Composing Equipotent Teams

Mark Cieliebak\textsuperscript{1}, Stephan Eidenbenz\textsuperscript{2}, and Aris Pagourtzis\textsuperscript{3}

\begin{itemize}
  \item \textsuperscript{1} Institute of Theoretical Computer Science, ETH Zürich
cieliebak@inf.ethz.ch
  \item \textsuperscript{2} Basic and Applied Simulation Science (CCS-5), Los Alamos National Laboratory\textsuperscript{†} eidenben@lanl.gov
  \item \textsuperscript{3} Department of Computer Science, School of ECE, National Technical University of Athens, Greece\textsuperscript{‡} pagour@cs.ntua.gr
\end{itemize}

Abstract. We study the computational complexity of \textsc{$k$ Equal Sum Subsets}, in which we need to find \textit{k} disjoint subsets of a given set of numbers such that the elements in each subset add up to the same sum. This problem is known to be \textsc{NP}-complete. We obtain several variations by considering different requirements as to how to compose teams of equal strength to play a tournament. We present:

- A pseudo-polynomial time algorithm for \textsc{$k$ Equal Sum Subsets} with $k = O(1)$ and a proof of strong \textsc{NP}-completeness for $k = \Omega(n)$.
- A polynomial-time algorithm under the additional requirement that the subsets should be of equal cardinality $c = O(1)$, and a pseudo-polynomial time algorithm for the variation where the common cardinality is part of the input or not specified at all, which we proof \textsc{NP}-complete.
- A pseudo-polynomial time algorithm for the variation where we look for two equal sum subsets such that certain pairs of numbers are not allowed to appear in the same subset.

Our results are a first step towards determining the dividing lines between polynomial time solvability, pseudo-polynomial time solvability, and strong \textsc{NP}-completeness of subset-sum related problems; we leave an interesting set of questions that need to be answered in order to obtain the complete picture.

1 Introduction

The problem of identifying subsets of equal value among the elements of a given set is constantly attracting the interest of various research communities due to its numerous applications, such as production planning and scheduling, parallel processing, load balancing, cryptography, and multi-way partitioning in VLSI design, to name only a few. Most research has so far focused on the version where

\textsuperscript{†} LA–UR–03:1158; work done while at ETH Zürich.
\textsuperscript{‡} Work partially done while at ETH Zürich, supported by the Human Potential Programme of EU, contract no HPRN-CT-1999-00104 (AMORE).
the subsets must form a partition of the given set; however, the variant where we skip this restriction is interesting as well. For example, the TWO EQUAL SUM SUBSETS problem can be used to show NP-hardness for a minimization version of PARTIAL DIGEST (one of the central problems in computational biology whose exact complexity is unknown) \cite{2}. Further applications may include: forming similar groups of people for medical experiments or market analysis, web clustering (finding groups of pages of similar content), or fair allocation of resources.

Here, we look at the problem from the point of view of a tournament organizer: Suppose that you and your friends would like to organize a soccer tournament (you may replace soccer with the game of your choice) with a certain number of teams that will play against each other. Each team should be composed of some of your friends and – in order to make the tournament more interesting – you would like all teams to be of equal strength. Since you know your friends quite well, you also know how well each of them plays. More formally, you are given a set of \( n \) numbers \( A = \{a_1, \ldots, a_n\} \), where the value \( a_i \) represents the excellence of your \( i \)-th friend in the chosen game, and you need to find \( k \) teams (disjoint subsets of \( A \)) such that the values of the players of each team add up to the same number.

This problem can be seen as a variation of BIN PACKING with fixed number of bins. In this new variation we require that all bins should be filled to the same level while it is not necessary to use all the elements. For any set \( A \) of numbers, let \( \text{sum}(A) := \sum_{a \in A} a \) denote the sum of its elements. We call our problem \( k \) EQUAL SUM SUBSETS, where \( k \) is a fixed constant:

**Definition 1 (\( k \) EQUAL SUM SUBSETS).** Given is a set of \( n \) numbers \( A = \{a_1, \ldots, a_n\} \). Are there \( k \) disjoint subsets \( S_1, \ldots, S_k \subseteq A \) such that \( \text{sum}(S_1) = \ldots = \text{sum}(S_k) \)?

The problem \( k \) EQUAL SUM SUBSETS has been recently shown to be NP-complete for any constant \( k \geq 3 \) \cite{3}. The NP-completeness of the particular case where \( k = 2 \) has been shown earlier by Woeginger and Yu \cite{8}. To the best of our knowledge, the variations of \( k \) EQUAL SUM SUBSETS that we study in this paper have not been investigated before in the literature.

We have introduced parameter \( k \) for the number of equal size subsets as a fixed constant that is part of the problem definition. An interesting variation is to allow \( k \) to be a fixed function of the number of elements \( n \), e.g. \( k = \frac{n}{q} \) for some constant \( q \). In the sequel, we will always consider \( k \) as a function of \( n \); whenever \( k \) is a constant we simply write \( k = O(1) \).

The definition of \( k \) EQUAL SUM SUBSETS corresponds to the situation in which it is allowed to form subsets that do not have the same number of elements. In some cases this makes sense; however, we may want to have the same

\footnote{Under a strict formalism we should define \( A \) as a set of elements which have values \( \{a_1, \ldots, a_n\} \). For convenience, we prefer to identify elements with their values. Moreover, the term “disjoint subsets” refers to subsets that contain elements of \( A \) with different indices.}