## Close Enough Routing Problems

## Ángel Corberán ${ }^{1}$

Isaac Plana ${ }^{1}$, Miguel Reula ${ }^{1}$ \& José M. Sanchis ${ }^{2}$

\author{

1. Universitat de València <br> 2. Universidad Politécnica de Valencia
}

Fuengirola, June 23-25, 2021

## Outline

(9) Introduction
(2) The Distance-constrained CEARP
(3) The Min-Max CEARP (with N. Bianchessi)
(4) The Profitable CEARP (with N. Bianchessi)
(5) Future work

## Outline

(9) Introduction
(2) The Distance-constrained CEARP
(3) The Min-Max CEARP (with N. Bianchessi)

4 The Profitable CEARP (with N. Bianchessi)
(5) Future work

## Close Enough TSP

## Gulczynski, Heath, Price (2006)

"Historically when a utility company measures the monthly usage of a customer, a meter reader visits each customer and physically reads the usage value at each site. Radio frequency identification (RFID) tags at customer locations can remotely provide data if the tag reader is within a certain radius of the tag. This changes the routing problem from one of a standard TSP to what we call a Close Enough TSP (CETSP). Thus the route lengths of the meter readers can be drastically reduced by developing heuristics that exploit this close enough feature."
"We consider such a meter reading routing problem where each customer is modeled as a point in the plane. Additionally there is a point that represents the depot for the meter reader. A CETSP tour must begin and end at the depot and travel within the required radius, $r$, of each customer. For simplicity in the cases tested here the meter reader was not restricted to a road network. All distances are Euclidean and the objective is to minimize the total distance traveled."

## Close Enough TSP

## Gulczynski, Heath, Price (2006)



(Left) An example of a supernode set on 100 nodes, with radius 9 , and the depot located at $(50,10)$. The circles represent the customer nodes, and the asterisks are the supernodes.
(Right) An example of tiling the plane with regular hexagons of side length $r=1.5$ units. All customers (small circles) in a given tile are within $r$ units of the center of that tile (*).

## Close Enough TSP

## Gulczynski, Heath, Price (2006)



(Left) An example CETSP tour on 100 nodes, with radius 9, and the depot located at $(50,10)$. The circles represent the customers, and the asterisks are the supernodes
(Right) An improved tour.

## Applications. Wireless Sensor Networks

## Potdar, Sharif, and Chang (2009)


"WSNs are mostly used in applications ranging from civil and military to environmental and healthcare monitoring"
"The wireless communication channel provides a medium to transfer the information extracted from the sensor node to the exterior world which may be a computer network and inter-node communication [5]. However, WSN using IEEE 802.15.4 Wireless Personal Area Network protocol (WPAN) or Bluetooth is complicated and costly. Using RFID to implement wireless communication is relatively simple and cheap."

## Applications. Meter reading

## Shuttelworth, Golden, Smith, Wasil (2008)

"With the widespread availability of radio frequency identification (RFID) technology in the late 2000s, an RFID tag can be attached to a meter. A truck equipped with a reading device traverses streets in the service area and collects data from the meters automatically. It is not necessary to visit each meter individually because a meter can be read from a predefined distance. In the automated meter reading (AMR) problem, the utility seeks vehicle routes that cover all meters (customers) in the service area and minimize the total distance traveled."


A neighborhood on a route.

## Applications. Meter reading

Shuttelworth, Golden, Smith, Wasil (2008)

(Left) A traditional route through a neighborhood. (Right) An AMR route.

Automatic Meter Reading is an arc routing problem

## Automatic Meter Reading Problem

- Each meter has a transmitter.
- A vehicle has a receiver than can read the meters located closer than a given distance $r$.



## Automatic Meter Reading Problem



## Automatic Meter Reading Problem



## Automatic Meter Reading Problem



## Automatic Meter Reading Problem

The AMR Problem (as a CETSP) was first studied in:

- Shuttleworth, Golden, Smith \& Wasil, "Advances in Meter Readings: Heuristic Solution of the Close Enough Traveling Salesman problem over a Street Network" (2008).
- It is equivalent to the following arc routing problem the Generalized Directed Rural Postman Problem introduced by Drexl $(2007,2014)$.
- We will use the name Close Enough.


## Introduction

The Close Enough ARP (CEARP) and some variants have been studied in

- Hà, Bostel, Langevin \& Rousseau $(2012,2014)$
- Ávila, Corberán, Plana \& Sanchis $(2016,2017)$
- Aràoz, Fernández, Franquesa (2017)
- Cerrone, Cerulli, Golden, Pentangelo (2017)
- Renaud, Absi, Feillet (2017)
- Russo, Cerrone, Di Placido (2019)


## Introduction

When the network is very large, a single vehicle is not able to perform all the services, and several vehicles must be considered.

To balance the routes of the different vehicles we can:

- limit the distance traveled by each vehicle: the Distance-constrained CEARP,
- use a min-max objective function: the Min-Max CEARP.


## Outline

## (9) Introduction

## (2) The Distance-constrained CEARP

## (3) The Min-Max CEARP (with N. Bianchessi)

4 The Profitable CEARP (with N. Bianchessi)
(5) Future work

## DC-CEARP. Definition

- $G=(V, A)$ strongly connected directed graph.
- Vertex 1 is the depot.
- There is a cost $d_{i j} \geq 0$ associated with each $\operatorname{arc}(i, j) \in A$.
- There is a family $\mathbb{H}=\left\{H_{1}, \ldots, H_{L}\right\}$, of arc sets, each $H_{c} \subseteq A$ ( $H_{c}$ are the arcs from which a "customer" $c$ can be served).
- There is a fleet of $K$ vehicles, and
- There is a limit $D_{\max }$ on the length of the routes.

DC-CEARP is defined as to find a set of $K$ routes of minimum total length, such that for each customer $c$, at least one arc in $H_{c}$ is traversed, and each route length does not exceed $D_{\text {max }}$.

## DC-CEARP. Notation

We use the following notation:

- $A_{R}=H_{1} \cup \cdots \cup H_{L}$; arcs that can service the customers.
- $A_{N R}=A \backslash A_{R}$.
- Given $S \subset V, \delta^{+}(S)=\{(i, j) \in A \mid i \in S, j \notin S\}$, $\delta^{-}(S)=\{(i, j) \in A \mid i \notin S, j \in S\}$, and $\delta(S)=\delta^{+}(S) \cup \delta^{-}(S)$.
- Given a set of variables $x_{i j}$ indexed on the $\operatorname{arcs}(i, j) \in A$, and given $F \subset A, \quad x(F)=\sum_{(i, j) \in F} x_{i j}$


## A new DC-CEARP formulation

Ávila, Corberán, Plana \& Sanchis (2017) proposed 4 formulations. We present here a new one combining the best characteristics of them.

- For each arc $(i, j) \in A$ and for each vehicle $k$, $x_{i j}^{k}=$ number of times that arc $(i, j)$ is traversed by vehicle $k$.
- For each customer $c$ and each vehicle $k$, $z_{c}^{k}=1$, if $c$ is served by vehicle $k$, 0 , otherwise.
- For each customer $c$, each vehicle $k$, and each required arc $(i, j)$, $y_{i j}^{k c}=1$, if $c$ is served by vehicle $k$ from $(i, j), 0$, otherwise
Note that $\sum_{(i, j) \in H_{c}} y_{i j}^{k c}=z_{c}^{k}$


## $F_{x z y}+$ Formulation

$$
\begin{align*}
\text { Minimize } & \sum_{k \in \mathbb{K}} \sum_{(i, j) \in A} d_{i j} x_{i j}^{k} \\
\sum_{(i, j) \in A} d_{i j} x_{i j}^{k} \leq D_{\text {max }} & \forall k \in \mathbb{K}  \tag{1}\\
x^{k}\left(\delta^{+}(i)\right)=x^{k}\left(\delta^{-}(i)\right) & \forall i \in V, \forall k \in \mathbb{K}  \tag{2}\\
\sum_{k \in \mathbb{K}} \sum_{(i, j) \in H_{c}} y_{i j}^{k c}=1 & \forall c \in \mathbb{H}  \tag{3}\\
x_{i j}^{k} \geq y_{i j}^{k c} & \forall(i, j) \in A_{R}, \forall c \in \mathbb{H}, \forall k \in \mathbb{K}  \tag{4}\\
\sum_{(i, j) \in H_{c}} y_{i j}^{k c}=z_{c}^{k} & \forall c \in \mathbb{H}, \forall k \in \mathbb{K}  \tag{5}\\
x^{k}\left(\delta^{+}(S)\right) \geq z_{c}^{k}-x^{k}\left(H_{c} \cap A(V \backslash S)\right) & \forall S \subset V \backslash\{1\}, \forall c \in \mathbb{H}, \forall k \in \mathbb{K}  \tag{6}\\
x_{i j}^{k} \geq 0 \text { and integer } & \forall(i, j) \in A, \forall k \in \mathbb{K}  \tag{7}\\
z_{c}^{k} \in\{0,1\} & \forall c \in \mathbb{H}, \forall k \in \mathbb{K}  \tag{8}\\
y_{i j}^{k c} \in\{0,1\} & \forall(i, j) \in A_{R}, \forall c \in \mathbb{H}, \forall k \in \mathbb{K}
\end{align*}
$$

## Valid inequalities

- Other connectivity inequalities.
- Parity inequalities.
- $\mathrm{K}-\mathrm{C}$ and $\mathrm{K}-\mathrm{C}_{02}$ inequalities.
- Path-bridge inequalities.
- Max-distance inequalities.
- Symmetry-breaking inequalities.

For a single vehicle: dissagregate inequalities, and
For a subset $\Omega \subseteq\{1, \ldots, K\}$ of vehicles: $\Omega$-aggregate inequalities.

## Max-distance inequalities

Suppose a set of customers $F^{H}$ that can not be served with just one vehicle:


$$
\begin{aligned}
& \sum_{k=1}^{K} x^{k}\left(\delta^{-}(S)\right) \geq 2, \quad \text { and } \\
& \sum_{k \neq k^{\prime}}^{K} x^{k}\left(\delta^{-}(S)\right) \geq 1, \quad \forall k^{\prime}
\end{aligned}
$$

(Max-distance- $x$ ineq.)

## Max-distance inequalities

Suppose a set of customers $F^{H}$ that can not be served with just one vehicle:


## For each $k$ :

$\sum_{c \in F^{H}} z_{c}^{k} \leq\left|F^{H}\right|-1, \quad$ and

$$
\sum_{c \in F^{H}} y^{k c}\left(H_{c}\right) \leq\left|F^{H}\right|-1
$$

(Max-distance-z ineq.)

## Max-distance inequalities

How do we know if set $F^{H}$ can not be served with a single vehicle?
$\Rightarrow$ Solving the CEARP defined on $G$ with set of customers $F^{H}$ by using the B\&C for the CEARP by Ávila et al. (2015).

## Comparison of separation strategies and cutting-plane algorithms

To analyze the contribution of the valid inequalities and the separation algorithms, we compare the gaps in the root node and the performance profiles (Dolan and Moré, 2002) of the versions of our B\&C procedure using different combinations of separation algorithms.

## Comparison of separation strategies and cutting-plane algorithms

For instance, we considered different options regarding the max-distance inequalities. Here, three new versions were implemented and compared with the "whole algorithm". Performance profiles, average gaps in the root node, and other measures follow.

|  | \# opt | Gap0 (\%) | Time0 (scs) | Time (scs) |
| :--- | :---: | :---: | :---: | :---: |
| $V 1234$ | 46 | 5.874 | 252.38 | 1031.19 |
| $V 123$ | 46 | 9.835 | 114.34 | 902.39 |
| $V 123+4(a)$ | 46 | 7.738 | 136.99 | 1036.11 |
| $V 123+4(b)$ | 46 | 6.655 | 241.18 | 1054.21 |

Table: Results on the subset of 48 instances - max-distance

## Comparison of separation strategies and cutting-plane algorithms

We compute the performance ratio $r_{p, s}=t_{p, s} / \min \left\{t_{p, s}: s \in \mathcal{S}\right\}$,
where $t_{p, s}$ is the computing time required by algorithm $s$ to solve instance $p$. If $s$ is not able to solve the instance $p$ within the time limit, we set $r_{p, s}=\infty$.

Thus, the performance profile of each version $s$,

$$
\rho_{s}(\tau)=\frac{\left|\left\{p \in \mathcal{P}: r_{p, s} \leq \tau\right\}\right|}{|\mathcal{P}|}
$$

describes the percentage of instances that can be solved by $s$ within a factor $\tau \geq 1$ compared to the fastest algorithm.

Note, for example, that $\rho_{s}(1)$ is the percentage of instances for which algorithm $s$ is the fastest and that $\rho_{s}(\infty)$ is the percentage of instances that are solved by algorithm $s$ within the time limit.

## Comparison of separation strategies and cutting-plane algorithms



Figure: Impact of the max-distance inequalities: Performance profile

## Comparison of the new B\&C with that of Ávila et al.



Figure: The proposed B\&C (Alg 1) versus that of Ávila et al. (Alg 2)

## New B\&C versus that of Ávila et al.

|  | Gap0(\%) | Final Gap(\%) |
| :--- | :---: | :---: |
| Algorithm 0 (Full) | 14.20 | 8.98 |
| Algorithm 1 (New) | 14.36 | 7.80 |
| Algorithm 2 (Ávila et al.) | 15.35 | 9.23 |

Table: Results on the 11 instances not solved by any algorithm

|  | \# opt | Gap(\%) | Final Gap(\%) |
| :--- | :---: | :---: | :---: |
| Algorithm 0 (Full) | 7 | 12.07 | 4.98 |
| Algorithm 1 (New) | 10 | 11.86 | 3.68 |
| Algorithm 2 (Ávila et al.) | 5 | 12.94 | 5.27 |

Table: Results on the 27 instances not solved by at least one algorithm

## Outline

(1) Introduction
(2) The Distance-constrained CEARP
(3) The Min-Max CEARP (with N. Bianchessi)

4 The Profitable CEARP (with N. Bianchessi)
(5) Future work

## Min-Max CEARP. Definition

The Min-Max Close-Enough Arc Routing Problem consists of finding a set of $K$ routes, starting and ending at the depot, servicing all the customers and minimizing the length of the largest route.

Two different formulations:

- Arc-based formulation
- Route-based set covering formulation


## Arc-based formulation

$x_{i j}^{k}=$ number of times that vehicle $k$ traverses arc $(i, j) \in A$,
$z_{c}^{k}=1$, if $c$ is served by vehicle $k$

$$
\begin{aligned}
\text { Minimize } & w \\
\sum_{(i, j) \in A} d_{i j} x_{i j}^{k} \leq w & \forall k \in \mathbb{K} \\
x^{k}\left(\delta^{+}(i)\right)=x^{k}\left(\delta^{-}(i)\right) & \forall i \in V, \forall k \in \mathbb{K} \\
\sum_{k \in \mathbb{K}} z_{c}^{k}=1 & \forall c \in \mathbb{H} \\
\sum_{(i, j) \in H_{c}} x_{i j}^{k} \geq z_{c}^{k} & \forall c \in \mathbb{H}, \forall k \in \mathbb{K} \\
x^{k}\left(\delta^{+}(S)\right) \geq z_{c}^{k}-x^{k}\left(H_{c} \cap A(V \backslash S)\right) & \forall S \subset V \backslash\{1\}, \forall c \in \mathbb{H}, \forall k \in \mathbb{K} \\
x_{i j}^{k} \geq 0 \text { and integer } & \forall(i, j) \in A, \forall k \in \mathbb{K} \\
z_{c}^{k} \in\{0,1\} & \forall c \in \mathbb{H}, \forall k \in \mathbb{K}
\end{aligned}
$$

## Route-based set covering formulation

Let $R^{k}$ be the set of feasible routes for vehicle $k$.
For each $r \in R^{k}$, let $d^{k r}$ be the length of the route. Moreover, for each customer $c \in \mathbb{H}$ and each $r \in R^{k}$, let $s_{c}^{k r}=1$ if the route $r$ serves customer $c$ and 0 otherwise. Then, let
$\lambda^{k r}=1$, if the route $r \in R^{k}$ is assigned to vehicle $k$, and
$w^{k}=$ length of the route assigned to vehicle $k$.

## Route-based set covering formulation

$$
\begin{aligned}
\text { Minimize } & w^{1} \\
\sum_{k \in \mathbb{K}} \sum_{r \in R^{k}} s_{c}^{k r} \lambda^{k r} \geq 1 & \forall c \in \mathbb{H} \\
\sum_{r \in R^{k}} \lambda^{k r}=1 & \forall k \in \mathbb{K} \\
\sum_{r \in R^{k}} d^{k r} \lambda^{k r}-w^{k} \leq 0 & \forall k \in \mathbb{K} \\
w^{k}-w^{k+1} \geq 0 & \forall k=1, \ldots, K-1 \\
\lambda^{k r} \in\{0,1\} & \forall k \in \mathbb{K}, \forall r \in R^{k}
\end{aligned}
$$

MASTER PROGRAM

## Route-based set covering formulation

Minimize $\quad w^{1}$

$$
\begin{array}{rll}
\sum_{k \in \mathbb{K}} \sum_{r \in R^{k}} s_{c}^{k r} \lambda^{k r} \geq 1 & \forall c \in \mathbb{H} & \mu_{c} \in \mathbb{R}^{+}, \quad \forall c \in \mathbb{H}, \\
\sum_{r \in R^{k}} \lambda^{k r}=1 & \forall k \in \mathbb{K} & \theta_{k} \in \mathbb{R}, \quad \forall k \in \mathbb{K}, \\
\sum_{r \in R^{k}} d^{k r} \lambda^{k r}-w^{k} \leq 0 & \forall k \in \mathbb{K} & \rho_{k} \in \mathbb{R}^{-}, \quad \forall k \in \mathbb{K}, \\
w^{k}-w^{k+1} \geq 0 & \forall k=1, \ldots, K-1 & \sigma_{k} \in \mathbb{R}^{+}, \quad k=1, \ldots, K-1 \\
\lambda^{k r} \in\{0,1\} & \forall k \in \mathbb{K}, \forall r \in R^{k} &
\end{array}
$$

MASTER PROGRAM

## Branch-and-price (BP) algorithm

In a BP algorithm, at each node of the branch-and-bound tree, the linear relaxation of the Master Problem (LMP) is solved iteratively by means of column generation.
The starting point is to define the LMP over a subset $\tilde{R} \subseteq \bigcup_{k \in \mathbb{K}} R^{k}$ of the feasible routes for the vehicles (reduced LMP, RLMP).
At each iteration, column generation alternates between the optimization of the RLMP and the solution of pricing problems (PPs). The former allows to retrieve optimal dual variable values with respect to set $\tilde{R}$. The latter generates negative reduced cost route variables $\lambda^{k r}$ to be included in the RLMP, if any.
When no negative reduced cost variable is found, the optimal solution of the RLMP is also the optimal solution of the LMP. Branching is finally required to ensure the integrality of the solution.

## Column generation

Let us consider the LMP at the root node of the branch-and-bound tree. The dual of LMP is:
$\operatorname{Max} \quad \sum_{c \in \mathbb{H}} 1 \cdot \mu_{c}+\sum_{k \in \mathbb{K}} 1 \cdot \theta_{k}+\sum_{k \in \mathbb{K}} 0 \cdot \rho_{k}+\sum_{k=1}^{K-1} 0 \cdot \sigma_{k}$

$$
\begin{aligned}
\sum_{c \in \mathbb{H}} s_{c}^{k r} \mu_{c}+\theta_{k}+d^{k r} \rho_{k} \leq 0 & \forall k \in \mathbb{K}, \forall r \in R^{k} \\
-\rho_{1}+\sigma_{1} \leq 1 & \\
-\rho_{k}+\sigma_{k}-\sigma_{k-1} \leq 0 & \forall k=2, \ldots K-2 \\
-\rho_{K}-\sigma_{K-1} \leq 0 &
\end{aligned}
$$

## Column generation

There is one distinct PP for each vehicle $k \in \mathbb{K}$.
In particular, given the duals $(\boldsymbol{\mu}, \boldsymbol{\theta}, \boldsymbol{\rho}, \boldsymbol{\sigma})$, the PP for vehicle $k \in \mathbb{K}$ consists of finding a minimum reduced cost route to be assigned to the vehicle, where the reduced cost $\bar{c}^{k r}(\boldsymbol{\mu}, \boldsymbol{\theta}, \boldsymbol{\rho})$ of route $r \in R^{k}$ to be assigned to the vehicle is:

$$
\begin{equation*}
\bar{c}^{k r}(\boldsymbol{\mu}, \boldsymbol{\theta}, \boldsymbol{\rho})=-\sum_{c \in \mathbb{H}} s_{c}^{k r} \mu_{c}-\theta_{k}-d^{k r} \rho_{k} \tag{10}
\end{equation*}
$$

A solution (a route) corresponds to a negative reduced cost $\lambda^{k r}$ variable if its value (reduced cost) is less than 0.

## Pricing problem

The PP associated with vehicle $k \in \mathbb{K}$ can be formulated as:

$$
\begin{aligned}
\operatorname{Min}-\sum_{c \in \mathbb{H}} \mu_{c} z_{c}^{k}-\sum_{(i, j) \in A} d_{i j} \rho_{k} x_{i j}^{k} & \\
x^{k}\left(\delta^{+}(i)\right)=x^{k}\left(\delta^{-}(i)\right) & \forall i \in V \\
x^{k}\left(\delta^{+}(S)\right) \geq z_{c}^{k}-x^{k}\left(H_{c} \cap A(V \backslash S)\right) & \forall S \subset V \backslash\{1\}, \forall c \in \mathbb{H} \\
\sum_{(i, j) \in H_{c}} x_{i j}^{k} \geq z_{c}^{k} & \forall c \in \mathbb{H} \\
x_{i j}^{k} \geq 0 \text { and integer } & \forall(i, j) \in A \\
z_{c}^{k} \in\{0,1\} & \forall c \in \mathbb{H},
\end{aligned}
$$

where $\mu_{c} \geq 0 \forall c \in \mathbb{H}$, and, $-d_{i j} \rho_{k} \geq 0$ for each $(i, j) \in A$.

## Pricing problem

The PP formulation is equivalent to

$$
\begin{aligned}
-\operatorname{Max} \sum_{c \in \mathbb{H}} \mu_{c} z_{c}^{k}-\sum_{(i, j) \in A}\left(-d_{i j} \rho_{k}\right) x_{i j}^{k} & \\
x^{k}\left(\delta^{+}(i)\right)=x^{k}\left(\delta^{-}(i)\right) & \forall i \in V \\
x^{k}\left(\delta^{+}(S)\right) \geq z_{c}^{k}-x^{k}\left(H_{c} \cap A(V \backslash S)\right) & \forall S \subset V \backslash\{1\}, \forall c \in \mathbb{H} \\
\sum_{(i, j) \in H_{c}} x_{i j}^{k} \geq z_{c}^{k} & \forall c \in \mathbb{H} \\
x_{i j}^{k} \geq 0 \text { and integer } & \forall(i, j) \in A \\
z_{c}^{k} \in\{0,1\} & \forall c \in \mathbb{H}
\end{aligned}
$$

Note that it is not mandatory to service all the customers.

## Pricing problem

We call Profitable Close Enough Arc Routing Problem (PCEARP) to this problem. We have studied it and have proposed a branch-and-cut algorithm for its solution.

If an upper bound $W$ is available for $w^{1}, R^{k}$ can be restricted to include feasible routes such that $d^{k r} \leq W-1$, and we can include in the PP formulation the constraint:

$$
\sum_{(i, j) \in A} d_{i j} x_{i j}^{k} \leq W-1
$$

## Branch and Price. Branching rules

As mentioned, when no negative reduced cost variable is found, the optimal solution of the RLMP is the optimal solution of the LMP. If it is not integer, branching is required.

Let $(\bar{\lambda}, \overline{\mathbf{w}})$ be the optimal solution of the current RLMP. When $(\overline{\boldsymbol{\lambda}}, \overline{\mathbf{w}})$ is fractional, we apply the following branching rules:

- First, we consider an application of the Ryan and Foster's branching rule:

For each pair of customers $c^{\prime}$ and $c^{\prime \prime}$, we define $\alpha_{c^{\prime} c^{\prime \prime}}=\sum_{k \in \mathbb{K}} \sum_{r \in R^{k}} s_{c^{\prime}}^{k r} s_{c^{\prime \prime}}^{k r} \lambda^{k r}$ as the sum of the $\lambda^{k r}$ values associated with routes that serve both customers $c^{\prime}$ and $c^{\prime \prime}$.

We select $\alpha_{c^{\prime} c^{\prime \prime}}^{*}$ closest to 0.5 such that $0<\alpha_{c^{\prime} c^{\prime \prime}}^{*}<1$. On one branch, we set $\alpha_{c^{\prime} c^{\prime \prime}}^{*}=0$, meaning that $c^{\prime}$ and $c^{\prime \prime}$ must be served in different routes (and vehicles). Then, $z_{c^{\prime}}^{k}+z_{c^{\prime \prime}}^{k} \leq 1$ is added to the formulation associated with each $k \in \mathbb{K}$. On the other, we set $\alpha_{c^{\prime} c^{\prime \prime}}^{*}=1$, meaning that they have to be served in the same route by the same vehicle. Then, we add $z_{c^{\prime}}^{k}-z_{c^{\prime \prime}}^{k}=0$.

- When the solution is fractional and there is no $c^{\prime}$ and $c^{\prime \prime}$ such that $0<\alpha_{c^{\prime} c^{\prime \prime}}^{*}<1$, we branch on the fractional use of an arc by vehicle $k \in \mathbb{K}$.


## Computational results. B\&C vs B\&P

We studied the performance of the branch-and-cut and branch-and-price methods and compared them.

## Computational results


(a) Performance profile- Overall

(c) Performance profile-3 vehicles

(b) Performance profile-2 vehicles

(d) Performance profile-4 vehicles

## Computational results

|  | Instances with LB |  |  |  |  | Instances without LB |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | \# Inst | BC |  | BP |  | BC |  |  |
|  |  | \# opt Time | Gap(\%) | \# opt Time | Gap(\%) | \# Inst | \# opt Time | Gap(\%) |
| 2 | 59 | 59245.0 | 0.00 | 531081.6 | 0.38 | 12 | 73764.1 | 4.09 |
| 3 | 57 | 353032.0 | 1.47 | 432064.0 | 1.19 | 7 | 07200.0 | 11.67 |
| 4 | 41 | 164724.6 | 5.43 | 252996.6 | 1.93 | 5 | 07200.0 | 22.54 |
| 5 | 31 | 56279.3 | 8.32 | 153770.0 | 3.80 | 2 | 07200.0 | 25.31 |
| M6 | 42 | 36741.5 | 7.68 | 214101.6 | 2.78 | 2 | 07200.0 | 19.72 |
|  | 230 | 1183733.9 | 3.86 | 1572580.2 | 1.76 | 28 | 75727.5 | 10.89 |

Table: Gap comparison on the instances with and without LB computed by the BP algorithm

## Outline

(9) Introduction
(2) The Distance-constrained CEARP
(3) The Min-Max CEARP (with N. Bianchessi)

4 The Profitable CEARP (with N. Bianchessi)
(5) Future work

## The Profitable CEARP. Definition

- $G=(V, A)$ strongly connected directed graph.
- Vertex 1 is the depot.
- There is a cost $d_{i j} \geq 0$ associated with each $\operatorname{arc}(i, j) \in A$.
- There is a family $\mathbb{H}=\left\{H_{1}, \ldots, H_{L}\right\}$, of arc sets, each $H_{c} \subseteq A$ ( $H_{c}$ are the arcs from which a "customer" $c$ can be served).
- There is a profit $p_{c} \geq 0$ collected (only once) if $c$ is serviced.

The PCEARP is defined as to find a tour on G maximizing the difference between the sum of profits collected and the total length of the tour.

## The Profitable CEARP. Formulation

$$
\begin{aligned}
& \text { Maximize } \sum_{c \in \mathbb{H}} p_{c} z_{c}-\sum_{(i, j) \in A} d_{i j} x_{i j} \\
& x\left(\delta^{+}(i)\right)=x\left(\delta^{-}(i)\right) \forall i \in V \\
& x\left(\delta^{+}(S)\right) \geq z_{c}-x\left(H_{c} \cap A(V \backslash S)\right) \forall S \subset V \backslash\{1\}, \forall c \in \mathbb{H} \\
& x\left(H_{c}\right) \geq z_{c} \forall c \in \mathbb{H} \\
& x_{i j} \geq 0 \text { and integer } \forall(i, j) \in A \\
& 1 \geq z_{c} \geq 0 \text { and integer } \forall c \in \mathbb{H},
\end{aligned}
$$

Other valid inequalities: Parity and K-C inequalities.

## The Profitable CEARP. Polyhedron

$\operatorname{PCEARP}(G)=\operatorname{conv}\left\{(x, z) \in \mathbb{Z}^{|A|+|\mathbb{H}|}:(x, z)\right.$ is a PCEARP tour $\}$.
It can be seen that $\operatorname{PCEARP}(G)$ is an unbounded polyhedron. If $G$ is strongly connected, $\operatorname{dim}(\operatorname{PCEARP}(G))=|A|+|\mathbb{H}|-|V|+1$.
Under mild conditions, the following inequalities are facet inducing:

- $x_{i j} \geq 0$,
- $z_{c} \geq 0$ and $z_{c} \leq 1$,
- $x\left(H_{c}\right) \geq z_{c}$,
- Connectivity inequalities, and
- Parity inequalities.


## The Profitable CEARP. Computational results

We have designed and implemented a heuristic and a branch-and-cut algorithm with separation procedures for:

- Connectivity inequalities,
- Parity inequalities, and
- K-C inequalities


## Computational results

Table: Characteristics of the sets of instances

|  | \#Inst | $\|V\|$ | $\|A\|$ |  | $\left\|A_{R}\right\|$ |  | $\left\|A_{N R}\right\|$ |  | $\|\mathbb{H}\|$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  | Min | Max | Min | Max | Min | Max | Min | Max |
| Albaida | 24 | 116 | 259 | 305 | 124 | 172 | 109 | 162 | 18 | 33 |
| Madrigueras | 24 | 196 | 453 | 544 | 224 | 305 | 197 | 281 | 22 | 47 |
| Random50 | 12 | 50 | 296 | 300 | 105 | 292 | 7 | 193 | 10 | 100 |
| Random75 | 12 | 75 | 448 | 450 | 143 | 438 | 10 | 305 | 15 | 150 |
| Random100 | 12 | 100 | 498 | 500 | 134 | 490 | 10 | 366 | 20 | 200 |
| Random150 | 12 | 150 | 749 | 750 | 256 | 731 | 19 | 493 | 30 | 300 |
| Random200 | 12 | 200 | 997 | 1000 | 321 | 972 | 27 | 679 | 40 | 400 |
| Random300 | 12 | 300 | 1498 | 1500 | 502 | 1457 | 43 | 998 | 60 | 600 |
| Random400 | 12 | 400 | 1999 | 2000 | 675 | 1936 | 63 | 1324 | 80 | 800 |

## Computational results

|  | [0.65 $\mu, 1.05 \mu]$ |  | [ $0.80 \mu, 1.20 \mu]$ |  | [0.95, $1.35 \mu]$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Min | Max | Min | Max | Min | Max |
| Albaida | 181.8 | 293.8 | 223.8 | 335.9 | 265.8 | 378.0 |
| Madrigueras | 167.4 | 271.0 | 206.4 | 309.7 | 245.1 | 348.6 |
| Random50 | 93.5 | 151.3 | 115.2 | 173.1 | 136.8 | 194.5 |
| Random75 | 80.7 | 130.5 | 99.3 | 149.3 | 118.0 | 168.0 |
| Random100 | 70.2 | 113.9 | 86.7 | 130.1 | 102.8 | 146.3 |
| Random150 | 61.3 | 99.2 | 75.3 | 113.2 | 89.7 | 127.6 |
| Random200 | 49.2 | 79.5 | 60.5 | 91.2 | 72.0 | 102.5 |
| Random300 | 38.9 | 63.2 | 48.0 | 72.3 | 57.1 | 81.3 |
| Random400 | 36.1 | 58.8 | 44.6 | 67.3 | 53.1 | 75.8 |

Table: Average $\min /$ max profit per instance set and profit interval

## Computational results

|  | \#lnst | Heuristic |  |  | B\&C |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | \# opt | Gap | Time | \# opt | Gap0 | Gap | \# nodes | Time |
| Alb | 72 | 59 | 0.44 | 0.9 | 72 | 0.09 | 0.00 | 1.2 | 1.0 |
| Mad | 72 | 39 | 3.03 | 4.0 | 72 | 1.87 | 0.00 | 10.5 | 12.3 |
| R50 | 36 | 23 | 0.77 | 17.8 | 36 | 0.90 | 0.00 | 5.6 | 0.5 |
| R75 | 36 | 13 | 2.38 | 29.6 | 36 | 0.32 | 0.00 | 11.1 | 1.2 |
| R100 | 36 | 13 | 2.69 | 33.0 | 36 | 1.19 | 0.00 | 1454.4 | 51.2 |
| R150 | 36 | 6 | 7.00 | 44.5 | 34 | 1.15 | 0.19 | 2620.4 | 269.5 |
| R200 | 36 | 6 | 10.84 | 46.4 | 31 | 1.87 | 0.21 | 4994.8 | 763.3 |
| R300 | 36 | 2 | 18.54 | 53.4 | 19 | 3.79 | 2.13 | 6064.1 | 2023.1 |
| R400 | 36 | 0 | 21.67 | 60.0 | 16 | 3.09 | 2.23 | 4181.9 | 2190.8 |
|  | 396 | 161 | 6.44 | 26.8 | 352 | 1.48 | 0.43 | 1759.6 | 484.2 |

Table: Heuristic and B\&C results in all instances

## Computational results

|  | \| \# Inst | Heuristic |  |  | B\&C |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | \# opt | Gap | Time | \# opt | Gap0 | Gap | \# nodes | Time |
| $[0.65 \mu, 1.05 \mu]$ | 132 | 68 | 7.84 | 26.4 | 119 | 2.16 | 0.51 | 1208.0 | 401.9 |
| $[0.80 \mu, 1.20 \mu]$ | 132 | 52 | 6.36 | 27.0 | 118 | 1.13 | 0.37 | 1888.6 | 471.4 |
| $[0.95 \mu, 1.35 \mu]$ | 132 | 41 | 5.11 | 26.9 | 115 | 1.14 | 0.43 | 2182.3 | 579.3 |
|  | 396 | 161 | 6.44 | 26.8 | 352 | 1.48 | 0.43 | 1759.6 | 484.2 |

Table: Heuristic and B\&C results by profit intervals

## Outline

## (9) Introduction

## (2) The Distance-constrained CEARP

## (3) The Min-Max CEARP (with N. Bianchessi)

4 The Profitable CEARP (with N. Bianchessi)
(5) Future work

## Future work

- To study the multiple vehicle version of the PCEARP, the Team Orienteering CEARP:

We consider mandatory and potential customers. A fleet of vehicles with maximum distance constraints. The goal is to maximize the collected profits associated with the potential customers.

## Close Enough Routing Problems

## Thanks for your attention!

