ПРИМЕНЕНИЕ ДИСТАНЦИОННОГО ПРАКТИКУМА ПО ПРОГРАММИРОВАНИЮ ДЛЯ ПРОВЕРКИ РЕШЕНИЙ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ЭКОНОМИКИ**[[1]](#footnote-1)**

Андрианов Игорь Александрович,

доцент, кандидат технических наук, доцент,

Вологодский государственный университет, г. Вологда

**Аннотация**. В статье рассматривается использование системы дистанционного практикума по программированию для автоматизации проверки задач курса «Исследование операций». Описаны особенности интеграции дистанционного практикума с системой математических вычислений GNU Octave и языком программирования Python. Приведён пример использования полученных результатов.

**Ключевые слова**: исследование операций, GNU Octave, Python, электронное обучение, дистанционное обучение.

В последние годы электронное и дистанционное обучение активно развивается. При этом значительный скачок произошёл в связи с недавней пандемией, когда большинству студентов нашей страны пришлось начать учиться дистанционно, а вузам – срочно организовывать дистанционное обучение сразу по всем учебным предметам.

Дистанционная электронная форма обучения имеет как свои плюсы, так и минусы. Одним из плюсов является возможность предлагать курсы не только для собственных студентов, но и делать их публичными – бесплатно или за небольшую плату. Многие вузы делают это уже давно: в настоящее время на платформах Stepik, Coursera, OpenEdu, а также на собственных площадках вузов можно найти сотни учебных курсов, доступных для всех желающих.

Однако, при создании таких курсов возникает проблема проверки заданий практических и лабораторных работ: ведь при большом количестве студентов один преподаватель физически не способен это делать. Решением проблемы является организация автоматической проверки.

Заметим, что автоматическая проверка полезна и для небольших групп: студенты могут сдавать работы в любое время и тут же узнавать результат, а преподаватель избавляется от рутинной работы, и может освободившееся время посвятить индивидуальным занятиям с обучающимися или научной работе.

В данной статье рассматривается реализация автоматической проверки заданий для дисциплины «Исследование операций». Данная дисциплина входит в учебные планы как «программистских», так и «экономических» направлений подготовки. Важной частью курса является применение методов математического программирования – аналитических и численных методов решения задач на нахождение наибольшего или наименьшего значения функции нескольких переменных при наличии ограничений на область изменения переменных. Особое применение такие задачи находят в экономике, так как многие величины, встречающиеся здесь (например, производительность труда, прибыль, издержки производства и др.) зависят от многих других величин – переменных. Практически всегда на эти переменные наложены некоторые ограничения. Требуется спланировать работу экономической системы так, чтобы добиться экстремального значения некоторой функции при условии соблюдения ограничений, наложенных на переменные [1].

Студенты, прошедшие курс, должны научиться выполнять формализацию задачи (то есть перевод её из исходного текстового вида на язык формул), а затем решать её, используя специализированное программное обеспечение. Отметим, что в результате формализации задачи достаточно часто получается математическая задача линейного (в том числе целочисленного) либо нелинейного программирования.

Для решения подобных задач существует целый ряд программных пакетов. Из наиболее популярных отметим MATLAB и его открытый аналог GNU Octave, а также язык программирования Python, который достаточно прост в освоении и имеет большое количество дополнительных библиотек. Для того, чтобы автоматизировать проверку заданий, выполненных в среде Octave и на языке Python, мы добавили поддержку данного ПО в систему дистанционного практикума по программированию, разработанную на кафедре АВТ ВоГУ. Приведём вначале краткое описание данной системы.

Дистанционный практикум по программированию разработан и используется на кафедре АВТ уже в течение нескольких лет, он доступен по адресу [atpp.vstu.edu.ru/acm](http://atpp.vstu.edu.ru/acm). При проверке решений используется следующая концепция. Решение задачи (программа пользователя) рассматривается в виде чёрного ящика. Автоматизированная проверяющая система компилирует и запускает решение пользователя с различными (неизвестными студенту) входными данными и сравнивает результат с верными данными. Решение считается верным, если пройдены все тесты. Возможна также настройка системы на выставление частичных баллов для конкретных задач – в этом случае участник может получить от 0 до 100 баллов в зависимости от количества верных ответов, выданных его программой.

Система имеет ряд функциональных возможностей. В частности, она может формировать и показывать подсказки в случае, если присланное решение оказалось неверным. По каждой задаче система отбирает лучшие решения по критерию быстродействия и объёму использованной памяти. Участник, решивший задачу, получает доступ к лучшим решениям других студентов и может сравнить с ними своё решение. Такой соревновательный момент дополнительно повышает интерес к занятиям. Поддерживается автоматическое сравнение одной и той же задачи на предмет плагиата. Также в системе ведётся полная история работы каждого учащегося, реализована возможность генерации различных отчётов [2].

Чтобы добавить в данную систему поддержку программного пакета GNU Octave, потребовалось разобраться, возможно ли работать с Octave через интерфейс командной строки, а также каким образом выполняется ввод/вывод через стандартные потоки. Изучение документации и эксперименты показали, что для работы с Octave в командной строке применяется специализированная утилита octave-cli, при этом файл с исходным кодом (решением задачи) обязан иметь расширение “.m”. Для работы с потоками ввода-вывода можно использовать функции scanf и printf.

Язык Python уже был нами добавлен в систему ранее. Для поддержки курса «Исследование операций» потребовалось лишь установить дополнительные математические библиотеки numpy и scipy.

Рассмотрим пример конкретной задачи, которая давалась студентам в курсе. Имеется три склада с запасами продукции на них *S*1, *S*2 и *S*3 тонн. Эту продукцию нужно доставить трём потребителям, потребности которых составляют *D*1, *D*2 и *D*3 тонн. Составить план перевозок, при котором затраты на перевозку минимальны, а все потребности удовлетворены. Тарифы перевозок заданы матрицей *C*, где *C*ij – стоимость доставки одной тонны груза со склада *i* потребителю *j*.

*Входные данные*. Первая строка входных данных содержит три вещественных числа *S*1, *S*2, *S*3. Вторая строка входных данных содержит три вещественных числа *D*1, *D*2, *D*3 (где 1 ≤ *D*1, *D*2, *D*3 ≤ 1000). Следующие три строки содержат по три вещественных числа – элементы матрицы *C*.

*Выходные данные*. Вывести одно вещественное число – наименьшую суммарную стоимость перевозок (с погрешностью не более 10-4).

В таблице 1 приведены примеры студенческих решений данной транспортной задачи на GNU Octave и на языке Python.

Таблица 1 – Примеры студенческих решений транспортной задачи на разных языках

|  |  |
| --- | --- |
| Пример студенческого решения на Octave | Пример студенческого решения на Python |
| s = scanf("%f", 3);  d = scanf("%f", 3);  c = [scanf("%f", 3); scanf("%f", 3);  scanf("%f", 3)];  A = [  eye(3), eye(3), eye(3);  ones(1, 3), zeros(1, 6);  zeros(1, 3), ones(1, 3), zeros(1,3);  zeros(1, 6), ones(1, 3);  ];  b = [d; s];  [xopt, fopt, err, extra] = glpk(c, A, b, zeros(9, 1), [d; d; d], "SSSUUU", "CCCCCCCCC", 1);  printf("%f", fopt); | from scipy.optimize import linprog  b\_ub = list(map(float,input().split()))  b\_eq = list(map(float,input().split()))  T = [list(map(float,input().split())) for i in range(3)]  c = [T[0][0], T[0][1], T[0][2],  T[1][0], T[1][1], T[1][2],  T[2][0], T[2][1], T[2][2]]  A\_ub = [[1, 1, 1, 0, 0, 0, 0, 0, 0],  [0, 0, 0, 1, 1, 1, 0, 0, 0],  [0, 0, 0, 0, 0, 0, 1, 1, 1]]  A\_eq = [[1,0,0,1,0,0,1,0,0],  [0, 1, 0, 0, 1, 0, 0, 1, 0],  [0, 0, 1, 0, 0, 1, 0, 0, 1]]  res = linprog(c, A\_ub, b\_ub, A\_eq, b\_eq)  print(res.fun) |

В обоих случаях студенты свели данную задачу к задаче линейного программирования и воспользовались готовой функцией для её решения. Интересно, что данные решения были написаны и сданы в проверяющую систему в конце марта 2020 года, то есть когда ВоГУ уже перешёл на дистанционную форму обучения.

На рисунке 1 показан пример отчёта системы о верных решениях данной задачи. Из отчёта видно, кто именно уже сдал задачу, а также в каких решениях есть подозрение на плагиат (показано красным).



Рисунок 1 – Пример отчёта системы о проверенных решениях

Таким образом, опыт применения полученных результатов показал, что их можно успешно использовать при реализации дистанционного электронного обучения для автоматизации проверки заданий по курсу «Исследование операций», а также для похожих курсов (например, «Математическое моделирование»).

Список цитируемых источников

1. Исследование операций: методические указания к лабораторным работам: [направления: 09.03.01 - Информатика и вычислительная техника, 09.03.04 - Программная инженерия] Ч. 1 / сост.: И.А. Андрианов, О.В. Авдеева, О.И. Микрюкова. – Вологда: ВоГУ, 2016. – 35 с.

2. Андрианов, И.А. Архитектура электронного ресурса для обучения информационно-коммуникационным технологиям / И.А. Андрианов, И.А. Петров, С.Ю. Ржеуцкая // материалы шестой Всероссийской научно-технической конференции «Вузовская наука – региону». – Вологда: Вологодский государственный технический университет, 2008. – С. 44-47.

1. Работа выполнена при поддержке РФФИ (проект №19-01-00103). [↑](#footnote-ref-1)