Задание кафедры

Написать программу, реализующую один из методов оптимизации.  
Проверить работу программы тестами из лабораторных работ. Сравнить и  
проанализировать результаты. Программа должна позволять вводить тесты из файла и из формы ввода, а также сохранять отчет о результатах работы.

Вариант 30 - Комбинаторный эвристический алгоритм

Ход работы

1 Описание метода и алгоритма решения

Основная идея комбинаторного эвристического алгоритма состоит в дискретизации интервалов значений независимых переменных и реализации случайного поиска на полученной дискретной решетке, узлы которой представляют собой возможные значения переменных.

Поиск значения переменной осуществляется на этой фиксированной решетке путем квазислучайных процедур, проводимых отдельно для каждой переменной.  
Шаг1. Задаем допустимую начальную точку и полагаем . Затем для каждой переменной *i,* выполняемследующую последовательность вычислений.

Шаг2.Проводим оптимизацию по i-й переменной, зафиксировав  
остальные. Для этого

1) Выбираем случайным образом возможные значения i-ой  
переменной (например, используя формулу (12.1), или с  
предварительным разбиением интервала изменения переменной на q  
равных подынтервалов так, чтобы каждому допустимому значению  
переменной соответствовал один подынтервал,что предпочтительнее)  
для нахождения q дополнительных допустимых точек с  
лучшим значением целевой функции по сравнению с текущей  
базовой точкой.

2) Авторы алгоритма утверждают, что q должно лежать в  
диапазоне от 3 до 5. Если такие точки получить не удается,  
повторить шаг 2 для переменной i+1.

Шаг3. Проводим «упреждающий» поиск.

а) Для каждого из q допустимых решений, найденных на шаге  
2.1, проводим случайный выбор одного из q возможных  
значений переменной (i+1) для определения допустимого  
значения этой переменной, дающего лучшее значение  
целевой функции по сравнению с

б) Выбираем наилучшую из q допустимых точек. Фиксируем  
значение переменной i, соответствующее этой точке, как  
оптимальное.

в) Принимаем . Если , выполняем шаг 2 для следующей  
переменной . В противном случае переходим на шаг 4.

Шаг4. Проводим случайный поиск для определения наилучшего  
значения n-ой переменной при фиксированных значениях  
других переменных, соответствующих текущей базовой точке.  
Найденную точку принимаем за новую базовую точку, а  
значение целевой функции в ней – за новое значение

Шаг5. Проверяем условие окончания вычислений

Если условие выполняется вычисления заканчиваем . Если не выполняется полагаем и возвращаемся на шаг 2.

2 Руководство оператора

2.1 Назначение программы

Программа позволяет найти минимум функции многих переменных с ограничениями-неравенствами, используя комбинаторно эвристический алгоритм.

1) Ввод данных. Эта функция позволяет ввести данные вручную пользователем. Доступна сразу после того, как пользователь зашел на веб-страницу.

2) Поиск минимума с описанием шагов выполнения программы в виде отчета. Эта функция дает возможность пользователю найти минимум функции при помощи комбинаторно эвристического алгоритма. Доступна сразу после того, как пользователь ввел корректные исходные данные.

3) Загрузка исходных данных. Эта функция дает возможность пользователю загрузить исходные данные, вместо их ввода через интерфейс программы. Доступна сразу после того, как пользователь зашел на веб-страницу, при наличии исходных файлов с данными в определённом формате.

4) Сохранения исходных данных. Эта функция дает возможность пользователю сохранить введенные им через интерфейс программы исходные данные, чтобы в дальнейшем была возможность осуществить их загрузку в программу. Доступно пользователю после ввода всех входных данных или загрузки входных данных из файла.

5) Сохранение результата работы программы. Эта функция дает возможность пользователю сохранить осуществленные программой вычисления в процессе нахождения минимума функции. Доступна после того, как пользователь запустил поиск минимума функции в программе и дождался окончания выполнения расчетов.

Приложение реализовано с использованием таких языков программирования, как HTML5, CSS3, JavaScript.

2.2 Условия выполнения программы

Данная программа требует для своего выполнения наличия файлов программы с исходным кодом и интернет-браузер для исполнения данного кода. Программа не требует подключения к сети и может быть выполнена в среде любой операционной системы, в которой установлен интернет-браузер.

Аппаратные требования соответствуют аппаратным требованиям установленного интернет-браузера.

Программные требования – наличие установленного интернет-браузера, версия которого вышла после 2016 года.

2.3 Выполнение программы

Интерфейс программы представлен на рисунке 1.

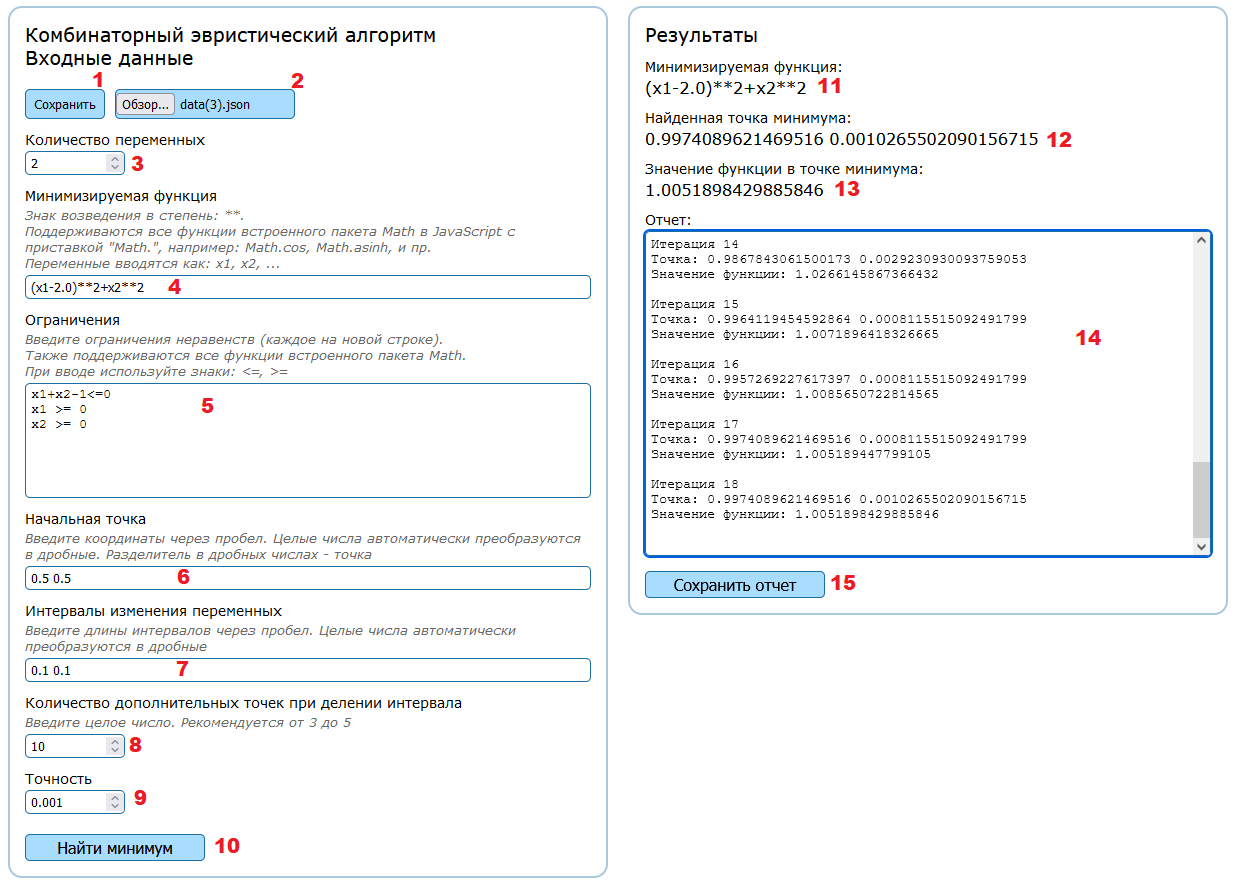


Рисунок 1 - Интерфейс программы

1) Кнопка сохранения введенных в программу исходных данных. При нажатии, в зависимости от настроек браузера может высветиться окно, где будет предложено, что пользователь хочет сделать со сформированным файлом. Сохранение будет произведено только в случае того, если пользователь правильно ввел исходные данные.

2) Кнопка загрузки исходных данных в программу. При нажатии открывается окно выбора файла, предопределенное операционной системой.

3) Поле ввода количества переменных в уравнении. По умолчанию – 2.

4) Поле ввода уравнения.

5) Поле ввода ограничений-неравенств для уравнения.

6) Поле ввода начальной точки поиска.

7) Поле ввода интервала, в пределах которого будут находиться возможные значения каждой из переменных для последующих вычислений согласно алгоритму.

8) Поле ввода количества подынтервалов, на которое необходимо разбить введенный выше интервал для указания на каждом из них случайного значения изменяемой переменной согласно алгоритму.

9) Необходимая точность значения искомого минимума функции.

10) Кнопка начала расчета минимума функции. Расчет начинается только, если пользователь ввел корректные исходные данные.

11) Поле, показывающее минимизируемую функцию.

12) Поле, показывающее найденную точку минимума.

13) Поле, показывающее значение функции в данной точке минимума.

14) Поле, содержащее отчет с осуществленными программой вычислениями в процессе нахождения минимума функции. В нем может находиться отчет из последней сессии работы пользователя с программой.

15) Кнопка сохранения отчета программы. Интерфейс сохранения аналогичен кнопке 1.

Шаги пользователя:

1) Пользователь открывает веб-страницу. Все справочные данные указаны возле соответствующего поля ввода.

2) Пользователь либо вводит данные в поля 3-9, при этом ввод в поле 5 необязателен; либо нажимает на кнопку 2 для выбора файла, содержащего данные программы в определенном формате. Если пользователь загрузил данные из файла, он может редактировать необходимые ему поля. После редактирования он может сохранить данные полей, нажав на кнопку 1.

3) Пользователь нажимает на кнопку 10 для начала расчета минимума функции при помощи комбинаторно эвристического алгоритма. Если пользователь совершил ошибку при вводе данных, то одно из сообщений об ошибке, описанных в пункте «сообщение оператору» данного документа, высветится либо под этой кнопкой, либо возле соответствующего поля ввода из полей 3-9.

4) Дождавшись окончания расчета, пользователь может скачать результат, нажав на кнопку 15.

5) Для повторного использования пользователю не нужно предпринимать дополнительных действий.

2.4 Сообщение оператору

1) «Начальная точка не удовлетворяет ограничениям» - некорректное значение в поле 6. Исходная точка не удовлетворяет условиям, описанным в поле 5.

2) «Слишком много итераций при поиске» - поиск минимума функции занял более 2000 итераций главного цикла, согласно алгоритму программы.

3) «Проверьте корректность ввода начальной точки» - некорректный синтаксис описания координат начальной точки. Появляется возле поля 6.

4) «Проверьте корректность ввода интервалов» - некорректный синтаксис описания интервалов. Появляется возле поля 7.

5) «Ошибка в ходе вычислений. Проверьте входные данные на правильность» - возникла исключительная ситуация в ходе выполнения расчетов.

Отчет о работе программы

В отчете приводится описания работы каждой итерации, содержащее значение точки и функции в данной точке при приближении к точке минимума функции. На последней итерации находится значение найденной точки минимума и функции в данной точке.

3 Результаты тестирования программы

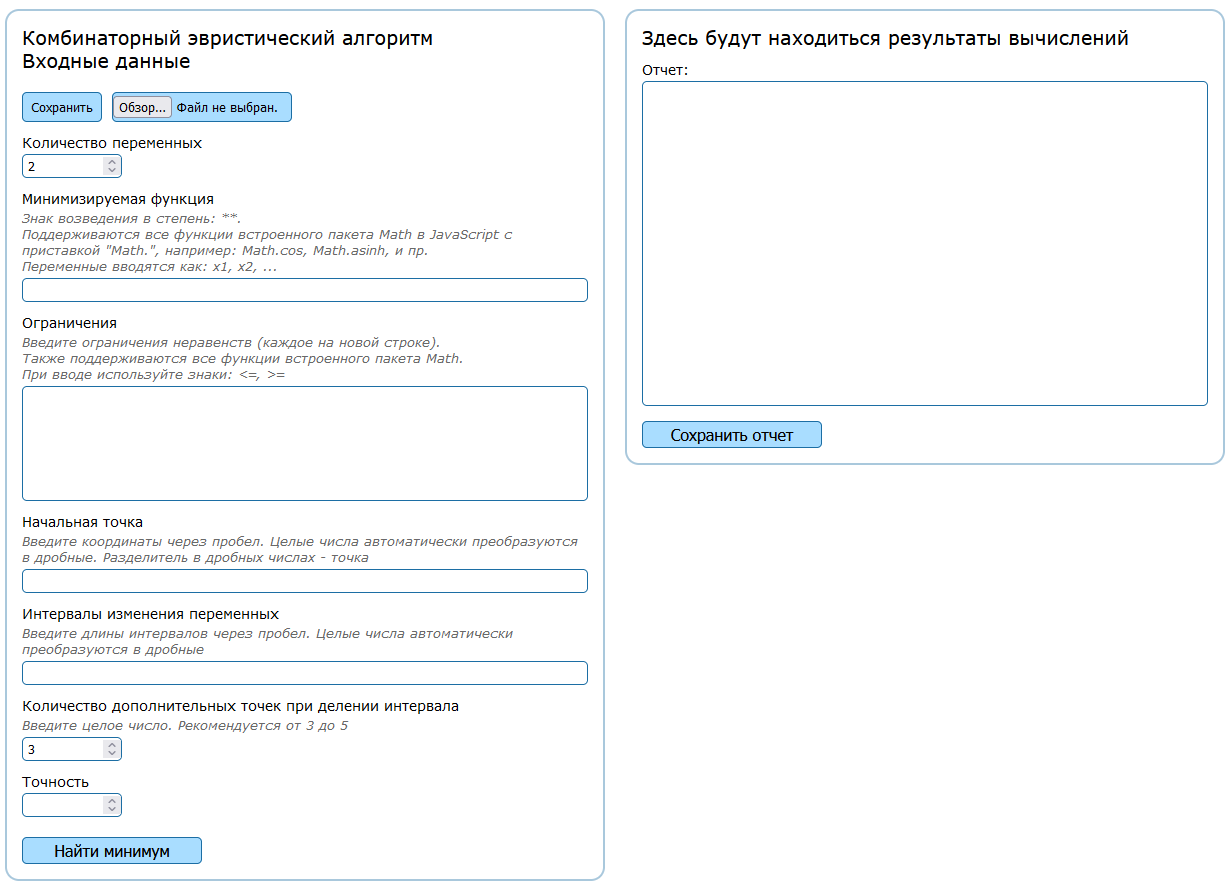


Рисунок 2 – Открытие веб-страницы

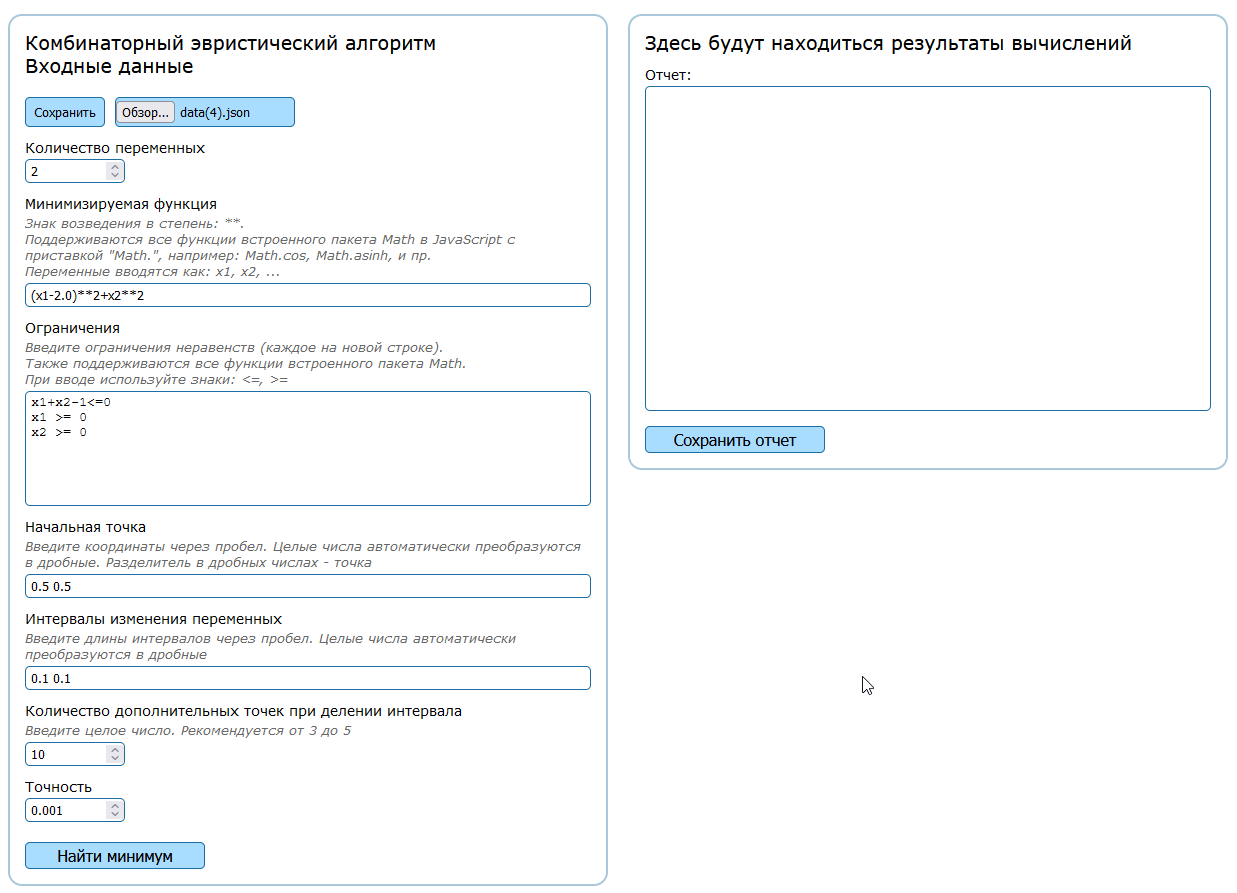


Рисунок 4 – Ввод данных из файла

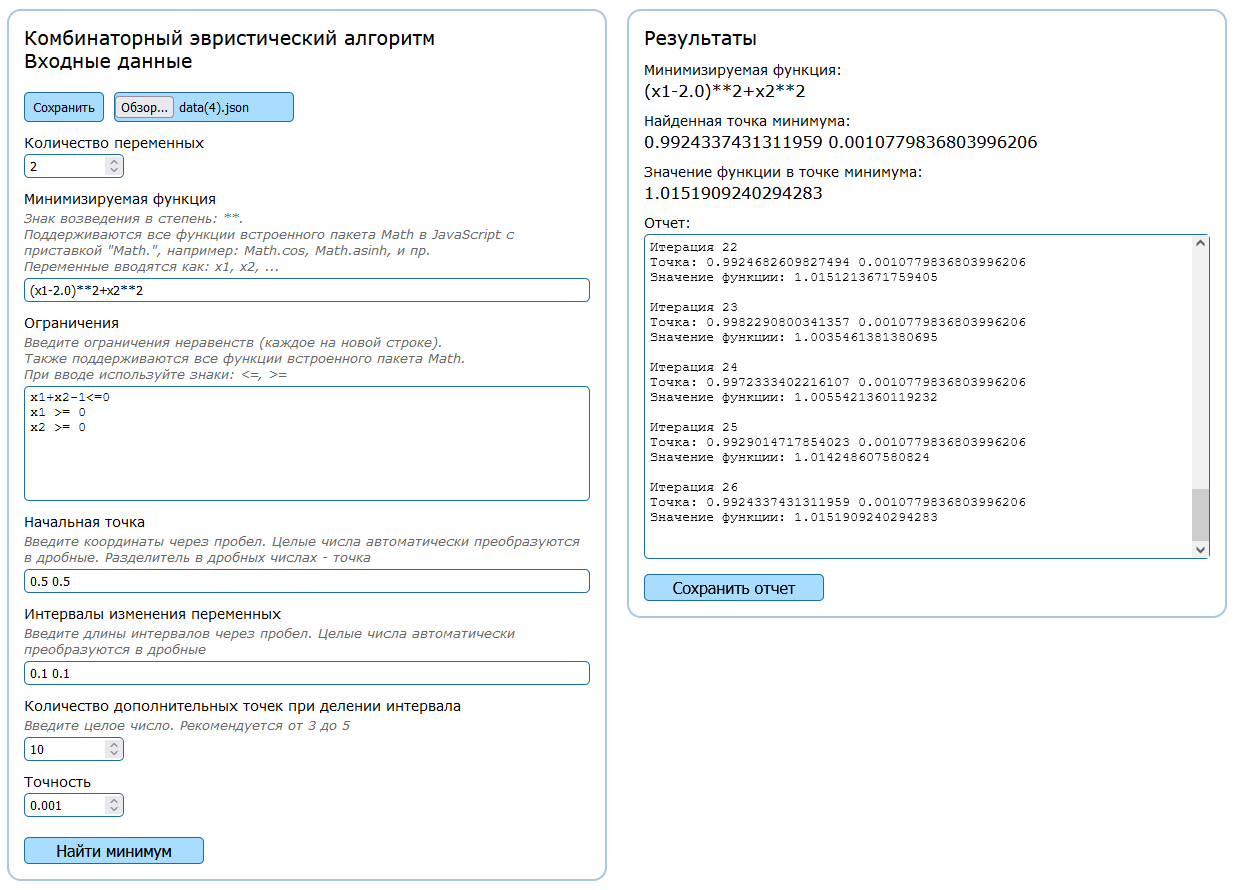


Рисунок 5 – Результат вычислений после нажатия на кнопку 10