

# Greedy algorithms for scheduling periodic message

Maël Guiraud<sup>1,2</sup> and Yann Strozecki<sup>1</sup>

<sup>1</sup>David Laboratory, UVSQ

<sup>2</sup>Nokia Bell Labs France

**Abstract**—A recent trend in mobile networks is to centralize in distant data-centers processing units which were attached to antennas until now. The main challenge is to guarantee that the latency of the periodic messages sent from the antennas to their processing units and back, fulfills protocol time constraints. The problem is then to propose a sending scheme from the antennas to their processing units and back without contention and buffer.

We study a star shaped topology, where all contentions are on a single arc shared by all antennas. We present several greedy heuristic to solve PAZL. We study their experimental efficiency and we use them to prove that when the load of the network is less than 44%, there is always a solution to PAZL. We also prove that for random lengths of the arcs, most of the instances have a solution when the load is less than 45%.

## I. INTRODUCTION

## II. MODEL

Describe the general model, and objective to optimize: no buffering. It is a scheduling/packing problem, could call each element a task or just an element. Notion of depth.

+ the simplified version for the star with a picture + a set of number  $d_1, \dots, d_n$  and two values  $P, \tau$ . Notion of load. [?]

## III. BASIC ALGORITHMS

Notion of partial solution, how to extend it (and notation). All algorithms are greedy and most can work online. Once a route is placed, it is not changed, hence in a partial solution we are only interested in the set of positions used in the backward and forward period. If  $n$  tasks has been placed, we denote them by  $x_1, \dots, x_n$  and  $y_1, \dots, y_n$ .

### A. Depth one

Algorithm of increasing quality

First fit, analyzed naively  $n/4$ , then better through compactness  $n/3$ .

Meta interval, load of  $n/3$ .

First fit in order of the  $d_i \bmod \tau$   $n/2$ , as good as naive first fit for  $\tau = 1$  (all alg degenerate to this one for  $\tau = 1$ ).

### B. General graph

Use the coherent routing property. Algo first fit  $n/4$  for any  $\tau$  and  $P$ , and any DAG of depth  $k$ .

Copy the results of the previous article and propose heuristics to chose among several positions/candidates (compactness heuristic).

### C. Experimental results

Results better in practice: give the data Two heuristics to test:

- heuristic to build super compact assignment (among the compact assignments possible, chose the one which maximize the gain on the second bloc)
- heuristic to maximize the free position of the remaining elements

Quality of the results explained by the average analysis done later

## IV. WHY $\tau$ CAN BE ASSUMED TO BE ONE

Rank the  $d_i$  by value, and compute  $d_i + d_n \bmod \tau$ . We allow buffering, but the worst time should not increase ! Bufferize each route during  $\tau - (d_i + d_n \bmod \tau)$ . All routes have the same remainder  $\bmod \tau$ , can assume they are of size one. A bit more complex on the general graphs, proofs on the depth of the graph. Should take into account the length of the graph.

## V. ABOVE 1/2

There are easier cases, all routes are distinct or all routes are the same. There is a solution for load 1 in these two cases. With two distinct values, solution for load almost one. Insert a figure here. We give a method which always finds a solution for load  $1/2 + \epsilon$  as soon as a constant fraction of the routes are distinct.

First, if there are  $2\epsilon P$  routes with the same shift, then load can  $1/2 + \epsilon$  be easily attained.

To go above  $1/2$  of load, we use a two pass greedy algorithm. It gives a better bound than a similar one pass greedy algorithm but it also drastically simplifies the proof. We use the notion of potential of a route in a partial solution and we show how to produce partial solutions with good potential, which can then be extended into a complete solution.

**Definition 1.** The potential of a route of shift  $s$  in a partial solution (whether fixed or not in the partial solution), is the number of integers  $i \in [P]$  such that  $i$  is used in the forward

window and  $i + s \bmod P$  is used in the backward window. We denote by  $Pot(S)$  the sum of potentials of the routes in the partial solution  $S$ .

**Definition 2.** The potential of a position  $i$  in a partial solution is the number of routes of shift  $s$  such that  $i + s$  is used in the partial solution.

The potentials of the positions satisfy the following simple invariant.

**Lemma 1.** The sum of potentials of all positions of the forward window in a partial solution of size  $k$  is  $kn$ .

We then link  $Pot(S)$  to the potential of the positions in the forward window.

**Lemma 2.** The sum of potentials of all used positions in the forward window in a partial solution  $S$  is equal to  $Pot(S)$ .

We now describe the algorithm to solve our problem with load  $1/2 + \epsilon$ . The first pass build a partial solution of size  $P/2$  by choosing at each step the route and the position which increases  $Pot(S)$  the most. We partition the routes in three sets:  $R_1$  the  $\epsilon P$  routes of largest potential in  $S$ ,  $R_2$  the routes placed in  $S$  and not in  $S_1$  and  $R_3$  the free routes not in  $R_1$ .

We consider the partial solution  $S'$  obtained by freeing the routes of  $R_1$  in  $S$ . Then,  $S'$  is extended into a complete solution by using any greedy algorithm, such as first fit, using first the solutions of  $R_3$  and finally  $R_1$ .

**Lemma 3.** We assume that there are at least  $P/2$  distinct routes. Given a partial solution  $S$  of size  $k \leq P/4$ , such that  $Pot(S) = \frac{k^2 n}{2P}$ , there is a free route which can be placed at some position, adding  $kn/P + 1$  of value.

*Proof.* By Lemma 2, the sum of the potential of the used positions in  $S_k$  is  $Pot(S) = k^2 n / 2P$ . Since there are  $k$  used positions, their average value is  $kn/2P$ . By Lemma 1, the sum of potentials is  $kn$ , hence the average value of a position is  $kn/P$ . As a consequence of these two facts, there is a free position of value more than the average  $kn/P$ .

Since there are at least  $P/2$  distinct routes, and  $k$  are used, there are at least  $P/2 - k$  free distinct routes. Since  $k \leq P/4$ , there are more distinct routes available than fixed route, hence any free position can be occupied by at least one free route.

Hence we can always put a route on the free position of potential  $\frac{kn}{P}$ . Moreover, fixing a route add one to its own potential, which increases the potential of the partial solution by  $kn/P + 1$ .  $\square$

Encore pas bon, on doit s'arrêter à  $P/4$  et pas  $P/2$  pour garantir de toujours trouver une route qui couvre, ce qui donne un gain de seulement  $1/64$ .

**Theorem 4.** The two-pass algorithm always solves positively our problem with load  $1/2 + 1/32$ , when all routes have a

distinct shift.

*Proof.* Using Lemma 3 we can prove by induction on  $k$  that for the partial solution  $S_k$  with  $k$  routes produced in the first pass of the algorithm,  $Pot(S_k) \geq \frac{k^2 n}{2P}$ . The property for  $k = 1$  is clearly true.

We now assume the property true for  $S_k$ . We can always assume that  $Pot(S) = \frac{k^2 n}{2P}$ , otherwise we put the additional value on some arbitrary free position and solve this modified instance (améliorer l'explication).

The algorithm selects the route which increases the potential the most. By Lemma 3, we can improve the potential by  $kn/P + 1$ . Hence  $Pot(S_{k+1}) \geq kn/P + 1 + Pot(S_k)$ .  $Pot(S_{k+1}) \geq kn/P + 1 + k^2 n / 2P \geq (k+1)^2 n / 2P$ , which proves the induction.

At the end of the first pass, we have a potential of  $(P/2)^2 n / 2P$ , since  $k = P/2$ . We want that the best  $\epsilon P$  routes to be of potential  $2\epsilon P, 2\epsilon P + 2, \dots, 4\epsilon P$ . Sort all routes by decreasing potential and assume that the previous condition is not met, that is the  $i$  routes in order of potential is of potential less than  $4\epsilon P - 2i$ . The potential of a single route is bounded by  $P/2$  since each placed route contribute at most one to its potential. Therefore, the first  $i$  routes are of potential at most  $P/2$  and the following ones of potential at most  $4\epsilon P - 2i$ . Therefore the potential is less than  $iP/2 + (4\epsilon P - 2i)(n - i)$ . This function is decreasing for  $i \leq \epsilon P$ , hence the potential should be less than  $4\epsilon n$ .

We know that the potential is at least  $(P/2)^2 n / 2P$  and we want it to be larger than  $4\epsilon P n$  so that the condition on the routes of largest potential is met. Hence we must satisfy the following equation:

$$(P/2)^2 n / 2P \geq 4\epsilon P n$$

$$\epsilon \leq 1/32$$

With this choice of  $\epsilon$ , all routes in  $R_1$  have potential at least  $2\epsilon$  in  $S'$ . Hence when they are considered in the second pass, they always can be placed, which proves that the algorithm works.  $\square$

TODO: que peut-on dire quand il n'y a pas que des shifts distincts. Si on réapplique plusieurs fois la méthode, on obtient mieux, peut-être compliqué à calculer.

We can add the fact that all free routes should be of potential less than  $2\epsilon$ . Either they are in the group of largest potential, and it decreases the necessary potential of the other ones by  $P/2$  each. Or they are not but if they are of potential  $2\epsilon$ , we can use them in the solution and again decrease the potential. The equation we get is thus  $Pot(S) \geq 4\epsilon P^2 / 2 + 2\epsilon^2 P$  and solving it gives a slightly better bound.

## VI. ALGORITHMS FOR RANDOM INSTANCES

a)  $\tau = 1$ : We analyze the following process, called **Uniform Greedy** or UG. For each element in order, chose one admissible position uniformly at random. We analyze the probability that Uniform Greedy solve the problem, averaged over all possible instances. It turns out that this probability, for a fixed load strictly less than one goes to zero when  $m$  grows.

Définir l'ensemble des solutions de taille  $n$  parmi  $m$ .

**Theorem 5.** *Given an instance of size  $n$  uniformly at random UG produces a solution uniformly at random or fail.*

*Proof.* Regarder mes notes partielles pour compléter ça.  $\square$

Let us denote by  $P(m,n)$  the probability that UG fails at the  $n$ th steps assuming it has not failed before.

**Theorem 6.** *We have*

$$P(m,n) = \frac{\binom{n}{2n-m}}{\binom{m}{n}}.$$

*In particular,  $P(m,n) \leq f(\lambda)^m$ , where  $f(\lambda) < 1$ .*

*Proof.* Probability independent of the shift of the  $n$  element, can say it is 0. It is the probability that two sets of size  $n$  in  $[m]$  are of union  $[m]$ . It is the same as the probability that it contains a given set of size  $m - n$ . Could find an asymptotic online.  $\square$

Can we make the same argument for a deterministic algorithm? The not average version of the argument is the previous proof.

## VII. GREEDY AND DELAY

Tradeoff between waiting time and load. Can we prove  $0.5 + \epsilon$  load for  $f(\epsilon, n, \tau)$  waiting time ? No idea yet.

## VIII. NON GREEDY ALGORITHM

Greedy + swap one element if necessary. Can we guarantee a solution for a larger load ? Conjecture, yes for  $\lambda = 2/3$ . Seems hard for group theoretic reason (how to avoid subgroups of  $\mathbb{Z}/p\mathbb{Z}$  which are a problem)

## IX. LOWER BOUNDS

Example/family of examples for which some greedy alg fail. Example/family of examples with a given load such that there are no feasible solution.

## X. NP-HARDNESS

Are we able to prove NP-hardness ?