I'll assign roles as Person A (Data/Modeling Specialist), Person B (Implementation Specialist), Person C (Evaluation/Reporting Specialist). You can adjust based on your actual skills.

Task Division by Phases

Each phase builds on the previous, with shared deliverables (e.g., code on GitHub, reports on shared docs).

Days 1-2 (Dec 29-30: Data Processing)

Person A (Lead: Data Specialist): Clean and process the Excel data (extract rooms, capacities, courses, groups; estimate missing params like sizes). Deliver: Processed CSV/JSON dataset.

Person B (Support): Help parse data into code-readable format (e.g., Python dicts).

Person C (Support): Validate data against real FSTM sources (quick surveys if possible) and document assumptions.

Group Goal: Ready dataset by end of Day 2.

Days 3-5 (Dec 31-Jan 2: Algorithm Implementation)

Person B (Lead: Implementation Specialist): Code the Simulated Annealing algorithm (initial solution generation, neighborhood search, cooling schedule, constraint penalties using the formulas). Deliver: Working prototype script.

Person A (Support): Integrate processed data into the code (e.g., load capacities, slots).

Person C (Support): Test basic runs for feasibility (no hard violations) and debug.

Group Goal: Prototype generating timetables by end of Day 5.

Days 6-8 (Jan 3-5: Evaluation)

Person C (Lead: Evaluation Specialist): Run experiments on FSTM data, compute metrics (e.g., hard/soft violations, runtime), compare with manual timetable. Deliver: Results spreadsheet/report.

Person B (Support): Optimize code (e.g., add Genetic Algorithms variant if SA struggles).

Person A (Support): Analyze outputs for patterns (e.g., gap reduction) and suggest improvements.

Group Goal: Performance insights by end of Day 8.

Days 9-10 (Jan 6-7: Finalization)

Person C (Lead: Reporting Specialist): Compile full report (LaTeX doc, code docs, results).

Person B (Support): Finalize code (clean, comment, GitHub repo).

Person A (Support): Add modeling refinements (e.g., better encoding) and conclude with recommendations.

Group Goal: Complete system and report by Jan 7.

General Guidelines

Collaboration: Daily 15-min stand-ups (e.g., via Zoom) to sync. Use GitHub for code pulls, Trello for task tracking.

Risk Mitigation: If one falls behind, rotate (e.g., Person A helps coding).

Workload Balance: Each has ~30-40% lead time, with supports to avoid burnout.

Justification: This division maximizes efficiency—data prep first, then coding, then testing—mirroring metaheuristic workflows.