

УДК 004.932.721

Разработка алгоритма распознавания сложных накладывающихся геометрических объектов

Минков О.В., Ручкин К.А.

Донецкий национальный технический университет
oleg_spy93@mail.ru

Минков О.В., Ручкин К.А. Разработка алгоритма распознавания сложных накладывающихся геометрических объектов. Рассмотрен алгоритм бактериальной оптимизации для нахождения и распознавания геометрических фигур, в частности окружностей, а также различных его модификация. Проведен анализ классического алгоритма с его улучшенными версиями. Предложена своя модификация для одновременного нахождения окружностей.

Ключевые слова: распознавание, компьютерное зрение, эволюция, BFOA, геометрические фигуры.

Введение

В современном мире очень интенсивно развиваются различные системы компьютерного зрения, что в свою очередь позволяет эффективно взаимодействовать человеку и технике. Одной из самых важных и нужных целей которые ставит перед собой системы компьютерного зрения, является задача автоматического распознавания образов, решение которой позволит корректно оценивать окружающую среду, а затем выполнять в ней необходимые действия. Но данное направление является достаточно обширным и может применяться абсолютно в разных сферах жизнедеятельности. Например, это может пригодиться в строительной сфере, в сфере медицины, образования, машиностроения вплоть до военного дела. Соответственно меняется и сам объект распознавания, будь то человек, машина, животное, растение, строение и так далее.

1 Постановка задачи

В рамках данной работы мы будем рассматривать алгоритм бактериальной оптимизации для нахождения и распознавания геометрических фигур, в частности окружностей, а также различных его модификаций. Проведем небольшой анализ классического алгоритма с его улучшенными версиями. Затем рассмотрим модификацию для одновременного нахождения окружностей и приведем графики работы по скорости и по точности.

2 Анализ предметной области

В настоящее время достаточно большое

количество работ посвящено исследованию применения эволюционных методов для решения задач распознавания изображений. Рассмотрим эти работы более подробно.

2.1 Эволюционные алгоритмы

В статье S. Das, S. Dasgupta, A. Biswas, A. Abraham [1] представлен алгоритм, для автоматического обнаружения круглых форм на сложных и шумных изображениях. Алгоритм основан на гибридной технике, состоявшей из моделируемого отжига и дифференциальной эволюции. Новая нечеткая целевая функция была получена на граничной карте входного изображения. Минимизация этой функции с гибридным алгоритмом обжига дифференциальной эволюции приводит к автоматическому обнаружению кругов на изображении. Моделирование результатов заканчивается по нескольким синтетическим, а также естественным изображениям с изменяемой сложностью, это утверждают эффективность предложенной техники с точки зрения ее предельной точности, скорости и надежности.

В статье E. Cuevas, M. Gonzalez, D. Zaldivar, M. Perez-Cisneros [2] представлен новый и эффективный метод для извлечения эллипсов на изображении. Данный подход использует эволюционный алгоритм для имитации коллективного поведения животных, предполагая, что общий процесс обнаружения является проблемой мультимодальной оптимизации. В этом алгоритме, поисковые агенты эмулируют группу животных, которые взаимодействуют друг с другом, используя простые биологические правила, которые моделируются как эволюционные операторы. В свою очередь, такие операторы применяются к

каждому объекту, который рассматривается как полная группа, имеющая память для хранения оптимального решения (эллипсов), так называемый принцип конкуренции. Детектор использует комбинацию из 5 граничащих точек в качестве параметров для определения кандидатов в эллипсы (возможные решения), в то время как функция соответствия определяет, присутствуют ли эти кандидаты на реальном изображении. Руководствуясь значениями функции соответствия. Набор закодированных кандидатов эллипса эволюционируют через эволюционный алгоритм, так, что лучшие кандидаты могут быть вставлены в реальные эллипсы на изображении. Сразу же, по окончании оптимизационного процесса, выполняется анализ встроенной памяти, для того, чтобы найти лучшее полученное решение (лучший эллипс) и важные локальные минимумы (оставшиеся эллипсы). Экспериментальные по нескольким сложным синтетическим и естественным изображениям подтвердили эффективность предложенной методики в отношении точности, скорости и надежности.

В статье S. Fazli, S. Fathi Ghiri [3] был рассмотрен алгоритм для обнаружения круглых форм реального мира, на сложных и зашумленных изображениях без использования обычного преобразования Хафа. Алгоритм базируется на одной из самых недавних техник роевого интеллекта, которая называется Искусственная Пчелиная Колония (ИПК). Наш алгоритм — это новое применение ИПК. Новая целевая функция была получена на граничной карте желаемого изображения. Алгоритм искусственных пчелиных колоний минимизирует целевую функцию и это приводит к автоматическому обнаружению кругов на цифровом изображении.

2.2 Алгоритм BFO и различные его модификации

Одной из особенностей алгоритма BFO является нахождение накладывающихся объектов. Накладывающимися объектами выступают как пересекающиеся объекты, так и объекты, которые располагаются почти друг под другом.

В статье Y. Liu, K. M. Passino [4] мы объясняется социальное поведение бактерий E.Coli и M.Xanthus занимающихся поиском пищи и разработаем имитационную модель, основанную на принципах теории о поиске пищи, и рассмотрим поиск пищи как оптимизацию. Это дает нам новые модели

поведения и новые методы распределенной не градиентной оптимизации. Кроме того, мы покажем, что модели обоих видов бактерий демонстрируют свойство, открытое Грюнбаумом, которое говорит о том, что их поведение при поиске пищи является социальным, что дает им возможность передвигаться по шумным градиентам в питательных веществах. Это обеспечивает соединения между эволюционными силами в алгоритме социального поиска пищи и распределенной не градиентной оптимизации для глобальной оптимизации на шумных поверхностях. Далее рассмотрим некоторые модификации алгоритма бактериальной оптимизации.

В статье B. Hernandez-Ocana, Ma. Del Pilar Pozos-Parra, E. Mezura-Montes [5] рассматривается улучшенная версия модифицированного алгоритма бактериальной оптимизации для решения численного ограничения оптимизационных проблем. Добавлено 4 механизма:

- два оператора плавления, один отвечает за исследование, а другой сфокусирован на исследовании поискового пространства, где рассматривается динамический механизм для обработки значения длины шага;
- искаженный механизм для более подходящего начального роя, где бактерии разделяются на 3 группы, две из которых размещаются вблизи краев поискового пространства, а последняя разбросана по всей поверхности пространства поиска;
- локальный поисковый оператор;
- уменьшение шага воспроизведения для решения проблемы преждевременной конвергенции.

60 хорошо известных тестовых проблем из двух критериев решаются в трех экспериментах. Первый эксперимент направлен на предоставление предварительных данных о подходящем поведении нового механизма. Второй эксперимент дает углубленное сравнение новой версии с ее предыдущей на основе конечных результатов и четырех показателей производительности. Третий эксперимент сравнивает производительность предложенного алгоритма с пятью современными алгоритмами, вдохновленными природой, предназначенными для работы с ограниченным пространством непрерывного поиска. Результаты показывают, что предложенный алгоритм обеспечивает лучшую производительность по сравнению с его предшественником, улучшая его способность достигать допустимого региона и генерируя

лучшие решения, в то же время получая конкурентоспособную производительность по сравнению с самыми современными алгоритмами.

В статье R. Panda, M. K. Naik [6] представлен модифицированный алгоритм бактериальной оптимизации, который называется кроссоверный алгоритм бактериальной оптимизации, который наследует кроссоверные техники от генетических алгоритмов. Это может быть использовано для импровизации значения оптимальной целевой функции. Идея использования кроссовермеханизма состоит в том, чтобы искать близлежащие места потомства (50 процентов бактерий), потому что они производятся в разных случайных местах. В традиционном алгоритме бактериальной оптимизации поиск начинается с некоторой позиции (50 процентов бактерий дублируются) что не желательно. Для оценки эффективности учитываются семь различных эталонных функций. Также приведено сравнение с результатами предыдущих методов, чтобы выявить эффективность предложенного алгоритма.

В статье Chin-Ling Lee, Cheng-Jian Lin [7] представлен алгоритм бактериальной оптимизации на основе адаптационной стратегии для решения сложных оптимизационных задач (SABFO). Предлагаемый алгоритм SABFO использует стратегический подход на этапе хемотаксиса традиционного алгоритма бактериальной оптимизации (BFO). Предлагаемый метод позволяет каждой бактерии плавать на разные расстояния, а также увеличивает бактериальное разнообразие. Для проверки производительности SABFO используются пять задач оптимизации нелинейных контрольных функций. Результаты моделирования показывают, что SABFO получает лучшие глобальные оптимальные решения, чем другие методы.

3 Алгоритм бактериальной оптимизации

Алгоритм бактериальной оптимизации (BFOA) предложенный Кевином Пассино (Kevin Passino, 2002), является новым пришествием в семействе оптимизационных алгоритмов, вдохновленных природой. Ключевой идеей этого нового алгоритма является применение групповой стратегии кормления роя бактерий *E.coli* при оптимизации функций. Бактерии ищут питательные вещества,

чтобы максимизировать энергию, получаемую за единицу времени. Отдельная бактерия также общается с другими, отправляя сигналы. Бактерия принимает решения о кормлении после рассмотрения двух предыдущих факторов. Процесс, в котором бактерия движется, делая небольшие шаги при поиске питательных веществ, называется хемотаксисом. Ключевая идея BFOA - имитировать хемотаксическое движение виртуальных бактерий в пространстве поиска проблем.

P: размер поискового пространства,

S: общее количество бактерий в популяции,

Nc: количество шагов хемотаксиса,

Ns: Длина плавания.

Nre: количество шагов воспроизведения,

Ned: количество событий выбывания-рассеивания,

Ped: вероятность выбывания-рассеивания,

C(i): размер шага, выполненного в случайном направлении, указанном кувирканием.

Теория поиска пищи, основывается на предположении, что животные ищут и получают питательные вещества, для того чтобы увеличить потребление энергии E, за единицу времени T, затраченного на поиск пищи. Следовательно, они пытаются максимизировать функцию типа E/T (или максимизировать свой долгосрочный средний уровень потребления энергии). Максимизация такой функции обеспечивает источники питательных веществ для выживания и дополнительного времени для других важных видов деятельности (например, борьба, бегство, спаривание, размножение, сон или строительство жилья). Такие виды деятельности, как строительство убежища и поиск партнера иногда очень схоже с тем, как это делает медведь. Ясно, что поиск пищи очень различается у разных видов. Травоядные животные, как правило, легко находят пищу, но должны есть много. Хищникам же обычно трудно найти пищу, но им не нужно столько есть, поскольку их пища имеет высокую энергетическую ценность. «Окружающая среда» устанавливает структуру питательных веществ, которые можно получить от других организмов которые пригодны для пищи, от геологических ограничений, таких как, реки, горы и погодные условия, и это создает ограничения для получения этой пищи (например, маленькие порции еды могут находиться на большом расстоянии друг от друга). Поиска пищи может

быть рискован из-за хищников, жертва может быть подвижной, поэтому ее нужно преследовать, а физиологические характеристики ограничивают его возможности и конечный успех. Теория бактериальной оптимизации объясняется следующими шагами:

- хемотаксис;
- роение;
- воспроизведения;
- ликвидация – рассеивания.

3.1 Хемотаксис

Этот процесс симулирует движение бактерий *E.Coli* через плавание и кувыркания с помощью жгутиков. Биологически бактерия *E.Coli* может двигаться двумя разными способами. Это может быть плавание за некоторый период времени в одинаковом направлении или же кувыркание, и на всем периоде жизни это два режима чередуются. Пусть $O^i(j, k, l)$ будет i -ая бактерия на j -ом шаге хемотаксиса, на k -ом шаге воспроизведения и на l -ом шаге выбывания-рассеивания $C(i)$ – размер шага, взятом в случайном направлении, который был определен кувырканием (длина одного рывка). Тогда в вычисляемом хемотаксисе движение бактерии может быть представлено следующим выражением:

$$X^i(j+1, k, l) = X^i(j, k, l) + C(i) \frac{\Delta}{\sqrt{\Delta^T \Delta}}, \quad (3.1)$$

где Δ обозначает вектор в случайном направлении, элементы которого лежат в $[-1, 1]$.

3.2 Роение

Интересное групповое поведение наблюдалось у некоторых подвижных разновидностей бактерий включая *E.Coli* и *S. Typhimurium*, где замысловатые и стабильные пространственно-временные паттерны (рои) формируются в полутвердых питательных средах. Группа бактерий *E.coli* располагается в перемещающемся кольце и перемещается по направлению увеличения питательных веществ, когда помещается среди полутвердой матрицы с одним питательным химиоэффектом. Бактерии, которые стимулированы высоким уровнем сукцината, выделяют аспартат аттрактанта, который помогает им собираться в группы и таким образом перемещаться как концентрированные образцы роев с высокой бактериальной плотностью.

3.3 Воспроизведение

Бактерии, у которых меньше всего здоровья, в конечном итоге умирают, когда каждая из здоровых бактерий (у которой значение целевой функции минимально) беспорядочно расщепляется на две бактерии, которые затем помещаются в одно и то же место. Это делает количество бактерий неизменным.

3.4 Ликвидация – рассеивания

Постепенные или внезапные изменения в локальной среде, где живут бактерий, могут происходить по различным причинам. События могут происходить так, что все бактерии в регионе погибают или группа, рассеивается в новую часть окружающей среды. Например, значительное повышение локальной температуры может убить группу бактерий, которые в данный момент находятся в регионе с высокой концентрацией питательных веществ. События могут происходить таким образом, что все бактерии в регионе погибают или группа, рассеивается на новое место. В течение длительных периодов времени такие события распространяли различные типы бактерий в каждую часть нашей окружающей среды от нашего кишечника к горячим источникам и подземным средам. Для моделирования этих феноменов в BFOA некоторые случайные бактерии выбывают с очень маленькой вероятностью, в то время как новые бактерии в случайном порядке инициализируются по всей поверхности поискового пространства.

События ликвидация и рассеивания имеют эффект возможного разрушения хемотаксического прогресса, но они также оказывают положительное влияние на хемотаксис, так как рассеивание может расположить бактерии вблизи источников хорошего питания. С широкой точки зрения выбывания и рассеивание являются составными частями подвижного поведения на уровне популяции на дальнем расстоянии.

3.5 Алгоритм бактериальной оптимизации для предложенного метода

Шаг 1: Инициализация параметров S , N_s , N_{re} , N_{ed} , P_{ed} и $C(i)$, ($i = 1, 2, \dots, S$).

Выбираем начальные значения для θ^i , $i = 1, 2, \dots, S$. Это должно быть сделано в областях, где существует оптимальное значение. Контрольные переменные (θ^i) представляют собой реактивную мощность (для SVC) и линейное реактивное сопротивление (для

TCSC). Они случайным образом распределены по все области оптимизационного пространства. После того как вычисления θ завершены, значение P (позиция каждого члена в популяции S -бактерий) обновляется автоматически, и завершающий тест делается для максимального количества указанных итераций.

Шаг 2: Цикл выбывания-рассеивания: $l = l + 1$.

Шаг 3: Цикл воспроизведения: $k = k + 1$.

Шаг 4: Цикл хемотаксиса: $j = j + 1$.

Шаг 4.1: Выполним хемотаксический шаг для i -ой бактерии $i = 1, 2, \dots, S$, следующим образом:

Шаг 4.2: Рассчитаем стоимость $J(i, j, k, l)$.

Шаг 4.3: Пусть $J(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta^i(j, k, l), P(j, k, l))$.

Шаг 4.4: Пусть $J_{\text{последнее}} = J(i, j, k, l)$ сохраняем это значение, пока не найдется лучшее.

Шаг 4.5: Кувыркание: Генерируем случайный вектор $\Delta(i) \in R^p$, где каждый элемент $\Delta_m(i)$, $m = 1, 2, \dots, p$, это число лежащее в диапазоне $[-1, 1]$. где R – действительное число. Новое положение рассчитывается по следующей формуле:

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (3.2)$$

где $C(i) > 0$ – размер шага в определённом направлении, позволяющий моделировать процесс кувыркания.

Шаг 4.6: Рассчитываем $J(i, j+1, k, l)$.

Рассчитывается общая стоимость. Если функция стоимости минимальна, то выполняем следующий шаг, иначе переходим к этапу (4.3).

Шаг 4.7: Плавание.

Шаг 4.7.1: Пусть $m = 0$ (количество шагов плавания).

Шаг 4.7.2: До тех пор пока $m < N_s$

Пусть $m = m + 1$. Если $J(i, j+1, k, l) < J_{\text{последнее}}$ (если оно улучшается),

пусть $J_{\text{последнее}} = J(i, j+1, k, l)$, и пусть

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}, \quad (3.3)$$

и используем это $\theta^i(j+1, k, l)$, чтобы рассчитать новый $J(i, j+1, k, l)$. Иначе, пусть $m = N_s$. Конец цикла.

Шаг 4.8: Если $i \neq S$, переходим к следующей бактерии ($i + 1$).

Шаг 5: Если $j < N_c$ перейти к шагу 3. В этом случае продолжаем хемотаксис, бактерии еще живы.

Шаг 6: Воспроизведение. В

Шаг 6.1: Для заданных k и l , и для каждой $i = 1, 2, \dots, S$, пусть $J^i_{\text{здоровье}} =$

$\sum_{j=1}^{N_c+1} J(i, j, k, l)$ будет здоровье i -ой бактерии. Сортируем бактерии и параметр $C(i)$ в порядке возрастания стоимости $J_{\text{здоровье}}$.

Шаг 6.2: S_r – ая бактерия с наивысшим $J_{\text{здоровье}}$ значением умирает, а остальные бактерии с лучшими значениями, разделяются.

Шаг 7: Если $k < N_{re}$, переходим к шагу 2. В данном случае мы не достигли указанного количества шагов воспроизведения.

Шаг 8: Выбывание – рассеивание.

Для $i = 1, 2, \dots, S$, с вероятностью P_{ed} , устранить и рассеять каждую бактерию. Исключить бактерию и поместить ее в случайное место на поисковом пространстве оптимизации. Если $l < N_{ed}$, перейдите к шагу 1, иначе конец.

В таблице 3.1 приведены основные параметры алгоритма бактериальной оптимизации.

Таблица 3.1 – Основные параметры алгоритма бактериальной оптимизации

№	Параметр	Значение
1	Количество бактерий, S	50
2	Максимальное количество шагов, N_s	4
3	Количество шагов хемотаксиса, N_c	100
4	Количество шагов воспроизведения, N_{re}	4
5	Количество шагов выбывания – рассеивания, N_{ed}	2
6	Вероятность, P_{ed}	0.25
7	Размер шага, $C(i)$	0.1

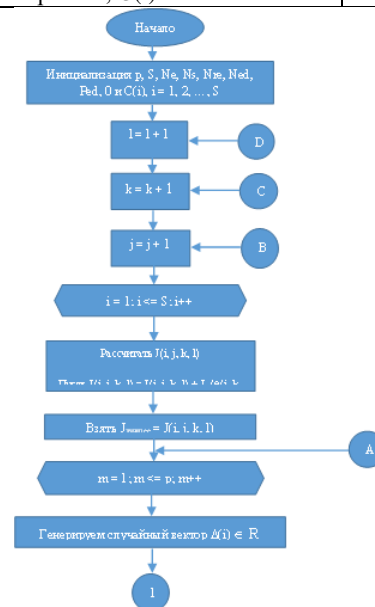


Рисунок 3.1 - Блок-схема работы классического алгоритма

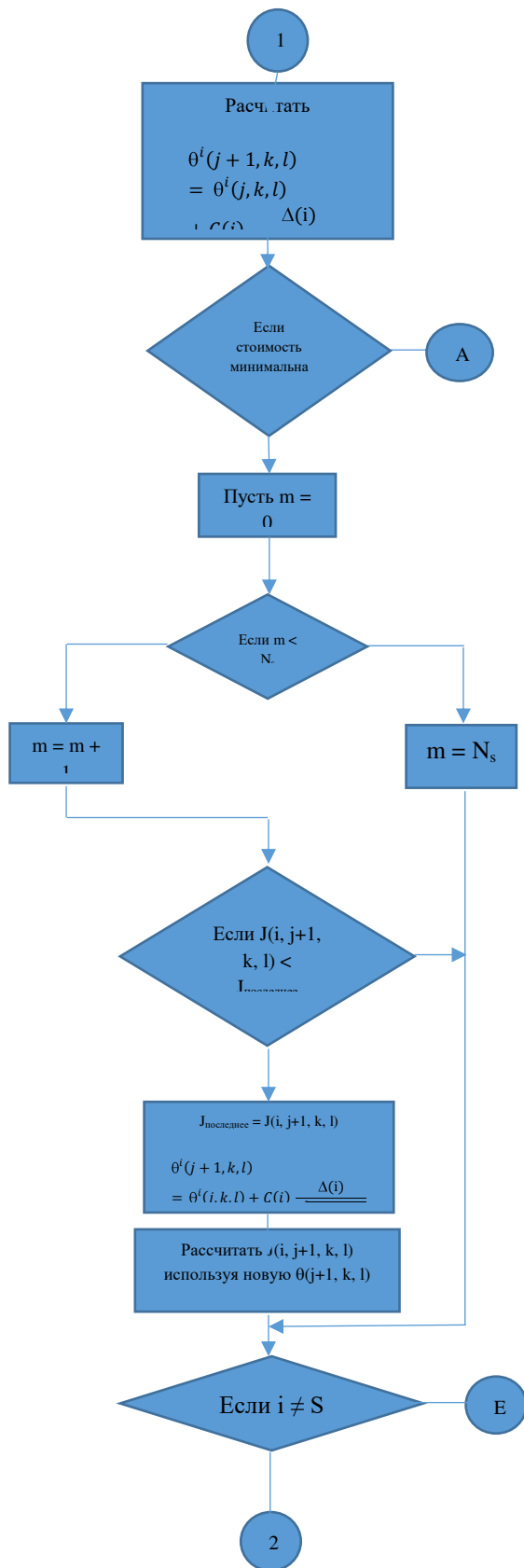


Рисунок 3.2 - Продолжение блок-схемы работы

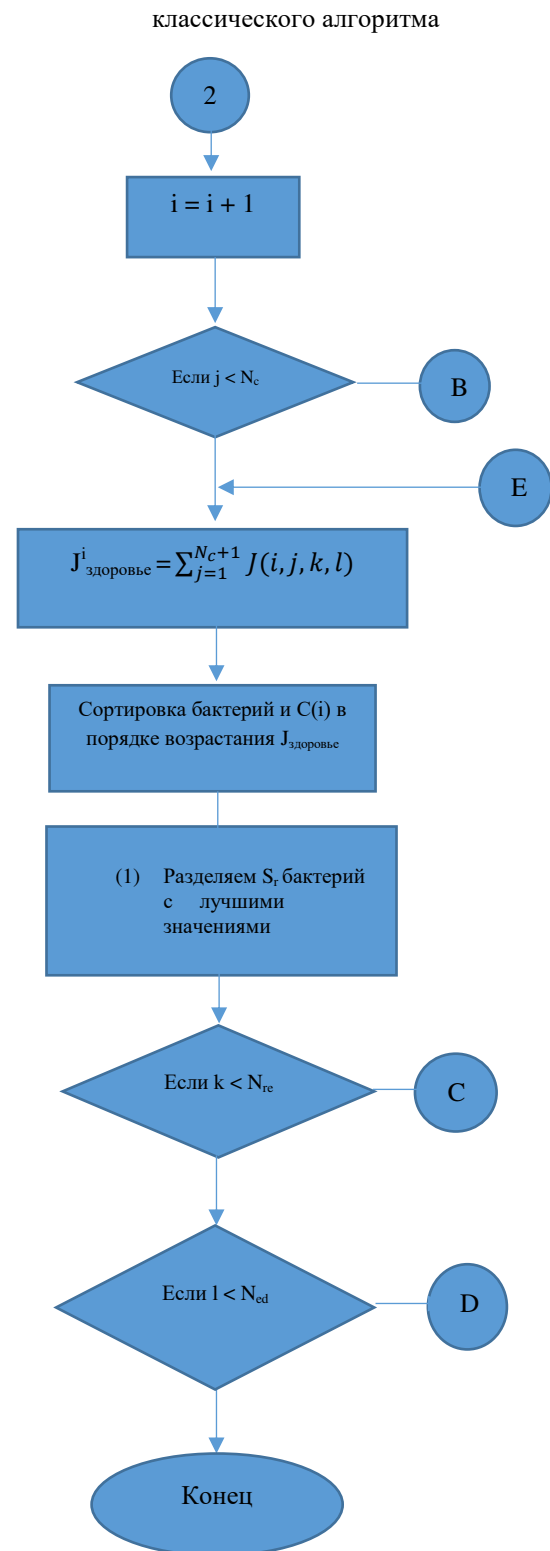


Рисунок 3.3 – Продолжение работы классического алгоритма

3.7 Предлагаемая модификация

На каждом шаге в цикле выбывания-

рассеивания, по окончании самого цикла, мы считаем количество групп методом формального элемента. Каждая группа бактерий представляет собой некоторый объект на изображении, не обязательно окружность. Желаемое количество объектов задается на фазе инициализации. Как только количество групп станет равным заданному, алгоритм прекращает свою работу. После этого высчитываются средние значения для X , Y и радиуса в каждой группе, эти значения и являются окружностями.

3.8 Блок схема предлагаемой модификации

На рисунке 3.4 приведена часть блок схемы самой модификации.

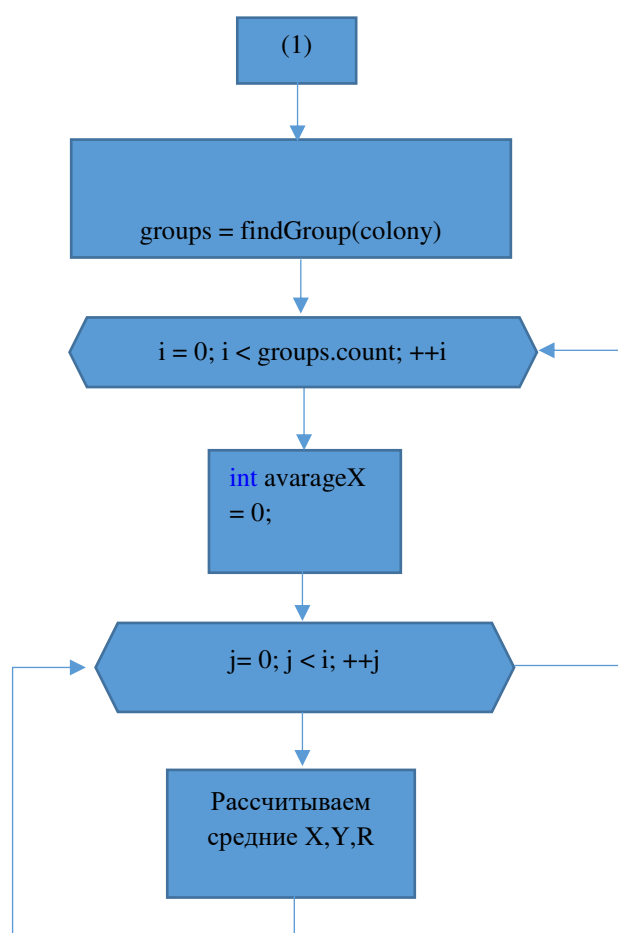


Рисунок 3.4 – Блок-схема предлагаемой модификации

4 Метод формального элемента

Алгоритм Форель (ФОРмальный Элемент) является примером эвристического дивизимного алгоритма классификации. В

основе работы алгоритма Форель лежит использование гипотезы компактности: близким в содержательном смысле объектам в геометрическом пространстве признаков соответствуют обособленные множества точек, так называемые «сгустки». Если расстояние между центром n -го таксона и точкой k этого таксона обозначить S_{nk} , то сумма расстояний между центром и всеми точками k этого таксона будет равна:

$$P_n \sum_{k=1}^L S_{nk} \quad (4.1)$$

где, P_n – расстояние между центром n -го таксона и всеми точками этого таксона; S_{nk} – расстояние между центром n -го таксона и точкой k этого таксона. Сумма таких внутренних расстояний для всех n таксонов равна:

$$P = \sum_{n=1}^N P_n \quad (4.2)$$

Целью работы алгоритма Форель является найти такое разбиение множества объектов на n таксонов, чтобы величина P была минимальной. Работа алгоритма заключается в перемещении гиперсферы определенного радиуса в геометрическом пространстве до получения устойчивого центра тяжести наблюдений, попавших в эту гиперсферу. До начала работы алгоритма признаки объектов нормируются так, чтобы их значения находились между нулем и единицей.

5 Тесты

Проведем некоторые эксперименты. Будем запускать алгоритм по 10 раз на разных изображениях и высчитаем точность нахождения окружностей, а также среднюю скорость работы. Полученные данные представлены на рисунках 5.2, 5.4, а исходные данные на рисунках 5.1, 5.3.

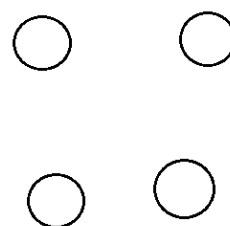


Рисунок 5.1 – Исходное изображение с четырьмя кругами

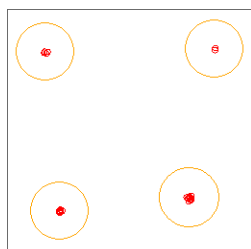


Рисунок 5.2 – Полученные результаты на изображении с четырьмя кругами

Точность нахождения: 55%. Среднее время 4.45

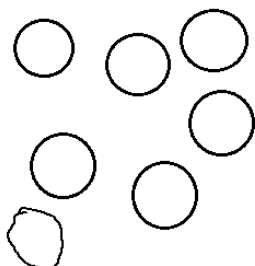
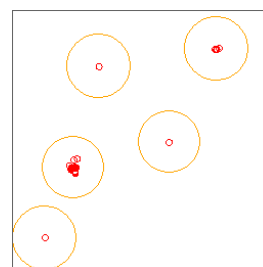


Рисунок 5.3 – Исходное изображение с семью кругами



Точность нахождения 65.7%. Среднее время 5.5

Выводы

Мы рассмотрели алгоритм бактериальной оптимизации для нахождения и распознавания окружностей, а также различных его модификаций. Провели анализ классического алгоритма с его улучшенными версиями. Затем предложили модификацию для одновременного нахождения окружностей и привели таблицу работы по скорости и по точности. Как видно из таблицы, наша модификация еще не ищет все сто процентов кругов, но делает это значительно быстрее классического алгоритма. Увеличение точности планируется за счет разброса бактерий, не на лучшие позиции, а по всей поверхности исключая уже найденные окружности, для того чтобы бактерии смогли найти другие локальные минимумы.

Рисунок 5.4 – Полученные результаты на изображении с семью кругами

Минков О.В., Ручкин К.А. Разработка алгоритма распознавания сложных накладываются геометрических объектов. Рассмотрен алгоритм бактериальной оптимизации для нахождения и распознавания геометрических фигур, в частности окружностей, а также различных его модификация. Проведен анализ классического алгоритма с его улучшенными версиями. Предложена своя модификация для одновременного нахождения окружностей.

Ключевые слова: распознавание, компьютерное зрение, эволюция, BFOA, геометрические фигуры.

Minkov O.V, Ruchkin K.A. Development of an algorithm for recognizing complex overlapping geometric objects. The algorithm of bacterial optimization for finding and recognizing geometric figures, in particular circles, as well as various modifications thereof, is considered. The analysis of the classical algorithm with its improved versions is carried out. We propose our own modification for the simultaneous determination of circles.

Key words: recognition, computer vision, evolution, BFOA, geometric figures.

Статья поступила в редакцию 25.5.2017

Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко