Accelerating a local search algorithm for large instances of the independent task scheduling problem with the GPU

Frederic Pinel Johnatan Pecero Pascal Bouvry

Computer Science and Communications
University of Luxembourg

GreenDays@Paris

UNIVERSITÉ DU LUXEMBOURG

Outline

Motivation

Initial algorithm

Adaptation

Large instances

Conclusion

Motivation

- Independent tasks
- Makespan, combines several perspectives:
  - User: flowtime
  - Provider: load balance, energy (low machine heterogeneity)

Parallel CGA

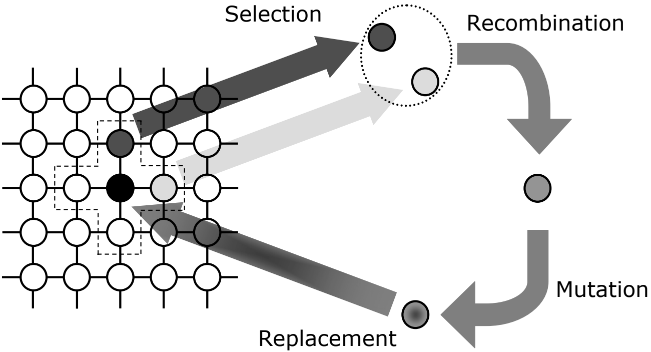
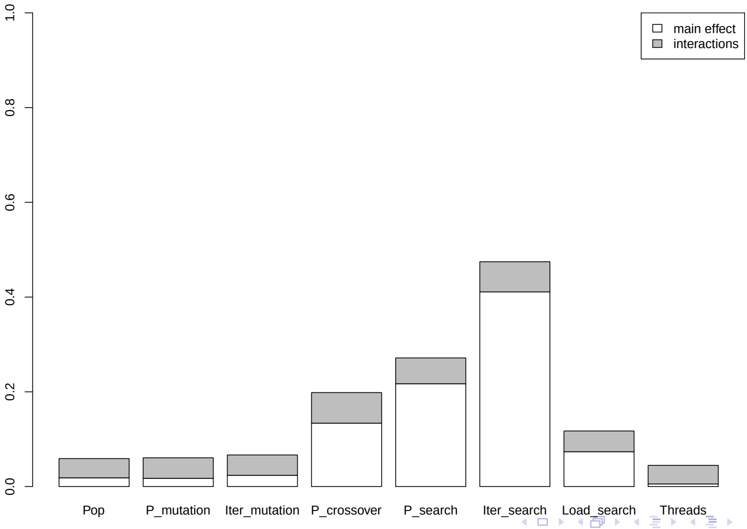


Figure: Generating solution

Parallel CGA

- Parallel asynchronous cellular genetic algorithm
- Initialized with heuristic (Min-Min)
- Local search

Feedback


1.0
0.8
0.6
0.4
0.2
0.0
□ main effect
▨ interactions
Pop
P_mutation
Iter_mutation
P_crossover
P_search
Iter_search
Load_search
Threads

Adaptation

- Simplified algorithm
- Min-Min, incremental formulation
- Increased local search, complete-state formulation

Results

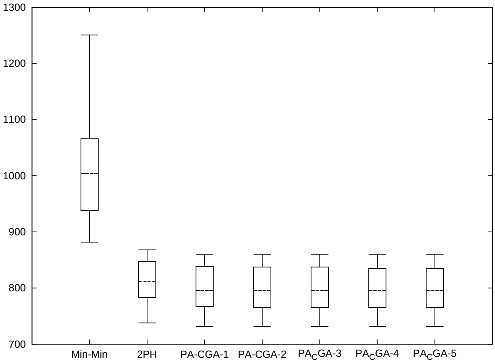


Figure: Consistent, high-h. tasks, low-h. machines

Results

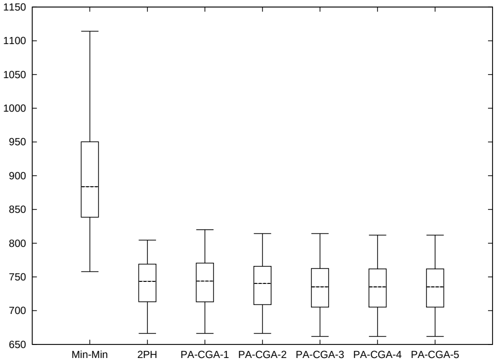


Figure: Semi-consistent, high-h. tasks, low-h. machines

Results

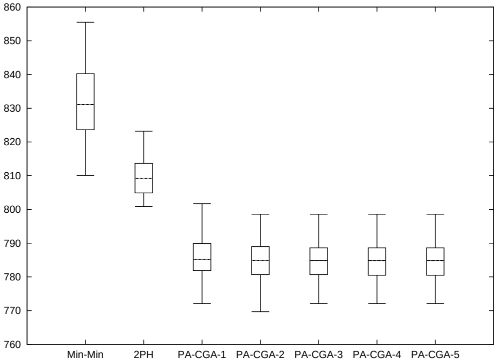


Figure: Consistent, low-h. tasks, low-h. machines

Results

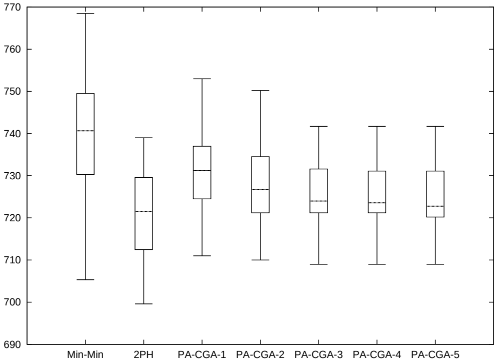
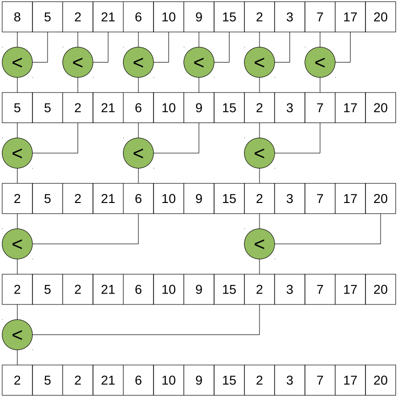


Figure: Semi-consistent, low-h. tasks, low-h. machines

Min-Min on GPU



8 5 2 21 6 10 9 15 2 3 7 17 20

5 5 2 21 6 10 9 15 2 3 7 17 20

2 5 2 21 6 10 9 15 2 3 7 17 20

2 5 2 21 6 10 9 15 2 3 7 17 20

2 5 2 21 6 10 9 15 2 3 7 17 20

Figure: Parallel reduction in Min-Min

Min-Min runtime

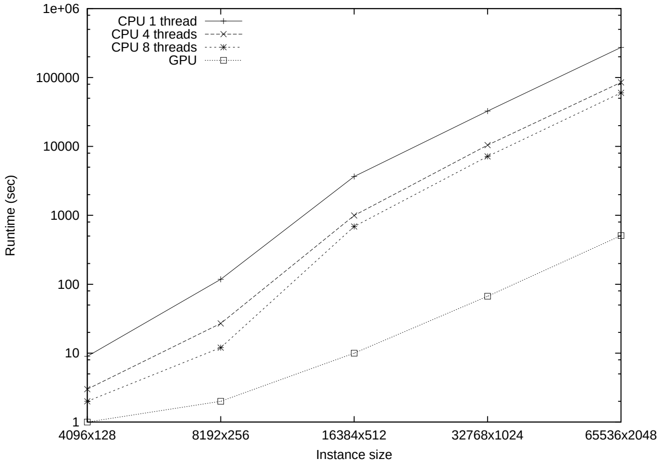


Figure: Runtime Min-Min

Performance

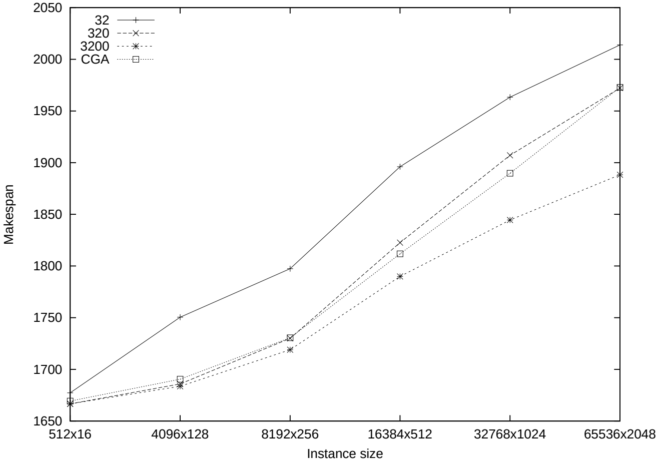


Figure: Makespan

Performance

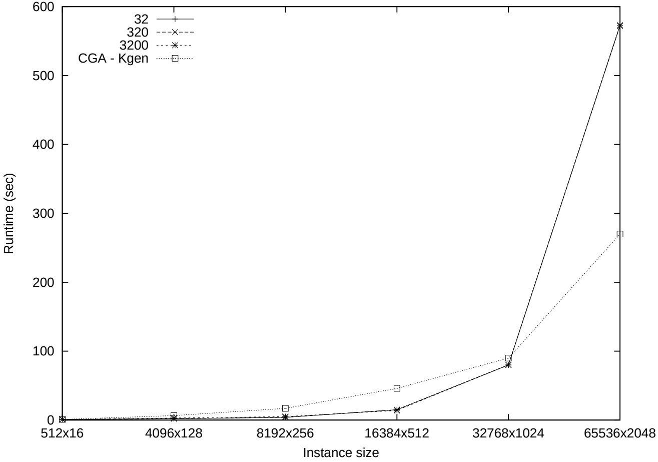


Figure: Runtime

Conclusion

- Failure
- Solution → Feedback → loop
- Learning process

Machine learning opportunities

- Learn on problem instance
  - Task profiling
  - Co-scheduling
- Learn allocation rules
  - Adapt (parameters, heuristics)
  - Algorithm (oracle: solved instances)

Questions

Thank you.