A Comparison Between a GPU-Based Method for Combinatorial Optimization and the Fixstars Amplify Annealing Engine

Kojiro Noborio,
Noriyki Fujimoto

Background and Objectives

- Combinatorial optimization problems, such as QUBO and QAP, play a critical role in many real-world applications, including logistics, scheduling, and circuit layout
- As problem sizes increase, achieving sufficient performance with conventional sequential computing becomes increasingly difficult
- Recently, specialized hardware platforms such as Ising machines and quantum annealers have attracted significant attention
- Meanwhile, the massive parallelism of general-purpose GPUs offers the potential to achieve high performance without relying on specialized hardware
- The objective of this study is to **develop a fast and accurate combinatorial optimization solver using general-purpose GPUs** and to compare its performance with the Fixstars Amplify Annealing Engine (AE)

The Proposed Method

- The proposed solver is based on **Simulated Annealing (SA)**
- SA enables escape from local optima by probabilistically accepting worse solutions
- In this study, SA is designed for GPU execution based on the following principles
- A **Specified Time** parameter is introduced to explicitly control the program execution time
  - The temperature schedule is automatically adjusted according to the specified execution time
- To fully exploit GPU parallelism, **multiple SA replicas are executed concurrently**
  - All replicas share a **common temperature** during the search process
- For neighborhood search, the **simplest 2-opt operation** is employed for both TSP and QAP instances

Experimental Setup

- **Fixstars Amplify Annealing Engine (AE)** is adopted as the baseline solver
- **Official benchmark statistics publicly released by Fixstars** are used for comparison
  （https://amplify.fixstars.com/benchmark/#/clients/FixstarsClient）
- To ensure fair comparison, the **GPU is fixed to NVIDIA A100** across all experiments
- Each benchmark instance is evaluated using **20 or 100 independent runs**
  - The number of runs is **matched to the evaluation conditions reported by Fixstars**

Future Works

- Application to a broader range of combinatorial optimization problems and benchmark instances
- Systematic comparison with optimization solvers developed by other companies and research institutions
- Further improvement and acceleration of both the algorithm and GPU implementation
- Extension to large-scale problems motivated by real-world applications
- We are currently developing a solver that **solves combinatorial optimization problems by specifying only the problem type, instance, and execution time**
- The solver is designed to be **usable without any GPU programming expertise**, requiring only a compatible GPU and its driver
- The range of supported problem classes is being continuously expanded toward practical deployment

Results and Discussion

- Under identical execution-time constraints, the proposed method achieves **higher-accuracy solutions than Amplify AE for larger problem instances**
- For the QAP benchmark *sk056*, a **943× speedup in TTS(1%)** is observed
- The performance gap between the proposed method and Amplify AE **becomes more pronounced as the problem size increases**
- The convergence behavior is found to depend on the **problem type and the implementation of neighborhood search**
  - In particular, **TSP exhibits rapid convergence in the later stages of the search compared to QAP**

Table 1: Comparison of the Proposed GPU Solver with Fixstars Amplify AE on TSP Instances

Instance	Specified Time[s]	Amplify AE		Proposed GPU Solver	
		%best err	TTS[ms]	%best err	TTS[ms]
burma14	1	0	78(0%)	0	372(0%)
ulysses16	3	0	168(0%)	0	1447(0%)
bayg29	3	0	151(0%)	0	2920(0%)
eil51	30	0	3034(0%)	0	30464(1%)
kroA100	120	0	60234(1%)	0.36	116489(2%)
pr124	1000	0.87	33285129(1%)	0.06	984567(1%)
pr136	1000	1.39	675340(5%)	0.61	1020789(1%)
pr144	1000	1.81	705390(5%)	0.10	971038(1%)
si175	1000	0.63	70149275(1%)	0.12	883881(1%)
pr299	1000	41.2	15243078(50%)	1.35	991149(5%)

Table 2: Comparison of the Proposed GPU Solver with Fixstars Amplify AE on QAP Instances

Instance	Specified Time[s]	Amplify AE		Proposed GPU Solver	
		%best err	TTS[ms]	%best err	TTS[ms]
chr12a	1	0	73.50(0%)	0	685(0%)
chr18a	1	0	127.0(0%)	0	1605(1%)
esc32a	30	0	1526(0%)	0	537(5%)
sko56	30	0.39	839380(1%)	0.01	889(1%)
sko100a	120	1.40	68994(5%)	0.08	1666(1%)

Notes
- **err** denotes the relative error, where **err = 0** indicates that the optimal solution has been reached
- **TTS (Time to Solution)** represents the time required to reach the specified relative error with **99% probability**, assuming that the time to solution follows a **normal distribution**

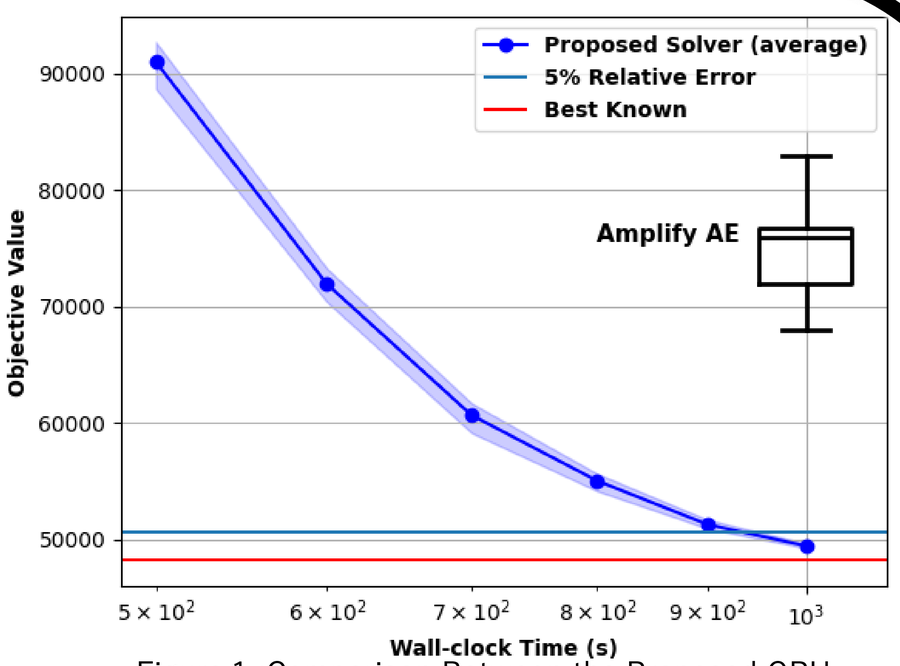

Figure.1 :Comparison Between the Proposed GPU Solver and Fixstars Amplify AE on “pr299”

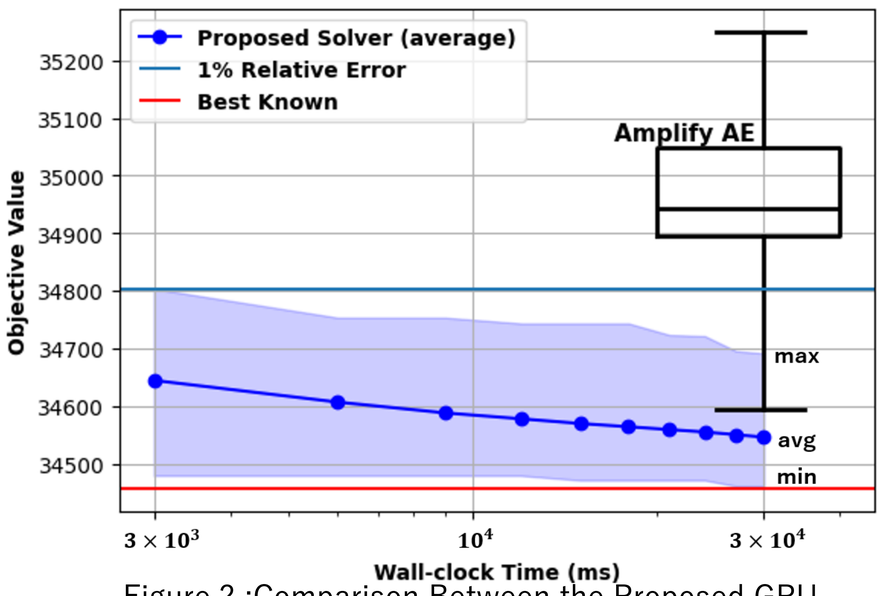

Figure.2 :Comparison Between the Proposed GPU Solver and Fixstars Amplify AE on “sko56”