TP – Constraint Satisfaction Problems Queens/GAs/CW/PP

CP468-A | Group-18

Dr. Ilias Kotsireas

Fri, Oct 31, 2025

Name	Student ID	Student email	Groupwork Accountability Measures
Daniel Gordon	169032316	gord2316@mylaurier.ca	Person A – CSP model design and constraints
Sam Oreskovic	169068090	ores8090@mylaurier.ca	Person B – AC-3 algorithm implementation and queue tracking
Kyle Fernandes (Group Rep)	169069584	fern9584@mylaurier.ca	Person C – Input parser and queue visualization
Yusuf Tanveer	169053297	tanv3297@mylaurier.ca	Person F – Documentation & Integration lead (assembled report and verified code organization)
Ammar Zarzour	169066197	zarz6197@mylaurier.ca	Person E – Testing and performance analysis

CP468 Term Project – N-Queens with MIN-CONFLICTS (Fair 5-Person Work Distribution Plan)

🎯 Project Goal

Implement and analyze the **MIN-CONFLICTS local-search algorithm** for solving the N-Queens problem.
Test performance for **n = 10, 100, 1 000, 10 000, 100 000, 1 000 000**, and document results, analysis, and visual output.

👥 Team Roles (Balanced Workload)

Member	Role	Core Responsibilities	Deliverables
Person A – Algorithm Lead	Designs and implements the **MIN-CONFLICTS algorithm**	• Implement initialization (random board) • Write main loop (min-conflicts selection) • Implement conflict-count function • Optimize runtime for large n • Comment code and document logic	`min_conflicts.py` (core algorithm) + 1-page *Algorithm Design* write-up
Person B – Board Representation & Validation Developer	Handles **data structures** and **solution checking**	• Implement board representation (array / list) • Write `is_solution()` function to verify valid configuration • Add utilities for random restarts and statistics • Co-write *Design Choices* section	`board_utils.py` + solution-validator code + ½ of *Design Choices* section

Person C – Testing & Performance Analyst	Focuses on **experiment runs and performance measurement**	• Run algorithm for all required n-values • Measure execution time and iteration count • Collect success/failure rates over multiple runs • Create result tables and performance plots • Draft *Results & Discussion* section	CSV logs + plots + tables + 2 pages *Results & Discussion*
Person D – Visualization & Poster Designer	Builds **graphical outputs** and the **poster for demo**	• Write a visualizer for small n (ASCII or Matplotlib board) • Generate conflict heatmaps or runtime graphs from C's data • Design demo/poster showing algorithm behavior • Write *Visualization & Interpretation* paragraph	`visualizer.py` + poster + figures + short report section

Person E – Documentation & Integration Manager	Integrates code + finalizes submission	• Combine all modules into one runnable script/package • Write `README.md` (install, compile, execute) • Compile the final **PDF report** (cover sheet + sections + references) • Proofread formatting and consistency • Handle final .zip submission	`README.md` + `CP468_NQueens_TermProject.pdf` + complete code bundle

Why This Is Fair

- **All five members code:** A (Basic algorithm) + B (Board + validation) + C (Run scripts) + D (Visualization) + E (Integration scripts).
- **All five write:** A/B write *Design*, C writes *Results*, D writes *Visualization*, E writes *Intro/Conclusion + Formatting*.
- **All test:** C runs full tests, others verify sub-modules.
- **All present:** each speaks during demo (see below).

🗓️ Suggested 6-Week Timeline

Week	Focus	Lead(s)	Support
1	Research MIN-CONFLICTS, assign tasks	E	All
2	Implement algorithm & board representation	A + B	—
3	Verify correctness on small n (10, 100)	B + C	A
4	Performance testing on larger n + optimizations	C	A B
5	Visualization & poster creation	D	C E
6	Report integration + demo prep	E	All

🧠 Demo Responsibilities

Section	Speaker
Intro + overview of problem	E
Algorithm logic (MIN-CONFLICTS)	A
Data structures + solution check	B
Results & scaling analysis	C
Visualization & poster walkthrough	D

🧩 Final Deliverables Checklist

✅ `min_conflicts.py` (core algorithm)
✅ `board_utils.py` (board & validator)
✅ `tests/` (folder with results and timings)
✅ `visualizer.py` (graphical output)
✅ `README.md` (execution instructions)
✅ `CP468_NQueens_TermProject.pdf` (design document + poster)