

8. ОПТИМАЛЬНОЕ ПРОЕКТИРОВАНИЕ НА ОСНОВЕ РЕШЕНИЯ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

8.1. Основные понятия линейного программирования

Задачи, в которых отыскивается минимум или максимум некоторой функции, зависящей от многих переменных при наличии ограничений на эти переменные называется задачами *математического программирования*.

Функция F , максимум или минимум которой определяется называется *целевой функцией*.

Если целевая функция и ограничения *линейно* зависят от переменных, то такие задачи называются задачами *линейного программирования*

Таким образом, математическая модель любой задачи линейного программирования включает в себя:

- максимум или минимум целевой функции (критерий оптимальности);
- систему ограничений в форме линейных уравнений и неравенств;
- требование неотрицательности переменных

Решения, удовлетворяющие системе ограничений (1) условий задачи и требования неотрицательности переменных (2) называются *допустимыми*

Решения, удовлетворяющие одновременно и требованиям минимизации (максимализации) (3) целевой функции называются *оптимальными*

Выражения (1), (2), (3) – это общий вид задачи линейного программирования (ЗЛП), которую часто называют *основной задачей*

Если система ограничений (1) состоит только из равенств (уравнений), а переменные v_i являются неотрицательными, то такая ЗЛП называется *канонической*. К канонической форме можно преобразовать любую ЗЛП.

Если система ограничений состоит из одних линейных неравенств типа « \leq » или « \geq », то такая форма ЗЛП называется *стандартной*. Любую ЗЛП можно привести к стандартной форме.

Правило приведения ЗЛП к каноническому виду

1. Если в исходной задаче некоторое ограничение (например, первое) было неравенством, то оно преобразуется в равенство, введением в левую часть некоторой неотрицательной переменной, при чем в неравенство « \leq » эта дополнительная переменная вводится со знаком плюс, а в случае неравенства « \geq » со знаком минус

$$a_{11}v_1 + a_{12}v_2 + \dots + a_{1n}v_n \leq b_1 \quad (4)$$

Вводим переменную

$$v_{n+1} = b_1 - a_{11}v_1 - a_{12}v_2 - \dots - a_{1n}v_n.$$

Тогда неравенство (4) запишется в виде

$$a_{11}v_1 + a_{12}v_2 + \dots + a_{1n}v_n + v_{n+1} = b_1 \quad (5)$$

В каждое из неравенств вводится своя «уравнивающая» переменная, после чего система ограничений становится системой уравнений

2. Если в исходной задаче некоторая переменная не подчинена условию неотрицательности, то ее заменяют (в целевой функции и во всех ограничениях) разностью неотрицательных переменных $v_k = v_k - v_l$, где $v_k \geq 0, v_l \geq 0$.

3. Если в ограничениях правая часть отрицательная, то следует умножить это ограничение на (-1)

4. Наконец, если исходная задача была задачей на максимум, то введением новой целевой функции $F_1 = -F$ мы преобразуем нашу задачу на максимум функции F в задачу на минимум функции F_1 .

Пример 1

Привести к каноническому виду задачу

$$\begin{cases} v_1 - v_2 \geq 1 \\ v_1 - 2v_2 \leq 1 \\ v_1 + v_2 \leq 3 \end{cases}$$
$$v_1 \geq 0, v_2 \geq 0, \quad F = v_1 + 2v_2 \rightarrow \max$$

Вводим дополнительные переменные v_3, v_4, v_5 . При этом в первое неравенство введем неотрицательную переменную v_3 со знаком минус, а во второе и третье – со знаком плюс переменные v_4 и v_5 . Запишем задачу в виде

$$\begin{cases} v_1 - v_2 - v_3 = 1 \\ v_1 - 2v_2 + v_4 = 1 \\ v_1 + v_2 + v_5 = 3 \end{cases}$$
$$v_j \geq 0, \quad j = 1, 2, \dots, 5$$

Переведем \max на \min , домножив целевую функцию на (-1)

$$F = -v_1 - 2v_2 + 0v_3 + 0v_4 + 0v_5 \rightarrow \min,$$

7.2. Симплекс-метод и основные утверждения линейного программирования

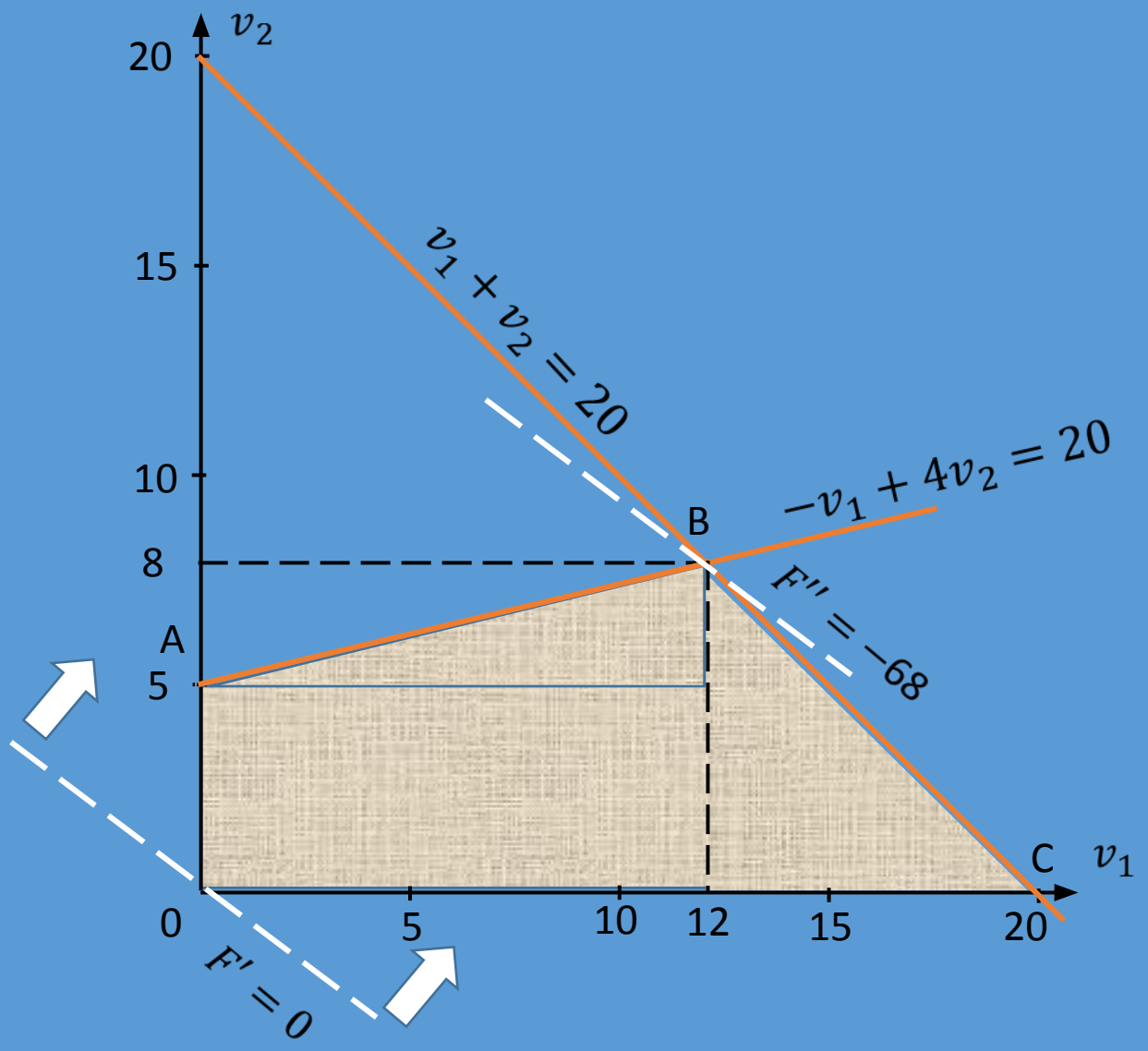
Геометрическая интерпретация ЗЛП и метода ее решения для двух переменных

Пример 2

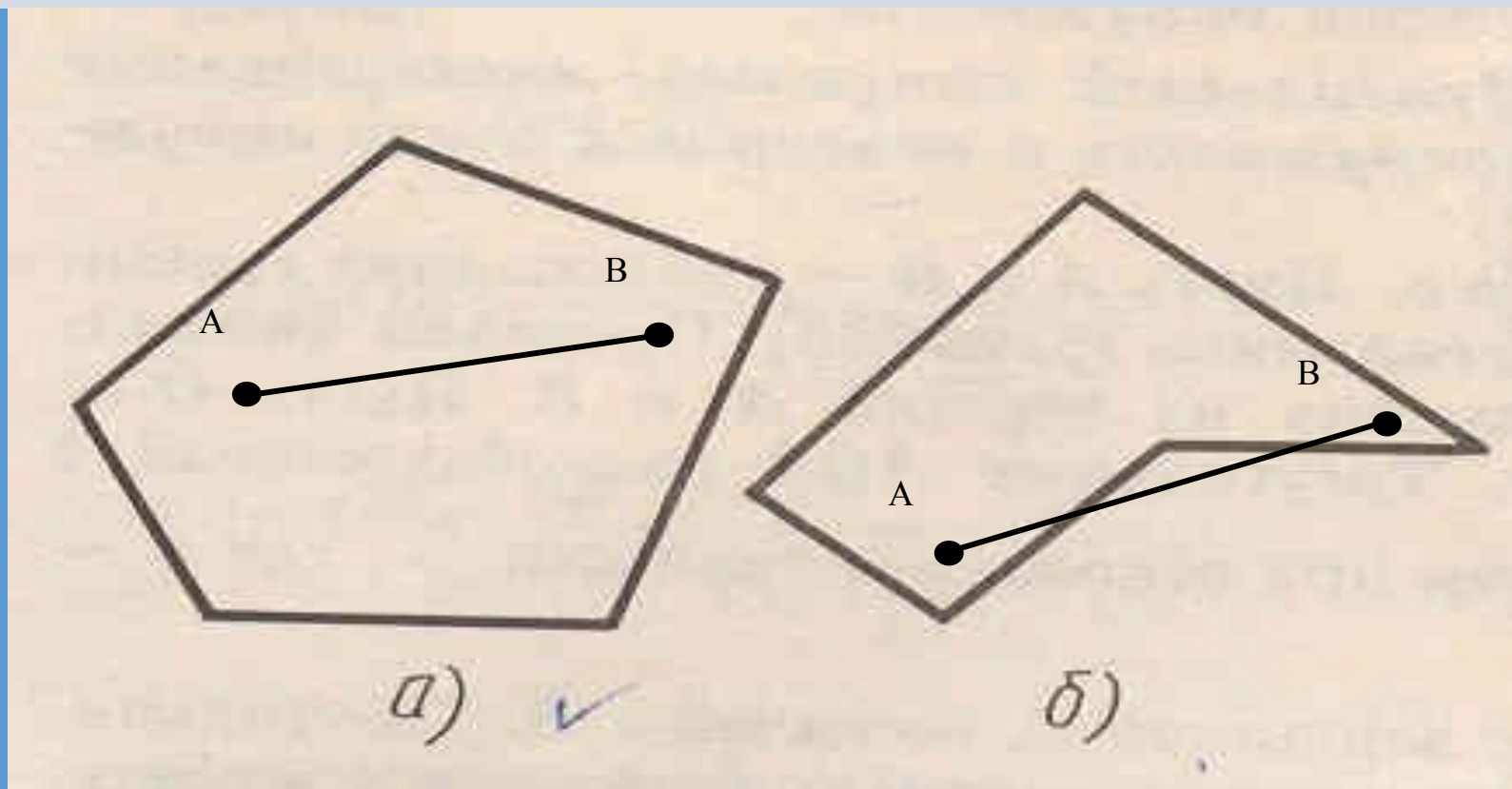
Требуется минимизировать функцию $F = -3v_1 - 4v_2 \rightarrow \min$

При наличии следующих ограничений

$$\begin{aligned} v_1 + v_2 &\leq 20 \\ -v_1 + 4v_2 &\leq 20 \\ v_1, v_2 &\geq 0, \end{aligned}$$



Множество точек называется выпуклым, если вместе с его двумя любыми точками ему принадлежит и весь отрезок, соединяющий их (см. рис. а – выпуклый многоугольник (множество), б - невыпуклый).



1. Множество всех допустимых решений системы ЗЛП является выпуклым с конечным числом угловых точек (вершин). В частном случае, когда в систему ограничений-неравенств входят только две переменные, это множество можно изобразить на плоскости (пр.2)
2. Симплекс-метод применим к любой ЗЛП в канонической форме
3. В канонической форме система ограничений – это система линейных уравнений, причем количество уравнений m меньше, чем число переменных n ($m < n$) (пр. 1). Можно выбрать m неизвестных, которые выражаются через остальные переменные. Эти m неизвестных называются *базисными*, остальные - *свободными*.
4. Придавая определенные значения свободным переменным и вычисляя базисные, можно получить различные решения системы ограничений. Решения, у которых свободные переменные равны нулю называются *базисными*. Базисное решение называется *допустимым базисным решением* или *опорным решением*, если в нем значение переменных (базисных) неотрицательны.

5. Опорные решения всегда соответствуют одной из вершин многоугольника ограничений (пр.2)

6. При n переменных каждое уравнение- ограничение представляет собой плоскость в n -мерном пространстве. Фигура, образованная этими плоскостями, образует область допустимых значений переменных и называется *симплексом*. Для рассмотренного примера симплекс представляет собой многоугольник OABCD

7. Вывод 5 можно распространить и на случай многомерной задачи, т.е. опорные решения ЗЛП соответствуют вершинам симплекса ограничений, в которых неотрицательны m базовых переменных и равны нулю остальные $n-m$ переменных

8. Оптимальное решение ЗЛП, если оно существует, является одним из опорных решений (пр. 2)

Симплексный метод

Геометрический смысл симплексного метода состоит в переходе от одной вершины многогранника ограничений к соседней, в которой ЦФ принимает лучшее (или по крайней мере, не худшее) значение до тех пор, пока не будет найдено оптимальное решение - вершина, где достигается оптимальное значение ЦФ.

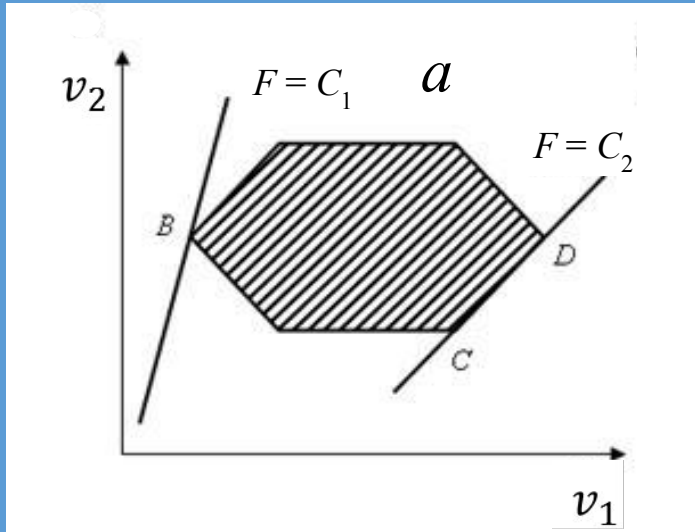
Процесс применения симплексного метода предполагает реализацию трех его основных элементов.

1. Способ определения какого-либо первоначального опорного решения.
2. Правило перехода к лучшему (точнее, нехудшему) опорному решению
3. Критерий проверки оптимальности найденного решения

Алгоритм решения ЗЛП графическим методом

1. Записывают уравнения прямых, соответствующих ограничениям типа (4), и строят их на плоскости v_1Ov_2 .
2. Определяют области, в которых выполняются ограничения задачи.
3. Определяют область допустимых решений задачи как область пересечения m полуплоскостей, соответствующих m ограничениям задачи.
4. Определяют направление возрастания (убывания) целевой функции F . Это можно сделать двумя способами. Самый простой способ - построить две линии уровня функции $F = C_1$; $F = C_2$; (C_1, C_2 - произвольные константы, $C_1 \neq C_2$), и по их расположению определить направление возрастания (убывания) функции.
5. Определяют граничную точку (точки) области допустимых решений, в которых целевая функция принимает максимальное или минимальное значение
6. Вычисляют значения найденной точки, решая совместно уравнения, задающие прямые, на пересечении которых находится эта точка, или выявляя уравнение граничной прямой области допустимых решений, с которой совпадает линия уровня целевой функции.

Возможны следующие варианты областей допустимых решений

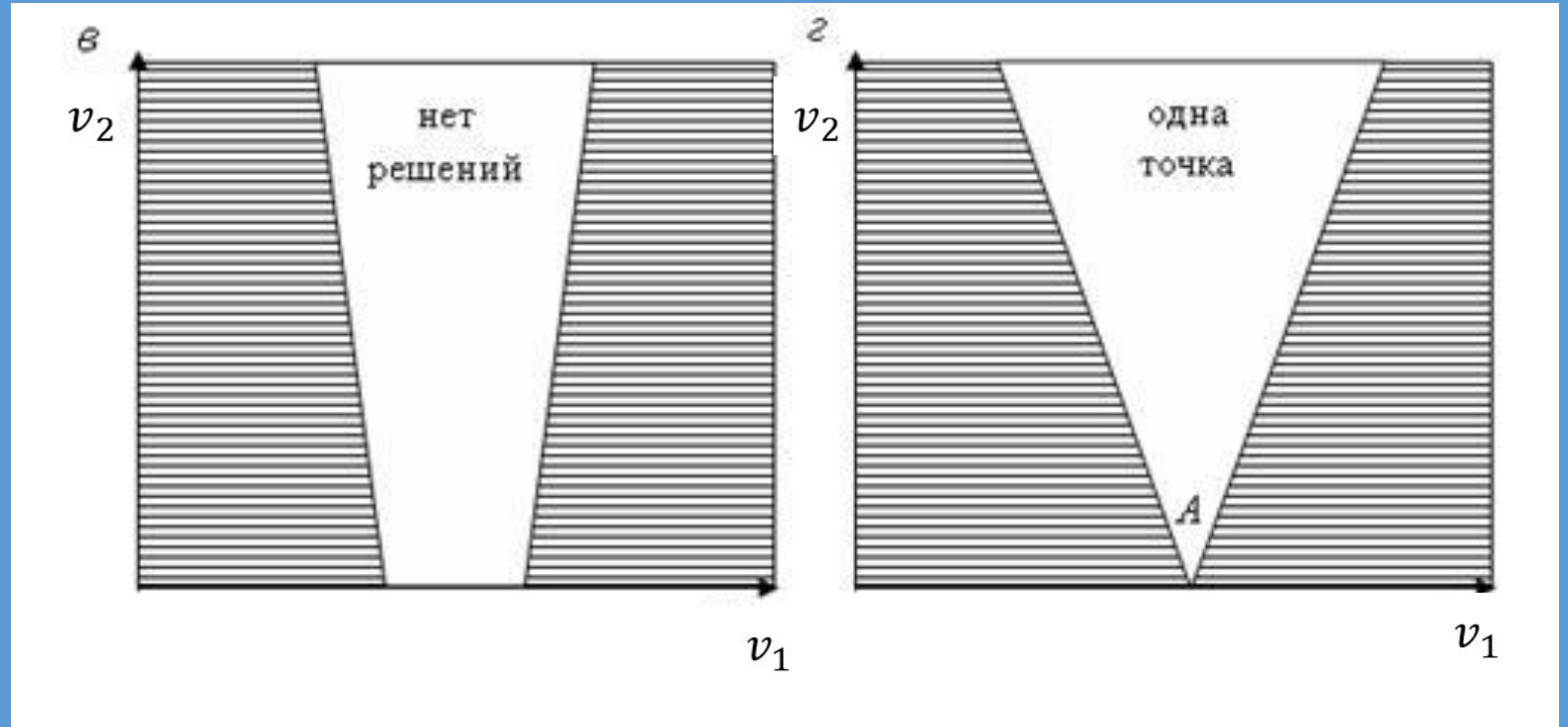


a - единственное решение – точка B ,

бесконечно много решений – отрезок CD ;

b – нет решений (область ограничений несовместимо);

z - только одна допустимая точка.

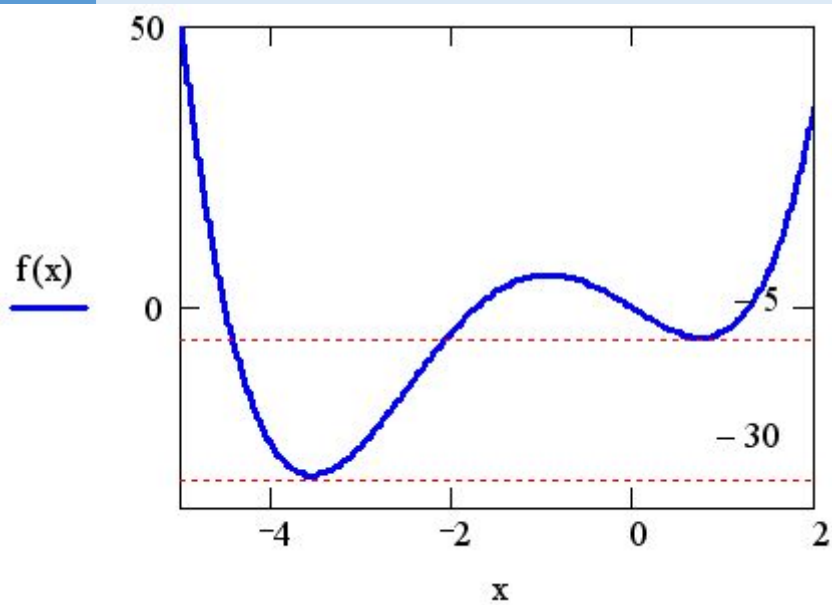


9. Оптимальное проектирование на основе решения задачи нелинейного программирования

9.1. Общие сведения

Нелинейное программирование (НП) – это математическое программирование, в котором ЦФ или ограничения являются нелинейными функциями.

Если функция имеет один минимум (максимум) в заданной области, то ее называют *одноэкстремальной* (униmodalной), если же более одного, то *многоэкстремальной*. Каждый минимум многоэкстремальной функции называется *локальным*, а наименьший из них – *глобальным*.



Если ограничения на внутренние параметры отсутствуют, то минимум называется *безусловным*, в противном случае функция имеет *условный минимум*.

Поиск минимума в n -мерном пространстве осуществляется итерационными методами. На каждой итерации необходимо решить две задачи: 1 – выбрать направление «движения» из заданной исходной или полученной на предыдущем шаге (итерации) точки; 2 – выполнить оптимальный шаг в данном направлении. Вторая задача – это одномерный поиск.

9.2. Методы одномерного поиска оптимального решения

Рассмотрим методы определения минимума ЦФ одной переменной $F(V)$ в случае отсутствия ограничений на область допустимых значений переменной V (безусловная минимизация), т.е. необходимо найти

$$F_{min} = F(V) \text{ при } V = V_{opt}$$

В большинстве случаев проектирование РЭС функция $F(V)$ задана алгоритмически и при решении задачи нахождения минимума используются численные методы

При численных методах определяется не точное значение V_{opt} , некоторый интервал неопределенности δV , внутри которого лежит точка V_{opt} .

Таким образом, задача нелинейного программирования заключается в следующем:

Найти минимум $F(V)$ при $V \in [a, b]$ в предположении, что функция $F(V)$ в этой области имеет один минимум, функция $F(V)$ – унимодальная.

Алгоритм поиска минимума $F(V)$ должен за наименьшее число обращений к функции $F(V)$ найти минимальный или заданный (по ширине) интервал δV , любая точка внутри которого принимается за оптимальное значение $V_{\text{опт}}$.

8.2.1. Метод равномерного поиска

Этот метод относится к пассивным методам (без обучения в процессе поиска) и заключается в вычислении функции $F(V)$ с заданным шагом δV на интервале $[a, b]$. Из вычисленных значений функции $F(V)$ выбирается наименьшее

Метод используется для грубого определения максимума (минимума) или для исследования поведения функции в заданном интервале.

Иногда этот метод несколько модернизируют.

Берут большой шаг δV . Определяют интервал неопределенности, т.е. интервал, в пределах которого лежит минимум

Левую границу нового интервала берем за новое значение a' , а правую за b' . Полученный новый интервал неопределенности вновь разбиваем на несколько частей и т.д., пока $\delta V \leq \delta V_{\text{рп}}$.

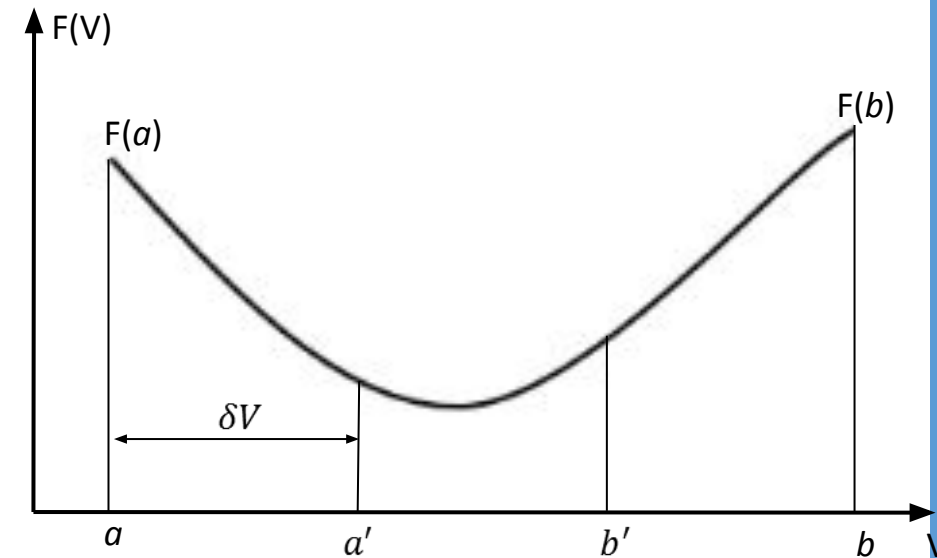
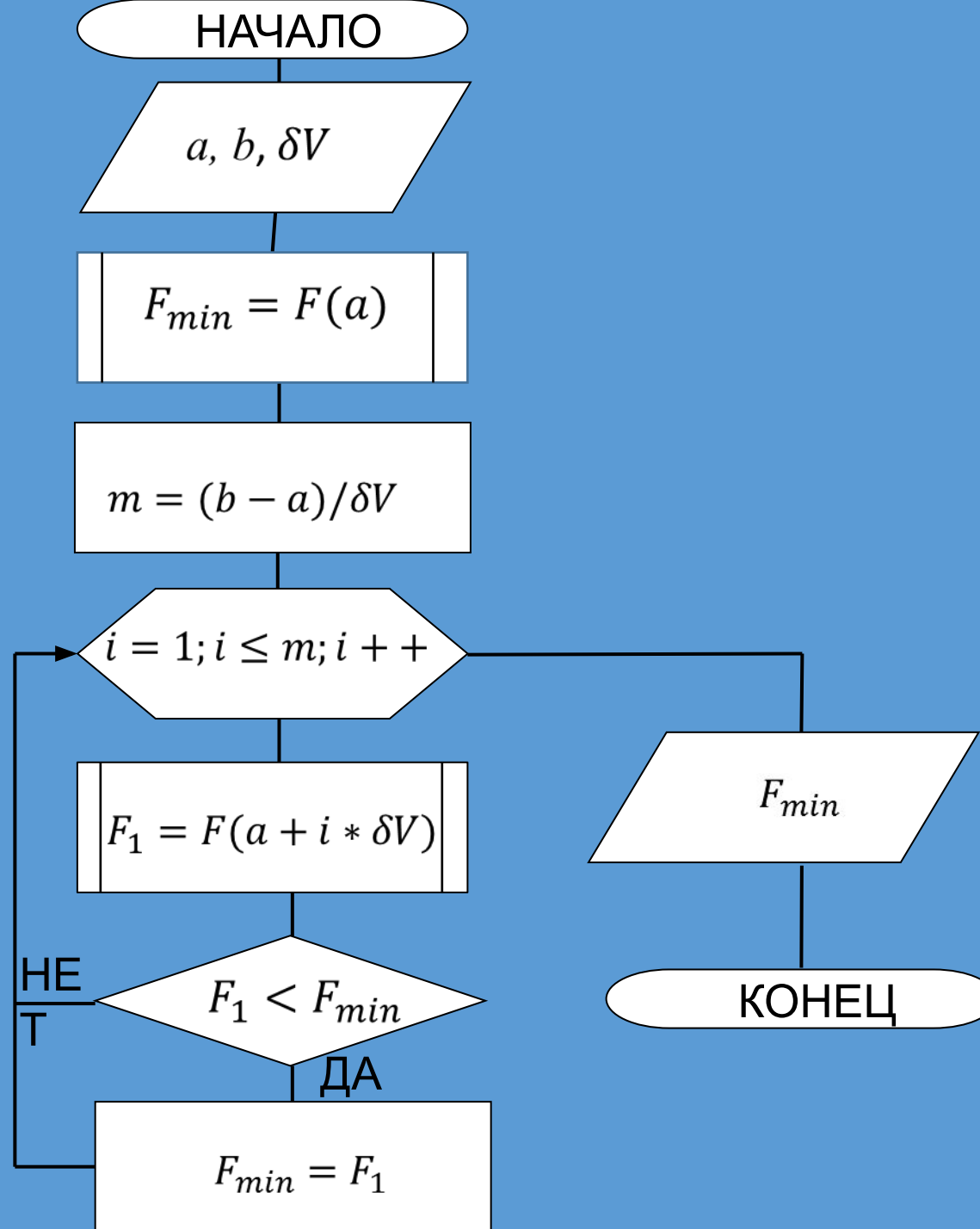
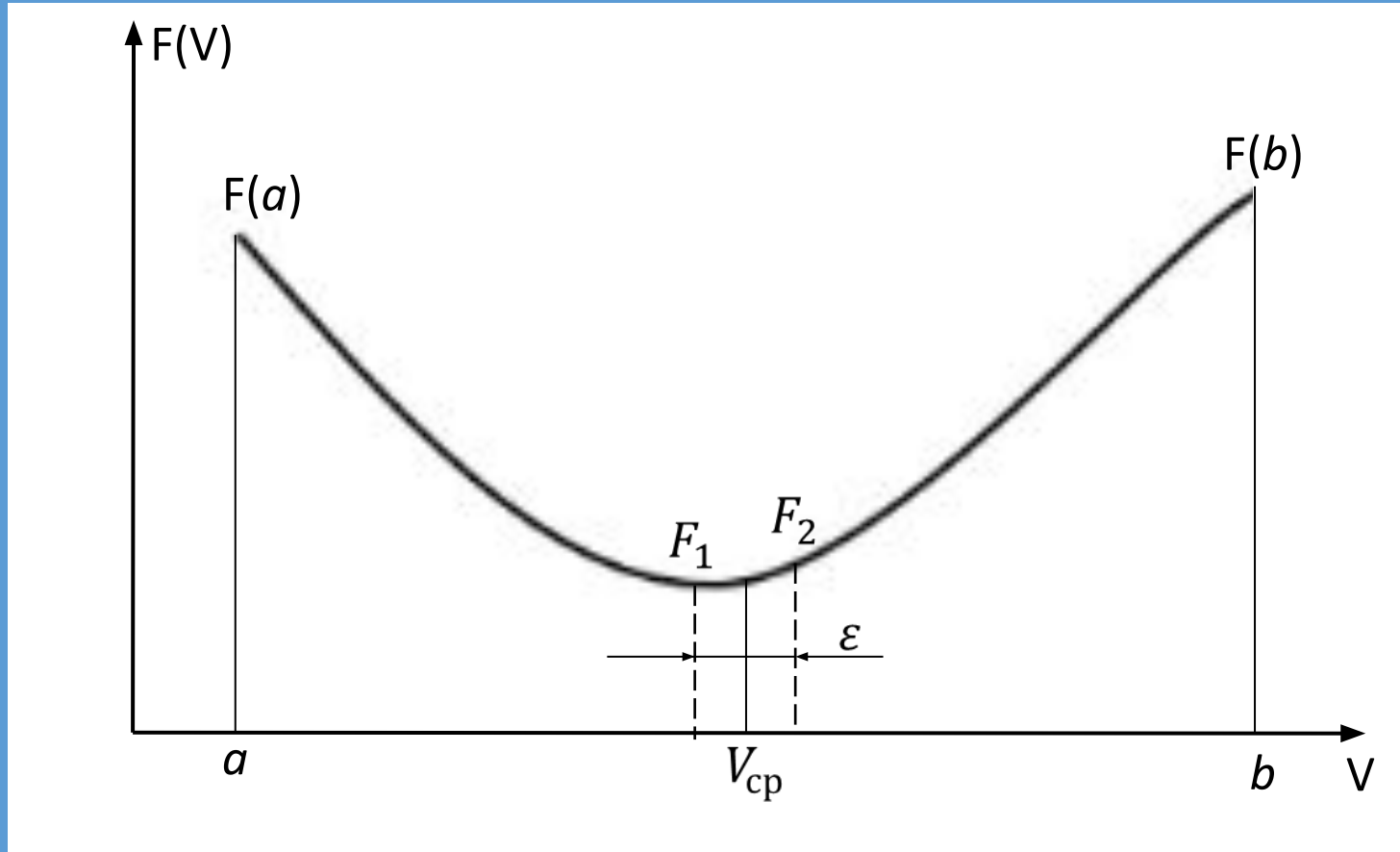
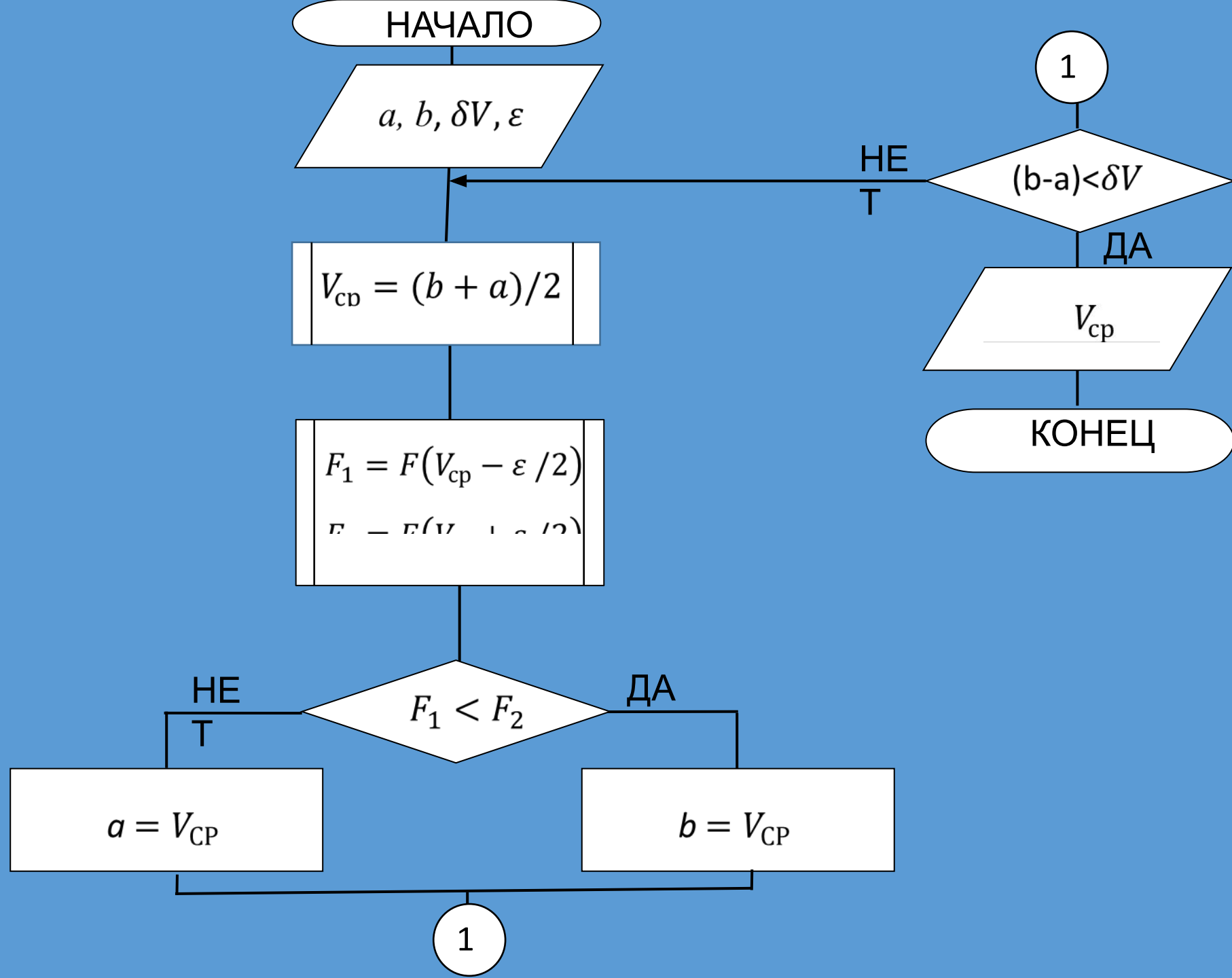


Схема алгоритма метода равномерного поиска



9.2.2. Метод деления пополам (метод дихотомии)





Сравним ИН метода равномерного поиска $\delta V_{\text{рп}}$ и метода дихотомии $\delta V_{\text{дп}}$ при одинаковом числе обращений m . Понятно, что чем меньше полученный ИН, тем лучше, тем точнее определяется значение $V_{\text{опт}}$.

Для равномерного поиска

$$\delta V_{\text{рп}} = \frac{b - a}{m}$$

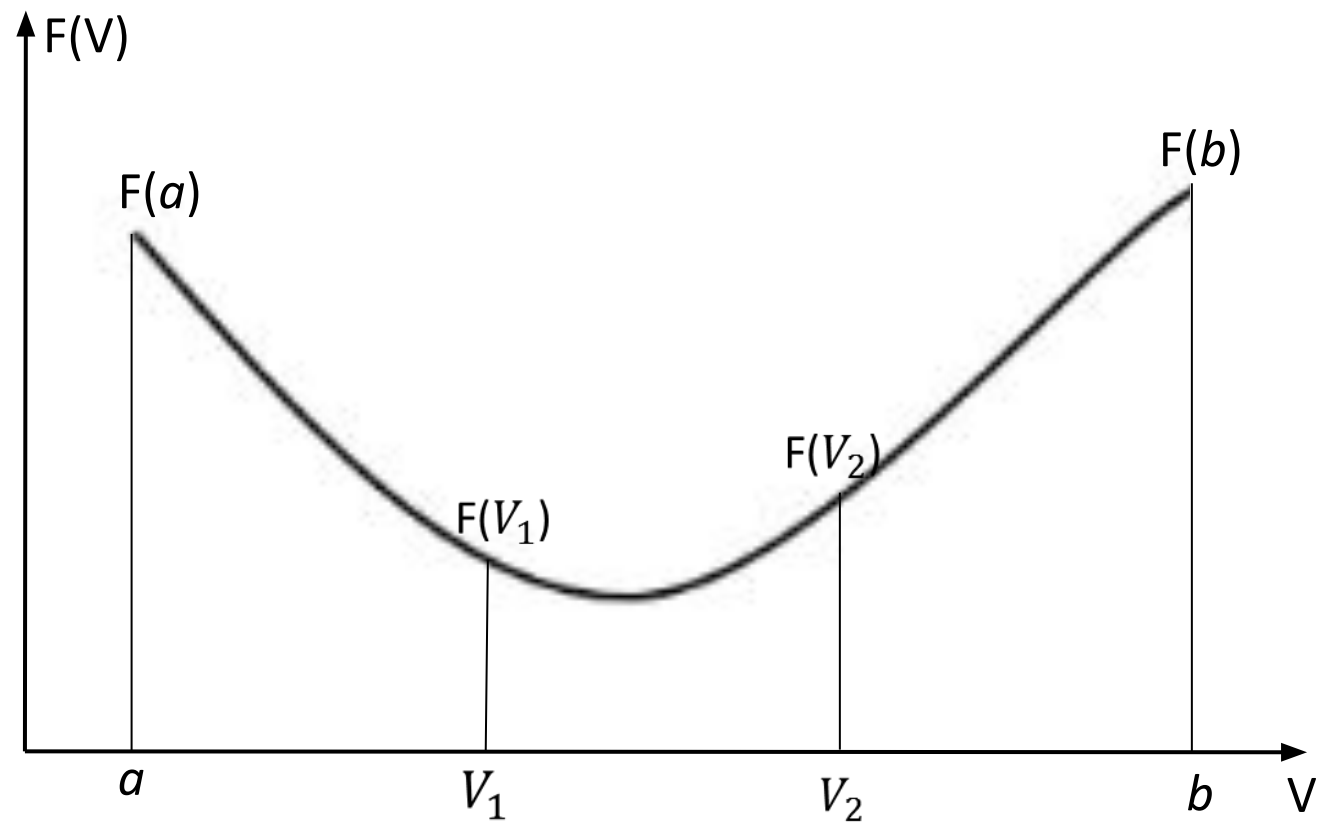
Для деления пополам

$$\delta V_{\text{дп}} = \frac{b - a}{2^{m/2}}$$

Таким образом, выигрыш метода деления пополам составит

$$\delta V_{\text{рп}} / \delta V_{\text{дп}} = \frac{2^{m/2}}{m}$$

Так при $m = 10$ это выигрыш составит 3,2, т.е. в 3,2 раза ИН в методе половинного деления будет уже.



1. $F(V_1) > F(V_2)$, то за ИН принимаем $[V_1, b]$, а интервал $[a, V_1]$ отбрасываем.
2. $F(V_1) < F(V_2)$, то ИН принимаем $[a, V_2]$.
3. $F(V_1) = F(V_2)$, то ИН $[V_1, V_2]$

9.2.3. Метод Фибоначчи

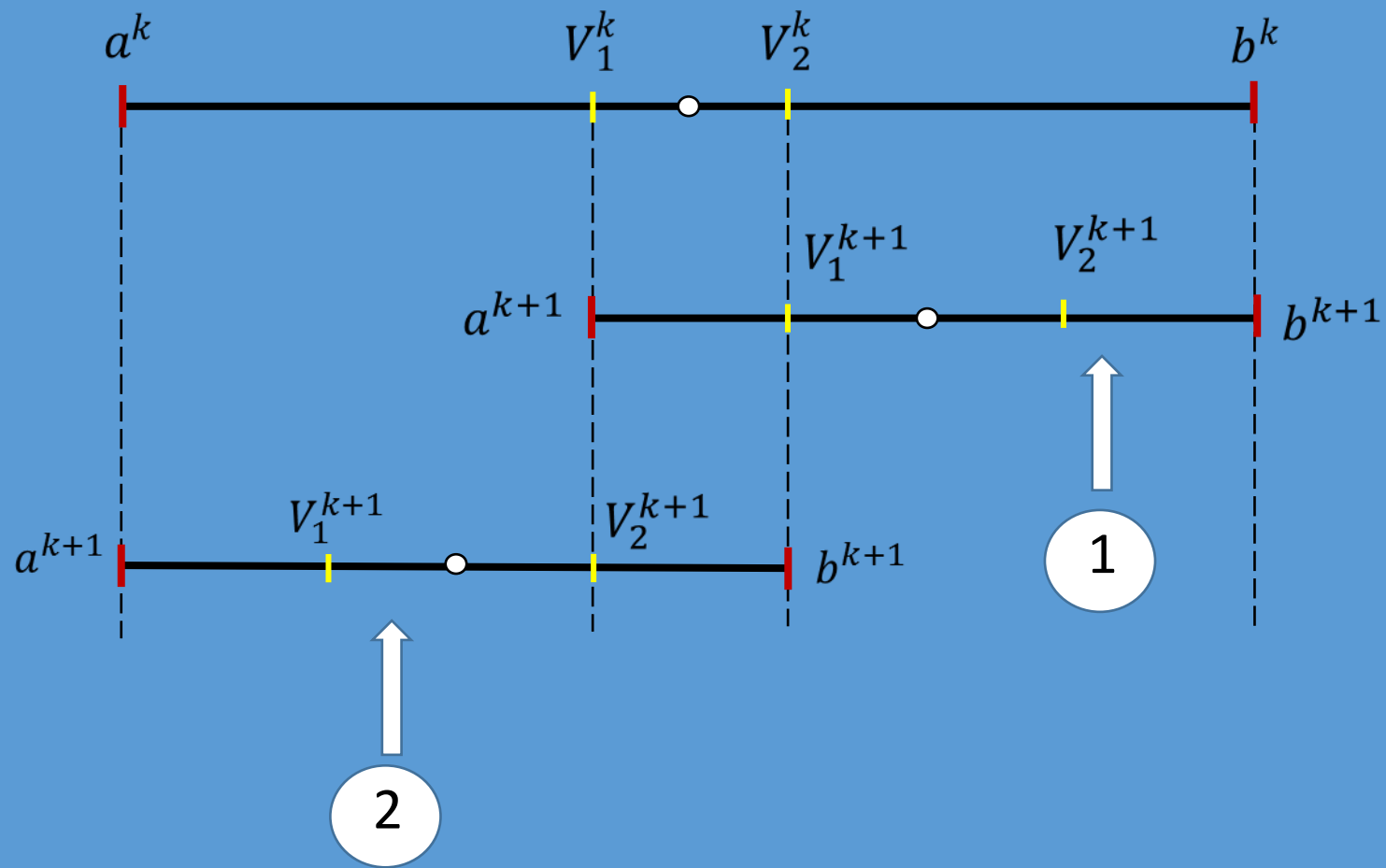
Метод Фибоначчи основан на последовательности чисел Фибоначчи $\{F_V\}$, которая определяется следующим образом:

$\Phi_{m+1} = \Phi_m + \Phi_{m-1}$, где $m = 1, 2, 3, \dots$ $\Phi_0 = 1$, $\Phi_1 = 1$, последовательность Фибоначчи имеет вид 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

Предположим, что на k -й итерации ИН равен $[a^k, b^k]$, тогда оптимальное расположение точек V_1 и V_2 будет определяться выражением

$$\begin{aligned} V_1^k &= a^k + (b^k - a^k) \frac{\Phi_{n-k-1}}{\Phi_{n-k+1}}, \\ V_2^k &= a^k + (b^k - a^k) \frac{\Phi_{n-k}}{\Phi_{n-k+1}} \end{aligned} \quad (1),$$
$$k = 1, 2, \dots, n - 1,$$

где n – заданное общее число вычислений функции.



1 $F(V_1^k) > F(V_2^k)$

2 $F(V_1^k) \leq F(V_2^k)$

Пусть $F(V_1^k) > F(V_2^k)$. Найдем ИН на $k+1$ итерации.

$$b^{k+1} - a^{k+1} = b^k - V_1^k.$$

Подставляем V_1^k из (1), получим

$$\begin{aligned} b^{k+1} - a^{k+1} &= b^k - V_1^k = b^k - a^k - (b^k - a^k) \frac{\Phi_{n-k-1}}{\Phi_{n-k+1}} = (b^k - a^k) \left[1 - \frac{\Phi_{n-k-1}}{\Phi_{n-k+1}} \right] = \\ &= (b^k - a^k) \left[\frac{\Phi_{n-k+1} - \Phi_{n-k-1}}{\Phi_{n-k+1}} \right] = (b^k - a^k) \left[\frac{\Phi_{n-k}}{\Phi_{n-k+1}} \right] \end{aligned}$$

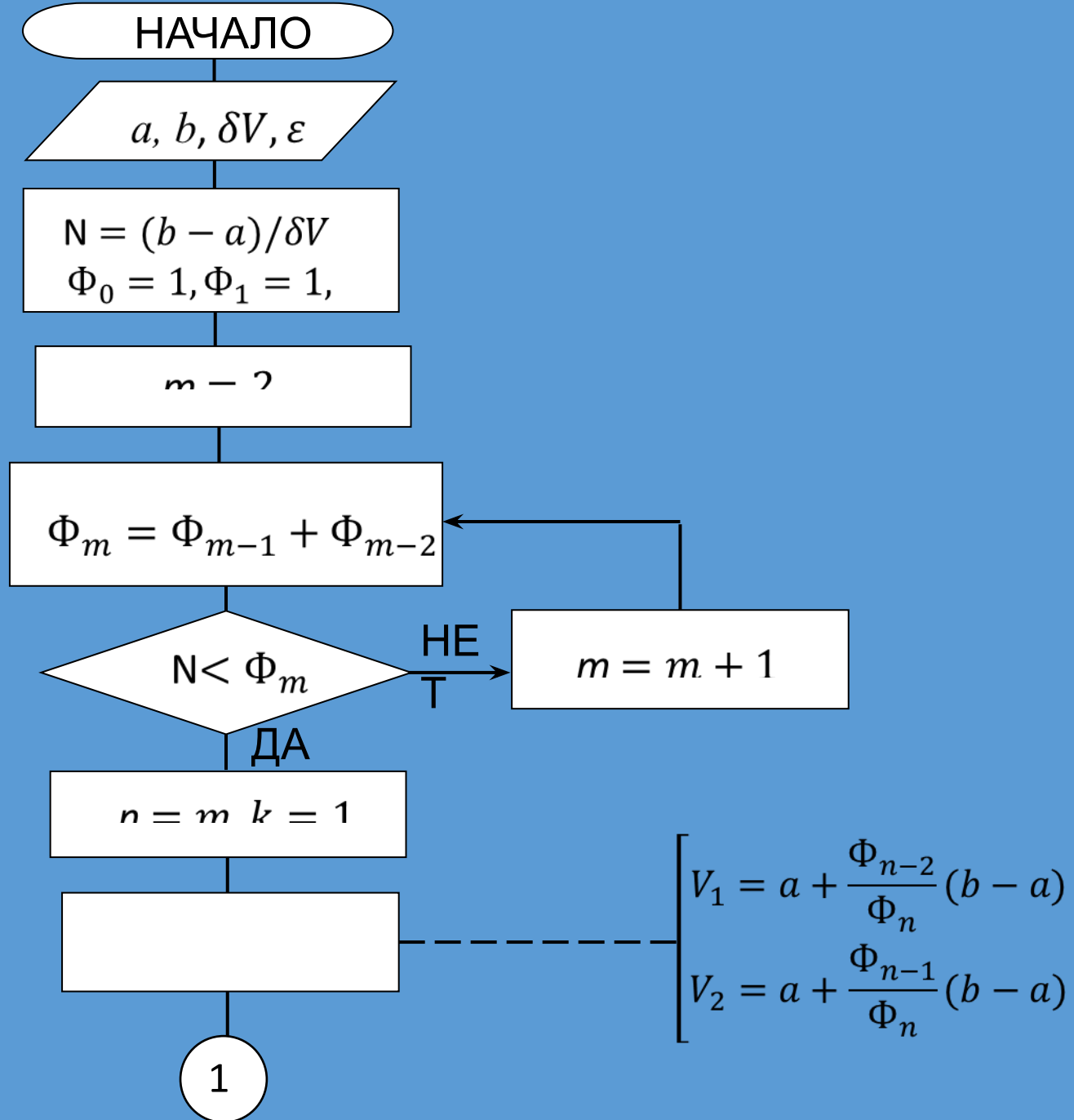
Вывод.

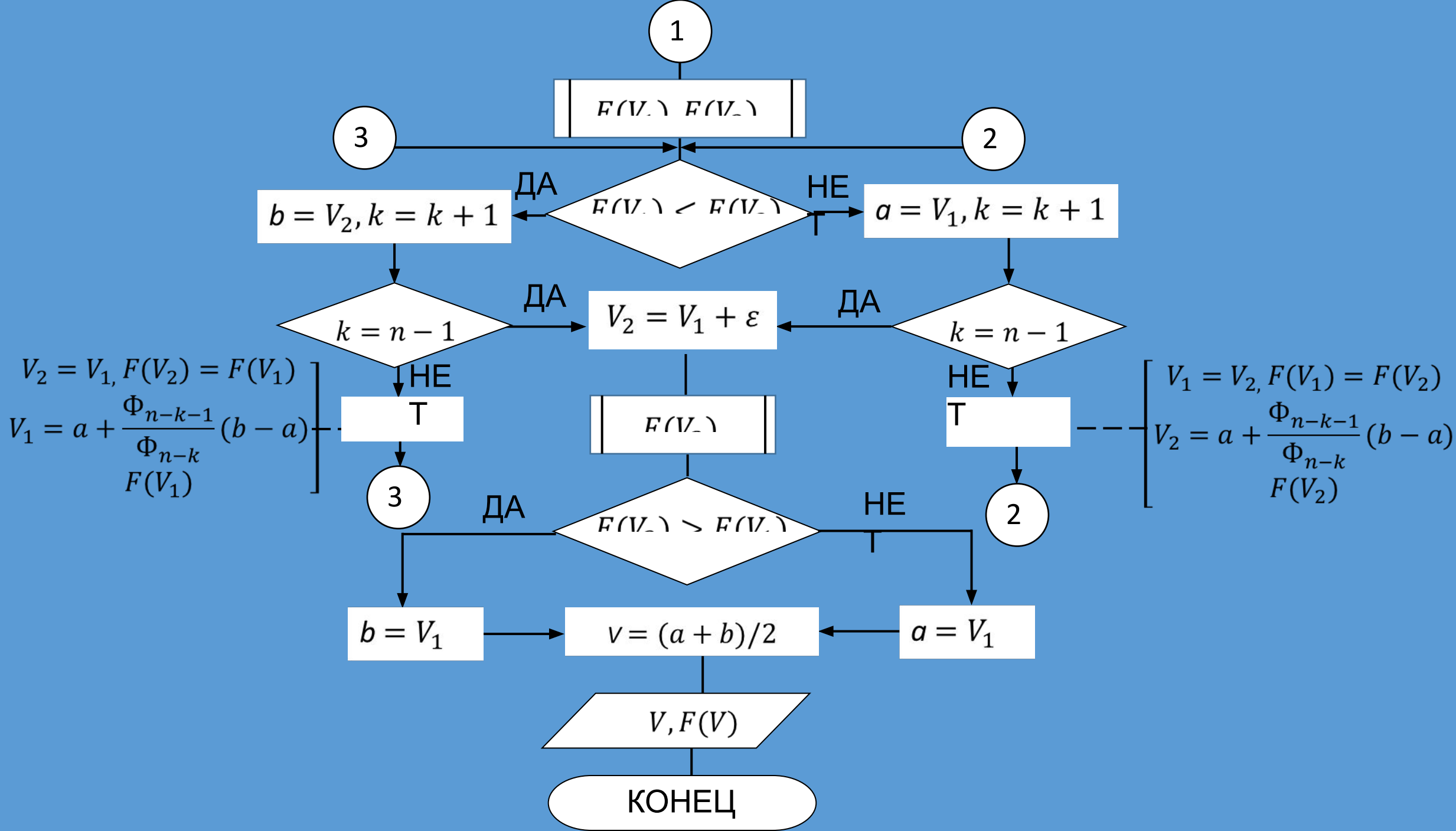
В методе Фибоначчи ИН сжимается с коэффициентом

$$\frac{\Phi_{n-k}}{\Phi_{n-k+1}},$$

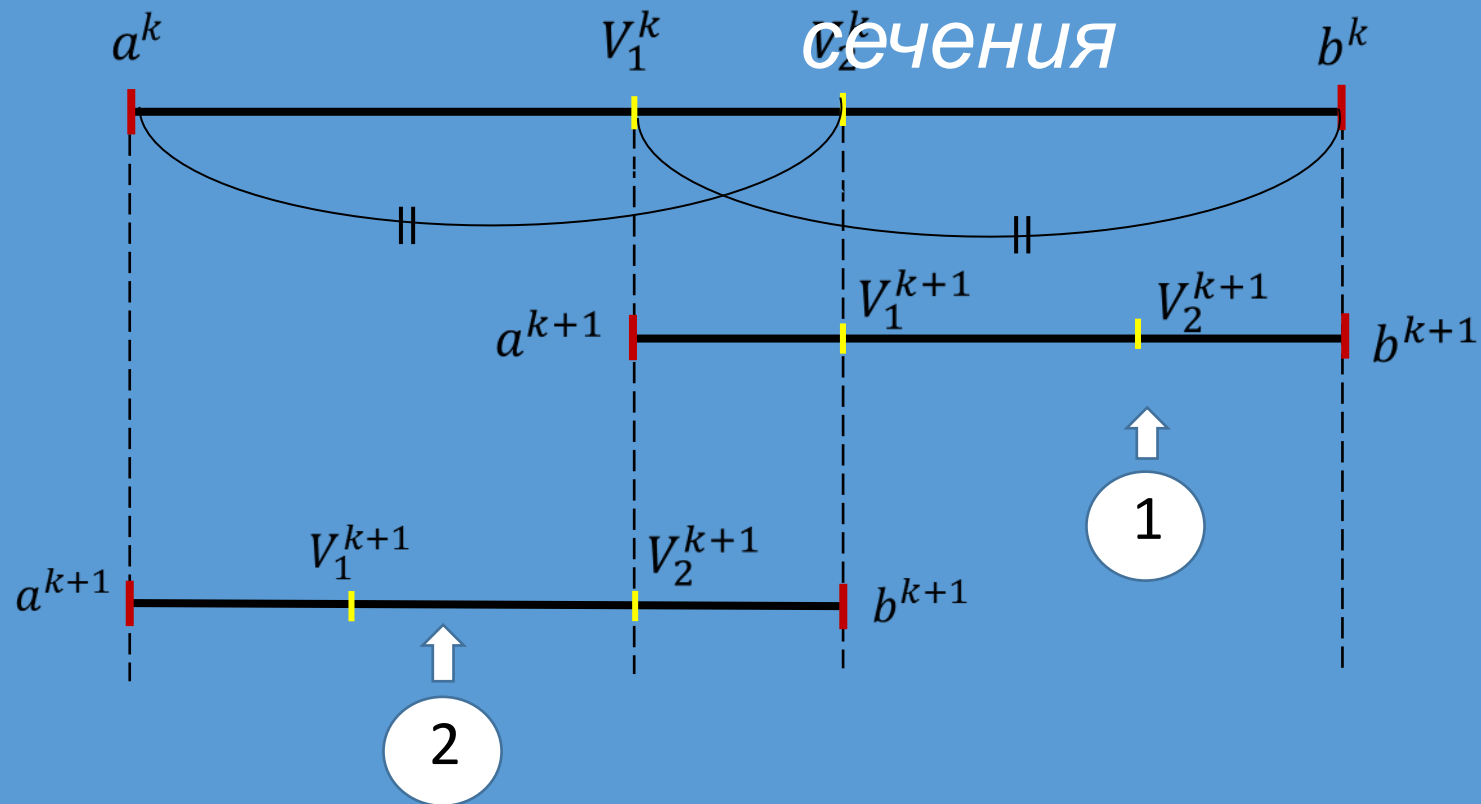
который меняется от шага к шагу. После n итераций длина ИН сократится с $b - a$ до $(b - a)/\Phi_n$.

СХЕМА АЛГОРИТМА МЕТОДА ФИБОНАЧЧИ





9.2.4. Метод золотого сечения



1. Длина нового ИН $a^{k+1} \div b^{k+1}$ не зависит от результата на k -й итерации, т.е. от того выполняется ли неравенство $F(V_1^k) > F(V_2^k)$ или $F(V_1^k) \leq F(V_2^k)$

2. Должно выполняться равенство

$$b^k - V_1^k = V_2^k - a^k \text{ (см. слайд) или } V_1^k + V_2^k = a^k + b^k. \quad (1)$$

Из этого следует, что, если

$$\text{то } V_1^k = a^k + (1 - \alpha)(b^k - a^k), \text{ где } 0 < \alpha < 1, \quad (2)$$

$$V_2^k = a^k + \alpha(b^k - a^k). \quad (3)$$

Действительно, если сложить (2) и (3), то получим (1).

Пусть $F(V_1^k) \leq F(V_2^k)$. Тогда будем иметь

$$b^{k+1} - a^{k+1} = V_2^k - a^k.$$

Воспользовавшись выражением (3), получим

$$b^{k+1} - a^{k+1} = \alpha(b^k - a^k).$$

Таким образом, α — это коэффициент сжатия ИН при переходе от k -й итерации к $k+1$. Причем, α не зависит от k , т.е., в отличие от метода Фибоначчи коэффициент сжатия не меняется от шага к шагу.

Для новой итерации V_1^{k+1} и V_2^{k+1} выбирают так (см. слайд), чтобы либо $V_1^{k+1} = V_2^k$, либо $V_2^{k+1} = V_1^k$. Тогда на $k+1$ итерации потребуется только одно новое вычисленное значение

Можно доказать, что для этого коэффициент сжатия α должен быть равен 0,618. При это на первой итерации необходимы два вычисления функции в точках V_1, V_2 , а на всех последующих - только одно, т.к. либо $V_1^{k+1} = V_2^k$ либо $V_2^{k+1} = V_1^k$.

Таким образом, длина ИН на каждом шаге сжимается с коэффициентом 0,618. После n вычислений длина ИН составит

$$\delta V = 0,618^{n-1}(b - a).$$

Сравним метод ЗС и метода Фибоначчи. При больших n ИН, найденные по этим методам, относятся, как

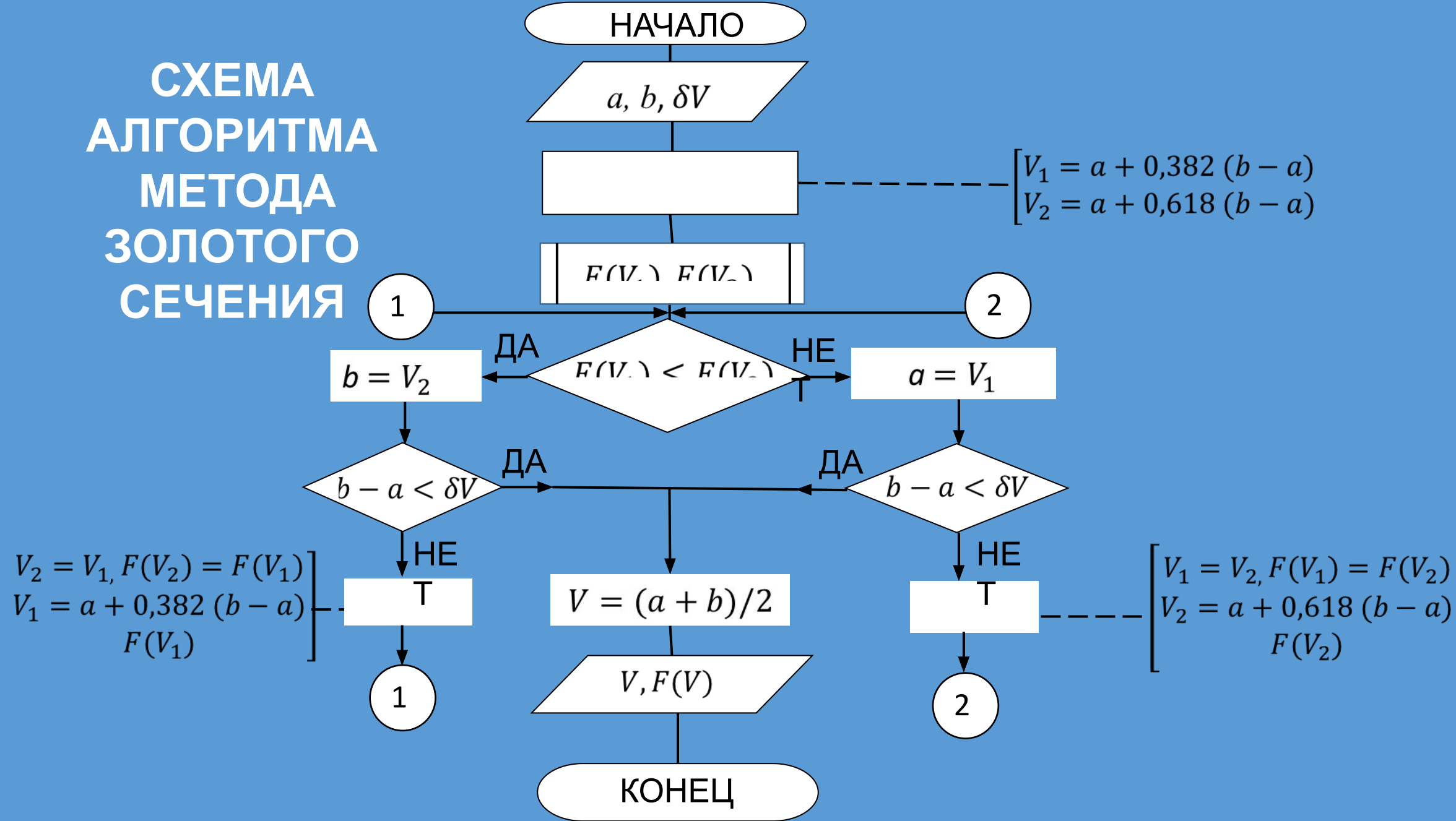
$$\frac{\delta V_{зс}}{\delta V_{ф}} = 1,17,$$

т.е. окончательный ИН в методе ЗС лишь на 17% больше, чем МФ.

Резюме.

Наиболее эффективен метод Фибоначчи, далее метод ЗС, затем метод половинного деления. При больших n методы ЗС и Фибоначчи становятся практически эквивалентными.

СХЕМА АЛГОРИТМА МЕТОДА ЗОЛОТОГО СЕЧЕНИЯ



9.3. Минимизация функций многих переменных

9.3.1. Основные определения

Возможны два подхода к решению задачи отыскания минимума ЦФ многих переменных $F(\bar{V}) = F(v_1, v_2, \dots, v_n)$ при отсутствии ограничений на диапазон изменения неизвестных.

□ Первый подход лежит в основе *косвенных методов оптимизации* и сводит решение задачи оптимизации к решению системы нелинейных уравнений, являющихся следствием условий экстремума функции многих переменных. Однако решение систем нелинейных уравнений - задача весьма сложная и трудоемкая

➤ На практике используют второй подход к минимизации функций, составляющий основу *прямых методов*. Суть их состоит в построении последовательности векторов $\bar{V}^0, \bar{V}^1, \dots, \bar{V}^m$, таких, что $F(\bar{V}^0) > F(\bar{V}^1) > \dots > F(\bar{V}^m)$.

Переход (итерация) от точки \bar{V}^k к точке \bar{V}^{k+1} , $k = 0, 1, 2, \dots$, состоит из двух этапов:

✓ выбор направления движения из точки \bar{V}^k ;

✓ определение шага вдоль этого направления.

Методы построения таких последовательностей часто называют *методами спуска*, так как осуществляется переход от больших значений функций к меньшим.

Математически методы спуска описываются соотношением

$$\overline{V}^{k+1} = \overline{V}^k + a_k \overline{P}^k, \quad k = 0, 1, 2, \dots,$$

где \overline{P}^k - вектор, определяющий направление спуска; a_k - длина шага.

Различные методы спуска отличаются друг от друга способами выбора двух параметров - направления спуска и длины шага вдоль этого направления.

Качество сходящихся итерационных методов оценивают по скорости сходимости

В методах спуска решение задачи теоретически получается за бесконечное число итераций. На практике вычисления прекращаются при выполнении некоторых критериев (условий) останова итерационного процесса. Например, это может быть условие малости приращения аргумента

$$\left| \overline{V}^k - \overline{V}^{k-1} \right| < \varepsilon$$

или функции

$$\left| F(\overline{V}^k) - F(\overline{V}^{k-1}) \right| < \gamma$$

Методы поиска точки минимума называются *детерминированными*, если оба элемента перехода от \overline{V}^k к \overline{V}^{k+1} (направление движения и величина шага) выбираются однозначно по доступной в точке \overline{V}^k информации

Если же при переходе используется какой-либо случайный механизм, то алгоритм поиска называется *случайным поиском минимума*.

Детерминированные алгоритмы безусловной минимизации делят на классы в зависимости от вида используемой информации.

- ❖ Если на каждой итерации используются лишь значения минимизируемых функций, то метод называется *методом нулевого порядка*
- ❖ Если, кроме того, требуется вычисление первых производных минимизируемой функции, то имеют место методы *первого порядка*
- ❖ При необходимости дополнительного вычисления вторых производных – это будут *методы второго порядка*.

9.3.2. Численные методы безусловной оптимизации нулевого порядка

Метод покоординатного спуска

Это наиболее простой способ. Направления спуска выбирается вдоль одной из координатных осей. Это позволяет поочередно изменять все независимые переменные v_1, v_2, \dots, v_n так, чтобы на каждой из них достигалось наименьшее значение целевой функции.

Очередность варьирования независимых переменных при этом устанавливается произвольно и не меняется в процессе поиска. В результате многомерный поиск заменяется последовательностью одномерных поисков с любой стратегией минимизации функции одной переменной (см. методы, описанные выше).

Данный метод эффективен в случае единственного минимума функции. Для уменьшения числа вычислений величину шага a меняют при каждом переходе от поиска минимума по одной переменной к поиску минимума по другой переменной.

