Глава 37. МЕТОД, ИМИТИРУЮЩИЙ ПОВЕДЕНИЕ СТАИ СЕРЫХ ВОЛКОВ

37.1. Постановка задачи

Дана целевая функция $f(x) = f(x_1, x_2, ..., x_n)$, определённая на множестве допустимых решений $D \subseteq R^n$.

Требуется найти условный глобальный максимум функции $f(x)$ на множестве D, т.е. такую точку $x^* \in D$, что

$$f(x^*) = \max_{x \in D} f(x), \tag{37.1}$$

где $x = (x_1, x_2, ..., x_n)^T$, $D = \{x \mid x_i \in [a_i, b_i], i = 1,2,...,n\}$.

Задача поиска минимума функции $f(x)$ сводится к задаче поиска максимума путем замены знака перед функцией на противоположный: $f(x^*) = \max_{x \in D} f(x) = -\min_{x \in D}[-f(x)]$. Функция $f(x)$ может быть многоэкстремальной, поэтому искомое решение в общем случае неединственное.

37.2. Стратегия поиска решения

Метод серых волков (Grey Wolf Optimizer – GWO) имитирует охоту стаи серых волков за жертвой [x]. Он относится к методам роевого интеллекта, в которых используется иерархия лидерства в стае и особый механизм охоты, заключающийся в отслеживании и приближении к жертве, ее последующем окружении и финальном нападении.

В начале работы метода случайным образом, используя предположение о равномерном распределении, на множестве допустимых решений D генерируется некоторый набор начальных точек (волков в стае): $I = \{x^j = (x_1^j, x_2^j, ..., x_n^j)^T, j = 1,..., NP\} \subset D$, где x^j – вектор координат волка с номером j, NP – количество волков в стае. Поскольку в процессе охоты положение жертвы точно не известно вследствие ее постоянного движения (а в задаче оптимизации не известно положение точки экстремума), то члены стаи ориентируются на лидеров, полагая, что они обладают большей информацией о положении жертвы (точке экстремума).

В стае волков, где каждый волк характеризуется своей позицией в области допустимых решений, выбираются три последовательно лучших (α, β, γ) по величине целевой функции $f(x)$: $x^\alpha, x^\beta, x^\gamma$. Все волки в стае меняют свое положение с учетом сравнения своей текущей позиции с этими тремя наилучшими:

$$x^j(k+1) = \frac{x^{j,1}(k+1) + x^{j,2}(k+1) + x^{j,3}(k+1)}{3},$$

$$x^{j,1}(k+1) = x^\alpha(k) - A_\alpha \otimes D_\alpha^j(k),$$
$$x^{j,2}(k+1) = x^\beta(k) - A_\beta \otimes D_\beta^j(k),$$
$$x^{j,3}(k+1) = x^\gamma(k) - A_\gamma \otimes D_\gamma^j(k),$$

$$D_\alpha^j(k) = \left| C_\alpha^j \otimes x^\alpha(k) - x^j(k) \right|,$$
$$D_\beta^j(k) = \left| C_\beta^j \otimes x^\beta(k) - x^j(k) \right|,$$
$$D_\gamma^j(k) = \left| C_\gamma^j \otimes x^\gamma(k) - x^j(k) \right|,$$

где $\otimes$ – операция поэлементного произведения векторов по Адамару, k – номер итерации, $x^j(k+1), x^j(k)$ – следующее и текущее положения волков, $j = 1,...,NP$; $A_\alpha, A_\beta, A_\gamma$ – векторы, определяемые по правилу $A_m = 2a \otimes r_1 - a,\ m = \alpha, \beta, \lambda$; r_1 – n-мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке $[0,1]$; a – вектор с одинаковыми компонентами, уменьшающимися линейно по закону $a_i = 2(1 - \frac{k}{K}),\ i = 1,...,n;$ K – максимальное число итераций; $C_\alpha, C_\beta, C_\gamma$ – векторы, определяемые по правилу $C_m = 2r_2, m = \alpha, \beta, \lambda$, r_2 – n-мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке $[0,1]$. Имеется модификация, в которой $a_i = 2(1 - \frac{k^2}{K^2}),\ i = 1,...,n$ [x].

Общая схема работы метода серых волков представлена на рис. 37.1.

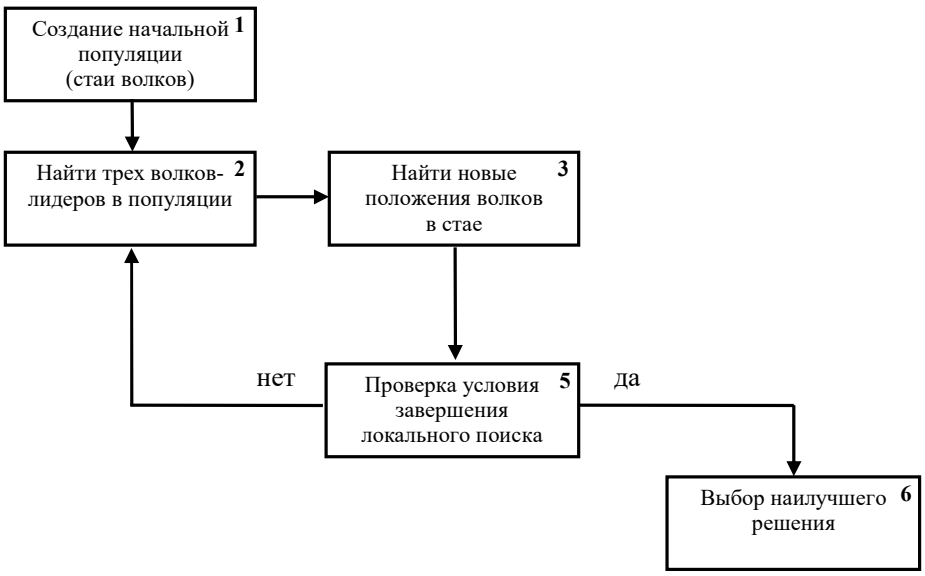


Рис. 37.1. Общая схема работы метода, имитирующего поведение стаи серых волков

37.3. Алгоритм решения задачи

Шаг 1. *Генерация начальной популяции.*
Шаг 1.1. Задать параметры метода:

- число элементов в популяции NP;

- максимальное число итераций K;

Положить $k = 1$ (счетчик числа итераций).

Шаг 1.2. Используя равномерный закон распределения на множестве D, сгенерировать начальную популяцию

$$I_k = \{x^j(k) = (x_1^j(k), x_2^j(k), ..., x_n^j(k))^T, j = 1, ..., NP\} \subset D.$$

Шаг 1.3. Для каждого волка в стае вычислить соответствующее значение целевой функции: $f\left(x^1(k)\right), \ldots, f\left(x^{NP}(k)\right)$.

Шаг 1.4. Среди сгенерированных частиц найти три наилучших решения, которым соответствуют наименьшие значения целевой функции:

$x^\alpha = \arg\min_{j \in \{1,...,NP\}} f\left(x^j(k)\right)$ – лучшее решение; x^β – второе по величине функции;

x^γ – третье по величине функции.

Шаг 2. *Вычисление параметров.*

Для каждого волка в стае с номером j найти:

а) a – вектор с одинаковыми компонентами $a_i = 2(1 - \frac{k}{K}),\ i = 1, ..., n,$ или $a_i = 2(1 - \frac{k^2}{K^2}),\ i = 1, ..., n,$ в зависимости от используемой модификации;

б) $A_\alpha, A_\beta, A_\gamma$ – векторы, определяемые по правилу

$$A_m = 2a \otimes r_1 - a,\ \ m = \alpha, \beta, \lambda;$$

где r_1 – n-мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке $[0,1]$; $\otimes$ – операция поэлементного произведения векторов по Адамару;

в) $C_\alpha, C_\beta, C_\gamma$ – векторы, определяемые по правилу

$$C_m = 2r_2,\ \ m = \alpha, \beta, \lambda,$$

где r_2 – n-мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке $[0,1]$.

Шаг 3. *Генерация новой стаи.*

Шаг 3.1. Найти новые положения волков в стае

$$D_\alpha(k) = \left| C_\alpha \otimes x^\alpha(k) - x^j(k) \right|,$$
$$D_\beta(k) = \left| C_\beta \otimes x^\beta(k) - x^j(k) \right|,$$
$$D_\gamma(k) = \left| C_\gamma \otimes x^\gamma(k) - x^j(k) \right|,$$

$$x^{j,1}(k+1) = x^\alpha(k) - A_\alpha \otimes D_\alpha(k),$$
$$x^{j,2}(k+1) = x^\beta(k) - A_\beta \otimes D_\beta(k),$$
$$x^{j,3}(k+1) = x^\gamma(k) - A_\gamma \otimes D_\gamma(k),$$

$$x^j(k+1)=\frac{x^{j,1}(k+1)+x^{j,2}(k+1)+x^{j,3}(k+1)}{3},\quad j=1,...,NP.$$

Шаг 3.2. Для каждой волка в стае вычислить соответствующее ему значение целевой функции: $f\left(x^1(k+1)\right),\ldots,f\left(x^{NP}(k+1)\right)$.

Шаг 3.3. Найти новые три наилучших решения, которым соответствуют наименьшие значения целевой функции:

$$x^\alpha=\arg\min_{j\in\{1,...,NP\}} f\left(x^j(k+1)\right),$$

x^β – второе по величине функции;

x^γ – третье по величине функции.

Шаг 4. *Проверка условия завершения поиска.*

Если $k=K,$ то процесс поиска завершить, перейти к шагу 5, а иначе положить $k=k+1$ и перейти к шагу 2.

Шаг 5. *Выбор решения из последней популяции.*

Закончить работу алгоритма. В качестве решения (приближенного) задачи $f(x^*)=\min_{x\in D} f(x)$ выбрать волка с положением x^α, которому соответствует наименьшее значение целевой функции.

З а м е ч а н и е.

1. Компоненты векторов A_m, где $m=\alpha,\beta,\lambda,$ являются случайными на отрезке $[-a_i,a_i]$, при этом a_i уменьшается от 2 до нуля. При $|A_{mi}|\geq 1$ волк может удаляться от жертвы, осуществляя исследование множества допустимых решений, а при $|A_{mi}|<1$, $i=1,...,n$, волк приближается к жертве, нападая на нее.

2. Если новое положение волка на шаге 3.1 не принадлежит множеству допустимых решений, следует генерировать параметры метода заново до тех пор, пока ограничения не будут выполнены. Второй способ – если какая-то компонента вышла за границы отрезка $[a_i,b_i]$, то в качестве нового значения выбрать соответствующую границу.

3. На шаге 3.1 можно искать абсолютные величины каждой компоненты правых частей или вычислять эвклидово расстояние и полагать все компоненты равными.

Рекомендуемые значения параметров: $step_{init}=0,05;$ $\rho=0,3;$ $decay_{\min}=0,999;$ $decay_{\max_{ini}}=0,99;$ $decay_{\max_{end}}=0,95.$

25.4. Программное обеспечение

На основе изложенного алгоритма разработана программа поиска глобального экстремума функций. Для ее создания использовалась среда разработки Microsoft Visual Studio, язык программирования C#.

Возможности программы позволяют изучить алгоритм метода, а также влияние параметров метода на результат его работы. В список выбираемых функций включены стандартные многоэкстремальные тестовые функции двух переменных, для которых известно точное решение (табл. П.1).

На рис. 25.2 представлено главное окно метода, имитирующего поведение стаи рыб, где пользователь может выбрать вид целевой функции, задать множество допустимых решений и параметры метода, просматривать результаты работы.

Разработанная программа предусматривает возможность анализа работы метода по шагам. На рис. 25.3 представлено окно пошаговой работы метода, где изображена общая схема метода, отображаются результаты работы после выполнения каждого шага (график изменения наилучшего значения целевой функции, а также графическое изображение популяции) и после завершения работы метода.

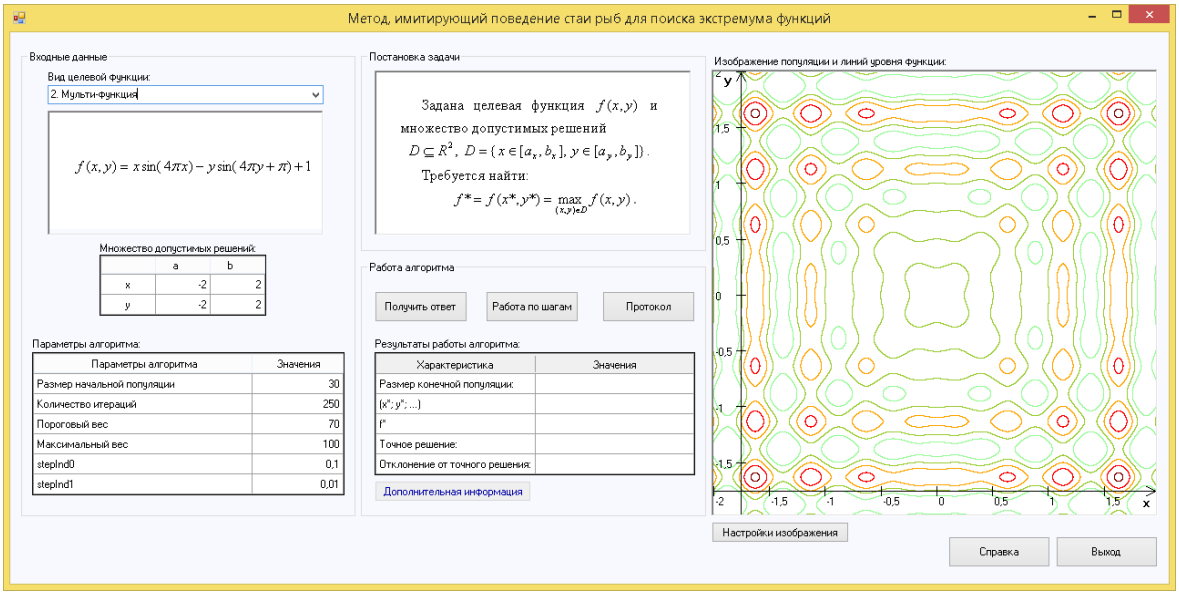


Рис. 25.2. Главное окно программы метода, имитирующего поведение стаи рыб

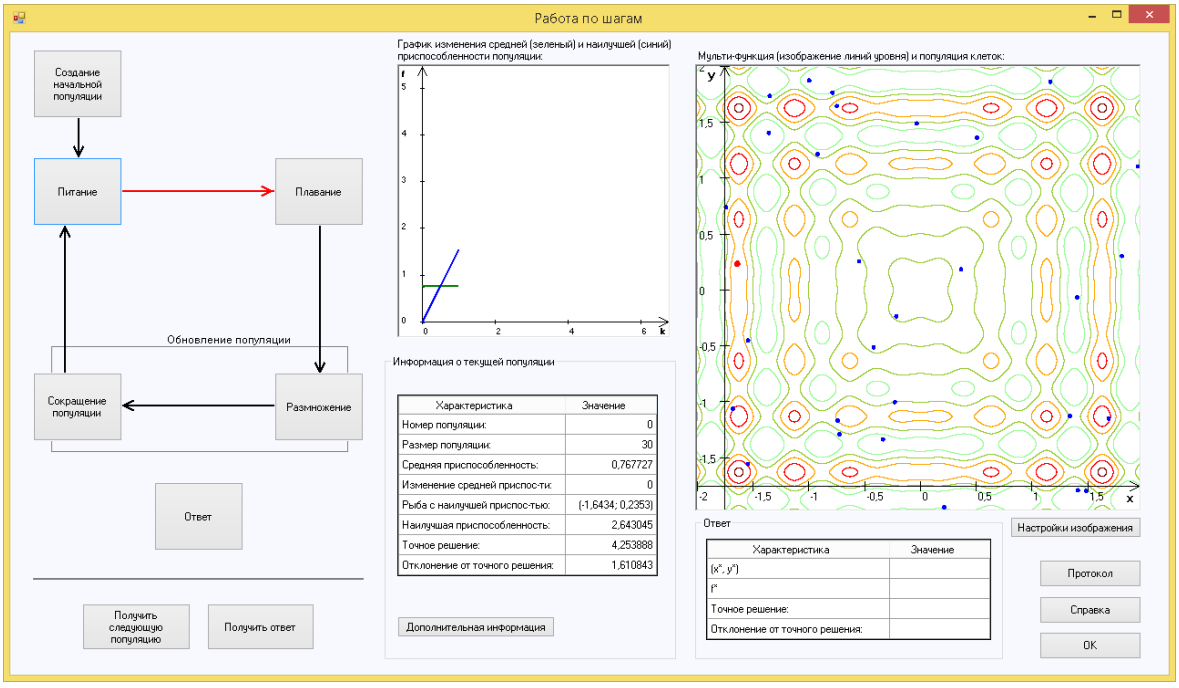


Рис. 25.3. Окно пошаговой работы метода, имитирующего поведение стаи рыб

25.5. Тестовые примеры

Пример 25.1. Найдем глобальный максимум корневой функции (табл. П.1). Зададим множество допустимых решений $x, y \in [-2;2]$. Выберем следующие параметры алгоритма:

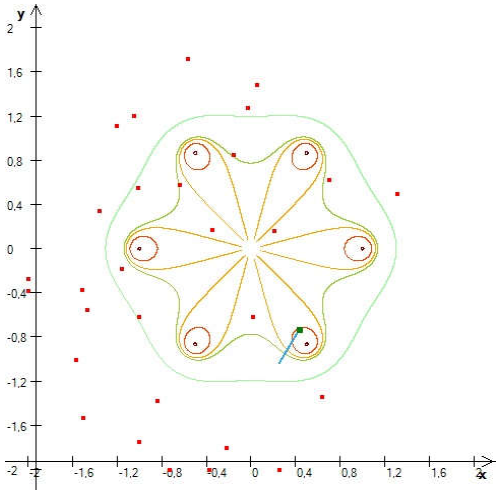
- число элементов в популяции $NP = 30$;
- максимальное число итераций $K_{\max} = 200$;
- угол поворота $\theta = \frac{\pi}{4}$;
- минимальное значение радиуса спирали $r_{\min} = 0{,}81$;
- максимальное значение радиуса спирали $r_{\max} = 0{,}91$;
- параметры для определения радиуса спирали $r_{\max} = 0{,}91$;

На рис. 24.5 представлена популяция на начальной ($k = 1$), промежуточных ($k = 10,\ k = 25$) и конечной ($k = 200$) итерациях. Зеленым цветом обозначено наилучшее решение в популяции на текущей итерации.

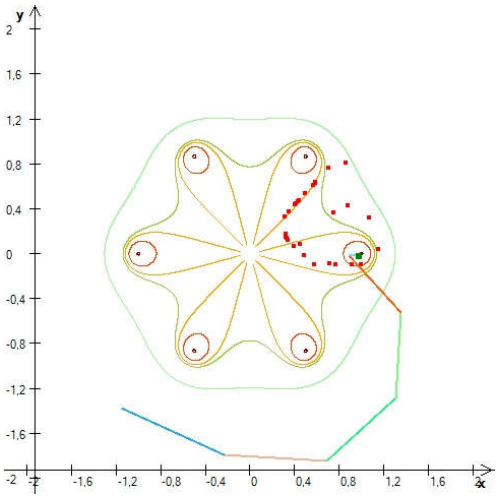
Результаты работы алгоритма:

- наилучшее решение $(x^*; y^*) = (1; 0)$;
- значение целевой функции $f(x^*, y^*) = 1$;
- отклонение от точного решения $\Delta = 0$.

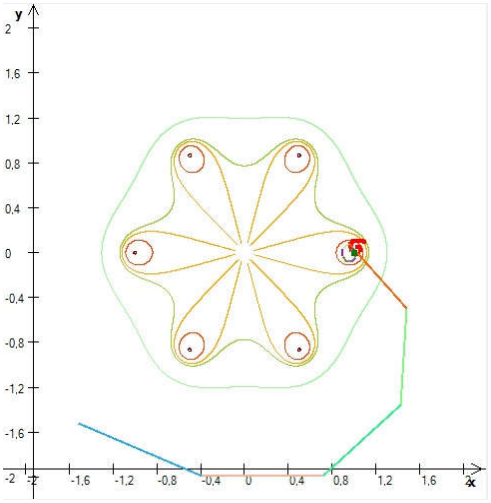
График изменения наилучшего значения целевой функции представлен на рис. 24.6.



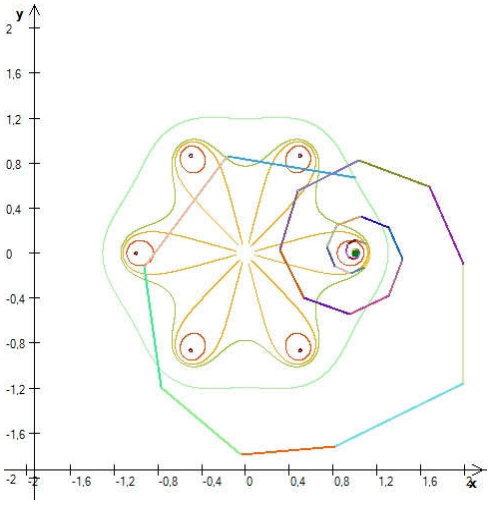
$k = 1$



$k = 10$



$k = 25$



$k = 200$

Рис. 24.5. Начальная, промежуточные и конечная итерации

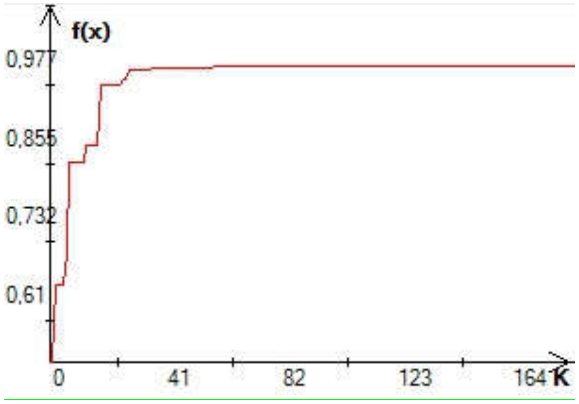


Рис. 24.6. График изменения наилучшего значения целевой функции

Пример 25.2. Найдем глобальный максимум функции Розенброка (табл. П.1). Зададим множество допустимых решений $x \in [-3;3]$, $y \in [-1;5]$. Выберем следующие параметры алгоритма:

- число элементов в популяции $NP = 50$;
- максимальное число итераций $K_{\max} = 200$;
- угол поворота $\theta = \frac{\pi}{4}$;
- минимальное значение радиуса спирали $r_{\min} = 0{,}81$;
- максимальное значение радиуса спирали $r_{\max} = 0{,}91$;
- константы для определения радиуса спирали $c_1 = 0{,}81; c_2 = 0{,}91$.

На рис. 24.7 представлена популяция на начальной ($k = 1$), промежуточных ($k = 5, k = 15$) и конечной ($k = 200$) итерациях. Зеленым цветом обозначено наилучшее решение в популяции на данной итерации.

Результаты работы алгоритма:

- наилучшее решение $(x^*; y^*) = (0,963; 0,926)$;
- значение целевой функции $f(x^*, y^*) = -0,0014$;
- отклонение от точного решения $\Delta = 0,0014$.

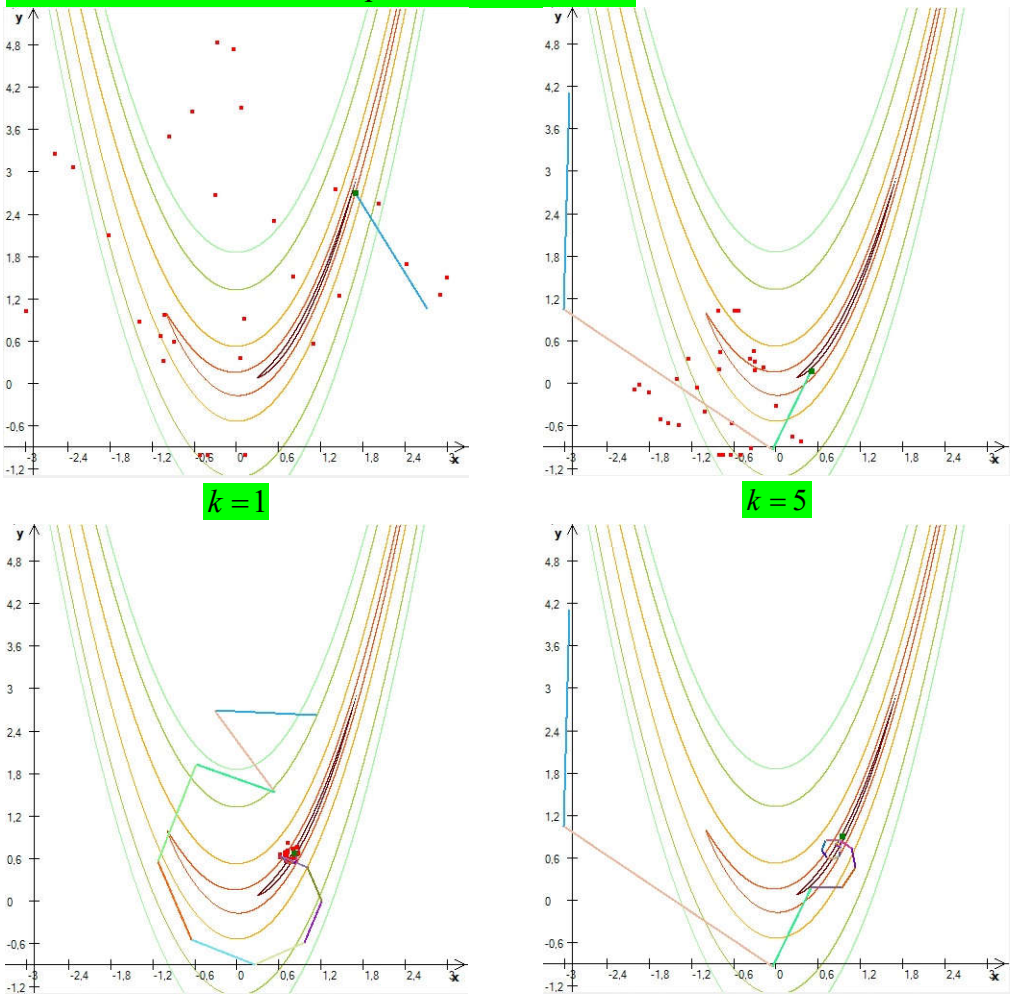


Рис. 24.7. Начальная, промежуточные и конечная популяции

График изменения наилучшего значения целевой функции представлен на рис. 24.8.

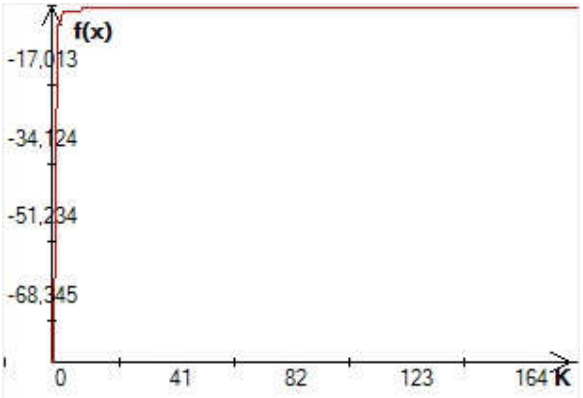


Рис. 24.8. График изменения наилучшего значения целевой функции

25.6. Анализ эффективности метода

25.6.1. Рекомендации по выбору параметров

Размер популяции NP определяет количество вычислений целевой функции на каждой итерации. Для задачи с большой областью допустимых решений рекомендуется брать большее значение параметра NP. Рекомендуемые значения параметра $NP \in [30;40]$.

Число итераций $ITER$ определяет, как долго будет продолжаться поиск новых решений. Чем больше $ITER$, тем более точным будет решение. Для рассмотренного набора стандартных функций рекомендуемые значения в зависимости от сложности функции $ITER \in [1000;10000]$.

Максимальный вес W_{scale}, который может набрать особь. По истечению времени поиска, выбирается рыба с максимальным весом W_{scale}, и в качестве ответа берется ее положение. Рекомендованное значение этого параметра $W_{scale} \in [1000;5000]$.

Пороговый вес thr определяет тех рыб, которые будут допущены к размножению. Рекомендованное значение параметра $thr \in [900;4500]$.

Шаги индивидуального плавания $step_{ind}^{k}$ и коллективного плавания $step_{vol}^{k}$.

Рекомендуемые варианты начального и конечного значений шага:

а) $step_{ind,initial} = 0,001 \cdot \max_i [b_i - a_i]$, $step_{ind,final} = 0,00001 \cdot \max_i [b_i - a_i]$,

$step_{vol,initial} = 0,01 \cdot \max_i [b_i - a_i]$, $step_{vol,final} = 0,0001 \cdot \max_i [b_i - a_i]$,

б) $step_{ind,initial} = 0,1 \cdot \max_i [b_i - a_i]$, $step_{ind,final} = 0,0001 \cdot \max_i [b_i - a_i]$,

$step_{vol,initial} = 0,1 \cdot \max_i [b_i - a_i]$, $step_{vol,final} = 0,001 \cdot \max_i [b_i - a_i]$,

в) $step_{ind,initial} = 0,01 \cdot \max_i [b_i - a_i]$, $step_{ind,final} = 0,00001 \cdot \max_i [b_i - a_i]$,

$step_{vol,initial} = 0,01 \cdot \max_i [b_i - a_i]$, $step_{vol,final} = 0,0001 \cdot \max_i [b_i - a_i]$,

г) $step_{ind,initial} = 0,1 \cdot \max_i [b_i - a_i]$, $step_{ind,final} = 0,0001 \cdot \max_i [b_i - a_i]$,

$step_{vol,initial} = 0,1 \cdot \max_i [b_i - a_i]$, $step_{vol,final} = 0,001 \cdot \max_i [b_i - a_i]$.

25.6.2. Анализ работы метода при различных значениях параметров

В данном разделе приводится статистический анализ и сравнение работы метода, имитирующего поведение стаи рыб, при различных значениях его параметров. Исследуемый метод применялся к некоторым функциям из набора тестовых функций (табл. П.1). Для каждой функции проводились серии из 100 решений одной и той же задачи с одними и теми же значениями параметров. Для полученной выборки

$\{f^1, f^2, ..., f^{100}\}$ вычислялись среднее значение отклонения полученного решения от точного $\overline{\Delta f} = \frac{1}{100}\sum_{i=1}^{100}\Delta f_i$, где $\Delta f_i = \left|f(x^*) - f^i\right|$; наименьшее значение отклонения $\Delta f_{best} = \min_i \Delta f_i$; среднеквадратическое отклонение $\overline{\sigma}_f = \sqrt{\overline{S}_{100}}$, где $\overline{S}_{100} = \frac{1}{100}\sum_{i=1}^{100}(\Delta f^i - \overline{\Delta f})^2$; количество успехов n_{ycn} (успехом считалось попадание лучшей точки в ε-окрестность точного решения, $\varepsilon = \frac{\max_{i=1,..,n}\left|b_i - a_i\right|}{1000}$). Результаты, полученные для каждой функции, представлены в табл. 25.1–25.3.

Таблица 25.1. Влияние параметров метода. Корневая функция

Параметры метода						$\overline{\Delta f}$	Δf_{best}	$\overline{\sigma}_f$	n_{succ}
NP	*ITER*	*W*	W_{scale}	$step_{ind0}$	$step_{ind1}$				
30	10000	4500	5000	0,1	0,01	0,001215	0,000016	0,00069	100
20	10000	4500	5000	0,1	0,01	0,001055	0,000012	0,000572	100
10	5000	4500	5000	0,1	0,01	0,012728	0,000056	0,094832	99
20	5000	4000	5000	0,1	0,01	0,004717	0,000008	0,023464	99
20	5000	4000	4500	0,1	0,01	0,002062	0,000009	0,001096	100
20	5000	200	250	0,1	0,01	0,001699	0,000078	0,000932	100
20	5000	4500	5000	0,08	0,01	0,001419	0,000096	0,000703	100
20	10000	4500	5000	0,3	0,01	0,002643	0,000089	0,001545	100
20	10000	4500	5000	0,1	0,008	0,001632	0,000117	0,00082	100
20	10000	4500	5000	0,1	0,03	0,002507	0,000009	0,001283	100

Таблица 25.2. Влияние параметров метода. Мульти-функция

Параметры метода						$\overline{\Delta f}$	Δf_{best}	$\overline{\sigma}_f$	n_{succ}
NP	*ITER*	*W*	W_{scale}	$step_{ind0}$	$step_{ind1}$				
30	10000	4500	5000	0,1	0,01	0,789304	0	0,744677	12
50	10000	4500	5000	0,1	0,01	0,750076	0	0,798666	32
50	10000	4000	5000	0,1	0,01	0,940749	0	0,800207	22
50	10000	4700	5000	0,1	0,01	0,912874	0	0,650055	14

50	10000	4500	4800	0,1	0,01	0,797011	0	0,720034	26
50	10000	4500	5200	0,1	0,01	0,965895	0	0,739039	19
50	10000	4500	5000	0,08	0,01	0,539011	0	0,640241	34
50	10000	4500	5000	0,2	0,01	0,836956	0	0,656606	19
50	10000	4500	5000	0,1	0,009	0,882147	0	0,763311	19
50	10000	4500	5000	0,1	0,02	0,875453	0,000001	0,661915	19

Таблица 25.3. Влияние параметров метода. Функция Розенброка

Параметры метода						$\overline{\Delta f}$	Δf_{best}	$\overline{\sigma}_f$	n_{succ}
NP	*ITER*	*W*	W_{scale}	$step_{ind\,0}$	$step_{ind\,1}$				
30	10000	4500	5000	0,1	0,01	0	0	0	100
40	10000	4500	5000	0,1	0,01	0	0	0	100
40	10000	4000	5000	0,1	0,01	0	0	0	100
40	10000	4800	5000	0,1	0,01	0	0	0	100
40	10000	4500	4800	0,1	0,01	0	0	0	100
40	10000	4500	5200	0,1	0,01	0,0069	0	0,068654	99
40	10000	4500	5000	0,2	0,01	0	0	0	100
40	10000	4500	5000	0,09	0,01	18,010518	0	179,202387	99
40	10000	4500	5000	0,1	0,02	0,087777	0	0,785772	98
40	10000	4500	5000	0,1	0,009	0	0	0	100

Анализ работы метода спиральной динамики показал, что с большинством тестовых функций метод справляется успешно. В серии из 100 запусков удается найти глобальный минимум функций с большой точностью или точное решение.

При выборе функций с большой областью допустимых решений, таких как синусоидальная функция Швефеля, близкий к точному решению результат получается только при большом размере популяции *NP*.

Метод спиральной динамики не всегда может найти изолированный глобальный минимум функции Шаффера, сходясь к одному из локальных минимумов.

При работе с функцией Розенброка метод находит область, в которой находится глобальный минимум, но не всегда успевает сойтись к точке глобального минимума.

Mirjalili S., *Mirjalili S.M., Lewis A.* Grey wolf optimizer //Advances in Engineering Software. 2014. Vol. 69. P. 46–61.

Mittal N., *Singh U., Sohi B.S.* Modified grey wolf optimizer for global engineering optimization // Applied Computational Intelligence and Soft Computing. 2016. Article ID 7950348. http://dx.doi.org/10.1155/2016/7950348