r/ThingsYouDidntKnow • 3 mo. ago
TheStocksGuy

Instant Query - Theory using SAT, TSP, FLT to think of how the wheel would be predetermined

To explore how combining Fermat's Last Theorem (FLT), the Traveling Salesman Problem (TSP), and the Boolean Satisfiability Problem (SAT) can help solve complex path calculations, let's break down how we can merge each concept's constraints, solution spaces, and feasibility checks into a framework for optimized and validated paths. By harnessing the unique strengths of each, we can structure paths that are not only efficient but also logically sound and free from impossible configurations.

1. Using Fermat's Last Theorem (FLT) to Define Constraints on Feasible Paths

- **FLT as a Constraint**: Fermat's Last Theorem tells us that for any integer ($n > 2$), there are no positive integer solutions to ($a^n + b^n = c^n$). This fact creates a kind of **constraint on possible values** that we can leverage as we set up paths.
- **How We Apply It**: Instead of using FLT to find a solution, we use it to **exclude impossible configurations**. For any path segment or point that satisfies ($a^n + b^n = c^n$) with ($n > 2$), we discard it from our solution space. This essentially maps out "infeasible zones," where potential paths are invalid, helping us eliminate impossible paths early in the calculation.

2. Applying Traveling Salesman Problem (TSP) Techniques for Path Optimization

- **TSP for Path Calculation**: The TSP's core focus is to find the **optimal (typically shortest) route** that visits each "city" (or point) exactly once. Here, each "city" in our space represents a feasible configuration of values (like (a), (b), (c) coordinates) that doesn't violate FLT constraints.
- **How We Adapt It**: By treating feasible configurations as nodes or "cities," we can use TSP methods to find the most efficient path through the space that **minimizes costs** (whether distance, error, or computation). This lets us avoid unnecessary calculations and focus on paths that meet our criteria, such as minimal computational expense or optimal resource use. Each path then represents a possible sequence of solutions, all of which avoid infeasible FLT zones.

3. Using SAT for Consistency and Feasibility Checks

- **SAT for Logical Constraints**: SAT is a logic-based framework that ensures **feasibility and logical consistency** across multiple conditions. By encoding our constraints from FLT and TSP as logical clauses, SAT can serve as a "feasibility filter" at each step in the path.
- **Application in the Framework**: For every configuration on a potential path, SAT can verify that no values satisfy any restricted configurations (like ($a^n + b^n = c^n$) for ($n > 2$)). It also ensures that each path meets our conditions, such as avoiding previously traversed configurations and maintaining an optimal distance. If a

Skip to main content

Create

Combined Framework to Achieve Optimized, Feasible Paths

1. **Initializing the Solution Space**:
   - We begin by setting up a wide range of potential configurations (values for (a), (b), (c)) and define possible paths between them, often visualized as points in a multi-dimensional space or nodes in a graph.
2. **Applying FLT Constraints to Exclude Infeasible Paths**:
   - Using FLT as a filter, we remove configurations that satisfy ($a^n + b^n = c^n$) for any ($n > 2$), creating boundaries around infeasible zones within our space. This helps us identify and retain only the paths where solutions could exist.
3. **Optimizing Path Selection with TSP**:
   - With feasible configurations defined, TSP optimization techniques help us find the shortest, least costly path. Each valid point is treated as a "node" or "city," and the optimal path ensures we move through this solution space as efficiently as possible. This step is crucial for balancing computational resources with solution accuracy.
4. **Filtering for Logical Consistency with SAT**:
   - For each calculated path, SAT constraints check that logical conditions are consistently met. This includes ensuring that paths don't revisit configurations that should be unique, verifying that no segment of the path enters infeasible FLT regions, and upholding minimal distance conditions. This real-time verification step ensures our framework remains optimized and logically valid throughout.

By combining these methods, we create a robust framework capable of narrowing down solutions within a massive, complex space, removing infeasible configurations, and optimizing paths that satisfy all constraints. This approach leverages the best of FLT's exclusion principles, TSP's optimization structure, and SAT's logical filtering to create a single, highly structured solution path.

1 0 Share Approved 3 months ago

Post Insights

Only the post author and moderators can see this

230	100%	0	0
Total Views	Upvote Rate	Comments	Total Shares

Hourly views for first 48 hours

Some insights are no longer available because this post is older than 45 days

Add a comment