Parallel Solver for Vehicle Routing Problem

A status/idea report

2010

Parallel Solver - Ambitions

Why parallelize?

- Faster computation times
- Better solutions
- More robust solutions

Where parallelize?

- Algorithmic level, metaheuristics
- Iteration level, neighborhood evaluation (generation)
- Solution level
- ...?

Parallel Solver - Ambitions cont'd

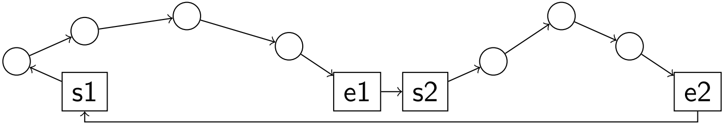
Platforms:

- Use of GPU: Large number of computational units, ideally single instruction multiple data
- Multicore, thread-based parallelism: Relatively few computational units, high level of independence, shared memory or communication
- Multiprocessor similar to multicore
- ...?

Model

Used model based on paper "A Unified Modeling and Solution Framework for Vehicle Routing and Local Search-based Metaheuristics" by Stefan Irnich

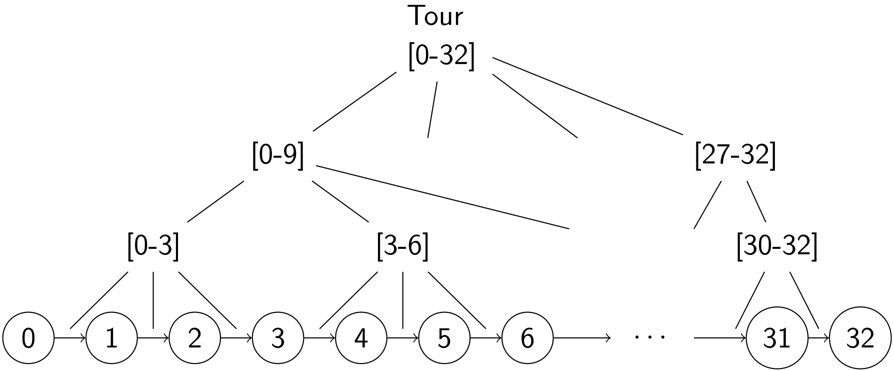
- Solution represented as a giant route or giant tour



- At each node i: interval $[a_i, b_i]$ given, in which resource must lie
- For each allowed path $i \rightarrow j$ a resource extension function (REF) $f_{i,j}$ given
- Resource feasible path: Can find T_i such that for all nodes $f_{i,j}(T_i) \leq T_j$
- For now: Only classical REFs: $f_{i,j}(T) = \max(a_j, T + t_{i,j})$

Segment - Hierarchy

Why classical REF? Simple, can build segment hierarchy



Aggregation:

[3-6] contains: 3 → 5, 3 → 6 and 4 → 6 and maybe inverse

[0-9] contains: 0 → 6, 0 → 9 and 3 → 9 and maybe inverse

Segment - Hierarchy cont'd

- Why segment hierarchy? Gives constant time feasibility check

  Example: Exchange two nodes, e.g. 5 and 20:

path up to first:	$0 \rightarrow 4$:	$0 \rightarrow 3$, $3 \rightarrow 4$
reconnect first:	$4 \rightarrow 20$:	
	$20 \rightarrow 6$:	
path to second:	$6 \rightarrow 19$:	$6 \rightarrow 9$, $9 \rightarrow 18$, $18 \rightarrow 19$
reconnect second:	$19 \rightarrow 5$:	
	$5 \rightarrow 21$:	
path to end:	$21 \rightarrow 32$:	$21 \rightarrow 27$, $27 \rightarrow 32$

- Maximum number of segments in one path: 2l-1 (l: depth of hierarchy)
- How to do feasablity check with segments, see paper(s) by Irnich
- Effort to create hierarchy: $O(n^{2^l/(2^l-1)})$

So far

Working currently on a CPU implementation to have reference CPU version

Next goal: Implement neighbourhood evaluation on GPU
Probable challanges:

- Memory management
- Organize evaluation and hierarchy traversal in such a way that each thread in a warp follows same execution pattern

Thank you for your attention!