Grover for satisfiability problems

Exercise: 3-SAT Problem

The 3-Satisfiability Problem (3-SAT) is a classic problem in computational complexity theory and an important example of an NP-complete problem. It asks whether an assignment of Boolean variables exists that satisfies a given Boolean formula. The formula is expressed in conjunctive normal form (CNF), where each clause contains exactly three literals. A literal is a Boolean variable or its negation, and a clause is a disjunction (logical OR) of literals. The entire formula is a conjunction (logical AND) of these clauses. Consider the Boolean formula below:

$$F = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3)$$

In this formula, each clause is enclosed in parentheses and contains exactly three literals combined using the logical OR operator $\vee$. The entire formula F is a conjunction of these clauses combined using the logical AND operator $\wedge$.

The goal is to determine whether an assignment of truth values (true or false) exists to the variables x_1, x_2, x_3 such that the entire formula F evaluates to true. This involves checking all possible combinations of truth values for the variables and identifying those that satisfy all the clauses simultaneously.

Tasks:

1) Propose a solvable 3-SAT boolean formula, different from the one in the text.

2) Implement Grover's algorithm to find a solution. Comment on the results obtained.

3) Bonus: Choose a real-world problem that can be cast as a 3-SAT problem, e.g, scheduling. Assess the algorithm's efficiency in finding a solution.

More examples of SAT problems:

- **Software and Hardware Verification**: SAT is a crucial tool in model checking, which verifies the correctness of hardware and software designs by checking if a design satisfies its specifications.
- **Artificial Intelligence (AI)**: Many AI problems can be expressed as SAT problems, such as classical planning, automated theorem proving, and determining if a graph is colorable.
- **Scheduling:** SAT can be used to solve scheduling problems, finding a valid schedule that satisfies all constraints.
- **Design of Experiments**: It helps in designing experiments by determining a set of conditions that will lead to a successful outcome.
- **Optimization**: The Max-SAT variant is used for optimization problems where you want to find the best solution that satisfies the maximum number of clauses, such as in FPGA routing and bioinformatics.
- **E-commerce**: Some e-commerce problems can be formulated and solved using SAT.
- **Formal Specification**: Languages like Alloy use SAT solvers to automatically check properties of a given specification, making it easier to find bugs in software models.