При минимизации двумерной функции Розенброка в области $D = \{X \mid x_1, x_2 \in [-1,2; 1,2]\}$ (см. Приложение Б) B-алгоритм также достиг 100 % успеха и потребовал по крайней мере в два раза меньшего числа итераций по сравнению с указанными алгоритмами. В случае четырехмерной функции Розенброка в той же области поиска B-алгоритму потребовалось большее число испытаний для достижения минимума со 100 % успехом.

3.3.3 Алгоритм колонии искусственных пчел

Алгоритм колонии искусственных пчел (*Artificial Bee Colony, ABC*) предложил Карабога (*D. Karaboga*) в 2005 г. для задачи непрерывной безусловной оптимизации. Впоследствии алгоритм модифицирован в работах того же автора и Бастурка (*B. Basturk*) и распространен на задачу условной оптимизации. ABC-алгоритм хорошо зарекомендовал себя при решении как унимодальных, так и многоэкстремальных задач оптимизации. Суть алгоритма состоит в комбинировании алгоритмов локального и глобального поиска с целью достижения оптимального баланса между ними.

Рассмотрим задачу глобальной условной минимизации в гиперкубе

$$D = \left\{X \mid x^- \le x_i \le x^+, i \in [1:|X|]\right\} \subset R^{|X|}.$$

Напомним, что произвольный гиперпараллелепипед Π с помощью линейных преобразований компонент вектора X легко трансформировать в указанный гиперкуб.

В ABC-алгоритме популяция содержит три группы пчел: рабочие пчелы, пчелы-исследователи и пчелы-разведчики. На разных этапах алгоритма рабочие пчелы превращаются в пчел-разведчиков и наоборот, так что в каждый данный момент времени популяция состоит из множеств рабочих пчел и пчел-исследователей, имеющих равные мощности. Каждому разрабатываемому участку (источнику нектара) ставится в соответствие одна рабочая пчела. Если некоторый участок перестает представлять интерес для популяции, то этот участок объявляется *заброшенным* и соответствующая рабочая пчела становится пчелой-разведчиком.

Введем следующие обозначения: $s_i^w, i \in [1:|S^w|]$ – рабочие пчелы; $s_j^o, j \in [1:|S^o|]$ – пчелы-исследователи; $|S| = |S^w| + |S^o|$ – общее число пчел в рое. Текущие положения рабочих пчел и пчел-исследователей определяют векторы их координат X_i^w, X_j^o соответственно, которым придаем смысл положений источников нектара. Количество нектара в его источнике полагаем равным значению фитнес-функции в соответствующей точке. Заброшенным объявляем участок, который не изменяется в течение заданного числа итераций.

Схема ABC-алгоритма включает в себя этап инициализации популяции и повторяющиеся итерации поискового процесса, каждая из которых состоит из следующих стадий: разработка источников рабочими пчелами; разработка источников пчелами-исследователями; поиск новых источников пчелами-разведчиками; завершение итераций. Рассмотрим указанные шаги алгоритма.

Инициализация популяции заключается в генерации в гиперкубе D начальных решений (координат участков) по формуле

$$X_i^w(0) = U_{|X|}(x^-; x^+),\ i \in [1:|S^w|] \tag{3.24}$$

и вычислении количеств нектара в них

$$\varphi(X_i^w(0)) = \varphi_i^w(0) = \begin{cases} \dfrac{1}{1+f(X_i^w(0))}, & f(X_i^w(0)) \geq 0, \\ 1+abs\left(f(X_i^w(0))\right), & f(X_i^w(0)) < 0. \end{cases} \tag{3.25}$$

Заметим, что формула (3.25) сводит задачу минимизации целевой функции $f(X)$ к задаче максимизации фитнес-функции.

Решение $X_i^w(0)$ интерпретируем как источник, подлежащий разработке рабочей пчелой s_i^w.

Стадия разработки участков. На этой стадии каждая рабочая пчела $s_i^w, i \in [1:|S^w|]$ пытается отыскать в окрестности своего текущего участка X_i^w новый участок X_i' с большим количеством нектара по следующей схеме:

1) генерируем точку X_i',

2) вычисляем количество нектара в этой точке и сравниваем его с количеством нектара в источнике X_i^w,

3) если количество нектара в новой точке больше, то перемещаем рабочую пчелу в эту точку.

Генерацию точки X_i' выполняем по формуле

$$x'_{i,j} = \begin{cases} x_{i,j}^w + U_1(-1;1)\left(x_{i,j}^w - x_{k,j}^w\right), & j = l, \\ x_{i,j}^w, & j \neq l, \end{cases} \tag{3.26}$$

где $l = U_1[1:|X|]$, $k = U_1[1:|S^w|]$, $k \neq i$. Другими словами, новый источник находим на прямой, параллельной случайно выбранной координатной оси $0x_l$. Источник на этой прямой располагаем на случайном расстоянии от источника X_i^w, пропорциональном расстоянию по оси $0x_l$ до случайно выбранного другого источника X_k^w (рис. 3.35).

Отметим следующее обстоятельство. Из формулы (3.26) следует, что чем меньше расстояние между источниками X_i^w, X_k^w, тем меньше расстояние между X_i^w и новой точкой X_i'. Поскольку на завершающих итерациях алгоритма пчелы s_i^w, $i \in [1:|S^w|]$ концентрируются вблизи глобального максимума фитнес-функции, указанное расстояние, а тем самым и шаг поиска с ростом числа итераций уменьшаются. В результате уменьшается широта обзора и повышается точность локализации искомого максимума.

Рассмотренная схема генерации точки X_i' может привести к ситуации, когда эта точка оказывается за пределами области поиска D. В соответствии с алгоритмом *ABC* в этом случае в качестве результирующей точки используем проекцию точки X_i' на ближайшую границу этой области.

Количество нектара φ_i' в найденных точках X_i', $i \in [1:|S^w|]$ вычисляем по формуле, аналогичной формуле (3.25). Если имеет место ситуация

$\varphi_i' > \varphi_i^w$, то полагаем $X_i^w = X_i'$, иначе оставляем вектор X_i^w без изменений.

Стадия разработки источников пчелами-исследователями. На этой стадии рабочие пчелы делятся информацией о полезности своих источников с пчелами-исследователями, которые на основе полученной информации принимают решение о том, к какому источнику им целесообразно отправиться для его дальнейшей разработки. Чем выше качество источника, тем больше пчел-исследователей к нему будет направлено. Пчелы-исследователи разрабатывают окрестности выбранного источника аналогично рабочим пчелам. Полученную информацию о качестве источника пчелы-исследователи передают рабочим пчелам, разрабатывающим данный источник. На основе этой информации те принимают решение, следует ли им переключаться на разработку нового источника или нет. После этого пчелы-исследователи возвращаются обратно в улей.

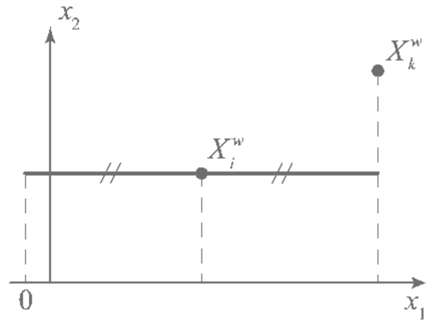

Рис. 3.35. К схеме локального поиска: участок X_i' с равной вероятностью находится в любой из точек выделенного отрезка

Выбор источника, к которому должна направиться пчела-исследователь, осуществляем алгоритмом рулетки. Так что вероятность ξ_i^o того, что пчела-исследователь s_i^o выберет источник X_i^w, определяет формула

$$\xi_i^o = \frac{\varphi_i^w}{\sum\limits_{j \in [1:|S^w|]} \varphi_j^w}, \; i \in [1:|S^o|]. \tag{3.27}$$

Пчела-исследователь осуществляет локальный поиск в окрестности выбранного источника $X_i^w, i \in [1:|S^w|]$ по рассмотренной схеме стадии разработки источников. Каждая из пчел-исследователей, разрабатывающая источник X_i^w, передает рабочей пчеле s_i^w, «ответственной» за этот источник, соответствующую полезность, вычисленную по формуле (3.25). На основании информации от всех пчел-исследователей, разрабатывающих источник X_i^w, рабочая пчела s_i^w принимает решение о переносе источника в новую точку.

Стадия поиска новых источников пчелами-разведчиками. Если в течение b_t итераций полезность источника $X_i^w, i \in [1:|S^w|]$ не удается улучшить, то источник переводим в разряд заброшенных. Соответствующая этому источнику рабочая пчела становится пчелой-разведчиком. Автор алгоритма *ABC* предлагает использовать значение параметра b_t, равное $|X| \cdot |S^w|$.

По схеме инициализации популяции для всех пчел-разведчиков генерируем новые источники и вновь переводим всех этих пчел в ранг рабочих пчел.

Стадия завершения итераций. На данной стадии проверяем выполнение условия завершения итераций, и если оно выполнено, заканчиваем вычисления. В противном случае возвращаемся к стадии разработки источников рабочими пчелами. Как всегда, в качестве условия окончания итераций может быть использовано, например, условие достижения максимального числа итераций или условие наступления стагнации.

Общая схема алгоритма *ABC* имеет следующий вид.

1) По формулам (3.24), (3.25) инициализируем популяцию s_i^w рабочих пчёл в источниках $X_i^w, i \in [1:|S^w|]$.

2) Вычисляем исходное приближение к решению

$$\varphi(\tilde{X}^*) = \max_{i \in [1:|S^w|]} \varphi(X_i) = \tilde{\varphi}^*.$$

3) Для каждой рабочей пчелы s_i^w по формуле (3.26) генерируем точку X_i' и по формуле, аналогичной формуле (3.25), вычисляем ее полезность φ_i'. Если $\varphi_i' > \varphi_i^w$, то источник нектара из точки X_i^w переносим в точку X_i'.

4) По формуле (3.27) для каждого из источников нектара X_i^w вычисляем вероятность ξ_i^o того, что пчела-исследователь выберет его для разработки. С помощью алгоритма рулетки назначаем каждой пчеле-исследователю источник, который ей следует разрабатывать, и отправляем ее к этому источнику.

5) Для каждой пчелы-исследователя по формуле (3.26) генерируем точку X_i' и по формуле, аналогичной формуле (3.25), вычисляем ее полезность φ_i'.

6) На основании информации о результатах локального поиска, выполненного пчелами-исследователями, разрабатывающими каждый данный источник нектара, принимаем решение о перемещении или не перемещении соответствующей рабочей пчелы в новый источник.

7) Если решение в источнике X_i^w не улучшается в течение b_t итераций, то данный источник исключаем из процесса разработки и соответствующую ему рабочую пчелу объявляем пчелой-разведчиком.

8) Для каждой пчелы-разведчика по формуле (3.24) генерируем новый источник нектара и объявляем ее рабочей пчелой, разрабатывающей этот источник.

9) В каждом текущем источнике нектара $X_i^w, i \in [1:|S^w|]$ вычисляем значение фитнес-функции $\varphi(X_i)$ и определяем максимальное из этих значений $\varphi(X_{i_j}^w) = \max_{i \in [1:|S^w|]} \varphi(X_i^w)$. Если $\tilde{\varphi}^* < \varphi(X_{i_j}^w)$, то полагаем $\tilde{\varphi}^* = \varphi(X_{i_j}^w)$.

10) Проверяем условие окончания итераций. Если это условие выполнено, то принимаем $X^* = \tilde{X}^*$ и завершаем вычисления. Иначе переходим к шагу 3.

Кроме общих для всех популяционных алгоритмов, свободным параметром алгоритма является параметр b_t – максимальное число неудачных итераций.

Пример 3.6. Рассмотрим задачи минимизации сферической функции, а также функций Розенброка, Растригина и Экли (Приложение Б). Множеством допустимых значений вектора варьируемых параметров во всех случаях является гиперкуб

$$D = \{X \mid -20 \le x_i \le 20, i \in [1:|X|]\}.$$

Полагаем, что фитнес-функция обратна целевой функции.

Используем число рабочих пчел (а значит, и источников нектара), равное $|S^w| = 20$, так что общее число пчел в рое $|S| = 40$. Максимальное число неудачных итераций в источнике принимаем равным $b_t = |S^w| \cdot |X|$. В качестве критерия окончания итераций используем стагнацию вычислительного процесса в течение $\delta_t = 20$ итераций. Размерность вектора X варьируем в пределах от двух до 16.

Скорость сходимости алгоритма *ABC* при решении указанных задач иллюстрируют рис. 3.36, 3.37. Рисунки показывают высокую скорость сходимости алгоритма даже в случае высокой размерности пространства поиска и даже для таких сложных многоэкстремальных функций, как функции Растригина и Экли.

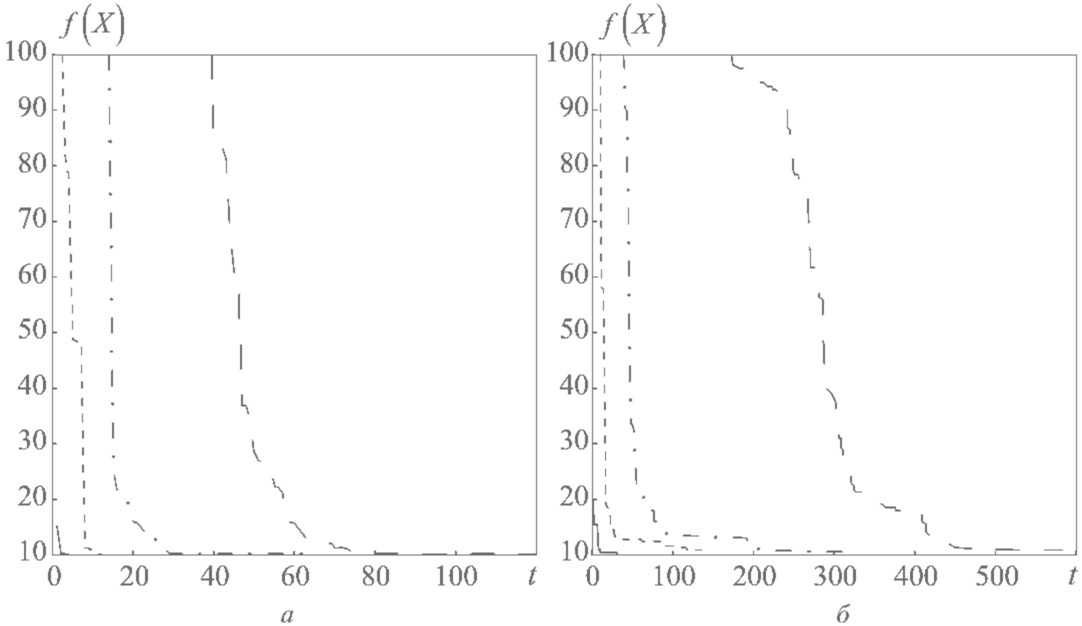

Рис. 3.36. Сходимость *ABC*-алгоритма: *а* – сферическая функция: —— $|X| = 2$, ···· $|X| = 4$, -·- $|X| = 8$, – – $|X| = 16$; *б* – функция Розенброка: —— $|X| = 2$, ···· $|X| = 4$, -·- $|X| = 8$, – – $|X| = 16$

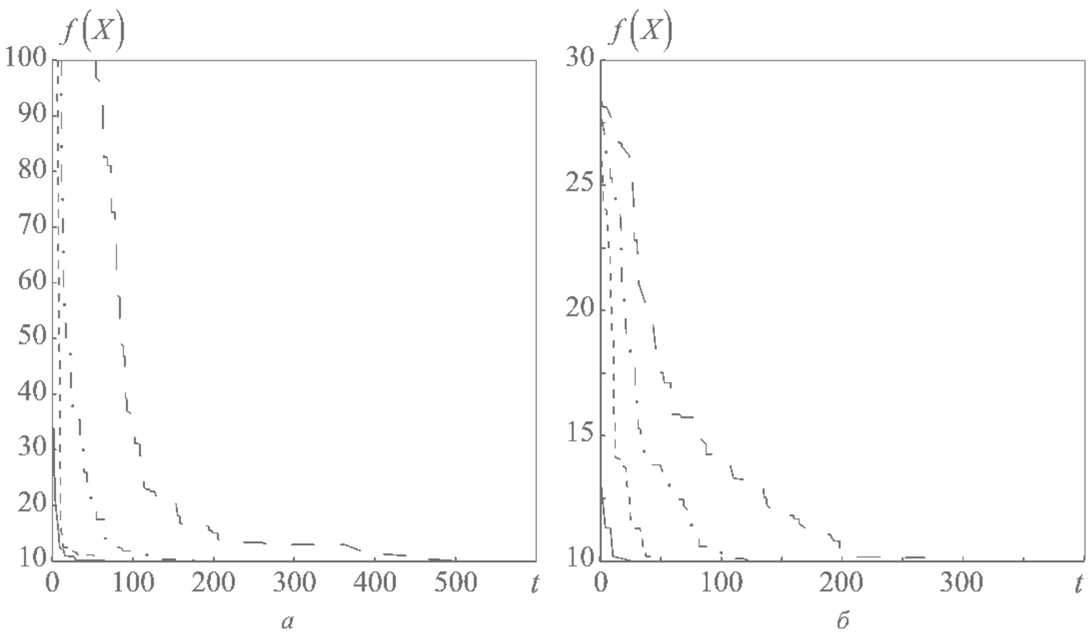

Рис. 3.37. Сходимость *ABC*-алгоритма: *а* – функция Растригина: —— $|X| = 2$, ···· $|X| = 4$, -·- $|X| = 8$, – – $|X| = 16$; *б* – функция Экли: —— $|X| = 2$, ···· $|X| = 4$, -·- $|X| = 8$, – – $|X| = 16$

Для исследования зависимости скорости сходимости алгоритма от размерности задачи воспользуемся мультистартом с числом запусков алгоритма $\hat{n} = 30$. В качестве меры скорости сходимости используем усредненную величину t_{end}, то есть усредненное число итераций до начала стагнации вычислений. Результаты исследования иллюстрируют табл. 3.3 и рис. 3.38.

Таблица 3.3 **Среднее число итераций $\overline{t}_{end}$**

X	Среднее число итераций $\overline{t}_{end}$			
	Сферическая функция	Функция Розенброка	Функция Растригина	Функция Экли
2	82	33	116	154
4	155	92	239	287
8	312	254	463	574
16	626	589	942	1137

Табл. 3.3 и рис. 3.38 показывают, что в широком диапазоне варьирования размерности задачи имеет место практически линейная зависимость величины $\overline{t}_{end}$ от этой размерности.

Результаты исследования показывают, что *ABC*-алгоритм обеспечивает 100% локализацию глобального экстремума таких топологически сложных функций, как функции Растригина и Экли, даже при их высокой размерности, равной 16. В то же время алгоритм показывает гораздо худшие результаты при оптимизации овражной функции Розенброка, особенно при ее высокой размерности. Это обстоятельство связано с тем, что дно оврага функции Розенброка не ориентировано вдоль какой-либо оси системы координат $0x_1x_2...x_{|X|}$, а локальный поиск в алгоритме *ABC* осуществляется вдоль этих координатных осей. В результате пчела, расположенная вблизи дна оврага, постоянно «наталкивается» на крутые склоны оврага и не может улучшить свою позицию. В такой ситуации найти глобальный экстремум могут только пчелы-разведчики, осуществляющие глобальный поиск. Однако точность их способа поиска намного ниже точности локального поиска рабочих пчел. Исправить ситуацию мог бы другой алгоритм локального поиска, например, известный алгоритм Хука – Дживса или алгоритм Розенброка.

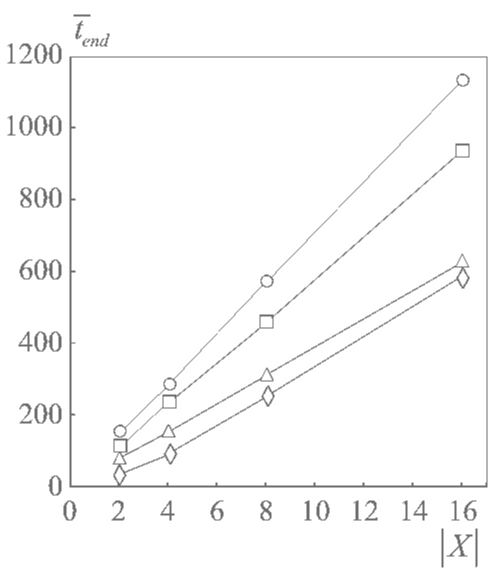

Рис. 3.38. Скорость сходимости *ABC*-алгоритма $\overline{t}_{end}$ в функции размерности задачи $|X|$: △ – сферическая функция; ◊ – функция Розенброка; □ – функция Растригина; ○ – функция Экли

Значительный интерес представляет зависимость вероятности локализации глобального экстремума ξ в функции размера популяции $|S|$. В качестве оценки $\tilde{\xi}$ этой вероятности используем отношение числа успешных запусков алгоритма (когда глобальный максимум был локализован) к общему числу запусков $\hat{n} = 30$.

Результаты исследования для функции Растригина размерности $|X| = 8$ представлены в табл. 3.4. Исследование выполнено при следующих значениях свободных параметров алгоритма: $b_t = |S^w| \cdot |X|$; $\delta_t = 10$.

Таблица 3.4 **Оценка вероятности локализации глобального экстремума $\tilde{\xi}$ функции Растригина в функции размера популяции $|S|$**

$\lvert S$	6	10	14	20	30	40
$\tilde{\xi}$	0,07	0,60	0,83	0,90	0,97	1,00

Табл. 3.4 показывает, что вероятность локализации глобального экстремума сложной многоэкстремальной функции Растригина достаточно высокой размерности ($|X| = 8$) быстро растет с ростом размера популяции и уже при общем числе пчел в популяции, равном 30, достигает величины 0,97●

Известно значительное число модификаций *ABC*-алгоритма, имеющих целью повышение его эффективности. Например, в работе Ал. А. Олейника, С. А. Субботина (2008 г.) предложена модификация, основные положения которой заключаются в следующем.

Если общее число пчел в данном источнике X_i достигло своего порогового значения $n_{\max}$, то одна из них (пусть это будет рабочая пчела s_i^w) изменяет свои координаты по правилу, близкому к правилу (3.26),

$$x'_{i,j} = \begin{cases} x_{i,j}^w + u_{sign}^{\pm 1}\, b_x, & j = l, \\ x_{i,j}^w, & j \neq l, \end{cases} \quad i \in [1:|S^w|],\ j \in [1:|X|],$$

где $l = U_1(1:|X|)$ – случайная натуральная величина, определяющая координатную ось, вдоль которой перемещается пчела s_j^w; $u_{sign}^{\pm 1}$ – случайная дискретная величина, которая с равной вероятностью принимает значения ± 1 и определяет направление смещения пчелы s_i^w; b_x – свободный параметр, шаг смещения.

Полезность φ_i^w, $i \in [1:|S^w|]$ источника нектара X_i^w определяет формула

$$\varphi_i^w = \begin{cases} 1, & \tilde{\varphi}_i^w > 1, \\ \tilde{\varphi}_i^w, & \tilde{\varphi}_i^w \in [\delta_\varphi; 1], \\ 0, & \tilde{\varphi}_i^w < \delta_\varphi, \end{cases} \tag{3.28}$$

где

$$\tilde{\varphi}_i^w = 1 - \overline{\varphi}_i^w + U_1(-b_n; b_n). \tag{3.29}$$

Здесь $\delta_\varphi > 0$ – порог полезности источника (рекомендованное значение равно 0,1); $b_n > 0$ – уровень случайных возмущений (рекомендованное значение равно 0,1); $\overline{\varphi}_i^w$ – нормированная полезность источника нектара X_i^w, рассчитанная на основе его минимальной и максимальной полезностей на протяжении всех итераций $[1:\ t]$.

На текущей итерации вероятность ξ_i^w выполнения пчелой s_i^w процедуры вербовки других пчел оцениваем в соответствии с выражением

$$\xi_i^w = \frac{a_i}{b_\xi},\ i \in [1:|S^w|],$$

где $a_i = \max\left(\varphi_i^w - b_a \overline{\varphi}^w,\ 0\right)$ – привлекательность источника X_i^w, $b_\xi > 0$ – параметр, управляющий вероятностью ξ_i^w; $\overline{\varphi}^w$ – средняя текущая полезность всех $|S^w|$ участков; $b_\alpha > 0$ свободный коэффициент, регулирующий влияние величины $\overline{\varphi}^w$.

Вероятность того, что незанятый фуражир станет разведчиком, определяет формула

$$\xi^o = \exp\left(-\frac{a^2}{2b_o}\right),$$

где $a = \sum_{i \in [1:|S^w|]} a_i$ – суммарная текущая привлекательность всех источников, $b_o > 0$ – свободный параметр.

Авторами рассматриваемой модификации выполнено сравнение ее эффективности с каноническим алгоритмом роя частиц. Исследование показало, что при оптимизации сложных многомерных многоэкстремальных функций данная модификация обеспечивает более высокую точность локализации глобального экстремума.

3.3.4. Гибридный алгоритм

Алгоритм предложен в работах С. А. Субботина, Ан. А. Олейника и Ал. А. Олейника (2008 г.) и существенно использует алгоритм оптимизации путем имитации отжига.

Рассмотрим задачу глобальной условной максимизации в гиперпараллелепипеде *П*. Схема алгоритма включает в себя следующие основные шаги.

1) Инициализация алгоритма.

2) Выбор пчел, на основе которых с помощью процедуры скрещивания будут получены новые пчелы.

3) Скрещивание отобранных агентов.

4) Моделирование процесса вербовки пчел.

5) Локальная оптимизация найденных лучших решений с помощью алгоритма имитации отжигом.

6) Выбор лучшей пчелы.

7) Выбор пчел-разведчиков.

8) Обновление динамических параметров алгоритма.

9) Завершение итераций.

Последовательно рассмотрим перечисленные шаги алгоритма.

Инициализация алгоритма. Этап инициализации включает в себя следующие действия. Задаем начальные координаты $X_i(0)$ пчел-разведчиков (источников нектара)

$$x_{i,j}(0) = x_j^- + U_1(0;1)\ \left(x_j^+ - x_j^-\right),\ i \in [1:|S^o|],\ j \in [1:|X|]. \qquad (3.30)$$