Quasi-Greedy Numerical Semigroups

Hebert Pérez-Rosés
José Miguel Serradilla-Merinero
Maria Bras-Amorós

Dept. of Computer Science and Mathematics
Universitat Rovira i Virgili, Avda. Paisos Catalans
26, Tarragona, 43007, Catalonia, Spain

Abstract

Motivated by the change-making problem, we extend the notion of greediness to sets of generators of numerical semigroups, and we provide an algorithm to decide that property.

1. The change-making problem and greedy sets

In the change-making problem we are given a set of integer coin denominations $S = \{s_1, s_2, \ldots, s_t\}$, with $0 < s_1 < \ldots < s_t$, and a target amount $k \in \mathbb{N}_0$, where $\mathbb{N}_0$ is the set of nonnegative integers. The goal is to represent k using as few coins as possible. Traditionally, an additional requirement is imposed, namely that $s_1 = 1$. In this work we explore the consequences of dropping that requirement, so that the ideas of the change-making problem can be generalized to numerical semigroups.

More formally, we are looking for a *payment vector* $(a_1, \ldots, a_t)$, such that: 1. $a_i \in \mathbb{N}_0$ for all $i = 1, \ldots, t$, 2. $\sum_{i=1}^{t} a_i s_i = k$, and 3. $\sum_{i=1}^{t} a_i$ is minimal.

The payment vector $\mathbf{a} = (a_1, a_2, \ldots, a_t)$ that satisfies Conditions 1, 2 and 3 above is called a *minimal payment vector*, or *minimal representation* of k with respect to S, and we denote it by $\textsc{MinRep}_S(k)$. If $\mathbf{a}$ is a minimal representation of k, then $\textsc{MinCost}_S(k) = \sum_{i=1}^{t} a_i$. Note that $\textsc{MinRep}_S(k)$ is not necessarily unique, but $\textsc{MinCost}_S(k)$ is.

A traditional approach for addressing the change-making problem is the *greedy strategy*, which proceeds by first choosing the coin of the largest possible denomination, subtracting it from the target amount as many times as possible, then doing the same with the second-largest coin, and so on.

If the set of coin denominations $S = \{s_1, s_2, \ldots, s_t\}$ is such that $1 < s_1 < \ldots < s_t$, then not all integers are representable, regardless of the strategy that we choose for finding the representation. Given a representable k, we call it *greedy-representable* if there exists a payment vector $\mathbf{a} = (a_1, a_2, \ldots, a_t)$ that is obtained by the greedy strategy mentioned above.

Definition 1. *Given a set of denominations $S = \{s_1, s_2, \ldots, s_t\}$ and a given target amount k, the* greedy payment vector*, or* greedy representation *of k with respect to S, is the payment vector $\boldsymbol{a} = \textsc{GreedyRep}_S(k) = (a_1, a_2, \ldots, a_t)$ produced by the greedy strategy (if it exists), and $\textsc{GreedyCost}_S(k) = \sum_{i=1}^{t} a_i$.*

If k is greedy-representable, the greedy representation is not necessarily the best or the most efficient representation of k. In other words, given the greedy representation vector $\mathbf{a} = (a_1, a_2, \ldots, a_t)$, the sum $\displaystyle\sum_{i=1}^{t} a_i$ is not necessarily minimal. However, for some specific sets S the greedy representation is indeed minimal for any greedy-representable k. Such a set S is called *greedy*, *orderly* or *canonical* [2]. Greedy sets were used in [2] to construct circulant network topologies with efficient routing algorithms.

2. Quasi-greedy sets and quasi-greedy numerical semigroups

If $\gcd(s_1, s_2, \ldots, s_t) = 1$ then S generates a numerical semigroup $\mathbb{S}$, and all integers $k > f(\mathbb{S})$ are representable, where $f(\mathbb{S})$ denotes the Frobenius number of $\mathbb{S}$. The denominations $s_1, s_2, \ldots, s_t$ are the *generators* of $\mathbb{S}$, and we write $\mathbb{S} = \langle S \rangle$. We also denote by G the set of gaps of $\mathbb{S}$. For the basic concepts and results about numerical semigroups see [3].

Even if k is representable, we cannot still guarantee that k be greedy-representable. This leads us to consider a modification of the greedy strategy for representing numbers, which we call *quasi-greedy*. The quasi-greedy algorithm behaves as the greedy algorithm whenever possible. It is described in Algorithm 1.

Algorithm 1: QUASI-GREEDY REPRESENTATION METHOD

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

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

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

Definition 2. *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* $Q\text{GREEDYREP}_S(k)$*, is the payment vector* $\mathbf{a} = (a_1, a_2, \ldots, a_t)$ *produced by Algorithm 1, and* $Q\text{GREEDYCOST}_S(k) = \sum_{i=1}^{t} a_i$.

All representable numbers $k \in \langle S \rangle$, $k > 0$, have a quasi-greedy representation. Moreover, if k is greedy-representable, then the quasi-greedy representation is just the greedy representation.

Again, the quasi-greedy representation is not necessarily the best or the most efficient representation of k, i.e. given the quasi-greedy representation vector $\mathbf{a} = (a_1, a_2, \ldots, a_t)$, the sum $\sum\limits_{i=1}^{t} a_i$ is not necessarily minimal. However, for some specific sets S the quasi-greedy representation is indeed minimal for any representable k, which leads us to the following definition:

Definition 3. *Let* $S = \{s_1, s_2, \ldots, s_t\}$ *be a set of generators with* $1 < s_1 < s_2 < \ldots < s_t$ *and* $\gcd(s_1, s_2, \ldots, s_t) = 1$*, such that Algorithm 1* 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.

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

3. Algorithmic identification of quasi-greedy sets

If S is not quasi-greedy, then there must exist some k s.t. $\textsc{MinCost}_S(k) < \textsc{QGreedyCost}_S(k)$. Such a k is called a *counterexample*. The smallest counterexample must lie in some finite interval, the *critical range*. This is the basis for the algorithmic identification of quasi-greedy sets. Our main result is a generalization of Theorem 2.2 of [1] to numerical semigroups:

Theorem 1. *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*

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

Theorem 1 is the starting point for the algorithmic identification of quasi-greedy sets and quasi-greedy numerical semigroups. The algorithm must simply look for a counterexample in the critical range, and if we cannot find one, then we can conclude that S is quasi-greedy, as well as $\mathbb{S} = \langle S \rangle$.

With the aid of Theorem 1 and the ensuing algorithm we have been able to find several quasi-greedy semigroups of embedding dimension three.

References

[1] Kozen, D. and S. Zaks: Optimal bounds for the change-making problem. *Theoretical Computer Science* 123, 377–388 (1994).

[2] Pérez-Rosés, H., M. Bras-Amorós and J.M. Serradilla-Merinero: "Greedy routing in circulant networks". *Graphs and Combinatorics* **38**, 86 (2022). DOI: `https://doi.org/10.1007/s00373-022-02489-9`

[3] Rosales, J.C. and P.A. García-Sánchez: *Numerical Semigroups*. Springer (2009).