УДК 004.021

В. С. Бурмако, О. И. Наранович
Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

МОДИФИЦИРОВАННЫЙ АЛГОРИТМ МЕТОДА МОНТЕ-КАРЛО

Введение. В настоящее время человечество вынуждено постоянно принимать какие-либо решения — будь то покупка одежды или развитие предприятия. Принятие решений всегда связано с анализом множества различных факторов. Существуют методы принятия решений, которые применимы в том случае, если анализируемые факторы можно представить в числовом виде. В таком случае можно формализовать задачу с помощью аналитических выражений, которые поддаются оценке с помощью некоторого алгоритма. Однако число факторов может быть достаточно большим или же их анализ будет требовать больших вычислений для получения точной оценки. Для таких ситуаций используются различные методы оптимизации принятия решений, такие как методы экспертных оценок, методы линейного, нелинейного, дискретного программирования и др. Одним из методов решения оптимизационных задач является метод Монте-Карло.

В настоящее время метод Монте-Карло имеет необычайно широкий спектр приложений. В числе наиболее известных: задачи переноса излучений — ядерные реакторы, атмосферная оптика и др.; задачи газовой динамики — метод Бёрда, моделирование процессов коагуляции; задачи финансовой математики — моделирование управления ценными бумагами и рыночных ситуаций; задачи массового обслуживания — моделирование сложных производственных систем, систем связи и компьютерных сетей [1, с. 5].

Целью исследования является установление возможности повышения скорости работы алгоритма метода Монте-Карло.

Основная часть. Для того чтобы решить оптимизационную задачу методом Монте-Карло, необходимо построить математическую модель, которая может состоять из целевой функции и ограничений. Под целевой функцией понимается некоторое математическое выражение, для которого необходимо найти экстремум на множестве допустимых значений. Множество допустимых значений формируется исходя из ограничений.

Данный метод наиболее эффективен для решения на ЭВМ, так как требует большого количества итерационных вычислений. Суть данного метода состоит в том, что в некотором указанном диапазоне случайно генерируются точки, в каждой из них просчитывается значение целевой функции. Данное действие повторяет указанное количество итераций. В результате всех вычислений выбирается точка, в которой значение функции достигает искомого экстремума.

Так как существует шанс того, что программа сгенерирует повторные точки, было предложено модифицировать данный метод, чтобы избежать вычислений для повторно сгенерированных точек. Суть алгоритма модификации метода Монте-Карло состоит в том, что во время генерации точек, если сгенерированная точка отличается от одной из ранее сгенерированных менее чем на некоторое заданное число, то она отбрасывается. В результате вычисления проводятся только для тех точек, которые не были отброшены на этапе генерации.

Для реализации данной модификации был выбран язык программирования Java и решено помещать данные о выбранных точках в контейнер HashSet, так как он автоматически не сохраняет повторные значения в случае попытки их записи. Класс HashSet служит для создания коллекции, где для хранения элементов используется хеш-таблица. Для хранения данных в хеш-таблице применяется механизм хеширования, где содержимое ключа служит для определения однозначного значения, называемого хеш-кодом. Этот хеш-код служит далее в качестве индекса, по которому сохраняются данные, связанные с заданным ключом. Преимущество хеширования заключается в том, что оно обеспечивает постоянство времени выполнения методов add() (добавляет элемент в коллекцию), contains() (определяет, есть ли элемент в коллекции), remove() (удаляет элемент из коллекции) и size() (возвращает размер коллекции) даже для крупных множеств [2, с. 656]. За счет этого преимущества мы можем не тратить дополнительное время на поиск повторений среди сгенерированных точек, а предоставить решение о добавлении новой точки во множество входных значений классу HashSet.

Для исследования алгоритма модификации была выбрана оптимизационная задача на поиск значения целевой функции, стремящейся к максимуму: $F = -x_1^2 - x_2^2 + 2x_1 + 2x_2 \to \max$ при следующих ограничениях:

$$\begin{cases} -x_1 - x_2 \le 0; \\ 2x_1 + 2x_2 \ge 0; \\ x_1 \ge 0; x_2 \ge 0. \end{cases}$$

Для обоих алгоритмов генерировались идентичные значения точек в указанном диапазоне и рассчитывалось время выполнения методов по отдельности. Также для графического отображения зависимостей времени работы алгоритмов от входных параметров обозначим отношение количества итераций (k) к количеству всех возможных точек в указанном диапазоне (N) как R, т. е. $R = \frac{k}{N}$.

По сравнению с алгоритмом метода Монте-Карло модификация более требовательна к объему памяти, так как вынуждена запоминать все значения, но при этом скорость ее выполнения возрастает с увеличением отношения R, т. е. с увеличением вероятности того, что будет сгенерирована точка, значение функции для которой уже было просчитано. Данные о результатах исследования скорости вычисления алгоритма метода Монте-Карло и его модификации приведены на рисунке 1. Пример работы программы представлен на рисунке 2.

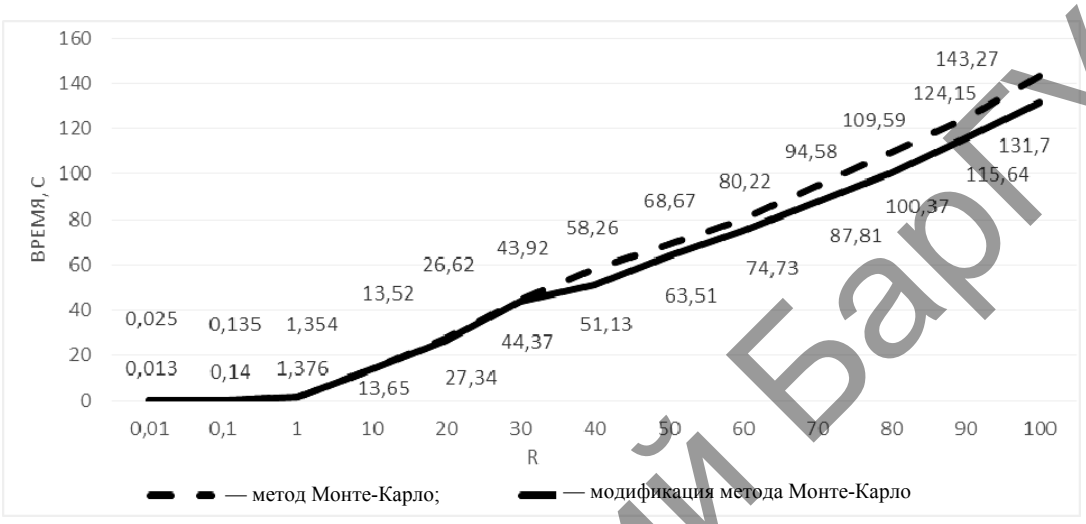

Рисунок 1 — Графическое представление зависимостей времени работы алгоритмов от входных параметров

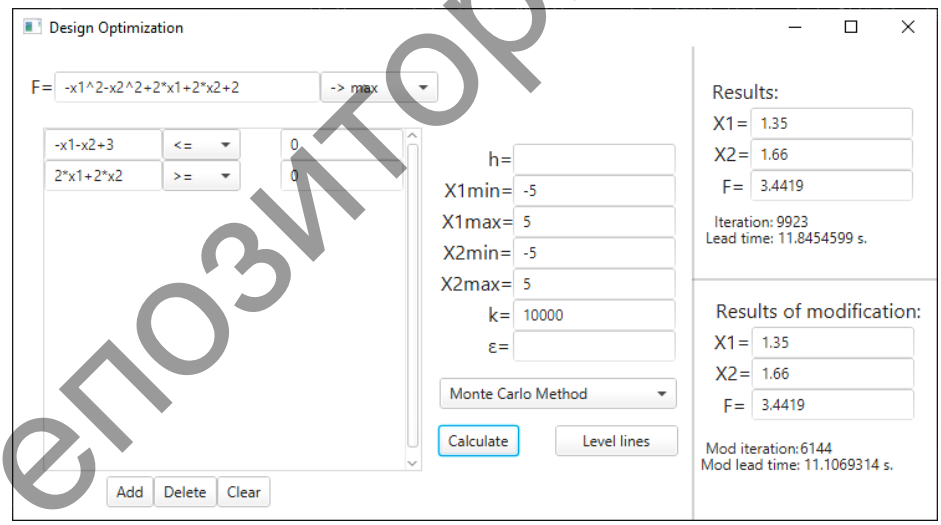

Рисунок 2 — Результаты решения задачи обоими алгоритмами

Заключение. Модифицированный алгоритм ускоряет вычисления для большого количества итераций, что достаточно важно при решении оптимизационных задач с большим количеством ограничений и сложной структурой целевой функции. Но стоит заметить, что данная модификация достаточно требовательна к ресурсам памяти компьютера. Следовательно, данную модификацию целесообразно применять для вычислений на компьютерах с хорошим запасом оперативной памяти, а требования к производительности компьютера минимальны.

Список цитируемых источников

1. *Ермаков, С. М.* Метод Монте-Карло в вычислительной математике / С. М. Ермаков. — М. : С.-Петерб. гос. ун-т, 2009. — 192 с.
2. *Шилдт, Г.* Java. Полное руководство / Г. Шилдт. — 10-е изд. — М. ; СПб. : Альфакнига, 2018. — 1488 с.