Combinatorial Problem Solving and Optimization through Diffusion Model Based P-metaheuristics

Andreas Robert Brünner
Georg Simon Ohm University of Applied Sciences Nuremberg, Germany

Abstract

Combinatorial problem solving and optimization play a key role in tackling major problem domains in science and industry, among them genome sequencing, planet finding, logistics and VLSI. They require extensive computational power and a high memory throughput due to the complexity of the problem domains, which are mostly NP-complete/NP-hard, multi-dimensional or multimodal.

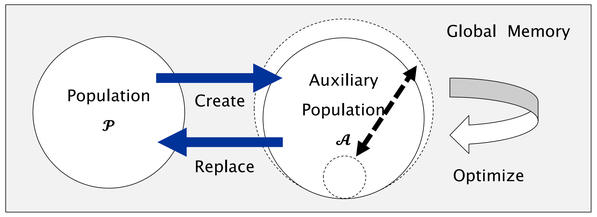
Those requirements can be currently best met by modern GPUs, which are not only able to offer a higher instruction but also a higher memory throughput than most recent CPUs. GPUs therefore present themselves as ideal candidates for satisfying the high demands on computational hardware.

A generic library was designed and implemented, which exploits the superior capabilities of GPUs for combinatorial problem solving and optimization. The library hereby employs a diffusion model based P-metaheuristic called cellular genetic algorithms (cGAs), that provides optimal or near optimal solutions and can be mapped efficiently to the architecture of modern GPUs.

Motivation

Many combinatorial solvers that execute on CPUs can increase their performance dramatically by employing the massive-parallel capabilities of modern GPUs. This is especially true for combinatorial solvers that employ population based approximate methods like cGAs.

CGAs are a P-metaheuristic that maintains a divers one-, two-, or three-dimensional set of potential solutions to a problem, in order to improve upon it iteratively. Potential solutions are hereby constantly transferred between a main and an auxiliary population, which further increases the demands on memory throughput.



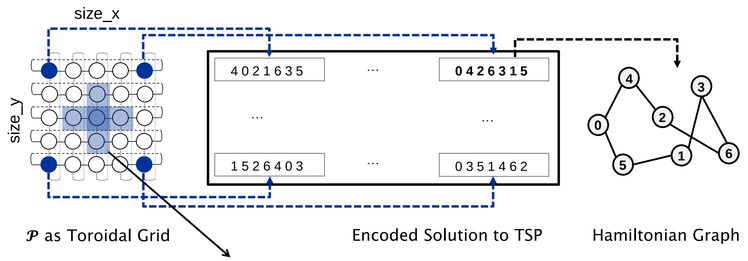
The challenge of employing cGAs as combinatorial solvers lies in their mapping to the architecture of GPUs. This mapping is non trivial due to the manifold features of modern GPUs, which provide various types of on-chip and on-board memory, varying greatly with respect to bandwidth and capacity. A mapping therefore requires the efficient use of the given memory hierarchy, exploitation of data locality and reuse of data as much as possible.

Objective Target

The objective target is to design and implement a generic library to illustrate the mapping of the concept of cGAs to modern GPU architectures and to evaluate the effectiveness of cGAs in the context of combinatorial optimization. **A major focal point during design and implementation of this mostly memory-bound optimization method is the exploitation of the GPU's superior memory bandwidth**.

The 2D Euclidean TSP in the context of VLSI serves as a baseline for the evaluation of the library's effectiveness and the achieved hardware exploitation. A subset of the TSPLIB from Bonn University hereby provides necessary test sets.

Concept



$$\mathcal{N}^{vN}_{(x_c,y_c)}(x,y) = \{(x,y): |x - x_c| + |y - y_c| \leq r\},$$
where r is the radius of the neighborhood and (x_c, y_c) denotes the center cell of the neighborhood

Optimization operations of cGAs are restricted to neighborhoods. This principle of spatial isolation provides an effective mechanism for balancing the explorative and exploitive properties of an optimization and allows for a concurrent optimization on a per neighborhood basis. The overlapping of neighborhoods provide the diffusion of information through the population, which is necessary to improve solutions cooperatively towards a global optimum.

Course of Action

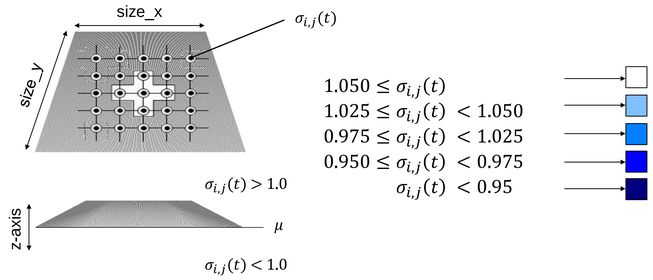
The concept of cGAs was first analyzed and then mapped to the architecture of modern GPUs by taking NVIDIA's Compute Unified Device Architecture into account. Problem domain agnostic cGA components were hereby separated from problem domain dependent ones to allow for an easy extension towards other problem domains. Cell selection was designed and implemented so that virtually all cell update szenarios are feasible. Self-configuring operators were introduced that deduce necessary execution parameters from concrete optimization workloads and thus hide the execution model of GPUs. Crossover operators were classified and an efficient implementation with respect to the execution model of GPUs was proposed. A class of crossover operators hereby particularly benefits from fast on-chip memory, leading to an additional 4x speed-up.

Results

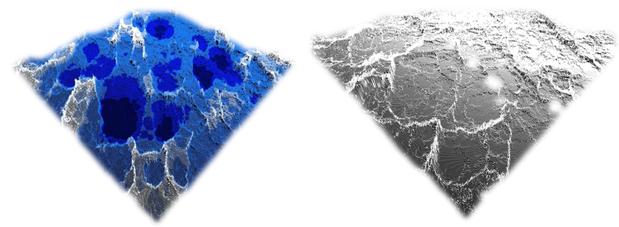
- The library solved the two VLSI test sets "xqf131" with 131 and "xqg237" with 237 2D-coordinates from TSPLIB with 0.43% and 1.94% above optimality in approximately two and four minutes respectively using a population of 256² solution vectors.
- The library currently achieves an effective memory throughput on a GeForce GTX 680 that matches or exceeds the theoretical bandwidth of most recent CPUs.
- The overall high occupancy rate of ~0.8 and higher in addition to a minimal branch divergence rate furthermore suggest that the concept of CGAs can be efficiently mapped to GPUs and therefore provide an effective and efficient method to tackle large and complex combinatorial optimization problems.

Visualization

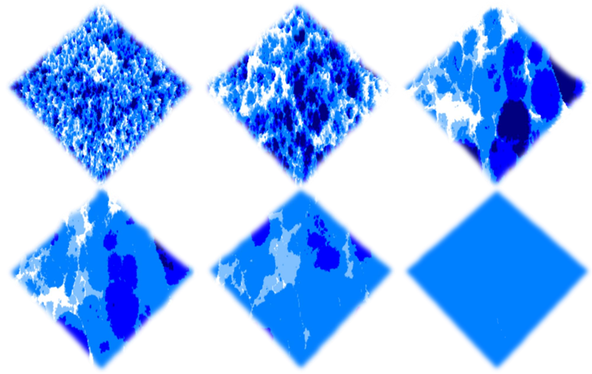
A visualization component of the library maps the relative solution quality of a two-dimensional cellular population to a three-dimensional mesh with each vertex in the mesh representing the relative quality of a single solution candidate. This component shows the optimization progress in real-time, which in turn allows to determine the convergence behaviour of concrete optimization instances and thus helps to setup effective parameter sets.



Valleys of the resulting solution quality landscape depict better solutions to a problem, which in context of the 2D Euclidean TSP are smaller routes. Each dark blue valley hereby represents at least one local/global optimum.



The landscape depicts the convergence of the search with a continually decreasing diversity of solutions until finally one best solution is determined.



Georg Simon Ohm University of Applied Sciences
Nuremberg, Germany

Special thanks for support and encouragement:
Prof. Christoph von Praun
Prof. Florian Gallwitz