Research Diary

PhD Research Journal

Author: **PhotonZhang**

Start Date: 5 February 2026

Field: **Computer Science**

Contents

1 30 July 2025 - Literature Review 3
1.1 Research Focus 3

2 31 July 2025 - Experimental Work 3
2.1 Algorithm Implementation 3

3 1 August 2025 - Code Development 4
3.1 Algorithm Optimization 4

4 2 August 2025 - Key Insights 5
4.1 Research Breakthrough 5

5 3 August 2025 - Weekly Summary 6
5.1 Progress Review 6

6 4 August 2025 - Research Planning 7
6.1 Future Directions 7

1 30 July 2025 - Literature Review

1.1 Research Focus

Today's focus is on reviewing recent papers in production optimization and scheduling algorithms.

Paper Reading

Paper Title: Advanced Genetic Algorithms for Job Shop Scheduling [1]
Authors: Johnson, M. and Smith, A.
Main Content:

- Introduced adaptive mutation rates for genetic algorithms
- Applied to job shop scheduling problems with 100+ jobs
- Achieved 15% improvement over traditional methods
- Proposed hybrid approach combining GA with local search

Personal Thoughts: This paper provides excellent insights into adaptive parameter tuning. The hybrid approach is particularly interesting and could be applied to our manufacturing scheduling problems. The 15% improvement is significant for industrial applications.
Relevance Score: ***** (5/5)
Related Papers:

- [2] - Traditional scheduling methods
- [3] - Benchmark comparison

2 31 July 2025 - Experimental Work

2.1 Algorithm Implementation

Testing the genetic algorithm implementation on our production dataset.

Experiment Log

Experiment Name: Genetic Algorithm Performance Testing
Objective: Evaluate the performance of our genetic algorithm implementation on real manufacturing data
Experimental Setup:

- Dataset: Real production data from Factory A (500 orders, 30 machines)
- Algorithm: Custom genetic algorithm with adaptive parameters
- Parameters: Population size=100, generations=200, mutation rate=0.1
- Comparison: Against traditional FCFS (First Come First Serve)
- Metrics: Makespan, resource utilization, computational time

Results:

Method	**Makespan (hours)**	**Utilization (%)**	**Time (min)**
FCFS	120.5	65.2	0.1
Genetic Algorithm	98.3	78.9	45.2
Improvement	18.4%	21.0%	-

Key Findings:

- Genetic algorithm reduces makespan by 18.4%
- Resource utilization improved by 21.0%
- Computational time is acceptable for offline planning
- Algorithm converges within 150 generations

Issues and Thoughts: The results are promising but computational time might be too high for real-time applications. Consider implementing parallel processing or hybrid approaches for faster execution.

3 1 August 2025 - Code Development

3.1 Algorithm Optimization

Working on improving the genetic algorithm implementation.

Code Snippet

```
% Improved genetic algorithm with adaptive parameters def
adaptive_genetic_algorithm(population, max_generations=200):  """
Enhanced genetic algorithm with adaptive mutation and crossover rates
""" best_fitness = 0 best_solution = None
% Adaptive parameters mutation_rate = 0.1 crossover_rate = 0.8
for generation in range(max_generations):  % Selection using tournament
selection parents = tournament_selection(population, tournament_size=3)
% Adaptive crossover offspring = adaptive_crossover(parents,
crossover_rate)
% Adaptive mutation offspring = adaptive_mutation(offspring,
mutation_rate)
% Evaluate fitness fitness_scores = [evaluate_fitness(ind) for ind in
offspring]
% Update best solution max_fitness = max(fitness_scores) if
max_fitness > best_fitness:  best_fitness = max_fitness best_solution
= offspring[fitness_scores.index(max_fitness)]
% Adaptive parameter adjustment if generation % 20 == 0:  mutation_rate
= adjust_mutation_rate(mutation_rate, generation) crossover_rate =
adjust_crossover_rate(crossover_rate, generation)
% Population replacement population = replace_population(population,
offspring)
return best_solution, best_fitness
```

4 2 August 2025 - Key Insights

4.1 Research Breakthrough

Made an important discovery about parameter adaptation.

Important Note

Important Discovery: Adaptive parameter adjustment significantly improves algorithm performance!
Key findings:

- Dynamic mutation rate reduces premature convergence
- Adaptive crossover rate improves solution diversity
- Parameter adjustment every 20 generations is optimal
- Performance improvement: 25% better than fixed parameters

Next Steps:

- Implement this approach in our main algorithm
- Test on larger datasets
- Prepare paper for conference submission
- Consider patent application for the adaptive method

5 3 August 2025 - Weekly Summary

5.1 Progress Review

Comprehensive review of this week's research progress.

Daily Summary

This Week's Achievements:

- Completed literature review of 15 papers on genetic algorithms
- Implemented and tested adaptive genetic algorithm
- Achieved 25% performance improvement over baseline
- Identified key parameters for industrial applications
- Prepared draft for conference paper

Next Week's Plan:

- Implement parallel processing for faster execution
- Test algorithm on multiple factory datasets
- Compare with other optimization methods (PSO, SA)
- Write detailed methodology section
- Prepare presentation for research group meeting

Challenges Encountered:

- Computational time still too high for real-time use
- Need more diverse test datasets
- Parameter tuning requires more systematic approach

Inspiration Notes: The adaptive parameter approach could be a novel contribution to the field. Consider extending this to other optimization algorithms. The industrial application potential is very high - could lead to significant cost savings in manufacturing.

6 4 August 2025 - Research Planning

6.1 Future Directions

Planning next phase of research.

Paper Reading

Paper Title: Machine Learning in Production Optimization [4]
Authors: Chen, L. and Wang, H.
Main Content:

- Applied reinforcement learning to production scheduling
- Used neural networks for parameter prediction
- Achieved 30% improvement over traditional methods
- Real-time adaptation to changing conditions

Personal Thoughts: This paper opens up exciting possibilities for our research. The combination of ML and optimization could be the next breakthrough. We should explore this direction while maintaining our focus on industrial applications.
Relevance Score: ***** (5/5)
Research Ideas:

- Combine our adaptive GA with ML parameter prediction
- Use ML for real-time parameter adjustment
- Apply to dynamic production environments

References

[1] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778.

[2] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[4] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).