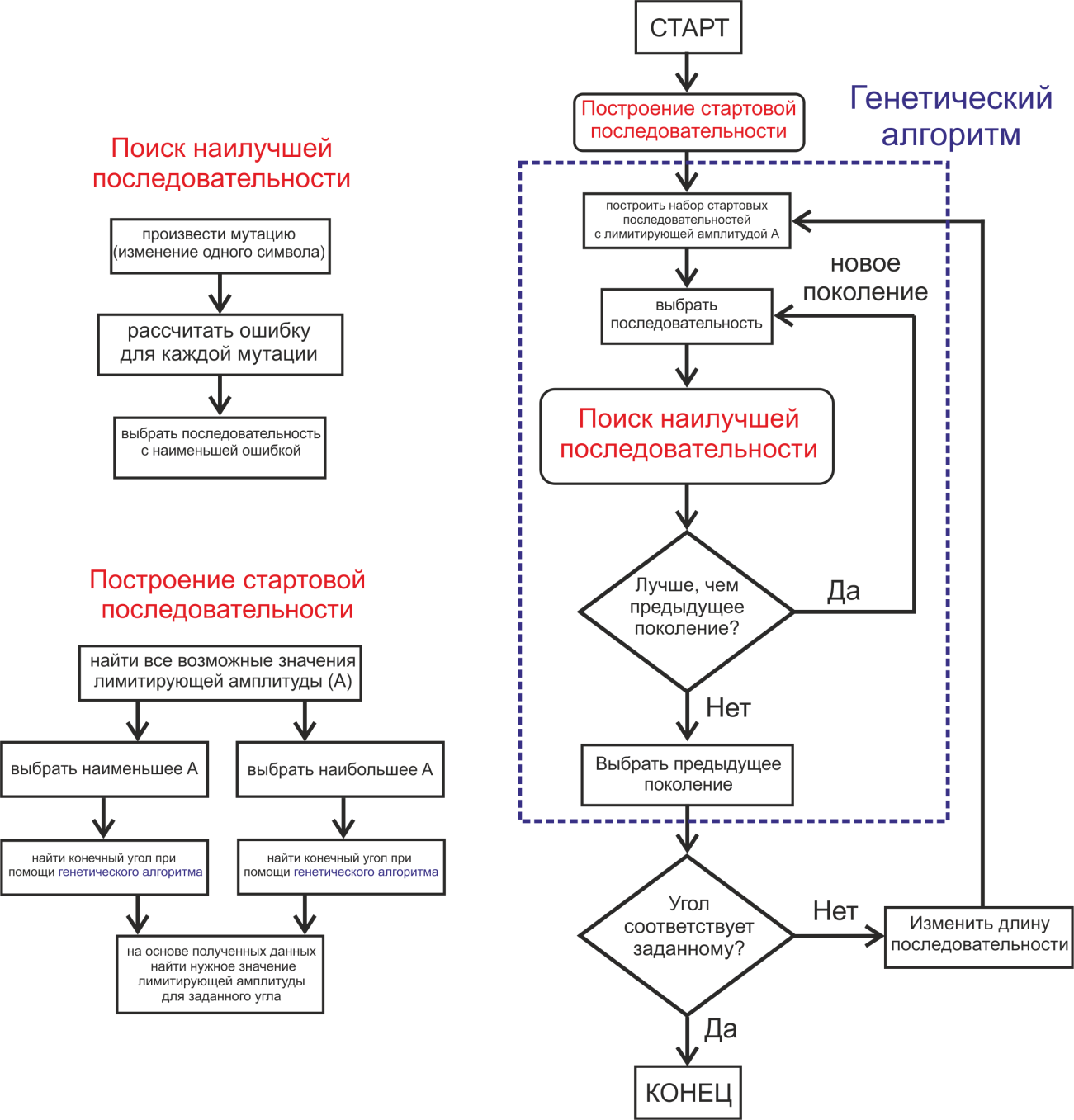
**Optimized High Fidelity Sequence Search**

Общий принцип работы алгоритма:  


1. Находим все возможные значения лимитирующей амплитуды для заданных параметров
2. Выбираем самое маленькое значение амплитуды из найденных. Строим для него стартовую последовательность и применяем у ней «тело» алгоритма, изменяя последовательность до тех пор, пока не будет найдена наименьшая утечка. Находим угол поворота всей последовательности.
3. Аналогично для самого большого значения амплитуды и некоторого среднего значения (из найденных в п.1). При помощи интерполяции находим, какое значение амплитуды даст угол, близкий к желаемому.
4. Строим стартовую последовательность, используя найденное значение амплитуды. Применяем «тело» алгоритма, находим угол.
5. Если угол не соответствует желаемому, тогда начинаем изменять длину последовательности. Увеличиваем, если надо уменьшить угол или уменьшаем длину, если надо увеличить угол.
6. При изменении длины, стартовая последовательность строится заново, но выбранной в начале значение амплитуды не меняется. Считаем утечку и угол.
7. Продолжаем до тех пор, пока не найдем последовательность с нужным углом.

Примечание: в реальности нужно значение угла может «проскочить» (пример: нам нужен угол 0.002, при изменении длины угол меняется с 0.0019 до 0.0021, минуя нужное значение), поэтому ищется не точное значение угла, а значение в некотором диапазоне

Скрипты и функции:

* main.m – основной файл, через который запускается скрипт
* GenSearch.m – функция генетического алгоритма, которая ищет последовательность с минимальной утечкой для заданных параметров , , и M (длина последовательности)
* Fidelity.m – считает утечку для заданной последовательности
* RotateCheck.m – проверяет на какой угол поворачивает состояние кубита вся последовательность
* NewThetaOptimizer.m – находит, какой угол поворота одного импульса даёт поворот всей последовательности, близкий к π/2
* ChangingOneElement.m – для заданной последовательности выдаёт все возможные мутации изменения одного символа в ней
* CreateAmpThresholds.m – рассчитывает для заданных параметров , и M все возможные значения параметра лимитирующей амплитуды для построения стартовой последовательности
* CreateStartSCALLOP.m – строит стартовую последовательность для заданных параметров , , , M и рассчитанного при помощи CreateAmpThresholds значения лимитирующей амплитуды стартовую последовательность
* WriteSequence.m – выводит последовательность на экран
* CountFidelity.m – вспомогательный скрипт, не используемый в алгоритме. Считает утечку для заданной последовательности и всех нё параметров