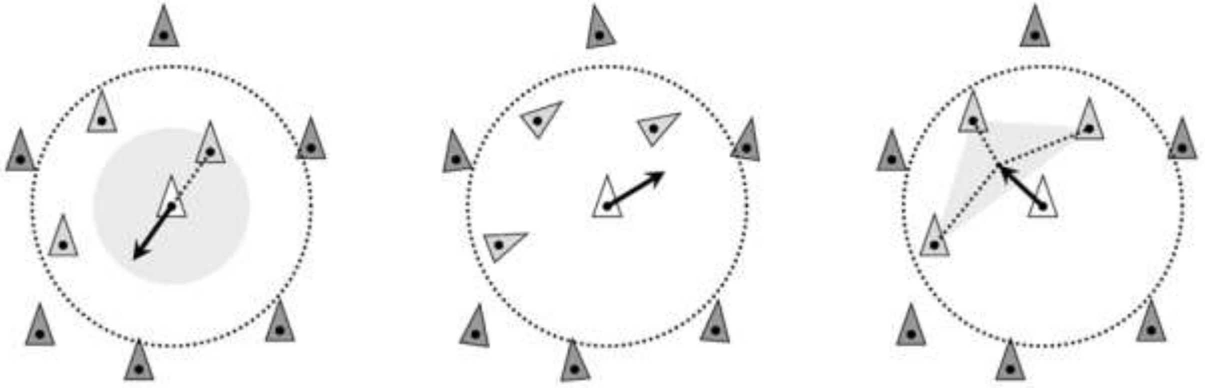
РОЕВЫЕ АЛГОРИТМЫ

Правила поведения птиц в модели Рейнольдса



Основной роевой алгоритм

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:

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

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

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

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 $\widehat{y_j}(t)$ –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:

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

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

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;

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

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

где n_s – общее число частиц в рое.

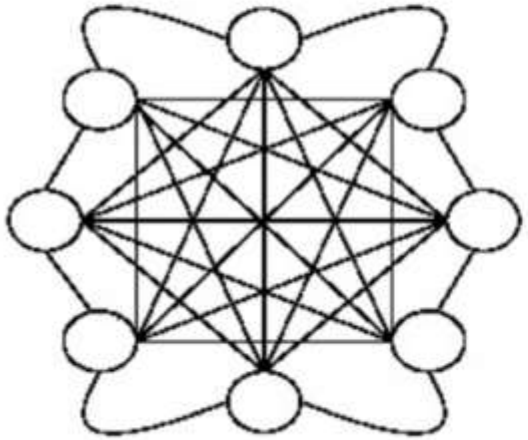
В процессе поиска решения описанные действия выполняются для каждой частицы роя.

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 роем (или его некоторым ближайшим окружением) глобального оптимума.

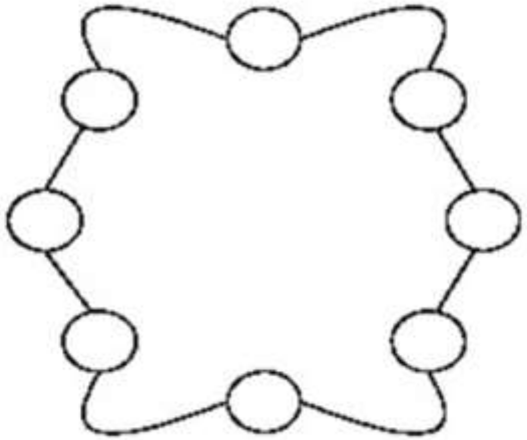
x_2
$\hat{y}(t)$
$x(t+1)$
новая
скорость
социальная скорость
инерциальная
скорость
$x(t)$
когнитивная скорость
$y(t)$
x_1

x_2
$\hat{y}(t+1)$
социальная скорость
$x(t+2)$
новая скорость
инерциальная
скорость
$x(t+1)$
когнитивная скорость
$x(t)$
$y(t+1)$
x_1

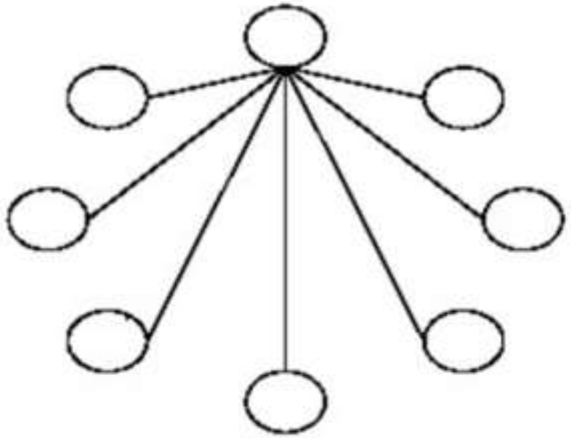
Локальный роевой алгоритм



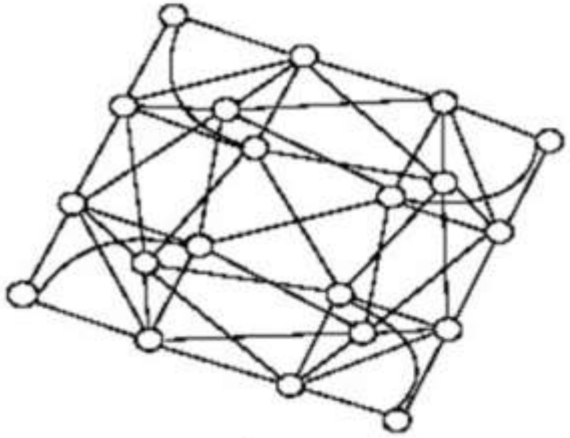
а) звезда



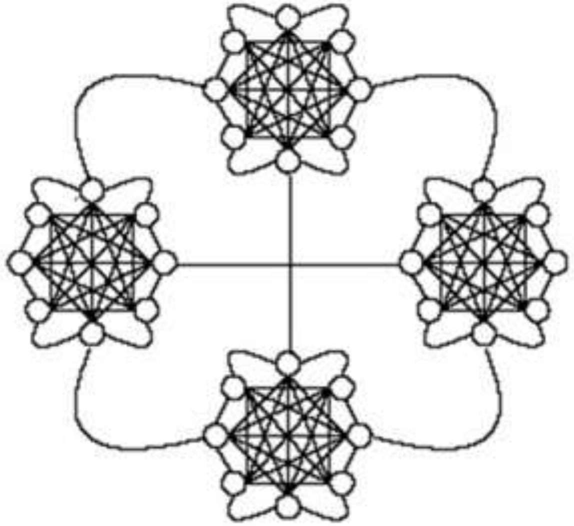
б) кольцо



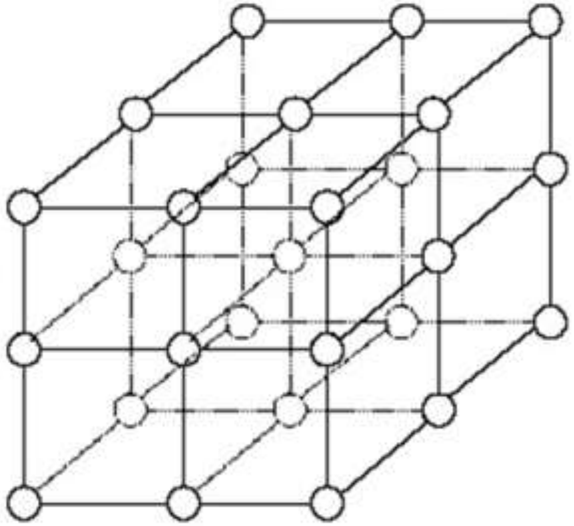
в) колесо



г) пирамида



д) четыре кластера



е) решение Фон Неймана

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:

$$v_{ij}(t+1) = v_i(t) + c_1 r_{1j}(t)\left[y_{ij}(t) - x_{ij}(t)\right] + \ c_2 r_{2j}(t)\left[\widehat{y_j}(t) - x_{ij}(t)\right] \quad (9.5)$$

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

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

$$\widehat{y_j}(t+1) \in \{N_i\} | f\left(\widehat{y_i}(t+1)\right) = min\{f(x)\}, \forall x \in N_i \quad (9.6)$$

где

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

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

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;

Основные аспекты роевых алгоритмов

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:

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

где $r_j \sim U(0,1)$

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.

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

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

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

$$R_{max} = \left||x_m - \hat{y}|\right|, \mathrm{m} = 1, \dots, \mathrm{n}_s$$
$$\left||x_m - \hat{y}|\right| \geq \left||x_i - \hat{y}|\right|, \forall i = 1, \dots, \mathrm{n}_s$$

Для учета изменений целевой функции часто используют следующее отношение:

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

Если $f'(t) < \varepsilon$ для определенного числа последовательных итераций, то можно считать, что рой сошелся.