Local Area Subset Row Inequalities for Efficient Exact Vehicle Routing

Udayan Mandal [1,2], Amelia Regan[2, 3], Julian Yarkony [4]
[1]Stanford University, Palo Alto, CA
[2]University of California, Irvine, CA
[3]University of Washington, Seattle, WA
[4] Laminaar Optimization Research Group, La Jolla, CA

September 2022

Abstract

In this research we consider an approach for improving the efficiency and tightness of column generation (CG) methods for solving vehicle routing problems. Our approach can be immediately extended to CG methods for problems in other areas in logistics, as well as problems with similar structure in other application areas. This work builds upon recent work on Local Area (LA) routes. LA routes rely on pre-computing (prior to any call to pricing during CG) the lowest cost elementary sub-route (called an LA arc) for each tuple consisting of the following: **(1)** a customer to begin the LA arc, **(2)** a customer to end the LA arc, which is far from the first customer, **(3)** a small set of intermediate customers nearby the first customer. LA routes are constructed by concatenating LA arcs where the final customer in a given LA arc is the first customer in the subsequent LA arc. A Decremental State Space Relaxation (DSSR) method is used to construct the lowest reduced cost elementary route during the pricing step of CG. We demonstrate that LA route based solvers can be used to efficiently tighten the standard set cover vehicle routing relaxation using a variant of subset row inequalities (SRI). SRI are a class of valid inequalities that can significantly tighten the standard set cover linear programming (LP) relaxation for vehicle routing problems such as the Capacitated Vehicle Routing Problem (CVRP). However, SRI are difficult to use in practice as they alter the structure of the pricing problem in a manner that makes pricing difficult. SRI in their simplest form state that the number of routes servicing two or three members of a given set of three customers can not exceed one. We introduce LA-SRI, which in their simplest form state that the number of LA arcs (in routes in the solution) including two or more members of a set of three customers (excluding the final customer of the arc) cannot exceed one. Higher order generalizations of SRI also exist and can be applied to LA-SRI. We exploit the structure of LA arcs inside a Graph Generation based formulation to accelerate convergence of CG. We apply our LA-SRI to CVRP and demonstrate that we tighten the LP relaxation, often making it equal to the optimal integer solution, and solve the LP efficiently without altering the structure of the pricing problem.

1 Introduction

In this research we introduce a new class of valid inequalities called Local Area-subset row inequalities (LA-SRI), which serve as a component in column generation (CG) solutions (Barnhart et al. 1996, Gilmore and Gomory 1961) to vehicle routing problems (VRP) (Desrochers et al. 1992, Feillet 2010). While our approach can be applied to more general vehicle routing problems, we describe and conduct experiments using the Capacitated Vehicle Routing Problem (CVRP), which we define as follows.
CVRP is associated with the following terms: **(1)** a depot located in space, **(2)** a set of customers with integer demands located in space, **(3)** and a set of homogeneous vehicles with integer capacity. Vehicles are assigned to routes, where each route satisfies the following:
(1) each route starts and ends at the depot, **(2)** the total demand of the customers on a route does not exceed the capacity of a vehicle, **(3)** the cost of the route is the total distance traveled. The CVRP problem selects a set of routes with the goal of minimizing the total distance traveled while ensuring that each customer is serviced at least once (though an optimal solution services each customer exactly once).

CVRP and other vehicle routing problems can be solved using with compact linear programming (LP) relaxations or expanded LP relaxations (Desrochers et al. 1992) (where the expanded LP relaxation is often referred to as the set cover formulation). Compact LP relaxations have a variable for each tuple consisting of a vehicle and pair of destinations, each either at a depot or customer, denoting the first and second destination. The associated variable is used to indicate if the given vehicle travels from the first destination to the second destination. Expanded LP relaxations have one variable for each possible route, and each such variable is used to indicate that the given route is selected. While both the compact and expanded LP relaxations have the same optimal integer linear programming (ILP) solution objective, these formulations can have different optimal LP solution objectives. The expanded LP relaxation is often tighter and is never looser than the compact LP relaxation and hence is often preferred in practice (Costa et al. 2019). The number of possible routes can grow rapidly in the number of customers and often can not be easily enumerated much less considered in optimization. Hence the expanded LP relaxation is solved using CG, which imitates the revised simplex, where the pricing operation is a resource constrained shortest path (RCSP) problem (Baldacci et al. 2011, Desrochers et al. 1992). The RCSP is NP-hard but is efficiently solvable at the scale of many practical problems (Desrosiers and Lübbecke 2005).

Dual stabilization approaches are employed to accelerate the convergence of CG in terms of time or iterations and does so by altering the sequence of dual solutions generated but does not alter the structure of the pricing problem (Marsten et al. 1975, Du Merle et al. 1999, Haghani et al. 2021). The stabalization mechanism which we employ in this paper is Graph Generation (GG) (Yarkony et al. 2021). To apply GG, we must be able to map any given column to a small directed acyclic graph for which any path from source to sink describes a feasible column. This structure is easily satisfied for vehicle routing and crew scheduling problems; and other such problems where pricing is a resource constrained shortest path problem (RCSP). Such graphs are then added to the restricted master problem (RMP) when the corresponding column is generated during pricing. The use of GG does not weaken the expanded LP relaxation. At any given iteration of CG, GG permits the RMP to express a much wider set of columns than those generated during pricing. As a result, incorporating GG to faster convergence of CG since fewer iterations of CG are needed.

When graphs are large the GG RMP can become a computational bottleneck which is easily circumvented using Principled Graph Management (PGM) (Yarkony and Regan 2022). PGM exploits the following properties of the GG RMP: **1:** the addition of a small number of edges to a GG graph has the effect of adding a large number of columns to the GG RMP without significantly increasing GG RMP solution time; **2:** most edges are not active in the final solution to the GG RMP at any iteration of GG; **3:** computing the lowest reduced cost column associated with any graph is a simple shortest path problem (**not a RCSP**).

PGM solves the GG RMP in a manner akin to CG by alternating between the following two steps: **a**: solving the RMP over a subset of the edges in the graphs; **b.**: adding edges to the subset under consideration associated with the lowest reduced cost column. PGM terminates when the optimal solution to the RMP is solved, which is achieved when no negative reduced cost columns exist in the graphs.

In standard applications of GG to vehicle routing such as Capacitated Vehicle Routing Problem (CVRP), each route is associated with a graph representing the ordering of customers in the route. A route is consistent with that ordering if by removing customers from the ordering in the graph we obtain the route. The construction of orderings is motivated by the observation that customers that are in similar physical locations should be in similar positions on the ordered list. Thus given a route (Yarkony et al. 2021) iterates through the customers not in the original route and inserts the given customer behind their nearest customer in the original route.

CG only enumerates a small subset of the possible routes to produce an LP solution. Solving the ILP over these routes produces a value that is often very close to the ILP solution over all possible routes (Barnhart et al. 1996). Hence to generate provably optimal integer solutions branch & price (Barnhart et al. 1996) methods are employed which branch on terms in the compact relaxation and solve the expanded LP relaxation at any given node in the branch-bound tree using CG. Branch & price methods do not disrupt the structure of the pricing problem, unlike cutting plane methods, which can be employed jointly with branch & price (and the product is called branch & cut & price (Desrosiers and Lübbecke 2011)) to tighten the relaxation. Cutting plane methods add valid inequalities to the LP relaxation which are satisfied by every integer solution but not every fractional solution. Cutting plane methods add valid inequalities as needed when the LP solution violates such an inequality.

Subset row inequalities (SRI) (Jepsen et al. 2008) are one class of SRI, which in practice significantly tightens the expanded LP relaxation for vehicle routing problems. We describe a simple SRI below: the number of routes servicing two or three members of a given set of three customers cannot exceed one. We

describe a more elaborate SRI below: the number of routes servicing two or three members of a given set of five customers plus two times the number of routes servicing four or five of those customers can not exceed two. SRI of low order (defined over small number of customers such as 3,4,5) that are violated can be easily identified and hence can be satisfied in a cutting plane manner. SRI are challenging to use in standard CG formulations since they alter the structure of the pricing problem. This is because each SRI creates an additional resource that must be kept track of by the RCSP solver.

In order to gain the advantages of SRI without altering the structure of the pricing problem, which is a key goal, we apply LA routes (Mandal et al. 2022). LA routes are a superset of feasible elementary routes (where elementary means that a customer is included no more than once), and a subset of feasible ng-routes (Baldacci et al. 2011). n LA route is described as a sequence of LA arcs. Each LA arc starts at a customer, then contains sequence of customers nearby the first customer in the arc, and ends with a customer far away from the first customer in the arc. An LA arc contains no customer more than once. The final customer on an LA arc is the same as the first customer on the subsequent LA arc in a route. LA routes can be efficiently priced over, and can be used with a Decremental State Space Relaxation (Righini and Salani 2009) to produce the lowest reduced cost elementary routes. Hence any elementary route can be written as a sequence of LA arcs.

We weaken the LP relaxation over elementary routes enforcing SRI marginally by enforcing SRI constraints over LA arcs and not routes. We refer to the weakened SRI as Local Area-SRI (LA-SRI). We now express LA-SRI over LA arcs for a simple example used previously as follows. We enforce that the number of LA arcs in routes in our solution servicing two or three members of a given set of three customers can not exceed one. All customers in an LA arc except the last customer in that LA arc are serviced by that arc. LA-SRI are designed to effectively enforce SRI defined over customers localized in space. The dual variables for LA-SRI in the RMP are associated with the cost terms over LA arcs in pricing, leaving the structure of the pricing problem unmodified when LA-SRI are added.

In this paper, we apply LA-SRI in a manner so as to exploit the structure of LA routes inside a GG solver (Yarkony et al. 2021). We relax the consistency criteria (for routes in GG) to enforce that routes associated with a given graph only use LA arcs that satisfy the following: All intermediate customers must come before/after the last/first customer in the LA arc respectively in the associated ordering of the graph; and the first customer in the LA arc must come before the last customer in the LA arc in the associated ordering of the graph. This change does not loosen the expanded LP relaxation but decreases the number of iterations of GG required to solve the expanded LP relaxation.

We organize this document as follows. In Section 2 we review the existing literature on valid inequalities used with CG, and branch & price. In Section 3 we provide a mathematical review of CG for CVRP, LA routes, and SRI. In Section 4 we describe LA-SRI, and show how the structure of the pricing problem is preserved when adding LA-SRI. In Section 5 we describe an efficient GG/PGM based solver for the expanded LP relaxation that is able to consider LA-SRI. In Section 6 we provide an experimental validation of our approach. In Section 7 we conclude and discuss extensions.

2 Literature Review

In this section we briefly review the most related literature on valid inequalities, and branch & price (Barnhart et al. 1996, 2000). Valid inequalities are used to tighten the expanded LP relaxation (Costa et al. 2019) and are added to the expanded LP relaxation in a cutting plane manner. In a cutting plane method, valid inequalities iteratively remove optimal fractional solutions from the LP solution space. Valid inequalities do not remove any integer solutions from the feasible space. Large classes of valid inequalities exist. One key class that is of use in vehicle routing and combinatorial optimization problems in operations research (and computer vision) are subset row inequalities (SRI) (Jepsen et al. 2008, Yarkony et al. 2020, Wang et al. 2017). The most trivial variant of SRI for CVRP enforces that the number of vehicles including two or more customers from a given set of three does not exceed one. More elaborate SRI exist as we will see later in the document. SRI are rank one Gomory-Chitival cuts. The use of SRI alters the structure of the CG pricing problem, requiring the resource constrained shortest path solver for pricing to keep track of additional resources.

Another key class of valid inequalities are capacity inequalities (Archetti et al. 2011). Capacity inequalities ensure that the number of selected arcs (where an arc indicates travel from one customer/depot immediately to another customer/depot) composing the route entering a subset of customers is no less than the minimum number of vehicles required to service all customers in that subset. Capacity inequalities are challenging

(and are in fact NP-hard) to identify (Ralphs et al. 2003) (identification is often referred to as separation). However, capacity inequalities have the key advantage over SRI that they do not alter the structure of the pricing problem. This is because the dual variables of capacity inequalities are associated with the traversals of arcs present in the pricing problem. The reduced cost of a route thus remains the sum of the costs of the edges in the pricing graph leaving the structure of the pricing problem unmodified.

An alternative or complement to valid inequalities is branch & price (Barnhart et al. 1996) (which in its complementary form is called branch & cut & price (Desrosiers and Lübbecke 2011)). In branch & price, branching is done on variables present in the compact LP relaxation. For example, we would branch whether a given customer u should be followed by a customer v in a route (Ryan and Foster 1981). This branching process does not alter the structure of the pricing problem and only adds weights with infinite value on the prohibited edges over which pricing is done.

Valid inequalities (and branching on variables of the compact form) are challenging to use with dual optimal inequalities (DOI) (Ben Amor et al. 2006) as DOI that are correct for the expanded LP relaxation need not be correct when valid inequalities or branching are applied (Haghani et al. 2021). However violated DOI can be removed on demand in a cutting plane manner.

3 Mathematical Review

In this section we review the core mathematical techniques required to understand the contributions of this paper.

3.1 Column Generation for CVRP

We now consider the Capacitated Vehicle Routing Problem (CVRP), which we define briefly as follows. We are given a set of customers located in a metric space with integer demands, a number of homogeneous vehicles with common capacity, and a depot located in the same metric space. We seek to cover the customers with a set of ordered lists (of customers) called routes. Each route is serviced by a unique vehicle so as to minimize the total distance traveled, while ensuring that no vehicle services more demand than it has capacity. Vehicles start and end at the depot.

We now describe CVRP formally using the following notation. We use N to denote the set of customers which we index by u. We use N^+ to denote N augmented with the starting/ending depot denoted $-1, -2$ which are often at the same location. We use K to denote the number of vehicles available for use each with a capacity d_0.We use d_u to denote the demand of $u \in N^+$ where $d_{-1} = d_{-2} = 0$. We use Ω to denote the set of feasible routes which we index by l. We use $a_{ul} = 1$ if route l services customer u. We use use $a_{uvdl} = 1$ if route u leaves $u \in N^+ - (-2)$ with d units of capacity remaining then travels immediately to $v \in N^+ - (-1 \cup u)$. We use F to denote the set of feasible values (uvd) for a_{uvdl} which is defined as $u \in N^+ - (-2), v \in (N^+ - (-1, u)), (d_0 - d_u)[v \neq -2] \geq d \geq d_v$. We use c_{uv} to denote the distance between any pair $u \in N^+, v \in N^+$. Using c_{uv}, we write the cost of a route l which we denote as c_l as the total distance traveled below.

$$c_l = \sum_{uvd \in F} c_{uv} a_{uvdl} \quad \forall l \in \Omega \tag{1}$$

We now describe the necessary and sufficient conditions that a set of values $\{a_{ul}, a_{uvdl}\}$ satisfy for l to lie in Ω with annotation below.

$$\sum_{-1,u,d \in F} a_{-1,u,d,l} = 1 \quad \forall l \in \Omega \tag{2a}$$

$$a_{ul} = \sum_{\substack{v \in N^+ \\ d_0 - d_u \geq d \geq d_v}} a_{uvdl} \quad \forall l \in \Omega, u \in N \tag{2b}$$

$$\sum_{\substack{v \in N^+ \\ (v,u,d) \in F}} a_{vudl} = \sum_{\substack{v \in N^+ \\ (u,v,d-d_u) \in F}} a_{uv,d-d_u,l} \quad \forall l \in \Omega, u \in N, d \geq d_u \tag{2c}$$

$$a_{uvdl} \in \{0, 1\} \quad \forall uvd \in F \tag{2d}$$

$$a_{ul} \in \{0, 1\} \quad \forall u \in N \tag{2e}$$

In (2a) we enforce that the route starts at the starting depot. In (2b) we enforce that the the a_{uvdl} and a_{ul} terms are consistent . In (2c) we enforce that a vehicle must leave a customer u with the amount of resource it entered with minus d_u. In (2d),(2e) we enforce that a_{ul}, a_{uvdl} are binary valued so that a set of actions is specified for the route, and that there is no fractional servicing of customers.

We now consider optimization for CVRP below as a set cover problem. Given our definition of Ω the standard CVRP LP relaxation is written using decision variables θ_l where $\theta_l = 1$ if route l is selected in our solution and otherwise set $\theta_l = 0$. We write the minimum weight set cover LP relaxation over Ω as $\Psi(\Omega)$ below, with dual variables written in [] as π; with exposition after the equations.

$$\Psi(\Omega) = \min_{\theta \geq 0} \sum_{l \in \Omega} c_l \theta_l \tag{3a}$$

$$\sum_{l \in \Omega} a_{ul} \theta_l \geq 1 \quad \forall u \in N \quad [\pi_u] \tag{3b}$$

$$\sum_{l \in \Omega} \theta_l \leq K \quad [-\pi_0] \tag{3c}$$

In (3a) we minimize the total cost of the routes used. In (3b) we ensure that each customer is covered (serviced) at least once (though an optimal solution will service each customer exactly once). In (3c) we ensure that no more than K vehicles are used in the solution.

Since the cardinality of the set of routes Ω can grow exponentially in the number of customers we cannot trivially solve (3). Instead CG (Barnhart et al. 1996, Desrochers et al. 1992) is employed to solve (3). CG constructs a sufficient subset of Ω denoted Ω_R s.t. solving (3) over Ω_R provides an optimal solution to (3) over Ω. To construct Ω_R, we iterate between **(1)** solving (3) over Ω_R, which is referred to as the restricted master problem (RMP) and written as $\Psi(\Omega_R)$ and **(2)** identifying at least one $l \in \Omega$ with negative reduced cost, which are then added to Ω_R. Typically the lowest reduced cost column (member of Ω) is generated. We write the selection of this column as optimization below using $\bar{c}_l$ to denote the reduced cost of $l \in \Omega$.

$$\min_{l \in \Omega} \bar{c}_l \tag{4a}$$

$$\bar{c}_l = c_l + \pi_0 - \sum_{u \in N} a_{ul} \pi_u \quad \forall l \in \Omega \tag{4b}$$

The operation in (4) is referred to as pricing. CG terminates when pricing proves no column with negative reduced cost exists in Ω. This certifies that CG has produced the optimal solution to (3). We typically initialize Ω_R with columns corresponding to artificial variables that have prohibitively high cost to use in an optimal solution but can be used to create a feasible solution. This can be done by creating $|N|$ variables each of which covers a customer u with prohibitively high cost but without using a vehicle. Pricing is attacked as an elementary resource constrained shortest path problem (RCSP) and may be treated with a variety of algorithms including but not limited to dynamic programming based labeling algorithms(Desrosiers and Lübbecke 2005).

3.2 Pricing using Local Area Routes Solver

In this section we review the Local Area (LA) route relaxation based solver of (Mandal et al. 2022) for CVRP pricing problems. This section is abbreviated from (Mandal et al. 2022) and adapted to the use of notation later in this paper. To learn more about LA routes, refer to (Mandal et al. 2022), notably for the use of dominance criteria during pricing (Desaulniers et al. 2005). LA route solvers employ a Decremental State Space Relaxation (DSSR) (Righini and Salani 2009) over LA routes, which are a superset of elementary routes (and LA routes are a subset of the popular ng-routes (Baldacci et al. 2011)) and are easily priced over. DSSR gradually decreases the subspace of LA routes considered until the optimal LA route is elementary.

We organize this section as follows. In Section (3.2.1) we define the notation and basic information associated with LA routes. In Section (3.2.2) we provide the definition for LA routes. In Section (3.2.3) we describe the use of LA routes inside of DSSR to produce the the lowest reduced cost route elementary route. In Section (3.2.4) we describe the computation of the lowest reduced cost LA route using the Bellman-Ford Algorithm with efficient computation of edge weights for the associated graph specified in Section 3.2.5. In Section (3.2.6) we describe replace Bellman-Ford with Dijkstra's algorithm which has better computational

performance; this is an intermediate step leading to Section (3.2.7) where we replace Dijkstra's algorithm with A* (Dechter and Pearl 1985) for further computational efficiency. In Section (3.2.8) we describe the reduction in the space of LA routes for DSSR so as to improve the efficiency of pricing.

3.2.1 Local Area Structure of Reduced Cost

We now describe costs and terms associated with computing the lowest reduced cost route as done in (Mandal et al. 2022). Our description employs the following notation. We use $N_u \subseteq N - u$ to denote the set of customers nearby u for a given $u \in N$. We refer to N_u as the LA neighbors of customer u. The selection of the cardinality of N_u involves computational trade offs seen later in the document. In (Mandal et al. 2022) N_u is set to be the 10 nearest neighbors of u in for each $u \in N$. We use Ω_y where $y = (u, v, d)$ for some $(u \in N,\, v \in (N^+ - (N_u \cup u \cup -1)), d_0 - d_v \geq d \geq d_u)$, which we index by p, to denote the set of elementary paths p meeting the following properties.

- Path p starts at u, ends at v and all intermediate customers denoted N_p lie in the LA neighborhood of u; meaning that $N_p \subseteq N_u$.
- The total demand serviced in path p prior reaching v is d meaning $\sum_{w \in N_p \cup u} d_w = d$.
- We use c_p to denote the total travel distance (cost) of path p. The ordering of the intermediate customers N_p in path p is denoted ζ_p and is set so as to minimize c_p.

We use Y, which we index by y, to denote the set of $(u \in N,\, v \in (N^+ - (N_u \cup u \cup -1)), d \geq d_u)$ for which Ω_y is non-empty.

The LA routes approach seeks to find a resource constrained shortest path on a directed acyclic multi-graph with the following properties. It is constructed so that any elementary path from source to sink has total cost on traversed edges equal to $\bar{c}_l$ and that the set of elementary paths from source to sink is exactly Ω. Thus finding the lowest reduced cost elementary path solves (4). We define this multi-graph as follows using I, E to denote the set of nodes and edges respectively.

- We index I using i or j. There is one node corresponding to the source $(-1, d_0)$ and the sink $(-2, 0)$. There is one node for each u, d where $u \in N$ and $d_0 \geq d \geq d_u$.
- We connect $(-1, d_0)$ to node (u, d) for each $d \geq d_u$. This edge is associated with weight $\bar{c}_{ij} = c_{uv} + \pi_0$. Traversing this edge indicates that the vehicle leaves the depot on a route servicing exactly d units of demand on the route, where its first customer is u. We use N_{ij} to denote the set of customers associated with a given edge. This edge is associated with servicing no customers meaning $N_{ij} = \{\}$.
- For each $y \in Y, p \in \Omega_y$ (where $y = (u, v, d)$) and nodes in $i \in I, j \in I$ $i = (u, d_1), j = (v, d_1 - d)$ we connect i to j with an edge with weight $\bar{c}_{ij} = c_p - \sum_{w \in N_p \cup u} \pi_u$. Traversing this edge indicates that upon arriving at u with d_1 units of demand remaining that the vehicle travels on path p servicing the customers in N_p then proceeds immediately to v. Here $N_{ij} = u \cup N_p$.

Given a path from source to sink containing the arcs $\hat{E}$ this path is elementary (services no customer more than once) if the edges in $\hat{E}$ have disjoint N_{ij} sets. We use $a_{ijl} = 1$ if route l uses edge ij. Thus we write the reduced cost of a route l as follows in terms of $\bar{c}_{ij}$.

$$\bar{c}_l = \sum_{ij \in E} a_{ijl} \bar{c}_{ij} \tag{5}$$

We use p_{ij} to denote the path corresponding to edge ij where $p_{ij} = 0$ if $i = (-1, d_0)$ The solver for LA route employs a Decremental State Space Relaxation (DSSR) (Righini and Salani 2009) exploiting the fact that for c_p terms can be computed in advance as long as $|N_u|$ does not become exceedingly large.

3.2.2 LA routes: Component for Pricing

In this section we define LA routes which we use to solve pricing. We express LA routes using the following terms $M_u \subseteq N - u$ which we refer to as ng-neighbors as introduced by (Baldacci et al. 2011) and used by many different researchers. The sets $M_u \forall u \in N$ grow over the course of DSSR (Righini and Salani 2009).

We use Ω^{LA} to denote the set of paths from source to sink in the graph in Section 3.2.1 some of which are not elementary and hence $\Omega \subseteq \Omega^{LA}$. To assist in our description we use u^l_k to refer to the $k'th$ customer visited in the route l. We use Q^l to refer to the set of special indexes. The special indexes correspond to the nodes visited on the route. Thus $q^l_j = k$ if the first j LA arcs on the route l contain a total of k customers; meaning $\sum_{ij \in E^{lj}} |N_{ij}|$; where E^{lj} is the first j edges starting from the source on route l. The feasibility of an LA route given M is written as follows.

$$(u = u^l_{k_1} = u^l_{k_2}) \rightarrow \exists k_3 \quad \text{s.t. } k_1 < k_3 < k_2, \quad k_3 \in Q^l, \quad v = u^l_{k_3}, \quad u \notin M_v \qquad \forall \{u \in N, k_2 > k_1\} \tag{6}$$

The premise of (6) (left hand side of the $\rightarrow$ in (6)) is true if the same customer is at indexes k_1 and k_2 and that customer is u. The inference (right hand side of the $\rightarrow$ in (6)) is that there must exist a customer at special index k_3 that lies between k_1 and k_2 and does not consider u to be a ng-neighbor.

3.2.3 Decremental State Space Relaxation

DSSR generates the lowest reduced cost elementary route by iterating between solving for the lowest reduced cost LA route given M followed by augmenting M.

We describe the procedure to use DSSR alongside LA routes as follows. We define the N_u for all $u \in N$ to be the composed of the nearest customers to customer u. We initialize $M_u = \{\}$ $\forall u \in N$, but these M_u sets increase in size over iterations of DSSR. We define M_{-1}, M_{-2} to be initialized empty and remain empty. We then iterate between the following two steps until the lowest reduced cost LA route generated is elementary.

1. Solve for the lowest reduced cost LA route, also denoted as l, as a shortest path problem (not a resource constrained shortest path problem). This procedure is described in Section 3.2.4.

2. Find a cycle of customers in the selected route l (if it exists). Consider that the cycle identified starts ends with customer u at indexes k_1, k_2 where $k_2 > k_1$. Now, for each intermediate customer $k_3 \in Q^l$ for which $k_2 \geq k_3 > k_1$ add u to $M_{u^l_{k_3}}$. In Section 3.2.4 we shall see that smaller M_u sets are computationally desirable for pricing over LA routes (which is step 1 of DSSR) (Mandal et al. 2022). Thus, intelligent cycle selection is done so as to keep M_u sets small which we discuss in Section 3.2.8.

3.2.4 Pricing over LA routes

In this section we discuss generating the lowest reduced cost LA route given ng-neighbors sets M, as required in step **(1)** in DSSR as a simple shortest path computation which we solve can solve with Bellman-Ford. To assist in this description we introduce the following terms. For any given tuple $z = (u, v, M_1 \subseteq M_u, M_2 \subseteq M_v, d)$ we use Ω_z to denote the subset of Ω_y where $y = (u, v, d)$ s.t. $p \in \Omega_z$ has N_p satisfy all of the following properties: $|N_p \cap M_1| = 0$, $M_2 = M_v \cap (u \cup M_1 \cup N_p)$. We use Z, which we index by z, to denote the set of non-empty Ω_z.

Given any dual solution π, we define $\bar{c}_z$ as the cost of the lowest reduced cost path in Ω_z as written below.

$$\bar{c}_z = \min_{p \in \Omega_z} \bar{c}_p \tag{7}$$

We define minimizing path in (7) as p^z. Using terms p^z we describe the following graph with vertex and edge sets denoted I^M, E^M where the lowest reduced cost LA route in Ω^{LA} corresponds to the lowest cost path from the source to the sink.

- We index I^M with i, j. There is one node in I^M corresponding to the source denoted $(-1, \{\}, d_0)$, and one node for the sink denoted $(-2, \{\}, 0)$. For each node $(u, d) \in I$ (all nodes are defined in this manner in I excluding source and sink), create one node (u, M_1, d) in I^M for each tuple u, M_1, d satisfying the following: $M_1 \subseteq M_u, d_0 - \sum_{w \in M_1} d_w \geq d \geq d_u$.

- We connect $(-1, \{\}, d_0)$ to node $(u, \{\}, d)$ for each $u, \{\}, d\}$ in I^M. This edge is associated with weight $\bar{c}_{ij} = c_{uv} + \pi_0$. Traversing this edge indicates that the vehicle leaves the depot on a route servicing exactly d units of demand, where its first customer serviced is u.

- For each $z \in Z$ (where $z = (u, v, M_1, M_2, d)$) and nodes in $i \in I^M, j \in I^M$ where $i = (u, M_1, d_1), j = (v, M_2, d_1 - d)$ we connect i to j with an edge with weight $\bar{c}_{ij} = c_{p^z} - \sum_{w \in N_{p^z} \cup u} \pi_u$. Traversing this edge indicates that upon arriving at u with d_1 units of demand remaining that the vehicle travels on path p^z meaning that it services the customers in N_{p^z} after leaving u then proceeds immediately to v.

Given I^M, E^M we can solve pricing using the Bellman-Ford algorithm to find the shortest path from -1 and -2 in E^M, since E^M may have negative weights but no negative weight cycles as E^M describes a directed acyclic graph.

The set of paths from source to sink in I^M, E^M does not include all LA routes. However we now establish that this set of paths includes the lowest reduced cost LA route which is sufficient in order to complete step (1) of DSSR. In order to permit every LA route to be expressed we use a multi-graph I^M, E^{M+} where E^{M+} has all edges in E^M plus the following additional edges. For each $z \in Z, p \in \Omega_z - p^z$ (where $z = (u, v, M_1, M_2, d)$) and nodes in $i \in I^M, j \in I^M$ where $i = (u, M_1, d_1), j = (v, M_2, d_1 - d)$ we connect i to j with an edge with weight $\bar{c}_{ij} = c_p - \sum_{w \in N_p \cup u} \pi_u$. Traversing this edge indicates that upon arriving at u with d_1 units of demand remaining that the vehicle travels on path p meaning that it services the customers in N_p after leaving u then proceeds immediately to v. Observe that the lowest cost path from source to sink (I^M, E^{M+}), does not use any edge between a pair of nodes i, j other than the one with lowest possible cost. Thus the lowest cost path in (I^M, E^{M+}) uses only edges in E^M. Therefore the lowest cost path on (I^M, E^M) is the lowest reduced cost LA route.

3.2.5 Fast Computation of LA Arc Costs

In this subsection we describe the pre-computation of lowest cost component path terms, a process that ensures that pricing using LA routes is computationally fast. This pre-computation is done once prior to the first iteration of CG and never needs to be repeated. This section shows that c_p terms are easy to compute using a dynamic program when the size of N_u sets are small.

Let P^+ be defined as the set of tuples of the form $p = (u, v, \hat{N})$ where for each $p \in P^+$ there exists a $u_1 \in N$ s.t. $(u \in (N_{u_1} \cup u_1), v \in N^+ - (\hat{N} \cup u \cup u_1), \hat{N} \subseteq N_{u_1})$. Let $c_{uv\hat{N}}$ for any $\{u, v, \hat{N}\} \in P^+$ denote the cost of the lowest cost path starting at customer u, ending at v, and visiting all customers in $\hat{N}$ which we denote as $c_{uv\hat{N}}$. We write $c_{uv\hat{N}}$ recursively below with helper terms $c_{u_2,u_2,\{\}} = 0$ and $c_{u_2,v_2,\{\}} = c_{u_2,v_2}$ for all $u_2 \in N^+, v_2 \in N^+$ to describe the base cases.

$$c_{u,v,\hat{N}} = \min_{w \in \hat{N}} c_{u,w} + c_{w,v,\hat{N}-w} \quad \forall (u, v, \hat{N}) \in P^+ \tag{8a}$$

In Alg 1 we generate the c_p terms efficiently by iterating over the elements of P^+ in the order of increasing sizes of intermediate customer sets, using (8a) to evaluate c_p.

Algorithm 1 Fast Computation of c_p, ζ_p terms

```
for f = 1 : max_{(u,v,N̂)∈P+} |N̂| do
    for p ∈ P+ , p = {u, v, N̂}, |N̂| = f do
        w ← arg min_{w∈N̂} c_uw + c_{w,v,N̂−w}
        c_{u,v,N̂} ← c_uw + c_{w,v,N̂−w}
        ζ_{u,v,N̂} ← [w, ζ_{w,v,N̂−w}]
    end for
end for
```

3.2.6 Replacing Bellman Ford with Dijkstra' Algorithm

In this section we transform the weights of E^M to an equivalent representation with no negative edge weights so that we can apply Dijkstra's algorithm instead to find the shortest path. Applying Dijkstra's algorithm to an equivalent representation of E^M is desirable due to the improved asymptotic time complexity of Dijkstra's algorithm compared to the Bellman-Ford algorithm. Using Dijkstra's algorithm to find the shortest path from source to sink, only the p^z terms corresponding to expanded nodes are required to be computed. In comparison, Bellman-Ford algorithm requires the computation of all possible p^z terms, making Bellman-Ford far slower computationally.

We define the demand of a node to be the demand associated with the associated customer with 0 demand associated with depots. Thus we define $d_{\{-1,\{\},d_0\}} = d_{\{-2,\{\},0\}} = 0$ and $d_i = d$ where i corresponds to (u, M_1, d). Observe that for any path starting at $(-1, \{\}, d_0)$ and ending at $(-2, \{\}, d_0)$ where the edges included are the set $\hat{E}^M$ the following property holds: $\sum_{ij \in \hat{E}} (d_i + ([i = -1] * d_0) - d_j) = d_0$. Let us offset $\bar{c}_{ij}$

where $i = (u, M_1, d_1)$ and $j = (v, M_2, d_2)$ by adding $\eta * (d_1 - d_2)$ to $\bar{c}_{ij}$ where η is the smallest value sufficient to ensure that all edge terms are non-negative. We define below the value of η.

$$\eta = - \min_{\substack{y \in Y \\ y=(u_1,v_1,d) \\ p \in \Omega_y}} \frac{\bar{c}_p}{d} \tag{9}$$

Observe that cost of every path from source to sink is increased by exactly ηd_0. Thus the optimal path from source to sink is not modified. By subtracting ηd_0 from the cost of the path generated by Dijkstra's algorithm we recover the reduced cost of the lowest reduced cost LA route.

3.2.7 Accelerating Decremental State Space Relaxation with A*

In this section we exploit information from the first iteration of DSSR to solve subsequent iterations efficiently using the A* algorithm (Dechter and Pearl 1985). The A* algorithm operates similarly to Dijkstra's algorithm except that it expands the un-expanded node $i \in I^M$ with the lowest sum of the distance to reach the node (denoted g_i) from the source and a heuristic (h_i). This heuristic h_i is a lower bound on the cost of the lowest cost path to reach the sink starting from that node.

Dijkstra's algorithm expands all nodes with shorter distance from source than the distance from the source to the sink meaning that all nodes. Thus in Dijkstra's Algorithm a node $i \in I^M$ is expanded if $g_i < g_{-2,\{\},0}$ (and potentially some nodes for which $g_i = g_{-2,\{\},0}$). In contrast A* expands all nodes where the distance from the source plus the lower bound to the sink is less than the distance from the source to the sink. Thus in A* each node $i \in I^M$ is expanded for which $g_i + h_i < g_{-2,\{\},0}$ (and potentially some nodes for which $g_i + h_i = g_{-2,\{\},0}$). Given a more accurate heuristic (meaning larger h_i values)far fewer nodes are expanded in A* than Dijkstra's algorithm.

Computation of the heuristic should be easy so as to permit efficient search. We describe a high quality easily computed heuristic as follows. Given empty M_u sets (as is the case during the first iteration of DSSR), we compute the shortest distance from each node $(u, \{\}, d)$ to the sink. We denote this heuristic as h_{ud} and refer to the graph where all ng-neighbor sets are empty as the initial graph which we denote as (I^0, E^0) with edge weights as described in Section 3.2.6 (meaning that the edge weight on ij is $\bar{c}_{ij} + \eta(d_i - d_j)$). We associate each h_{ud} term to nodes of the form $i = (u, M_1, d)$ for each $M_1 \subseteq M_u$. The h_{ud} terms are computed exactly (via Bellman-Ford) once for each call to pricing and not for each iteration of DSSR. This initial graph is much smaller than the graph in later iterations of DSSR so the computations of h_{ud} terms is not expensive. The heuristic of $h_{i=(-1,\{\},d_0)}$ is the cost of the shortest path on (I^0, E^0) from $(-1, \{\}, d_0)$ to $(-2, \{\}, 0)$ as computed by Bellman-Ford.

The A* algorithm requires that the heuristic meet the property of consistency meaning that accuracy of the heuristic for the children is no less than that of the parent (Dechter and Pearl 1985). Below we formally describe this property of consistency in the context of our our domain.

$$h_i \leq h_j + \bar{c}_{ij} + \eta(d_i - d_j) \quad \forall (ij) \in E^M \tag{10}$$

Observe that the h_{ud} terms provide a consistent heuristic for A* since the set of LA routes on the graph where all ng-neighbor sets are empty is a super-set of the set of LA routes on any graph described by (I^0, E^0) in DSSR when the ng-neighbor sets are non-empty.

3.2.8 Cycle Selection for Augmenting ng-Neighbor Sets

In this section, we describe how we select the violating cycle in step **(2)** of DSSR in order to optimize the computational time taken during pricing. In the step **(2)** of DSSR we select the cycle that causes the the least number of nodes to be added to the graph (I^M, E^M), since adding extra nodes in the graph I^M may require additional nodes to be expanded during A*. This is crucial as the number of nodes (u, M_1, d) in the graph (I^M, E^M) can grow exponentially in terms of $|M_u| \quad \forall u \in N$.

3.3 Subset Row Inequalities for CVRP

In this section we consider the use of the celebrated subset row inequalities (SRI) (Jepsen et al. 2008) to tighten the CVRP LP relaxation. SRI can dramatically tighten the LP relaxation for many CVRP problems (Costa et al. 2019). SRI are rank 1 Gomory-Chitaval cuts. SRI are satisfied for all feasible integer solutions

for CVRP but not necessarily for all fractional solutions. The use of SRI significantly tightens the CVRP LP relaxation. However, introducing SRI to the CVRP LP makes the pricing problem harder since one resource is added for each SRI.

We organize the remainder of this section as follows. In 3.3.1 we display how the LP relaxation produces a different solution from the ILP solution with an example, which motivates the use of SRI to help improve the LP approximation. In 3.3.2 we summarize the constraints placed by SRI, and a standard example of SRI. In 3.3.3 we show how we can derive SRI from the set cover constraint present in the LP relaxation for CVRP. In 3.3.4 we show how we can incorporate SRI into the LP relaxation and also the lowest reduced cost formula.

3.3.1 Motivating Example

Here we will describe an example illustrating the difference between the ILP and LP solutions to the same problem.

In our problem, we have a depot in San Diego (SD), and vehicles of capacity 2. There is no constraint on the number of vehicles used to service all customers. There are 3 customers in New York City (NYC), each with unit demand (demand of one). We denote these customers as u_1, u_2, and u_3. Customer u_1 is located in the Upper East Side, customer u_2 is located in the Upper West Side, and customer u_3 is located in Staten Island. The distances between customers and the depot correspond approximately to the actual geographical locations, so the distance from u_1 to u_2 is smaller than the distance from u_2 and u_3 and u_1 and u_3. We describe the distance from the depot to any of our customers as 0.5, the distance between u_1 and u_2 as a very comparatively small value ϵ, and the distance from u_2 and u_3 as well as the distance from u_1 and u_3 as 10ϵ.

Below, we define the four routes shared amongst our ILP and LP solutions with the associated costs c_l and the associated binary variables a_{ul} denoting the presence of customers in a route l. Note the customers in a_{ul} are visited in route l in an order so as to minimize cost, which is reflected by c_l.

- Route l_1: defined by $a_{u_1 l_1} = a_{u_2 l_1} = 1$ and $c_{l_1} = 1 + \epsilon$. Our total cost is defined by a route like the following. There is a cost of 0.5 to travel from the depot to u_1, a cost of ϵ to travel from u_1 to u_2, and a cost of 0.5 to travel from u_2 to the depot.

- Route l_2: defined by $a_{u_3 l_2} = 1$ and $c_{l_2} = 1$. Our total cost is defined by the following route. There is a cost of 0.5 to travel from the depot to u_3, and a cost of 0.5 to travel from u_3 to the depot.

- Route l_3: defined by $a_{u_1 l_3} = a_{u_3 l_4} = 1$ and $c_{l_3} = 1 + 10\epsilon$. Our total cost is defined by a route like the following. There is a cost of 0.5 to travel from the depot to u_1, a cost of 10ϵ to travel from u_1 to u_3, and a cost of 0.5 to travel from u_3 to the depot.

- Route l_4: defined by $a_{u_2 l_4} = a_{u_3 l_4} = 1$ and $c_{l_4} = 1 + 10\epsilon$. Our total cost is defined by a route like the following. There is a cost of 0.5 to travel from the depot to u_2, a cost of 10ϵ to travel from u_2 to u_3, and a cost of 0.5 to travel from u_3 to the depot.

One such optimal ILP solution sets decision variables $\theta_{l_1} = \theta_{l_2} = 1$. This has a total cost of $2 + \epsilon$, which is very close to 2.

One such optimal LP solution sets decision variables $\theta_{l_1} = \theta_{l_3} = \theta_{l_4} = \frac{1}{2}$. This has a total cost of $1.5 + 10.5\epsilon$, which is very close to 1.5.

The significant difference between the LP solution and ILP solutions motivates us to find additional constraints for the LP relaxation for the overall CVRP problem. With such constraints, which will be the SRI, we aim to have an LP relaxation which produces a solution much closer to the ILP solution for CVRP problem instances.

3.3.2 Formal Description of Subset Row Inequalities

In this section we provide a formal definition of SRI,with examples. The simplest example of SRI for CVRP is written as follows. For any given set of three unique customers denoted $N_\delta \subseteq N$ the number of routes including 2 or more members of N_δ can not exceed one.

$$\sum_{l \in \Omega} [2 \leq \sum_{u \in N_\delta} a_{ul}] \theta_l \leq 1 \quad \forall N_\delta \subseteq N, |N_\delta| = 3 \tag{11}$$

Enforcing (11) makes the LP relaxation in Section 3.3.1 tight meaning that the LP has objective 2 which is the same as the ILP, even when all possible routes are considered.

A more general version of SRI states that for any set of customers N_δ and scalar m_δ, the number of routes using $|m_\delta|$ or more customers is bounded by $\lfloor \frac{|N_\delta|}{m_\delta} \rfloor$. For example, for N_δ sets of size 5 and $m_\delta = 3$ for each set, the number of routes using 3 or more customers for each set N_δ is bounded by one. In another example, we include N_δ sets of size 5, and enforce that the number of routes using 2 or more customers in each set N_δ is bounded by two. We write a form for both examples and more cases below:

$$\sum_{l \in \Omega} \theta_l \left[m_\delta \leq \sum_{u \in N_\delta} a_{ul} \right] \leq \left\lfloor \frac{|N_\delta|}{m_\delta} \right\rfloor \quad \forall N_\delta \subseteq N, m_\delta \in \mathbb{Z}_+ \tag{12}$$

A complete form of SRI weights the decision variables describing routes by $\lfloor \frac{\sum_{u \in N_\delta} a_{ul}}{m_\delta} \rfloor$ which we derive in Section 3.3.3. This complete form allows us to express even more restrictions on routes covering subsets of N_δ sets, ensuring that constraining the LP relaxation using the complete form instead of the form defined by 12 produces a solution closer to the ILP form.

$$\sum_{l \in \Omega} \theta_l \left\lfloor \frac{\sum_{u \in N_\delta} a_{ul}}{m_\delta} \right\rfloor \leq \left\lfloor \frac{|N_\delta|}{m_\delta} \right\rfloor \quad \forall N_\delta \subseteq N, m_\delta \in \mathbb{Z}_+ \tag{13}$$

We can show that the complete form places more restrictions with the following example. In the complete form, for a set N_δ with size 5 and corresponding to $m_\delta = 2$, there is a constraint that the number of selected routes containing 2 or 3 customers in N_δ in addition to two times the number of routes containing 4 or 5 customers in N_δ cannot exceed 2. With this example, we correspondingly observe that equation (13) with a set N_δ of size 5 and corresponding $m_\delta = 2$ describes the same constraints given by equation (12) for the same set N_δ of size 5 but utilizing two different m_δ values 2 and 4 in a single inequality.

We also observe that equation (13) may constrain the LP relaxation when $\lfloor \frac{|N_\delta|}{m_\delta} \rfloor < \frac{|N_\delta|}{m_\delta}$, however, the LP relaxation will not be constrained when $\lfloor \frac{|N_\delta|}{m_\delta} \rfloor = \frac{|N_\delta|}{m_\delta}$. Failing to constrain the LP relaxation with such SRI values adds constraints without changing the value produced from an LP relaxation without SRI.

We prove the above statements using an example, and then extrapolate this example to satisfy the generality of our observations. In our example, we see that for a SRI defined by a N_δ set of size 4 and corresponding $m_\delta = 2$ that the expression that the right hand side of equation (13) is unmodified when the floor operation is not used, but the left hand side of equation (13) can be modified when the floor operation is not used.

We now establish formally that for an SRI δ to tighten (3) it is a necessary condition that $\lfloor \frac{|N_\delta|}{m_\delta} \rfloor < \frac{|N_\delta|}{m_\delta}$. Observe that if the floor operation is removed from both the RHS and LHS of (13) then the induced equation is the average of the constraints enforcing that each customer is covered exactly once each of which is satisfied by any solution to (3). Now observe that the average of satisfied inequalities (with $\leq$ facing the same direction) is also a satisfied inequality. If $\lfloor \frac{|N_\delta|}{m_\delta} \rfloor = \frac{|N_\delta|}{m_\delta}$ then the floor operation does not decrease the RHS but may decrease the LHS of (13). Thus we have established the claim.

3.3.3 Derivation of Subset Row Inequalities

Below we derive SRI as seen in (13), given for any $N_\delta \subseteq N$,$m_\delta \in \mathbb{Z}_+$ s.t. $\lfloor \frac{|N_\delta|}{m_\delta} \rfloor < \frac{|N_\delta|}{m_\delta}$ with annotation provided below the equations.

$$\sum_{l\in\Omega} \theta_l (\sum_{u\in N_\delta} a_{ul}) \geq |N_\delta| \tag{14a}$$

$$\sum_{l\in\Omega} \theta_l (\sum_{u\in N_\delta} a_{ul}) = |N_\delta| \tag{14b}$$

$$\sum_{l\in\Omega} \theta_l \frac{\sum_{u\in N_\delta} a_{ul}}{m_\delta} = \frac{|N_\delta|}{m_\delta} \tag{14c}$$

$$\left\lfloor \sum_{l\in\Omega} \theta_l \frac{\sum_{u\in N_\delta} a_{ul}}{m_\delta} \right\rfloor = \left\lfloor \frac{|N_\delta|}{m_\delta} \right\rfloor \tag{14d}$$

$$\sum_{l\in\Omega} \left\lfloor \theta_l \frac{\sum_{u\in N_\delta} a_{ul}}{m_\delta} \right\rfloor \leq \left\lfloor \frac{|N_\delta|}{m_\delta} \right\rfloor \tag{14e}$$

$$\sum_{l\in\Omega} \theta_l \left\lfloor \frac{\sum_{u\in N_\delta} a_{ul}}{m_\delta} \right\rfloor \leq \left\lfloor \frac{|N_\delta|}{m_\delta} \right\rfloor \tag{14f}$$

1. In (14a): We write the sum of inequalities of the standard cover $\sum_{l\in\Omega} a_{ul}\theta_l \geq 1$ over N_δ. Let θ^*, θ^+ be any fractional solution/binary valued solutions to the 3 respectively. Both θ^* and θ^+ satisfy (14a) since the sum of satisfied inequalities (with $\geq$ facing the same way) is also satisfied.

2. In (14b) we observe that for any optimal integer solution no customer is visited more than once thus we replace the $\geq$ with an $=$.

3. In (14c): We divide both sides by positive integer m_δ. Both θ^* and θ^+ satisfy (14b) since dividing both sides of an equality by a positive number produces another satisfied equality.

4. In (14d) we round down to the nearest integer on both sides using a floor operations. Since the LHS of (14c) is equal to the RHS of (14c), this equality is also satisfied for both θ^* and θ^+.

5. In (14e) we move the floor operations into the sum. Since the floor of a sum of values is no less than the sum of floor of each of those values, then both θ^* and θ^+ satisfy (14e).

6. In (14f) we move θ outside of the floor operation. Since θ_l^+ is binary valued for any $l \in \Omega$ by definition then $\lfloor \theta_l \frac{\sum_{u\in N_\delta} a_{ul}}{m_\delta} \rfloor = \theta_l \lfloor \frac{\sum_{u\in N_\delta} a_{ul}}{m_\delta} \rfloor$ hence θ^+ satisfies (14f). However (14f) may not be satisfied for non-binary θ^*. In fact for the example in Section 3.3.1 this inequality is not satisfied for the optimal fractional LP solution.

We use Δ which we index by δ to denote the set of SRI which we are considering in our problem, where δ is associated with N_δ, m_δ. We rewrite (14f) using $a_{\delta l}, b_\delta$ to help compact our notation, and define these terms as follows.

$$a_{\delta l} = \lfloor \frac{\sum_{u\in N_\delta} a_{ul}}{m_\delta} \rfloor \quad \forall l \in \Omega, \delta \in \Delta \tag{15a}$$

$$b_\delta = \lfloor \frac{|N_\delta|}{m_\delta} \rfloor \quad \forall \delta \in \Delta \tag{15b}$$

We now write a compacted form of (14f) as follows for any given $\delta \in \Delta$.

$$\sum_{l\in\Omega} \theta_l a_{\delta l} \leq b_\delta \quad \forall \delta \in \Delta \tag{16}$$

3.3.4 Column/Row Generation to Solve Optimization using SRI

In this section we consider a mechanism to solve optimization over all of Ω, Δ. Note that for clarification we refer to our column generation problem as a Column/Row Generation CGR) problem to better reflect the inclusion of the subset row inqeualities.

Given any subset of SRI denoted $\Delta_R \subseteq \Delta$ and set of columns $\Omega_R \subseteq \Omega$ we denote the Restricted Master Problem (RMP) as $\Psi^*(\Omega_R, \Delta_R)$ which we write formally below with dual variables next to the associated constraints.

$$\Psi^*(\Omega_R, \Delta_R) = \min_{\theta \geq 0} \sum_{l \in \Omega_R} c_l \theta_l \tag{17a}$$

$$\sum_{l \in \Omega_R} a_{ul} \theta_l \geq 1 \quad \forall u \in N \quad [\pi_u] \tag{17b}$$

$$\sum_{l \in \Omega_R} \theta_l \leq K \quad [-\pi_0] \tag{17c}$$

$$\sum_{l \in \Omega_R} a_{\delta l} \theta_l \leq b_\delta \quad \forall \delta \in \Delta_R \quad [-\pi_\delta] \tag{17d}$$

Given a solution θ to (17) and set Δ we write the identification of the most violated SRI set N_δ and associated value m_δ for $\delta \in \Delta$ (which is called separation) as follows.

$$\min_{\delta \in \Delta} b_\delta - \sum_{l \in \Omega_R} a_{\delta l} \theta_l \tag{18}$$

Typically new constraints are only added once the MP is solved optimally given fixed subset of SRI $\Delta_R \subseteq \Delta$. The pricing operation given π is written below.

$$\min_{l \in \Omega} \bar{c}_l \tag{19a}$$

$$\bar{c}_l = c_l + \pi_0 - \sum_{u \in N} \pi_u a_{ul} + \sum_{\delta \in \Delta_R} \pi_\delta a_{\delta l} \quad \forall l \in \Omega \tag{19b}$$

Pricing using (19a) is tricky to solve as the addition of the π_δ terms alters the structure of the pricing problem. This is known to be very problematic. Each $\delta \in \Delta_R$ introduces an additional resource to the resource constrained shortest path (RCSP) solver that must be kept track of. In Alg 2 we write the CG problem with SRI added in a cutting plane manner.

In computational studies using SRI, computational benefit is achieved by using Δ for which N_δ is small (typically 3) for all $\delta \in \Delta$ (Costa et al. 2019, Wang et al. 2017, Yarkony et al. 2020).

4 LA Arcs Encoding Subset Row Inequalities

In this section we tighten the CG LP relaxation with a variant of subset row inequalities (SRI) that is computationally efficient to consider. Specifically we introduce we provide a marginally weakened form of

Algorithm 2 Basic Column/Row Generation with SRI

$\Omega_R \leftarrow$ from user
$\Delta_R \leftarrow$ from user typically empty
repeat
  repeat
    $\theta, \pi \leftarrow$ Solve $\Psi^*(\Omega_R, \Delta_R)$
    $l_* \leftarrow \arg\min_{l \in \Omega} \bar{c}_l$
    $\Omega_R \leftarrow \Omega_R \cup l_*$
  until $\bar{c}_{l_*} \geq 0$
  $\delta_* \leftarrow \arg\min_{\delta \in \Delta} b_\delta - \sum_{l \in \Omega_R} a_{\delta l} \theta_l$
  $\Delta_R \leftarrow \Delta_R \cup \delta_*$
until $b_{\delta_*} - \sum_{l \in \Omega_R} a_{\delta_* l} \theta_l \geq 0$
Return last θ generated. This can be used inside branch-cut-price.

SRI, which we call LA-SRI, where the use of LA-SRI does not alter the structure of the pricing problem when using the Local Area (LA) routes relaxation based solver (Mandal et al. 2022). This permits the use of large numbers of LA-SRI defined over large subsets of customers to be used without adding computational difficulty to pricing.

We organize this section as follows. In Section 4.1 we describe SRI using LA arcs in a manner that does not permit efficient pricing. In Section 4.2 we introduce LA-SRI. In Section 4.3 we describe how the use of LA-SRI can be placed directly into the edge weights used in the LA route pricing problem meaning that the structure pricing is unmodified. In Section 4.4 we contrast SRI and LA-SRI so as to demonstrate where LA-SRI perform differently than SRI.

4.1 Formulating SRI over LA Arcs

We introduce the following notation to keep track of the inclusion of SRI in LA -arcs.

Let for any $y \in Y, p \in \Omega_y, \delta \in \Delta$ where $y = (u, v, d)$ us define $a_{p\delta}$ as follows.

$$a_{p\delta} = \sum_{w \in N_p \cup u} [w \in N_\delta] \quad \forall y \in Y, y = (u, v, d), p \in \Omega_y \tag{20}$$

Recall that N_p contains all customers in the LA arc p excluding u, v. We use $a_{pl} = 1$ if route l includes LA arc p.

Observe that for any given $l \in \Omega$ that we can write the sum of customers in a given subset corresponding to an SRI in terms of the inclusion of customers in the associated LA arcs.

$$\sum_{u \in N_\delta} a_{ul} = \sum_{\substack{y \in Y \\ p \in \Omega_y}} a_{p\delta} a_{pl} \quad \forall l \in \Omega, \delta \in \Delta \tag{21}$$

Using $a_{p\delta}$ we rewrite (14f) as follows. We begin by we (14f) in terms of $a_{p\delta}$ using (21)

$$\sum_{l \in \Omega} \theta_l \lfloor \frac{\sum_{\substack{y \in Y \\ p \in \Omega_y}} a_{p\delta} a_{pl}}{m_\delta} \rfloor \leq \lfloor \frac{|N_\delta|}{m_\delta} \rfloor \quad \forall \delta \in \Delta \tag{22}$$

4.2 LA-SRI: Formulating SRI in Marginally Weakened Form over LA Arcs

In this section we relax (22) so as to permit efficient inclusion in our LP relaxation. Starting from (22) we move the floor operation inside of the sum producing the following.

$$\sum_{l \in \Omega} \theta_l \lfloor \frac{\sum_{\substack{y \in Y \\ p \in \Omega_y}} a_{p\delta} a_{pl}}{m_\delta} \rfloor \geq \sum_{l \in \Omega} \sum_{\substack{y \in Y \\ p \in \Omega_y}} \theta_l a_{pl} \lfloor \frac{a_{p\delta}}{m_\delta} \rfloor \quad \forall \delta \in \Delta \tag{23}$$

Thus we enforce the following

$$\sum_{l \in \Omega} \sum_{\substack{y \in Y \\ p \in \Omega_y}} \theta_l a_{pl} \lfloor \frac{a_{p\delta}}{m_\delta} \rfloor \leq \lfloor \frac{|N_\delta|}{m_\delta} \rfloor \quad \forall \delta \in \Delta \tag{24a}$$

However since the LHS of (24a) is no greater than (22) the inequality (24a) is weaker than (22). However we shall see that these inequalities are easier to price over. Observe that the larger the size of the LA-neighborhood the less the gap between (24a) and (22) tends to become (though this need not be monotonic as we will see in Section 4.4). Let $a^*_{p\delta}$ be defined for efficient exposition as follows.

$$a^*_{p\delta} = \lfloor \frac{a_{p\delta}}{m_\delta} \rfloor \quad \forall p \in \cup_{y \in Y} \Omega_y, \quad \delta \in \Delta \tag{25}$$

Let $a^*_{\delta l}$ be the LHS value associated with valid inequality δ with variable l in (24a) which is defined as follows.

$$a^*_{\delta l} = \sum_{\substack{p \in \Omega_y \\ y \in Y}} a_{pl} a^*_{p\delta} \quad \forall l \in \Omega, \delta \in \Delta \tag{26}$$

Below we write the RMP enforcing all LA-SRI for Δ_R as described in (24a), which we denote as $\Psi(\Omega_R, \Delta_R)$.

$$\Psi(\Omega_R, \Delta_R) = \min_{\theta \geq 0} \sum_{l \in \Omega_R} c_l \theta_l \tag{27a}$$

$$\sum_{l \in \Omega_R} a_{ul} \theta_l \geq 1 \quad \forall u \in N \quad [\pi_u] \tag{27b}$$

$$\sum_{l \in \Omega_R} \theta_l \leq K \quad [-\pi_0] \tag{27c}$$

$$\sum_{l \in \Omega_R} a^*_{\delta l} \theta_l \leq b_\delta \quad \forall \delta \in \Delta_R \quad [-\pi_\delta] \tag{27d}$$

4.3 Pricing under LA-SRI

We now consider efficient pricing which we show is unmodified. This formulation for pricing over LA routes can be trivially adapted to consider LA-SRI.

The reduced cost of a route l in (27) is written in terms of dual variables π as follows.

$$\bar{c}_l = c_l + \pi_0 - \sum_{u \in N} \pi_u a_{ul} + \sum_{\delta \in \Delta_R} \pi_\delta a^*_{\delta l} \quad \forall l \in \Omega \tag{28}$$

We now rewrite (28) in terms of helper variable $\bar{c}_p$ which is the reduced cost associated with path p with is defined below.

$$\bar{c}_p = c_p - \sum_{w \in N_p \cup u} \pi_w + \sum_{\delta \in \Delta_R} \pi_\delta a^*_{p\delta} \quad \forall y \in Y, p \in \Omega_y, y = (u, v, d) \tag{29}$$

The reduced cost of a route $l \in \Omega$ (or Ω_{LA}) can now be written as follows using $a_{ij,l} = 1$ edge ij lies in route l and otherwise is $a_{ijl} = 0$

$$\bar{c}_l = (28) = \sum_{ij \in E} a_{ijl} \bar{c}_{ij} \quad \forall l \in \Omega \tag{30a}$$

$$\bar{c}_{(-1,d_0),(u,d)} = c_{-1u} + \pi_0 \quad \forall u \in N, d \geq d_u \tag{30b}$$

$$\bar{c}_{i,j} = c_p - \sum_{w \in N_p \cup u} \pi_w + \sum_{\delta \in \Delta_R} a^*_{p\delta} \pi_\delta \quad \forall i, j \in E, i \neq (-1, d_0), p = p_{ij}, i = (u, d_1), j = (v, d_2) \tag{30c}$$

Thus pricing can be carried out identical to as in the standard LA routes. The only difference is that the edge weights are changed. However no additional resources are used.

4.4 Tightness of LA-SRI Relative to SRI

In this section we compare tightening effect of a given SRI compared to the associated LA-SRI. We show when the relaxation of (22) to (24a) weakens the LA-SRI relative to the SRI and when it does not. However, we observe with well chosen and large LA-neighborhoods that the relaxation is not significantly weakened. We provide substantial proof for these phenomena with an analysis of examples below, and define these examples over the following problem domain.

In our examples, suppose that we have a depot in San Diego (SD), and vehicles of capacity 201. We have three customers in New York City (NYC), named NYC1, NYC2, and NYC3, and thee customers in San Diego, named SD1, SD2, SD3. NYC and SD are far from each other but the associated customers of each city are nearly co-located. Each NYC customer has a demand of 100 and the SD customer has a demand of 1. Each customer has neighbor sizes of 2, so each customer in NYC considers one another to be an LA neighbor, and each customer in San Diego considers one another to be an LA neighbor. This is because we set LA neighbor sets by closest location.

1. Consider a case with SRI defined to be N_δ=[NYC1,NYC2,SD1] and m_δ=2. Now consider routes l_1 and l_2. Route l_1 has the sequence of customers (with depot added) as follows [-1, NYC1, NYC3, SD1, SD2, -2] and route l_2 has the sequence [-1, NYC1, NYC2, SD2, SD3, -2]. Note that the special indexes are 1,3 for both l_1 and l_2. Observe that for route l_1 the associated variables are defined to be $a_{\delta l_1} = 1$ and $a^*_{\delta l_1} = 0$. However, for route l_2 observe that $a_{\delta l_2} = 1$ and $a^*_{\delta l_2} = 1$.

2. Consider a case with SRI defined to be N_δ=[NYC1,NYC2,NYC3] and m_δ=2. Now consider routes l_3 and l_4. Route l_3 has the sequence [-1, NYC1, NYC2, SD1, -2] and route l_4 has the sequence [-1, NYC1, SD1, NYC2, -2]. Note that the special indexes are 1,3 for l_3 and 1,2,3 for l_4. Observe that for route l_3 the associated variables are defined to be $a_{\delta l_3} = a^*_{\delta l_3} = 1$. However, for route l_4 observe that $a_{\delta l_4} = 1$ and $a^*_{\delta l_4} = 0$.

When solving the CVRP LP relaxation in (27) with the LA neighborhoods defined above alongside all SRI of $|N_\delta| = 3, m_\delta = 2$, we enforce that at least two routes visiting at least one customer in SD and at least one customer in NYC are used. The two routes must visit different customers in SD and NYC. When solving the CVRP LP formulation with LA neighborhoods and arcs but without SRI, there is no such constraint on routes used. Given our relaxed formulation of LA-SRI constraints, one would think that larger LA neighborhoods always tightens the bound at the expense of greater computational cost during iterations of DSSR in pricing. However, we find that increasing the size of LA neighborhoods may not always tighten the bound, which we demonstrate with an example below.

Consider that we add to each the LA neighborhoods in customers in SD so that the LA neighbors of customers in SD are defined as follows: $N_{SD1} = \{SD2, SD3, NYC1\}$,$N_{SD2} = \{SD1, SD3, NYC2\}$,$N_{SD3} = \{SD1, SD2, NYC3\}$. These LA neighborhood sets are still defined using closest distance. Now, consider the following solution to the MP $\Psi(\Omega, \Delta)$ using $\theta_{l_1} = \theta_{l_2} = \theta_{l_3} = \theta_{l_4} = \frac{1}{2}$ where routes l_1, l_2, l_3, l_4 are defined as sequences of the following LA arcs.

- $l_1 = \{(-1, SD1), (SD1, NYC1, NYC2), (NYC2, -2)\}$
- $l_2 = \{(-1, SD2), (SD2, NYC2, NYC3), (NYC3, -2)\}$,
- $l_3 = \{(-1, SD3), (SD3, NYC3, NYC1), (NYC1, -2)\}$,
- $l_4 = \{(-1, SD1), (SD1, SD2, SD3, -2)\}$

Observe that this solution uses $\frac{3}{2}$ cross country routes but no integer solution can use fewer than 2 cross country routes.

Based on the examples above we hypothesize that SRI defined over customers localized in space are crucial for tightening the expanded LP relaxation of CVRP (and other VRP). Furthermore, we observe that it is unlikely for an optimal solution to the expanded LP relaxation to use routes that frequently return to areas of customers localized in space after leaving such areas. This is because including such routes in an LP solution would lead to sub-optimal cost. An example of such an route is [-1, NYC1, SD1, NYC2, -2]; as instead, the customers can be serviced at lower cost using the route [-1, NYC1, NYC2, SD1, -2]. Thus, for violated SRI defined over customers localized in space, a key opportunity for the corresponding LA-SRI to not be violated is when LA arcs exist for which the following holds: the penultimate customer of the LA arc is nearby the final customer of the LA arc. By avoiding the construction of LA neighborhoods that allow for presence of such LA arcs, we diminish the possibility of weakening the LP relaxation using LA-SRI instead of SRI.

5 Stabilization of Optimization Exploiting LA Route Structure

In this section we describe a stabilized version of CG adapted to efficiently solve the CG master problem (MP) that does not alter pricing. We achieve this by altering the sequence of dual solutions generated by CG algorithm with the aim that fewer iterations of CG are required to solve the MP. At each iteration of CG a more computationally intensive RMP is solved that includes a larger set of columns than those generated during pricing. Solving over this set of columns does not dramatically increase the time taken since a simple computational structure is used to ensure that an exponential number of such columns can be encoded with a finite number of variables. This set of columns is written as $\cup_{l \in \Omega_R} \Omega_l$ where Ω_l contains an exponential number of columns related to column l. We refer to Ω_l as the family of column l. As a result, when solving

the RMP in our stabilized version of CG, we solve $\Psi(\cup_{l\in\Omega_R}\Omega_l, \Delta_R)$ instead of $\Psi(\Omega_R, \Delta_R)$ to accelerate the convergence of CG. In this section we develop an efficient solver for this approach that does not enumerate all $\cup_{l\in\Omega_R}\Omega_l$ and does not employ expensive pricing operations.

Families of columns are used in Graph Generation (GG) methods (Yarkony et al. 2021, Yarkony and Regan 2022). In this paper we exploit LA routes to describe families that are larger than those of (Yarkony et al. 2021, Yarkony and Regan 2022, Mandal et al. 2022) and permit the introduction of LA-SRI, which can not be done using the original GG families.

We organize this section as follows. In Section 5.1 we formally characterize the families in our formulation. In Section 5.1 we describe the generation of a family from a column. In Section (5.3) we describe the new RMP over graphs associated with families of columns leading to faster convergence inspired by CG. In Section (5.4) we consider a fast way to solve the RMP inspired by Principled Graph Management (PGM)(Yarkony and Regan 2022).

5.1 Families of Columns for CVRP

For any given route l we associate it with a strict total order of the members N^+ with β^l where $-1, -2$ have the smallest and largest β^l values respectively. A route $\hat{l} \in \Omega$ lies Ω_l if and only if LA arcs consistent with the ordering β^l are used. An LA arc p in Ω_y (for y=(u, v, d)) is consistent with the ordering β^l if $\beta^l_u < \beta^l_w < \beta^l_v$ for any $w \in (N_p$ where N_p is a set of the intermediate customers in route l; and $\beta^l_u < \beta^l_v$. It is mandatory that route l lies in Ω_l for all $l \in \Omega$. We use Ω^l_y to refer to the subset of $p \in \Omega_y$ for which p is consistent with ordering β^l. We use Y^l to denote the subset of $y \in Y$ for which $|\Omega^l_y| \neq 0$. We use E^l to denote

5.2 Generation of Ordering Given a Route

In this section we describe the procedure to generate β^l terms given a route l from (Yarkony et al. 2021). This construction is motivated by the observation that customers that are in similar physical locations should be in similar positions on the ordered list. Having customers in such an order ensures that a route in Ω_l can visit all customers close together without leaving the area and then coming back. We use a set N_l to denote the customers in the route l. We initialize the ordering with the customers in N_l sorted in order from first visited to last visited. Then, we iterate over $u \in (N - N_l)$ and insert u behind the customer in (N_l) nearest to u. We insert u that are closer to the depot than any customer in N_l at the beginning of the list. We observe that by using the aforementioned construction that l lies in Ω_l for all $l \in \Omega$. We use $E^l \subseteq E^0$ (where node/edge sets I^0, E^0 is defined in Section 3.2.7) to denote the set of edges in E^0, $(u, d_1), (v, d_2)$ for which $\beta^l_u < \beta^l_v$ and in addition where either $u = -1$ or $\Omega^l_{y=(u,v,d_1-d_2)}$ is non-empty.

5.3 Restricted Master Problem Using Stabilization

In this section we describe a new RMP formulation that efficiently describes $\Psi(\cup_{l\in\Omega_R}\Omega_l, \Delta_R)$.

We now define an LP over the graphs defined by the nodes,edge sets I^0, E^l for each $l \in \Omega_R$. We use the following decision variables with associated cost terms defined below.

- We use decision variable x^l_p to be a variable which denotes if a p composing a path in the family l is used in the solution, and how many vehicles (can be fractional or 1) are used for p. There is one such variable for each $p \in \cup_{y\in Y^l}\Omega^l_y, l \in \Omega_R$.
- We use decision variable x^l_{ij} to be a variable which denotes if an edge ij in a route in corresponding to family l is used in the solution, and how many vehicles (can be fractional or 1) are used for ij. There is one such variable for each $l \in \Omega_R, ij \in E^l$.
- We use decision variable θ_l to be a variable which denotes if a route l is used in the solution, and how many vehicles (can be fractional or 1) is used for that route l.
- We use c_l to denote the cost associated with traveling through route l.
- We use c_p to denote the cost associated with traveling in path p, ensuring that the cost of traveling through the intermediate customers is minimized.
- We use $c_{-1,(u,d)}$ to denote the cost associated with traveling from the source to customer u, which is also simply the distance from the source to customer u.

Using our decision variables x, θ we write $\Psi(\cup_{l\in\Omega_R}\Omega_l, \Delta_R)$ as a LP below which we refer to as $\Psi^+(\Omega_R, \Delta_R)$ (with equivalence proven in Appendix A) We provide annotation of this RMP below the equations.

$$\Psi^+(\Omega_R, \Delta_R) = \Psi(\cup_{l\in\Omega_R}\Omega_l, \Delta_R) = \min_{\substack{x\geq 0\\ \theta\geq 0}} \sum_{l\in\Omega_R} c_l\theta_l + \sum_{\substack{l\in\Omega_R\\ y\in Y^l\\ p\in\Omega_y^l}} c_p x_p^l + \sum_{\substack{l\in\Omega_R\\ u\in N\\ d_0\geq d\geq d_u}} c_{-1u} x^l_{(-1,d_0),(u,d)} \tag{31a}$$

$$\sum_{l\in\Omega_R} \theta_l + \sum_{l\in\Omega_R}\sum_{ij\in E^l} [i = (-1, d_0)]x_{ij}^l \leq K \quad [-\pi_0] \tag{31b}$$

$$\sum_{l\in\Omega_R} a_{ul}\theta_l + \sum_{\substack{l\in\Omega_R\\ y\in Y^l\\ p\in\Omega_y^l}} x_p^l a_{up} \geq 1 \quad \forall u \in N \quad [\pi_u] \tag{31c}$$

$$\sum_{l\in\Omega_R} a^*_{\delta l}\theta_l + \sum_{\substack{l\in\Omega_R\\ y\in Y^l\\ p\in\Omega_y^l}} a^*_{\delta p} x_p^l \leq b_\delta \quad \forall \delta \in \Delta_R \quad [-\pi_\delta] \tag{31d}$$

$$\sum_{p\in\Omega_y^l} x_p^l = \sum_{\substack{ij\in E^l\\ i=(u,d_1)\\ j=(v,d_1-d)}} x_{ij}^l \quad \forall y \in Y^l; y = (u, v, d), l \in \Omega_R, i \in I^0 - ((-1, d_0) \cup (-2, 0)) \tag{31e}$$

$$\sum_{ij\in E^l} x_{ij}^l = \sum_{ji\in E^l} x_{ji}^l \quad \forall i \in I^0, \quad u_i \notin \{-1, -2\}, l \in \Omega_R \tag{31f}$$

- (31a): We seek to minimize the sum of the costs of the LA arcs and edges (from the depot) and routes used in our solution.
- (31b): We enforce that no more that K vehicles leave the depot by ensuring that the sum of edges from the source and the sum of other routes used in our solution do not exceed K.
- (31c): We enforce that each customer is serviced in at least one LA arc or route by checking for all customers that the sum of decision variables a_{ul} (which denotes if a route l contains customer u) and a_{up} (which denotes if a LA arc p contains customer u) corresponding to a customer u is at least one.
- (31d): We enforce all LA-SRI over the LA arcs and routes which form our solution. This is done by taking the sum of $a^*_{\delta l}$ for all routes $l \in \Omega_R$ and $a^*_{\delta p}$ and ensuring that this sum does not exceed b_δ for all LA-SRI $\delta \in \Delta_R$.
- (31f): We enforce that the edges selected for each family l describe a set of routes. This means that for every family l, that the total weight of incoming edges is equal to that of the outgoing edges on all nodes corresponding to customers $u \in N$.
- (31e): We enforce that the the LA arcs selected are consistent with the edges selected for a given family. This is done by ensuring that the number of paths also matches the number of edges starting at a customer u, ending at a customer v, and servicing demand d.

The CG optimization procedure used to solve $\Psi(\Omega, \Delta_R)$ alternates between solving (31) (meaning $\Psi^+(\Omega_R, \Delta_R)$) and adding to Ω_R the lowest reduced cost column $l \in \Omega$. We price over θ (never x) to add in a new column l to Ω_R. Then, we add to the RMP the x terms associated with l. We never consider θ terms when solving the RMP as they are redundant given x. However we include the θ terms in (31) so as to make clear that pricing for (31) is identical to previous pricing problems. Given our solver for $\Psi(\Omega, \Delta_R)$ we solve $\Psi(\Omega, \Delta)$ and produce an optimal solution using the column/row generation approach described in Alg 2.

5.4 Efficient Solution to the Restricted Master Problem

In this section we consider the efficient solution of (31) by adapting the ideas of Principled Graph Management (PGM) (Yarkony and Regan 2022). Observe that (31) may have a very large number of variables as $|\Omega_R|$ grows and for problems where $|\Omega_y|$ is large for some $y \in Y$; thus making the solution to (31) at each iteration

of CG intractable. Thus we seek to construct a small set of edges, paths denoted $\hat{E}^l \subseteq E^l \quad \forall l \in \Omega_R, \quad \hat{\Omega}^l_y \subseteq \Omega^l_y \quad (y \in Y^l \ \forall l \in \Omega_R)$ respectively s.t. solving (31) over these terms (denoted $\Psi^+(\Omega_R, \Delta_R, \{\hat{E}^l\}, \{\hat{\Omega}^l_y\})$) yields the same solution as (31); where for short hand we use $\{\hat{E}^l\}, \{\hat{\Omega}^l\}$ to describe the $\hat{E}^l, \hat{\Omega}^l_y$ for each $l \in \Omega_R, (y \in Y^l, l \in \Omega_R)$. We define $\Psi^+(\Omega_R, \Delta_R, \{\hat{E}^l\}, \{\hat{\Omega}^l_y\}))$ formally below.

$$\Psi^+(\Omega_R, \Delta_R, \{\hat{E}^l\}, \{\hat{\Omega}^l_y\}) = \min_{x \geq 0} \sum_{\substack{l \in \Omega_R \\ y \in Y^l \\ p \in \hat{\Omega}^l_y}} c_p x^l_p + \sum_{\substack{l \in \Omega_R \\ u \in N \\ d_0 \geq d \geq d_u \\ (-1,d_0),(u,d) \in \hat{E}^l}} c_{-1u} x^l_{(-1,d_0),(u,d)} \tag{32a}$$

$$\sum_{l \in \Omega_R} \sum_{ij \in \hat{E}^l} [i = (-1, d_0)] x^l_{ij} \leq K \quad [-\pi_0] \tag{32b}$$

$$\sum_{\substack{l \in \Omega_R \\ y \in Y^l \\ p \in \hat{\Omega}^l_y}} x^l_p a_{up} \geq 1 \quad \forall u \in N \quad [\pi_u] \tag{32c}$$

$$\sum_{\substack{l \in \Omega_R \\ y \in Y^l \\ p \in \hat{\Omega}^l_y}} a^*_{\delta p} x^l_p \leq b_\delta \quad \forall \delta \in \Delta_R \quad [-\pi_\delta] \tag{32d}$$

$$\sum_{p \in \hat{\Omega}^l_y} x^l_p = \sum_{\substack{ij \in \hat{E}^l \\ i = u, d_1 \\ j = v, d_1 - d}} x^l_{ij} \quad \forall y \in Y^l; y = (u, v, d), l \in \Omega_R, i \neq (-1, d_0), \tag{32e}$$

$$\sum_{ij \in \hat{E}^l} x^l_{ij} = \sum_{ji \in \hat{E}^l} x^l_{ji} \quad \forall i \in I^0, \quad u_i \notin \{-1, -2\}, l \in \Omega_R \tag{32f}$$

To solve (31) we alternate between the following two steps.

- Solve $\Psi^+(\Omega_R, \Delta_R, \{\hat{E}^l\}, \{\hat{\Omega}^l_y\})$ producing the dual solution π. This is fast since the sets $\hat{\Omega}^l_y$ and $\hat{E}^l$ are generally much smaller than Ω^l_y, E^l respectively.
- Iterate over $l \in \Omega_R$ and compute the lowest reduced cost route in Ω_l denoted $\hat{l}^l$. The computation of $\hat{l}^l$ is a simple shortest path computation (not a resource constrained shortest path problem) and corresponds to the lowest cost path on the graph with edge set E^l where the edge weights are defined as follows.

$$\bar{c}_{(-1,d_0),(u,d)} = c_{-1,u} + \pi_0 \tag{33a}$$

$$\bar{c}_{ij} = \min_{\substack{p \in \Omega^l_y \\ y = (u, v, d_1 - d_2)}} \bar{c}_p \quad \forall i = (u, d_1), j = (v, d_2), i, j \in E^l \tag{33b}$$

$$\bar{c}_p = c_p - \sum_{u \in N} a_{up} \pi_u + \sum_{\delta \in \Delta_R} a^*_{\delta p} \pi_\delta \quad \forall p \in \Omega^l_y, y \in Y^l \tag{33c}$$

Observe that via the ordering of β^l that the route $\hat{l}^l$ must be elementary as all paths from source to sink describe elementary routes. If $\hat{l}^l$ has negative reduced cost then we update the edges,paths associated with route l. We denote the edges used in this route as $E^l(\hat{l})$ and the paths used in this route for a given y in Y^l as $\Omega^l_y(\hat{l})$. Next we augment each $\hat{E}^l, \hat{\Omega}^l_y (\forall y \in Y^l)$ with the edges with $E^l(\hat{l}^l)$ and $\Omega^l_y(\hat{l}^l)$. When we find that $\hat{l}^l$ has non-negative reduced cost for each $l \in \Omega_R$, we terminate since we have solved (31) optimally.

We refer to the operation of generating edge weights and computing the associated shortest path as the RMP-Shortest Path computation (RMP-SP).

We must initialize the $\{\hat{\Omega}^l_y\}, \{\hat{E}^l\}$ terms in order to ensure that (32) has a feasible solution. In experiments we found that using $\hat{\Omega}^l_y, \hat{E}^l$ to be the terms generated during RMP-SP for a route l over the entire course of CG optimization thus far worked well. We could just as easily use all edges that are associated with

non-zero x^l_{ij}, x^l_p values in the final solution produced last time (31) was solved using (32). In our experiments (which had no maximum number of vehicles) prior to the first iteration of CG we included all edges and paths corresponding to the route starting at the depot, visiting a single customer u, and then returning to the depot (for each $u \in N$). This means that for the initial l in CG, denoted l_0 we define $\hat{E}^{l_0}$ to include $(-1, d_0), (u, d_u)$,and $(u, d_u), (-2, 0)$ for each $u \in N$; and set $\hat{\Omega}^{l_0}_y$ as empty except for terms $y = (u, -2, d_u)$ (for each $u \in N$) which is set to contain the path starting at u ending at -2 and visiting no intermediate customers.

In Alg 3 we provide a formal description of our fast optimization of (31) using RMP-SP which we annotate below.

- In Line 1 we initialize $\Omega_R, \{\hat{E}^l\}, \{\hat{\Omega}^l_y\}$ from the user.
- In the loop defined in Lines 2- 15 we construct a sufficient set of $\{\hat{E}^l\}, \{\hat{\Omega}^l_y\}$ to solve $\Psi^+(\Omega_R, \Delta_R)$ exactly meaning $\Psi^+(\Omega_R, \Delta_R, \hat{E}, \hat{\Omega}) = \Psi^+(\Omega_R, \Delta_R)$.
  - In Line 3 we solve $\Psi^+(\Omega_R, \Delta_R, \{\hat{E}^l\}, \{\hat{\Omega}^l_y\})$ as an LP producing π, x terms.
  - In Lines 4-10 we iterate over $l \in \Omega_R$ and augment $\{\hat{E}^l\}, \{\hat{\Omega}^l_y\}$ using RMP-SP.
    - In Lines 5 we generate the lowest reduced cost column in the family Ω_l which we denote as $\hat{l}^l$. It is important to note that this is a simple shortest path and not a hard pricing problem.
    - In Line 6 we determine if $\hat{E}^l, \hat{\Omega}^l_y$ need to be augmented which is true if $\hat{l}^l$ has negative reduced csost. In Line 7 we add any edges used in $\hat{l}^l$ to $\hat{E}^l$ expanding the set of edges. In Line 8 we similarly, add any paths used in $\hat{l}^l$ to the corresponding $\hat{\Omega}^l_y$.
  - In Line 11 we terminate the loop once no path in any graph E^l for $l \in \Omega_R$ has negative reduced cost, since at this point we have solved $\Psi^+(\Omega_R, \Delta_R)$ exactly.
- In Lines 12- 15 we describe an optional step to ensure that $\hat{E}^l, \hat{\Omega}^l_y$ do not grow too large causing computational difficulty. In practice we found this unnecessary. In this step we remove all terms for which the corresponding x^l_{ij},x^l_p are zero valued for the solution of (31) for the final solution to (32).

Given Alg 3 we describe the complete optimization approach for solving $\Psi(\Omega, \Delta)$ using column/row generation in Alg 4 with annotation below.

- In Lines 1-2 we initialize from the user the set of columns in Ω_R. In our experiments this is a single column l randomly generated with edge,path sets $\hat{E}^l, \hat{\Omega}^l_y \quad \forall y \in Y^l$ that can be used to generate all routes containing a single customer (one for each $u \in N$).

Algorithm 3 Fast Solver for (31)

1: $\Omega_R, \{\hat{E}^l\}, \{\hat{\Omega}^l_y\} \leftarrow$ from user
2: **repeat**
3: $x, \pi \leftarrow$ Solve (32) over $\hat{E}^l, \hat{\Omega}^l_y$
4: **for** $l \in \Omega_R$ **do**
5: $\hat{l}^l \leftarrow \arg\min_{l \in \Omega_l} \bar{c}_l$. Fast and easy shortest path calculation (not resource constrained).
6: **if** $\bar{c}_{\hat{l}^l} < 0$ **then**
7: $\hat{E}^l \leftarrow \hat{E}^l \cup E^l(\hat{l}^l)$
8: $\hat{\Omega}^l_y \leftarrow \hat{\Omega}^l_y \cup \Omega^l_y(\hat{l}^l)$ for each $y \in Y^l$
9: **end if**
10: **end for**
11: **until** $\bar{c}_{\hat{l}^l} \geq 0$ for all $l \in \Omega_R$
12: **for** $l \in \Omega_R$ **do**
13: $\hat{E}^l \leftarrow \{ij \in \hat{E}^l \quad s.t. \quad x^l_{ij} > 0\}$ OPTIONAL NOT USED IN EXPERIMENTS
14: $\hat{\Omega}^l_y \leftarrow \{p \in \hat{\Omega}^l_y \quad s.t. \quad x^l_y > 0\}$ for each $y \in Y^l$. OPTIONAL NOT USED IN EXPERIMENTS
15: **end for**
16: Return last $x, \pi, \{\hat{E}^l\}, \{\hat{\Omega}^l_y\}$

- In Line 3 we initialize the set Δ_R to be empty.
- In Lines 4-12 we solve the master problem over all Ω, Δ, which is written $\Psi(\Omega, \Delta)$.
  - In Lines 5-9 we solve $\Psi(\Omega, \Delta_R)$.
    - In Line 6 we solve $\Psi^+(\Omega_R, \Delta_R)$ using Alg 3.
    - In Line 7 we find the lowest reduced cost route l_* using the LA route algorithm (Mandal et al. 2022). Recall that $\bar{c}_l$ is defined in (28) for each $l \in \Omega$.
    - In Line 8 we add the column l_* to the Ω_R set.
    - In Line 9 we terminate CG when the lowest reduced cost route in Ω has non-negative reduced cost.
  - In Lines 10-11 we find one or more violated LA-SRI which are then added to Δ_R. In our experiments we add the 30 most violated LA-SRI (or all violated LA-SRI if less than 31 are violated)
  - In Line 12 we terminate optimization when no LA-SRI are violated since we have solved $\Psi(\Omega, \Delta)$ optimally.
- Line 13: Return x providing a primal solution that can be fed into branch & price.

It is often useful to generate an approximate optimal solution to the optimal integer linear programming (ILP) solution to CVRP. To do this efficiently we solve the RMP in $\Psi^+(\Omega_R, \Delta_R, \{\hat{E}^l\}, \{\hat{\Omega}^l\})$ as described in (32) as an ILP given $\{\hat{E}^l\}, \{\hat{\Omega}^l_y\}$ generated during Alg 4. We found that this produces quality integer solutions in practice.

6 Experiments

In this section we demonstrate the value of LA-SRI to tighten the expanded LP relaxation for the Capacitated Vehicle Routing Problem (CVRP) without altering the structure of the pricing problem. We consider a dataset of CVRP problem instances that vary by numbers of customers and vehicle capacity. We perform experiments using Alg 4 with different parameters for the LA-SRI (meaning the of range of $|N_\delta|$ terms for $\delta \in \Delta$) and the sizes of LA neighbor sets (sizes of N_u). Larger sizes of sets Δ,$|N_u|$ have the potential to tighten the expanded LP relaxation at the expense of greater computation time. In these experiments, pricing and determination of the lowest reduced cost route is solved using Decremental State Space Relaxation (Righini and Salani 2008, 2009) (DSSR) over LA routes as described in Section 3.2.3.

We organize this section as follows. In Section 6.1 we consider the parameterizations used for our column/row generation (CRG) solver as described in Alg 4 and CVRP problems in our data set. In Section 6.2 we discuss the hardware and software implementation details for our experiments. In Section 6.3 we quantify tightening of the LP relaxation achieved by adding LA-SRI. In Section 6.4 we describe experiments that we conduct, which show the value of LA-SRI in the context of efficiently tightening the expanded LP relaxation for CVRP. In Section 6.5 we discuss the results of these experiments.

Algorithm 4 Complete Solver for the Master Problem

1: $\Omega_R, \leftarrow$ from user
2: $\hat{E}^l, \hat{\Omega}^l_y$ from user for all $l \in \Omega_R$,$(y \in Y^l, l \in \Omega_R)$ respectively
3: $\Delta_R \leftarrow \{\}$
4: **repeat**
5: **repeat**
6: $x, \pi, \{\hat{E}^l\}, \{\hat{\Omega}^l_y\} \leftarrow$ Solve for (31), using Alg 3
7: $l_* \leftarrow \arg\min_{l \in \Omega} \bar{c}_l$ via the LA routes pricing algorithm
8: $\Omega_R \leftarrow \Omega_R \cup l_*$
9: **until** $\bar{c}_{l_*} \geq 0$
10: $\delta_* \leftarrow$ Most violated constraint over (31d) given x. More than one violated constraint can be added.
11: $\Delta_R \leftarrow \Delta_R \cup \delta_*$
12: **until** (31d) is not violated for δ_*
13: Return last x generated by line 6. This x can be used inside branch & price.

6.1 Algorithms and Data Sets Compared

In this section, we describe the problems in our data set and the CRG solver parameterizations considered. We considered CVRP problem instances with the following parameter possibilities for (number of customers,vehicle capacity): (20,4);(30,5);(40,8). Each customer has unit demand in each CVRP problem instance. We generate 10 problem instances for each CVRP parameter possibility (meaning 10 for each of (20,4);(30,5);(40,8)). Each instance randomizes the locations of customers and the starting and ending depot.

We used a total of twelve parameterizations for our CRG solver described in Alg 4. The trade off is an increase in computation time vs possible improvement in the tightness of the LP relaxation. Larger LA neighbor sets and larger classes of LA-SRI can tighten the LP relaxation but may increase computation time. Each CRG parameterization is associated with a different number of LA neighbors per customer, and different classes of LA-SRI used. We set LA neighbor sets for a given customer u with size x to be the x closest customers (by spatial distance) to u (excluding u since $u \notin N_u$ by definition). We had one solver parameterization for each pair of unique choices for the number of LA neighbors and the LA-SRI parameterization. We considered four choices for the number of LA neighbors per customer $(4, 6, 8, 10)$ and three choices for LA-SRI parameter values denoted a, b, c which are described below using: $\Delta_{\alpha,\beta}$ to denote the set of LA-SRI consisting of α (meaning $|N_\delta| = \alpha$) unique customers in N and for which $m_\delta = \beta$.

- option a: $\Delta \leftarrow \Delta_{3,2}$.
- option b: $\Delta \leftarrow \Delta_{3,2} \cup \Delta_{4,3}$
- option c: $\Delta \leftarrow \Delta_{3,2} \cup \Delta_{5,2} \cup \Delta_{5,3}$. Note that by definition of LA-SRI in (16) $\Delta_{4,3}$ is dominated by $\Delta_{5,3}$ and hence is not included in option c. By dominated we mean that any solution satisfying all LA-SRI in $\Delta_{5,3}$ will also satisfy all LA-SRI in $\Delta_{4,3}$ but not vice versa.

We use F which we index by f to denote the set of CRG solver parameterizations.

6.2 Implementation Details

All code is implemented in MATLAB and the LP solved in the RMP at each iteration of Alg 3 (which is called in Alg 4) are solved using the MATLAB "linprog" solver with default options. All LPs are solved from scratch each time. In future work, we intend to use CPLEX and C++/C and not solve LPs from scratch. All experiments were run on a 2014 Macbook pro running Matlab 2016. For pricing we used the LA routes code of (Mandal et al. 2022) using the specific efficiencies in that code with edge weights generated using (7).

To solve the ILP we use the MATLAB "intlinprog" solver with default options and maximum run time of 180 seconds.

6.3 Quantifying Tightening of the LP Relaxation

In this section we quantify the amount of tightening of the LP relaxation using a measure of "relative increase", which describes how much of the gap between the optimal integer solution and the lower bound (the LP value) is closed by adding LA-SRI. We define the relative increase of a CRG solver parameterization to be the improvement of the tightness of the solution induced by adding LA-SRI relative to the best known integer solution. We define this term formally using the following terms.

- $\Psi(\Omega)$: We use $\Psi(\Omega)$ to denote the LP value prior to adding any LA-SRI. This is identical for all CRG solvers $f \in F$.
- $\Psi^f(\Omega, \Delta)$: We use $\Psi^f(\Omega, \Delta)$ to denote the LP value for the CRG solver f after all LA-SRI are satisfied.
- UB: We use UB^f to denote the upper bound generated by CRG solver f at termination of column/row generation as done by solving the RMP as an ILP.
- UB: We use UB to denote the lowest cost upper bound generated using terms from any CRG solver for a particular CVRP problem instance. Thus $UB = \min_{f \in F} UB^f$.

We now define relative increase below.

$$\text{relative increase}(f) = \frac{UB - \Psi^f(\Omega, \Delta)}{UB - \Psi(\Omega)} \quad \forall \quad UB > \Psi(\Omega), f \in F \tag{34}$$

Observe that a relative increase(f) of 1 indicates that the CRG solver f makes the LP relaxation tight. A relative increase of 0 indicates that CRG solver f does not tighten the LP relaxation using the LA-SRI. All other relative increase values lie linearly in between $[0, 1]$ depending on the LP value of CRG solver f. Observe that (34) is not defined for the case where the LP relaxation is tight prior to adding any LA-SRI, since the denominator (and the numerator) would be 0.

6.4 Experiments Conducted

We split our experiments into two categories. The first category, represented by plots in 1, shows the comparisons in time taken averaged over problem instances for various components for CRG solver parameterizations. The second category, represented by plots in 2, shows the comparisons in relative increase for various CRG solver parameterizations for CVRP problem instances.

On the top right of every figure, there is a box representing the parameter values for the CRG solver, which is given by a distinct mark and color combination on the plot. Furthermore, every plot has a title describing the type of comparison (either "timing" or "relative increase") and the CVRP parameter values, where the number of customers is denoted by N, and the capacity is denoted by d_0. Results are described conditioned on problem size (N, d_0).

In plots shown by Fig 1, the y-axis provides the average time of any such timing category over the 10 CVRP problem instances used for a given CVRP parameterization; where the timing categories are written below and associated with the x-axis.

- Total time: x=1 corresponds to the total optimization time taken, which is the sum of the terms below.
- Pricing time: x=2 corresponds to the time used in during pricing.
- RMP-LP time: x=3 corresponds to the time used for calls to the LP solver for $\Psi^+(\Omega_R, \Delta_R, \hat{E}, \hat{\Omega})$ generated during the RMP solver of $\Psi^+(\Omega_R, \Delta_R)$ as described in Section 5.4.
- RMP-Shortest Path time: x=4 corresponds to the time used for finding shortest paths (and the associated overhead) during the generation of $\hat{E}^l$ and $\hat{\Omega}^l$ as described in 5.4.
- Separation time: x=5 corresponds to the time used for identifying violated LA-SRI.
- ILP time: x=6 corresponds to the time required for solving the ILP to generate a feasible integer solution.
- Pre-computation time: x=7 corresponds to the time required for pre-computing c_p as described in Alg 1.

In plots shown by Fig 2, the x-axis is used to represent the problem instance for the CVRP parameterization given by the title. The y-axis provides the relative increase of the solution given by a CRG solver's parameterization for the appropriate problem instance.There is also a blue circle in the same location of the CRG solver parameterization which provides the greatest relative increase for a particular problem instance and CVRP parameterization. It is important to note that some problem instances do not have associated CRG solver values plotted; this is because we ignore problem instances where the CRG solver provides a tight solution before introducing LA-SRI as the relative increase would be undefined.

6.5 Analysis

In this section we analyze the results for the experiments conducted in Section 6.4.

In results shown by Fig 1, we see for each CVRP parameterization that a large amount of optimization time is spent on pricing, with the time spent on solving the LP during RMP becoming increasingly negligible for CVRP parameterizations involving larger amounts of customers. It is important to note that although the time taken to find the shortest path for stabilization does take significant time, the time can be reduced by parallelizing the shortest path computation. Similarly, the time taken for separation to generate new LA-SRI sets can be reduced by using parallelization. As a result, LA-SRI computation time will be even more heavily dominated by other components of optimization (like pricing) in more sophisticated versions.

In results shown by 2, we see for each CVRP parametrization that the use of LA-SRI often improves the tightness of the LP solver's solution. However, we occasionally see that there are LA solver parameterizations

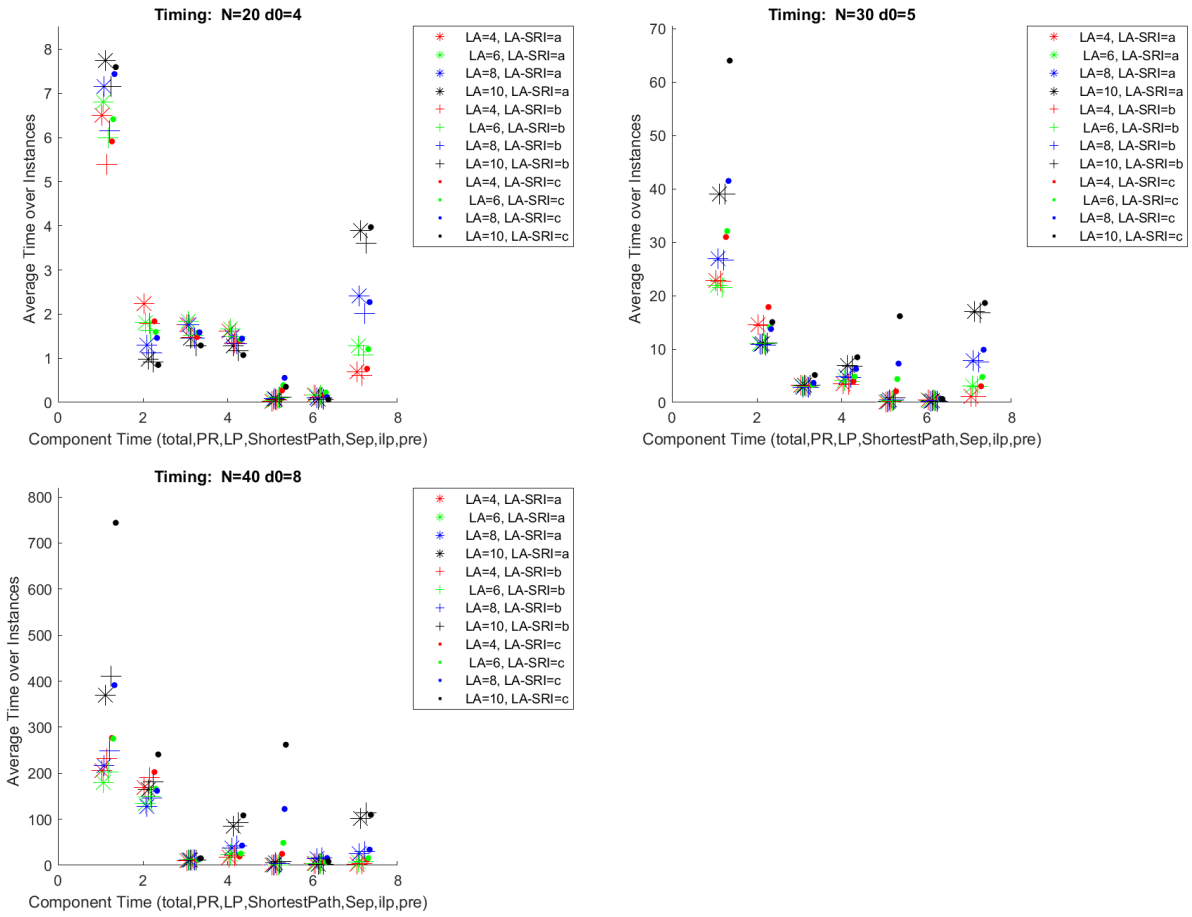

Figure 1: In these plots, we describe the amount of time taken by each component of optimization for each solver. The y-axis describes the average time taken in a procedure for a CRG solver parameterization. Each data point describes the average time taken by a given CRG solver parameterization for a given component of optimization on problems in our dataset for a given CVRP parameterization. The x-axis is used to represent categories in time taken for the CRG solver. We use numbers to represent these categories on the x-axis, and we explicitly define them here: Total x=1; Pricing x=2; RMP-LP x=3; RMP-Shortest Path x=4; Separation x=5; ILP x=6; Preprocessing x=7.

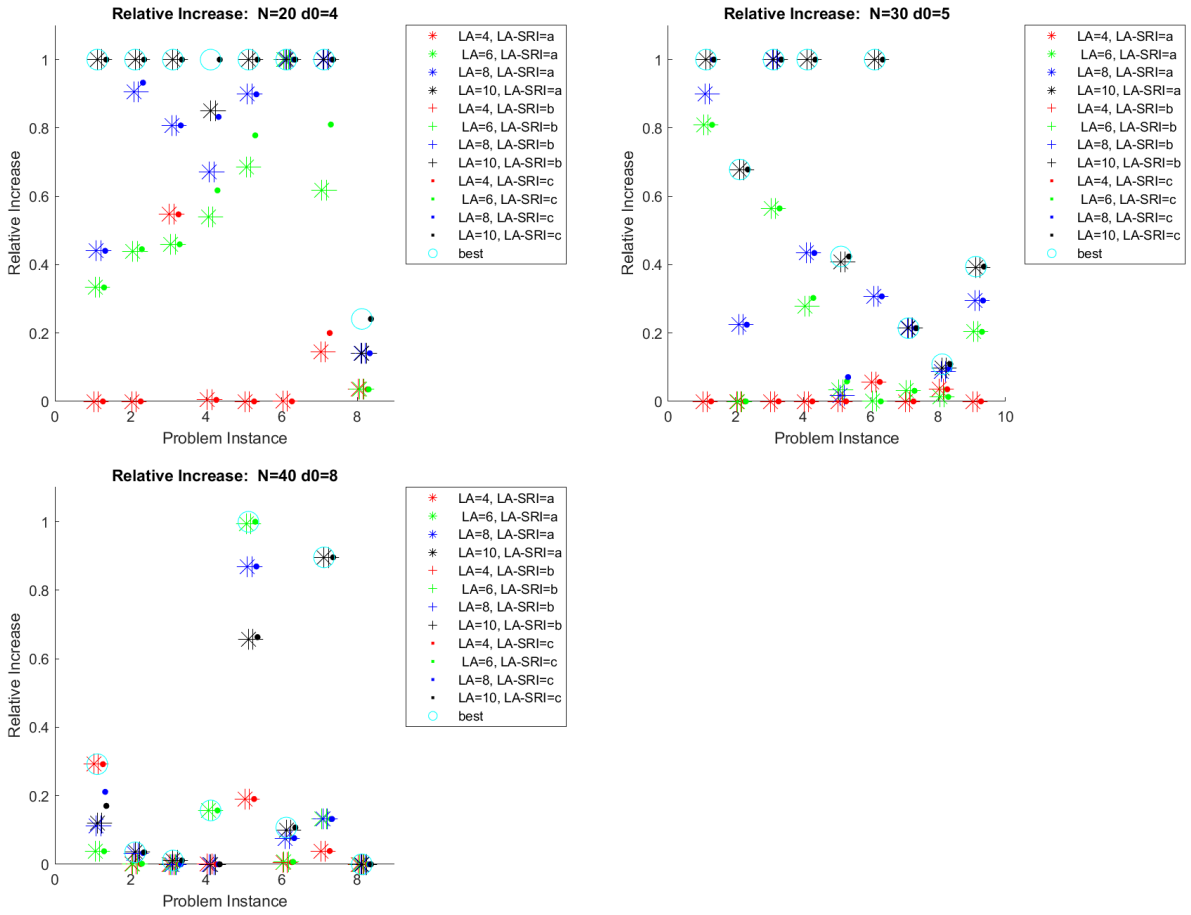

Figure 2: In these plots, we describe the relative solution tightness for each CRG solver parameterization. The y axis describes the relative increase in the tightness by a given solver parameterization. Each data point describes the "relative increase" in tightness for the CRG solver parameterization described by the legend for a CVRP problem instance for a given CVRP parameterization. The relative increase is defined by (34). Observe that a y-coordinate value of 1 for a given problem instance x and solver f indicates that the LP relaxation became tight when LA-SRI are added for the problem instance x for solver f. We use the "best" label in the legend to refer to the largest relative increase over all solvers $f \in F$. We also ignore problem instances where the LP solver provides a tight solution before introducing LA-SRI.

where the use of LA-SRI does not improve the tightness of the LP solver's solution. When looking at the effectiveness of CRG solver parameterizations with the same LA-SRI option, we see that generally having larger LA neighbor sets improves tightness, but there are exceptions to this rule as described in Section 4.4.The use of more LA-SRI in the optimization can lead to tighter solutions, but we observe that the LA-SRI of option (a) often aligns with the tightest possible solution and is computationally fastest to solve over.

Thus we conclude by stating that the use of LA-SRI tends not to significantly add to the time taken for optimization, but often does tighten the corresponding LP relaxation.

7 Conclusions

In this document we adapt subset row inequalities (SRI) (Jepsen et al. 2008) to efficiently tighten the expanded LP relaxation for vehicle routing problems, producing Local Area-subset row inequalities (LA-SRI). We integrate LA-SRI alongside LA routes in a manner that ensures that the structure of the pricing problem is not altered (Mandal et al. 2022). We demonstrate that our formulation with LA-SRI allows for an accelerated solution based on Graph Generation/Principled Graph Management (GG/PGM) (Yarkony et al. 2021, Yarkony and Regan 2022). We apply our approach to the Capacitated Vehicle Routing Problem (CVRP), though our approach can be applied to other vehicle routing problems (Desrochers et al. 1992, Costa et al. 2019). We demonstrate that the expanded LP relaxation is indeed significantly tightened using LA-SRI and that computation of an optimal LP solution remains tractable.

We observe that for some problem instances, the number of LA neighbors alters the improvement in the LP value. Furthermore, an increase in the size of LA neighbor sets does not monotonically tighten the LP. However, larger LA neighbor sets do increase computation time. Thus, in future work we seek to develop an algorithm that constructs the LA neighbor sets for customers in an efficient manner so as to maximally tighten the LP. In this process, we would aim to maximally tighten the LP bound given that the number of LA neighbors per customer does not exceed a user defined value (thus describing a computation budget).

In future work, we intend to explore the use of time windows in our formulations. This may involve different LA neighbor settings depending on the time when the vehicle leaves a customer.

We also intend to use the original SRI in our formulation when they are violated by the LP solution but the associated LA-SRI are not. In addition we can adapt other classes of valid inequalities to use with LA arcs as we did SRI such as capacity inequalities (Archetti et al. 2011). These adaptations will be done to allow the class of valid inequalities considered to easily fit into the existing pricing procedure, making the addition of such valid inequalities computationally efficient.

References

C. Archetti, N. Bianchessi, and M. G. Speranza. A column generation approach for the split delivery vehicle routing problem. *Networks*, 58(4):241–254, 2011.

R. Baldacci, A. Mingozzi, and R. Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5):1269–1283, 2011.

C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1996.

C. Barnhart, C. A. Hane, and P. H. Vance. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326, 2000.

H. Ben Amor, J. Desrosiers, and J. M. Valério de Carvalho. Dual-optimal inequalities for stabilized column generation. *Operations Research*, 54(3):454–463, 2006.

L. Costa, C. Contardo, and G. Desaulniers. Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science*, 26(1), 2019.

R. Dechter and J. Pearl. Generalized best-first search strategies and the optimality of a. *Journal of the ACM (JACM)*, 32(3):505–536, 1985.

G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors. *Column Generation*. Springer, New York, 1st edition, 2005.

M. Desrochers, J. Desrosiers, and M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40(2):342–354, 1992.

J. Desrosiers and M. E. Lübbecke. A primer in column generation. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, pages 1–32. Springer, New York, NY, 2005.

J. Desrosiers and M. E. Lübbecke. Branch-price-and-cut algorithms. *Encyclopedia of Operations Research and Management Science. John Wiley & Sons, Chichester*, pages 109–131, 2011.

O. Du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized column generation. *Discrete Mathematics*, 194 (1-3):229–237, 1999.

D. Feillet. A tutorial on column generation and branch-and-price for vehicle routing problems. *4or*, 8(4):407–424, 2010.

P. Gilmore and R. Gomory. A linear programming approach to the cutting-stock problem. *Operations Research*, 9 (6):849–859, 1961.

N. Haghani, C. Contardo, and J. Yarkony. Smooth and flexible dual optimal inequalities. *Informs Journal on Optimization, in press, arXiv preprint arXiv:2001.02267*, 2021.

M. Jepsen, B. Petersen, S. Spoorendonk, and D. Pisinger. Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56(2):497–511, 2008.

U. Mandal, A. Regan, and J. Yarkony. Local area routes for vehicle routing problems. *arXiv preprint arXiv:2207.04520*, 2022.

R. E. Marsten, W. Hogan, and J. W. Blankenship. The boxstep method for large-scale optimization. *Operations Research*, 23(3):389–405, 1975.

T. K. Ralphs, L. Kopman, W. R. Pulleyblank, and L. E. Trotter. On the capacitated vehicle routing problem. *Mathematical programming*, 94(2):343–359, 2003.

G. Righini and M. Salani. New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks: An International Journal*, 51(3):155–170, 2008.

G. Righini and M. Salani. Decremental state space relaxation strategies and initialization heuristics for solving the orienteering problem with time windows with dynamic programming. *Computers & Operations Research*, 36 (4):1191–1203, 2009.

D. M. Ryan and B. A. Foster. An integer programming approach to scheduling. *Computer scheduling of public transport urban passenger vehicle and crew scheduling*, pages 269–280, 1981.

S. Wang, S. Wolf, C. Fowlkes, and J. Yarkony. Tracking objects with higher order interactions via delayed column generation. In *Proc. 20th International Conference on Artificial Intelligence and Statistics*, pages 1132–1140, Fort Lauderdale, Florida, 2017.

J. Yarkony and A. Regan. Principled graph management. *arXiv preprint arXiv:2202.01274*, 2022.

J. Yarkony, Y. Adulyasak, M. Singh, and G. Desaulniers. Data association via set packing for computer vision applications. *Informs Journal on Optimization*, 2(3):167–191, 2020.

J. Yarkony, N. Haghani, and A. Regan. Graph generation: A new approach to solving expanded linear programming relaxations. *arXiv preprint arXiv:2110.01070*, 2021.

A Restricted Master Problem Equivalence

In this section we establish that $\Psi^+(\Omega_R, \Delta_R) = \Psi(\cup_{l\in\Omega_R}\Omega_l, \Delta_R)$. We establish that in two parts. In Section A.1 we demonstrate that $\Psi^+(\Omega_R, \Delta_R) \leq \Psi(\cup_{l\in\Omega_R}\Omega_l, \Delta_R)$. In Section A.2 we establish that $\Psi^+(\Omega_R, \Delta_R) \geq \Psi(\cup_{l\in\Omega_R}\Omega_l, \Delta_R)$. Since $\Psi^+(\Omega_R, \Delta_R) \leq \Psi(\cup_{l\in\Omega_R}\Omega_l, \Delta_R)$ and $\Psi^+(\Omega_R, \Delta_R) \geq \Psi(\cup_{l\in\Omega_R}\Omega_l, \Delta_R)$ then $\Psi^+(\Omega_R, \Delta_R) = \Psi(\cup_{l\in\Omega_R}\Omega_l, \Delta_R)$.

A.1 First Side of the Inequality

In this section we establish that $\Psi^+(\Omega_R, \Delta_R) \geq \Psi(\cup_{l\in\Omega_R}\Omega_l, \Delta_R)$. Consider the optimal solution to (31) denoted (x, θ). Observe that if the x terms are zero valued then $\Psi^+(\Omega_R, \Delta_R) = \Psi(\Omega_R, \Delta_R)$ and $\Psi(\Omega_R, \Delta_R) \geq \Psi(\cup_{l\in\Omega_R}\Omega_l, \Delta_R)$ by definition thus establishing the claim.

We now transform the solution x, θ to a solution to $\Psi(\cup_{l\in\Omega_R}\Omega_l, \Delta_R)$ with identical cost to that of x, θ. Each step of this process decreases the number of x terms with non-zero value. We now describe an individual step of this process. If x is not zero valued by (31f), there must exist a path of non-zero valued x^l_{ij} terms starting at the source node $(-1, d_0)$ and ending at the sink node $(-2, 0)$ for some $l \in \Omega_R$. Select any such path on the graph associated with l, and let the edges on the path be denoted $\hat{E}^l$. By (31e) for each $ij \in \hat{E}^l$ (except the edge including the source) there must exist a $p \in \Omega^l_y$ (for $i = (u, d_1)$,$j = (v, d_2)$, $y = (u, v, d_1 - d_2)$

s.t. $x^l_p > 0$. For each $ij \in \hat{E}^l$ (excluding the edge connected to the $(-1, d_0)$) let p_{ij} be any such $p \in \Omega^l_y$ s.t. $x^l_p > 0$. Let $\hat{l}$ denote a route corresponding to crossing all $ij \in \hat{E}^l$ and using the paths in p_{ij} as the intermediate nodes. Note that $\hat{l}$ is an elementary feasible route that lies in Ω_l (but need not lie in Ω_R) by definition of E^l, Y^l. Now consider the following altered solution given a tiny constant $\alpha > 0$.

$$\theta_{\hat{l}} \leftarrow \theta_{\hat{l}} + \alpha \tag{35a}$$

$$x^l_{ij} \leftarrow x^l_{ij} - \alpha \quad \forall ij \in \hat{E}^l \tag{35b}$$

$$x^l_p \leftarrow x^l_p - \alpha \quad \forall p \in \cup_{\substack{ij \in \hat{E}^l \\ i \neq (-1,d_0)}} p_{ij} \tag{35c}$$

Observe that the change in cost is zero by the definition of the cost of a route.Thus we need to establish that a non-zero value α exists s.t. the change induced in (35) is feasible. Below we set α to the largest possible value s.t. (35) does not alter non-negativity.

$$\alpha \leftarrow \min\{\min_{ij \in \hat{E}^l} x^l_{ij}, \min_{\substack{ij \in \hat{E}^l \\ i \neq (-1,d_0)}} x^l_{p_{ij}})\} \tag{36}$$

Since all x terms in (36) are positive then α is positive. Observe that this solution has fewer non-zero valued x terms than prior to the update.

A.2 Second Side of the Inequality

In this section we establish that $\Psi^+(\Omega_R, \Delta_R) \geq \Psi(\cup_{l \in \Omega_R} \Omega_l, \Delta_R)$ using proof by contradiction. Consider any solution producing optimal terms x, θ to $\Psi(\cup_{l \in \Omega_R} \Omega_l, \Delta_R)$. If $\Psi^+(\Omega_R, \Delta_R) \geq \Psi(\cup_{l \in \Omega_R} \Omega_l, \Delta_R)$ then a contradiction is created establishing the claim.

We construct as solution x (with zero valued θ) as follows to $\Psi^+(\Omega_R, \Delta_R)$ with the same objective as that of θ over $\Psi(\cup_{l \in \Omega_R} \Omega_l, \Delta_R)$. Let $\Omega_{R2} = \cup_{l \in \Omega_R} \Omega_l$. For any $\theta_l > 0$ let Q_l be any route in Ω_R for which $l \in \Omega_{Q_l}$ (where Ω_{Q_l} is simply $\Omega_{\hat{l}}$ where $\hat{l} = Q_l$) . We now construct x as follows.

$$x^{\hat{l}}_{ij} \leftarrow \sum_{\substack{l \in \Omega_{R2} \\ \hat{l} = Q_l}} a_{ijl} \theta_l \quad \forall (ij \in E^{\hat{l}}, \hat{l} \in \Omega_R) \tag{37a}$$

$$x^{\hat{l}}_p \leftarrow \sum_{\substack{l \in \Omega_{R2} \\ \hat{l} = Q_l}} a_{pl} \theta_l \quad \forall p \in \Omega^{\hat{l}}_y, y \in Y^{\hat{l}}, \hat{l} \in \Omega_R \tag{37b}$$

Observe that this solution is feasible and has cost identical to $\Psi(\cup_{l \in \Omega_R} \Omega_l, \Delta_R)$ creating a contradiction hence establishing the claim.