

Нахождение максимума функции с несколькими переменными 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 правовой информатики

Аннотация

В данной статье описывается метод решения задачи на нахождение максимального количества некоторого продукта, используя алгоритм поиска косяком рыб с помощью языка программирования PHP.

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

Finding the maximum of a function with several variables using the search algorithm for a school of fish

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 Legal Informatics

Abstract

This article describes the method of solving the problem of finding the maximum quantity of a certain product using the search algorithm for a fish by using the PHP programming language.

Keywords: PHP, algorithm search for a school of fish, maximum of function.

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. В настоящее время все популярнее становятся методы, основанные на популяционных алгоритмах. Популяционный алгоритм представляет собой эвристический алгоритм поиска, используемый для

решения задач оптимизации и моделирования путём коллективного движения и нахождения лучшего значения функции. Существует множество популяционных алгоритмов: муравьиный алгоритм, метод роя частиц, алгоритм поиска косяком рыб и так далее. Остановимся более подробно на последнем популяционном алгоритме, который показал себя наиболее эффективно в сравнении с другими эвристическими методами.

Цель данного исследования заключается в рассмотрении способа решения задачи, в которой необходимо найти максимум функции, используя алгоритм поиска косяком рыб и язык программирования PHP, и выявить эффективность данного алгоритма. Описание задачи: «На Предприятии производят продукт высокого качества, который зависит от количества работающих станков и количества рабочих. Необходимо найти оптимальное соотношение между числом рабочих предприятия и числом работающих станков для заготовки данного продукта в максимальном количестве за 2 недели, если математически количество произведенного продукта выражено следующей функцией (Рис. 1)

$$z = y\sqrt{x} - 2y^2 - x + 14y$$

Рис 1 – Функция задачи

где z – количество произведенного продукта в тысячах ед., y – количество рабочих в сотнях, x – количество работающих станков в сотнях».

В статье А.Н. Куликова подробно рассказывается об алгоритме поиска косяком рыб и показывается эффективность данного метода. Так же им была разработана «Программа оптимизации методом косяком рыб», которая предназначена обеспечить более быстрый и эффективный поиск локальных экстремумы для функции многих переменных, чем стандартные алгоритмы перебора, благодаря популяционному алгоритму, действие которого основано на поведении косяка рыб в реальных условиях поиска пищи [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. В ходе их исследований были проанализированы основные параметры алгоритма поиска косяком рыб для практической реализации алгоритма в качестве решения проблем относительной локализации. [2]. Также они полностью рассмотрели алгоритм поиска косяком рыб и выявили зависимость результатов при нахождении экстремумов функции от вводимых значений. Ими же было выяснено, что алгоритм поиска косяком рыб является универсальным методом для поиска экстремума независимо от сложности функции [3]. А.И. Галиев, С.Ф. Чермошнецев рассмотрели способ представления решения задачи канальной трассировки, а также эвристики, имитирующие биологические процессы, происходящие при поиске косяком рыб места скопления пищи [4]. М.М. Егин рассматривает принцип работы алгоритма

косяка рыб в многомерной глобальной оптимизации. Он же установил, что при решении задачи глобальной оптимизации для функции со сложным ландшафтом рекомендуется использовать FSS, так как классические алгоритмы зачастую приводят к неверному решению, а генетический алгоритм работает медленнее [5]. 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 [6]. 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 [7]. Х.Т.Ну и др. классифицировали рыбный косяк по нескольким направлениям, и, в связи с этим был предложен новый алгоритм поиска косяком рыб, который объединил достоинства стратегии самоадаптации, стратегии мутации и гибридной стратегии в социальном поведении косяка рыб. Экспериментальные результаты подтвердили превосходство нового алгоритма с точки зрения, как качества решения, так и скорости конвергенции [8].

В данном исследовании мы поставили задачу, в которой необходимо найти оптимальное соотношение между числом рабочих предприятия и числом работающих станков для заготовки данного продукта в максимальном количестве за 2 недели, на языке программирования РНР.

Для начала рассмотрим решение задачи вручную (Рис. 2), без применения алгоритма поиска косяка рыб, с помощью онлайн-сервиса [9].

$$z = y \cdot (x^{1/2}) - (2 \cdot y^2) - x + 14 \cdot y$$

1. Найдем частные производные.

$$\frac{\partial z}{\partial x} = -1 + \frac{y}{2\sqrt{x}}$$

$$\frac{\partial z}{\partial y} = \sqrt{x} - 4 \cdot y + 14$$

2. Решим систему уравнений.

$$-1 + \frac{y}{2\sqrt{x}} = 0$$

$$\sqrt{x} - 4 \cdot y + 14 = 0$$

Получим:

а) Из первого уравнения выражаем x и подставляем во второе уравнение:

$$x = 4 \cdot (2 \cdot y - 7)^2$$

$$\frac{y}{4\sqrt{(2 \cdot y - 7)^2}} - 1 = 0$$

$$\text{Откуда } y_1 = 28/9; y_2 = 4$$

Данные значения y подставляем в выражение для x . Получаем: $x_1 = 196/81; x_2 = 4$

Количество критических точек равно 1.

$$M_1(4;4)$$

Рис 2а – Решение задачи вручную

3. Найдем частные производные второго порядка.

$$\frac{\partial^2 z}{\partial x \partial y} = \frac{1}{2\sqrt{x}}$$

$$\frac{\partial^2 z}{\partial x^2} = -\frac{y}{4 \cdot x^{3/2}}$$

$$\frac{\partial^2 z}{\partial y^2} = -4$$

4. Вычислим значение этих частных производных второго порядка в критических точках

$M(x_0; y_0)$.

Вычисляем значения для точки $M_1(4;4)$

$$A = \frac{\partial^2 z}{\partial x^2(4;4)} = -\frac{1}{8}$$

$$C = \frac{\partial^2 z}{\partial y^2(4;4)} = -4$$

$$B = \frac{\partial^2 z}{\partial x \partial y(4;4)} = \frac{1}{4}$$

$AC - B^2 = 7/16 > 0$ и $A < 0$, то в точке $M_1(4;4)$ имеется максимум $z(4;4) = 28$

Вывод: В точке $M_1(4;4)$ имеется максимум $z(4;4) = 28$;

Рис 2б – Решение задачи вручную

Теперь реализуем алгоритм поиска косяком рыб на языке программирования РНР. Для начала создадим поля под ввод значений, необходимых для дальнейших вычислений, в html-форме (Рис. 3).

Введите размер популяции

Введите количество итераций

Введите нижнюю границу поиска
 X1 Z1

Введите верхнюю границу поиска
 X2 Z2

Введите начальный радиус поиска

Введите конечный радиус поиска

Введите максимальный вес агента

Рассчитать

Рис 3 – Значения для работы алгоритма

Объявляем необходимые массивы, которые будем использовать в дальнейшем (Рис. 4).

```

$array=array_fill(0,$raz_pop,$max_vec/2);//Массив веса рыб
$array1=array_fill(0,$raz_pop, 0);//Массив вычисленных значений функции при смещении
$array2=array_fill(0,$raz_pop, 0);//Массив вычисленных значений функции до смещения
$array3=array_fill(0,$raz_pop, 0);//Разница двух предыдущих массивов
$pos2=array_fill(0,$raz_pop, 0);//Значения 1-й переменной со всеми сдвигами
$pos3=array_fill(0,$raz_pop, 0);//Значения 2-й переменной со всеми сдвигами
$ran1=array_fill(0,$raz_pop, 0);//Значения 1-й переменной без сдвигов
$ran2=array_fill(0,$raz_pop, 0);//Значения 2-й переменной без сдвигов
$kk=$nach-$kones;//коэффициент сдвига

```

Рис 4 – Объявленные массивы

Выбираем случайное положение для каждого агента и заносим его в массив (Рис. 5).

```

for($n=0;$n<=$raz_pop-1;$n++){
    $pos2[$n]=rand($verx,$verx1);
    $pos3[$n]=rand($niz,$niz1);
    $ran1[$n]=$pos2[$n];
    $ran2[$n]=$pos3[$n];
}

```

Рис 5 – Выбор случайного положения агента

Далее начинаем наш цикл, вычисляя значения функции для каждого случайно выбранного положения агента. Здесь же считаем значение функции уже для положения агентов со сдвигом. Сравниваем полученные значения функции и если при новом положении оно оказалось лучше, то закрепляем его, а иначе считаем, что его не было (Рис. 6).

```

for($i=1;$i<=$iter;$i++)
{
    echo ' № итерации '.$i.' <br />';
    for($j=1;$j<=$raz_pop;$j++){
        $array2[$j]=10*(sin(0.1*$ran1[$j])+sin(0.1*$ran2[$j]));
        $pos2[$j]=$pos2[$j]+ rand(-1, 1)*$kk;
        $pos3[$j]=$pos3[$j]+ rand(-1, 1)*$kk;
        $array1[$j]=10*(sin(0.1*$pos2[$j])+sin(0.1*$pos3[$j]));

        if($array1[$j]<$array2[$j]){
            $pos2[$j]=$ran1[$j];
            $pos3[$j]=$ran2[$j];
        }
        else{
            $ran1[$j]=$pos2[$j];
            $ran2[$j]=$pos3[$j];
        }
    }
}

```

Рис 6 – Поиск лучшего значения функции

Закрепляем успех в индивидуальной стадии, для чего вычисляем характеристику «Вес». После этого рыбы совершают инстинктивно-коллективную стадию плавания, в ходе которой высчитывается величина «Общий шаг миграции» (Рис. 7).

```

$array3[$j]=$array2[$j]-$array1[$j];
$m+=$(pos2[$j]-$ran1[$j])*$array3[$j];
$m3+=$(pos3[$j]-$ran2[$j])*$array3[$j];
    $m1+=$array3[$j];
        if($m1==0) $m1=1;
        if($m==0) $m=1;
        if($m3==0) $m3=1;
        if($array3[$j]==0) $array3[$j]=1;
    }

$m2=$m/$m1;//Общий шаг миграции
$m4=$m3/$m1;//Общий шаг миграции
for($l=1;$l<=$raz_pop;$l++){
    if(($array3[$l]>$max) and ($array3[$l]!=0))
    {
        $max=$array3[$l];
    }
}

$t=array_sum($array);

for($k=1;$k<=$raz_pop;$k++){
    $array[$k]+=$array3[$k]/$max;//Вес каждой рыбы
    if($array[$k]>$max_vec) $array[$k]=$max_vec;
    if($array[$k]<0) $array[$k]=1;
}

```

Рис 7 – Вес каждой рыбы и косяка в целом

Вычисляем центр тяжести косяка и смещаем текущее положение каждого агента на вычисленный ранее общий шаг миграции (Рис. 8).

```

for($f=1;$f<=$raz_pop;$f++){
    $pos2[$f]= $pos2[$f] + $m2;
    $pos3[$f]= $pos3[$f] + $m4;
    $bary4+=$pos3[$f]*$array[$f];
    $bary+=$pos2[$f]*$array[$f];
    $bary1+=$array[$f];
}
$bary3=$bary/$bary1;
$bary2=$bary4/$bary1;

```

Рис 8 – Центр тяжести косяка и смещение на общий шаг миграции

Высчитываем окончательные положения агентов после каждой итерации, учитывая центр тяжести всего косяка и коэффициент сдвига. Если при этом вес косяка в целом увеличился, значит рыбы находятся в верном положении и следует сузить круг поиска и наоборот, если он уменьшился, то следует расширить круг (Рис. 9).

```

if(array_sum($array)>$t){
$pos2[$n]=$pos2[$n]+(($kk^2)*rand(0,1)*($pos2[$n]-$bary3)/(sqrt(pow($pos2[$n]-$bary3,2))));
$pos3[$n]=$pos3[$n]+(($kk^2)*rand(0,1)*($pos3[$n]-$bary2)/(sqrt(pow($pos3[$n]-$bary2,2))));
}
else {
    $pos2[$n]=$pos2[$n]-(($kk^2)*rand(0,1)*($pos2[$n]-$bary3)/(sqrt(pow($pos2[$n]-$bary3,2))));
    $pos3[$n]=$pos3[$n]-(($kk^2)*rand(0,1)*($pos3[$n]-$bary2)/(sqrt(pow($pos3[$n]-$bary2,2))));
}
if($pos2[$n]>$verx1) $pos2[$n]=$verx1;
if ($pos2[$n]<$verx) $pos2[$n]=$verx;
if($pos3[$n]>$niz1) $pos3[$n]=$niz1;
if ($pos3[$n]<$niz) $pos3[$n]=$niz;
}
}

```

Рис 9 – Окончательное смещение агентов в конце итерации

Конечным шагом перед следующей итерацией становится подсчет коэффициента сдвига и определение максимума среди вычисленных значений функции (Рис. 10).

```

$kk=($nach-$konec)/($i+1);
if($maxim<max($array1)){
$maxim=max($array1);
}

```

Рис 10 – Коэффициент сдвига и максимум функции

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 (Рис. 11).

Введите размер популяции

Введите количество итераций

Введите нижнюю границу поиска
X1 Z1

Введите верхнюю границу поиска
X2 Z2

Введите начальный радиус поиска

Введите конечный радиус поиска

Введите максимальный вес агента

Рис 11 – Необходимые параметры для работы алгоритма

Максимальное значение функции, вычисленное на 1 итерации (Рис. 12).

№ итерации 1
 1агент: Значения переменных со сдвигом. 1 переменная 0 2 переменная 0
 Значение функции без сдвига 0
 Значение функции со сдвигом 0
 2агент: Значения переменных со сдвигом. 1 переменная 9 2 переменная -2
 Значение функции без сдвига -51
 Значение функции со сдвигом -51
 3агент: Значения переменных со сдвигом. 1 переменная 3 2 переменная -17
 Значение функции без сдвига -256.85640646055
 Значение функции со сдвигом -848.44486372867
 4агент: Значения переменных со сдвигом. 1 переменная 0 2 переменная 10
 Значение функции без сдвига 6
 Значение функции со сдвигом -60
 5агент: Значения переменных со сдвигом. 1 переменная 0 2 переменная 0
 Значение функции без сдвига 0
 Значение функции со сдвигом 0
 6агент: Значения переменных со сдвигом. 1 переменная 0 2 переменная 1
 Значение функции без сдвига 7.6457513110646
 Значение функции со сдвигом 12
 7агент: Значения переменных со сдвигом. 1 переменная 9 2 переменная 3
 Значение функции без сдвига 24
 Значение функции со сдвигом 24
 8агент: Значения переменных со сдвигом. 1 переменная 0 2 переменная -9
 Значение функции без сдвига -288
 Значение функции со сдвигом -288
 9агент: Значения переменных со сдвигом. 1 переменная 0 2 переменная -19
 Значение функции без сдвига -340
 Значение функции со сдвигом -988
 10агент: Значения переменных со сдвигом. 1 переменная 0 2 переменная -10
 Значение функции без сдвига 0
 Значение функции со сдвигом -340
 Сравнение веса косяка рыб. Вес который стал 252.54048488978 Вес который был 250
 коэффициент сдвига 4.5
 Максимальное значение этой и предыдущих итераций 24

Рис 12 – Значения функции на 1 итерации

Не совсем точное значение максимума функции, поэтому переходим на последнюю итерацию (Рис. 13).

№ итерации 30
 1агент: Значения переменных со сдвигом. 1 переменная 3.5308721158847 2 переменная -3.7073146973022
 Значение функции без сдвига -80.859617159558
 Значение функции со сдвигом -89.887914719033
 2агент: Значения переменных со сдвигом. 1 переменная 3.5202678974431 2 переменная 4.4520047295205
 Значение функции без сдвига 27.958660687492
 Значение функции со сдвигом 27.520125195381
 3агент: Значения переменных со сдвигом. 1 переменная 1.0713354224563 2 переменная 3.0762521895673
 Значение функции без сдвига 26.308086958378
 Значение функции со сдвигом 26.25362527598
 4агент: Значения переменных со сдвигом. 1 переменная 4.4554726321099 2 переменная 3.56225550441
 Значение функции без сдвига 27.938724486175
 Значение функции со сдвигом 27.555981409016
 5агент: Значения переменных со сдвигом. 1 переменная 3.3566866399831 2 переменная 3.5101260109573
 Значение функции без сдвига 27.970114396581
 Значение функции со сдвигом 27.574102335495
 6агент: Значения переменных со сдвигом. 1 переменная 1.7057896118533 2 переменная 2.6872137875054
 Значение функции без сдвига 27.33757057502
 Значение функции со сдвигом 24.982626771808
 7агент: Значения переменных со сдвигом. 1 переменная 7.8146194582255 2 переменная 4.5869977439309
 Значение функции без сдвига 27.430886931022
 Значение функции со сдвигом 27.145039808606
 8агент: Значения переменных со сдвигом. 1 переменная 0.92084176512537 2 переменная -0.88212819995191
 Значение функции без сдвига -31.742980901023
 Значение функции со сдвигом -15.673431522251
 9агент: Значения переменных со сдвигом. 1 переменная 3.7237074221661 2 переменная 3.7158647218356
 Значение функции без сдвига 27.99956404414
 Значение функции со сдвигом 27.853568196717
 10агент: Значения переменных со сдвигом. 1 переменная 1.2523773463703 2 переменная 3.0058942270778
 Значение функции без сдвига 26.122704754582
 Значение функции со сдвигом 26.123227829841
 Сравнение веса косяка рыб. Вес который стал 256.65405721515 Вес который был 256.65707668459
 коэффициент сдвига 0.29032258064516
 Максимальное значение этой и предыдущих итераций 27.99956404414

Рис 13 – Значения функции на последней итерации

Как видно из рисунка 13 максимум функции найден и примерно равен 27,99956, в то время как при аналитическом решении задачи было получено значение 28. Таким образом видно, что значения практически совпадают, а недостаток алгоритма — это приближённость данных. Но к плюсам можно отнести его быстроедействие. Иногда этот фактор является основным в работе, и даже минимальная погрешность не столь важна.

На деле же не всегда получается сразу найти необходимый результат, который напрямую зависит от вводимых пользователем значений. Приходится искать оптимальные параметры для работы алгоритма, что является его несомненным минусом. Для наглядности протестируем наш алгоритм при разных вводимых параметрах (Таблица 1).

Таблица 1 – Вычисление значений функции при разных вводимых параметрах

Название параметра	Номер эксперимента					
	1	2	3	4	5	6
Размер популяции	10	20	25	40	50	30
Количество итераций	10	20	25	10	50	30

Нижняя граница поиска	-20, -20	-15, 15	-50, -50	-45, -45	-10, -10	-20, -20
Верхняя граница поиска	20, 20	15, 15	50, 50	100, 100	40, 40	20, 20
Начальный радиус поиска	20	15	50	100	20	20
Конечный радиус поиска	5	3	1	10	2	1
Максимальный вес агента	20	40	100	200	45	50
Значение функции при последней итерации	24.4296	27.4595	27.9275	25.7834	27.9976	27.9989

Проведя все эксперименты можно сделать вывод, что для получения более точного максимального значения следует вводить такие параметры, где размер популяции и количество итерации должны быть относительно большими, нижняя и верхняя границы поиска находиться между собой не в сильно большом промежутке, начальный и конечный радиусы поиска отличаться не на сильно маленькое число, а максимальный вес агента должен лежать в промежутке между числами 10 и 100.

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. 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. 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
 7. Дружинина М.А. Особенности работы алгоритма движения косяка рыб // Современное состояние и перспективы развития технических наук. Уфа: Башкирский государственный университет, 2014. С. 23-25.
 8. 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.
 9. Поиск экстремумов функции URL: <https://math.semestr.ru/math/extremum.php> (дата обращения: 14.05.18).
 10. Маринчук А.С. Нахождение максимума функции с помощью алгоритма поиска косяком рыб // Постулат. 2018. №1.