

Лекционный материал по дисциплине
(заполняется по каждому виду учебного материала)

ДИСЦИПЛИНА Системный анализ данных в СППР
(полное наименование дисциплины без сокращений)

ИНСТИТУТ ИТ

КАФЕДРА Вычислительной техники
полное наименование кафедры

ВИД УЧЕБНОГО МАТЕРИАЛА Лекция
(в соответствии с пп.1-11)

ПРЕПОДАВАТЕЛЬ Сорокин А.Б.
(фамилия, имя, отчество)

СЕМЕСТР пятый
(указать семестр обучения, учебный год)

9. ЛЕКЦИЯ. РОЕВЫЕ АЛГОРИТМЫ

Роевые алгоритмы (РА), также, как и эволюционные, используют популяцию особей – потенциальных решений проблемы и метод стохастической оптимизации, который навеян (моделирует) социальным поведением птиц или рыб в стае или насекомых в рое. Аналогично эволюционным алгоритмам здесь начальная популяция потенциальных решений также генерируется случайным образом и далее ищется оптимальное решение проблемы в процессе выполнения РА. Первоначально в РА предпринята попытка моделировать поведение стаи птиц, которая обладает способностью порой внезапно и синхронно перегруппироваться и изменять направление полета при выполнении некоторой задачи. В отличие от ГА здесь не используются генетические операторы, в РА особи (называемые частицами) летают в процессе поиска в гиперпространстве поиска решений и учитывают успехи своих соседей. Если одна частица видит хороший (перспективный) путь (в поисках пищи или защиты от хищников), то остальные частицы способны быстро последовать за ней, даже если они находились в другом конце роя. С другой стороны, в рое, для сохранения достаточно большого пространства поиска, должны быть частицы с долей «сумасшествия» или случайности в своем поведении (движении).

Алгоритм роя является методом численной оптимизации, поддерживающий общее количество возможных решений, которые называются частицами или агентами, и перемещая их в пространстве к наилучшему найденному в этом пространстве решению, всё время находящемуся в изменении из-за нахождения агентами более выгодных решений.

Самая первая компьютерная модель роя частиц была придумана ещё в далёком 1986 Крейгом Рейнольдсом. Он, занимаясь созданием графической модели, придумал довольно простые правила поведения для частиц роя, действуя по которым, рой выглядел крайне похожим на реальный аналог птичьего роя. Классическая модель роя частиц была создана лишь в 1995 году Расселом Эберхартом и Джеймсом Кеннеди. Их модель отличается тем, что частицы-агенты роя, помимо подчинения неким правилам обмениваются информацией друг с другом, а текущее состояние каждой частицы характеризуется местоположением частицы в пространстве решений и скоростью перемещения.

9.1. Естественная мотивация

Работы в области роевого интеллекта были инспирированы исследованиями разнообразных реальных колоний. Например, метод роя частиц основан на исследовании и моделировании коллективного поведения стай птиц,

проведенного Крейгом Рейнольдсом. Целью этих исследований было построение реалистической анимации полета птиц. Особенностью поведения птичьих стай является то, что в них нет никакого единого центра управления, тем не менее вся стая демонстрирует весьма цельное поведение. Рейнольдсом были сформулированы три простых принципа, которых должна придерживаться каждая отдельная птица (рис.9.1) :

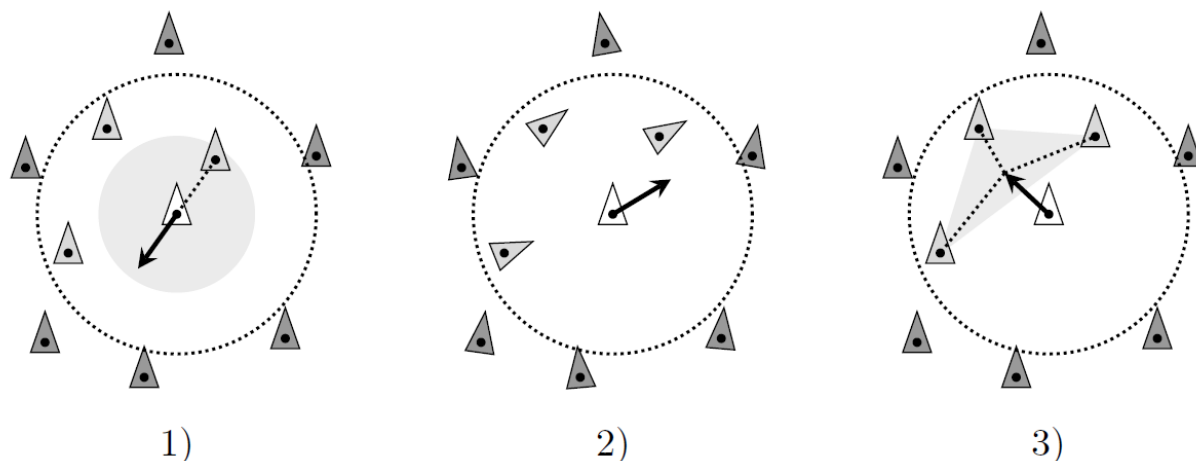


Рис. 9.1. Правила поведения птиц в модели Рейнольдса

- 1) каждая птица старается не приближаться к другим птицам (или другим объектам) на расстояние меньше некоторой заданной величины;
- 2) каждая птица старается выбрать свой вектор скорости наиболее близким к среднему вектору скорости среди всех птиц в своей локальной окрестности;
- 3) каждая птица старается расположиться в геометрическом центре масс своей локальной окрестности.

Первый принцип предназначен для того, чтобы избежать столкновений среди птиц, второй — координирует скорость и направление полета, третий — заставляет каждую птицу не отрываться от стаи (а в идеале — расположиться внутри стаи). Иногда эти принципы входят в противоречие друг с другом, в этих случаях целесообразно ввести приоритет отдельных правил и следовать правилам с наивысшим приоритетом.

Моделирование, основанное на перечисленных принципах, показывает весьма реалистическое поведение стай, способных, в частности, разделяться на несколько групп при встрече с препятствиями (домами, деревьями) и затем, спустя некоторое время, вновь соединяться в общую стаю.

Переход от моделирования коллективного поведения к коллективной оптимизации основан на следующей биологической идее: организмы объединяются в колонии для улучшения своих условий существования — каждый организм в колонии в среднем имеет больше шансов на выживание в борьбе с

хищниками, колония может более эффективно производить поиск, обработку и хранение пищи по сравнению с отдельными особями и т. д. Другими словами любая колония организмов в течение всего времени своего существования с той или иной степенью эффективности решает различные оптимизационные задачи, чаще всего — многокритериальные (например, максимизация количества пищи с одновременной минимизацией потерь от хищников).

9.1 Основной роевой алгоритм

РА использует рой частиц, где каждая частица представляет потенциальное решение проблемы. Поведение частицы в гиперпространстве поиска решения все время подстраивается в соответствии со своим опытом и опытом своих соседей. Кроме этого, каждая частица помнит свою лучшую позицию с достигнутым локальным лучшим значением целевой (фитнесс-) функции и знает наилучшую позицию частиц - своих соседей, где достигнут глобальный на текущий момент оптимум. В процессе поиска частицы роя обмениваются информацией о достигнутых лучших результатах и изменяют свои позиции и скорости по определенным правилам на основе имеющейся на текущий момент информации о локальных и глобальных достижениях. При этом глобальный лучший результат известен всем частицам и немедленно корректируется в том случае, когда некоторая частица роя находит лучшую позицию с результатом, превосходящим текущий глобальный оптимум. Каждая частица сохраняет значения координат своей траектории с соответствующими лучшими значениями целевой функции, которые обозначим y_i , которая отражает когнитивную компоненту. Аналогично значение глобального оптимума, достигнутого частицами роя, будем обозначать \hat{y} , которое отражает социальную компоненту. Таким образом, каждая частица роя подчиняется достаточно простым правилам поведения (изложенным ниже формально), которые учитывают локальный успех каждой особи и глобальный оптимум всех особей (или некоторого множества соседей) роя.

Каждая i -я частица характеризуется в момент времени t своей позицией $x_i(t)$ в гиперпространстве и скоростью движения $v_i(t)$. Позиция частицы изменяется в соответствии со следующей формулой:

$$x_i(t+1) = x_i(t) + v_i(t+1), \text{ где } x_i(0) \sim (x_{min}, x_{max}) \quad (9.1)$$

Вектор скорости $v_i(t+1)$ управляет процессом поиска решения и его компоненты определяются с учетом когнитивной и социальной составляющей следующим образом:

$$v_{ij}(t+1) = v_i(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}(t) - x_{ij}(t)] \quad (9.2)$$

Здесь $v_i(t)$ - j -ая компонента скорости ($j = 1, \dots, n_x$) i -ой частицы в момент времени t , $x_{ij}(t)$ - j -я координата позиции i -й частицы, c_1 и c_2 — положительные

коэффициенты ускорения (часто полагаемые 2), регулирующие вклад когнитивной и социальной компонент, $r_{1j}(t)$ и $r_{2j}(t) \sim (0,1)$ - случайные числа из диапазона $[0,1]$, которые генерируются в соответствии с нормальным распределением и вносят элемент случайности в процесс поиска. Кроме этого $y_{ij}(t)$ - персональная лучшая позиция по j -й координате i -ой частицы, а $\hat{y}_j(t)$ - лучшая глобальная позиция роя, где целевая функция имеет экстремальное значение.

При решении задач минимизации персональная лучшая позиция в следующий момент времени $(t + 1)$ определяется следующим образом:

$$y_i(t + 1) = \begin{cases} y_i(t) & \text{if } f(x_i(t + 1)) \geq f(y_i(t)) \\ x_i(t + 1) & \text{if } f(x_i(t + 1)) < f(y_i(t)) \end{cases} \quad (9.3)$$

где $f: R^{n_\infty} \rightarrow R$ фитнесс-функция. Как и в эволюционных алгоритмах фитнесс-функция измеряет близость текущего решения к оптимуму.

Существует два основных подхода в оптимизации роя частиц, под названиями *lbest* и *gbest*, отличающиеся топологией соседства, используемой для обмена опытом между частицами. Для модели *gbest* лучшая частица определяется из всего роя. Глобальная лучшая позиция (*gbest*) $\hat{y}_j(t)$ в момент t определяется в соответствии с

$$\hat{y}_j(t) \in \{y_0(t), \dots, y_{n_s}(t)\} | f(y_i(t)) = \min\{y_0(t), \dots, y_{n_s}(t)\} \quad (9.4)$$

где n_s – общее число частиц в рое.

В процессе поиска решения описанные действия выполняются для каждой частицы роя.

Алгоритм 9.1. Глобальный роевой алгоритм.

Создание инициализации n_x -мерного роя;

repeat

for каждой частицы $i = 1, \dots, n_s$ **do**

// определить персональную лучшую позицию

if $f(x_i) < f(y_i)$ **then**

$y_i = x_i$

end

// определить глобальную лучшую позицию

if $f(y_i) < f(\hat{y})$ **then**

$(\hat{y}) = y_i$

end

end

for каждой частицы $i = 1, \dots, n_s$ **do**

коррекция скорости согласно (9.2)

коррекция скорости согласно (9.1)

end

until критерий останова;

Рассмотрим влияние различных составляющих при вычислении скорости частицы в соответствии с (9.2). Первое слагаемое в (9.2) $v_i(t)$ сохраняет предыдущее направление скорости i -й частицы и может рассматриваться как момент, который препятствует резкому изменению направления скорости и выступает в роли инерционной компоненты. Когнитивная компонента $c_1 r_1 (y_i - x_i)$ определяет характеристики частицы относительно ее предистории, которая хранит лучшую позицию данной частицы. Эффект этого слагаемого в том, что оно пытается вернуть частицу назад в лучшую достигнутую позицию. Третье слагаемое $c_2 r_2 (\hat{y} - x_i)$ определяет социальную компоненту, которая характеризует частицу относительно своих соседей. Эффект социальной компоненты в том, что она пытается направить каждую частицу в сторону достигнутого роум (или его некоторым ближайшим окружением) глобального оптимума.

Графически это наглядно иллюстрируется для двумерного случая, как это показано на рис.9.2.



а) Скорость во времени t

b) Скорость во времени $t+1$

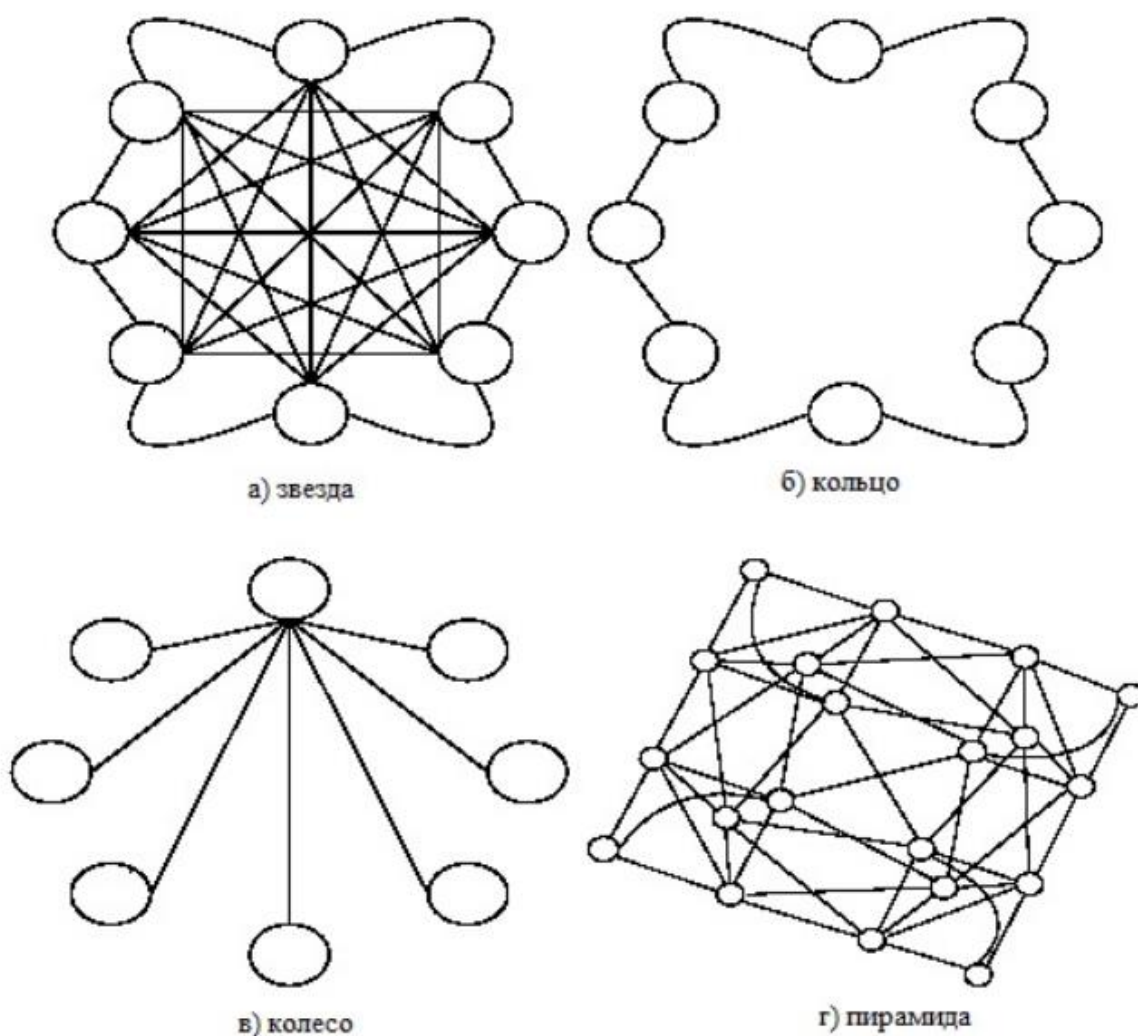
Рис. 9.1. Геометрическая иллюстрация изменения позиции и скорости частицы.

Представленный основной роевой алгоритм часто называют глобальным РА (Global Best PSO), поскольку здесь при коррекции скорости частицы используется информация о положении достигнутого глобального оптимума, которая определяется на основании информации, передаваемой всеми частицами роя. В противоположность этому подходу иногда используется локальный РА, где при коррекции скорости частицы используется информация, передаваемая только в каком-то смысле ближайшими соседними частицами роя.

9.2. Локальный роевой алгоритм

Локальный РА (Local Best PSO) использует для коррекции вектора скорости частицы только локальный оптимум, который определяется на множестве соседних (ближайших в некотором смысле) частиц. То есть считается, что данной частице может передавать полезную информацию только ее ближайшее окружение. При этом отношение соседства задается некоторой «социальной» сетевой структурой, которая образует перекрывающиеся множества соседних частиц, которые могут влиять друг на друга. Соседние частицы обмениваются между собой информацией о достигнутых лучших результатах и поэтому стремятся двигаться в сторону локального в данной окрестности оптимума. Характеристики роевого локального алгоритма сильно зависят от структуры используемой «социальной сети». Поток информации через «социальную сеть»

зависит от: 1) степени связности узлов сети, 2) числа кластеров, 3) среднего расстояния между узлами сети. В сильно связанной социальной сети большинство частиц могут общаться друг с другом, что способствует быстрому распространению информации о достигнутых оптимумах и вследствие этого высокой скорости сходимости процесса поиска решения в отличие от мало связанных сетей. Однако это часто достигается ценой преждевременной сходимости к локальным экстремумам. С другой стороны, для мало связанных сетей с большим числом кластеров возможна ситуация, когда пространство поиска покрывается неудовлетворительно, вследствие чего трудно получить глобальное оптимальное решение. Каждый кластер содержит сильно связанные особи и покрывает только часть пространства поиска. Сетевая структура обычно содержит несколько кластеров, которые слабо связаны между собой. Следовательно, исследуется информация только в ограниченной части пространства поиска. В РА используются различные социальные структуры, типовые сетевые структуры которых представлены на рис.12.2.



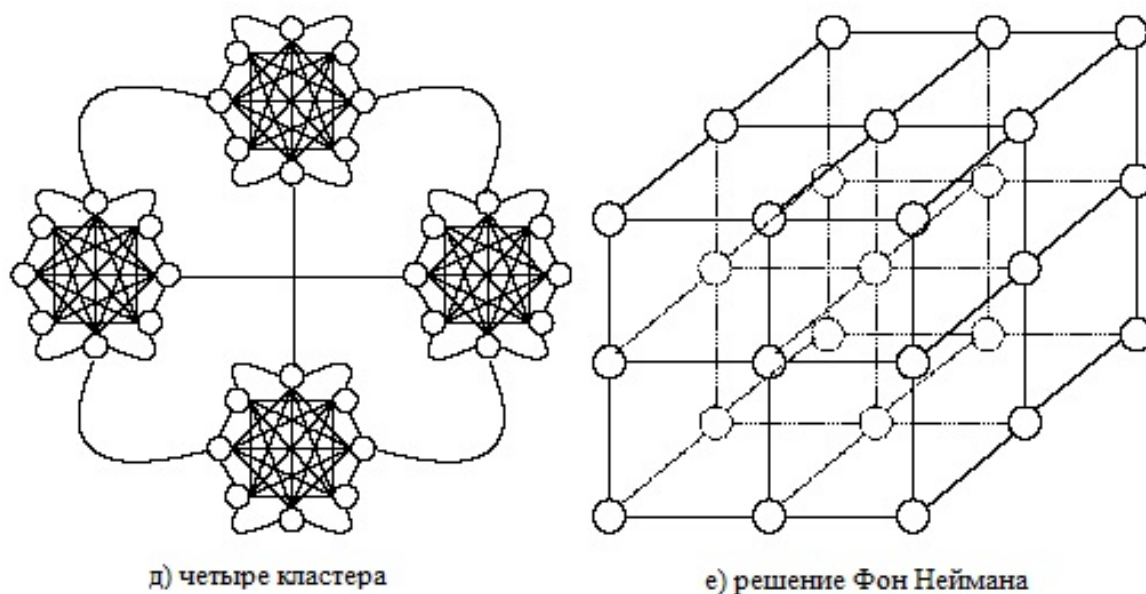


Рис. 9.2. Типовые социальные сетевые структуры.

На рис.9.2 а) представлена структура «звезда», где все частицы связаны друг с другом (образуют полный граф) и могут соответственно обмениваться информацией. В этом случае каждая частица стремится сместиться в сторону глобальной лучшей позиции, которую нашел рой. Очевидно, что основной РА, рассмотренный в предыдущем разделе, использует по умолчанию фактически структуру «звезда» на всем рое.

На следующем рис. 9.2. б) приведена сетевая структура «кольцо», где каждая частица общается со своими n_N ближайшими соседями. При $n_N = 2$ каждая частица связана только с двумя ближайшими соседями по кольцу, как это показано на рисунке. В этом случае каждая частица пытается сместиться в сторону лучшего соседа. Следует отметить, что множества соседних окружений перекрываются и вследствие этого возможен обмен информацией не только между ближайшими соседями. Поэтому данная структура позволяет находить и глобальный экстремум, но с меньшей скоростью. Данная структура лучше зарекомендовала себя при решении мульти-модальных задач (со многими экстремумами).

В сетевой структуре «колесо», показанной на рис.9.2 в), особи фактически изолированы друг от друга. Только одна частица выступает в качестве «фокальной точки», через которую идет обмен информации. В этом случае «фокальная частица» сравнивает характеристики всех соседних частиц и стремится в сторону лучшего соседа. Если новая позиция «фокальной частицы» имеет лучшие характеристики, то она сообщает это всем своим соседям. Данная сетевая структура замедляет распространение хороших решений через рой.

Социальная структура «пирамида» образует трехмерный каркас, как это показано на рис. 9.2 г).

Далее на рис.9.2 д) представлена четырех-кластерная социальная структура, в которой четыре кластера (клики) связаны друг с другом двумя соединениями.

Наконец, на рис.9.2 е) приведена социальная структура, где частицы объединены в решетку, которая часто называется социальной сетевой структурой фон Неймана. Данный тип согласно проведенным экспериментальным исследованиям показывает лучшие результаты при решении многих задач.

Следует отметить, что нет единого рецепта по использованию некоторой сетевой структуры. Для различных задач эффективными могут быть различные сетевые структуры, определяющие отношение соседства. В целом полно связная структура (звезда) дает лучшие результаты для унимодальных задач, в то время, как слабо связные структуры предпочтительнее для мульти модальных задач.

При коррекции скорости частицы в локальных РА вклад данной частицы пропорционален расстоянию между ней и лучшей позицией своего окружения, которое задается одной из рассмотренных сетевых структур. Таким образом, скорость частицы вычисляется следующим образом:

$$v_{ij}(t + 1) = v_i(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)] \quad (9.5)$$

где $\hat{y}_j(t)$ - лучшая позиция, которая найдена по координате j соседями частицы i .

Для модели *lbest* рой разделяется на перекрывающиеся окрестности частиц. При этом локальная лучшая позиция $\hat{y}_j(t)$ определяется как лучшая позиция в окружении N_i в соответствии с выражением

$$\hat{y}_j(t + 1) \in \{N_i\} | f(\hat{y}_j(t + 1)) = \min\{f(x)\}, \forall x \in N_i \quad (9.6)$$

где

$$N_i = \{y_{i-nN_i}(t), y_{i-nN_i+1}(t), \dots, y_{i-1}(t), y_i(t), y_{i+1}(t), \dots, y_{i+nN_i}(t)\} \quad (9.7)$$

при числе соседей nN_i . Здесь локальная лучшая позиция относится к лучшей позиции соседнего окружения.

Следует отметить, что, в основном, РА частицы в окружении не связаны друг с другом. Выбор соседей выполняется на основе индексов частицы. Однако, кроме этого, разработаны методы, где соседнее окружение формируется на основе пространственной близости.

Существуют, по крайней мере, две причины, по которым отношение соседства по индексам предпочтительнее:

1. экономия в вычислительных ресурсах, так как не требуется пространственное упорядочение частиц, где необходимы вычисления евклидовых расстояний между частицами со сложностью $O(n_s^2)$;

2. это способствует распространению информации о хороших решениях для всех частиц, безотносительно их текущего расположения в пространстве поиска.

Ниже представлен псевдокод локального РА.

Алгоритм 9.2. Локальный роевой алгоритм.

Создание инициализации n_x -мерного роя;

repeat

for каждой частицы $i = 1, \dots, n_s$ **do**

// определить персональную лучшую позицию

if $f(x_i) < f(y_i)$ **then**

$y_i = x_i$

end

// определить лучшую позицию окружения

if $f(y_i) < f(\hat{y})$ **then**

$\hat{y}_i = y_i$

end

end

for каждой частицы $i = 1, \dots, n_s$ **do**

коррекция скорости согласно (9.6)

коррекция скорости согласно (9.1)

end

until критерий останова не выполнен;

Рассмотренные две версии глобального и локального РА похожи в том смысле, что в обоих присутствует социальная компонента изменения скорости частицы, которая направляет ее (в конечном счете) в сторону глобальной лучшей позиции. Это возможно вследствие того, что локальные соседние области перекрываются.

Но существуют, по крайней мере, два основных различия между этими двумя подходами относительно их характеристик (свойств) сходимости:

1. благодаря большему взаимодействию частиц в глобальном РА он сходится быстрее, чем локальный РА. Однако эта быстрая сходимость достигается ценой сужения пространства поиска.

2. вследствие большего разнообразия потенциальных решений локальный РА менее подвержен преждевременной сходимости к локальным экстремумам.

Часто сетевые социальные структуры (например, такие как "кольцо") позволяют улучшить характеристики РА для многих задач.

9.3. Основные аспекты роевых алгоритмов

Далее рассмотрим некоторые аспекты представленных выше роевых алгоритмов А9.1 и А9.2, к которым относятся вопросы инициализации, условий останова, вычисления фитнес-функций и т.п. Как было показано ранее, процесс поиска решения в РА согласно А9.1 и А9.2 является итеративным, который продолжается пока не выполнено условие останова. Одна итерация содержит все шаги основного цикла **repeat ...until**, включая определение персональных и глобальных лучших позиций и коррекцию скорости каждой частицы. В каждой итерации производится оценка значений фитнес-функций частиц. Оценка качества потенциального решения выполняется на основе вычисления значения фитнес-функции, которая характеризует данную задачу оптимизации. В основном РА выполняется n_s вычислений фитнес-функции за итерацию, где n_s число частиц в рое.

На первом этапе РА производится инициализация роя и управляющих параметров алгоритма. При этом определяются начальные значения скоростей, позиций и персональных лучших позиций частиц, коэффициенты ускорения c_1 , c_2 и т.п. Для локальных РА необходимо также определить отношение соседства и размер окрестности ближайших соседей.

Обычно позиции частиц инициализируются таким образом, чтобы пространство поиска покрывалось равномерно. Следует отметить, что эффективность РА существенно зависит от начального разнообразия множества потенциальных решений, то есть от того, как первоначально покрыто пространство поиска и как распределены частицы в нем. Если некоторые области пространства поиска не покрыты начальным роем, то РА будет трудно найти оптимум в том случае, когда он расположен в не охваченном участке. В этом случае РА может найти такой оптимум благодаря моменту частицы, который может направить ее в неисследованную область.

Предположим, что оптимум расположен внутри области, определяемой двумя векторами x_{min} , x_{max} , которые представляют минимальные и максимальные значения по каждой координате. Тогда эффективным методом инициализации начальной позиции частиц является:

$$x(0) = x_{min,j} + r_j(x_{max,j} - x_{min,j}), \forall j = 1, \dots, n_x, \forall i = 1, \dots, n_x \quad (9.8)$$

где $r_j \sim U(0,1)$

Начальные скорости частиц при этом можно положить нулевыми $v_i(0) = 0$. В другом варианте начальные скорости можно инициализировать случайными

значениями из некоторого диапазона, но делать это необходимо осторожно. Случайная инициализация позиций частиц уже определяет начальное движение в случайных направлениях. Если, однако, выполняется случайная инициализация скоростей, то их значения не должны быть слишком большими, чтобы частицы не вышли из «зоны интереса», что может существенно ухудшить сходимость.

Персональная лучшая позиция для каждой частицы определяется позицией частицы в момент $t = 0$, то есть $y_i(0) = x_i(0)$. Различные схемы инициализации позиций частиц исследованы для покрытия пространства поиска: на основе последовательностей. При решении реальных задач важно, прежде всего, чтобы частицы равномерно покрывали пространство поиска.

Следующий аспект РА касается условия останова процесса поиска решения, где необходимо учитывать следующее:

- критерий не должен вызывать преждевременной сходимости РА в локальных оптимумах;
- критерий должен предотвращать чрезмерно большие вычисления вследствие частой оценки фитнес-функции частиц.

В настоящее время предложен ряд критериев останова, основными из которых являются следующие.

Останов по максимальному числу итераций (при превышении заданного порога). Очевидно, что в случае малого порога числа итераций процесс может остановиться до того, как будет найдено хорошее решение. Это критерий обычно используется совместно с критерием сходимости. Данный критерий полезен, когда необходимо за ограниченное заданное время найти лучшее решение.

Останов по найденному приемлемому решению. Предположим, что x^* представляет оптимум для целевой функции f . Тогда критерий окончания можно сформулировать в терминах близости найденного лучшего решения x к оптимуму $f(x_i) \leq |f(x^*) - \varepsilon|$, то есть при достижении достаточно малой ошибки ε . Значение порога ошибки ε необходимо выбирать осторожно. Если значение ε слишком велико, то очевидно процесс поиска может остановиться, если найдено не очень хорошее решение. С другой стороны, если значение слишком мало, то процесс может вовсе не сойтись. Это особенно характерно для основного (глобального) РА. Кроме этого, данный критерий предполагает априорное знание оптимального значения, которое не всегда известно, кроме случаев минимизации ошибки (например, в процессе обучения).

Останов по отсутствию улучшения решения за заданное число итераций. Есть различные способы измерения улучшения получаемых решений. Например, среднее изменение позиции частиц мало, то можно предположить, что процесс

поиска сошелся. С другой стороны, если среднее значение скорости частицы за некоторое число итераций примерно равно нулю, то возможны только незначительные изменения позиции частиц и процесс поиска можно остановить. К сожалению, данный критерий требует ввода двух параметров: 1) диапазона изменения итераций, 2) пороговое значение наблюдаемых величин.

Останов при стремлении нормализованного радиуса роя к нулю. Определим нормализованный радиус роя следующим образом:

$$R_{norm} = \frac{R_{max}}{diameter(S)}, \quad (9.9)$$

где $diameter(S)$ - диаметр начального роя и максимальный радиус R_{max} определяется следующим образом

$$R_{max} = ||x_m - \hat{y}||, m = 1, \dots, n_s$$

$$||x_m - \hat{y}|| \geq ||x_i - \hat{y}||, \forall i = 1, \dots, n_s$$

Можно считать, что алгоритм сошелся, если $R_{norm} < \varepsilon$. Если ε слишком велико, то процесс поиска может закончиться раньше, чем будет найдено хорошее решение. С другой стороны при малом значении ε может потребоваться слишком большое число итераций для формирования компактного роя.

Останов по малому значению наклона (крутизны) целевой функции. Рассмотренные критерии учитывают только относительное расположение частиц в пространстве поиска и не принимают во внимание информацию о крутизне целевой функции. Для учета изменений целевой функции часто используют следующее отношение:

$$f'(t) = \frac{f(\hat{y}(t)) - f(\hat{y}(t-1))}{f(\hat{y}(t))} \quad (9.10)$$

Если $f'(t) < \varepsilon$ для определенного числа последовательных итераций, то можно считать, что рой сошелся. Этот критерий сходимости, по мнению многих специалистов, превосходит приведенные ранее, так как он учитывает динамические характеристики роя. Однако использование крутизны целевой функции в качестве критерия останова может привести к преждевременной сходимости в локальном экстремуме. Поэтому целесообразно использовать данный критерий в сочетании с приведенным ранее критерием нормализованного радиуса роя.

Следует отметить, что сходимость по приведенным критериям, в общем случае, может не соответствовать достижению оптимума (глобального или локального). В данном случае сходимость означает, что рой достиг состояния равновесия, когда частицы стремятся к некоторой точке в пространстве поиска (в общем случае необязательно точке оптимума).

9.4 Основные параметры роевых алгоритмов

Эффективность РА зависит от ряда параметров, к которым относятся: размерность задачи, число частиц, коэффициенты ускорения, вес инерции, тип и размер соседнего окружения, число итераций, коэффициенты, определяющие вклад когнитивной и социальной компонент. В случае наложения ограничений на возможные скорости частиц необходимо также определить максимальные значения и некоторые коэффициенты. Далее рассмотрим основные параметры РА.

Размер роя, число частиц n_s , играет большую роль: чем больше частиц, тем больше разнообразие потенциальных решений (при хорошей схеме инициализации, обеспечивающей однородное распределение частиц). Большое число частиц позволяет покрыть большую часть пространства поиска за итерацию. С другой стороны, большое число частиц повышает вычислительную сложность итерации и при этом РА может вырождаться в случайный параллельный поиск. Хотя бывают случаи, что большее число частиц ведет к уменьшению числа итераций при поиске хороших решений. Экспериментально показано, что РА способны находить оптимальное решение с малым размером роя от 10 до 30 частиц. В общем случае оптимальный размер роя зависит от решаемой задачи и определяется экспериментально.

Размер соседнего окружения определяет степень влияния социальной компоненты в локальных РА. Чем меньше соседей у частицы, тем меньше ее взаимодействие с окружением. Малый размер определяет малую скорость сходимости, но дает большую надежность поиска оптимума. Чем меньше размер окружения, тем меньше чувствительность к локальным экстремумам. Для использования преимуществ малого и большого размера окружения часто стартуют с малым размером соседней окрестности и далее в процессе поиска размер окружения увеличивают пропорционально числу выполненных итераций. Такой подход обеспечивает хорошее начальное разнообразие и быструю сходимость частиц к перспективной области поиска.

Число итераций, обеспечивающее нахождение хорошего решения, зависит от решаемой задачи. При малом числе процесс поиска может не успеть сойтись. С другой стороны, большое число итераций, естественно, повышает вычислительную сложность.

Коэффициенты ускорения c_1 и c_2 вместе со случайными векторами r_1 и r_2 определяют вклад когнитивной и социальной компонент в результирующую скорость частицы. При $c_1 = c_2 = 0$ частицы летают с прежней скоростью пока не достигнут (по инерции) границы пространства поиска. Если $c_1 > 0$ и $c_2 = 0$, частица не зависит от остальных особей. Каждая частица находит свою лучшую

позицию в своем окружении путем замены лучшей позиции в том случае, если текущая позиция лучше. В случае $c_2 > 0$ и $c_1 = 0$, весь рой стремится к одной точке \hat{y} . Эксперименты показывают, что эффективность поиска увеличивается при балансе этих коэффициентов, т.е. при $c_1 \approx c_2$. Если $c_1 = c_2$, то частицы стремятся к средней точке между y_i и \hat{y} . Часто при решении задач полагают $c_1 = c_2$, но в общем случае отношение этих коэффициентов зависит от решаемой задачи. При $c_1 \gg c_2$ каждая частица больше стремится к своей лучшей позиции, что в результате ведет к чрезмерному блужданию частиц. Наоборот $c_2 \gg c_1$ определяет большее стремление частиц к глобальному экстремуму. Обычно значения c_1 , c_2 в процессе поиска постоянны, но иногда используются адаптивные схемы, когда величины c_1 , c_2 изменяются.

9.5. Сравнение роевых и генетических алгоритмов

Сила генетических алгоритмов, прежде всего, в параллельной природе поиска решений. В ГА реализована псевдослучайная форма поиска решения, которая сохраняет кратные решения, уничтожает неперспективные и дает приемлемые решения. Генетические операторы позволяют даже слабым решениям оставаться в числе потенциальных решений и генерировать из них приемлемые. Использование генетических операторов обеспечивает успех поиска хорошего решения. Все ГА используют некоторую форму рекомбинации, позволяющую порождать новые решения, которые сохраняют хорошие качества родителей и с большой вероятностью показывают хорошие характеристики. Кроссинговер является основным оператором ГА, в то время как мутация используется намного реже.

Большинство эволюционных методов используют следующую схему:

1. Случайная генерация начальной популяции.
2. Вычисление значения фитнес-функции для каждой особи, определяющего ее близость к оптимуму.
3. Репродукция популяции на основе полученных значений фитнес-функции.
4. Конец при выполнении условий останова. В противном случае переход на 2.

Из этой схемы видно, что РА имеет много общего с ГА. Оба алгоритма стартуют из случайно генерированной популяции и используют оценку значений фитнес-функции. В обоих алгоритмах эволюция популяции и соответственно поиск решения основаны на случайных методах. Оба не гарантируют успех в поиске решения. Однако РА не имеет генетических операторов, подобных кроссинговеру и мутации. Здесь потенциальные решения-частицы

взаимодействуют и изменяют свои скорости. Частицы имеют память, что очень важно для алгоритма. Механизмы передачи информации в РА и ГА совершенно различны. В ГА хромосомы обмениваются информацией друг с другом. Поэтому вся популяция движется как единая группа в область оптимума. В РА только глобальная (локальная) лучшая позиция передается другим частицам. Это единственный механизм передачи информации. В процессе эволюции ведется поиск лучшего решения. По сравнению с ГА все частицы в большинстве случаев стремятся к лучшему решению быстрее. РА имеет много общего с эволюционными вычислениями в общем и с ГА в частности. Все три метода стартуют из случайно генерированной начальной популяции и используют оценку значений фитнес-функции. Преимущество РА также в том, что они, как правило, проще в реализации и имеют меньше параметров управления.

В настоящее время РА применяются при решении задач численной и комбинаторной оптимизации (существует дискретный вариант РА), обучении искусственных нейронных сетей, построении нечетких контроллеров и т.д. в различных областях науки техники:

- управление энергетическими системами;
- решение NP-трудных комбинаторных проблем;
- задачи календарного планирования;
- оптимизация в мобильной связи;
- оптимизация процессов пакетной обработки;
- оптимизация многокритериальных задач;
- обработка изображений;
- распознавание образов;
- кластеризация данных;
- биоинформатика;
- проектирование сложных технических систем и т.д.