Brian Rodriguez
U0853593

Quiz 6

The paper aims to simplify the implementation of Iterative local searches on parallel computing systems by introducing a new framework called iterative local champion search. The framework handles the nitty gritty details of creating a parallelized program. It allows the user to create template functions that are used when computing the optimal value in an iterative local search.

The template functions are the key part in using the framework. Depending on the hardware being used, the user needs to develop a serial CPU implementation, Single-GPU CUDA implementation or both. The functions are used to initialize the framework and also do the computations while running the algorithm. ILCS requires that the user also create a data-structure in which the champion can be represented. A champion is a representation for the most optimal value that has been found thus far.

The templated functions that are required are CPU/GPU_init, CPU/GPU_Exec and CPU/GPU_output. These are the functions that the user needs to implement in order to use the framework correctly. The init function essentially needs to be called by all of the cores that are being used for work and it returns the size of the data-structure being used as a champion. The execute function is repeatedly invoked by all of the threads, it is invoked with different seeds to do the work. Exec generates a solution and then improves it until a local optimum is reached and this solution is returned as a champion. The framework keeps track of the champion by inspecting the “quality” of the solution. This “quality” is a long type that is required to be inside of the data-structure. The output function is called on a periodic basis and returns the current champion.

ILCS provides a wrapper of implementing an iterative local search in parallel with the option of using CPUs and GPUs. This framework is simple, yet effective and gives the user as much power as they need but without completely allowing them to touch the inter-workings of creating tasks and threads.