A Constant Factor Approximation for Regret-Bounded Vehicle Routing

Zachary Friggstad, Chaitanya Swamy

The University of Waterloo

Flexible Network Design, Toronto
July 29, 2013
Vehicle Routing

A typical Vehicle Routing Problem (VRP): Given one or more vehicles located at some depots, find routes for them to visit some clients.

Travel distance often factors into the objective or constraints, e.g. TSP, Orienteering, Distance-Constrained VRP, Capacitated VRP, ...
Vehicle Routing

A typical Vehicle Routing Problem (VRP): Given one or more vehicles located at some depots, find routes for them to visit some clients.

Travel distance often factors into the objective or constraints, e.g. TSP, Orienteering, Distance-Constrained VRP, Capacitated VRP, ... 

However, this does not differentiate between clients close to the depot and clients far from the depot.
A Client-Centric View

We consider a vehicle routing problem with a single depot node $r$.

For a path $P$ starting at $r$ and for some $v \in P$, define the regret of $v$ along $P$ to be

$$d_P(v) - d(r, v)$$

This is the distance along $P$ to reach $v$ in excess of the $r - v$ distance delay. Since the $r - v$ distance delay is inevitable, this is a natural way to measure a client's satisfaction.
A Client-Centric View

We consider a vehicle routing problem with a single depot node $r$.

For a path $P$ starting at $r$ and for some $v \in P$, define the regret of $v$ along $P$ to be

$$d_P(v) - d(r, v)$$

This is the distance along $P$ to reach $v$ in excess of the $r - v$ distance.

Since the $r - v$ distance delay is inevitable, this is a natural way to measure a client’s satisfaction.
The Regret-Bounded Vehicle Routing Problem

Input
- Locations $V \cup \{r\}$ with $r$ being the root/depot.
- Symmetric metric distances $d(u, v)$ between locations:
  \[ d(u, v) \leq d(u, w) + d(w, v). \]
- A regret bound $R \geq 0$. 
The Regret-Bounded Vehicle Routing Problem

Input

• Locations \( V \cup \{r\} \) with \( r \) being the root/depot.
• Symmetric metric distances \( d(u, v) \) between locations:
  \[ d(u, v) \leq d(u, w) + d(w, v). \]
• A regret bound \( R \geq 0 \).

Goal

Cover \( V \) with the fewest rooted paths (starting at \( r \)) so that no client has regret more than \( R \) on their covering path.
Previous Work

Bock, Grant, Koenemann, and Sanita, 2011 - “School Bus Problem”

- Greedy Set Cover + Orienteering $\Rightarrow O(\log |V|)$-approximation.
- A 3-approximation in tree metrics.
Previous Work

Bock, Grant, Koenemann, and Sanita, 2011 - “School Bus Problem”

- Greedy Set Cover + Orienteering $\Rightarrow O(\log |V|)$-approximation.
- A 3-approximation in tree metrics.

The notion of regret has been studied before. Approximations for Minimum Excess Path lead to approximations for Orienteering. [Blum et al., 2003; Nagarajan and Ravi, 2007; Chekuri, Korula, and Pál, 2008].
Previous Work

Bock, Grant, Koenemann, and Sanita, 2011 - “School Bus Problem”
- Greedy Set Cover + Orienteering $\Rightarrow O(\log |V|)$-approximation.
- A 3-approximation in tree metrics.

The notion of regret has been studied before. Approximations for Minimum Excess Path lead to approximations for Orienteering. [Blum et al., 2003; Nagarajan and Ravi, 2007; Chekuri, Korula, and Pál, 2008].

Related problem: Distance-Constrained VRP. Cover $V$ using the fewest rooted cycles, each having distance at most $D \geq 0$.

Nagarajan and Ravi, 2008
- An $O(\min(\log D, \log |V|))$-approximation in general.
- A 2-approximation in tree metrics.
Main Result

An Integrality Gap Bound

We consider a configuration-style of LP relaxation.

Theorem

*Given an LP solution with value $k^*$ and polynomial support size, we can efficiently an integral solution which uses at most $(7 + 4\sqrt{3}) \cdot k^* + 1$ paths in polynomial time.*
Main Result

An Integrality Gap Bound

We consider a configuration-style of LP relaxation.

**Theorem**

*Given an LP solution with value $k^*$ and polynomial support size, we can efficiently an integral solution which uses at most $(7 + 4\sqrt{3}) \cdot k^* + 1$ paths in polynomial time.*

A Constant-Factor Approximation

Combining this with the $(2 + \epsilon)$-approximation for solving the LP yields a 28.36-approximation for Regret-Bounded VRP.
Highlights

Highlights:

• The LP is an example of the set-partitioning model for VRP.
  ○ Computationally, this approach has been observed to provide excellent lower bounds in related problems (column generation techniques help solve the LPs in practice) but few theoretical guarantees were known.

• New ideas to deal with regret/excess of a path and rounding configuration LPs in VRP.

• Can be viewed as a special case of Distance-Constrained VRP in a particular asymmetric metric (described soon).
An LP relaxation

Let \( C_R = \{\text{rooted paths } P : d_v(P) - d(r, v) \leq R \text{ for each } v \in P\} \).

\[
\text{minimize : } \sum_{P \in C_R} x_P \\
\text{subject to : } \sum_{P \in C_R \atop v \in P} x_P \geq 1 \quad \forall v \in V
\]

\[x \geq 0\]
An LP relaxation

Let \( C_R = \{ \text{rooted paths } P : d_v(P) - d(r, v) \leq R \text{ for each } v \in P \} \).

\[
\text{minimize : } \sum_{P \in C_R} x_P \\
\text{subject to : } \sum_{P \in C_R} x_P \geq 1 \quad \forall \ v \in V \\
\quad x \geq 0
\]

The dual separation problem is a Point-to-Point Orienteering problem. This has a \((2 + \epsilon)\)-approximation [Chekuri, Korula, and Pál, 2008].

\(\therefore\) we can solve the LP within a factor of \(2 + \epsilon\).
Preliminary Observations

Define the regret metric $d^{\text{reg}}$ over $V \cup \{r\}$ by

$$d^{\text{reg}}(u, v) := d(r, u) + d(u, v) - d(r, v)$$

Observations:

• $d^{\text{reg}}$ is an asymmetric metric.
• $d^{\text{reg}}(r, v) = 0$ for any $v \in V$.
• The $d^{\text{reg}}$-length of a rooted path $P$ is the regret of its endpoint.
• The $d$-length and $d^{\text{reg}}$-length of any cycle are equal.
Preliminary Observations

Define the regret metric $d^{\text{reg}}$ over $V \cup \{r\}$ by

$$d^{\text{reg}}(u, v) := d(r, u) + d(u, v) - d(r, v)$$

Observations:

- $d^{\text{reg}}$ is an asymmetric metric.
- $d^{\text{reg}}(r, v) = 0$ for any $v \in V$.
- The $d^{\text{reg}}$-length of a rooted path $P$ is the regret of its endpoint.
- The $d$-length and $d^{\text{reg}}$-length of any cycle are equal.
Preliminary Observations

In particular

Regret-Bounded VRP in \( d \equiv \) Distance-Constrained VRP in \( d^{reg} \)
Preliminary Observations

In particular

Regret-Bounded VRP in $d \equiv$ Distance-Constrained VRP in $d^{\text{reg}}$

Lemma

Given $\leq \alpha \cdot k^*$ paths covering $V$ with total $d^{\text{reg}}$-cost $\leq \beta \cdot k^* \cdot R$, we can efficiently find a feasible Regret-Bounded VRP solution using at most $(\alpha + \beta) \cdot k^*$ paths.
Preliminary Observations

In particular

Regret-Bounded VRP in \( d \equiv \) Distance-Constrained VRP in \( d^{\text{reg}} \)

Lemma

Given \( \leq \alpha \cdot k^* \) paths covering \( V \) with total \( d^{\text{reg}} \)-cost \( \leq \beta \cdot k^* \cdot R \), we can efficiently find a feasible Regret-Bounded VRP solution using at most \( (\alpha + \beta) \cdot k^* \) paths.

Proof.
Preliminary Observations

In particular

Regret-Bounded VRP in $d \equiv \text{Distance-Constrained VRP in } d^{\text{reg}}$

Lemma

Given $\leq \alpha \cdot k^* \text{ paths covering } V \text{ with total } d^{\text{reg}}\text{-cost } \leq \beta \cdot k^* \cdot R$, we can efficiently find a feasible Regret-Bounded VRP solution using at most $(\alpha + \beta) \cdot k^* \text{ paths.}$

Proof.

Break each path into paths of $d^{\text{reg}}\text{-length } \leq R$ and attach to $r$. \qed
In other words, it suffices to find $O(k^*)$ paths with total $d^{\text{reg}}$-cost $O(k^* \cdot R)$. 
Preliminary Observations

In other words, it suffices to find $O(k^*)$ paths with total $d^{\text{reg}}$-cost $O(k^* \cdot R)$.

**Side Note:** We can now easily get an $O(\log |V|)$-approximation for *asymmetric* Regret-Bounded VRP using known approximations for $k$-Person ATSP Path.

**Also:** $\alpha$-approximation for asymmetric Regret-Bounded VRP $\Rightarrow 2\alpha$-approximation for ATSP.
The Rounding

Suppose we have an LP solution $x^*$ with polynomial support size and value $k^*$.

Recall $k^* \leq (2 + \epsilon) \cdot OPT$. 
The Rounding

Suppose we have an LP solution $x^*$ with polynomial support size and value $k^*$.

Recall $k^* \leq (2 + \epsilon) \cdot OPT$.

**Easy case:** The union of all directed edges used by $\text{supp}(x^*)$ is acyclic.
View $x^*$ as a path decomposition of a flow $f$.

Notice $f$ has $d^{\text{reg}}$-cost at most $k^* \cdot R$ and satisfies

- $f(\delta^{\text{out}}(r)) \leq \lceil k^* \rceil$
- $f(\delta^{\text{in}}(v)) \geq 1$ for each $v \in V$
View $x^*$ as a path decomposition of a flow $f$.

Notice $f$ has $d^{\text{reg}}$-cost at most $k^* \cdot R$ and satisfies

- $f(\delta^{\text{out}}(r)) \leq \lceil k^* \rceil$
- $f(\delta^{\text{in}}(v)) \geq 1$ for each $v \in V$

Integrality of flows + $\text{supp}(f)$ being acyclic $\Rightarrow$ Can efficiently find $\leq \lceil k^* \rceil$ paths with total regret at most $k^* \cdot R$ which cover $V$. 
View $x^*$ as a path decomposition of a flow $f$.

Notice $f$ has $d^{\text{reg}}$-cost at most $k^* \cdot R$ and satisfies

- $f(\delta^{\text{out}}(r)) \leq \lceil k^* \rceil$
- $f(\delta^{\text{in}}(v)) \geq 1$ for each $v \in V$

Integrality of flows + $\text{supp}(f)$ being acyclic $\Rightarrow$ Can efficiently find $\leq \lceil k^* \rceil$ paths with total regret at most $k^* \cdot R$ which cover $V$.

Use the previous lemma to turn these into at most $2 \cdot k^* + 1$ paths covering $V$ with maximum regret $\leq R$. 
The Rounding

Things are not so simple if the flow described by $x^*$ contains cycles!

**High-Level Approach**

1) Shortcut the paths $P \in \text{supp}(x^*)$ past some clients to make their union acyclic.

2) If a client $v$ is removed from more than a $\frac{1}{2}$-fraction of their covering paths, then they are discarded them outright. We will also ensuring there is a cheap way to reintegrate them later.

3) Double the resulting acyclic flow and then round as before.
The Rounding

For a rooted path $P$, we define red and blue edges.

The cost of the red edges is at most $\frac{3}{2} \cdot d^{\text{reg}}(P)$ [Blum et al., 2003].

Deleting the blue edges naturally breaks $P$ into red intervals (some intervals may be singletons).
The Rounding

We now identify a forest $F$ and discard all but one particularly chosen node from each component.

Define a cut requirement function $f : 2^V \rightarrow \{0, 1\}$ by:

- $f(S) = 1$ if every $v \in S$ has $\geq \frac{1}{2}$ of its red intervals crossing $S$.
- $f(S) = 0$ otherwise.
The Rounding

Note:

- \( f \) is downward monotone: \( f(S) \geq f(T) \) for every \( \emptyset \subset S \subset T \).
- Every cut \( S \) with \( f(S) = 1 \) is crossed by a \( \frac{1}{2} \)-fraction of red edges:
  \[
  \sum_{e \in \delta(S)} \sum_{P: e \text{ is red on } P} x_P^* \geq \frac{1}{2}
  \]
- The total fractional \( d \)-cost of the red edges is at most \( \frac{3}{2} \cdot k^* \cdot R \).
The Rounding

Note:

• $f$ is downward monotone: $f(S) \geq f(T)$ for every $\emptyset \subsetneq S \subseteq T$.

• Every cut $S$ with $f(S) = 1$ is crossed by a $\frac{1}{2}$-fraction of red edges:

$$\sum_{e \in \delta(S)} \sum_{P : e \text{ is red on } P} x^*_P \geq \frac{1}{2}$$

• The total fractional $d$-cost of the red edges is at most $\frac{3}{2} \cdot k^* \cdot R$.

Thus, there is a forest $F$ with $d$-cost at most $6 \cdot k^* \cdot R$ satisfying $f(C) = 0$ for each component $C$ [Goemans and Williamson, 1994].
The Rounding

Note:

• \( f \) is downward monotone: \( f(S) \geq f(T) \) for every \( \emptyset \subseteq S \subseteq T \).
• Every cut \( S \) with \( f(S) = 1 \) is crossed by a \( \frac{1}{2} \)-fraction of red edges:

\[
\sum_{e \in \delta(S)} \sum_{P: e \text{ is red on } P} x^*_P \geq \frac{1}{2}
\]

• The total fractional \( d \)-cost of the red edges is at most \( \frac{3}{2} \cdot k^* \cdot R \).

Thus, there is a forest \( F \) with \( d \)-cost at most \( 6 \cdot k^* \cdot R \) satisfying \( f(C) = 0 \) for each component \( C \) [Goemans and Williamson, 1994].

Each component \( C \) has a node \( v \) where at least a \( \frac{1}{2} \)-fraction of \( v \)'s red intervals are contained in \( C \).

Let \( W \subseteq V \) consist of one such node from each component.
Forest $\Rightarrow$ Cycles

Standard TSP trick: convert each component of the forest to a cycle.

(White = W)
Forest $\Rightarrow$ Cycles

Standard TSP trick: convert each component of the forest to a cycle.

(White = $W$)

Since $d$- and $d^{\text{reg}}$-costs are equal for cycles, then the total $d^{\text{reg}}$-cost of these cycles is at most $12 \cdot k^* \cdot R$. 
Shortcutting the Paths

For each $P \in \text{supp}(x^*)$:

1) Mark each node in $V - W$ for removal (the black nodes).
Shortcutting the Paths

2) If a red interval contains more than one $W$-node, then mark them all for removal.

The dashed contours indicate components of the forest $F$ including these grey nodes.
Shortcutting the Paths

2) If a red interval contains more than one $W$-node, then mark them all for removal.

The dashed contours indicate components of the forest $F$ including these grey nodes.

**Important:** Each witness node $v \in W$ is marked for removal this way in at most a $\frac{1}{2}$-fraction of its covering paths.
3) Now shortcut $P$ past all marked nodes.
3) Now shortcut $P$ past all marked nodes.

After doing so for all $P \in \text{supp}(x^*)$:

- The fractional number of paths $k^*$ does not change.
- The $d^{\text{reg}}$-cost of each path does not increase.
- Each $v \in W$ lies on at least a $\frac{1}{2}$-fraction of the new paths.
- The union of the new paths is acyclic!
Wrap Up

Now we can round the acyclic flow described by $2x^*$ to get at most $\lceil 2k^* \rceil$ paths spanning $W$ with total $d^{\text{reg}}$-cost at most $2 \cdot k^* \cdot R$. 

Finally, applying the lemma finds at most $16 \cdot k^* + 1$ paths of maximum $d^{\text{reg}}$-cost spanning $V$: an $O(1)$-approximate solution!
Wrap Up

Now we can round the acyclic flow described by $2x^*$ to get at most $\lceil 2k^* \rceil$ paths spanning $W$ with total $d^{\text{reg}}$-cost at most $2 \cdot k^* \cdot R$.

Incorporating the cycles via their witness nodes and shortcutting finds $\lceil 2k^* \rceil$ paths spanning $V$ with total $d^{\text{reg}}$-cost at most $14 \cdot k^* \cdot R$. 
Wrap Up

Now we can round the acyclic flow described by $2x^*$ to get at most $\lceil 2k^* \rceil$ paths spanning $W$ with total $d^{\text{reg}}$-cost at most $2 \cdot k^* \cdot R$.

Incorporating the cycles via their witness nodes and shortcutting finds $\lceil 2k^* \rceil$ paths spanning $V$ with total $d^{\text{reg}}$-cost at most $14 \cdot k^* \cdot R$.

Finally, applying the lemma finds at most $16 \cdot k^* + 1$ paths of maximum $d^{\text{reg}}$-cost $R$ spanning $V$: an $O(1)$-approximate solution!
Extensions

Optimizations:

• Choose a different cutoff than $\frac{1}{2}$ in the definition of the cut requirement function.

• Tweaks to the definition of the cut requirement function and how we shortcut the paths to get the acyclic collection.
Extensions

Optimizations:

- Choose a different cutoff than $\frac{1}{2}$ in the definition of the cut requirement function.
- Tweaks to the definition of the cut requirement function and how we shortcut the paths to get the acyclic collection.

Consider the variant where we have $k$ vehicles and we want to minimize the maximum client regret.

- An $O(k^2)$-approximation.
- An $\Omega(k)$ “integrality gap” lower bound for the feasibility LP based on configurations.
Extensions

Optimizations:

• Choose a different cutoff than $\frac{1}{2}$ in the definition of the cut requirement function.
• Tweaks to the definition of the cut requirement function and how we shortcut the paths to get the acyclic collection.

Consider the variant where we have $k$ vehicles and we want to minimize the maximum client regret.

• An $O(k^2)$-approximation.
• An $\Omega(k)$ “integrality gap” lower bound for the feasibility LP based on configurations.

Thank You!