SOME FAMILIES OF GREEDY NUMERICAL SEMIGROUPS

Hebert Pérez-Rosés
(Joint work with Arnau Messegé-Buisan)

Universitat de Lleida
Spain

Contents

The change-making problem

Given an ordered 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

$$a_i \geq 0, \text{ for all } i = 1, \ldots, t$$

$$\sum_{i=1}^{t} a_i s_i = k,$$

$$\sum_{i=1}^{t} a_i \text{ is minimal.}$$

Note: There is an infinite supply of coins of every denomination.

Complexity

- The change-making problem is related to other combinatorial optimization problems (e.g. the **knapsack problem**, the **postage stamp problem**, etc.)
- Given a set of denominations $S = \{1, s_2, \ldots, s_t\}$ and a target amount k, the optimal representation for k can be found via **dynamic programming** in time $\mathcal{O}(tk)$.
- Time is exponential in the input size, if the numbers are represented in binary or decimal.
- In general, if the problem instance includes the set of denominations $S = \{1, s_2, \ldots, s_t\}$ and the target amount k, finding the best representation for k with respect to S is $\mathcal{NP}$-hard. (**Lueker, 1975**)

Greedy strategy

- We can use heuristics, such as the **greedy strategy**.
- The greedy algorithm for making change proceeds by always choosing in the first place the coin of the **largest possible denomination** and then applying the same procedure to the remainder.
- The greedy strategy is **not always optimal**. For example, consider a system with denominations 1, 4, 6, and suppose we want to make 8 cents:

(a) Greedy strategy

(b) Best strategy

Generalization of the change-making problem

- The greedy strategy for solving the change-making problem has been widely studied in its original formulation (e.g. **Adamaszek & Adamaszek, 2010**).
- We have relaxed the problem so that the system of denominations $S = \{s_1, s_2, \ldots, s_t\}$ does not necessarily include 1.
- In this case, **not all** integers are representable. However, if $\gcd(s_1, \ldots, s_t) = 1$ then **almost all** integers are representable.
- If $\gcd(s_1, \ldots, s_t) = 1$ then S generates a **numerical semigroup** $\mathbb{S} = \langle S \rangle$. Recall that

$$\mathbb{S} = \langle S \rangle = \left\{ \alpha_1 s_1 + \cdots + \alpha_t s_t : \alpha_1, \ldots, \alpha_t \in \mathbb{N}_0 \right\}$$

Generalization of the change-making problem

$$\mathbb{S} = \langle S \rangle = \left\{ \alpha_1 s_1 + \cdots + \alpha_t s_t : \alpha_1, \ldots, \alpha_t \in \mathbb{N}_0 \right\}$$

- $\left| \mathbb{N}_0 \setminus \mathbb{S} \right|$ is **finite**, i.e. **almost all** positive integers are representable.
- There exists a number $F(\mathbb{S})$ (the **Frobenius number**), such that all integers $k > F(\mathbb{S})$ are representable.
- If t is variable, computing $F(\mathbb{S})$ is $\mathcal{NP}$-hard (**Ramírez-Alfonsín, 1996**).
- There exists a unique **minimal** set of generators $\widehat{S}$, in the sense that any proper subset of $\widehat{S}$ fails to generate $\mathbb{S}$. The cardinality of $\widehat{S}$ is the **embedding dimension** of $\mathbb{S}$.

Greedy algorithm for the CMP in semigroups

Algorithm 1: GREEDY REPRESENTATION METHOD

```
Input  : Set of denominations $S = \{s_1, s_2, \ldots, s_t\}$, with
         $1 \leq s_1 < s_2 < \ldots < s_t$, $\gcd(s_1, s_2, \ldots, s_t) = 1$, and $k \in \langle S \rangle$.
Output: Greedy representation vector $\mathbf{a} = (a_1, a_2, \ldots, a_t)$.
if $k \notin \langle S \rangle$ then
    return "k is not representable";
end
Initialize a: $\mathbf{a} \leftarrow (0, 0, \ldots, 0)$;
$i \leftarrow t$;
while $k > 0$ do
    Let $q$ be the largest integer such that $k = qs_i + r$ and $r \in \langle S \rangle$;
    $a_i \leftarrow q$;
    $k \leftarrow r$;
    $i \leftarrow i - 1$;
end
return $a$;
```

Greedy algorithm (Cont)

Definition

For a given set of denominations $S = \{s_1, s_2, \ldots, s_t\}$, the **greedy payment vector** is the payment vector $(a_1, a_2, \ldots, a_t)$ produced by Algorithm 1 above, and it is denoted $\textsc{GreedyRep}_S(k)$. Additionally, $\textsc{GreedyCost}_S(k) = \sum_{i=1}^{t} a_i$.

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

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

Greedy sets (definition)

Definition (Greedy sets)

If the greedy algorithm (Algorithm 1 above) always produces an optimal payment vector for *any* given representable amount k, with respect to a given set $S = \{s_1, \ldots, s_t\}$ of denominations, then S is called **orderly**, **canonical**, or **greedy**.

If S is greedy and $\gcd(s_1, \ldots, s_t) = 1$, then the corresponding **numerical semigroup** $\mathbb{S} = \langle S \rangle$ is also called *greedy*.

1. A set S consisting of one or two denominations is always greedy.
2. Sets of cardinality ≥ 3 are investigated in a recent paper (**Pérez-Rosés, Serradilla-Merinero & Bras-Amorós, 2025**), where an algorithm is given to decide whether a given set S of denominations is greedy.

Algorithm for recognizing greedy sets

The algorithm works by looking for **counterexamples** within a certain **critical range**.

Definition (Counterexample)

Given a set $S = \{s_1, \ldots, s_t\}$ of denominations, with $\gcd(s_1, \ldots, s_t) = 1$, and a representable amount k, then k is a **counterexample** for S if

$$\textsc{GreedyCost}_S(k) > \textsc{MinCost}_S(k)$$

Theorem (PR, SM & BA, 2025)

S is greedy if, and only if, S does not have any counterexample k in the interval

$$\left[s_3 + s_1 + 2;\ F(\mathbb{S}) + s_t + s_{t-1}\right] \tag{1}$$

*(this interval is the **critical range** mentioned above)*

Complexity issues

- The running time of the above algorithm is $\mathcal{O}(ts_t)$
- Exponential in the number of digits
- In the **original setting**, i.e. when the system of denominations is $S = \{1, s_2, \ldots, s_t\}$, there exists a **polynomial-time algorithm** to decide if S is greedy (**Pearson, 2005**).
- In our **generalized setting**, $S = \{s_1 \geq 1, s_2, \ldots, s_t\}$, we still don't know if the problem is solvable in polynomial time.

Greedy sets of cardinality three

Lemma (PR, SM & BA, 2025)

Let $S = \{a, b, c\}$*, with* $1 < a < b < c$*, and* $\gcd(a, b, c) = 1$*, so that* $\mathbb{S} = \langle S \rangle$ *is not greedy. Then, the smallest counterexample* k *has the form* $k = by$*, and it is a solution of the Diophantine equation*

$$ax + cz = by, \tag{2}$$

where x, y, z *are* *<u>positive</u>* *integers, such that* $y < x + z$*.*

Theorem (PR, SM & BA, 2025)

The semigroups $\mathbb{S} = \langle n, n+1, n+2 \rangle$*, with* $n \geq 1$*, are greedy.*

New families of embedding dimension three

Theorem

If $S = \{a, b, c\}$ *is a set of positive integers with* $a < b < c$ *such that* $\gcd(a, b) = 1$ *and* $c \geq \frac{b^2}{a}$*, then* S *is greedy.*

Theorem

Let $S = \{a, b, c\}$*, with* $3 \leq a < b < c$*,* $\gcd(a, b) = 1$*, and* $c = F(a, b)$*, i.e.* $c = ab - a - b$*. Then* $\langle S \rangle$ *is greedy.*

Note: The case $b \leq \frac{1}{2}\left(a^2 - a + \sqrt{a^4 - 2a^3 - 3a^2}\right)$ is covered by the first theorem above.

Proof methodology in dimension three

- Emulate the algorithm.
- Find all the multiples of b that lie inside the critical range.
- For each such multiple by, determine whether it is a counterexample or not.
- I.e. determine whether by can also be represented as $ax + cz$, such that $y < x + z$.
- This methodology is difficult to generalize to higher dimensions.

Totally-greedy sets

If we have a greedy set S, a straightforward question is to determine which **subsets** of S are also greedy.

Definition

A set $S = \{s_1, s_2, \dots, s_t\}$ is **totally greedy**[a] if every prefix subset $\{s_1, s_2, \dots, s_k\}$, with $k \leq t$ is greedy.

---

[a] Also called normal, or totally orderly.

(Cowen, Cowen & Steinberg, 2008)

Obviously, a totally greedy set is also greedy, but the converse is not true in general. Take, for instance, the greedy set $\{1, 2, 5, 6, 10\}$, whose prefix subset $\{1, 2, 5, 6\}$ is not greedy.

Totally-greedy sets (cont)

Most coin systems used in the world are greedy (totally greedy)

(c) American coins

(d) Euro coins

Totally-greedy sets (cont)

One exception was the coin system of India in the 1960s and 1970s, which consisted of coins with values 1, 2, 3, 5, 10, 20, 25 and 50 paise.

(e) obverse (f) reverse

Figure: A rare coin: the 3-paisa coin

(**Counterexample**: $40 = 25 + 10 + 5 = 20 + 20$)

Totally-greedy sequences

The definition of totally greedy sets can be extended to **infinite sequences** in a straightforward way:

Definition

Let $S = \{s_n\}_{n=1}^{\infty}$ be a monotonically increasing integer sequence, with $s_1 \geq 1$. We say that S is **totally greedy** (or simply, greedy) if every **prefix subset** $S^{(t)} = \left\{s_1, s_2, \ldots, s_t\right\}$ is greedy.

Problem: Identify totally greedy sequences.

Totally-greedy sequences

- We have identified several classes of totally greedy sequences starting with $s_1 = 1$ (e.g. **Pérez-Rosés, Bras-Amorós and Serradilla-Merinero, 2022** and **Pérez-Rosés, 2025**)

- **Example**: Let $\{G_n\}_{n=1}^{\infty}$ be defined by the second-order linear recurrence

$$G_n = \begin{cases} 1 \text{ if } n = 1, \\ a \text{ if } n = 2, \\ pG_{n-1} + qG_{n-2}, \text{ if } n > 2, \end{cases} \tag{3}$$

where a, p, q are positive integers, with $p \geq q$ and $a > 1$.

Totally greedy sequences

Theorem (Pérez-Rosés, 2025)

Let $\{G_n\}_{n=1}^{\infty}$ be a type-1-sequence with $2 \leq a \leq p+q$. Then $\{G_n\}_{n=1}^{\infty}$ is totally greedy.

The following sequences are special cases of the above:

- $\{F_n\}_{n=1}^{\infty} = \{1, 2, 3, 5, 8, 13, \ldots\}$ (shifted Fibonacci numbers)
- $\{P_n\}_{n=1}^{\infty} = \{1, 2, 5, 12, 29, 70 \ldots\}$ (shifted Pell numbers)
- $P_n = 2P_{n-1} + P_{n-2}$, where $P_0 = 0$ and $P_1 = 1$.

Corollary

$\{F_n\}_{n=1}^{\infty}$ and $\{P_n\}_{n=1}^{\infty}$ are totally greedy.

A sequence which is not totally greedy

If $q > p$ we can no longer guarantee that $\{G_n\}_{n=1}^{\infty}$ is totally greedy.

Take for instance the (shifted) **Jacobstahl** numbers:

$\{\mathcal{J}_n\}_{n=2}^{\infty} = \{1, 3, 5, 11, 21, 43, 85 \ldots\}$, defined by $\mathcal{J}_0 = 0$, $\mathcal{J}_1 = 1$, and $\mathcal{J}_n = \mathcal{J}_{n-1} + 2\mathcal{J}_{n-2}$. Indeed, $\left\lceil \frac{\mathcal{J}_7}{\mathcal{J}_6} \right\rceil = 3 = m$, but $3 \cdot 21 - 43 = 20$, and $\textsc{GreedyCost}_{\mathcal{J}^{(6)}}(20) = 4 > m$.

Hence, $\mathcal{J}^{(7)} = \{1, 3, 5, 11, 21, 43\}$ is not greedy, and $\{\mathcal{J}_n\}_{n=2}^{\infty}$ is not totally greedy.

Totally-greedy sequences

- Now we want to identify totally greedy sequences having $s_1 > 1$.
- In particular, suppose we have a totally greedy sequence $S = \{s_n\}_{n=1}^{\infty}$ (possibly starting with $s_1 = 1$), and let us denote by ${}_rS$ the **suffix** sequence

  $$ {}_rS = \{s_r, s_{r+1}, \ldots\} = \{s_n\}_{n=r}^{\infty} $$

  (i.e. starting at the r-th term – Note that we have kept the original subscripts in order to avoid confusions).
- **Question**: For which values of r does ${}_rS$ remain totally greedy?

Totally-greedy sequences

- Let $\mathcal{F} = \{F_n\}_{n=0}^{\infty} = \{1, 1, 2, 3, 5, 8, 13, \ldots\}$ denote the Fibonacci sequence. In **PR, BA & SM, 2022** it was proved that ${}_1\mathcal{F} = \{1, 2, 3, 5, 8, 13, \ldots\}$ is **totally greedy**.
- Actually, the greediness of the (full) Fibonacci sequence has been known for a long time under a different guise (**Zeckendorf representation**).

Figure: Édouard Zeckendorf

Zeckendorf representation

Any non-negative integer k has a representation as a sum of Fibonacci numbers, such that

1. It does not contain two consecutive Fibonacci numbers, F_j and F_{j+1}, and
2. It does not contain two occurrences of a Fibonacci number F_j.

For the truncated Fibonacci sequence ${}_1\mathcal{F}^{(t)} = \{1, 2, 3, 5, \ldots F_t\}$, the largest digit, F_t, may appear more than once.

Example

The Zeckendorf representation of 17 is $13 + 3 + 1$. However, if we take the truncated sequence ${}_1\mathcal{F}^{(5)} = \{1, 2, 3, 5, 8\}$, the Zeckendorf-type representation of 17 becomes $8 + 8 + 1$.

Second suffix of Fibonacci sequence

Let us now consider the suffix sequence ${}_2\mathcal{F} = \{2, 3, 5, 8, 13, 21, \ldots\}$. We have proved the following result:

Theorem

The sequence ${}_2\mathcal{F}$ is totally greedy.

In other words, if we truncate ${}_2\mathcal{F}$ at any $t \geq 2$, then the resulting set ${}_2\mathcal{F}^{(t)}$ is greedy.

Note that if $t > 2$, then the Frobenius number $F(2, 3, \ldots, F_t) = 1$. Hence all positive integers, except 1, are **representable** with respect to the set ${}_2\mathcal{F}^{(t)}$.

Zeckendorf-type representation

We can obtain the **Zeckendorf-type representation** of a positive integer $k \geq 2$ with respect to ${}_2\mathcal{F}^{(t)}$ via the greedy algorithm. This Zeckendorf-type representation has the following properties:

1. It does not contain two consecutive Fibonacci numbers, F_j and F_{j+1}, with the possible exception of F_{t-1} and F_t, and
2. It does not contain two occurrences of a Fibonacci number F_j, with the possible exceptions of $F_2 = 2$ and $F_3 = 3$, which may appear twice (but not at the same time), and F_t, which may appear an arbitrary number of times.

Rewriting rules

We can also obtain the **Zeckendorf-type representation** of $k \geq 2$ from *any* Fibonacci-based representation $r(k)$ of k:

1. **Eliminate consecutive Fibonacci summands**: If $r(k)$ contains a pair of consecutive Fibonacci summands, F_j and F_{j+1}, for any $2 \leq j \leq t-2$, then replace the pair by F_{j+2}.
2. **Eliminate repetitions of a Fibonacci number**: If $r(k)$ contains two (or more) occurrences of a Fibonacci number F_j, for any $4 \leq j \leq t-1$, then replace $2F_j$ by $F_{j+1} + F_{j-2}$.
3. **Eliminate trio of** 2**'s**: If $r(k)$ contains three (or more) occurrences of 2, then replace a trio of 2's by two occurrences of 3.
4. **Eliminate trio of** 3**'s**: If $r(k)$ contains three (or more) occurrences of 3, then replace a trio of 3's by the trio $5+2+2$.

Further suffixes of the Fibonacci sequence

Unfortunately, the previous results cannot be generalized in a straightforward manner to **arbitrary suffixes** of the Fibonacci sequence starting at F_r. Take for instance $r = 3$, and take the truncated set $S = \{3, 5, 8\}$. It is easy to verify that $k = 20$ is a **counterexample** for S, since

$$\textsc{GreedyCost}_S(20) = 5,\ \textsc{GreedyRep}_S(20) = 8 + 3 + 3 + 3 + 3,$$

whereas

$$\textsc{MinCost}_S(20) = 4,\ \textsc{MinRep}_S(20) = 5 + 5 + 5 + 5.$$

Consequently, S is not greedy, and the suffix sequence

$$\{{}_3F_n\}_{n=0}^{\infty} = \{3, 5, 8, 13, 21, \ldots\}$$

is not totally greedy.

Bibliographic references - I

1. Pérez-Rosés, H. M. Bras and J.M. Serradilla-Merinero: Greedy routing in circulant networks. *Graphs and Combinatorics* **38**(86) (2022). `https://doi.org/10.1007/s00373-022-02489-9`.
2. Pérez-Rosés, H.: Totally Greedy Sequences Defined by Second-Order Linear Recurrences With Constant Coefficients. *Procs. ICCSA 2025*. LNCS vol. 15648, pp. 122–139. DOI: `https://doi.org/10.1007/978-3-031-97000-9_8`.
3. Pérez-Rosés, H. M. Bras and J.M. Serradilla-Merinero: Greedy sets and greedy numerical semigroups. *Communications in Algebra* **53**, 6 (2025), pp. 2442–2455. DOI: `https://doi.org/10.1080/00927872.2024.2444625`.

Bibliographic references - II

1. Adamaszek, A. and M. Adamaszek: Combinatorics of the change-making problem. *European Journal of Combinatorics* **31**, 47–63 (2010). DOI: `https://doi.org/10.1016/j.ejc.2009.05.002`
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. Ramírez-Alfonsín, J.L.: Complexity of the Frobenius problem. *Combinatorica* **16**, 143–147 (1996).
4. Pearson, D.: A polynomial-time algorithm for the change-making problem. *Operations Research Letters* **33**(3) 231–234 (2005). DOI: `https://doi.org/10.1016/j.orl.2004.06.001`

Network routing

Network routing is the process by which a path is chosen in order to transmit a message across a network.

- The path is chosen according to different criteria, e.g. the distance to be travelled by the message, fault-tolerance of the communication protocol, network congestion, etc.
- If the network topology is fixed, then a **static routing algorithm** can be implemented.

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

Circulant graphs and digraphs

Definition

A **circulant digraph** $C(n; S)$ is a Cayley graph on the cyclic group $\mathbb{Z}_n$, with connection set $S = \{s_1, \ldots, s_t\}$, such that S does not include any pair consisting of a generator s_i and its inverse.

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.

Definition

An undirected circulant graph $C(n; S)$ is a Cayley graph on the cyclic group $\mathbb{Z}_n$, with a symmetric connection set S (i.e. $S = S^{-1}$, or $S = -S$ in additive notation).

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

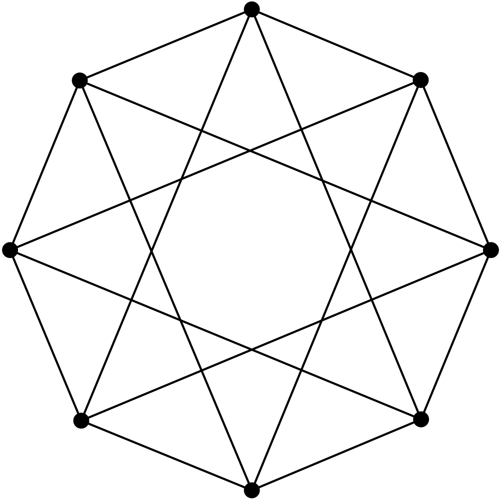

Figure: $C(8; \pm 1, \pm 3)$

Circulant graphs and supercomputers

Some supercomputer architectures based on (generalizations of) circulant graphs

- Cray 3TE
- IBM Blue Gene / L
- IBM Blue Gene / P

Figure: IBM Blue Gene / P

Change-making and greedy routing in circulant digraphs

Let $\overrightarrow{\Gamma}$ be a circulant digraph $C(n; 1, s_2, \ldots, s_t)$, where $S = \{1, s_2, \ldots, s_t\}$ is a greedy system of denominations, and let $k \in \{0, \ldots, n-1\}$ be an arbitrary vertex of $\overrightarrow{\Gamma}$. Then, the greedy payment vector $(a_1, a_2, \ldots, a_t)$ gives us the best route from vertex 0 to vertex k, by following the s_t-arcs a_t times, the s_{t-1}-arcs a_{t-1} times, and so on.

Definition

A circulant digraph $C(n; 1, s_2, \ldots, s_t)$ equipped with a greedy connection set will be called *greedy*, and the greedy payment vector will be the **greedy routing vector**.

Sending a message from 0 to 8 in $C(11; 1, 2, 5)$

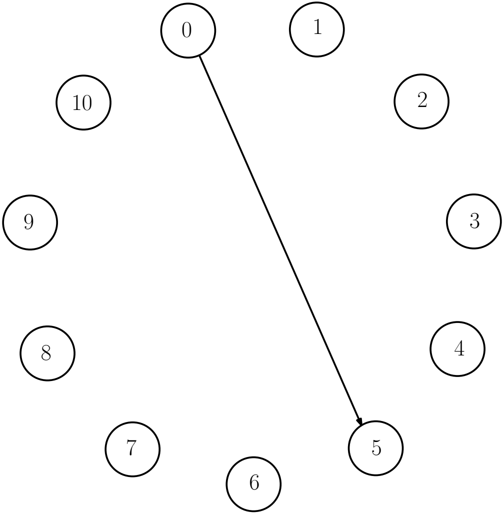

Figure: Step of length 5 in $C(11; 1, 2, 5)$

Sending a message from 0 to 8 in $C(11; 1, 2, 5)$

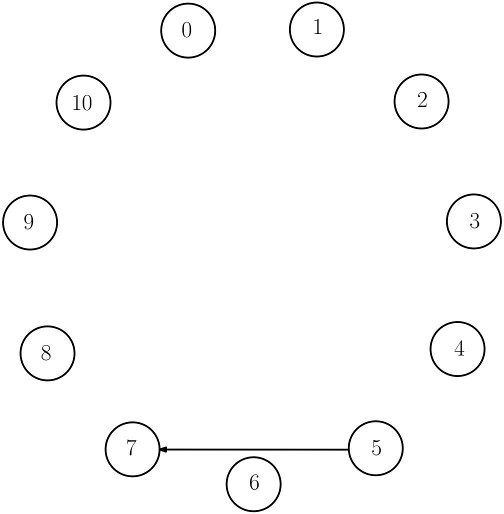

Figure: Step of length 2 in $C(11; 1, 2, 5)$

Sending a message from 0 to 8 in $C(11; 1, 2, 5)$

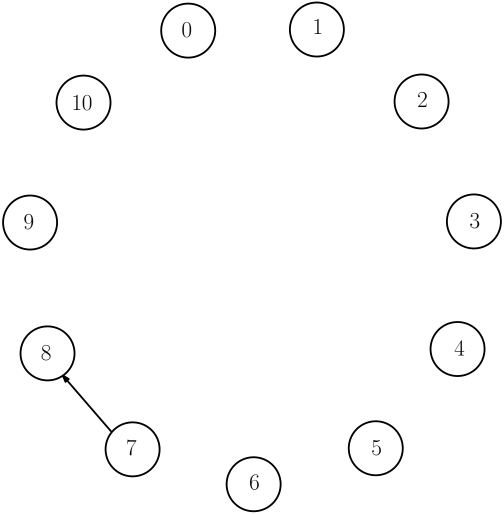

Figure: Step of length 1 in $C(11; 1, 2, 5)$

Further remarks on greedy routing

- If we know the routing vector in advance, we can follow the arcs in any order.
- However, the best feature of the greedy routing algorithm is that we don't need to compute the whole routing vector in advance; we can do it *online*.
- Little attention has been paid to the design of large circulant networks with efficient communication features.
- (**Pérez-Rosés, Bras-Amorós and Serradilla-Merinero, 2022**).