алгоритм случайной оптимизации;
- алгоритм спуска с горы;
- алгоритм имитации отжига;
- генетический алгоритм.

Алгоритм случайной оптимизации выбран в качестве эталонного. Принцип его действия заключается в случайном выборе заданного числа конфигураций, оценке их с помощью целевой функции и предоставлении лучшего найденного.

Алгоритм спуска с горы или градиентный спуск является жадным алгоритмом оптимизации. Его принцип заключается в выборе наилучшего соседнего решения с точки зрения целевой функции. Когда таких решений не найдено алгоритм заканчивает свою работу. Достоинством данного алгоритма является его быстрая сходимость, недостатками является его неспособность выходить из локальных минимумов функции и конфигураций, называемых плато. Плато является такой конфигурацией, при которой соседние конфигурации не изменяют значение целевой функции.

Алгоритм имитации отжига является примером не жадного алгоритма. Его особенностью является возможность так называемого шага в сторону. Шагом в сторону является возможность перехода к худшему решению с целью нахождения лучшего. Принцип алгоритма навеян физическими процессами, происходящими с металлом при процедуре, называемой отжиг.

Идея генетического алгоритма позаимствована у такого процесса как эволюция живых организмов. Данный алгоритм имеет два механизма для избегания скатывания в локальный минимум. Это скрещивание и мутация. Принципом алгоритма является формирование случайных конфигураций, называемых популяциями и дальнейшее их улучшение с помощью вышеупомянутых механизмов.

Для оценки описанных выше алгоритмов использовались следующие критерии: эффективность найденного решения; время, затраченное на выполнение;

использованное количество памяти. Все алгоритмы были реализованы на интерпретируемом языке PHP и были апробированы на одной машине. В качестве примера была использована описанная выше задача со следующими параметрами: 10 – количество поставщиков, 5 количество видов товара, 5% размер скидки при покупке товара в количестве более 50 единиц, 3% размер скидки при покупке товара более двух видов в размере более чем 20 единиц каждого. Скидка рассчитывается для каждого вида товара отдельно. Например: для товара F1 может быть использованы два вида скидок, в то время как для товара F2 только один.

В таблице 1 приведены размеры закупки товаров.

Таблица 1

Товар №1	Товар №2	Товар №3	Товар №4	Товар №5
50	34	17	69	33

В таблице 2 приведены результаты тестирования алгоритмов.

Таблица 2

Алгоритм	Среднее найденное решение	Лучшее найденное решение	Среднее затраченное время (сек)	Средняя использованная память (Мб)	Среднее количество шагов
Случайный	4517	3215	0.3	2.9	100
Спуск с горы	5921	5333	0.23	2.9	4
Имитация отжига	5823	4303	0.37	2.9	6
Генетический	3714	3217	28	3.14	100

В качестве заключения стоит сказать, что для различных конфигураций закупок эффективность алгоритмов может значительно отличаться. Но даже при условиях, описанных в данной работе видно, что генетический алгоритм намного более требователен к ресурсам, ввиду более высокой сложности его реализации и необходимости множества промежуточных вычислительных операций. Тем не менее, в условиях данной задачи он показал результат значительно лучше других. Стоит так же отметить алгоритм случайной оптимизации, показавший относительно неплохой результат. Успех этого алгоритма предположительно обусловлен относительно небольшой размерностью задачи.

### ***Список литературы***

1. Мхитарян В.С. Теория вероятностей и математическая статистика / В.С. Мхитарян В. [и др.]; под ред. В.С. Мхитаряна. – 2013.
2. Рассел С. Искусственный интеллект. Современный подход / С. Рассел, П. Норвиг. – Издательский дом «Вильямс», 2006.
3. Сегаран Т. Програмируем коллективный разум. – Символ-Плюс, 2008.
4. Таха Х. Введение в исследование операций. – Издательский дом «Вильямс», 2005.