

Parallel Hybrid Genetic Algorithm for TSP Using MPI: Design, Implementation, and Scalability Analysis

Dr. Minhua Huang
COSC6361.001, 002; Fall 2025
School of Engineering & Computer Sciences
A & M University at Corpus Christi

Points: 20
Out: Oct 5, , Due: Nov 23.

It is a group(of five students) project.

Objectives

This project aims to develop an MPI-based parallel hybrid genetic algorithm (GA) for the Traveling Salesman Problem (TSP), integrating tournament selection, Partially Mapped Crossover (PMX), inversion mutation, and 2-opt local search to optimize tour quality across 2, 4, 8, 16, and 32 processes, with a serial GA implemented as a baseline for comparison. The parallel algorithm's performance will be evaluated on TSPLIB benchmarks, specifically berlin52.tsp (52 cities), d198.tsp (198 cities), and pr439.tsp (439 cities), by measuring speedup (serial vs. parallel runtime), efficiency (speedup divided by number of processes), and solution quality (tour cost within 5% of known optima), while optimizing load balancing and minimizing communication overhead. Results will be documented in a comprehensive thesis draft with visualizations (e.g., speedup plots) and a 4–6 page workshop paper submitted to venues like ACM SRC or IEEE IPDPS workshops, highlighting scalability and solution quality.

Design and Implementation

Design

– MPI-Based Parallel Hybrid GA:

- *Island Model*: Divide population into 2, 4, 8, 16, or 32 subpopulations, each assigned to an MPI process, evolving independently.
- *Migration*: Broadcast top 5% tours every 50 generations using MPI_Bcast.
- *Load Balancing*: Dynamically adjust subpopulation sizes.
- *Operators*: Tournament selection ($k = 4$), PMX crossover, inversion mutation (probability 0.05), 2-opt local search.
- *Parameters*: Population size=200, generations=1000, crossover rate=0.8, mutation rate=0.05.

– Serial GA Baseline:

- *Representation*: Array of city indices.
- *Fitness*: Sum of Euclidean distances

$$L = \sum_{i=1}^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} + \sqrt{(x_n - x_1)^2 + (y_n - y_1)^2}$$

- *Operators*: Same as parallel GA.

– TSP Instances: Symmetric/asymmetric TSPLIB datasets (50–200 cities).

Implementation

- **Tools**: C with OpenMPI; standard C libraries.
- **Environment**: Multi-core PC (2, 4, 8, 16, or 32 cores).
- **Code Structure**:
 - `main.c`: Initialize MPI, distribute data, coordinate parallel execution.
 - `ga.c`: Serial GA baseline.
 - `parallel_ga.c`: MPI island model, migration, load balancing.

- **Key Functions with Mathematical Models:**

- `tournament_selection()`: Probability of selecting parent i :

$$P_i = \frac{f_i}{\sum_{j \in \text{tournament}} f_j}$$

where f_i is the fitness (inverse of tour length) of candidate i .

- `pmx_crossover()`: Offspring generation by mapping substrings:

$$\text{offspring} = \text{parent}_1[1..p] + \text{mapping}(\text{parent}_2[p+1..n])$$

ensuring valid TSP tour.

- `inversion_mutation()`: Reverse a subsequence $[i..j]$ with probability p_m :

$$\text{tour}'[k] = \text{tour}[j - (k - i)] \quad \text{for } k = i..j$$

- `two_opt()`: Swap edges $(i, i + 1)$ and $(j, j + 1)$ to reduce total length L :

$$\Delta L = (d_{i,j} + d_{i+1,j+1}) - (d_{i,i+1} + d_{j,j+1})$$

Apply swap if $\Delta L < 0$.

- `migrate_islands()`: Broadcast top tours using MPI_Bcast.

- **Profiling:** mpiP to measure collective communication overhead.

- **Performance Metrics:**

- **Speedup**: Ratio of serial GA runtime to parallel GA runtime.
- **Efficiency**: Speedup divided by the number of processes.
- **Solution Quality**: Tour cost compared to known TSPLIB optima (within 5% target).

- **Output**: Best tour, distance, timings, visualizations.

- **Workflow**: Weeks 1–2: Serial GA; Week 3: MPI parallelization; Week 4: Debug/optimize; Weeks 5–6: Experiments and documentation.

Report

- **Structure**: Introduction, Methodology, Results/Analysis, Conclusion.
- **Page Length**: Workshop paper: 4–6 pages; Thesis draft: 10–15 pages.
- **Content Focus**: Speedup, efficiency, solution quality, load balancing, collective communication overhead.
- **Deliverables**: Workshop paper (4–6 pages) and thesis draft (10–15 pages) for submission to ACM SRC or IPDPS workshops.
- **Timeline**: Weeks 5–6: Compile results, visualizations, write/edit report.

Oral Presentation (4 points)

- 20 slides of power point for a 20 minutes of oral presentation for each of the groups
- Demonstrate the project in front of the class
- Presentation Date:
 - Nov 24. 3 - 4 groups
 - Nov 26. 3 - 4 groups
 - Dec 1. 3 - 4 groups
 - Dec 3. 3 - 4 groups

Project Submission

Each of groups only needs to have one submission. The submission should be in the project-submission-entry of the leader on Canvas.

- ReadMe.txt – how to run your system
- Project source codes
- The screen shots of the outputs of your system
- Presentation slides
- Report