

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)

11. ЛЕКЦИЯ. АЛГОРИТМ ПЧЕЛИНОЙ КОЛОНИИ.

Алгоритм пчелиной колонии – один из полиномиальных эвристических алгоритмов для решения оптимизационных задач в области информатики и исследования операций. Относится к категории стохастических бионических алгоритмов, основан на имитации поведения колонии медоносных пчел при сборе нектара в природе.

11.1. Естественная мотивация

В обычной колонии пчел, например, *Apis mellifera* (медоносная пчела домашняя), предполагается, что пчелы со временем выполняют разные роли. В типичном улье может быть от 5000 до 20 000 особей. Взрослые особи (в возрасте от 20 до 40 дней), как правило, становятся фуражирами (foragers). Фуражиры обычно выполняют одну из трех ролей: активные фуражиры, фуражиры-разведчики и неактивные фуражиры.

Активные фуражиры летят к источнику нектара, обследуют соседние источники, собирают нектар и возвращаются в улей. Разведчики обследуют местность вокруг улья (площадью до 50 квадратных миль) в поисках новых источников нектара. Примерно 10% пчел-фуражиров в улье задействованы в качестве разведчиков (рис. 11.1.).

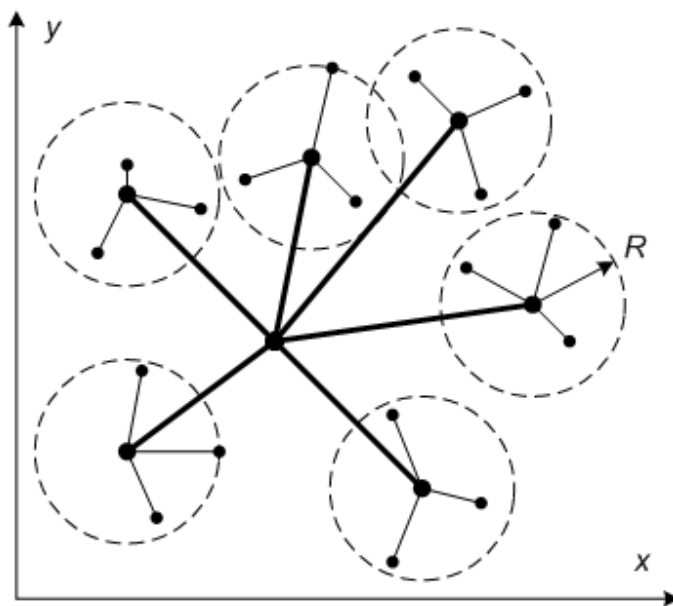


Рис. 11.1. Схематичное изображение стратегии разведки двумерного пространства (жирные линии — вылеты разведчиков, тонкие линии — уточнение решений рабочими пчелами)

В любой момент некоторое количество пчел-фуражиров неактивно. Они ждут неподалеку от входа в улей. Когда активные фуражиры и разведчики возвращаются в улей, то – в зависимости от качества источника нектара, который они только что посетили, - они могут исполнять виляющий танец (waggle dance)

перед ждущими неактивными пчелами. Есть довольно веские доказательства того, что этот виляющий танец несет информацию неактивным пчелам о местонахождении и качестве источника нектара. Неактивные фуражиры извлекают из виляющего танца эту информацию об источниках нектара и могут становиться активными фуражирами.

В целом, активный фуражир продолжает собирать нектар из конкретного источника до тех пор, пока он не истощится, после чего эта пчела становится неактивным фуражиром.

Исходя, из природного посыла, можно представить, что расположение глобального экстремума – это участок, где больше всего нектара, причем этот участок единственный, то есть в других местах нектар есть, но меньше. А пчелы живут не на плоскости, где для определения месторасположения участков достаточно знать две координаты, а в многомерном пространстве, где каждая координата представляет собой один параметр функции, которую надо оптимизировать. Найденное количество нектара представляет собой значение целевой функции в этой точке (в случае, если мы ищем глобальный максимум, если мы ищем глобальный минимум, то целевую функцию достаточно умножить на -1). Таким образом, пчелы могут собирать и отрицательное количество нектара.

11.2. Описание пчелиного алгоритма

Рассмотрим более подробно алгоритм применительно к нахождению экстремумов функции. При этом будем считать, что отыскивается глобальный максимум функции. В алгоритме каждое решение представляется в виде пчелы, которая знает (хранит) расположение (координаты или параметры многомерной функции) какого-то участка поля, где можно добыть нектар.

В начале алгоритма в точки, описываемые случайными координатами, отправляется некоторое количество пчел-разведчиков (пусть будет S пчел, от слова scout). Таким образом:

1 шаг: Необходимо задать количество пчел-разведчиков S . В точки со случайными координатами $X_{\beta,0} \in D$, отправляются пчелы-разведчики, где β – номер пчелы разведчика, $\beta \in [1: S]$, а 0 обозначает номер итерации в данный момент времени. Считаются значения целевой функции $F(X)$ в этих точках.

2 шаг: В области D с помощью полученных значений выделяют два вида участков (подобластей) d_{β} .

Первый вид содержит n лучших участков, которые соответствуют наибольшим или наименьшим значениям целевой функции, в зависимости от того решается задача на минимум или на максимум функции.

Второй m перспективных участков, соответствующих значениям целевой функции, наиболее близким к наилучшим значениям.

Подобласть d_β является подобластью локального поиска, представляющая собой гиперкуб в пространстве R^k с центром в точке $X_{\beta,0}$. Длина его сторон равна 2Δ , где Δ – параметр, называемый размером области локального поиска.

3 шаг: Сравнивается евклидово расстояние $\|X_{\beta,0} - X_{\gamma,0}\|$ между двумя агентами-разведчиками. Для точек $A = (x_1, x_2, \dots, x_n)$ и $B = (y_1, y_2, \dots, y_n)$ евклидово расстояние считается по формуле

$$d(A, B) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}$$

Если евклидово расстояние оказывается меньше фиксированной величины, то возможны два следующих варианта метода :

- поставить в соответствие этим агентам два различных пересекающихся участка d_β, d_γ (лучших и/или перспективных);
- поставить в соответствие тем же агентам один участок, центр которого находится в точке, соответствующей агенту с большим значением целевой функции. Из этих двух вариантов в работе используется второй вариант.

4 шаг: В каждый из лучших и перспективных участков посылаются по N и по M агентов, соответственно. Координаты этих агентов в указанных участках определяются случайным образом.

5 шаг: В полученных точках снова считается значение целевой функции $F(X)$, снова выбирается наибольшее или наименьшее значение. Точка, в которой значение функции является максимальным, становится центром новой подобласти.

6 шаг: Шаги 4 и 5 повторяются до тех пор, пока не будет получен искомый результат, если такой известен, либо до тех пор, пока полученные значения координат экстремумов и значений функции в них не повторятся τ раз, где τ — параметр останова.

Так же данный алгоритм, для нахождения максимума функции можно представить в виде блок-схемы представленной на рис. 10.2 (а, б).

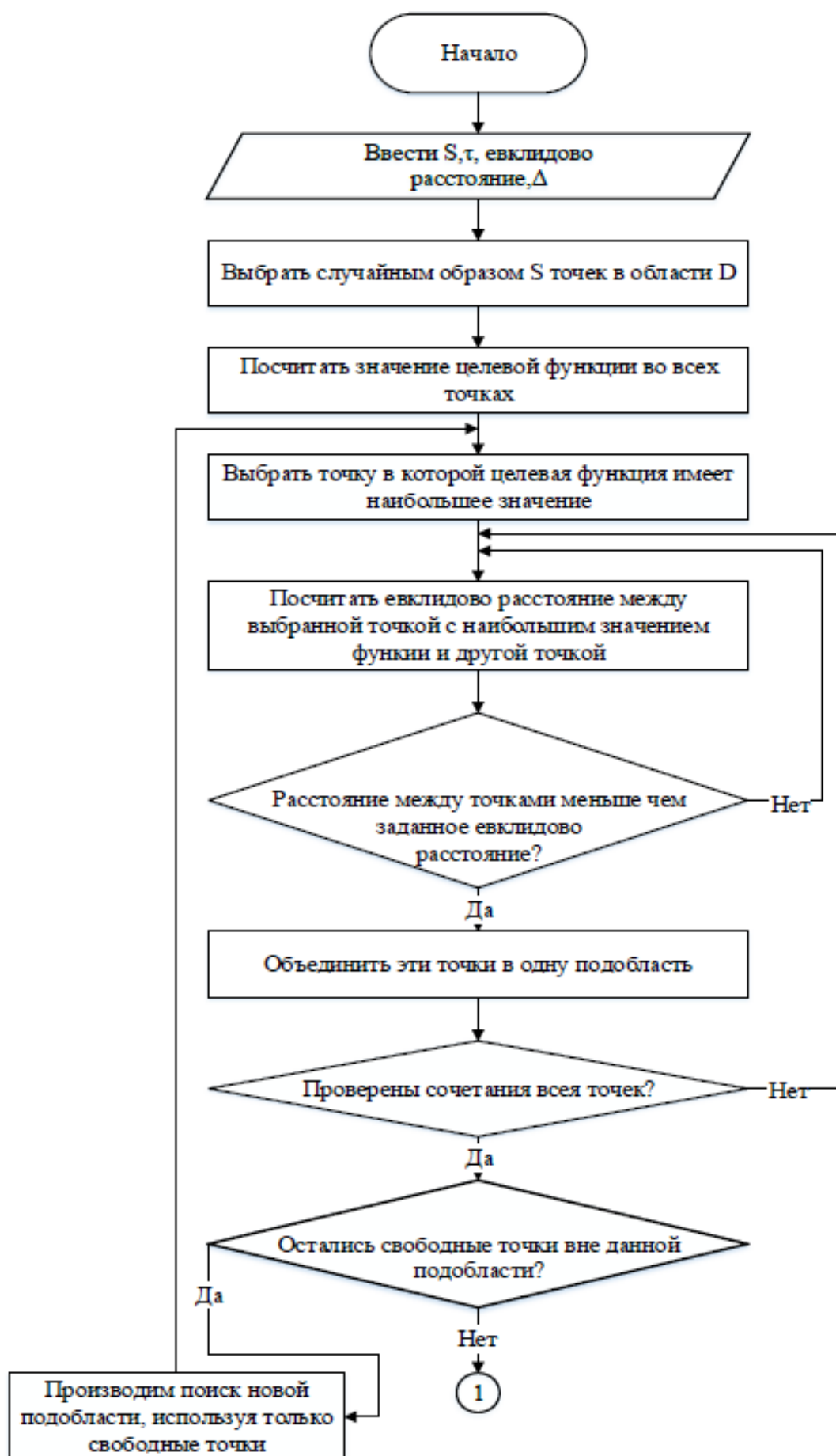


Рис. 10.2. (а). Блок-схема метода пчелиного роя

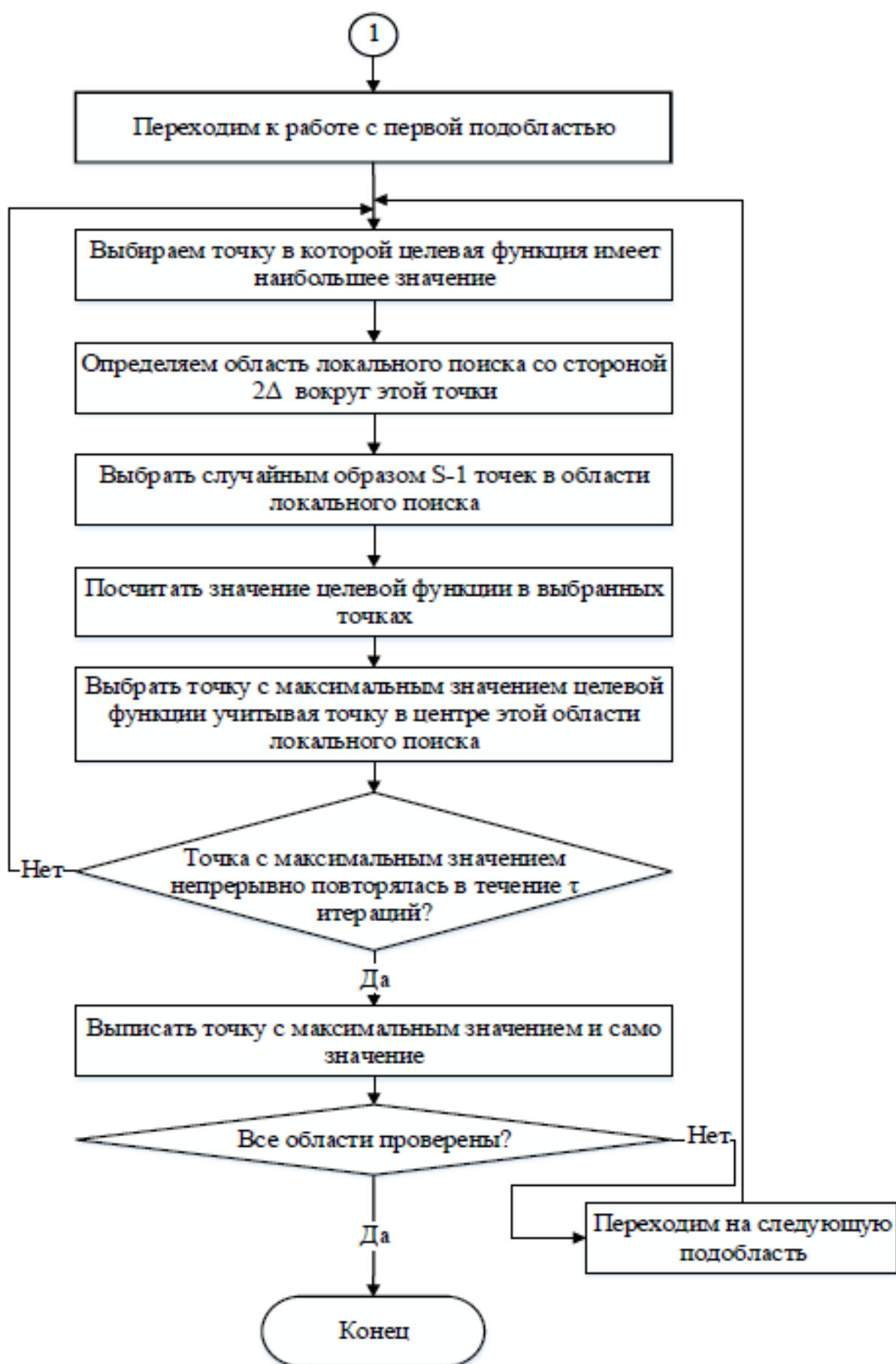


Рис. 10.2. (б). Продолжение блок-схемы метода пчелиного роя

11.3. Пример расчета итерации

Пусть в качестве целевой функции у нас выступает функция

$$f(x, y) = -(x^2 + y^2)$$

Исходник

Знак «-» в данном случае стоит, чтобы у функции был глобальный максимум, а не минимум. Известно, что глобальный (и единственный) максимум этой функции находится в точке $(0; 0)$, причем $f(0, 0) = 0$.

Зафиксируем необходимые параметры [15]:

- Количество пчел-разведчиков: 10;
- Количество пчел, отправляемых на лучшие участки: 5;
- Количество пчел, отправляемых на другие выбранные участки: 2;
- Количество лучших участков: 2;
- Количество выбранных участков: 3;
- Размер области для каждого участка: 10;

Пусть пчелы-разведчики попали на следующие, участки (список отсортирован по убыванию целевой функции):

$$f(15, 18) = -549$$

$$f(-30, -15) = -1125$$

$$f(22, -31) = -1445$$

$$f(18, 40) = -1924$$

$$f(-25, 47) = -2834$$

$$f(60, 86) = -10996$$

$$f(-91, -99) = -18082$$

$$f(17, -136) = -18785$$

$$f(-152, -1) = -22501$$

$$f(-222, 157) = -73933$$

Согласно, установленным параметрам, выбираются 2 лучшие точки:

$$f(15, 18) = -549$$

$$f(-30, -15) = -1125$$

Затем определяются другие 3 перспективных участка:

$$f(22, -31) = -1445$$

$$f(18, 40) = -1924$$

$$f(-25, 47) = -2834$$

В окрестности лучших точек будут отправлены по 10 пчел:

Для первой лучшей точки значение координат, которыми ограничивается участок будет:

$[15 - 10 = 5; 15 + 10 = 25]$ для первой координаты

$[18 - 10 = 8; 18 + 10 = 28]$ для второй координаты

И для второй точки:

$[-30 - 10 = -40; -30 + 10 = -20]$ для первой координаты

$[-15 - 10 = -25; -15 + 10 = -5]$ для второй координаты

Аналогично рассчитываются интервалы для выбранных участков:

$[12; 32] [-41; -21]$

$[8; 28] [30; 50]$

$[-35; 15] [37; 57]$

Необходимо заметить, что по каждой из координат размер области одинаков и равен 20, в реальности это не обязательно так.

В каждый из лучших интервалов отправляем по 5 пчел, а на выбранные участки по 2 пчелы. Причем, мы не будем менять положение пчел, нашедших лучшие и выбранные участки, иначе есть вероятность того, что на следующей итерации максимальное значение целевой функции будет хуже, чем на предыдущем шаге.

Теперь пусть на первом лучшем участке имеются следующие пчелы:

$$f(15, 18) = -549$$

$$f(7, 12) = -193$$

$$f(10, 10) = -100$$

$$f(16, 24) = -832$$

$$f(18, 24) = -900$$

Как видно, из расчета, уже среди этих новых точек есть такие, которые лучше, чем предыдущее решение .

Так же поступаем и со вторым лучшим участком, а затем аналогично и с выбранными участками. После чего среди всех новых точек снова отмечаются лучшие и выбранные, а процесс повторяется заново.

АЛГОРИТМ РОЯ СВЕТЛЯЧКОВ.

Существует около двух тысяч видов светлячков, которые способны светиться, производя ритмичные и короткие вспышки. Изучая поведение светлячков в естественной среде обитания, было замечено, что структура вспышек у каждого вида уникальна. Свечение у светлячков служит для коммуникации между особями. Каждая особь выделяет люциферин (особый пигмент, способствующий свечению), от количества которого зависит интенсивность свечения. Они используются для привлечения особей

противоположного пола, а также для привлечения потенциальной добычи. Кроме того, вспышки света могут выступать в качестве защитного механизма.

Алгоритм светлячков предложен в 2007 г. Янгом (X.-Sh. Yang). Модель поведения светлячков представлена следующим образом: все виды светлячков имеют возможность привлекать друг друга вне зависимости от пола; чем больше яркость светлячка, тем привлекательнее он для других особей; светлячки, с меньшей степенью привлекательности, притягиваются к более привлекательным особям; яркость излучения каждого светлячка, видимая другим светлячком, сокращается при увеличении расстояния между особями; если в окружении светлячка нет более ярких особей, то он перемещается произвольным образом.

Для формального описания модели используем вместо понятия «светлячок» понятие «агент». При инициализации поиска все агенты произвольным образом распределены в поисковом пространстве целевой функции. Каждый агент выделяет определенное количество люциферина и имеет свою собственную область принятия решений. Агент i рассматривает другого агента j как соседа, если он находится в пределах радиуса окрестности поиска агента i и уровень люциферина агента j выше, чем агента i , т.е. $I_j > I_i$. Локальная область принятия решений задается радиусом окрестности поиска для каждого i -го агента. Используя вероятностный механизм, каждый агент выбирает соседнего агента, у которого уровень люциферина выше, чем его собственный, и движется в его направлении. Иными словами, каждый агент движется в направлении того агента, у которого уровень свечения выше. Интенсивность свечения каждого агента определяется значением целевой функции в текущем положении. Чем выше интенсивность свечения, тем больше значение целевой функции. Кроме того, радиус окрестности поиска каждого агента зависит от количества агентов в этой области. Если в окрестности поиска находится малое количество агентов, то ее радиус увеличивается. В противном случае, радиус окрестности поиска сокращается. Иными словами, данный алгоритм имеет 4 глобальных этапа: начальное распределение агентов в пространстве поиска, обновление уровня люциферина, перемещение агентов в более перспективную область поиска, обновление радиуса окрестности поиска каждого агента (рис. 11.3).

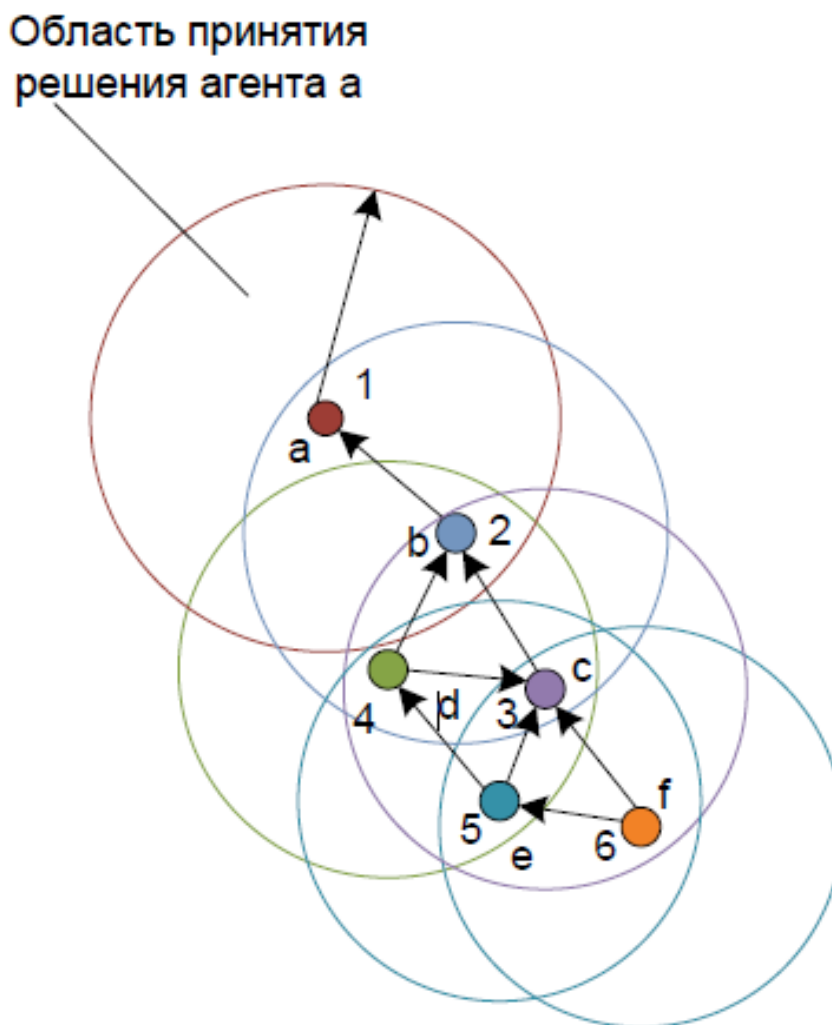


Рис. 11.3. Пример перемещения агента в пространстве поиска

На рис. 11.3 приведен направленный граф на 6 вершин, отражающий зависимость движения каждого агента от уровня люциферина, от которого зависит размер локальной области принятия решения. Агенты ранжируются в порядке возрастания уровня люциферина. В данном случае у агентов a, b, c и d уровень люциферина больше, чем у агента e , который расположен в пределах области локального принятия решений только агентов c и d , т.е. агент e имеет только два направления для перемещения.

Представим алгоритм, инспирированного поведением роя светлячков [19]:

Шаг 1. Инициализация входных параметров: параметра β для определения радиуса окрестности ($0 < \beta < 1$); параметра ρ определения уровня люциферина ($0 < \rho < 1$); параметра δ для генерации новой позиции ($0 < \delta < 1$); начального радиуса r_0 окрестности; максимального числа итераций n ; размера популяции k ; длины вектора позиции агента m ; минимальных x_{min} и максимальных x_{max} значений для вектора позиций.

Шаг 2. Задание функции цели $F(x) \rightarrow \min$, где x – вектор позиции жука-светляка.

Шаг 3. Создание случайным образом вектора лучшей позиции

$$x = (x_1, \dots, x_m), x_j = x_j^{\min} + (x_j^{\max} - x_j^{\min})rand(),$$

где $rand()$ – функция возвращающая равномерно распределённое случайное число в диапазоне $[0,1]$.

Шаг 4. Размещение в поисковом пространстве исходной популяции P ;

Шаг 5. Создание случайным образом вектора позиции x_k .

$$x_k = (x_{k1}, \dots, x_{km}), x_{kj} = x_j^{\min} + (x_j^{\max} - x_j^{\min})rand(),$$

Шаг 6. Изначально, все агенты имеют одинаковое количество люциферина. Инициализация количества люциферина l_k .

$$l_k = l_0$$

Инициализация радиуса окрестности r_k .

$$r_k = r_0$$

Шаг 7. Обновление уровня люциферина зависит от позиции агента в пространстве (значения его целевой функции). Все агенты на начальной итерации имеют одинаковый уровень люциферина, поэтому значение целевой функции каждого агента зависит от его положения в пространстве поиска. Уровень люциферина каждого агента увеличивается пропорционально измеряемым характеристикам агента (температура, уровень излучения). С точки зрения оптимизации это и является целевой функцией. Для моделирования процесса распада флуоресцирующего вещества производится вычитание части люциферина.

Вычисление уровня люциферина $l_k(t+1)$ (уровня свечения) i -го агента в момент времени t

$$l_k(t+1) = (1 - \rho)l_k(t) + \gamma F(x_k)(t+1), k \notin \overline{1, K}$$

где ρ – коэффициент ослабления уровня люциферина для моделирования процесса распада флуоресцирующего вещества, γ – коэффициент привлекательности светлячка, $F(x_k)(t+1)$ – значение целевой функции k -го агента в момент времени $t+1$.

Шаг 8. Миграция жуков-агентов (перемещение). Каждый агент выбирает того агента внутри радиуса окрестности поиска, у которого уровень люциферина выше, чем его собственный. Задание $N_i(t)$ – множества соседей i -го агента в момент времени t , r_k – радиуса окрестности поиска k -го светлячка в момент времени t .

Создание окрестности для k -го жука-светляка.

$$U_k = \{m | |x_m - x_k| < r_k, l_k < l_m, m \in \overline{1, K}\}$$

Шаг 9. При обновлении своего положения в пространстве поиска каждый агент на основе вероятностного механизма передвигается в направлении того агента, у которого уровень люциферина выше, чем его собственный (рис. 4.1).

Вычисление вероятности движения одного жука к другому.

$$P_{km} = \frac{l_m - l_k}{\sum_{s \in U_k} (l_s - l_k)}, m \in \overline{1, K}$$

Шаг 10. Выбор номера жука-светляка.

Если $\sum_{s=1}^{c-1} P_{ks} < rand() \leq \sum_{s=1}^c P_{ks}$, то $m = c$.

Агент k , используя метод колеса рулетки, выбирает агента m и перемещается в его направлении.

Шаг 11. Модификация позиции. Определяется обновленная позиция агента k по формуле

$$x_k(t+1) = x_k(t) + \delta \frac{x_m(t) - x_k(t)}{\|x_m(t) - x_k(t)\|}$$

Шаг 11. Модификация радиуса окрестности. Обновление радиуса окрестности r_k поиска по формуле

$$r_k = \min\{r_{max}, \max\{r_{min}, r_k + \beta(n_U - |U_k|)\}\},$$

где $|U_k|$ – размер текущей окрестности; n_U – параметр для управления количеством соседних агентов.

Шаг 11. Определение лучшего жука-светляка по функции цели.

$$k^* = \operatorname{argmin} F(x_k).$$

Если $F(x_k) < F(x^*)$, то $x^* = x_k$. Вывод результата x^* .

Соответственно, величины ρ , γ , δ , β , n_U – параметры алгоритма, значение которых определяется экспериментальным путем.

К положительным сторонам алгоритма можно отнести высокую точность получаемых результатов. К отрицательным качествам относится длительное время работы алгоритма при совсем не критичных значениях количества итераций и размером популяций. Наиболее важной особенностью алгоритма является зависимость между люциферинем и радиусом окрестности скопления агентов. Эта зависимость проявляется в том, что при увеличении люциферина уменьшается радиус поиска и при этом увеличивается точность работы алгоритма.