QUASIGREEDY NUMERICAL SEMIGROUPS

Hebert Pérez-Rosés

Joint work with Maria Bras-Amorós
and José M. Serradilla-Merinero

Departament d'Enginyeria Informàtica
i Matemàtiques
Universitat Rovira i Virgili

Contents

UNIVERSITAT ROVIRA i VIRGILI

The change-making problem

Given a set of coin denominations $S = \{s_1 = 1, s_2, \ldots, s_t\}$, with $s_1 < \ldots < s_t$, and a target amount k, the goal is to obtain k as a sum of coins, using as few coins as possible.

Mathematically, we are looking for a **payment vector** $(a_1, \ldots, a_t)$, such that

$$a_i \geq 0, \text{ for all } i = 1, \ldots, t$$
$$\sum_{i=1}^{t} a_i s_i = k,$$
$$\sum_{i=1}^{t} a_i \text{ is minimal.}$$

UNIVERSITAT ROVIRA i VIRGILI

Greedy algorithm

The **greedy algorithm** for making change proceeds by always choosing in the first place the coin of the largest possible denomination.

Algorithm 1: GREEDY PAYMENT METHOD

Input : The set of denominations $S = 1, s_2, \ldots, s_t$, with $1 < s_2 < \ldots < s_t$, and a quantity $k \geq 0$.

Output: Payment vector $(a_1, a_2, \ldots, a_t)$.

for $i := t$ **downto** 1 **do**
  $a_i := k$ div s_i;
  $k := k$ mod s_i;
end

Greedy algorithm (Cont)

Definition

For a given set of denominations $S = 1, s_2, \ldots, s_t$, the *greedy payment vector* is the payment vector $(a_1, a_2, \ldots, a_t)$ produced by Algorithm 1, and $\textsc{GreedyCost}_S(k) = \sum_{i=1}^{t} a_i$.

Bad news: The greedy payment vector is not necessarily optimal (i.e. $\textsc{GreedyCost}_S(k)$ is not always the minimum cost among all possible payment vectors)

But: But there do exist some sets of denominations S for which we can guarantee that the greedy payment vector is indeed optimal.

Example

Example 1:

- Let $S_1 = \{1, 4, 6\}$ and $S_2 = \{1, 2, 5\}$ be two sets of denominations
- $\textsc{GreedyCost}_{S_1}(8) = 3$ (not optimal)
- We can find a representation of 8 with two coins
- $\textsc{GreedyCost}_{S_2}(8) = 3$ (optimal)

UNIVERSITAT ROVIRA i VIRGILI

Greedy sets (definition)

Definition (Greedy sets)

If a set S of denominations *always* produces an optimal greedy payment vector for *any* given amount k, then S is called *orderly*, *canonical*, or *greedy*.

A set S consisting of **one** or **two** denominations is always greedy. For sets of cardinal 3 we have the following characterization:

Proposition (Adamaszek & Adamaszek, 2010)

The set $S = \{1, a, b\}$ (with $a < b$) is greedy if, and only if, $b - a$ belongs to the set

$$\mathfrak{D}(a) = \{a-1, a\} \cup \{2a-2, 2a-1, 2a\} \cup \ldots \{ma - m, \ldots ma\} \cup \ldots =$$
$$= \bigcup_{m=1}^{\infty} \bigcup_{s=0}^{m} \{ma - s\}$$

One-point theorem

The most powerful necessary and sufficient condition is given by the so-called **one-point theorem** (Theorem 2.1 in [Adamaszek & Adamaszek, 2010]):

Theorem

Suppose that $S = \{1, s_2, \ldots, s_t\}$ *is a greedy set of denominations, and* $s_{t+1} > s_t$. *Now let* $m = \left\lceil \frac{s_{t+1}}{s_t} \right\rceil$. *Then* $\hat{S} = \{1, s_2, \ldots, s_t, s_{t+1}\}$ *is greedy if, and only if,* $\textsc{GreedyCost}_S(ms_t - s_{t+1}) < m$.

Notice that

$$(m-1)s_t + 1 \leq s_{t+1} \leq ms_t,$$

by the definition of m

UNIVERSITAT ROVIRA i VIRGILI

Application to greedy routing

Greedy routing consists of always forwarding the message packet to the neighbour node that minimizes the distance to the target node, for some distance function defined on the nodes of the network.

- Makes sense in **geographically embedded networks**, and also in **circulant networks.**
- Does not always result in the shortest route to the target node, but in some networks it does.

UNIVERSITAT ROVIRA i VIRGILI

Circulant graphs and digraphs

Definition

A circulant graph (or digraph) $C(n; S)$ is a **Cayley graph** on the cyclic group $\mathbb{Z}_n$, with connection set $S = \{s_1, \ldots, s_t\}$.

I.e., every vertex i is connected by an arc to the vertices $i + s_1, i + s_2, \ldots, i + s_t$, where addition is performed modulo n.

Circulant graphs are **vertex-transitive**, so the problem of finding a route (optimal or not) from vertex i to vertex j, can be reduced to the problem of finding a route from vertex 0 to vertex k, where k is either $i - j$ or $j - i$.

UNIVERSITAT ROVIRA i VIRGILI

Example: Circulant graph on $\mathbb{Z}_8$

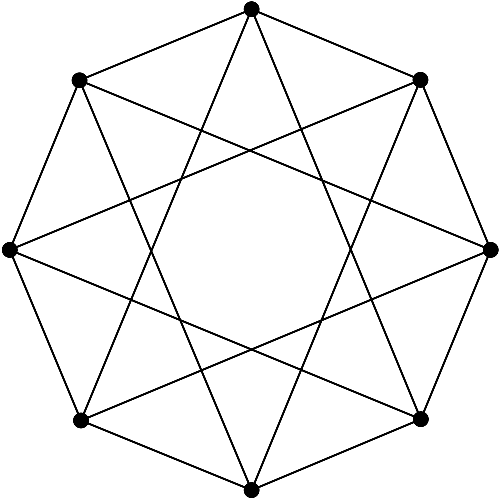

Figure: $C(8; \pm 1, \pm 3)$

1 The change-making problem and greedy sets

2 Generalization to numerical semigroups

UNIVERSITAT ROVIRA i VIRGILI

Quasi-greedy algorithm

Algorithm 2: QUASI-GREEDY REPRESENTATION METHOD

Input : The set of denominations $S = \{s_1, s_2, \dots, s_t\}$, with $1 \leq s_1 < s_2 < \dots < s_t$, $\gcd(s_1, s_2, \dots, s_t) = 1$, and an element $k \in \langle S \rangle$, $k > 0$.

Output: Quasi-greedy representation vector $\mathbf{a} = (a_1, a_2, \dots, a_t)$. (factorization)

for $i := t$ **downto** 1 **do**
  Let q be the largest integer such that $k = qs_i + r$ and $r \in \langle S \rangle$;
  $a_i := q$;
  $k := r$;
  if $k = 0$ **then**
    return $\boldsymbol{a}$;
  end
end

Quasi-greedy representation

Definition (Quasi-greedy representation and quasi-greedy cost)

For a given set of denominations $S = \{s_1, s_2, \ldots, s_t\}$, with $1 < s_2 < \ldots < s_t$ and $\gcd(s_1, s_2, \ldots, s_t) = 1$, and a given $k \in \langle S \rangle, k > 0$, the *quasi-greedy representation* of k with respect to S, denoted $\text{QG}\textsc{reedy}\text{R}\textsc{ep}_S(k)$, is the payment vector $\mathbf{a} = (a_1, a_2, \ldots, a_t)$ produced by Algorithm 2, and $\text{QG}\textsc{reedy}\text{C}\textsc{ost}_S(k) = \sum_{i=1}^{t} a_i$.

All representable numbers $k \in \langle S \rangle,\ k > 0$, have a quasi-greedy representation. In other words, Algorithm 2 always terminates and produces a factorization of k.

UNIVERSITAT ROVIRA i VIRGILI

Quasi-greedy set

Again, the quasi-greedy representation of k is not necessarily the best or the most efficient representation of k. However, for some specific sets S the quasi-greedy representation is indeed minimal for any representable k, which leads us to the following:

Definition (Quasi-greedy set)

Let $S = \{s_1, s_2, \dots, s_t\}$ be a set of generators with $1 < s_1 < s_2 < \dots < s_t$ and $\gcd(s_1, s_2, \dots, s_t) = 1$, such that Algorithm 2 *always* produces an optimal representation for *any* given $k \in \langle S \rangle$. Then S will be called *quasi-greedy*, and the semigroup $\mathbb{S} = \langle S \rangle$ will also be called *quasi-greedy*.

UNIVERSITAT ROVIRA i VIRGILI

Example

As before, sets of cardinality **two** are quasi-greedy, but that is not necessarily the case for sets of cardinality three or greater.

Example 2:

- Let $S_1 = \{3, 7, 10\}$ and $S_2 = \{3, 7, 11\}$ be two sets of 'denominations'
- $f(\mathbb{S}_1) = 11$ and $f(\mathbb{S}_2) = 8$
- $\textsc{QGreedyCost}_{S_1}(28) = 7$ (not optimal)
- We can find a representation of 28 with four 'coins'
- $\textsc{QGreedyCost}_{S_2}(28) = 4$ (optimal)
- Every integer larger than 8 has an optimal quasi-greedy representation in S_2 (we will see that later)

UNIVERSITAT ROVIRA i VIRGILI

Counterexamples and critical range

If S is **not** quasi-greedy, then there must exist some k such that $\textsc{MinCost}_S(k) < \textsc{QGreedyCost}_S(k)$. Such a number k is called a **counterexample.** The smallest counterexample must lie in some finite interval, the **critical range.**

Theorem

Let $S = \{s_1, s_2, \ldots, s_t\}$, with $1 < s_1 < s_2 < \cdots < s_t$ and $\gcd(s_1, \ldots, s_t) = 1$, so that $\mathbb{S} = \langle S \rangle$ is a numerical semigroup generated by S. If there exists a counterexample $k \in \mathbb{S}$ such that $\textsc{MinCost}_S(k) < \textsc{QGreedyCost}_S(k)$, then the smallest such k lies in the range

$$s_3 + s_1 + 2 \leq k \leq f(\mathbb{S}) + s_t + s_{t-1}. \tag{1}$$

UNIVERSITAT ROVIRA i VIRGILI

Example

Example 3:

- Let $S = \{5, 9, 14\}$
- $f(\mathbb{S}) = 31$
- The critical range is $[50; 54]$
- The number 54 is a counterexample. Let's check:
  - $\text{QGREEDYCOST}_S(54) = 9$
  - $\text{QGREEDYREP}_S(54) = 1 \cdot 14 + 8 \cdot 5$
  - $\text{MINCOST}_S(54) = 6$
  - $\text{MINREP}_S(54) = 6 \cdot 9$

UNIVERSITAT ROVIRA i VIRGILI

Witnesses

- Previous theorem is starting point for the algorithmic identification of quasi-greedy sets
- Look for a counterexample in the critical range, and if we cannot find one, then we can conclude that the given set S is quasi-greedy
- Implies calculating the minimal representation of all k in the critical range, which may be a costly process

Definition

Given a set of generators $S = \{s_1, s_2, \ldots, s_t\}$, with $1 < s_1 < s_2 < \ldots < s_t$ and $\gcd(s_1, s_2, \ldots, s_t) = 1$, a *witness* for S is any representable integer $k > 0$, such that $\textsc{QGreedyCost}_S(k) > \textsc{QGreedyCost}_S(k - s_i) + 1$ for some generator $s_i < k$.

Witnesses

Lemma

As in Definition 8, let $S = \{s_1, \ldots, s_t\}$ be a set of generators with $1 < s_2 < \ldots < s_t$ and $\gcd(s_1, \ldots, s_t) = 1$. Then

- *Every witness for S is a counterexample.*
- *The smallest counterexample (if it exists) is also a witness.*

Theorem

As in Definition 8 let $S = \{s_1, \ldots, s_t\}$ be a set of generators with $1 < s_1 < s_2 < \ldots < s_t$ and $\gcd(s_1, \ldots, s_t) = 1$. Then, S is quasi-greedy if, and only if, S does not have any witness k in the interval

$$s_3 + s_1 + 2 \leq k \leq f(\mathbb{S}) + s_t + s_{t-1}.$$

UNIVERSITAT ROVIRA i VIRGILI

Deciding if a set is quasi-greedy

Algorithm 3: DETERMINE WHETHER A SEMIGROUP DEFINED BY SET OF GENERATORS IS QUASI-GREEDY

Input : The set of denominations $S = \{s_1, s_2, \ldots, s_t\}$, with $1 < s_1 < s_2 < \ldots < s_t$, $\gcd(s_1, s_2, \ldots, s_t) = 1$.
Output: TRUE if $\langle S \rangle$ is quasi-greedy, and FALSE otherwise.

```
𝕊 := ⟨S⟩;
for k := s_3 + s_1 + 2 to f(𝕊) + s_t + s_{t-1} do
    t' := Smallest j such that k < s_j;
    for i := 1 to t' do
        if QGreedyCost_S(k) > QGreedyCost_S(k − s_i) + 1 then
            return FALSE;
        end
    end
end
return TRUE ;
```

More examples

Example 4:

- Let's go back to $S = \{5, 9, 14\}$
- Recall that $f(\mathbb{S}) = 31$ and the critical range is $[50; 54]$
- Let us check that 54 is a witness:
  - $\textsc{QGreedyCost}_S(54) = 9$
  - $\textsc{QGreedyCost}_S(54 - 9) = \textsc{QGreedyCost}_S(45) = 5$

Example 5:

- Let's go back to $S = \{3, 7, 11\}$
- Recall that $f(\mathbb{S}) = 8$ and the critical range is $[16; 26]$
- There is no witness in the critical range, therefore, S is quasi-greedy

UNIVERSITAT ROVIRA i VIRGILI

Final remarks

- Quasi-greedy semigroups with three generators are relatively abundant: We have sampled 90 semigroups with three generators between 2 and 15, and out of these, only 25 of them were *not* quasi-greedy.
- We have found a **family** of quasi-greedy semigroups with **three** generators, namely the semigroups generated by three consecutive integers: n, $n+1$ and $n+2$.
- Other families of quasi-greedy semigroups with three generators need to be identified.
- We know nothing about **four** generators.

UNIVERSITAT ROVIRA i VIRGILI

Bibliographic references - Change making problem

1. Adamaszek, A. and M. Adamaszek: Combinatorics of the change-making problem. *European Journal of Combinatorics* **31**, 47–63 (2010).
2. Cowen, L.J., R. Cowen and A. Steinberg: Totally Greedy Coin Sets and Greedy Obstructions. *The Electronic Journal of Combinatorics* **15** (2008), R90.
3. Kozen, D. and S. Zaks: Optimal bounds for the change-making problem. *Theoretical Computer Science* 123, 377–388 (1994).
4. Shallit, J.: What This Country Needs is an 18¢ Piece. *The Mathematical Intelligencer* **25**(2), 20–23 (2003).

UNIVERSITAT ROVIRA i VIRGILI

Bibliographic references - Circulant graphs and digraphs

1. Pérez-Rosés, H., M. Bras and J.M. Serradilla-Merinero: Greedy routing in circulant networks. *Graphs and Combinatorics* **38** (2022). DOI: `https://doi.org/10.1007/s00373-022-02489-9`
2. Hwang, F.K.: A survey on multi-loop networks. *Theoretical Computer Science* **299**, 107–121 (2003).
3. Wong, C.K. and D. Coppersmith: A combinatorial problem related to multimodule memory organizations. *Journal of the ACM* **21**, 392–402 (1974).