

Каждый пользователь может передвигать свой корабль при помощи джойстиков. При соприкосновении корабля и кристалла, последний будет перемещён в случайную точку, а время пользователя, подобравшего кристалл, увеличено.

По истечении времени одного из игроков на экран будет выведено сообщение о победе, если время игрока не дошло до нуля, и о поражении, если время дошло до нуля. При нажатии на кнопку перезапуска корабли примут изначальное положение, а кристалл выберет случайное положение.

Заключение. В практическом смысле эта работа показывает доступность и простоту разработки приложений с дополненной реальностью для образовательной сферы, в частности для закрепления аббревиатур в игровой форме, а также простоту создания сетевой составляющей для дополненной реальности.

Список цитируемых источников

1. Калько, А. И. Приложение с дополненной реальностью демонстрации виртуальных 3D-объектов и возможности их сетевого взаимодействия / А. И. Калько, Р. В. Мазура, О. Д. Хадарович // 73-я Международная студенческая научно-техническая конференция : материалы конференции, Астрахань, 17-22 апреля 2023 года. — Астрахань : Астраханский государственный технический университет, 2023. — С. 717—718.

2. Vuforia [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/Vuforia>. — Дата доступа: 20.04.2024.

УДК 004.023

Р. В. Мазура

Учреждение образования «Барановичский государственный университет»,
Барановичи, Республика Беларусь

ПРИМЕНЕНИЕ МНОГОПОПУЛЯЦИОННОГО ГЕНЕТИЧЕСКОГО АЛГОРИТМА В ЗАДАЧАХ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Введение. Линейное программирование (далее — ЛП) является мощным инструментом для решения широкого спектра оптимизационных задач, включая задачи планирования, логистики, производства и экономики. Несмотря на свою эффективность, точные методы решения ЛП-задач могут столкнуться с ограничениями в вычислительной мощности при работе с большими объемами данных или сложными моделями.

Основная часть. Для решения подобной проблемы можно использовать многопопуляционный генетический алгоритм (далее — МГА) благодаря его способности к обработке больших объемов данных и гибкости в работе с разнообразными типами задач.

МГА — это вид генетического алгоритма, который использует несколько независимо эволюционирующих популяций, которые обмениваются лучшими особями между собой.

Особь представляет собой одномерный массив значений.

Цель исследования: изучить насколько подходит многопопуляционный генетический алгоритм для решения задач линейного программирования.

Для разработки приложения с МГА был выбран редактор кода VS code, язык программирования Python 3.11, библиотека PyQt5 и pyqtgraph.

PyQt5 — это набор Python библиотек для создания графического интерфейса на базе платформы Qt5 от компании Digia [1].

PyQtGraph — это библиотека графики и пользовательского интерфейса для Python, которая обеспечивает функциональность, обычно необходимую при проектировании и научных приложениях. Его основные цели — предоставить быструю интерактивную графику для отображения данных (графики, видео и т. д.), а вторая — предоставить инструменты, помогающие в быстрой разработке приложений (например, деревья свойств, такие как используемые в Qt Designer) [2].

Для получения точного результата, с которым будет сравниваться МГА, выбран симплекс-метод за его точность.

Симплекс метод — это метод последовательного перехода от одного базисного решения (вершины многогранника решений) системы ограничений задачи линейного программирования к другому базисному решению до тех пор, пока функция цели не примет оптимального значения (максимума или минимума) [3].

Для анализа эффективности МГА будет рассмотрена следующая базовая математическая модель:

$$F(x) = 3x_1 + 4x_2 + 3x_3 + x_4 \rightarrow \max$$
$$\begin{cases} 5x_1 + 8x_2 + 4x_3 + 3x_4 \leq 480 \\ 2x_1 + 4x_2 + x_3 + 7x_4 \leq 130 \\ 7x_1 + 2x_2 + 3x_3 + 6x_4 \leq 80 \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

В качестве основного значения, по которому будет определяться отклонение МГА от симплекс-метода, будет выступать значение целевой функции.

Также кроме отклонения от симплекс-метода будет замеряться и время, за которой МГА приходит к конечному результату. Конечным результатом будет считаться значение, которое даёт наилучшая особь, при условии, что эта особь во всех популяциях оказалась лучшей на протяжении 3 последовательных поколений.

Всего будет проведено 8 запусков МГА, в которых количество используемые колоний будет соответствовать номеру попытки.

Общая схема работы МГА выглядит следующим образом (рисунок 1):

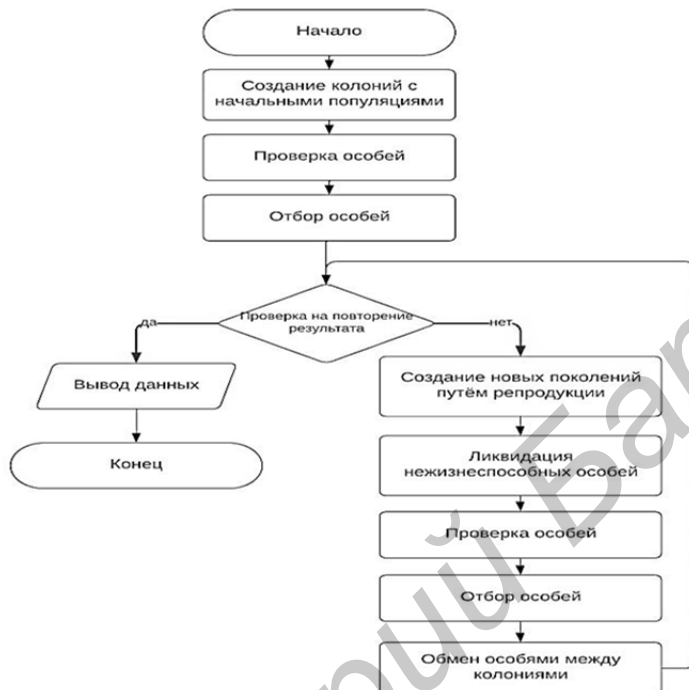


Рисунок 1 — Общая схема работы МГА

Результаты запусков, показанные на рисунке 2, свидетельствуют о том, что с увеличением количества колоний наблюдается увеличение времени, необходимого для нахождения лучшей особи. Интересно отметить, что в нашем случае 8 колоний нашли лучшую особь за меньшее время, чем 7 колоний, что свидетельствует о наличии некоторой случайности.

А результаты запусков, показанные на рисунке 3, свидетельствуют, что с увеличением количества колоний будет увеличиваться и точность нахождения значения целевой функции. Стоит заметить, что при 3-х колониях отклонение от симплекс-метода составляет 0 %, а это значит, что больше количество колоний для решения этой задачи не требуется. Однако, в контексте более сложных задач повышение точности может потребовать использование большего количества колоний, хотя это может привести к увеличению времени нахождения лучшей особи.

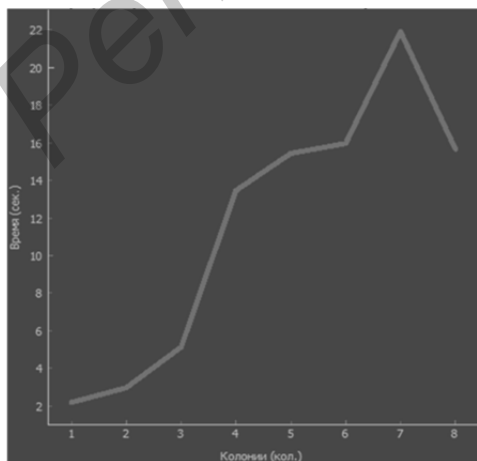


Рисунок 2 — График зависимости времени от количества колоний

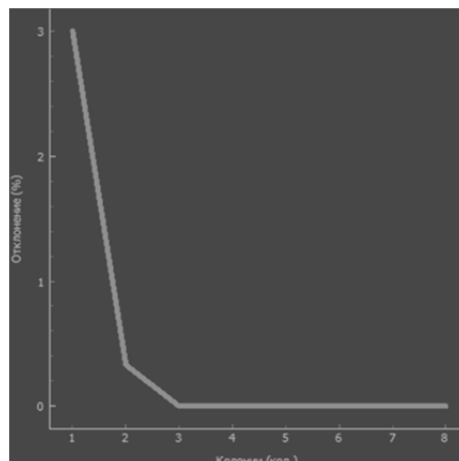


Рисунок 3 — График отклонения от симплекс-метода

Заключение. Результаты, показанные в данной статье, говорят о том, что МГА представляет собой перспективный инструмент для решения задач линейного программирования. Стоит отметить, что увеличение числа колоний МГА может увеличить точность, это может привести к увеличению времени выполнения алгоритма. Тем не менее, в контексте более сложных задач МГА представляет эффективный подход к решению задач оптимизации, обеспечивая гибкость и высокую точность в поиске оптимальных решений.

Список цитируемых источников

1. Python 3 для начинающих [Электронный ресурс]. — <https://pythonworld.ru/gui/pyqt5-firstprograms.html> . — Дата доступа: 20.04.2024
2. Introduction to PyQtGraph Module in Python [Электронный ресурс]. — Режим доступа: <https://www.geeksforgeeks.org/introduction-to-pyqtgraph-module-in-python/> . — Дата доступа: 20.04.2024
3. Решение симплекс методом задачи ЛП: пример и алгоритм [Электронный ресурс]. — Режим доступа: https://kpfu.ru/staff_files/F32587409/Reshenie_simpleks_metodom_zadachi_LP_primer_i_algorithm.pdf . — Дата доступа: 20.04.2024.

УДК 004.3

О. В. Марко

*Учреждение образования «Барановичский государственный университет»,
Барановичи, Республика Беларусь*

РАЗРАБОТКА СХЕМЫ ГИБОЧНОГО СТАНКА ПРОВОЛОКИ

Введение. Гибочные станки для проволоки являются важным оборудованием в промышленности, широко применяемым в производстве различных изделий, начиная от элементов мебели и автомобильных деталей, и заканчивая деталями для строительных конструкций. Спрос на точные, эффективные и универсальные гибочные машины растет вместе с развитием промышленных отраслей и появлением новых технологий.

Разработка схемы гибочного станка проволоки является сложным процессом, требующим учета множества факторов, включая требования по точности, скорости, мощности, типу обрабатываемых материалов и геометрии изделий. В современных условиях акцент делается на автоматизации, увеличении производительности и снижении издержек производства, что ставит перед разработчиками задачу создания инновационных решений [1].

Цель работы — исследовать процесс разработки схемы гибочного станка проволоки, выявление основных этапов проектирования, технологических решений и требований к конечному продукту. Объект исследования: современные тенденции и инновации в области разработки гибочного оборудования, направленные на повышение эффективности и конкурентоспособности производства.

Основная часть. Станок для гибки проволоки представляет собой высокоточное оборудование, специально разработанное для формирования проволочных деталей в трехмерном пространстве. Принцип его работы основан на использовании компьютерного управления и системы гибки, позволяющей точно контролировать процесс гибки проволоки [2].

Основные принципы работы такого станка:

1. Проволока подается с бобины или другого источника и проходит через систему подачи, которая точно контролирует скорость и напряжение проволоки.
2. Проволока подается в гибочный блок, где специальные клеммы или ролики удерживают ее, а гибочные инструменты, как правило, двигаются вдоль нескольких осей, чтобы создавать необходимую форму.
3. После завершения гибки проволока может быть обработана или отделана в соответствии с требованиями конечного продукта.

На рисунке 1 изображена модель гибочного станка [3].

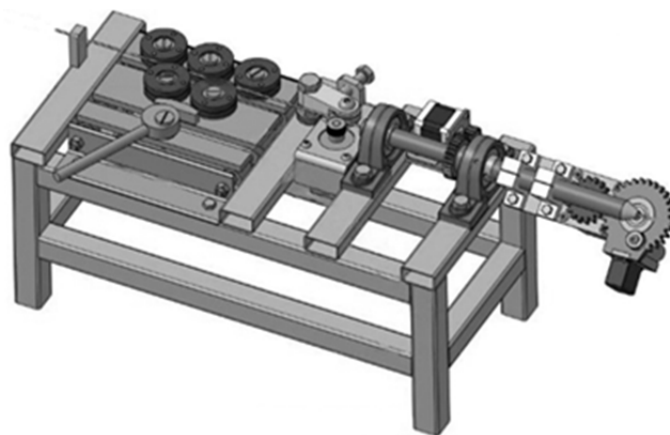


Рисунок 1 — Модель гибочного станка