Data Gathering in Sensor Networks with Data Mules: Global and Local Approaches

Pablo Luiz Araújo Munhoz[1,2]
Advisor: Lúcia Drummond [1]
Advisor: Philippe Michelon [2]

[1]Universidade Federal Fluminense
Instituto de Computação

[2]Université d'Avignon et des Pays de Vaucluse
Laboratoire Informatique d'Avignon

May, 04 de 2017

Agenda

Introduction

Introduction

- Wireless Sensor Networks (WSN) have received much attention in last decades
- Communication is exclusively wireless
- There are many practical applications
- The information routing is one of the main problems of WSN
- Mobile agent (data mule) is responsible to perform the network communication
- Data mule: greater processing and memory capacities, and energy availability
- Data transfer: mule in communication range of a sensor

Introduction

Two ways of dealing with the problem are studied:

- Global approach: complete knowledge about the network
  - Data Mule Scheduling Problem
- Local approach: no previous knowledge about the network
  - Data Mule Routing Problem

Motivation

Motivation

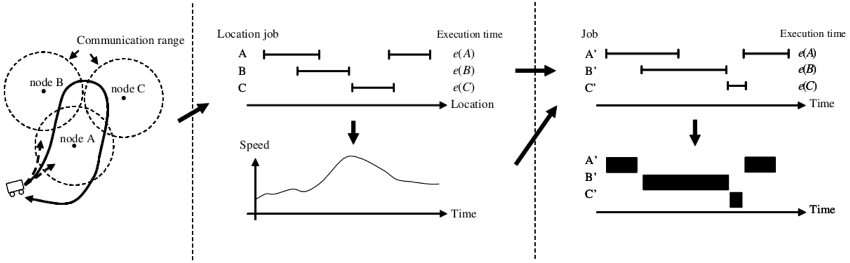
- Real world applications
  - Reduce the energy consumption (limited resource)
  - Reduce the latency of messages
- Challenge to develop efficient algorithms
  - $\mathcal{NP}$-Hard
- Emerging theme
  - Monitoring environmental parameters in remote areas
- There are few studies in literature that explore contribution of combinatorial optimization for this problem

Data Mule Scheduling Problem

Problem definition

Data Mule Scheduling Problem – DMSP

- Proposed by Zhao e Ammar (2003)
- Common objectives: **minimize the service time**, path distance, energy consumption, messages latency



(a) Path selection (b) Speed control (c) Job scheduling

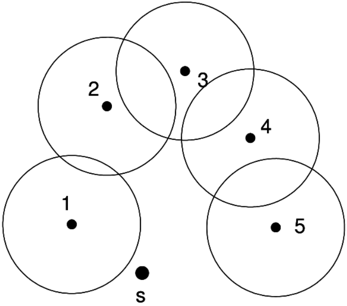
Related Works

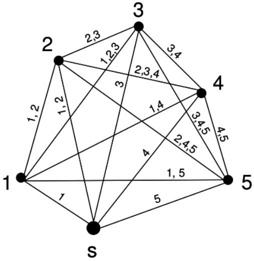
Table: Data mule based algorithms

Authors	**Sensors Movement**	**Data Mule**	**Characteristics**		**Algorithm**
			Dest.	**Communication**	
Zhao and Ammar (2003)	Static	Single	Sensors	End-to-end	Exact algorithm
Zhao *et al.* (2005)	Static	Multiple	Sensors	End-to-end	Assign and route algorithm for TSP
Bin Tariq *et al.* (2006)	Dynamic	Single	Sensors	End-to-end	Optimized Waypoints
Yuan *et al.* 2007	Static	Single	BS	End-to-end	Evolutionary Algorithm
Ma and Yang (2007)	Static	Single	BS	End-to-end	Clustering Algorithm
Xing *et al.* (2008)	Static	Single	Sinks	Multi-hop	Routing tree with pickup points
Rao *et al.* (2008)	Dynamic	Single	BS	Multi-hop	Ant colony
Ngai *et al.* (2009)	Static	Multiple	BS	End-to-end	Spanning tree based Algorithm
Sugihara *et al.* (2010)	Static	Single	BS	End-to-end	Mathematical Formulation
Sugihara *et al.* (2011)	Static	Single	BS	End-to-end	Shortest Path
Wichmann (2012)	Static	Single	Sinks	End-to-end	Constructive heuristics
Ma *et al.* (2013)	Static	Multiple	BS	End-to-end	Spanning tree covering algorithm
Our work	**Static**	**Single**	**BS**	**End-to-end**	

Problem representation

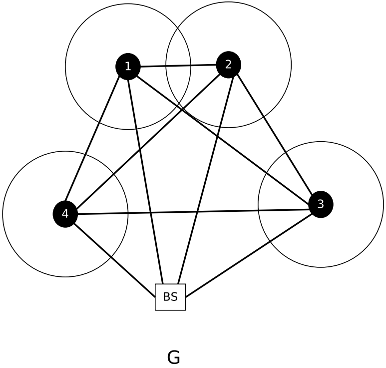
- Solution representation
  - Complete graph G
  - The nodes represents the sensors
  - The edges between two nodes represent the path
    - Have the information about the sensors that can be served

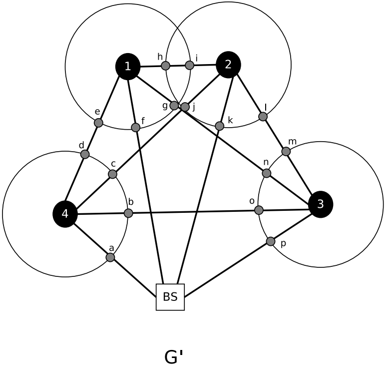




Problem representation

Pre-processing phase: fictitious nodes



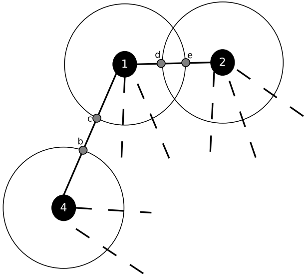


Formal definition - DMSP

- Be $G = (N, E)$ a complete graph
  - $N = \{0, 1, 2, ..., n\}$ and $N_a = N \setminus \{0\}$
  - Node 0 represents the Base Station
  - $E = \{(i, j) : i, j \in N, i \neq j\}$
  - $K = \{v_1, v_2, v_3, ...v_k, ..., v_p\}$
  - Each node $i \in N_a$ have:
    - s_i: service demand
    - r_i: radio range
    - $rate_i$: amount of data that the node i can transmit per time unit
- $G' = (N', E')$
  - $N' = N \cup N_f$
  - E'
  - Edges e
    - $c_{(p,q)}$: distance between the nodes p and q, $(p, q) \in E'$

Formal definition - DMSP

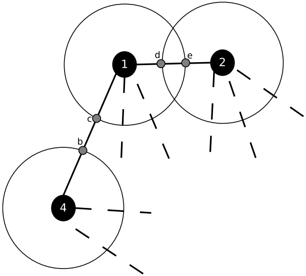
- Definitions
  - $A_i = \{(p,q) \in E' : \text{data from sensor } i \in N \text{ can be collect in } (p,q) \in E'\}$



- In this example:
  - $A_1 = \{(c,1),(1,d),(d,e),...\}$
  - $A_2 = \{(d,e),(e,2),...\}$

Formal definition - DMSP

- Definitions
  $Fict_{pq} = \{i : \text{the node } i \in N \text{ can be serve in the edge } (p,q) \in E'\}$



$Fict_{4b} = \{4\}$
$Fict_{de} = \{1,2\}$

Formal definition - DMSP

- Definitions
  - Time to traverse the edge $(p, q) \in E'$ using the speed $v_k \in K$
    - $t_{pq}^k = \dfrac{c_{pq}}{v_k} \qquad \forall (p, q) \in E', \forall k \in K$
  - Maximum amount of data collected in edge $(p, q) \in E'$ using the speed $v_k \in K$
    - $d_{pq}^k = t_{pq}^k \; rate_i \qquad \forall i \in N, \forall (p, q) \in A_i, \forall k \in K$

Studied cases

For the DMSP, three different variants were tackled:

- DMSP with Constant Speed
- DMSP with Discrete Speeds
- DMSP with Continuous Speeds

DMSP with Constant Speed

Consider that $|K| = 1$, i.e., there is only one available speed v for the data mule.

Two proposed mathematical formulations:

- Mathematical formulation using speed
- Mathematical formulation using time

DMSP with Constant Speed

Mathematical formulation using speed

Decision variables

$$x_{pq} = \begin{cases} 1, & \text{If the edge } (p,q) \in E' \text{ be used in the solution} \\ 0, & \text{Otherwise} \end{cases}$$

$$z_{pq} = \begin{cases} \text{Flow variable associated with each of the edges } (p,q) \in E' \\ z_{pq} \in \mathbb{Z}^+ \end{cases}$$

$$y_p = \begin{cases} 1, & \text{If the node } p \in N' \text{ belong to the current solution} \\ 0, & \text{Otherwise} \end{cases}$$

DMSP with Constant Speed

Mathematical formulation using speed

$$min \sum_{p \in N'} \sum_{q \in N'} t_{pq} x_{pq} \tag{1}$$

$$s.t.$$

$$\sum_{p \in N'} x_{pl} + \sum_{q \in N'} x_{lq} = 2y_l \qquad \forall l \in N' \tag{2}$$

$$\sum_{q \in N'} z_{0q} = 1 \tag{3}$$

$$\sum_{q \in N'} z_{lq} = \sum_{p \in N'} z_{pl} + y_l \qquad \forall l \in N' \setminus \{0\} \tag{4}$$

$$\sum_{p \in N'} z_{p0} = \sum_{p \in N'} y_p + 1 \tag{5}$$

$$x_{pq} \leq z_{pq} \qquad \forall (p,q) \in E' \tag{6}$$

$$x_{pq} \geq \frac{z_{pq}}{|N'| + 1} \qquad \forall (p,q) \in E' \tag{7}$$

$$\sum_{(p,q) \in A_i} d_{pq} x_{pq} \geq s_i \qquad \forall i \in N \tag{8}$$

$$x_{pq} \in \{0,1\} \qquad \forall (p,q) \in E' \tag{9}$$

$$y_p \in \{0,1\} \qquad \forall p \in N' \tag{10}$$

$$z_{pq} \geq 0 \qquad \forall (p,q) \in E', z_{pq} \in \mathbb{Z}^+ \tag{11}$$

DMSP with Constant Speed

Mathematical formulation using time

Decision variables

$$x_e = \begin{cases} 1, & \text{If the edge } e \in E' \text{ belongs to the solution} \\ 0, & \text{Otherwise} \end{cases}$$

$$z_{pq} = \begin{cases} \text{Flow variable related of each edge} \\ (p,q) \in E', z_{pq} \in \mathbb{Z}^+ \end{cases}$$

$$y_p = \begin{cases} 1, & \text{If the node } p \in N' \text{ is being used in the current solution} \\ 0, & \text{Otherwise} \end{cases}$$

$$H_e^i = \text{time spend by the mule in edge } e \in E' \text{ attending the sensor } i \in N$$

DMSP with Constant Speed

Mathematical formulation using time

Another definitions

- $T_e = \frac{c_e}{v}$
- Let $G'(N', W \cup U)$, where $E' = W \cup U$.
  - W: edges that can attend some node
  - U: edges that can not attend some node

DMSP with Constant Speed

Mathematical formulation using time

$$min \sum_{e \in W} \sum_{i \in N} H_e^i + \sum_{e \in U} \sum_{i \in N} H_e^i \tag{12}$$

$$s.t.$$

$$\sum_{p \in N'} x_{pl} + \sum_{q \in N'} x_{lq} = 2y_l \qquad \forall l \in N' \tag{13}$$

$$\sum_{q \in N'} z_{0q} = 1 \tag{14}$$

$$\sum_{q \in N'} z_{lq} = \sum_{p \in N'} z_{pl} + y_l \qquad \forall l \in N' \setminus \{0\} \tag{15}$$

$$\sum_{p \in N'} z_{p0} = \sum_{p \in N'} y_p + 1 \tag{16}$$

$$x_{pq} \leq \frac{z_{pq}}{2} \qquad \forall (p,q) \in E' \mid p \neq 0 \tag{17}$$

$$x_{pq} \geq \frac{z_{pq}}{|N'|+1} \qquad \forall (p,q) \in E' \tag{18}$$

$$x_{0q} = z_{0q} \qquad \forall q \in E' \tag{19}$$

$$\sum_{i \in Fict_e} H_e^i = T_e x_e \qquad \forall e \in E' \tag{20}$$

$$\sum_{i \in N} H_e^i = T_e x_e \qquad \forall e \in U \tag{21}$$

$$\sum_{e \in A_i} rate_i H_e^i \geq s_i \qquad \forall i \in N \tag{22}$$

$$x_e \in \{0,1\} \qquad \forall e \in E' \tag{23}$$

$$y_p \in \{0,1\} \qquad \forall p \in N' \tag{24}$$

$$z_e \geq 0 \qquad \forall e \in E', z_{pq} \in \mathbb{Z}^+ \tag{25}$$

$$H_e^i \geq 0 \qquad \forall e \in E', i \in N, H_e^i \in \mathbb{R}^+ \tag{26}$$

DMSP with Constant Speed

Heuristics

- Two constructive heuristics: IMB and CML
- Local search: RVND using *Swap*, *Shift* and *Swap(2,1)*
- Two metaheuristics based heuristics: GRVND and GVNS-RVND

DMSP with Constant Speed

Solution representation

- A vector is used as solution representation
- When a node is not present in a solution, it is added in the end of the vector

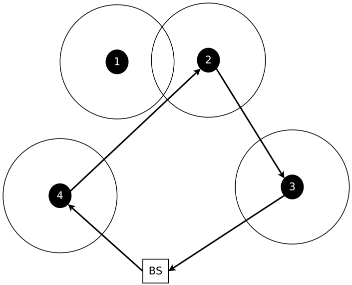


Figure: Representation of solution s

Example solution $s = [0, 4, 2, 3, 0, 1]$ represents the route made by the mule

DMSP with Constant Speed

Constructive heuristics

- IMB: based in Nearest Insertion
  - Criterion for add: $\frac{attendance}{distance}$
- CML: based in Longest Insertion
  - Criterion for remove: $\frac{distance}{attendance}$

DMSP with Constant Speed

RVND

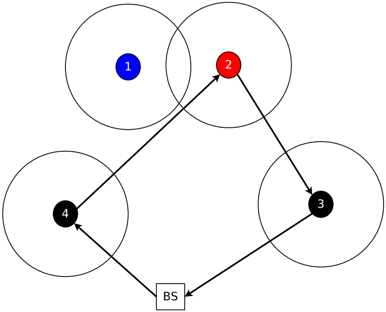
A local search based in VND (Variable Neighborhood Descent) using a random neighborhood ordering (RVND)

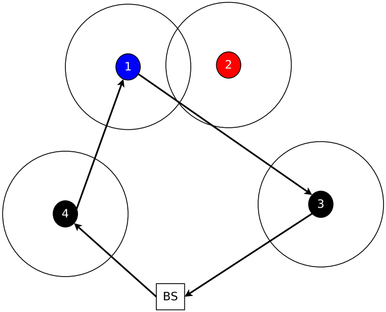
- N^{Swap}: swapping between two sensors in a given route
- N^{Shift}: reallocating a sensor to another point on the route
- $N^{Swap(2,1)}$: swapping between two consecutive sensors and another sensor belonging to the solution

DMSP with Constant Speed

Swap

Swap between nodes 2 and 1





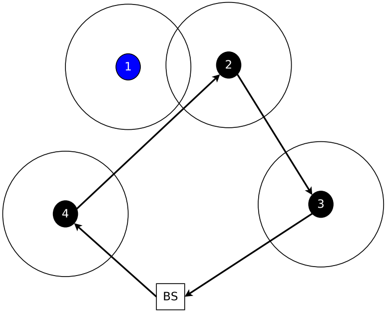
[0, 4, **2**, 3, 0 ,**1**]

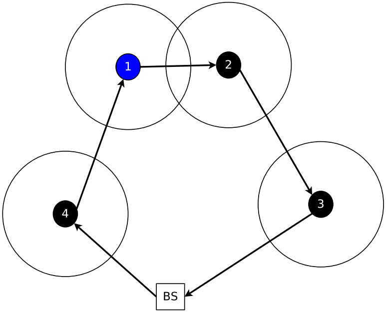
[0, 4, **1**, 3, 0, **2**]

DMSP with Constant Speed

Shift

Shift node 1 to position 1





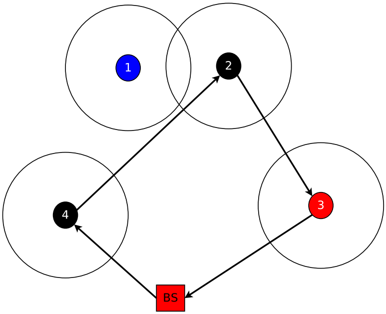
[0, 4, 2, 3, 0 ,**1**]

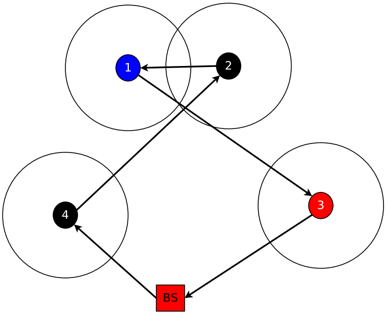
[0, 4, **1**, 2, 3, 0]

DMSP with Constant Speed

Swap(2,1)

Swap between nodes [3, 0] and 1





[0, 4, 2, **3**, **0** ,**1**]

[0, 4, 2, **1**, **3**, **0**]

DMSP with Constant Speed

Algorithm 1: RVND

Input: neighborhoods N^{Swap}, N^{Shift} and $N^{Swap(2,1)}$ in a random order, Solution s

$k \leftarrow 1$;
while $k \leq 3$ **do**
    Find the best neighbor $s' \in N^k(s)$;
    if $f(s') < f(s)$ **then**
        $s \leftarrow s'$;
        $k \leftarrow 1$;
    else
        $k \leftarrow k + 1$;
    end
end
return s

DMSP with Constant Speed

Metaheuristic based heuristics

GRASP framework

- Construction
- Local Search

VNS framework

- Construction
- Local Search: systematic change of neighborhood structures
- Shake procedure
- GVNS: more than one neighborhood in a local search

DMSP with Discrete Speeds

The data mule can choose between a set K of speeds at each edge $e \in E$, $|K| > 1$.
Mathematical formulation using a set of speeds K was proposed.

DMSP with Discrete Speeds

Mathematical formulation using set of speeds

Decision variables

$$x_{pq}^{k} = \begin{cases} 1, & \text{If the edge } (p,q)_k \in E' \text{ be used in the solution with } k \in K \\ 0, & \text{Otherwise} \end{cases}$$

$$z_{pq}^{k} = \begin{cases} \text{Flow variable associated with each of the edges } (p,q)_k \in E' \\ \ \text{with speed } k \in K,\, z_{pq}^{k} \in \mathbb{Z}^{+} \end{cases}$$

$$y_{p} = \begin{cases} 1, & \text{If the node } p \in N' \text{ belong to the current solution} \\ 0, & \text{Otherwise} \end{cases}$$

DMSP with Discrete Speeds

Mathematical formulation using set of speeds

$$min \sum_{k\in K}\sum_{p\in N'}\sum_{q\in N'} t_{pq}^k x_{pq}^k \tag{27}$$

$$s.t.$$

$$\sum_{k\in K}\sum_{p\in N'} x_{pl}^k + \sum_{k\in K}\sum_{q\in N'} x_{lq}^k = 2y_l \qquad \forall l \in N' \tag{28}$$

$$\sum_{k\in K}\sum_{q\in N'} z_{0q}^k = 1 \tag{29}$$

$$\sum_{k\in K}\sum_{q\in N'} z_{lq}^k = \sum_{k\in K}\sum_{p\in N'} z_{pl}^k + y_l \qquad \forall l \in N' \setminus \{0\} \tag{30}$$

$$\sum_{k\in K}\sum_{p\in N'} z_{p0}^k = \sum_{p\in N'} y_p + 1 \tag{31}$$

$$x_{pq}^k \le z_{pq}^k \qquad \forall (p,q) \in E', k \in K \tag{32}$$

$$x_{pq}^k \ge \frac{z_{pq}^k}{|N'|+1} \qquad \forall (p,q) \in E', k \in K \tag{33}$$

$$\sum_{k\in K} x_{pq}^k \le 1 \qquad \forall (p,q) \in E' \tag{34}$$

DMSP with Discrete Speeds

Mathematical formulation using set of speeds

$$x^k_{pq} = x^k_{p'q'} \qquad \forall (i,j) \in E, \quad \forall (p,q) \subseteq (i,j), \forall (p',q') \subseteq (i,j) \; | (p,q) \neq (p',q') \quad \forall k \in K \tag{35}$$

$$\sum_{k \in K} \sum_{(p,q) \in A_i} d^k_{pq} x^k_{pq} \geq s_i \qquad \forall i \in N \tag{36}$$

$$x^k_{pq} \in \{0,1\} \qquad \forall (p,q) \in E', k \in K \tag{37}$$

$$y_p \in \{0,1\} \qquad \forall p \in N' \tag{38}$$

$$z^k_{pq} \geq 0 \qquad \forall (p,q) \in E', k \in K, z^k_{pq} \in \mathbb{Z}^+ \tag{39}$$

DMSP with Continuous Speeds

The mathematical formulation considers only lower and upper bounds for the speeds of Data Mule ($speed_{min}$ and $speed_{max}$, respectively).

- Let $G'(N', W \cup U)$, where $E' = W \cup U$.
  - W: edges that can attend some node
  - U: edges that can not attend some node
- Two new definitions:
  - $T^e_{min} = \frac{c_e}{speed_{max}}$
  - $T^e_{max} = \frac{c_e}{speed_{min}}$

A mathematical formulation using time concepts was proposed.

DMSP with Continuous Speeds

Decision variables

$$
\begin{aligned}
x_e &= \begin{cases} 1, & \text{If the edge } e \in E' \text{ be used in the solution} \\ 0, & \text{Otherwise} \end{cases} \\
z_e &= \{ \text{ Flow variable associated with each of the edges } e \in E' \in \mathbb{Z}^+ \\
y_p &= \begin{cases} 1, & \text{If the node } p \in N' \text{ belong to the current solution} \\ 0, & \text{Otherwise} \end{cases} \\
H_e^i &= \text{time spend by the mule in edge } e \in E' \text{ attending the sensor } i \in N
\end{aligned}
$$

DMSP with Continuous Speeds

$$min \sum_{e \in W} \sum_{i \in Fict_e} H_e^i + \sum_{e \in U} \sum_{i \in N} H_e^i \tag{40}$$

$$s.t.$$

$$\sum_{p \in N'} x_{pl} + \sum_{q \in N'} x_{lq} = 2y_l \qquad \forall l \in N' \tag{41}$$

$$\sum_{q \in N'} z_{0q} = 1 \tag{42}$$

$$\sum_{q \in N'} z_{lq} = \sum_{p \in N'} z_{pl} + y_l \qquad \forall l \in N' \setminus \{0\} \tag{43}$$

$$\sum_{p \in N'} z_{p0} = \sum_{p \in N'} y_p + 1 \tag{44}$$

$$x_{pq} \leq \frac{z_{pq}}{2} \qquad \forall (p,q) \in E' \mid p \neq 0 \tag{45}$$

$$x_{pq} \geq \frac{z_{pq}}{|N'| + 1} \qquad \forall (p,q) \in E' \tag{46}$$

$$x_{0q} = z_{0q} \qquad \forall q \in E' \tag{47}$$

DMSP with Continuous Speeds

$$\sum_{i \in Fict_e} H_e^i \leq T_{max}^e x_e \qquad \forall e \in W \tag{48}$$

$$\sum_{i \in Fict_e} H_e^i \geq T_{min}^e x_e \qquad \forall e \in W \tag{49}$$

$$\sum_{i \in N} H_e^i = T_{min}^e x_e \qquad \forall e \in U \tag{50}$$

$$\sum_{e \in A_i} rate_i H_e^i \geq s_i \qquad \forall i \in N \tag{51}$$

$$x_e \in \{0, 1\} \qquad \forall e \in E' \tag{52}$$

$$y_p \in \{0, 1\} \qquad \forall p \in N' \tag{53}$$

$$z_e \geq 0 \qquad \forall e \in E', z_{pq} \in \mathbb{Z}^+ \tag{54}$$

$$H_e^i \geq 0 \qquad \forall e \in E', i \in N, H_e^i \in \mathbb{R}^+ \tag{55}$$

Computational Experiments and Analysis

Instances

Instances

- A set of instances was created
  - Sensors: 6, 7, 8, 9, 10, 11, 16, 21, 31, 41 and 51
  - Coordinates of the sensors were randomly defined
  - *Grid*: (0, 300) × (0, 300)
  - The Base Station in three distinct configurations: `Central`, `Eccentric` and `Random`
  - Radius range (r_i): [1,50]
  - Transmission rate ($rate_i$): [1,10]
  - Service demands (s_i): [1,20]
  - For each # sensors × Base station position: 50 instances
  - Total of 1650 instances

Computer experiments and Analysis

Computational Environment

- Tests executed in a Intel Core i7 3.40 Ghz computer, with 16 GB of RAM and Linux Ubuntu
- Exact method: C++ and CPlex 12.5.1 in its default configuration
- Exact method was executed for all instances with a timeout of 1 hour (3600 seconds)
- Each test was executed 10 times per instance

Computer experiments and Analysis

Constructive Heuristics and Local Search

After an exhaustive tests:

- IMB obtains the best results
  - AVG GAP: IMB (51.46%) $\times$ CML (68.07%)
- Neighborhood structures: *Swap*, *Shift* and *Swap(2,1)*
  - Best Improvement obtains the best results
  - MIN GAP, AVG GAP and improvement of a initial solution

Computer experiments and Analysis

Proposed Heuristics

- Constructive heuristic: IMB
- Local Search: RVND with Best Improvement
- GRVND and GVNS-RVND
- Same *StoppingCriterion*: 50 iterations without improving the current solution
- Evaluation criteria: *GAP*, *EQUAL* and *IMP*
- For the GVNS-RVND:
  - $shakePerc = 30\%$ of $iterMaxWithoutImp \rightarrow$ shake + 2 moves

Computer experiments and Analysis

Proposed Heuristics

# Sensors	Base Station	GRVND – AVG GAP				GVNS-RVND – AVG GAP			
		MIN	T(s)	AVG	T(s)	MIN	T(s)	AVG	T(s)
	central	0.13%	0.085	0.22%	0.092	**0.00%**	0.082	**0.00%**	0.087
6	**eccentric**	**0.00%**	0.081	0.03%	0.092	**0.00%**	0.013	**0.00%**	0.014
	random	0.23%	0.081	0.27%	0.088	**0.00%**	0.012	**0.00%**	0.014
	central	0.01%	0.152	0.26%	0.173	**0.00%**	0.162	**0.00%**	0.173
7	**eccentric**	**0.00%**	0.158	0.25%	0.173	**0.00%**	0.024	**0.00%**	0.026
	random	**0.00%**	0.162	0.05%	0.181	**0.00%**	0.025	**0.00%**	0.027
	central	**0.00%**	0.269	0.19%	0.306	**0.00%**	0.288	**0.00%**	0.312
8	**eccentric**	**0.00%**	0.278	0.38%	0.324	**0.00%**	0.043	**0.00%**	0.046
	random	0.02%	0.281	0.40%	0.324	**0.00%**	0.044	**0.00%**	0.047
	central	0.02%	0.413	0.42%	0.486	**0.00%**	0.449	**0.00%**	0.489
9	**eccentric**	0.21%	0.409	0.36%	0.480	**0.00%**	0.065	**0.00%**	0.069
	random	0.01%	0.415	0.15%	0.489	**0.00%**	0.067	**0.00%**	0.072
	central	0.05%	0.697	0.16%	0.799	**0.00%**	0.449	**0.00%**	0.489
10	**eccentric**	**0.00%**	0.683	0.22%	0.804	**0.00%**	0.098	**0.00%**	0.106
	random	0.02%	0.739	0.13%	0.833	**0.00%**	0.102	**0.00%**	0.111
	central	0.14%	1.068	0.43%	1.276	**0.00%**	0.510	**0.00%**	0.554
11	**eccentric**	**0.00%**	1.007	0.31%	1.284	**0.00%**	0.156	**0.00%**	0.170
	random	0.03%	1.071	0.12%	1.266	**0.00%**	0.154	0.01%	0.167
	central	0.07%	6.063	0.24%	7.033	**0.00%**	0.712	0.02%	0.838
16	**eccentric**	0.02%	6.208	0.11%	7.119	**0.00%**	0.692	**0.00%**	0.803
	random	0.04%	6.060	0.09%	7.050	**0.00%**	0.714	0.01%	0.805
	central	-1.50%	22.424	-1.40%	25.822	-1.54%	2.097	-1.44%	2.737
21	**eccentric**	-0.44%	10.305	-0.31%	12.421	-0.51%	2.094	-0.46%	2.663
	random	-0.29%	10.524	-0.18%	12.534	-0.35%	2.102	-0.29%	2.675
Average		-0.05%	2.901	0.12%	3.394	-0.10%	0.465	-0.09%	0.562

Computer experiments and Analysis

Proposed Heuristics

# Sensors	**Base Station**	**GRVND – AVG GAP**				**GVNS-RVND – AVG GAP**			
		MIN	**T(s)**	**AVG**	**T(s)**	**MIN**	**T(s)**	**AVG**	**T(s)**
	central	0.13%	0.085	0.22%	0.092	**0.00%**	0.082	**0.00%**	0.087
6	**eccentric**	**0.00%**	0.081	0.03%	0.092	**0.00%**	0.013	**0.00%**	0.014
	random	0.23%	0.081	0.27%	0.088	**0.00%**	0.012	**0.00%**	0.014
	central	0.01%	0.152	0.26%	0.173	**0.00%**	0.162	**0.00%**	0.173
7	**eccentric**	**0.00%**	0.158	0.25%	0.173	**0.00%**	0.024	**0.00%**	0.026
	random	**0.00%**	0.162	0.05%	0.181	**0.00%**	0.025	**0.00%**	0.027
	central	**0.00%**	0.269	0.19%	0.306	**0.00%**	0.288	**0.00%**	0.312
8	**eccentric**	**0.00%**	0.278	0.38%	0.324	**0.00%**	0.043	**0.00%**	0.046
	random	0.02%	0.281	0.40%	0.324	**0.00%**	0.044	**0.00%**	0.047
	central	0.02%	0.413	0.42%	0.486	**0.00%**	0.449	**0.00%**	0.489
9	**eccentric**	0.21%	0.409	0.36%	0.480	**0.00%**	0.065	**0.00%**	0.069
	random	0.01%	0.415	0.15%	0.489	**0.00%**	0.067	**0.00%**	0.072
	central	0.05%	0.697	0.16%	0.799	**0.00%**	0.449	**0.00%**	0.489
10	**eccentric**	**0.00%**	0.683	0.22%	0.804	**0.00%**	0.098	**0.00%**	0.106
	random	0.02%	0.739	0.13%	0.833	**0.00%**	0.102	**0.00%**	0.111
	central	0.14%	1.068	0.43%	1.276	**0.00%**	0.510	**0.00%**	0.554
11	**eccentric**	**0.00%**	1.007	0.31%	1.284	**0.00%**	0.156	**0.00%**	0.170
	random	0.03%	1.071	0.12%	1.266	**0.00%**	0.154	0.01%	0.167
	central	0.07%	6.063	0.24%	7.033	**0.00%**	0.712	0.02%	0.838
16	**eccentric**	0.02%	6.208	0.11%	7.119	**0.00%**	0.692	**0.00%**	0.803
	random	0.04%	6.060	0.09%	7.050	**0.00%**	0.714	0.01%	0.805
	central	-1.50%	22.424	-1.40%	25.822	-1.54%	2.097	-1.44%	2.737
21	**eccentric**	-0.44%	10.305	-0.31%	12.421	-0.51%	2.094	-0.46%	2.663
	random	-0.29%	10.524	-0.18%	12.534	-0.35%	2.102	-0.29%	2.675
Average		-0.05%	2.901	0.12%	3.394	-0.10%	0.465	-0.09%	0.562

Computer experiments and Analysis

Proposed Heuristics

# Sensors	Base Station	GRVND MIN EQUAL	GRVND MIN IMP	GRVND AVG EQUAL	GRVND AVG IMP	GVNS-RVND MIN EQUAL	GVNS-RVND MIN IMP	GVNS-RVND AVG EQUAL	GVNS-RVND AVG IMP
	central	48	0	47	0	**50**	0	**50**	0
6	**eccentric**	**50**	0	49	0	**50**	0	**50**	0
	random	49	0	48	0	**50**	0	**50**	0
	central	**50**	0	47	0	**50**	0	**50**	0
7	**eccentric**	49	0	47	0	**50**	0	**50**	0
	random	**50**	0	49	0	**50**	0	**50**	0
	central	**50**	0	45	0	**50**	0	**50**	0
8	**eccentric**	**50**	0	45	0	**50**	0	**50**	0
	random	49	0	46	0	**50**	0	**50**	0
	central	49	0	40	0	**50**	0	**50**	0
9	**eccentric**	48	0	42	0	**50**	0	**50**	0
	random	**50**	0	48	0	**50**	0	**50**	0
	central	49	0	44	0	**50**	0	**50**	0
10	**eccentric**	**50**	0	46	0	**50**	0	**50**	0
	random	49	0	46	0	**50**	0	**50**	0
	central	47	0	39	0	**50**	0	**50**	0
11	**eccentric**	**50**	0	41	0	**50**	0	**50**	0
	random	49	0	47	0	**50**	0	49	0
	central	47	0	44	0	**50**	0	49	0
16	**eccentric**	48	0	45	0	**50**	0	**50**	0
	random	49	0	46	0	**50**	0	**50**	0
	central	31	18	27	<u>18</u>	31	19	30	<u>19</u>
21	**eccentric**	39	9	35	<u>9</u>	41	9	41	<u>8</u>
	random	38	11	34	<u>10</u>	31	11	39	<u>11</u>

Computer experiments and Analysis

Proposed Heuristics

# Sensors	Base Station	GRVND				GVNS-RVND			
		MIN		AVG		MIN		AVG	
		EQUAL	IMP	EQUAL	IMP	EQUAL	IMP	EQUAL	IMP
	central	48	0	47	0	**50**	0	**50**	0
6	eccentric	**50**	0	49	0	**50**	0	**50**	0
	random	49	0	48	0	**50**	0	**50**	0
	central	**50**	0	47	0	**50**	0	**50**	0
7	eccentric	49	0	47	0	**50**	0	**50**	0
	random	**50**	0	49	0	**50**	0	**50**	0
	central	**50**	0	45	0	**50**	0	**50**	0
8	eccentric	**50**	0	45	0	**50**	0	**50**	0
	random	49	0	46	0	**50**	0	**50**	0
	central	49	0	40	0	**50**	0	**50**	0
9	eccentric	48	0	42	0	**50**	0	**50**	0
	random	**50**	0	48	0	**50**	0	**50**	0
	central	49	0	44	0	**50**	0	**50**	0
10	eccentric	**50**	0	46	0	**50**	0	**50**	0
	random	49	0	46	0	**50**	0	**50**	0
	central	47	0	39	0	**50**	0	**50**	0
11	eccentric	**50**	0	41	0	**50**	0	**50**	0
	random	49	0	47	0	**50**	0	49	0
	central	47	0	44	0	**50**	0	49	0
16	eccentric	48	0	45	0	**50**	0	**50**	0
	random	49	0	46	0	**50**	0	**50**	0
	central	31	18	27	<u>18</u>	31	19	30	<u>19</u>
21	eccentric	39	9	35	<u>9</u>	41	9	41	<u>8</u>
	random	38	11	34	<u>10</u>	31	11	39	<u>11</u>

Data Mule Routing Problem

Introduction

Data Mule Routing Problem

- Information exchange in intersection between sensors spatial coverages
- Sensors are distributed in a bi-dimensional space
- Communication range equal to r
- Responsible to collect all data and take them to a base station
- Reduce the number of exchanged messages in the network and, consequently, the spent energy for data transmission

Introduction

Data Mule Routing Problem

Data Mule Routing Problem definition

- Virtual backbones modelled as a Minimum Connected Dominating Set Problem
- Data mule has to serve each node of the WSN
- No knowledge about the global network
- Should visit a minimum number of nodes to serve all demands
- Local decision
- Neighborhood covers

Motivation

- Deal with realistic scenario
- Local view
- Different characteristics

Introduction

Data Mule Routing Problem

Data Mule Routing Problem definition

- $G = (V, E)$
  - $V(G)$ placed in an Euclidean plan
  - Each edge $(i, j) \in E(G)$ exists if i and j are within their communication range (Unit disk graph)
  - $N(i)$ contains the neighbour nodes of vertex i
- Edges have no weights
- $s \in V$ is the base station
- Data mule serves a node i when located in some node $j \in N(i)$
- Each edge traversed by the mule is included in the path
- Same edge can be traversed by the data mule more than once

Connected Dominating Set based Algorithms

Neighborhood Knowledge

Global view

- Das and Bharghavan (1997): All pairs shortest path
- Zhao *et al.* (2015): Distributed mathematical formulations

1-hop

- Alzoubi *et al.* (2002): MIS and dominating tree
- Funke *et al.* (2006): CDS using distance-2-coloring algorithm
- Islam *et al.* (2008): CDS using convex-hull and MIS
- Ghaffari (2014): CONGEST Model based in DS

2-hop

- Wu and Li (1999): shortest path for a DS calculation

Theoretical Remarks

Objective

Data Mule with Global View

Input:	A graph G, and a base station node $v \in V(G)$.
Goal:	Determine a minimum closed walk W of G such that $v \in V(W)$, and for all node $x \in V(G)$, $N[x] \cap V(W) \neq \emptyset$. That is, either $x \in V(W)$ or some neighbor y of x belongs to $V(W)$.

Theoretical Remarks

Lower Bounds

Denote $d(v, w)$ as the distance between v and w in G, and $d_v = \max_{w \in V(G)} d(v, w)$.

Lemma

Let $OPT(G, v)$ be an optimal solution value for Data Mule with Global View *on G with base station v. It holds that*

$$OPT(G, v) \geq 2(d_v - 1).$$

And such lower bound can be found in $O(m)$ time.

Lemma

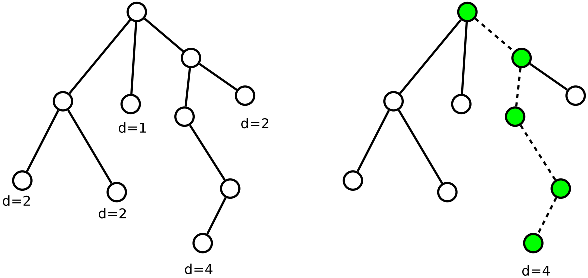
Given G, v and an integer $k \geq 1$. Let S be a set composed by the k most distant vertices of v, and T_k be a steiner tree to connect $\{v\} \cup S$. Let $LB_k = |T_k| - k + \min_{w \in S} d(v, w) - 1$. For all $k \geq 1$, it holds that

$$OPT(G, v) \geq LB_k.$$

Theoretical Remarks

Lower Bounds

LB_1

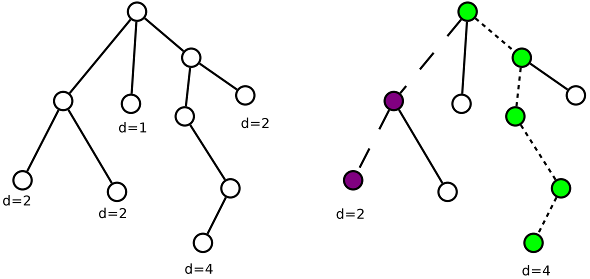


$$LB_1 = 2(d_v - 1) = 2(4 - 1) = 6$$

Theoretical Remarks

Lower Bounds

LB_2



$$LB_2 = |T_k| - k + \min_{w \in S} d(v, w) - 1$$
$$LB_2 = 6 - 2 + 2 - 1 = 5$$

Theoretical Remarks

Mathematical Formulation

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} x_{ij} \\
\text{s.t.} \quad & \sum_{j\in N(i)\cup\{i\}} y_j \geq 1, \forall i \in V && (56)\\
& \sum_{j\in\delta^+(i)} x_{ij} \geq y_i, \forall i \in V && (57)\\
& \sum_{j\in\delta^-(i)} x_{ji} \geq y_i, \forall i \in V && (58)\\
& \sum_{j\in\delta^+(i)} x_{ij} = \sum_{j\in\delta^-(i)} x_{ji}, \forall i \in V && (59)\\
& \sum_{j\in\delta^+(i)} x_{ij} \leq |N(i)| y_i, \forall i \in V && (60)\\
& \sum_{j\in\delta^-(i)} x_{ji} \leq |N(i)| y_i, \forall i \in V && (61)\\
& y_0 = 1 && (62)\\
& \sum_{i\in\bar{S}} \sum_{j\in S} x_{ij} \geq y_s, \forall S \subseteq V \setminus \{0\}, s \in S && (63)\\
& x_{ij} \in \{0,1\}, \forall (i,j) \in A && (64)\\
& y_i \in \{0,1\}, \forall i \in V && (65)
\end{aligned}
$$

Proposed Algorithms

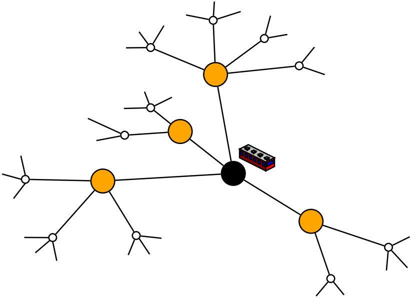
Local view

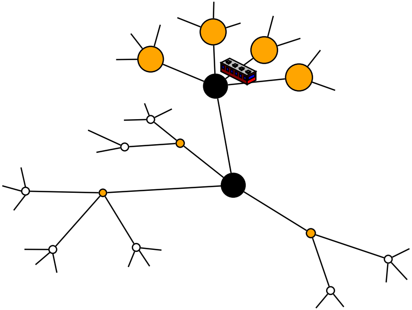
Two proposed algorithms:

- AlgNUM: where the mule decides his path based on the the number of uncovered nodes by the neighbours of the current sensor node.
- ALgCH: where the mule decision is based on the computation of convex-hulls of the current sensor node.

Proposed Algorithms

AlgNum



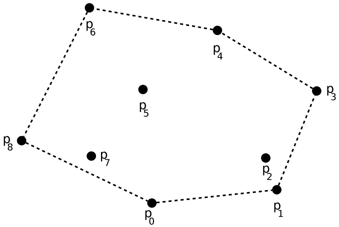


Proposed Algorihtms

AlgCH – Background

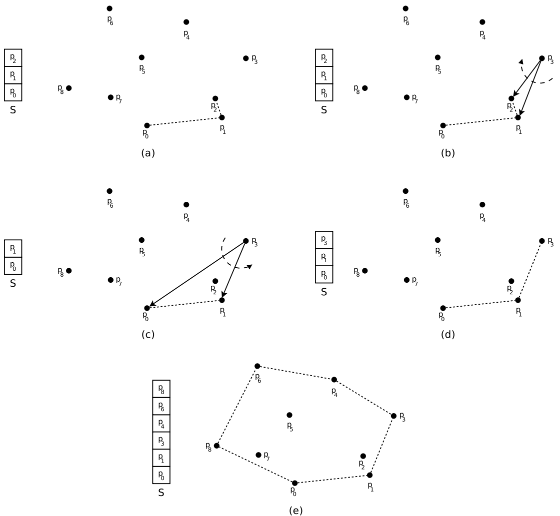
Definition of convex hull:

- Called $CH(Q)$
- All point in Q are unique
- At least three no co-linear points
- Graham's scan algorithm



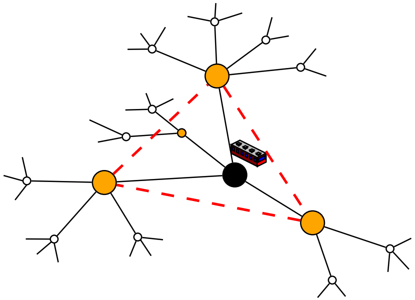
Proposed Algorithms

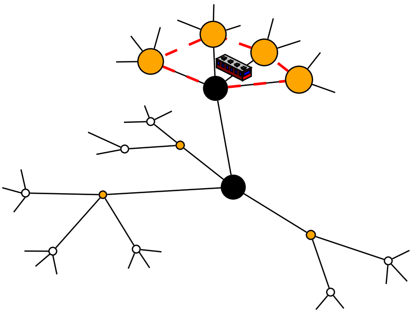
AlgCH – Background



Proposed Algorithms

AlgCH





Proposed Heuristics

Complexity Analysis

Local Time and Memory Complexities

Complexity	**Mule**		**Regular Sensor** u
	AlgNUM	**AlgCH**	
Memory	$\mathcal{O}(n)$ bytes	$\mathcal{O}(n)$ bytes	$\mathcal{O}(n)$ bytes
Time	$\mathcal{O}(n)$	$\mathcal{O}(n \lg n)$	$\mathcal{O}(1)$

Computer experiments and Analysis

Instances

Instances generated for the Close-enough Traveling Salesman Problem (Mennell, 2009)

- Euclidean plan
- Same acting range (communication range)
- Connected graph

Number of nodes: from 100 to 1000.

Computer experiments and Analysis

Computational Environment

Tests executed in a Intel Core i7 3.6 Ghz computer, with 16 GB of RAM and Linux Mint 18.

- LB_3: C++ and used a graph library, LEMON[1]
- Mathematical Formulation: C++ and used IBM ILOG CPLEX Optimizer v12.5.1
- Proposed algorithms: C++ and MPI

[1] LEMON – Library for Efficient Modeling and Optimization in Networks, available on https://lemon.cs.elte.hu

Computer experiments and Analysis

Theoretical Remarks

Inst.	LB_3		**Mathematical Formulation**			
	Sol_{LB_3}	**T(s)**	**LR**	$T_{LR}(s)$	Sol_{Math}	**T(s)**
kro100	4	0.01	3	0.01	4	1.31
rat195	4	0.07	3	0.02	4	4.27
team2_200	4	0.06	4	0.03	5	14.35
team3_300	32	0.09	19	0.07	74^{α}	17996.7
lin318	4	0.18	3.67	0.27	5^{α}	6012.51
rd400	6	0.28	5	0.97	6	7336.95
pcb442	6	0.38	4.14	0.88	6	37180.6
team6_500	3	0.66	3	2.20	3	225.29
dsj1000	6	3.00	4	2.50	8^{α}	24842.1
bonus1000	8	2.12	7.86	38.46	$22^{\alpha\beta}$	86400

$^{\alpha}$ the mathematical formulation used as input the best solution found by the four heuristics

$^{\beta}$ the optimal solution was not found in a time limit of 24 hours

Computer experiments and Analysis

Theoretical Remarks

Inst.	LB_3		**Mathematical Formulation**			
	Sol_{LB_3}	**T(s)**	**LR**	$T_{LR}(s)$	Sol_{Math}	**T(s)**
kro100	**4**	0.01	3	0.01	4	1.31
rat195	**4**	0.07	3	0.02	4	4.27
team2_200	4	0.06	4	0.03	5	14.35
team3_300	32	0.09	19	0.07	74^{α}	17996.7
lin318	4	0.18	3.67	0.27	5^{α}	6012.51
rd400	**6**	0.28	5	0.97	6	7336.95
pcb442	**6**	0.38	4.14	0.88	6	37180.6
team6_500	**3**	0.66	3	2.20	3	225.29
dsj1000	6	3.00	4	2.50	8^{α}	24842.1
bonus1000	8	2.12	7.86	38.46	$22^{\alpha\beta}$	86400

$^{\alpha}$ the mathematical formulation used as input the best solution found by the four heuristics

$^{\beta}$ the optimal solution was not found in a time limit of 24 hours

Computer experiments and Analysis

Scenarios

We implemented the algorithms in two scenarios:

- **With ACK**: the mule waits for neighborhood update
- **Without ACK**: The mule does not wait for neighborhood update

Computer experiments and Analysis

Computer experiments

Table: Computational results – locality sensitive heuristics

Inst.	Case 1 – With ACK				Case 2 – Without ACK			
	Sol_{NUM}	msgs	Sol_{CH}	msgs	Sol_{NUM}	msgs	Sol_{CH}	msgs
kro100	6	11108	6	10538	10.0	6664.7	15.3	5852.4
rat195	4	43006	4	42108	6.4	23040.8	10.4	21795.9
team2_200	10	30562	12	29016	14.6	17515.6	18.4	15212.2
team3_300	74	21742	74	19556	94.2	13586.5	88.6	10876.1
lin318	8	94250	10	91874	13.2	51019.1	13.4	46772.0
rd400	14	112694	14	108482	22.8	62300.5	17.2	54981.5
pcb442	12	169950	12	165778	17.4	90869.9	17.8	85027.3
team6_500	6	312252	6	307818	12.0	164828.5	9.8	154971.2
dsj1000	8	837514	10	828926	17.8	431716.1	19.9	418260.2
bonus1000	22	450712	26	441438	36.4	238016.4	35.4	222859.8

Computer experiments and Analysis

Computer experiments

Table: Computational results – locality sensitive heuristics

Inst.	**Case 1 – With ACK**				**Case 2 – Without ACK**			
	Sol_{NUM}	**msgs**	Sol_{CH}	**msgs**	Sol_{NUM}	**msgs**	Sol_{CH}	**msgs**
kro100	6	11108	6	10538	10.0	6664.7	15.3	5852.4
rat195	4	43006	4	42108	6.4	23040.8	10.4	21795.9
team2_200	10	30562	12	29016	14.6	17515.6	18.4	15212.2
team3_300	74	21742	74	19556	94.2	13586.5	88.6	10876.1
lin318	8	94250	10	91874	13.2	51019.1	13.4	46772.0
rd400	14	112694	14	108482	22.8	62300.5	17.2	54981.5
pcb442	12	169950	12	165778	17.4	90869.9	17.8	85027.3
team6_500	6	312252	6	307818	12.0	164828.5	9.8	154971.2
dsj1000	8	837514	10	828926	17.8	431716.1	19.9	418260.2
bonus1000	22	450712	26	441438	36.4	238016.4	35.4	222859.8

Computer experiments and Analysis

Computer experiments

Table: Maximum data mule local time

Inst.	Case 1 – With ACK		Case 2 – Without ACK	
	AlgNUM	**AlgCH**	**AlgNUM**	**AlgCH**
kro100	0.13	0.20	0.22	0.07
rat195	1.15	1.43	0.53	0.45
team2_201	0.36	0.73	0.86	0.85
team3_301	0.20	0.33	0.38	0.37
lin318	0.75	2.21	1.73	1.12
rd400	2.18	3.09	1.25	1.56
pcb442	1.72	2.27	1.73	1.60
team6_501	3.84	4.25	3.31	2.47
dsj1000	789.72	1213.97	290.25	242.11
bonus1001	9.13	7.91	6.30	3.62

Conclusions

Data Mule Scheduling Problem

- Three problems: Path Selection, Speed Control and Job Scheduling tackled simultaneously
- New set of instances created
- Mathematical formulations prove all optimal solution until 16 nodes
- Two robust heuristic proposed: GRVND and GVNS-RVND
- GVNS-RVND is more efficient in solution quality and computational time

Conclusions

Data Mule Routing Problem

- Lower Bound (global view)
  - A good lower bound calculation was proposed
  - LB_3 is better than LR of mathematical formulation
  - Some exact solutions found with the LB
  - Parameters to analyze the quality of solutions given by the locality heuristics
- Locality sensitive heuristics
  - AlgNum and AlgCH obtain good solutions
  - AlgCH proposed with the idea to obtain better solutions
  - Two scenarios (ACK): both approaches can be helpful

Future works

- DMSP
  - Extend the heuristics for the DMSP with Discrete and Continuous speeds may be done
    - Good results with constant speeds
    - Better improvement is possible with respect to the mathematical formulation
- DMRP
  - New compatible instances can be tested in order to explore other network characteristics
  - Extend for version where speed change is considered
  - Real tests can be done in order to observe the behavior of these algorithms

Data Gathering in Sensor Networks with Data Mules: Global and Local Approaches

Pablo Luiz Araújo Munhoz[1,2]
Advisor: Lúcia Drummond [1]
Advisor: Philippe Michelon [2]

[1]Universidade Federal Fluminense
Instituto de Computação

[2]Université d'Avignon et des Pays de Vaucluse
Laboratoire Informatique d'Avignon

May, 04 de 2017