“Motion Planning using Convex Optimization” **explain this project with Graph of Convex Sets" (GCS) approach**

Of course. A project on **Motion Planning using Convex Optimization** with the **Graph of Convex Sets (GCS)** approach is an excellent choice that aligns perfectly with your coursework and career interests. It's a cutting-edge technique in robotics that transforms a complex, non-convex problem into a solvable one.

Here is a detailed explanation of the project.

The Core Problem: Why Motion Planning is Hard

The fundamental challenge in robot motion planning is that the "free space"—all the valid configurations where the robot doesn't collide with obstacles—is typically a very complex, non-convex shape. Finding the "best" path (e.g., shortest, fastest, or most energy-efficient) through this complicated space is an inherently non-convex optimization problem. These problems are notoriously difficult to solve efficiently and often lack guarantees of finding the optimal solution.

The Solution: The Graph of Convex Sets (GCS) Approach

The GCS method, largely pioneered by researchers at MIT, is an elegant framework that tackles this challenge by breaking the problem down. Instead of dealing with the entire complex, non-convex free space at once, it first decomposes this space into a collection of simpler, overlapping *convex* regions. [1] [2] [3]

Think of it like this: Navigating a cluttered room is hard. But if you first identify a series of overlapping safe zones (like open floor patches), you can plan a path by moving from one safe zone to the next. Because these zones are simple (convex), planning within them is easy.

How GCS Works, Step-by-Step

1. **Decompose the Free Space into Convex Sets**: The first step is to cover the robot's collision-free space with a collection of overlapping convex sets, such as polytopes or ellipsoids. This can be done in the robot's 2D or 3D workspace or its more abstract, high-dimensional configuration space (which includes all its joint angles) . Each of these convex sets represents a "safe" region where planning is trivial because any two points within it can be connected by a straight line that is also guaranteed to be collision-free.
2. **Build the Graph**: These convex sets become the **vertices** (nodes) of a graph. An **edge** is created between two vertices if their corresponding convex sets overlap . This graph serves

as a high-level, abstract "roadmap" of the free space, showing how the robot can move between adjacent safe regions.[2] [1]

3. **Formulate a Joint Optimization Problem**: This is the key innovation of GCS. Instead of first finding a discrete path through the graph and then smoothing it out later, GCS formulates a *single, large optimization problem* that simultaneously finds the best sequence of convex sets *and* the optimal continuous trajectory through them. This is formulated as a **Mixed-Integer Convex Program (MICP)** :[4] [1] [2]
    - **Integer Variables**: These are binary (0 or 1) variables that correspond to each edge in the graph. An integer is set to '1' if the optimal path uses that edge and '0' otherwise. These variables solve the *discrete* part of the problem: which sequence of convex sets to traverse.
    - **Continuous Variables**: These variables define the actual smooth trajectory for the robot, often represented as a spline or a polynomial. These solve the *continuous* part of the problem: the exact path the robot follows.
    - **Convex Cost Function**: This defines the objective of the optimization. It's a convex function that can represent path length, travel time, control effort, or a weighted combination of these factors.[5] [1]
    - **Convex Constraints**: These are the rules of the problem. They ensure that the final trajectory remains within the chosen convex sets and respects the robot's physical limitations (e.g., maximum velocity and acceleration).[1]
4. **Solve the MICP**: While MICPs can be computationally hard in general, the GCS formulation has a specific structure that often allows for a very tight "convex relaxation." This means the problem can be solved efficiently by modern optimization solvers to find a high-quality, and often globally optimal, trajectory.[4] [1]

Why This is an Excellent Project for You

- **Directly Relevant to Your Course**: The entire framework is built on the core concepts of your ELL7100 course: convex sets, convex optimization, quadratic forms, and solving systems of linear equations.
- **State-of-the-Art Technique**: This is not just a textbook problem. GCS is a cutting-edge research topic actively being used to solve complex planning problems for drones, robot arms, and legged robots.[3] [2]
- **Provides Provable Guarantees**: Unlike many popular sampling-based planners (like RRT), GCS can provide guarantees on the feasibility and optimality of the generated path.[6] [5]
- **High-Impact for Career Goals**: Successfully implementing a GCS-based planner is a strong signal to companies like NVIDIA. It demonstrates a deep, practical understanding of the mathematical foundations behind modern autonomous systems.

⁂

1. https://arxiv.org/html/2410.08909v1
2. https://slides.com/russtedrake/planning-with-graphs-of-convex-sets-8adf9d
3. https://news.mit.edu/2023/new-optimization-framework-robot-motion-planning-1130

4. https://www.youtube.com/watch?v=BHd9yjNl2Dg
5. https://arxiv.org/html/2411.18913v2
6. https://www.arxiv.org/abs/2411.18913
7. https://nurse.plus/become-a-nurse/glasgow-coma-scale/
8. https://www.ncbi.nlm.nih.gov/books/NBK513298/
9. https://journals.sagepub.com/doi/10.1177/02783649241302419
10. https://www.youtube.com/watch?v=4zvVnUv3ZYw
11. https://en.wikipedia.org/wiki/Glasgow_Coma_Scale
12. https://www.youtube.com/watch?v=5gopIqGN5wY
13. https://www.youtube.com/watch?v=rSbpfAVF_7c
14. https://my.clevelandclinic.org/health/diagnostics/24848-glasgow-coma-scale-gcs
15. https://github.com/reso1/stgcs
16. https://dl.acm.org/doi/abs/10.1109/TRO.2023.3291463
17. https://www.glasgowcomascale.org
18. https://www.roboticsproceedings.org/rss20/p113.pdf
19. project.pdf
20. https://groups.csail.mit.edu/robotics-center/public_papers/Marcucci24a.pdf
21. https://groups.csail.mit.edu/robotics-center/public_papers/Wrangel24.pdf