Содержание

Введение

На сегодняшний день задача поисковой оптимизации является важнейшей задачей во многих областях. Например, картографические сервисы используют алгоритмы нахождения кратчайшего пути на графе для нахождения путей между физическими объектами, сервисы перевозки используют алгоритмы нахождения минимальной стоимости пути для сокращения расходов на дорожные затраты. Все эти сервисы стремятся выполнить свою задачу и сохранить как можно больше временных, денежных и человеческих ресурсов. Так возникают задачи оптимизации поиска кратчайшего пути.

Для решения задач поисковой оптимизации применяют так называемые популяционные алгоритмы (эволюционные алгоритмы; алгоритмы, использующие концепцию роевого интеллекта; алгоритмы, основанные на иных механизмах живой и неживой природы). Наиболее популярными и эффективными являются популяционные алгоритмы, вдохновленные живой природой [1] [2].

Цель данной научно-исследовательской работы — провести обзор популяционных алгоритмов, вдохновленных живой природой, для задачи поисковой оптимизации.

Для достижения поставленной цели необходимо решить следующие задачи:

- рассмотреть возможные способы решения задачи поисковой оптимизации;
- классифицировать популяционные алгоритмы, вдохновленные живой природой;
- сравнить описанные алгоритмы по предложенным критериям;
- отразить результаты сравнения рассмотренных алгоритмов в выводе.

1 Анализ предметной области

1.1 Описание популяционных алгоритмов

Популяционные алгоритмы предполагают одновременную обработку нескольких вариантов решения задачи оптимизации и представляют собой альтернативу классическим «траекторным» [3] поисковым алгоритмам, в которых в области поиска эволюционирует только один кандидат на решение этой задачи.

Все популяционные алгоритмы относятся к классу эвристических алгоритмов, то есть алгоритмов, для которых сходимость к глобальному решению не доказана, но экспериментально установлено, что в большинстве случаев они дают достаточно хорошее решение.

Термин *агент* используется в качестве общего названия членов популяции. В различных популяционных алгоритмах агенты называются индивидами, особями, частицами, муравьями, пчелами [1] [4].

Общая схема популяционных алгоритмов включает в себя следующие этапы.

1. *Инициализация популяции.* В области поиска тем или иным образом создается некоторое число начальных приближений к искомому решению задачи — инициализация популяции агентов;

2. *Миграция агентов популяции.* С помощью некоторого набора миграционных операторов, специфических для каждого из популяционных алгоритмов, агенты перемещаются в области поиска таким образом, чтобы в конечном счете приблизиться к искомому экстремуму оптимизируемой функции;

3. *Завершение поиска.* Проверется выполнение условий окончания итераций, и если они выполнены, завершаются вычисления, принимая лучшее из найденных положений агентов популяции за приближенное решение задачи. Если указанные условия не выполнены, происходит переход к выполнению этапа 2 [5].

При инициализации популяции могут быть использованы детерминированные и случайные алгоритмы. Формирование начальной популяции, агенты которой находятся вблизи глобального экстремума оптимизируемой функции, может существенно сократить время решения задачи. Однако обычно априорная информация о местоположении этого экстремума отсутствует, поэтому агентов начальной популяции распределяют равномерно по всей области поиска.

В качестве условия окончания поиска используют, как правило, условие достижения заданного числа итераций. Часто используют также условие стагнации алгоритма, когда лучшее достигнутое значение оптимизируемой функции не изменяется в течение заданного числа поколений. Могут быть использованы и другие условия, например условие исчерпания времени, отпущенного на решение задачи.

Из представленной общей схемы популяционных алгоритмов следует, что они обладают ярко выраженной модульной структурой, позволяющей легко получить большое число вариантов любого из алгоритмов путем варьирования и комбинирования правил инициализации популяции, миграционных операторов и условий завершения поиска [6].

Агенты популяции обладают следующими основными свойствами:

- автономность — агенты движутся в пространстве поиска, хотя бы частично, независимо друг от друга;
- стохастичность — процесс миграции агентов содержит случайную компоненту;
- ограниченность представления — каждый из агентов популяции обладает информацией лишь об исследуемой им части области поиска и, быть может, об окружении некоторых других агентов;
- децентрализация — отсутствие агентов, управляющих процессом поиска в целом;
- коммуникабельность — агенты тем или иным способом могут обмениваться между собой информацией о топологии (ландшафте) [7] оптимизируемой функции, выявленной в процессе исследования своей части области поиска.

Даже если стратегия поведения каждого из агентов популяции достаточно проста, указанные свойства агентов обеспечивают формирование так называемого роевого интеллекта популяции, проявляющегося в самоорганизации и сложном поведении популяции в целом.

Одной из особенностей популяционных алгоритмов оптимизации является то, что в подавляющем большинстве случаев для них имеется достаточно любопытная аналогия в человеческом обществе, живой или неживой природе. Так, известны популяционные алгоритмы эволюции разума, колонии муравьев, роя медоносных пчел, светлячков, гравитационного и электромагнитного поиска [8].

Для решения любой задачи оптимизации во множестве всех известных оптимизационных алгоритмов, включая популяционные алгоритмы, наверняка найдется хотя бы один алгоритм, который даст, как минимум удовлетворительные результаты. Однако нет и не может быть алгоритма, который мог бы обеспечить высокую эффективность при решении всех задач оптимизации.

Важнейшим понятием популяционных алгоритмов является понятие фитнес-функции (англ. fitnes-function). Важность функции обусловлена тем обстоятельством, что с ее помощью оценивают «качество» агентов популяции. Стратегически, в процессе миграции агенты движутся таким образом, чтобы приблизиться к глобальному экстремуму фитнес-функции.

С использованием понятия фитнес-функции можно сказать, что суть популяционных алгоритмов состоит в обеспечении более высокой средней приспособленности положений агентов данного поколения по сравнению с их приспособленностью в предыдущем поколении [9].

Одна из основных проблем популяционных алгоритмов — обеспечение баланса между интенсивностью поиска (скоростью сходимости алгоритма) и широтой поиска (диверсификацией поиска). Интенсификация поиска требует быстрой сходимости алгоритма, что означает быстрое уменьшение разнообразия популяции. Напротив, диверсификация поиска призвана обеспечить более широкий обзор пространства поиска и более высокую вероятность локализации глобального экстремума задачи.

Поскольку популяционные алгоритмы являются стохастическими, их эффективность, как правило, меняется в широких пределах в зависимости от удачности начального приближения, полученного на этапе инициализации

популяции. В этой связи для оценки эффективности данных алгоритмов используют многократные прогоны алгоритма, исходя из разных начальных приближений. Основными критериями эффективности популяционных алгоритмов являются надежность алгоритма — оценка вероятности локализации глобального экстремума, а также скорость его сходимости — оценка математического ожидания необходимого числа испытаний [1].

2 Классификация существующих алгоритмов

2.1 Роевой алгоритм

В основу роевого алгоритма (англ. Particle Swarm Optimization (PSO)) [10] положена социально-психологическая поведенческая модель толпы. Развитие алгоритма вдохновили такие задачи, как моделирование поведения птиц в стае и рыб в косяке. Целью было обнаружить базовые принципы, благодаря которым, например, птицы в стае ведут себя удивительно синхронно, меняя как по команде направления своего движения, так что стая движется как единое целое. К современному времени концепция алгоритма роя частиц развилась в высокоэффективный алгоритм оптимизации.

Существует значительное число алгоритмов роя частиц. В каноническом алгоритме, при определении следующего положения частицы учитывается информация о наилучшей частице из числа ее «соседей».

2.1.1 Канонический алгоритм роя частиц

Суть подхода, на котором основан алгоритм роя частиц, заключается в том, что глобальный максимум функции $f : \mathbb{R}^n \to \mathbb{R}$ ищется с помощью системы (роя), состоящей из m частиц. Частицы выполняют поиск, перемещаясь по пространству решений $\mathbb{R}^n$. Положение i-ой частицы задается вектором $x_i \in \mathbb{R}^n$, значение $f(x_i)$ определяет функцию качества этой частицы в текущий момент времени.

Каждая частица в рое обладает своей собственной скоростью $v_i \in \mathbb{R}^n$, которая определяет, как изменяются координаты частицы со временем:

$$x_i = x_i + \tau v_i,$$

где τ — некоторая единица измерения скорости (продолжительность одного такта работы алгоритма, например, можно положить $\tau = 1$). Ключевая особенность алгоритма роя частиц заключается в способе обновления скорости

отдельных частиц, которое выполняется по формуле

$$v_i = v_i + \alpha(p_i - x_i) + \beta(g - x_i).$$

Первое слагаемое в этой формуле представляет собой инерцию частицы — ее скорость на следующем временном шаге получается изменением текущей скорости. Вектор p_i, фигурирующий во втором слагаемом, служит простейшей моделью индивидуальной памяти — он равен лучшей точке траектории i-ой частицы за все время ее существования (от начала работы алгоритма до текущего момента времени). Говорят, что второе слагаемое реализует принцип простой ностальгии — каждая частица «хочет» вернуться в ту точку, где ею было достигнуто лучшее значение функции f [11].

Вектор g, участвующий в вычислении третьего слагаемого, представляет собой лучшую точку, обнаруженной за время своего существования всем роем, т. е. представляет собою некую коллективную память. Следовательно, само третье слагаемое определяет некоторую простую схему социального взаимодействия между отдельными частицами роя.

Другими словами, изменение скорости каждой частицы определяется как некая взвешенная сумма двух векторов, первый из которых направлен на лучшую точку, обнаруженную данной частицей, а второй — на лучшую точку, обнаруженную всем роем. Коэффициенты α и β могут выбираться из разных соображений. Численные эксперименты показали, что лучшей является вероятностная схема — либо оба коэффициента выбираются случайным образом из диапазона $[0, 1]$, либо значение α выбирается случайным образом из этого диапазона, а значение β полагается равным $1 - \alpha$.

Алгоритм роя частиц является, по сути, метаэвристикой, хорошо зарекомендовавшей себя при решении различных оптимизационных задач. Его отличительной особенностью от многих других алгоритмов является то, что для алгоритма роя частиц необходимо уметь вычислять только значение оптимизируемой функции, но не ее градиент. Т. е. функция, подлежащая оптимизации, не обязана быть дифференцируемой, более того, она может быть разрывной, зашумленной и т. п. С другой стороны, в силу того, что алгоритм оперирует понятием скорости частиц, необходимым условием его применимости является непрерывность области определения функции [9].

2.2 Алгоритм пчелиной колонии

2.2.1 Поведение пчел в природе

Пчелы могут выполнять поиск пищи одновременно во многих направлениях на расстоянии более чем 10 километров от улья. На первом этапе небольшое количество разведчиков выполняет случайный поиск мест с высоким содержанием нектара. При возвращении в улей пчелы исполняют специальный «танец» (являющийся, по сути, специфической формой коммуникацией между пчелами), в котором оказывается закодированной информация о расстоянии до найденного источника пищи, о направлении к этому источнику, и о качестве и количестве найденного там нектара. Чем ближе окажется источник и чем выше его качество, тем больше пчел последует за данным разведчиком. При возвращении в улей пчелы вновь исполняют свой танец, набирая новых последователей и т. д. Такая процедура позволяет пчелиной колонии достаточно эффективно выполнять на больших площадях поиск и сбор нектара [10].

Поведение пчелиной колонии является адаптивным. В то время как другие пчелы собирают нектар, разведчики продолжают искать новые перспективные места, что позволяет вести колонии мониторинг общей ситуации — как только некоторый источник нектара начинает истощаться, пчелы перестраиваются на другие источники, которые были за это время обнаружены разведчиками [11].

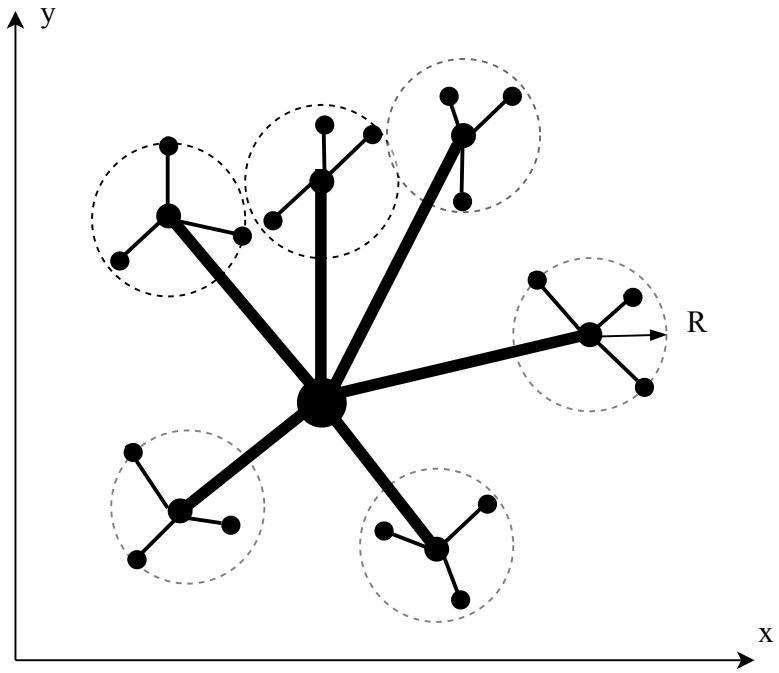


Рисунок 2.1 – Схематичное изображение стратегии разведки двумерного пространства (жирные линии — вылеты разведчиков, тонкие линии — уточнение решений рабочими пчелами)

2.2.2 Описание алгоритма

Рассмотрим задачу глобальной условной максимизации в гиперпараллелепипеде Π. Среди пчел $S = \{s_i, i \in [1 : |S|]\}$ выделяем $|S^\circ| < |S|$ пчел-разведчиков. Для простоты записи будем считать, что эти пчелы являются первыми в рое S:

$$S^\circ = \{s_i^\circ, i \in [1 : |S^\circ|]\}.$$

Остальные пчелы роя

$$S^w = \{s_i^w, i \in [1 : |S^w|]\} = \{s_i, i \in [|S^\circ| + 1 : |S|]\}$$

является рабочими пчелами; $|S^w| = |S| - |S^0|$.

Значения фитнес-функции $f(x)$, соответствующие текущему положению пчел s_i°, s_i^w, обозначим $f(x_i^\circ) = f_i^\circ$, $f(x_i^w) = f_i^w$.

Множество *элитных* участков обозначим $A^b = \{a_j^b, j \in [1 : |A^b|]\}$. Множество перспективных участков обозначим $A^p = \{a_k^p, k \in [1 : |A^p|]\}$.

Пологаем, что участки a_j^b, a_k^p представляют собой гиперпараллелепипеды в пространстве $R^{|x|}$, грани которых параллельны координатным плоскостям, центры находятся в точках x_j^b, x_k^p и длины сторон равны $2r_l^b$, $2r_l^p$ соответственно (рис. 2.2). Здесь $r_l^b, r_l^p, l \in [1 : |x|]$ — радиусы участков по соответствующим измерениям пространства поиска.

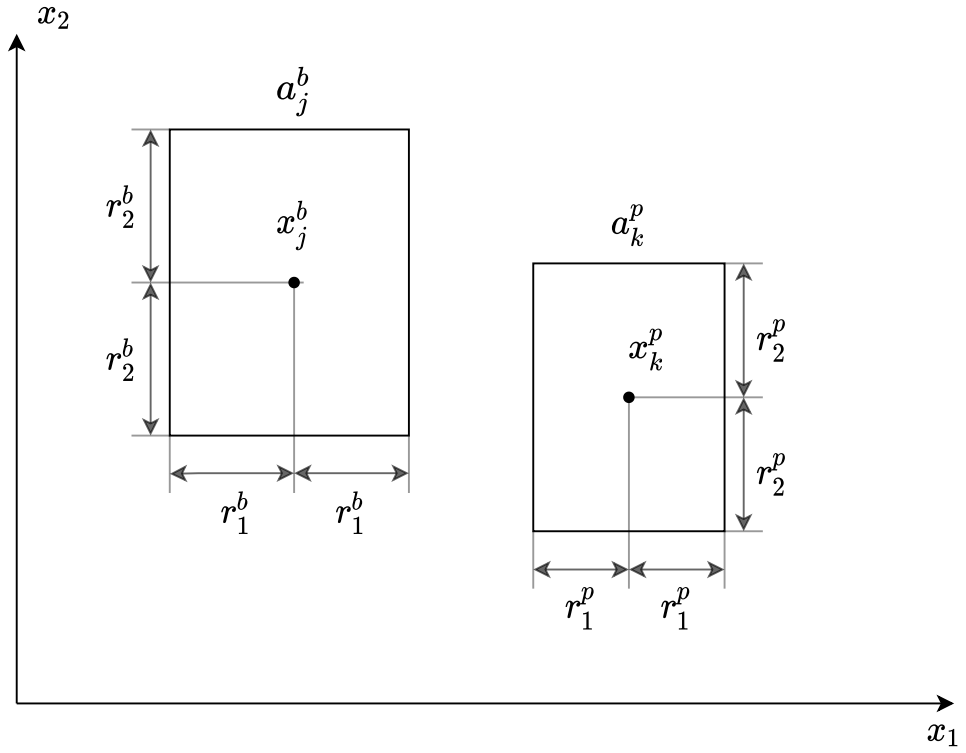


Рисунок 2.2 – К определению элитных и перспективных участков: $|x| = 2$

Во введенных обозначениях схема алгоритма имеет следующий вид.

1. Генерируем случайные точки $x_i, i \in [1 : |S^0|]$, равномерно распределенные во множестве Π, и отправляем в эти точки $|S^\circ|$, пчел-разведчиков s_i°. Вычисляем в точках x_i значения фитнес-функции f_i°, сортируем величины f_i° по убыванию и представляем в виде линейного списка;

2. Точки x_j, соответствующие первым $|A^b|$ элементам списка, объявляем центрами x_j^b элитных участков $a_j^b, j \in [1 : |A^b|]$. Аналогично точки $x_k, k \in [1 : |A^p|]$, соответствующие последующим $|A^p|$ элементам списка, объявляем центрами перспективных участков a_k^p;

3. В каждый из a_j^b элитных и a_k^p перспективных участков посылаем соот-

ветственно по n_b и n_p рабочих пчел $s_i^w \in S^w$, так что

$$n_b|A^b| + n_p|A^p| = |S^w|.$$

Координаты точек, в которые посылаются эти пчелы, полагаем случайными величинами, равномерно распределенными в соответствующих гиперпараллелепипедах a_j^b, a_k^p. Во всех точках, в которые посланы пчелы вычисляем значения фитнес-функции;

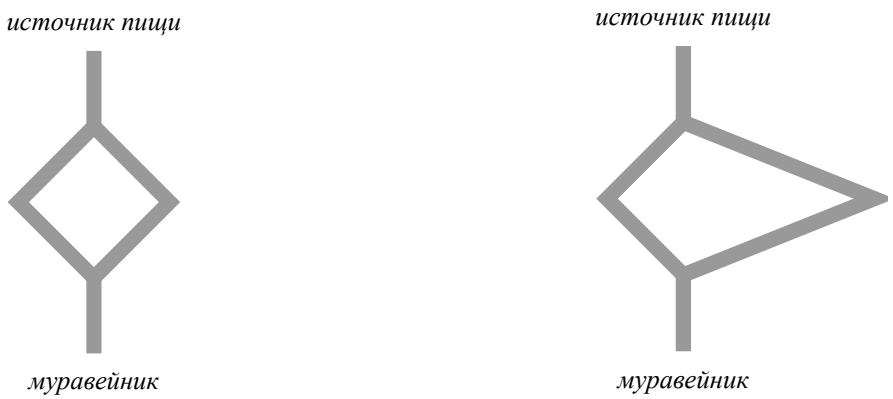
4. Проверяем выполнение условия окончания итераций. Если это условие выполнено, то в качестве решения задачи принимаем точку, соответствующую максимальному достигнутому значению фитнес-функции, и завершаем вычисления. В противном случае, приспособленности всех точек, найденных пчелами-разведчиками и рабочими пчелами, сортируем по убыванию и представляем в виде линейного списка. Переходим к шагу 2 [1] [12].

В качестве условия окончания итераций может быть использовано достижение заданного числа итераций t или стагнация вычислительного процесса в течение δ_t итераций.

2.3 Муравьиный алгоритм

2.3.1 Поведение муравьев в природе

Муравьиные алгоритмы основаны на следующей особенности поведения муравьев в природе. При поиске путей к источникам пищи муравьи помечают пройденный путь специальным веществом — *феромоном* [13]. Остальные муравьи, попадая на такой путь, с большой вероятностью начинают двигаться по этому пути, эта вероятность тем больше, чем больше концентрация феромона. Таким образом, феромон играет роль положительной обратной связи в системе — чем больше муравьев движется по помеченному пути, тем больше он становится привлекательным для других муравьев. В результате, через некоторое время большая часть муравьев будет передвигаться от



а) б)

Рисунок 2.3 – Схема опыта с двумя мостами

муравейника до найденного источника пищи по одному и тому же пути. Отличительными особенностями такого способа коммуникации являются:

1. Обмен информацией между муравьями является непрямым (безадресным), а производится путем изменения окружающей муравьев среды;

2. Информация является локально распределенной, другие муравьи могут получить доступ к ней, только переместившись в конкретную точку.

Интересной особенностью поведения муравьев является то, что найденный ими путь, при некоторых условиях, иногда оказывается *кратчайшим* постепенно за счет случайных флуктуаций, один из путей получает небольшое преимущество, которое очень быстро приводит к тому, что все муравьи переключаются на этот путь. Если повторять этот опыт несколько раз, то путем, связывающим муравейник с источником пищи. Этот факт был подтвержден опытами с настоящими муравьями. Схема одного такого классического опыта показана на рисунке 2.3. На рисунке 2.3.а имеется два пути одинаковой длины от муравейника до источника пищи. На начальном этапе опыта муравьи двигаются по обоим возможным направлениям, но оказывается, что в среднем каждый из двух таких путей (равной длины) выбирается примерно в половине случаев [14].

Иная ситуация наблюдается в том случае, если пути до пищи имеют разную длину 2.3.б. Результатом такого опыта, как правило, является нахождение муравьями кратчайшего из двух путей. Этот результат объясняется экспериментаторами тем, что короткие пути посещаются муравьями чаще,

чем длинные, поэтому они и феромоном помечаются чаще, что и приводит к усилению коротких путей по сравнению с длинными.

2.3.2 Описание муравьиного алгоритма

Описанная схема коммуникации муравьев легла в основу модели, разработанной Марко Дориго, и получившей название *муравьиного алгоритма*. В этой модели предполагается, что средой обитания муравьев является ненаправленный граф, в вершинах которого и располагаются муравьи. Все ребра графа являются помеченными, в качестве метки ребра используется действительное число, представляющее собой концентрацию феромона на этом ребре.

Перемещаются муравьи по ребрам графа. Муравей, находясь в вершине u графа, может переместиться на следующем шаге в вершину v только в том случае, если в графе имеется ребро (u, v). Выбор ребра для перемещения делается каждым муравьем на вероятностной основе, исходя из значений концентрации феромона на ребрах исходящих из вершины u — чем больше концентрация (относительно других ребер), тем вероятнее выбор этого ребра.

При этом, на выбор ребра могут влиять и другие факторы, обычно опеределяемые спецификой решаемой задачи. Например, каждый муравей может иметь ограниченную память, в которой хранится номер последней посещенной им вершины. Тогда, ребро, ведущее в эту вершину (по которому был произведен последний переход) исключается из списка возможных кандидатов на перемещение. Этот прием позволяет избежать короткого зацикливания муравья на одном ребре графа, но сохраняет возможность образования более длинных циклов[1].

Вероятность перехода из вершины i в вершину j определяется по формуле

$$p_{i,j} = \frac{(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}{\sum(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})},$$

где $\eta_{i,j}$ — расстояние от города i до j, $\tau_{i,j}$ — количество феромонов на ребре ij, α — параметр влияния длины пути, β — параметр влияния феромона.

Уровень феромона обновляется в соответствии с формулой

[1]Известен природный аналог такого рода циклов — так называемая «карусель смерти».

$$\tau_{i,j} = (1 - \rho)\tau_{i,j} + \Delta\tau_{i,j},$$

где ρ — доля феромона, которая испарится, $\tau_{i,j}$ — количество феромона на дуге ij, $\Delta\tau_{i,j}$ — количество отложенного феромона, вычисляется по формуле

$$\Delta\tau_{i,j} = \tau_{i,j}^0 + \tau_{i,j}^1 + ... + \tau_{i,j}^k,$$

где k — количество муравьев в вершине графа с индексами i и j.

Описание поведения муравьев при выборе пути:

- муравьи имеют собственную «память». Поскольку каждый город может быть посещён только один раз, то у каждого муравья есть список уже посещенных городов — список запретов. Обозначим через J_{ik} список городов, которые необходимо посетить муравью k, находящемуся в городе i;

- муравьи обладают «зрением» — видимость есть эвристическое желание посетить город j, если муравей находится в городе i. Будем считать, что видимость обратно пропорциональна расстоянию между городами;

- муравьи обладают «обонянием» — они могут улавливать след феромона, подтверждающий желание посетить город j из города i на основании опыта других муравьёв. Количество феромона на ребре (i, j) в момент времени t обозначим через $\tau_{i,j}(t)$;

- пройдя ребро (i, j), муравей откладывает на нём некоторое количество феромона, которое должно быть связано с оптимальностью сделанного выбора. Пусть $T_k(t)$ есть маршрут, пройденный муравьем k к моменту времени t, $L_k(t)$ — длина этого маршрута, а Q — параметр, имеющий значение порядка длины оптимального пути. Тогда откладываемое количество феромона может быть задано формулой

$$\Delta\tau_{i,j}^k = \begin{cases} Q/L_k, & \text{если } k\text{-ый муравей прошел по ребру } ij; \\ 0, & \text{иначе.} \end{cases}$$

На рисунке 2.4 приведена классификация ранее рассмотренных популяционных алгоритмов для задачи оптимизации поиска.

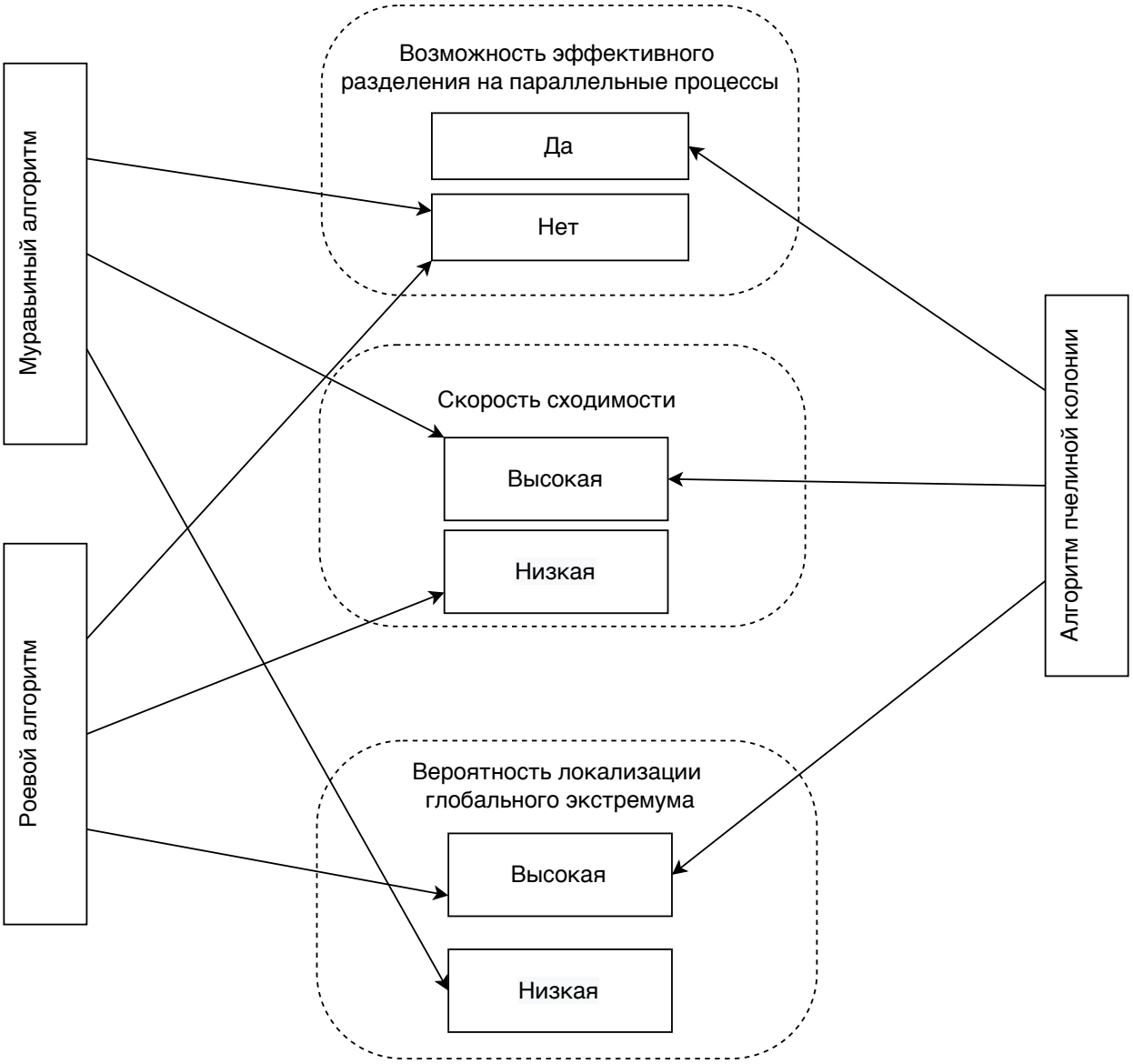


Рисунок 2.4 – Классификация популяционных алгоритмов, вдохновленных живой природой

Заключение

В ходе выполнения данной работы были рассмотрены популяционные алгоритмы, вдохновленные живой природой.

Среди популяционных алгоритмов, вдохновленных живой природой, было уделено внимание муравьиному и пчелиному алгоритмам, а также роевому алгоритму. Роевой алгоритм имеет наибольшую вероятность локализации глобального экстремума, однако низкая скорость сходимости делает его менее привлекательным среди пчелиного и муравьиного алгоритмов. Самая низкая вероятность локализации глобального экстремума оказалась у муравьиного алгоритма, однако данный алгоритм обладает высокой сходимостью.

На основе проделанной работы можно сделать вывод о том, что наиболее подходящим популяционным алгоритмом решения задачи поисковой оптимизации является алгоритм пчелиной колонии, так как данный алгоритм имеет высокую скорость сходимости, а также может быть реализован с помощью средств параллельного программирования, что делает его еще более эффективным.

Список литературы

1. Карпенко А. П. Современные алгоритмы поисковой оптимизации. МГТУ им. Н. Э. Баумана, 2017. с. 450.

2. Еремеев А. В. Популяционные алгоритмы // Федеральное государственное бюджетное образовательное учреждение высшего образования «Омский государственный университет им. Ф. М. Достоевского». 2016. С. 10—40.

3. Водолазский И. А., Егоров А. С. Роевой интеллект и его наиболее распространенные методы реализации // Москва, Молодой ученый №4. 2017. С. 1—8.

4. Ногин В. Д. Принятие решений в многокритериальной среде. Москва, «Физматлит», 2005. с. 142.

5. Штовба С. Д. Муравьиные алгоритмы // Донецк, Exponenta pro: математика в приложениях. 2015. С. 1—8.

6. Karaboga D., Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony algorithm // Kayseri, Journal of global optimization. 2007. P. 459—471.

7. Yang X. S. Nature-Inspired Algorithms and Applied Optimization. London, Springer International Publishing, 2018. p. 313.

8. Курейчик В. М., Кажаров А. А. Использование пчелиных алгоритмов // Известия Южного федерального университета. Технические науки. 2010. С. 1—7.

9. Teodorovi D., Dellorco M. Bee colony optimization–a cooperative learning approach to complex transportation problems. Publishing House of the Polish Operational and System Research, 2005. P. 51—60.

10. Ершов Н. М., Полуян С. В. Самоадаптация в алгоритмах роевой оптимизации // Вестник Российского университета дружбы народов. 2011. С. 1—14.

11. Леонов А. В., Литвинов Г. А. Применение алгоритма пчелиной колонии // Ростов-на-Дону, Инженерный вестник Дона. 2018. С. 1—11.

12. Yang X. S. Nature-Inspired Metaheuristic Algorithms. London, Springer International Publishing, 2010. p. 206.

13. Brownlee J. Clever Algorithms: Nature-Inspired Programming Recipes. Austin, Lulu, 2011. p. 438.

14. Ходашинский И. А., Горбунов И. В. Алгоритмы муравьиной и пчелиной колонии // Доклады Томского государственного университета систем управления и радиоэлектроники. 2017. С. 1—6.