# Project brief
# Combinatorial optimization of NP-hard problems using dynamic GPU programming

Kim Svensson ks6g10
Supervised by Dr. Sarvapali D. Ramchurn sdr

October 12, 2012

## Problem

With NP-hard problems and algorithms with a complexity class of P or greater, the nature of such problems allows only for a small change in sample size if the CPU execution speed double as execution speed and sample size of a algorithm is mutually exclusive. As such, computer scientists have to find other ways of computing difficult problems on large sets of data. The traditional approach have been to increase the CPU speed and the number of CPUs. However, such implementation is often costly and suffers from latency and architectural difficulties. The Graphical processing unit (GPU) have been commonly used to generate graphics for the user to see. However as the GPU is of a single instruction multiple data (SIMD) architecture with hundreds to thousands of programmable cores, researchers have been able to utilize GPUs to run concurrent algorithms faster than the equivalent more sequential algorithm bound to the CPU. This advance in computer science can allow for large computations previously unfeasible on the CPU by leveraging the parallel nature of the GPU.

## Goals

In order to measure any performance gain from using a GPU as an execution unit, a comparable CPU bound algorithm have to be implemented as a baseline. In order to achieve this the implementation will be designed after already known dynamic algorithms for said problem. Further on, a sequential algorithm is not suited for GPU so major parallelisation within bounds of the GPU have to be performed in order to leverage the GPU's power. During which time knowledge of concurrent programming and the GPU limitations will be explored and evaluated to determine the best suited methods. Last, the end goal is to develop a concurrent algorithm for the GPU which substantially lowers the time complexity of the problem evaluated.

## Scope

The scope of which problem to implement is set to be the combinatorial auction problem, however it may be subject to change. It should give an understanding and implementation of said problem running on a GPU, where you can draw conclusions on whether the implementation allows for a significant shorter run-time. If possible, the code for the GPU will be further optimized and refined to allow for an even shorter execution time. If time allows, a graphical representation application will be developed where it is applicable to enable a visualisation of problems and its paths depending on the input parameters and internal restrictions.