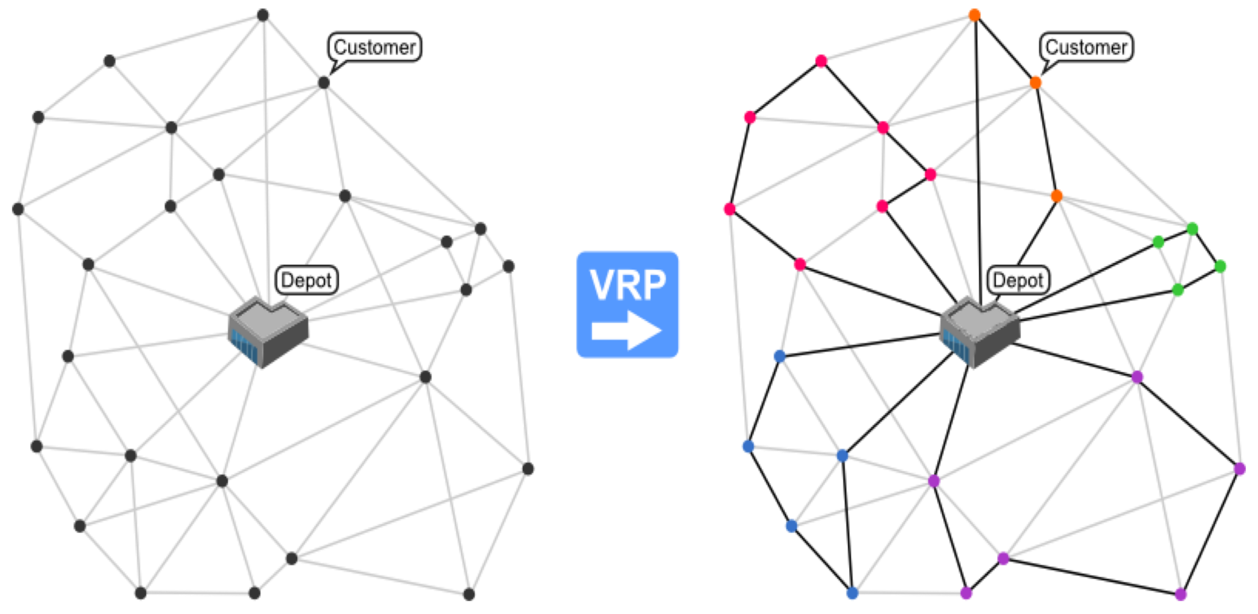
Scalable Branch-and-Price Implementation for the CVRP with SCoOL

CSE603: Parallel and Distributed Processing

Demetrios V. Papazaharias, Dr. Jaroslaw Zola

Introduction

SCoOL is a high-level programming model for solving combinatorial optimization problems in shared and distributed memory. In this work, we develop a branch and price algorithm with SCoOL to solve the capacitated vehicle routing problem.



Methods

Branch-and-price is a method for solving integer programs with many variables; it combines the algorithms of branch-and-bound and column generation.

Branch-and-Price

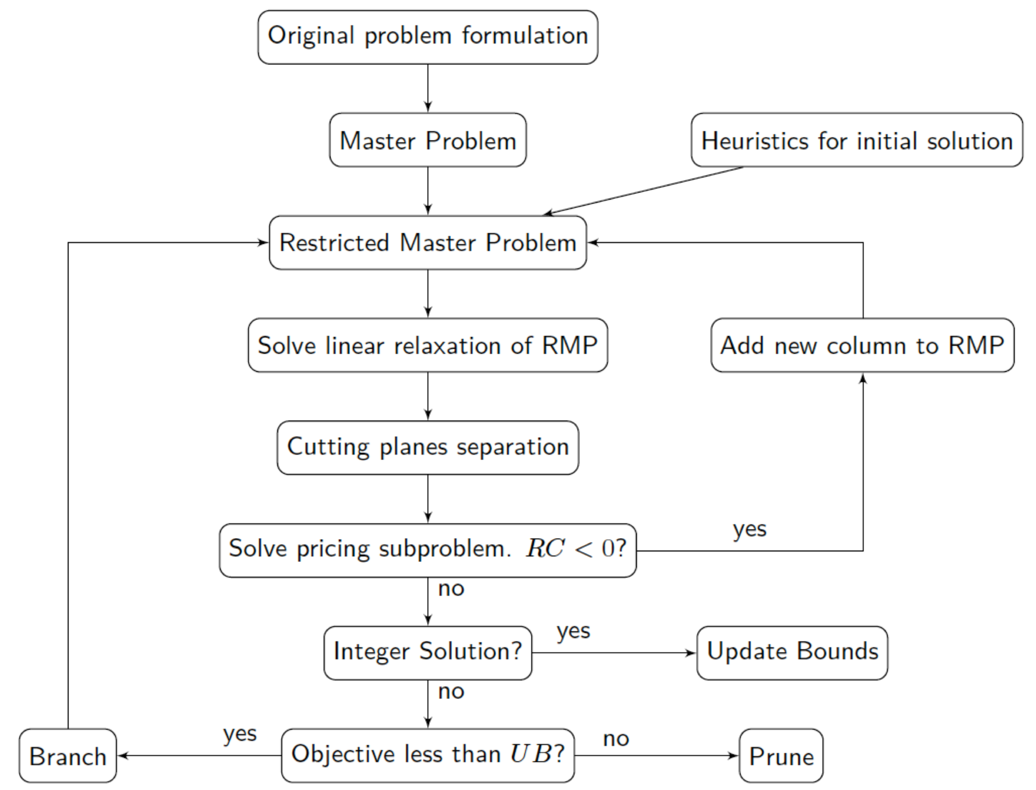
- **Branch-and-bound**: divide-and-conquer technique for IP
- **Column Generation**: efficient algorithm for very large LP

SCoOL

SCoOL operates off a user-defined ***state*** and ***task***, and handles the parallel assignment and execution

- **State:** global information for solver → **best solution**
- **Task:** how to process a point in the search space → **CG**

Branch-and-Price Algorithm



Column Generation

Restricted Master Problem – Set Partitioning

λ_r $-$ *1 if route r is selected, 0 otherwise*

a_i^r $-$ *1 if customer i is on route r, 0 otherwise*

c_r $-$ *total cost of route r*

Subproblem - Resource Constrained Shortest Path

x_{ij} $-$ *1 if arc (i, j) is selected, 0 otherwise*

π_i $-$ *dual variable of customer i from RMP*

c_{ij} $-$ *cost of arc (i, j)*

d_i $-$ *demand of customer i*

Q $-$ *capacity of the vehicle*

π

Restricted Master Problem

$$\begin{aligned} \min \quad & \sum_{r\in\Omega} c_r\lambda_r \\ \text{s.t.} \quad & \sum_{r\in\Omega} \lambda_r \le K; \\ & \sum_{r\in\Omega} a_i^r\lambda_r = 1, \quad i \in V_+; \\ & \lambda_r \ge 0, \quad r \in \Omega; \end{aligned}$$

Pricing Subproblem

$$\begin{aligned} \min \quad & \sum_{(i,j)\in A} (c_{ij} - \pi_i)\cdot x_{ij} \\ \text{s.t.} \quad & x \in \mathcal{X} \\ & \sum_{(i,j)\in A} d_i x_{ij} \le Q \\ & \sum_{(i,j)\in S\times S} x_{ij} \le |S| - 1 \quad S \subset V \setminus \{0\}, |S| \ge 2 \\ & x_{ij} \in \{0,1\} \quad (i,j) \in A \end{aligned}$$

λ

Implementation Details

We utilize a computing cluster from the Center of Computational Research (CCR) at UB. In particular, we test our results for shared memory.

- **Hardware:** Each node contains a 12-core Intel Xeon E5645 v3 2.4 GHz processor, 48 GB of RAM, and running Linux x64_64, CentOS 7.2
- **Software:** The restricted master problem and pricing subproblem were implemented using the Gurobi 9.1.1 API for C++
- **Parameters:** Using Gurobi's solution pool feature, we retain 10 best routes generated by the subproblem
- **Initialization:** In order to generate an initial set of columns we use a split tour construction heuristic and local search methods
- **Branching Rules:** Rather than branch on route variables λ_r, we select to branch on most fractional arc variable x_{ij}
- **Early Branching:** In order to reduce ramp up time to enable parallelism, we branch at suboptimal solutions at earlier levels of the branch-and-bound tree

Results

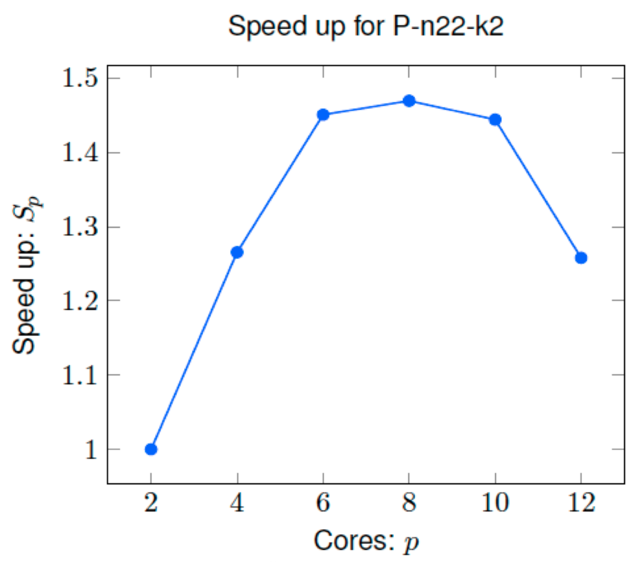
We tested several instances from the CVRPLIB, a library of benchmark instances from over 60 years of vehicle routing research

Instance	n	k	Optimal	Best Solution	Runtime (s)	Root time (s)	Columns	Task Depth
P-n16-k8	16	8	450	450	2.432	1.237	106	4
P-n19-k2	19	2	212	212	67.832	36.032	606	2
P-n22-k2	22	2	216	217*	196.658	76.35	638	3
P-n22-k8	22	8	603	601*	9.346	9.346	315	0
P-n23-k8	23	8	529	531*	9.737	9.737	536	0
P-n40-k5	40	5	458	461	7200	502.359	-	8
P-n50-k10	50	10	696	-	7200	502.935	-	9
P-n55-k10	55	10	694	-	7200	2146.77	-	5
P-n55-k15	55	15	989	-	7200	257.889	-	3
P-n60-k10	60	10	744	-	7200	418.92	-	9

Shared memory results with 12 cores; Gurobi using all cores

Results (cont.)

p	n	k	Runtime (s)	Root time (s)
2	22	8	594.177	0.759
4	22	2	469.555	0.732
6	22	2	409.601	0.742
8	22	2	404.371	0.740
10	22	2	411.454	0.812
12	22	2	472.365	0.732



Strong scaling test for shared memory for instance P-n22-k2; Gurobi using 2 cores

Discussion and Future Work

In this work we develop a parallel branch-and-price solver for the capacitated vehicle routing problem using the the SCoOL framework. While this implementation cannot beat the state of the art there are several improvements that can be made in future work:

- **Cutting Planes:** Iterative refinement of the restricted master problem to improve the quality of the linear relaxation. Also known as branch-cut-and-price
- **Stabilization:** Utilize "good" dual solutions in order to speed up the convergence of the CG algorithm
- **Pricing Subproblem:** The RSCP is an NP-Hard problem in itself; solve by dynamic programming

References

1. LLC Gurobi Optimization. Gurobi optimizer reference manual, 2021.
2. The blog of Logistics at MGEPS at UBV
3. Christian Prins, Nacima Labadi, and Mohamed Reghioui. Tour splitting algorithms for vehicle routing problems. International Journal of Production Research, 47(2):507–535, 2009.
4. Eduardo Uchoa, Diego Pecin, Artur Pessoa, Marcus Poggi, Thibaut Vidal, and Anand Subramanian. New benchmark instances for the capacitated vehicle routing problem. European Journal of Operational Research, 257(3):845–858, 2017.
5. Olivier Du Merle, Daniel Villeneuve, Jacques Desrosiers, and Pierre Hansen. Stabilized column generation. Discrete Mathematics, 194(1-3):229–237, 1999.

Industrial & Systems Engineering; Computer Science
School of Engineering & Applied Sciences
dvpapaza@buffalo.edu