

## Нахождение максимума функции с помощью алгоритма поиска косяком рыб

*Маринчук Александр Сергеевич*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

*Баженов Руслан Иванович*

*Приамурский государственный университет им. Шолом-Алейхема*

*к.п.н., доцент, зав. кафедрой информационных систем, математики и методик обучения*

### Аннотация

В данной статье описывается метод решения задачи на нахождение максимального объема коробки, вырезанной из листа жести, используя алгоритм поиска косяком рыб с помощью языка программирования PHP.

**Ключевые слова:** PHP, максимум функции, алгоритм поиска косяком рыб.

### Finding the maximum of the function using a artificial fish school algorithm

*Marinchuk Alexander Sergeevich*

*Sholom-Aleichem Priamursky State University*

*Student*

*Bazhenov Ruslan Ivanovich*

*Sholom-Aleichem Priamursky State University*

*Candidate of pedagogical sciences, associate professor, Head of the Department of Information Systems, Mathematics and teaching methods*

### Abstract

This article describes the method of solving the problem of finding the maximum volume of a box, cut from a sheet of sheet metal, using the search algorithm fish school search using the PHP programming language.

**Keywords:** PHP, maximum of function, artificial fish school algorithm

Существует множество способов для решения оптимизационных задач: метод линейного программирования, генетические методы, симплекс метод и другие. Целью таких задач является нахождение экстремума (максимума или минимума) функции, которые дадут наиболее оптимальный результат в решении таких задач.

Цель данного исследования заключается в рассмотрении способа решения задачи, в которой необходимо найти максимум функции, на основе алгоритма поиска косяком рыб, используя язык программирования PHP.

Описание задачи: «Из прямоугольного листа жести размером 5 сантиметров на 8 сантиметров надо изготовить открытую коробку наибольшего объема, вырезая квадратные уголки так, как показано на рисунке 1» [6].

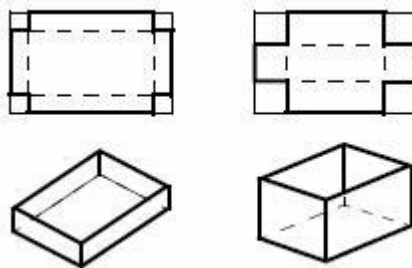


Рисунок 1 – Способ изготовления коробки

В данной задаче необходимо максимизировать объем коробки, полученной при вырезании из листа жести с помощью алгоритма поиска косяком рыб [1].

В статье А.Н. Куликова подробно рассказывается об алгоритме поиска косяком рыб и показывается эффективность данного метода [1]. В.А.Частикова и др. разработали алгоритм поиска косяком рыб и адаптировали метод для программирования движения квадрокоптеров без постоянного контроля со стороны человека с учетом индивидуальных особенностей отдельных агентов [2]. Также они полностью рассмотрели алгоритм поиска косяком рыб и выявили зависимость результатов при нахождении экстремумов функции от вводимых значений [3]. А.И.Галиев, С.Ф.Чермошенцев рассмотрели способ представления решения задачи канальной трассировки, а также эвристики, имитирующие биологические процессы, происходящие при поиске косяком рыб места скопления пищи [4]. В статье В.С.Васильева «Генетические алгоритмы в сравнении с оптимизационными методиками» рассматриваются основные понятия генетических алгоритмов, применение генетических алгоритмов к анализу и прогнозированию социально-экономических систем [5]. М.М.Егин рассматривает принцип работы алгоритма косяка рыб в многомерной глобальной оптимизации [7]. В статье Дружининой М.А. проведены исследования, показавшие особенности работы алгоритма движения косяка рыб, сформированы рекомендации по выбору значений, оптимальных параметров алгоритма движения косяка рыб, исходя из тонкостей решаемой задачи оптимизации [8]. Х.Т.Ну и др. классифицировали рыбный косяк по нескольким направлениям, и в связи с этим был предложен новый алгоритм поиска косяком рыб [9]. В статье Н.М.Jianga и др. были оптимизированы параметры насоса для усиления сглаженных комбинационных усилителей на основе алгоритма поиска косяком рыб [10].

Для начала рассмотрим решение задачи вручную, без применения алгоритма поиском косяка рыб.

Обозначим через  $X$  длину стороны вырезаемого квадрата. Тогда длины сторон прямоугольника уменьшатся на  $2X$  и объем коробки будет равен:

$$V = x(8 - 2x)(5 - 2x) = 4x^3 - 26x^2 + 40x$$

При этом  $X$  может меняться в следующих пределах:  $0 \leq x \leq 2,5$ . Заметим сразу, что в крайних точках отрезка  $[0; 2,5]$  объем равен нулю. Находим критические точки:  $V' = 12x^2 - 52x + 40$ ,  $V' = 0$  при  $X_1=1$ ,  $X_2=10/3$ . Отметим, что значение  $X_2$  не принадлежит области определения. При  $X = 1$  объем максимален:  $V=18$ .

Теперь реализуем алгоритм поиска косяком рыб на языке программирования PHP. Для начала создадим поля под ввод значений, необходимых для дальнейших вычислений.

Размер популяции

Количество итераций

Верхняя граница поиска  и нижняя граница поиска

Начальный  и конечный радиус поиска пищи вокруг агентов

Максимальный вес агента

Рисунок 2 – Вводимые параметры

Для каждого агента в начале выберем случайное положение в аквариуме и установим вес равный половине максимального.

```
$ran1=($niz+$verx)/2;
$ran=array_fill(0,$raz_pop+1,$ran1);
$array2=array_fill(0,$raz_pop+1, 0);
```

Рисунок 3 – Заносим в массивы вес и положение агентов

Далее идет основной цикл алгоритма, который будет продолжаться столько, сколько было указано в поле «число итераций». Затем внутри цикла вычисляется новое положение каждого агента и подставляется в функцию. Если значение оказалось лучше предыдущего, то положение фиксируется, а иначе считается, что перемещения не было.

```
for ($i=0;$i<=$iter-1;$i++)
{
    for ($j=0;$j<=$raz_pop;$j++){
        echo ' 1 переменная ' . $ves[$j];
        $array4[$j]=$pos2[$j];
        $pos2[$j]=$ves[$j]+ rand(-3, 3);
        echo ' 2 переменная ' . $pos2[$j];
        $array2[$j]=4*pow($ves[$j],3)-26*pow($ves[$j],2)+40*$ves[$j];;
        echo ' Первое значение функции ' . $array2[$j];
        $array1[$j]=4*pow($pos2[$j],3)-26*pow($pos2[$j],2)+40*$pos2[$j];
        echo ' Второе значение функции ' . $array1[$j];
        if ($array1[$j]<$array2[$j]){
            $pos2[$j]=$array4[$j];
        }
        else{
            $array4[$j]=$pos2[$j];
        }
    }
}
```

Рисунок 4 – Новое положение агентов и вычисление значений функции

Теперь вычисляем вес каждой рыбы, который равняется текущему весу агента плюс отношение изменения функции к максимальному изменению среди всей популяции.

```
$array3[$j]=$array1[$j]-$array2[$j];
```

Рисунок 5 – Вычисляем разность значений функции

```
for($l=0;$l<=$raz_pop;$l++){
    if(($array3[$l]>$max) and ($array3[$l]!=0))
    {
        $max=$array3[$l];
    }
}

for($k=0;$k<=$raz_pop;$k++){
    $array[$k]=($array3[$k]/$max)+$array[$k];
    echo ' Масса ' .$k.' Рыбы ' .$array[$k];
}
```

Рисунок 6 – Находим максимальное изменение функции и вычисляем вес каждого агента

После этого рыбы совершают следующую стадию плавания — инстинктивно-коллективную. Для всего косяка рыб высчитывается величина «Общий шаг миграции». Она равна отношению суммы разницы двух переменных, умноженных на изменение функции, к сумме изменения функции.

```
$m+=$pos2[$j]-$ves[$j]*$array3[$j];
$m1+=$array3[$j];

$m2=$m/$m1;
```

Рисунок 6 – Общий шаг миграции

Затем вся популяция смещается на подсчитанную величину.

```
for($f=0;$f<=$raz_pop;$f++){
    $pos2[$f]= $pos2[$f] + $m2;
```

Рисунок 7 – Смещение всей популяции

Перед следующей операцией надо вычислить центр тяжести всего косяка. Он рассчитывается как отношение суммы произведения положения каждого агента на его вес к сумме всего косяка.

```
$bary+=$pos2[$f]*$array[$f]
$bary1+=$array[$f];

}

$bary3=$bary/$bary1;
```

Рисунок 8 – Центр тяжести косяка

Перед последней операцией следует выяснить как изменился вес косяка в целом. Если он увеличился, значит рыбы находятся в верном положении и следует сузить круг поиска и наоборот, если он уменьшился, то следует расширить круг.

```
for ($n=0;$n<=$raz_pop;$n++) {
if (array_sum($array)>$t) {
$pos2[$n]=$pos2[$n]+(($ves*2)*rand(0,1)*($pos2[$n]-$bary3)/(sqrt(pow($pos2[$n]-$bary3,2))));
}
else {
$pos2[$n]=$pos2[$n]-(($ves*2)*rand(0,1)*($pos2[$n]-$bary3)/(sqrt(pow($pos2[$n]-$bary3,2))));
}
}
```

Рисунок 9 – Изменение положения агентов, зависящее от веса косяка

Теперь протестируем наш алгоритм на функции, которая описана в ручном методе решения нашей задачи. Применим алгоритм поиска косяком рыб для которого требуется ввести ряд значений: область исследования этой функции, а также параметры работы алгоритма. Область поиска обычно сравнивают с аквариумом, где плавает косяк рыб, который находится в поисках пищи (максимума функции).

Введем значения, характеризующие аквариум и агентов.

Размер популяции

Количество итераций

Верхняя граница поиска  и нижняя граница поиска

Начальный  и конечный радиус поиска пищи вокруг агентов

Максимальный вес агента

Рисунок 10 – Вводим необходимые значения

Номер тирации 0 1перемнная 2 2перемнная 2 Первое значение функции 8 Второе значение функции 8 Разница значений функции 0 1перемнная 2 2перемнная 3 Первое значение функции 8 Второе значение функции -6 Разница значений функции -14 1перемнная 2 2перемнная 3 Первое значение функции 8 Второе значение функции -6 Разница значений функции -14 1перемнная 2 2перемнная 2 Первое значение функции 8 Второе значение функции 8 Разница значений функции 0 Масса 0 Рыбы 5 Масса 1 Рыбы 6 Масса 2 Рыбы 6 Масса 3 Рыбы 0 Максимальное значение 8 Номер тирации 1

Рисунок 11 – Первая итерация

Максимальное значение функции, которое удалось найти рыбам это число 8. Не совсем точное значение максимума функции, поэтому переходим на последнюю итерацию.

Масса 3 Рыбы -1.8 Максимальное значение 18 Номер терации 4 1переменная 2  
 2переменная 2 Первое значение функции 8 Второе значение функции 8 Разница  
 значений функции 0 1переменная 2 2переменная 3 Первое значение функции 8 Второе  
 значение функции -6 Разница значений функции -14 1переменная 2 2переменная 3  
 Первое значение функции 8 Второе значение функции -6 Разница значений функции  
 -14 1переменная 2 2переменная 1 Первое значение функции 8 Второе значение  
 функции 18 Разница значений функции 10 Масса 0 Рыбы 6.6 Масса 1 Рыбы 5.6 Масса 2  
 Рыбы 7.6 Масса 3 Рыбы -0.8 Максимальное значение 18

Рисунок 12 – Последняя итерация

Как видно из рисунка 12 максимум функции найден, равен 18 . При аналитическом решении задачи тоже было получено значение 18. Видим, что значения совпадают.

Таким образом максимальный объем, который может иметь коробка, которую вырежут из листа жести размером 5 см. на 8 см. , равен  $18 \text{ см}^3$ .

В статье было рассмотрено решение задачи, в которой требовалось найти максимум функции с помощью алгоритма поиска косяком рыб на языке программирования PHP.

### Библиографический список

1. Куликов А.Н. Программа оптимизации, инспирированная поведением косяка рыб // Инноватика. 2014. №1. С. 33-42.
2. Частикова В.А., Дружинина М.А., Кекало А.С. Адаптация алгоритма поиска косяком рыб для оптимизации движения квадрокоптеров // Научные труды Кубанского государственного технологического университета. 2014. №S6. С. 97-100.
3. Частикова В.А., Дружинина М.А., Кекало А.С. Исследование эффективности алгоритма поиска косяком рыб в задаче глобальной оптимизации // Современные проблемы науки и образования. 2014. №4 URL: <https://www.science-education.ru/ru/article/view?id=14142>.
4. Галиев А.И., Чермошенцев С.Ф. Канальная трассировка сверхбольших интегральных схем на основе алгоритма косяка рыб // Современные материалы, техника и технологии. 2015. №3 (3) . С. 64-68.
5. Васильев В.С.1 Генетические алгоритмы в сравнении с оптимизационными методиками // Студенческая наука для развития информационного общества, сборник материалов 6 всероссийской научно-технической конференции. Ставрополь: Северо-Кавказский федеральный университет, 2017. С. 330-331.
6. Задача на поиск максимального объема коробки. URL: [http://files.school-collection.edu.ru/dlrstore/f65ed928-f396-4319-9dbd-651cba791e30/Modules\\_11-1.6-3/M11\\_1.6-3.html](http://files.school-collection.edu.ru/dlrstore/f65ed928-f396-4319-9dbd-651cba791e30/Modules_11-1.6-3/M11_1.6-3.html) (дата обращения: 15.01.2018).
7. Егин М.М. Алгоритм косяка рыб в многомерной оптимизации // Прикладные исследования и технологии ART 2016. Москва: Негосударственное образовательное учреждение высшего образования

- Московский технологический институт, 2016. С. 96-99
8. Дружинина М.А. Особенности работы алгоритма движения косяка рыб // Современное состояние и перспективы развития технических наук. Уфа: Башкирский государственный университет, 2014. С. 23-25.
  9. Hu X.T., Zhang H.Q., Li Z.C., Huang Y.A., Yin Z.P. A Novel Self-Adaptation Hybrid Artificial Fish-Swarm Algorithm // IFAC Proceedings Volumes. 2013. №5. С. 583-588.
  10. Jiang H. M., Xie K., Wang Y. F. Optimization of pump parameters for gain flattened Raman fiber amplifiers based on artificial fish school algorithm // Optics Communications. 2011. Т. 284. №. 23. С. 5480-5483.