

УДК 657.1:004

**РАЗРАБОТКА ПРОГРАММЫ ДЛЯ НАХОЖДЕНИЯ ОПТИМАЛЬНОГО РАСПРЕДЕЛЕНИЯ РЕСУРСОВ С ЦЕЛЬЮ МАКСИМИЗАЦИИ ПРИБЫЛИ**

**Ильичев В.Ю.,**

*к.т.н., доцент,*

*Калужский филиал ФГОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»,*

*Калуга, Россия*

**Юрик Е.А.,**

*к.т.н., доцент,*

*Калужский филиал ФГОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»,*

*Калуга, Россия*

**Аннотация**

Описывается исследование, ставящее целью создание средств для применения универсального метода оптимизации, метода дифференциальной эволюции, при решении одной из наиболее распространённых экономических задач – задачи оптимального распределения имеющихся ресурсов. Для использования данного метода предлагается использовать программу на свободно распространяемом языке Python, с рядом специально подобранных библиотек функций, предназначенных для обеспечения необходимого функционала.

Разобраны отличия описываемого метода от ранее применяемых, а также предоставляемые им дополнительные возможности для пользователя. Разработана и описана последовательность команд, необходимых для

подключения библиотек, осуществления ввода ограничений и целевой (оптимизируемой) функции прибыли, организации вывода результатов расчёта. Решение экономической задачи оптимального распределения ресурсов продемонстрировано наглядным примером, ход производимого расчёта представлен в графическом виде.

Сделаны выводы о преимуществах применения средств языка Python к обработке экономических данных, даны рекомендации по применению метода дифференциальной эволюции при решении прочих экономических задач. Кратко описаны планы по дальнейшему развитию разработанной методики в наиболее передовых экономических исследованиях.

**Ключевые слова:** экономические показатели, прибыль, распределение ресурсов, целевая функция, метод дифференциальной эволюции, язык Python.

### ***CREATING ORGANIZATION REVENUE REPORTS USING PYTHON***

***Ilichev V.Y.,***

*PhD, Associate Professor,*

*Kaluga Branch of Bauman Moscow State Technical University,*

*Kaluga, Russia*

***Yurik E.A.,***

*PhD, Associate Professor,*

*Kaluga Branch of Bauman Moscow State Technical University,*

*Kaluga, Russia*

#### **Annotation**

A study is described aimed at creating means for applying the universal optimization method, the method of differential evolution, when solving one of the most common economic problems - the problem of optimal distribution of available resources. To use

this method, it is proposed to use the program in the freely distributed language Python, with a number of specially selected function libraries designed to provide the necessary functionality.

The differences between the described method and the previously used ones, as well as the additional capabilities for the user provided to them, are analyzed. A sequence of commands necessary for connecting libraries, entering restrictions and the target (optimized) profit function, organizing the output of calculation results has been developed and described.

The solution of the economic problem of optimal distribution of resources is demonstrated by a clear example, the course of the calculation is presented in graphic form.

Conclusions were drawn on the advantages of applying Python language tools to the processing of economic data, recommendations were made on the use of the differential evolution method in solving other economic problems. Plans for the further development of the methodology in the most advanced economic research are briefly described.

**Keywords:** economic indicators, profit, resource allocation, objective function, differential evolution method, Python language.

**Введение.** В круг основных задач, решаемых в экономике, входят задачи оптимизации [1]. Их целью является нахождение таких параметров экономических систем, при которых достигается максимальная прибыль или минимальные затраты (функция, экстремум которой необходимо достичь, называется целевой функцией параметров).

Существует множество методов оптимизации, каждый из которых применим при нахождении решения для определённого класса систем [2]. К ним относятся, например, симплекс-метод, метод эллипсоидов, методы внутренней точки (методы линейного программирования [3], оперирующие системами

линейных уравнений и неравенств), метод неопределённых множителей Лагранжа, метод ветвей и границ (методы нелинейного программирования) [4].

Перечисленные методы являются «классическими», чисто математическими. Однако, в последнее время появились новые приёмы оптимизации, позволяющие использовать совершенно иной подход – использование аналогий, связанных с законами развития сложных биологических систем. Наиболее известными из них являются нейросетевое программирование (имитирующее работу мозга) [5] и эволюционные алгоритмы (реализующие законы эволюции живых организмов) [6].

Создание и обучение нейросетей является сложным и длительным процессом, причём для каждой системы и решаемой задачи необходимо подбирать наилучшую топологию нейронной сети, поэтому данный метод нельзя считать универсальным. Класс эволюционных методов уже может претендовать на универсальность, поскольку одними и теми же приёмами решаются задачи оптимизации разных типов [7].

Для одной из наиболее популярных сейчас сред программирования Python существует специальная библиотека функций для решения задач методом дифференциальной эволюции (одним из самых совершенных эволюционных методов) [8].

**Цель исследования.** Метод дифференциальной эволюции классифицируется как стохастический, поскольку его алгоритм предполагает использование случайных чисел. Этим обусловлено его большое преимущество перед остальными методами оптимизации – элемент случайности позволяет находить глобальные, а не локальные экстремумы оптимизируемой (целевой) функции. Случайными (в пределах заданного диапазона) являются начальные значения переменных, кроме того в процессе работы алгоритма переменные подвергаются «мутации» - осуществляется их принудительное отклонение от найденных текущих значений [9].

Метод дифференциальной эволюции перспективен тем, что с его помощью можно оптимизировать нелинейные, не дифференцируемые целевые функции. Также в задачах можно использовать большое количество переменных и ограничений, чем он выгодно отличается от прочих методов. Вычисления данным методом можно распараллелить на несколько процессоров.

Реализация данного метода для языка Python представлена в виде функции `Differential_evolution` библиотеки математических методов `Scipy` [10]. Данная функция позволяет осуществлять настройку её параметров (например, можно выбрать один из 12 алгоритмов поиска решения), хотя для простых задач показывает хорошие результаты и на параметрах по умолчанию. С особенностями организации дифференциальной эволюции можно ознакомиться в специальной литературе [8, 11], наша же задача - разработать процесс его практического использования для решения конкретной задачи оптимизации.

Таким образом, целью данного исследования является разработка методики и программы для решения одной из часто встречающихся экономических задач – задачи оптимального использования ресурсов с целью достижения максимальной прибыли [12] с использованием метода дифференциальной эволюции и вывод в графическом виде результатов работы написанной программы, наглядно показывающих особенности применённого метода.

**Материал и методы исследования.** Решаемую задачу можно сформулировать следующим образом. Предприятие имеет в своем распоряжении определенное количество ресурсов: рабочую силу, деньги, сырье, оборудование, производственные площади и т.п. Допустим, что имеются ресурсы трех видов: рабочая сила, сырье и оборудование, – в количестве соответственно 80 (чел./дней), 480 (кг) и 130 (станко/ч). Предприятие выпускает продукцию 4 видов. Информация о количестве единиц каждого ресурса, необходимом для производства единицы объёма продукции каждого вида, и доходах, получаемых предприятием от продажи единицы объёма каждого вида товаров, приведена в таблице 1.

Таблица 1. Исходные данные по ресурсам.

Ресурсы	Нормы расхода ресурсов на единицу изделия (детали)				Наличие ресурсов
	Тип 1	Тип 2	Тип 3	Тип 4	
Труд	7	2	2	6	80
Сырье	5	8	4	3	480
Оборудование	2	4	1	8	130
Доход от единицы	3	4	3	1	

Требуется найти такой план выпуска продукции, при котором будет максимальным доход от продажи всей выпущенной продукции.

Объём продукции каждого типа обозначим как  $x_1, x_2, x_3, x_4$ .

Доход предприятия, который необходимо максимизировать, является целевой функцией, и исходя из известного дохода от единицы объёма реализации каждого вида продукции его можно выразить

$$f(x) = 3x_1 + 4x_2 + 3x_3 + 1x_4.$$

Ограничения по ресурсам (исходя из табл. 1):

$$7x_1 + 2x_2 + 2x_3 + 6x_4 \leq 80;$$

$$5x_1 + 8x_2 + 4x_3 + 3x_4 \leq 480;$$

$$2x_1 + 4x_2 + x_3 + 8x_4 \leq 130.$$

Исходя из того, что объём выпускаемой продукции не может быть отрицательным числом, определяем дополнительные ограничения:

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0.$$

Разработанная программа на языке Python для решения задачи оптимального распределения ресурсов с помощью метода дифференциальной эволюции состоит из следующих блоков:

1. Импорт функций из математической библиотеки `Scipy.optimize`: `NonlinearConstraint` – используется для ввода представленных выше ограничений, `Bounds` – для задания области изменения каждой переменной, `Differential_evolution` – собственно, для использования метода дифференциальной эволюции.

2. Импорт дополнительной библиотеки Numpy [13] для работы с матрицами (заполняемыми текущими значениями переменных и целевой функции для дальнейшего построения графиков) и модуля Matplotlib [14] – для вывода результатов расчётов в графической форме.
3. Создание матриц, в которые после каждой итерации заносятся значения переменных и целевой функции.
4. Использование лямбда-функций, являющихся левыми частями накладываемых ограничений задачи [15].
5. Определение крайних допустимых значений ограничений по ресурсам с помощью функций NonlinearConstraint (в данном случае нижний предел ограничений равен нулю, а верхние пределы являются числами, указанными в правых частях ограничений по ресурсам).
5. Создание блока целевой функции.
6. Определение границ изменения каждой переменной с использованием функции Bounds из модуля Scipy. В данной задаче объёмы выпускаемой продукции не могут быть отрицательными, поэтому нижние пределы равны нулю, верхние пределы можно варьировать в процессе запуска решений, - оказалось, что в данном случае эти значения не могут превышать 200, поэтому это число и задано как верхний предел области допустимых значений каждой переменной.
7. Запуск функции Differential\_evolution, в параметрах которых указывается имя заданной выше целевой функции и даются ссылки на переменные и определённые для них ограничения. Вычисленные на каждой итерации значения переменных и целевой функции заносятся в созданные матрицы.
8. Предусматриваются команды построения графиков зависимости значений переменных и целевой функции от номера итерации. После решения задачи (нахождения максимума целевой функции) конечные значения этих параметров выводятся также в числовом виде на экран.

Максимальное количество итераций в функции `Differential_evolution` выбрано равным 1000 (что с запасом хватает для решения поставленной задачи), точность решения выбрана 0,001, что также является достаточным.

Код данной программы для Python занимает всего 40 строк, и может быть применён любым желающим для решения широкого круга экономических оптимизационных задач (его можно свободно скачать с сайта автора [16]).

**Результаты.** Примеры графического отображения процесса изменения вычисляемых переменных ( $x_1 - x_4$ ) при двух запущенных расчётах приведён на рис. 1. Вывод и оформление этих графиков осуществлён с использованием специальной библиотеки для работы с графическими данными Matplotlib для Python.

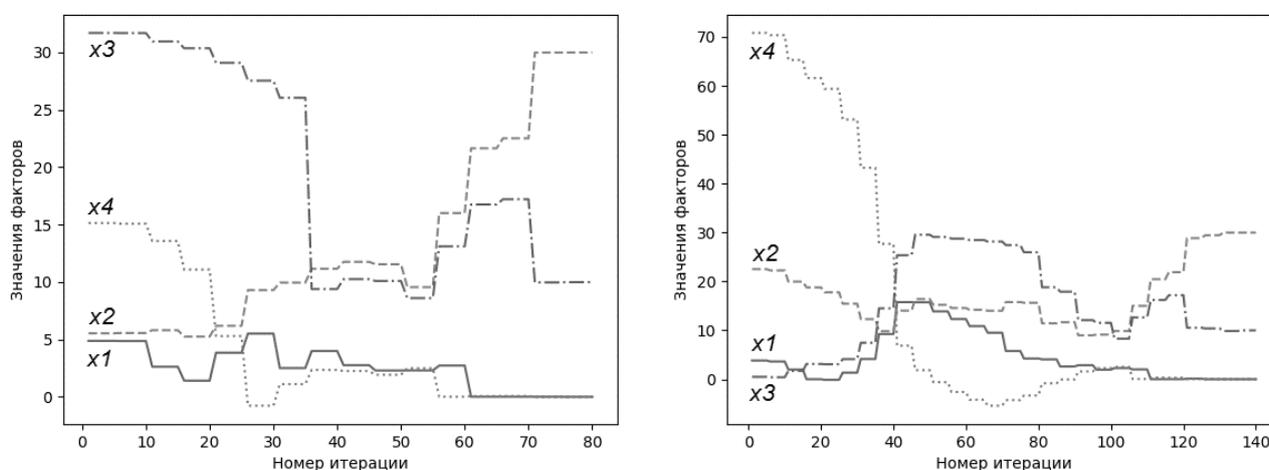


Рис. 1 – Графические зависимости изменения значений переменных (объёмов производства продукции  $x_1 - x_4$ ) в зависимости от номера итерации.

По графикам видно, что промежуточные значения переменных при каждом запуске расчёта могут сильно отличаться (тем сильнее, чем больше отклоняются начальные случайные значения переменных от оптимальных). Однако, получаемый результат расчёта (эволюционного процесса) всегда оказывается одинаковым. В данном случае значения переменных, обеспечивающие максимум целевой функции прибыли, составляют в конце расчёта:

$$x_1 = 0, x_2 = 10, x_3 = 30, x_4 = 0.$$

При первом запуске расчёта начальные значения переменных не так сильно отличаются от оптимальных, как при втором запуске. Благодаря этому в первом случае решение было найдено быстрее – за 80 итераций, по сравнению со 140 итерациями, которые понадобились во втором случае.

Результаты итераций, получаемых при использовании метода дифференциальной эволюции, называют популяциями. Из рис. 1 наглядно видно, что промежуточные популяции могут очень сильно изменяться, как и путь к решению задачи, зависящий от выбранной стратегии поиска. Данный путь при каждом запуске программы зависит от начальных значений переменных, выбранных программой случайно в рамках указанного диапазона. Более того, он зависит и от случайных мутаций, вносимых алгоритмом дифференциальной эволюции на каждом шаге расчёта. Решение считается найденным, когда мутации не оказывают существенного влияния на популяцию (целевая функция при каждой следующей итерации меняется только в диапазоне, определяемом указанной в функции `Differential_evolution` точностью решения).

Приведённые на рис. 2 графики показывают изменение текущих значений целевой функции  $f(x)$  в зависимости от номера итерации для тех же запусков программы, что и на рис. 1.

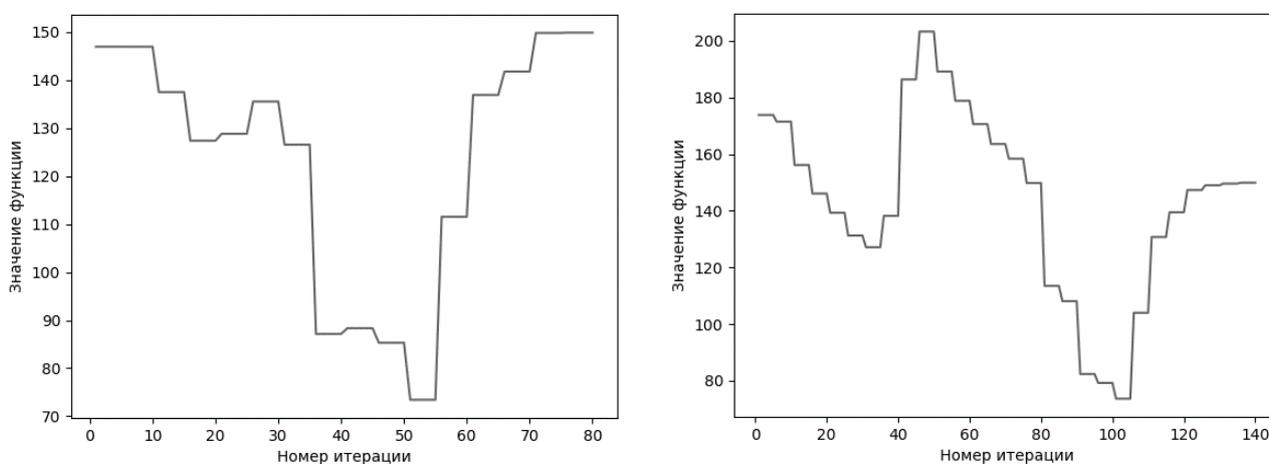


Рис. 2. Изменение целевой функции в зависимости от номера итерации.

По изменению целевой функции ещё более наглядно, как сильно меняются результаты итераций в процессе расчёта, но результат получается всегда одинаковым – в данном случае найденная максимальная прибыль составляет 150 денежных единиц. Вычисление прекращается, когда целевая функция перестаёт изменяться. Очень интересен тот факт, что на втором графике рис. 2 на некоторых популяциях целевая функция превышает найденную в результате максимальную прибыль, но расчёт продолжается, - это означает, что на данных популяциях не обеспечивалось выполнение всех установленных ограничений.

Исследованная в данной работе целевая функция является линейной, а для такого случая существуют и другие программы поиска оптимального решения, например, программа Eureka фирмы Borland, использующая методы линейного программирования. Рассматриваемая задача, решённая с её помощью, показала решение, идентичное представленному. Это указывает на правильность разработанной методики использования метода дифференциальной эволюции. При этом следует напомнить, что данный метод позволяет решать не только линейные задачи оптимизации.

**Обсуждение.** Цель исследования, состоящая в создании и апробации программы, решающей экономическую задачу оптимального распределения ресурсов с применением метода дифференциальной эволюции, полностью достигнута. Выбраны необходимые для реализации метода библиотеки и функции языка программирования Python, подобраны подходящие для решения задачи параметры. Немаловажно, что для наглядной демонстрации хода поиска решения удалось вывести графики изменения как значений переменных, так и значения целевой функции.

Метод дифференциальной эволюции показал простоту реализации на Python, быстрый поиск решения, гибкость подбора (в случае необходимости) дополнительных опций функции `Differential_evolution`, высокую точность получаемых результатов. Рассматриваемый в статье метод сильно отличается от других методов оптимизации также тем, что способен находить оптимальное

решение даже при исследовании очень сложных функций, со многими локальными экстремумами, и даже имеющими области неопределённости.

Код программы размещён на сайте автора для свободного скачивания, изучения и выполнения любых видов модификации и совершенствования [16]. С небольшими изменениями его можно с успехом использовать для решения широкого круга других экономических задач. Предприятия и организации могут использовать рассмотренную методику для автоматизации рутинных расчётов на основе актуального подхода.

**Заключение.** По результатам проведённой работы можно сделать следующие выводы:

- рассмотренный метод отличается высоким совершенством и поэтому его можно рекомендовать для решения многих экономических оптимизационных задач;
- код созданной программы отличается простотой и должен быть понятен даже начинающему программисту;
- возможен вывод как промежуточных, так и окончательных результатов оптимизации экономической системы как в текстовом, так и в графическом виде высокого качества.

Описанное исследование посвящено решению очень часто встречающейся в экономике задачи оптимального распределения ресурсов. При этом, к сожалению, следует констатировать, что на практике пока редко используется строго научный подход при расходовании денежных средств и других ресурсов, что приводит к неоправданно высоким затратам на инвестиционные проекты. Предлагаемый же метод позволяет на любом этапе подготовки и исполнения проекта быстро и точно перераспределить имеющиеся ресурсы с целью достижения наиболее высоких экономических показателей создаваемой или управляемой системы. Поэтому в дальнейшем предполагается использование разработанного метода дифференциальной эволюции для решения насущной проблемы выбора оптимальных предложений по поставке товаров и услуг, по

выполнению работ с учётом заранее оговоренных заказчиком ограничений (в процессе разыгрывания тендеров).

Также описанная работа доказывает высокую целесообразность использования языка Python для проведения исследований в экономической области, тем более, что описана лишь малая часть его возможностей.

### **Библиографический список:**

1. Ильичев В.Ю., Чухраев И.В., Чухраева А.И. Решение задачи перераспределения потоков газа на магистральных газопроводах методами линейного программирования. // Научные технологии. - 2020. - Т. 21. - № 1. - С. 11-17.
2. Ильичев В.Ю., Юрик Е.А., Антипов В.С. Оптимизация перераспределения потоков на магистральных газопроводах. // Научное обозрение. Технические науки. - 2019. - № 4. - С. 22-26.
3. Peng J., Roos C., Terlaky T. Новый и эффективный метод внутренних точек для линейной оптимизации. // Computational Technologies. - 2001. - Т. 6. - № 4. - С. 61-80.
4. Djukic R.R. Partial stability of multi attribute decision-making solutions for interval determined criteria weights - the problem of nonlinear programming. // Military Technical Courier. - 2020. - Т. 68. - № 3. - С. 488-529.
5. Ильичев В.Ю. Использование парсинга для создания базы метеорологических данных и разработка на её основе нейросетевой модели прогнозирования скорости ветра. // Системный администратор. - 2020. - № 10 (215). - С. 92-95.
6. Загинайло М.В., Фатхи В.А. Генетический алгоритм как эффективный инструмент эволюционных алгоритмов. // Инновации. Наука. Образование. - 2020. - № 22. - С. 513-518.
7. Ганцева Е.А. Эволюционный подход к структурному синтезу системы поисковой оптимизации. // Системы управления и информационные технологии. - 2008. - № 3 (33). - С. 51-55.

8. Mitrofanov S.A., Karaseva T.S. On efficiency of the differential evolution method. // Молодежь. Общество. Современная наука, техника и инновации. - 2019. - № 18. - С. 278-280.
9. Куликова Н.Л., Денисов А.А. Тестирование арифметических отношений в языках программирования по методу слабой мутации. // Ученые записки Российского государственного социального университета. - 2012. - № 9 (109). - С. 82-85.
10. Слащев И.С. Сравнение пакетов анализа данных Matlab, Scipy, SAS, SPSS. // Modern Science. - 2019. - № 11-2. - С. 257-259.
11. Газизова О.Р. Аспекты реализации алгоритма дифференциальной эволюции на языке Python. В сборнике: Информационные технологии в процессе подготовки современного специалиста. Межвузовский сборник научных трудов. - Липецк, - 2017. - С. 13-17.
12. Марданов Р.Ш., Султанов Р.А., Фатыхов А.Г. Стохастический подход к задаче об оптимальном использовании ресурсов. // Вестник Казанского государственного финансово-экономического института. - 2009. - № 2 (15). - С. 44-47.
13. Бирюков В.Е., Ликсин С.С. Использование библиотеки Numpy для повышения эффективности программного кода на языке Python. В сборнике: Новые информационные технологии и системы. Сборник научных статей XVI Международной научно-технической конференции. - 2019. - С. 159-161.
14. Смагин А.Н. Визуализация данных в Python при помощи библиотеки Matplotlib. В сборнике: Проблемы управления в социально-экономических и технических системах. Сборник научных статей Материалы XV Международной научно-практической конференции. - 2019. - С. 39-43.
15. Романенко Р.А., Стухальский А.Л., Прихожий А.А. Использование языков высшего уровня для решения прикладных задач. // Математические методы в технике и технологиях - ММТТ. - 2018. - Т. 9. - С. 97-100.

16. Решение задачи оптимального распределения ресурсов с помощью метода дифференциальной эволюции. [Электронный ресурс]. - Режим доступа - URL: <http://turbopython.ru/DEV> (дата обращения: 07.05.2021).

*Оригинальность 75%*