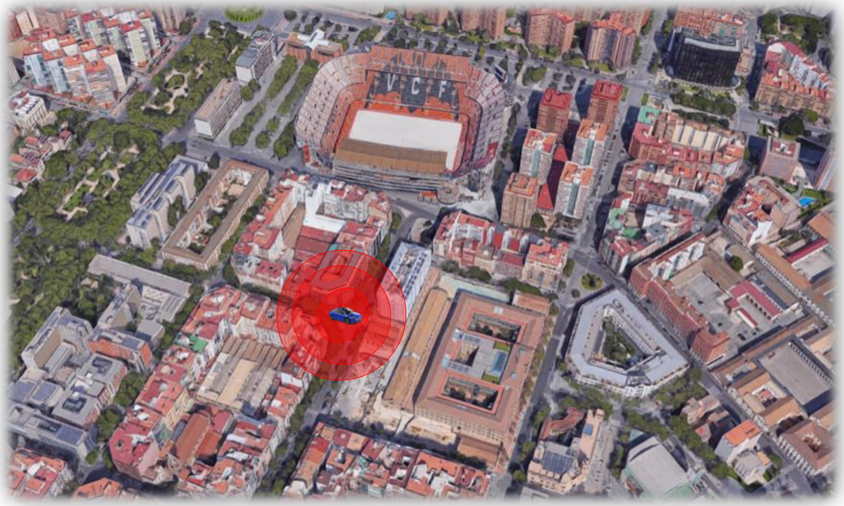
Feasible solutions for the Distance Constrained Close-Enough Arc Routing Problem

Á. Corberán [(1)], I. Plana [(1)], M. Reula [(1)] and J.M. Sanchis [(2)]

(1) University of Valencia
(2) Technical University of Valencia

IX International Workshop on Locational Analysis and Related Problems



VNIVERSITAT DE VALÈNCIA

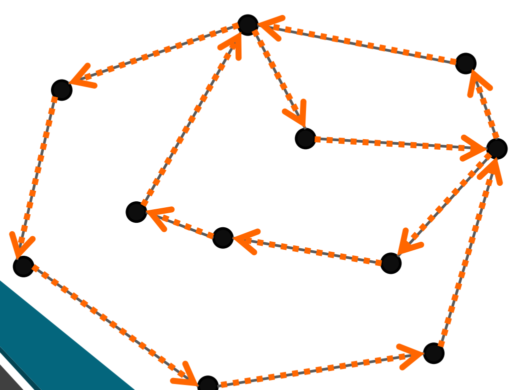
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Arc Routing Problems

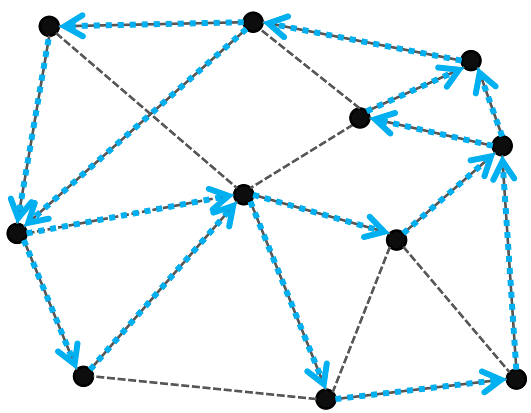
The goal of ARPs is to find one or several routes traversing a given set of arcs or/and edges that requires to be served.

Important ARPs :

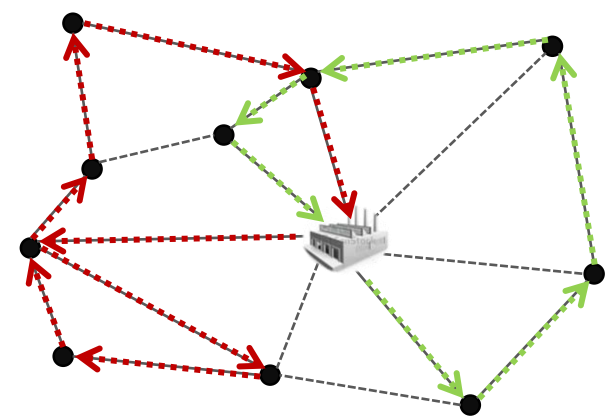
The Chinese Postman Problem (CPP)



The Rural Postman Problem (RPP)



The Capacitated Arc Routing Problem (CARP)



The Close Enough Arc Routing Problem

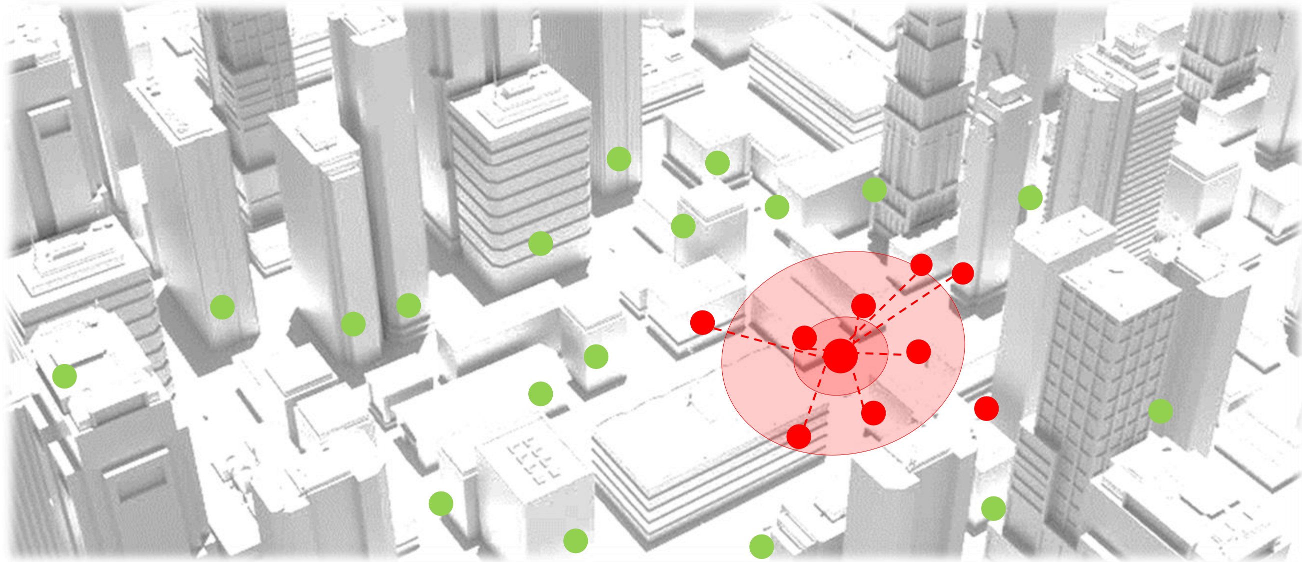
The **CEARP** considers that the service of a costumer is not associated with the traversal of an specific arc. The customer is served when the vehicle traverses any arc of a fixed subset of arcs.

*Each customer is served when **the vehicle gets closer than a certain distance r**.*

The problem consists of finding a **minimum cost tour** starting and ending at the depot, which traverses a set of arcs so that **all customers are served.**

Real-World Application

Automatic Meter Reading (water, electricity, gas)



Originally: done door to door.

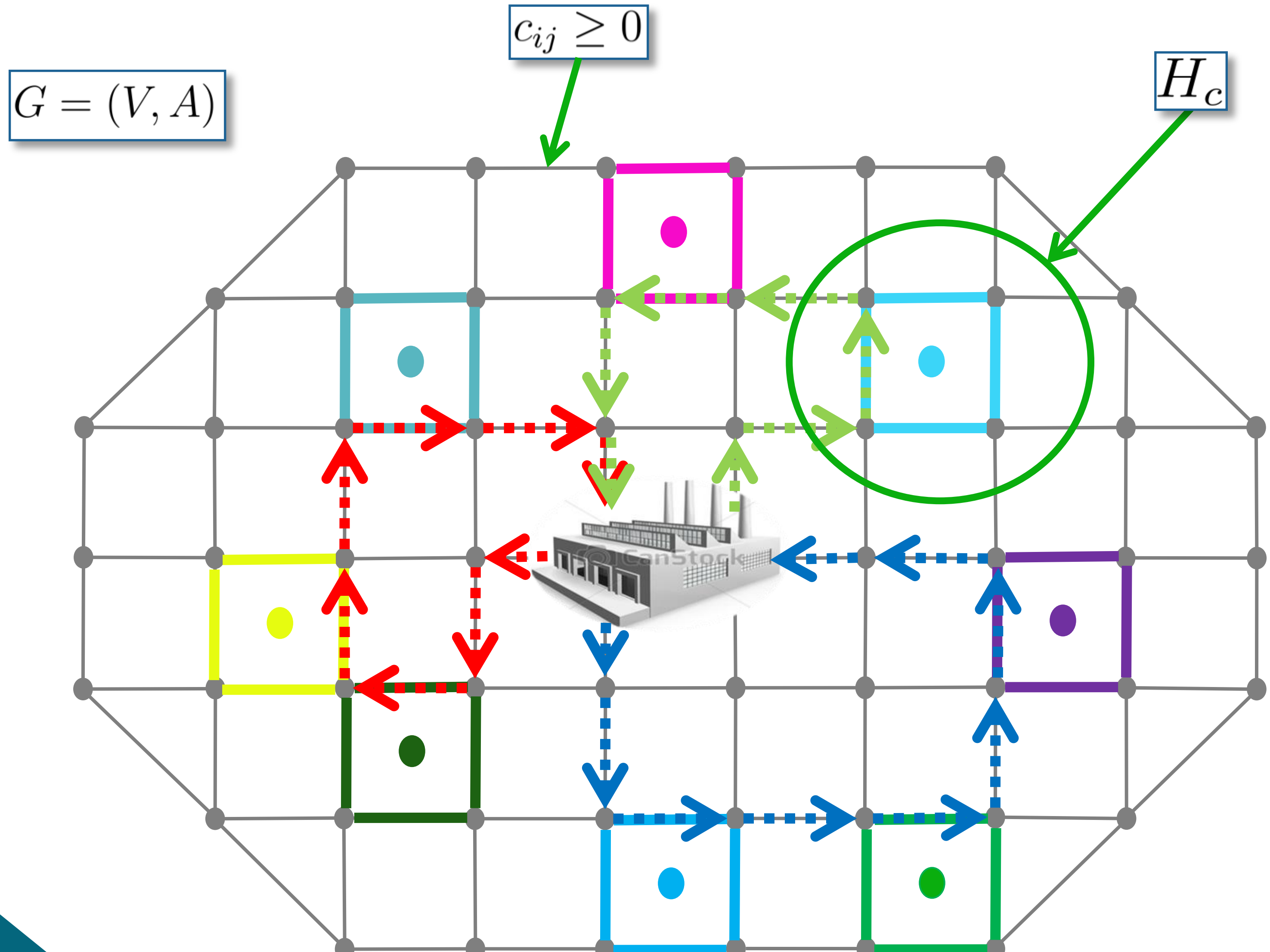
Nowadays: a vehicle with an installed receiver reads the data when it gets closer than a certain distance from each meter.

The Distance-Constrained CEARP with k-vehicles (DC-CEARP)

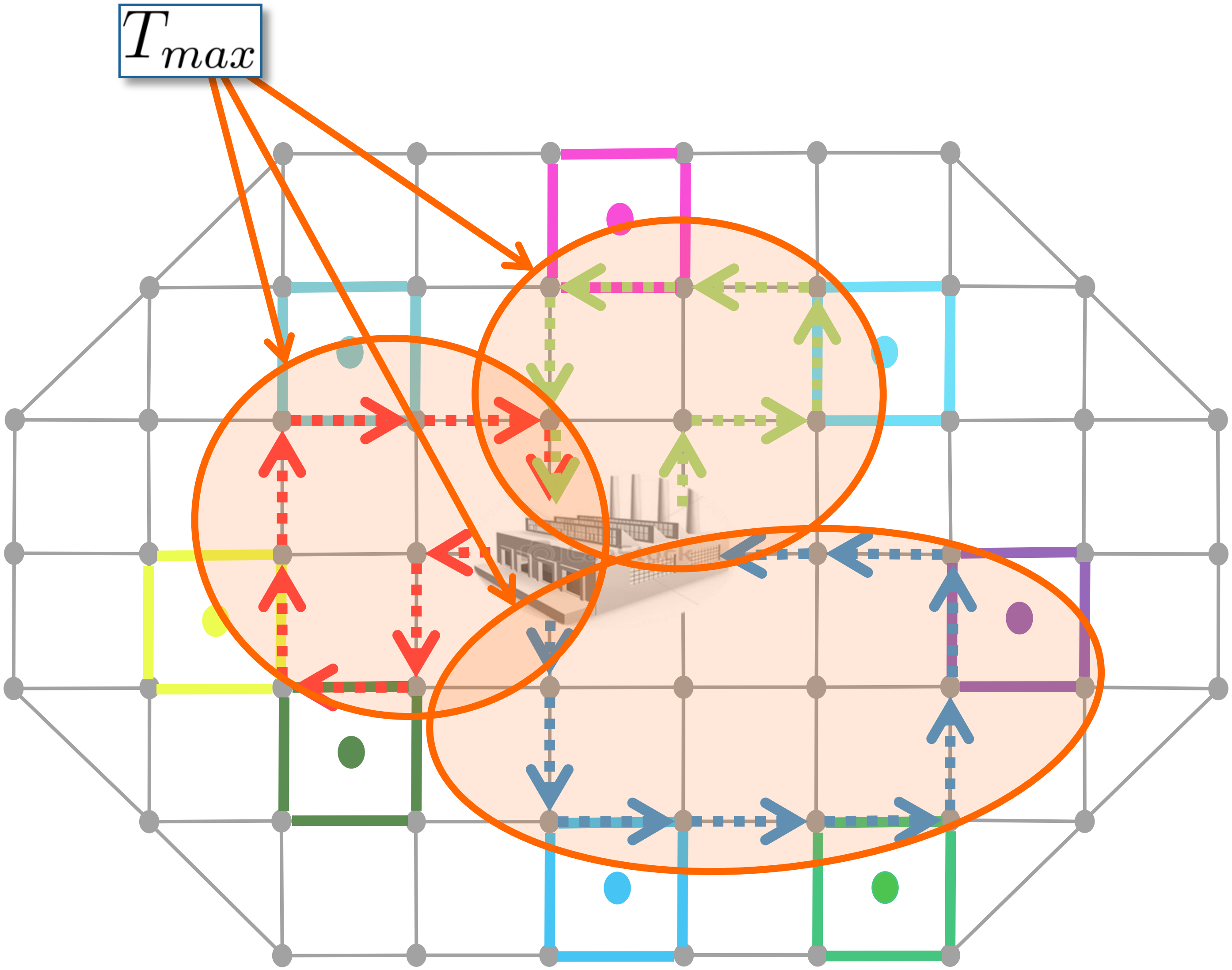
The **DC-CEARP** is a generalization of the CEARP using **k vehicles**, where the **maximum length** of each route is **limited**.

Find a set of ***k-routes*** of ***minimum total lenght***, such that each customer is served and each route length does not exceed a given limit.

Problem definition (k=3):



Problem definition (k=3):



Heuristic algorithm for the DC-CEARP

Input: G, $\mathbb{H}$, T_{max}, $Marg$, $iter\_max$, $time\_limit$

Output: S_{best}

$T_L \leftarrow (1+Marg) \times T_{max}$;

$iter \leftarrow 0$;

while $time\_limit$ is not reached AND $iter \leq iter\_max$ **do**

  for each Constructive algorithm **do**

    $S_c \leftarrow$ Constructive algorithm(T_L);

    $S_i \leftarrow$ Local-Search(S_c, T_L);

      2-Exchange(S_i, T_L);

      Destroy and Repair(S_i, T_L);

    $S_o \leftarrow$ Routes optimization(S_i, T_{max});

    if S_o is feasible and better than S_{iter}

      $S_{iter} \leftarrow S_o$;

  if S_{iter} is feasible and better than S_{best} **then**

    $S_{best} \leftarrow S_{iter}$;

    $iter \leftarrow 0$;

  else

    $iter \leftarrow iter + 1$;

Constructive Algorithm

Local Search

Route Optimization

Constructive Algorithms

1.- Parallel Constructive

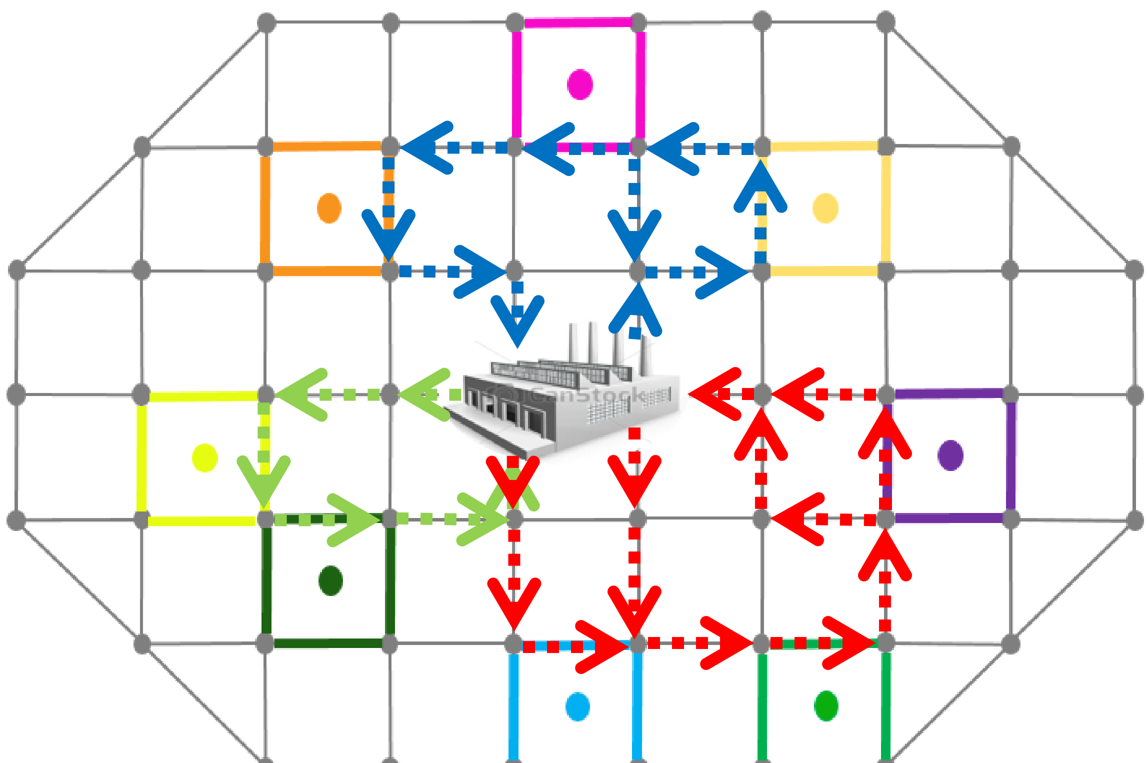
2.- Sequential Constructive

Parallel Constructive

1. ***Initialize all routes:***
   - ***Random Initialization***
   - ***Best Applicants Initialization***

2. ***Complete all routes***

```
procedure PARALLEL CONSTRUCTIVE(ConsPar, R, Asig)
    for i = 1 : K do
        InitializeRoute(Var, i,R, Asig);
    end for
    for j = 1 : m do
        if (AllocatedCustomer(j) == False) then
            CostumerAllocation(j, R, Asig);
        end if
    end for
end procedure
```



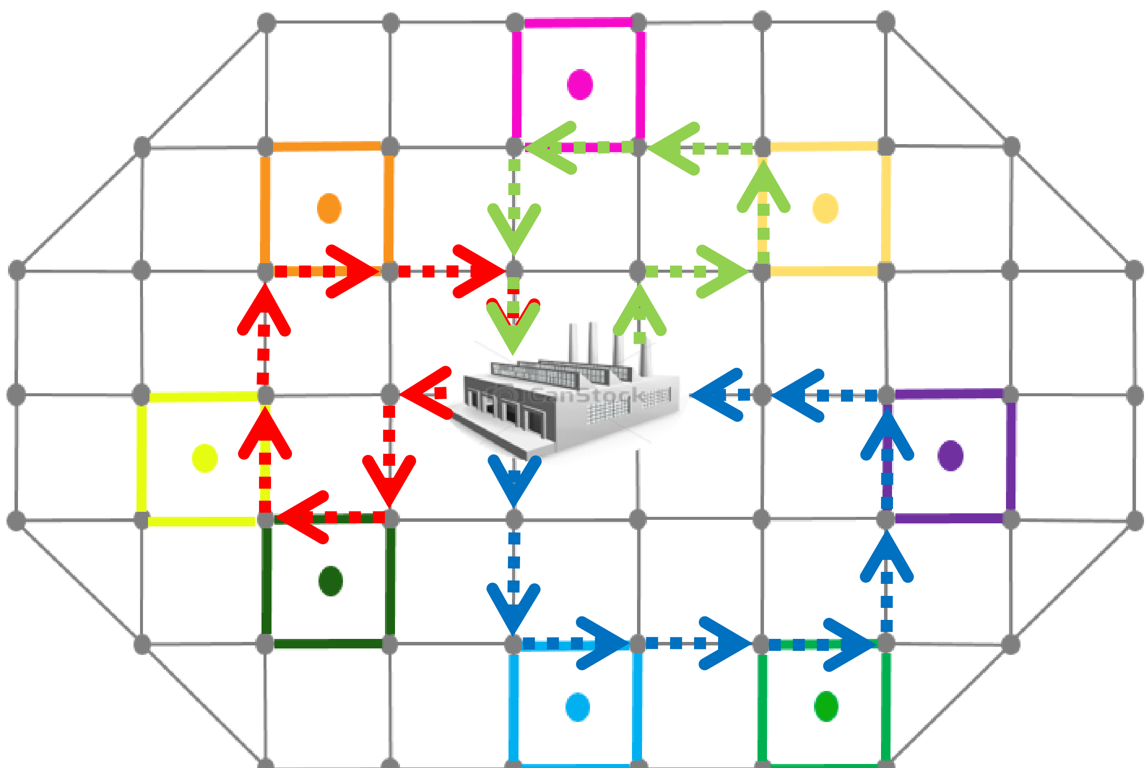
Heuristic algorithm for the DC-CEARP

Sequential Constructive

For each Route r:

1) *Initialize r*
   - *Random Initialization*
   - *Best Applicants Initialization*
2) *Complete r*

procedure SEQUENTIAL CONSTRUCTIVE($ConsSec$, R, $Asig$)
    for $i = 1 : K$ **do**
        InitializeRoute(Var, i,R, $Asig$);
        CompletionRoute(i,R, $Asig$, T_{Tope});
    end for
end procedure



Heuristic algorithm for the DC-CEARP

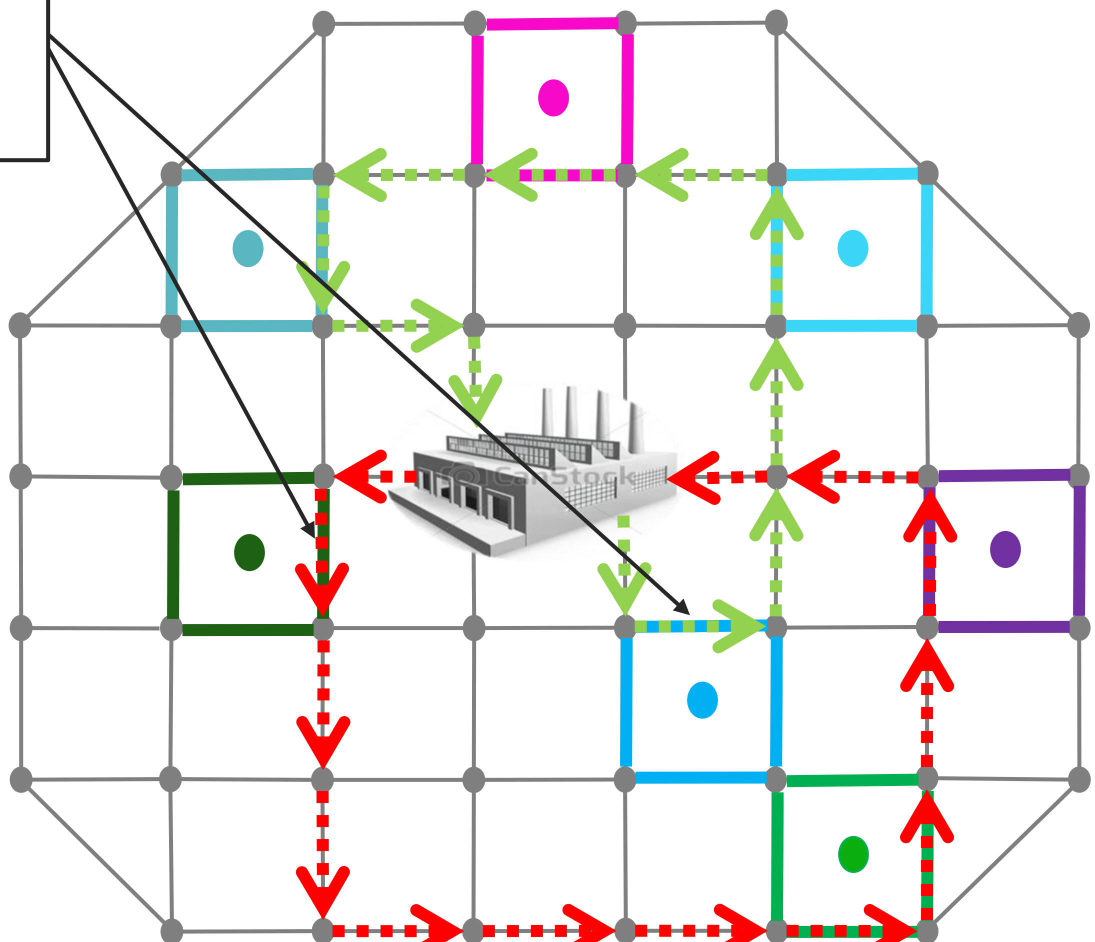
Local Search

2 - Exchange

Destroy and Repair

2-Exchange

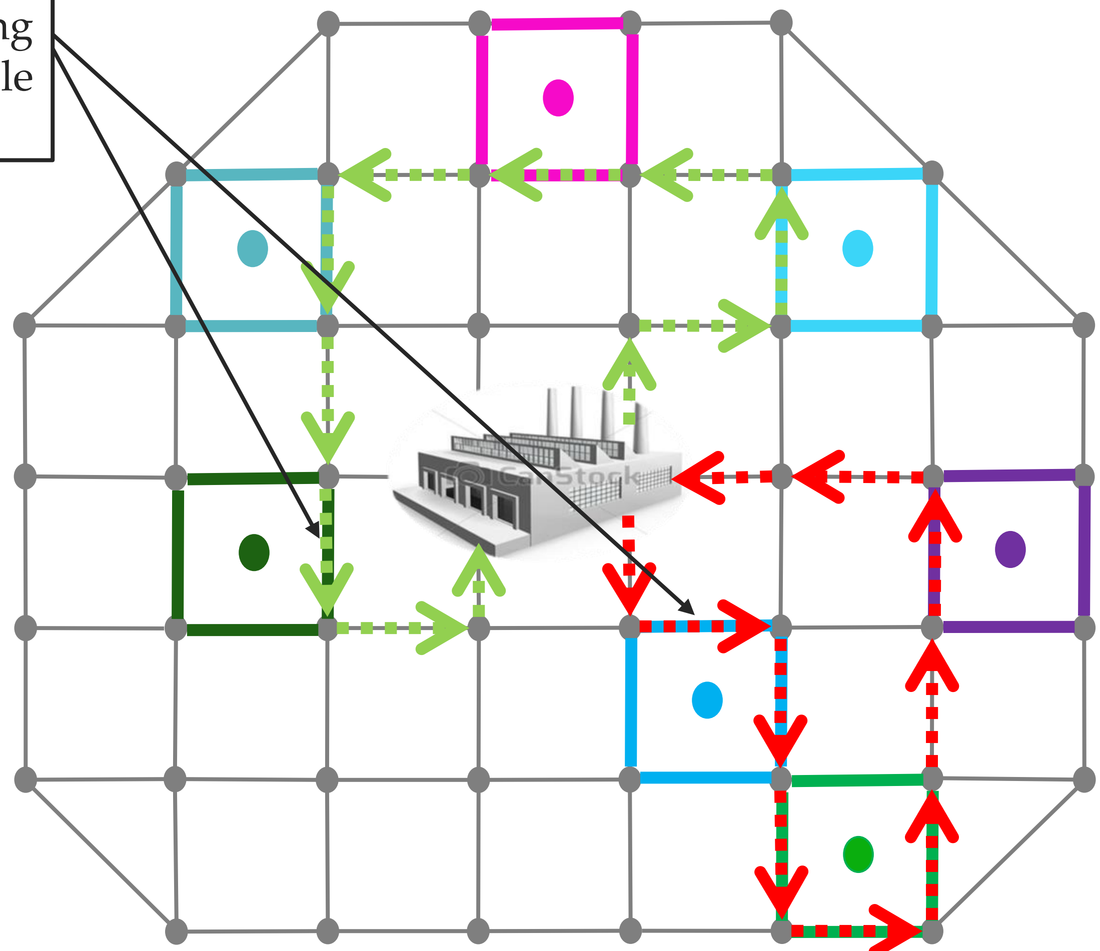
1. Select a required arc from two different routes.



2-Exchange

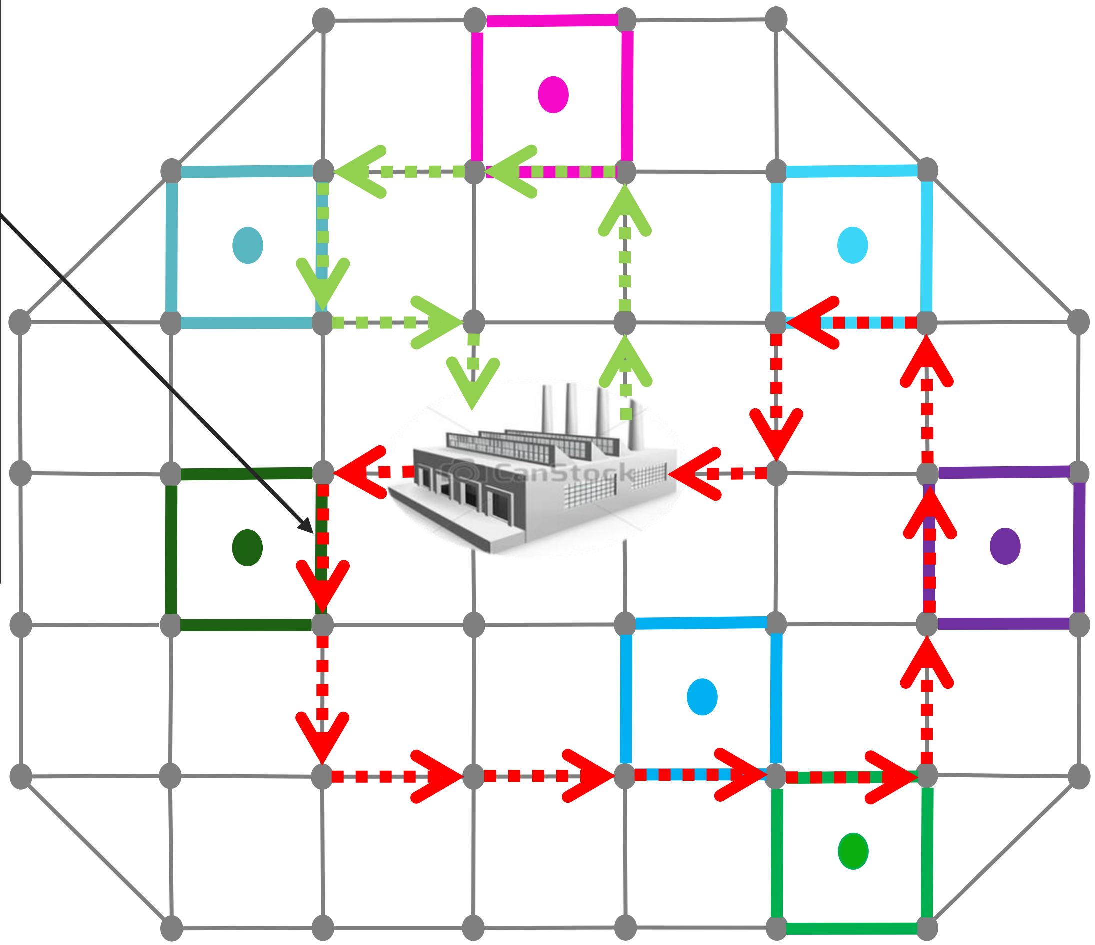
1. Select a required arc from two different routes.

2. The selected arcs are exchanged, introducing them in the best posible position.



Destroy and Repair

1. A required arc is randomly selected and removed.

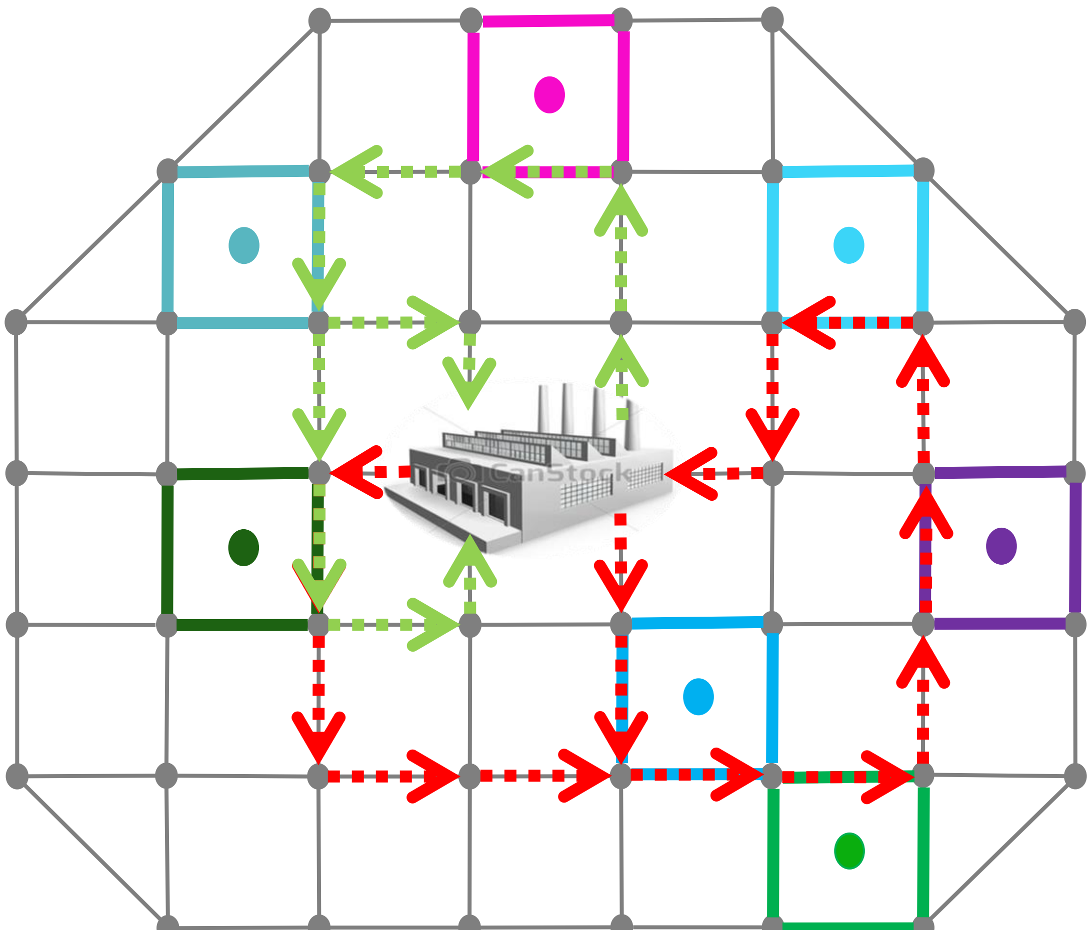


Destroy and Repair

1. A required arc is randomly selected and removed.
2. The resultant route is closed by the shortest path.
3. Are all costumers served?

   a. No: it is included in the best position of the nearest route.
   b. Yes: no changes are made.

The algorithm repeats the procedure removing 1, 2 and 5 arcs simultaneously



Route Optimization

Route Optimization

Input: G, $\mathbb{H}$, T_{max}, $Marg$, $iter\_max$,
Output: S_{best}

$T_L \leftarrow (1{+}Marg) \times T_{max}$;
$iter \leftarrow 0$;
while $time\_limit$ is not reached AN
  for each Constructive algorithm d
    $S_c \leftarrow$ Constructive algorithm(T_L
    $S_i \leftarrow$ Local-Search(S_c, T_L);
      2-Exchange(S_i, T_L);
      Destroy and Repair(S_i, T_L);
    $\boxed{S_o \leftarrow \text{Routes optimization}(S_i, T_{max});}$
    if S_o is feasible and better than S_{iter} **then**
      $S_{iter} \leftarrow S_o$;
  if S_{iter} is feasible and better than S_{best} **then**
    $S_{best} \leftarrow S_{iter}$;
    $iter \leftarrow 0$;
  else
    $iter \leftarrow iter + 1$;

Has the route been optimized before?

a. Yes! we have the minimum cost route that serve a set of customers.

b. No! we optimize the route using a B&C for the CEARP (Ávila et al. 2017)

Computational experiments

Computational experiments

Computing Environment:

- Intel Core i7-6700 @ 3.40GHz processor;
- 32 GBytes of RAM;
- Coded in C++;
- B&C with CPLEX 12.7;

Stopping rules:

1. Maximum number of iterations without improving the solution;
2. Time Limit.

Instances for the DC-CEARP

To test the performance of the heuristic algorithm, we used four different sets of instances.

	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert A_R\rvert$		$\lvert A_{NR}\rvert$		$\lvert\mathbb{H}\rvert$	
			Min	Max	Min	Max	Min	Max
Random50	50	300	105	292	7	193	10	100
Random75	75	450	143	438	10	305	15	150
Albaida	116	174	83	99	75	91	19	34
Madrigueras	196	316	152	181	135	164	23	48

Computational Results

Instances with known optimal solution

2 Vehicle	#Inst	#Opt	#No Opt	Gap(%)	#No Sol	Time (s) MH1	Time (s) B&C
Random50	12	9	3	4,48	0	13,48	45,90
Random75	12	8	4	2,67	0	22,09	388,5
Albaida	24	21	3	0,32	0	117,19	34
Madrigueras	24	18	6	1,20	0	323,19	224,1
	72	**56**	**16**	**2,17**	**0**	**118,99**	**173,13**

30%

3 Vehicle	#Inst	#Opt	#No Opt	Gap(%)	#No Sol	Time (s) MH1	Time (s) B&C
Random50	11	9	2	0,42	0	26,90	83,00
Random75	12	6	6	3,20	0	26,20	603,1
Albaida	24	22	2	1,15	0	152,33	89,9
Madrigueras	21	13	8	3,23	0	327,85	894,1
	68	**50**	**18**	**2,00**	**0**	**133,32**	**417,53**

69%

Computational Results

Instances with known optimal solution

4 Vehicle	#Inst	#Opt	#No Opt	Gap(%)	#No Sol	Time (s) MH1	Time (s) B&C
Random50	9	5	4	2,08	0	21,07	179,30
Random75	10	5	5	3,41	0	36,08	771,1
Albaida	21	18	3	1,23	0	159,47	338,7
Madrigueras	13	9	4	4,19	0	251,76	1625,2
	53	**37**	**16**	**2,73**	**0**	**117,09**	**728,58**

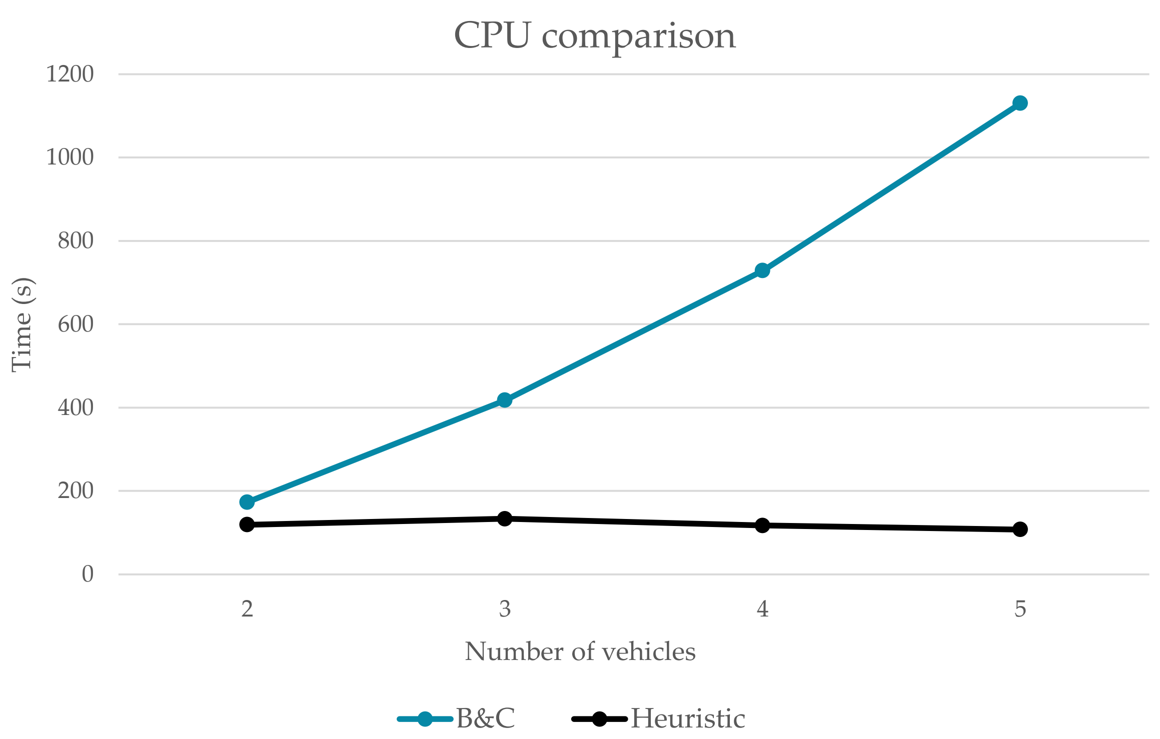
84%

5 Vehicle	#Inst	#Opt	#No Opt	Gap(%)	#No Sol	Time (s) MH1	Time (s) B&C
Random50	2	1	1	0,57	0	35,23	456,40
Random75	4	2	2	2,57	0	34,81	1301,1
Albaida	17	11	4	0,96	2	187,88	316,1
Madrigueras	5	4	1	3,47	0	172,04	2447,2
	28	**18**	**8**	**1,89**	**2**	**107,49**	**1130,20**

90%

Computational Results

Instances with known optimal solution



Computational Results

Instances with unknown optimal solution

	B&C	Matheuristic
# of feasible solutions	23/30	30/30
Average UB	14573.22	14528.73
# of best solutions	15/30	17/30
Average time	3600	405.34

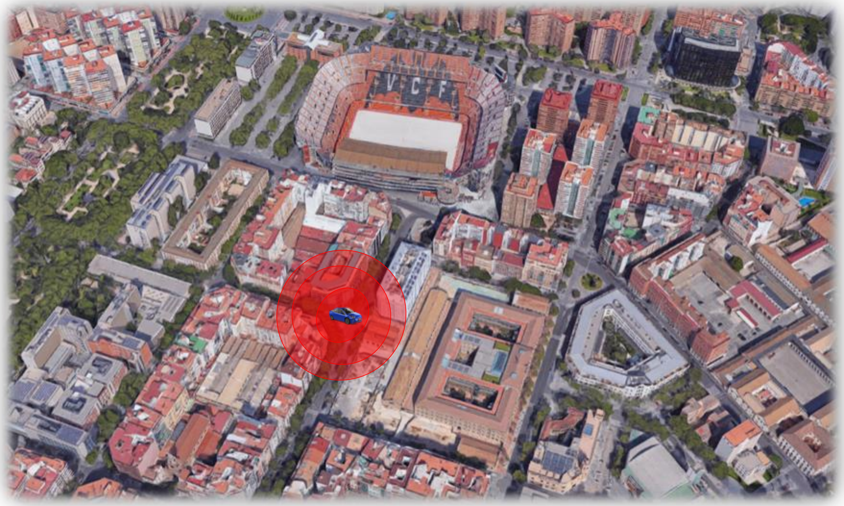
Future advancements

- Polyhedral study of the problem;
- Integration of the heuristic in a B&C framework;
- Inclusion of other real-world constraints;

Feasible solutions for the Distance Constrained Close-Enough Arc Routing Problem

Á. Corberán [(1)], I. Plana [(1)], M. Reula [(1)] and J.M. Sanchis [(2)]

Thanks for your attention!



IX International Workshop on Locational Analysis and Related Problems

VNIVERSITAT DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA