INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
SHORT ABSTRACT OF THESIS

Name of the Student : **Manoj Kumar**

Roll Number : **146101013**

Programme of Study : **Ph.D.**

Thesis Title:

Efficient Parallelization and Performance Analysis of Meta-heuristics on Many-core Platforms

Name of Thesis Supervisor(s) : **Dr. Pinaki Mitra**

Thesis Submitted to the Department/ Center : **CSE**

Date of completion of Thesis Viva-Voce Exam : **08 June, 2023**

Key words for description of Thesis Work : **Meta-heuristics, Multi-Core Architecture, GPU, CUDA**

SHORT ABSTRACT

Meta-heuristics are an efficient method for solving complex problems in science, engineering, and industry. They explore the solution space efficiently to generate a good solution in a reasonable time through a neighborhood or population-based local search. Even if the meta-heuristics do it efficiently, for large instances (practical problems of science, engineering, or industry), generation of neighborhood and evaluation of solution of single-solution based meta-heuristics or population-based meta-heuristics takes a tremendous amount of time.

We used this highly parallel meta-heuristics solver to run on modern days massively parallel Graphics Processing Unit (GPU) to reduce the execution further. In this work, we scheduled and mapped the application to take advantage of the GPU's architectural configuration to run efficiently by taking advantage of the management of local memory, shared memory, and global memory of the accelerator. Many GPU programmers use most coding styles or mapping strategies that are adhoc based. We automated this mapping or coding approach so that the accelerators are adequately used and improve the program's performance.

In the later phase, we analyzed the performance of meta-heuristics on GPU by using static and dynamic auto-tuning of application models on target architecture for better performance. In this study, we evaluated the performance of various meta-heuristics for the quadratic assignment problem (QAP), traveling salesman problem (TSP), and permuted perceptron problem (PPP) using a massively parallel machine such as a GPU. These meta-heuristics included iterated local search (ILS), simulated annealing (SA), genetic algorithm (GA), tabu search (TS), particle swarm optimization (PSO), and crow search algorithm (CSA). We achieved the highest speedup, 127:56 on GPU using PSO for QAP, speedup 50:53 using CSA for TSP, and speedup 165:62 using SA for PPP.