Efficent solutions for weight-balanced partitioning problems

Steffen Borgwardt\(^1\) and Shmuel Onn\(^2\)

\(^1\) sborgwardt@math.ucdavis.edu; University of California, Davis
\(^2\) onn@ie.technion.ac.il; Technion - Israel Institute of Technology, Haifa

Abstract. We prove polynomial-time solvability of a large class of clustering problems where a weighted set of items has to be partitioned into clusters with respect to some balancing constraints. The data points are weighted with respect to different categories and the clusters adhere to given lower and upper bounds on the total weight of their points with respect to each of these categories. Further the weight-contribution of a vector to a cluster can depend on the cluster it is assigned to. Our interest in these types of clustering problems is motivated by an application in land consolidation. Our framework maximizes an objective function that is convex in the summed-up utility of the items in each cluster. Despite hardness of convex maximization and many related problems, for fixed dimension and number of clusters, we are able to show that our clustering model is solvable in time polynomial in the number of items if the weight vectors come from a fixed, finite set. Finally, we exhibit a new model for land consolidation that can be approximated with provable and favorable error in polynomial time using our framework.

Keywords: constrained clustering, convex maximization, integer programming, land consolidation

1 Introduction

Partitioning a set of items while respecting some constraints is a frequent task in exploratory data analysis, arising in both operations research and machine learning; see e.g. [3,14]. We consider partitioning for which the sizes of the clusters are restricted with respect to multiple criteria. Our interest in these types of problems comes from an application in land consolidation [6,7,8,9]:

The farmers of many agricultural and private forest regions in Bavaria own a large number of small lots that are scattered over the whole region. For this, there is significant overhead driving and an unnecessarily high cost of cultivation. By means of lend-lease agreements, one can exchange the rights of cultivation for the existing lots in practice.

This corresponds to a combinatorial redistribution of the lots and can be modeled as a clustering problem where each lot is an item and each farmer is a cluster [9]. The lots can be represented by their midpoints in the Euclidean plane. They differ in several features like their size, quality of soil, shape, and attached subsidies, and some of these features are different for each farmer. For example some farmers may be eligible for subsidies if they cultivate some lots, and some may not. This situation often arises in private forests (where the cultivation refers to logging processes).

In the redistribution, the farmers have to receive lots that (approximately) sum up to their original total lots’ features in each category. The main goal is to create large connected pieces of land for each farmer. One way to do this is to use the geographical locations of the farmsteads of each farmer as a set of sites and then perform a weight-balanced least-squares assignment of the lots to these sites [5,6,11].

There are many other clustering applications where it is necessary to adhere to given bounds on the cluster sizes. For example, these include the modeling of polycrystals in the material sciences [1] and face recognition using meshes, where the original mesh is partitioned into parts of equal sizes to obtain an optimal running time for graph-theoretical methods that are applied to all of these parts [4].
Partitioning a weighted set of items into clusters of prescribed sizes (weight-balanced partitioning) is readily seen to be NP-hard, even for just two clusters and each item having just a single weight that is uniform for both clusters: deciding whether there is such a partition is at least as hard as the Subset Sum problem.

In the present paper, we consider the situation when these weight-balancing restrictions are defined using vectors from a fixed, finite domain. More precisely, each item specifies a vector of weights for each cluster (and these come from a fixed, finite domain). These vectors represent the weight the item contributes with respect to the different criteria if it is assigned to the respective cluster. The sizes of the clusters are bounded above and below with respect to all criteria.

Further, each item gives a vector of utility values with respect to each cluster representing the utility gained if the item is assigned to the corresponding cluster. (These vectors do not have to be from a fixed domain.) Each cluster ‘collects’ a total utility by summing up the utility of all its items’ utility vectors. We then maximize an objective function that is convex on the sum of utility vectors of each part.

These are the functions used for shaped partitioning [2,15,16,18], which encompass many of the objective functions commonly used in data analysis tasks, such as minimal-variance clustering; see e.g. [18]. They are intimately related to the studies of gravity polytopes [5,11]. Shaped partitioning is known to be NP-hard if either the dimension of the utility vectors or the numbers of clusters is part of the input [15], because it captures the hardness of convex maximization due to possible existence of exponentially many local optima [16]. For this, we fix the number of clusters and the dimension of the utility vectors in our analysis.

Our main result is to show that this framework then is polynomial-time solvable in the number of items. The tools we use are a combination of recent algebraic Graver bases methods [12,13,16,17] and geometric edge-directions and zonotope methods [15,18]. Further, we present a new, hard model for land consolidation that can be approximated with provably small error by modeling it using the expressive power of our framework.

The rest of the paper is organized as follows. In Section 2, we present a formal notation of the types of clustering models within our framework and state our main results. Section 3 is dedicated to the proofs of the main theorems. In Section 4, we then present to a new, hard model in land consolidation, and discuss why it can be approximated with favorable error in polynomial time.

2 Model and results

In the following, we partition \( n \) items \( \{1, \ldots, n\} \) into \( p \) clusters \( \pi_1, \ldots, \pi_p \). Each item \( j \) has a utility matrix \( C^j \in \mathbb{R}^{dp} \). Its \( i \)-th column \( C^j_{i} \) represents a vector of utility values gained if item \( j \) is assigned to cluster \( \pi_i \).

The utility of a clustering \( \pi = (\pi_1, \ldots, \pi_p) \) is \( f(\sum_{j \in \pi_1} C^j_{1}, \ldots, \sum_{j \in \pi_p} C^j_{p}) \), where \( f : \mathbb{R}^{dp} \rightarrow \mathbb{R} \) is a convex function. Note that \( \sum_{j \in \pi_i} C^j_{i} \) is the sum of utility vectors of the items in cluster \( \pi_i \).

Our task is to find a clustering of maximal utility under certain balancing constraints. We begin with a first model for these balancing constraints that we extend later on. First, we assume that we are given a fixed, finite set \( \Omega \subset \mathbb{Z}^p \) such that each item \( j \) has a weight vector \( w^j \in \Omega \). Informally, this vector contains a weight for each cluster \( \pi_i \) that it would contribute if assigned to \( \pi_i \). Further, each cluster \( \pi_i \) defines a total weight \( b_i \in \mathbb{Z} \) of the items that have to belong to it. Note that a finite cardinality \( m \) of \( \Omega \) can e.g. be achieved by bounding the absolute values of all components in the \( w^j \): for maximal absolute value \( \omega \), we obtain \( m \leq (2\omega + 1)^p \).

We write the corresponding set of restrictions on the cluster sizes as

\[
\sum_{j \in \pi_i} w^j_{i} = b_i \quad (i \in [p]).
\]
Let $\Pi$ denote the set of all partitions $\pi$ of $\{1, \ldots, n\}$. Then a full statement of this optimization problem $(P_1)$ would be

$$(P_1) \quad \max_{\pi \in \Pi} \ f(\sum_{j \in \pi_1} C_j^1, \ldots, \sum_{j \in \pi_p} C_j^p)
$$

$$\sum_{j \in \pi_i} w_j^i = b_i \quad (i \in [p])$$

Our first main result is polynomial-time solvability of this model for fixed $d$ and $p$.

**Theorem 1.** Suppose $d$ and $p$ are fixed and suppose there is a fixed set $\Omega \subset \mathbb{Z}^p$ such that all $w_j^i \in \Omega$. Then for every convex $f$ we can solve the problem $(P_1)$ in polynomial time.

In many applications, instead of having exact sizes of the partition parts, one is given lower and upper bounds on the sizes. To extend the above program to lower and upper bounds $b^+_i \in \mathbb{Z}$ on the total weights of the partition parts, we extend our formulation in several places. Formally, these constraints take the form

$$b^-_i \leq \sum_{j \in \pi_i} w_j^i \leq b^+_i \quad (i \in [p]).$$

For our later proofs, we now rewrite the corresponding optimization problem using only equalities. Let us introduce slack variables $s^\pm_i \in \mathbb{Z}$ for all $i \in [p]$. Then we can state the new optimization problem $(P_2)$ as

$$(P_2) \quad \max_{\pi \in \Pi} \ f(\sum_{j \in \pi_1} C_j^1, \ldots, \sum_{j \in \pi_p} C_j^p)
$$

$$\left(\sum_{j \in \pi_i} w_j^i\right) + s^+_i = b^+_i \quad (i \in [p])$$

$$\left(\sum_{j \in \pi_i} w_j^i\right) - s^-_i = b^-_i \quad (i \in [p])$$

$$s^\pm_i \geq 0 \quad (i \in [p])$$

With some modifications to the construction for $(P_1)$ in the proof of Theorem 1, we can show a similar statement for this more general class of problems.

**Theorem 2.** Suppose $d$ and $p$ are fixed and suppose there is a fixed set $\Omega \subset \mathbb{Z}^p$ such that all $w_j^i \in \Omega$. Then for every convex $f$ we can solve the problem $(P_2)$ in polynomial time.

Finally, we extend our model to allow for balanced weights with respect to $s$ different criteria. Instead of item $j$ listing a vector $w_j^i \in \mathbb{Z}^p$ of weights with respect to the clusters, it now has a matrix $W_j^i \in \mathbb{Z}^{sp}$ listing $s$-dimensional vectors $W_j^i$ of weights contributed to the cluster $\pi_i$ it is assigned to. All of these weight vectors come from a fixed, finite set $\Omega \subset \mathbb{Z}^{sp}$. As before, a finite cardinality $m$ of $\Omega$ can be achieved by bounding the absolute values of all components in the $W_j^i$: for maximal absolute value $\omega$, we now obtain $m \leq (2\omega + 1)^{sp}$.

Further, instead of $b^\pm \in \mathbb{Z}$, we now use $B^\pm_i \in \mathbb{Z}^*$, and likewise we have to use slack vectors $S^\pm_i \in \mathbb{Z}^*$ for all $i \in [p]$. This gives us an optimization problem $(P_3)$ as

$$(P_3) \quad \max_{\pi \in \Pi} \ f(\sum_{j \in \pi_1} C_j^1, \ldots, \sum_{j \in \pi_p} C_j^p)
$$

$$\left(\sum_{j \in \pi_i} W_j^i\right) + S^+_i = B^+_i \quad (i \in [p])$$

$$\left(\sum_{j \in \pi_i} W_j^i\right) - S^-_i = B^-_i \quad (i \in [p])$$

$$S^\pm_i \geq 0 \quad (i \in [p])$$

Even for this extension, we keep polynomial-time solvability.

**Theorem 3.** Suppose $d$ and $p$ are fixed and suppose there is a fixed set $\Omega \subset \mathbb{Z}^{sp}$ such that all $W_j^i \in \Omega$. Then for every convex $f$ we can solve the problem $(P_3)$ in polynomial time.
3 Proofs

The proofs for Theorems 1 to 3 have a common demeanor in that we exhibit polynomial-time and -size transformation of the corresponding problem to shaped partitioning for a set of constraints defined by an $n$-fold matrix. Generally, we aim for a problem statement of the form

$$\max \{ f(Cx) : A^{(n)}x = b, l \leq x \leq u, x \in \mathbb{Z}^N \},$$

where the $n$-fold matrix $A^{(n)}$ is derived from two matrices $A_1$ and $A_2$ by the standard construction in the form

$$A^{(n)} = \begin{bmatrix} A_1 \ldots A_1 \\ A_2 \\ \vdots \\ A_2 \end{bmatrix}.$$

Note that $n$ corresponds to the number of items; each of them gets a set of columns with one of the building blocks $A_1$ and $A_2$. The above problem statement is a variant of shaped partitioning.

To obtain polynomial-time-solvability, these matrices have to be polynomial-time and -size constructible from the input. In particular, $N$ has to be polynomial in the input, $A_1$ and $A_2$ have to be of constant size, and $C$ has to have a constant number of rows. Then the claim is a consequence of the polynomial-time solvability of convex integer programming as in [13].

Proof (Theorem 1). We prove the claim by transforming $(P_1)$ to a statement in the form

$$\max \{ f(Cx) : A^{(n)}x = b', l \leq x \leq u, x \in \mathbb{Z}^N \}.$$

Let us begin with the constraints. Recall that $m$ is the size of $\Omega$. Consider the two matrices

$$A_1 = \begin{bmatrix} w_1^1 & \ldots & w_1^m \\ w_2^1 & \ldots & w_2^m \\ \vdots & \ldots & \vdots \\ w_p^1 & \ldots & w_p^m \end{bmatrix} \in \mathbb{R}^{p \times mp}$$

and

$$A_2 = [1 \ldots 1] \in \mathbb{R}^{1 \times mp}.$$

Further, we define the vectors $b = (b_1, \ldots, b_p)^T$ and $1 = (1, \ldots, 1)^T \in \mathbb{R}^n$ to be able to write a system of equations $A^{(n)}x = b' = \begin{pmatrix} b \\ 1 \end{pmatrix}$. The variables $x \in \mathbb{R}^{n(mp)}$ here correspond to decision variables in the following way:

We have $x = (x^1, \ldots, x^n)^T$, where $x^j$ corresponds to the $j$-th column-block of the $n$-fold matrix, i.e. to the $j$-th 'copy' of $A_1$ and $A_2$. Each $x^j$ takes the form $x^j = (x^j_1, \ldots, x^j_m)^T$, i.e. it consists of $m$ blocks $x^j_i \in \mathbb{Z}^p$. Note that this construction yields $N = n(mp)$, which is polynomial in the input.

Further, let us define lower and upper bounds on $x$ in the form $l \leq x \leq u$: We choose lower bounds $l = 0 \in \mathbb{R}^N$ and define the upper bound vector $u = (u_1, \ldots, u^n)^T$ to consist of $n$ blocks $u^j = (u^j_1, \ldots, u^j_m)^T \in \mathbb{Z}^{mp}$, setting $u^j_i = 1 \in \mathbb{Z}^p$ if item $j$ has weight vector $w^j$ and $u^j_i = 0 \in \mathbb{Z}^p$ otherwise.

The system

$$A^{(n)}x = \begin{pmatrix} b \\ 1 \end{pmatrix}, l \leq x \leq u$$

can be derived in polynomial time and is of polynomial size, as we only use a polynomial number of copies of numbers from the original input (and of zeroes and ones). Let us discuss why it is equivalent to our original set of constraints. First, note that the lower and upper bounds force an integral solution $x$ to be a 0,1-solution.

The assignment of item $j$ to a cluster is determined by the decision variables $x^j$ corresponding to the $j$-th block of $A_1$ and $A_2$. The 'A2-block' of the system tells us that precisely one entry in $x^j$ is equal to 1. By the upper bounds on $x$, this can only be the case for an index
which corresponds to a correct combination of cluster \( \pi_i \) and weight contribution by item \( j \). For this, in the \( \{'A_{1}\}' \) block of the system, the correct weight for \( x_1 \) is added up in the equation to obtain total cluster size \( b_i \). Thus all items are assigned and all clusters obtain the correct total weight.

It remains to check whether the objective function can be written in form \( f(Cx) \), where \( C \) has a constant number of rows. Note that

\[
\sum_{j \in \pi_s} C^j = \sum_{j=1}^{n} \sum_{i=1}^{m} ((x_i^j)^T 1) C^j.
\]

Thus \( C \) takes the form

\[
C = [(C')^1 \ldots (C')^1 \ldots (C')^n \ldots (C')^n] \in \mathbb{R}^{dp \times n(mp)},
\]

with \( m \) consecutive copies for each of the \( (C')^j \). These are defined as

\[
(C')^j = \begin{bmatrix}
C^j_1 \\
C^j_2 \\
\vdots \\
C^j_p
\end{bmatrix} \in \mathbb{R}^{dp \times p}.
\]

The claim now follows from the number of \( dp \) rows being constant and observing that \( Cx \) yields the vector \( ((\sum_{j \in \pi_1} C^j_1)^T, \ldots, (\sum_{j \in \pi_p} C^j_p)^T)^T \).

Next, we prove Theorem 2 by extending the above construction. Our goal is to enter the slack variables \( s_i^+ \) into the model while preserving the \( n \)-fold structure of the program. To do so, we will duplicate them and move them into the building blocks of the matrix as follows.

**Proof (Theorem 2).**

Let \( A, b \) etc. refer to what we constructed for Theorem 1. We denote their corresponding new counterparts by a bar, e.g. \( \bar{A}, \bar{b} \). We now want a problem statement for \((P_2)\) of the form

\[
\max \{ f(\bar{C}x) : \bar{A}^{(n)}x = \bar{b}, \bar{l} \leq \bar{x} \leq \bar{u}, \bar{x} \in \mathbb{Z}^N \},
\]

where \( \bar{A}^{(n)} \) is an \( n \)-fold matrix derived by the standard construction using matrices \( \bar{A}_1 \) and \( \bar{A}_2 \).

For this formulation, we use a vector \( \bar{x} = (\bar{x}^1, \ldots, \bar{x}^n)^T \), where the column-blocks \( \bar{x}^j = ((s^j)^+, (s^j)^-, x^j)^T \) now also have copies of the slack variables \( (s^j)^+ = ((s^j_1)^+, \ldots, (s^j_p)^+) \) and \( (s^j)^- = ((s^j_1)^-, \ldots, (s^j_p)^-) \) for each of the original \( x^j \). Note that \( \bar{x} \in \mathbb{R}^\bar{N} \), where \( \bar{N} = n(2p+mp) \), which again is polynomial in the input.

\( \bar{A}^{(n)} \) is set to

\[
\bar{A}^{(n)} = \begin{bmatrix}
\bar{A}_1 & \cdots & \bar{A}_1 \\
\bar{A}_2 & \cdots & \bar{A}_2
\end{bmatrix},
\]

where

\[
\bar{A}_1 = \begin{bmatrix}
E & 0 & A_1 \\
0 & -E & A_1
\end{bmatrix} \in \mathbb{R}^{2p \times (2p+mp)} \quad \text{and} \quad \bar{A}_2 = [0 \quad A_2] \in \mathbb{R}^{1 \times (2p+mp)}
\]

with \( E \in \mathbb{R}^{p \times p} \) the unit-matrix and \( 0 \in \mathbb{R}^{p \times p} \) the 0-matrix. Further \( \bar{b}' = (\bar{b}, 1)^T \) with \( 1 = (1, \ldots, 1)^T \in \mathbb{R}^n \) and \( \bar{b} = (b_1^+, \ldots, b_p^+, b_1^-, \ldots, b_p^-)^T \) listing the upper and lower bounds on the cluster sizes.

Next, we define lower and upper bounds \( \bar{l}, \bar{u} \in \mathbb{R}^{\bar{N}} \) for \( \bar{x} \). As before, we use \( \bar{l} = 0 \in \mathbb{R}^{\bar{N}} \). Further, we use the upper bound vector \( \bar{u} = (\bar{u}^1, \ldots, \bar{u}^n)^T \), where \( \bar{u}^i = (\nu_i, \ldots, \nu_i)^T \in \mathbb{R}^{2p+mp} \) begins with \( 2p \) entries of value \( \nu = \sum_{i=1}^{p} \sum_{j=1}^{n} |w_i^j| \). Further \( \bar{u}^i = (0, \ldots, 0, u_i)^T \in \mathbb{R}^{2p+mp} \) begins

\[^3\text{In the notation of } \bar{b}' \text{ and in other places, we avoid double transposes when writing vectors for better readability when the context is clear. For example } c = (a, b)^T \text{ for two column vectors } a, b \text{ would be a column vector } c.\]
with $2p$ entries 0 for all $i > 1$. Note that in a feasible solution no slack variable ever is larger than $\nu$ and that $\nu$ has an encoding length that is polynomial in the input size of $(P_2)$.

Finally, we construct the new matrix $\tilde{C} = (\tilde{C}^1, \ldots, \tilde{C}^n) \in \mathbb{R}^{dp \times N}$ of $n$ building-blocks $	ilde{C}^j = [0 \; \ldots \; (C^j_1) \ldots (C^j_r)] \in \mathbb{R}^{dp \times 2p + mp}$, where $0 \in \mathbb{R}^{dp \times 2p}$ is a 0-matrix and there again are $m$ copies of the $(C^j_1)$ in the blocks $\tilde{C}^j$. Essentially, all parts that correspond to slack variables are ignored for the objective function value, i.e. $f(\tilde{C} \tilde{x}) = f(Cx)$.

The new system $A(n)\tilde{x} = \tilde{b}', \tilde{l} \leq \tilde{x} \leq \tilde{u}$, as well as the new objective function matrix $\tilde{C}$, extend the original system only by introducing an additional polynomial number of zeroes, ones and copies of numbers from the input. The matrices $A_1, A_2$ still have a constant number of rows and columns, and $\tilde{C}$ has a constant number of rows.

It remains to discuss why the new system is equivalent to the constraints of $(P_2)$. Clearly, the ‘$x$-part’ of $\tilde{x}$ still has to be a 0, 1-solution. These 0, 1-entries have the same role as in the original construction, as in the $A_2$-part the slack variables are only combined with zeroes.

The lower bounds on the slack variables force them to be non-negative. By the upper bounds, only the slack variables in the first block of the construction can be greater than zero. The other slack variables are only in the system to preserve the $n$-fold structure of the matrix. Note that it suffices to allow integral values for the slack variables (recall $\tilde{x} \in \mathbb{Z}^N$), as both the $w_i^j$ and the $b_i^{k_j}$ are integral. The rows of the $A_1$-blocks then guarantee that the cluster sizes plus the slack variables add up to the given lower and upper bounds. In particular, the clusters’ sizes lie in the given, bounded range. \hfill \Box

Finally, we prove Theorem 3 by extending the construction in the proof of Theorem 2 even further. We reuse the notation introduced for the proofs of both Theorem 1 and 2.

Proof (Theorem 3). We want a problem statement of $(P_3)$ of the form

$$\max \{ f(\tilde{C}x) : \tilde{A}^{(n)} \tilde{x} = \tilde{b}', \tilde{l} \leq \tilde{x} \leq \tilde{u}, \tilde{x} \in \mathbb{Z}^N \},$$

where $\tilde{A}^{(n)}$ is an $n$-fold matrix derived by the standard construction for matrices $\tilde{A}_1$ and $\tilde{A}_2$.

We use $\tilde{x} = (\tilde{x}^1, \ldots, \tilde{x}^n)^T$, where the column-blocks $\tilde{x}^j = ((S^j_1)^+, (S^j_2)^-, x^j)^T$ all contain their own copies of the slack vectors $(S^j_i)^+ = ((S^j_1)^+, \ldots, (S^j_l)^+) \in \mathbb{R}^{mp}$ and $(S^j_i)^- = ((S^j_1)^-, \ldots, (S^j_l)^-) \in \mathbb{R}^{mp}$. Herewith $\tilde{x} \in \mathbb{R}^N$ for $N = n(2sp + mp)$, which is polynomial in the size of the input.

For the $n$-fold construction of $\tilde{A}^{(n)}$, we now use

$$\tilde{A}_1 = \begin{bmatrix} E & 0 & A_1^T \\ 0 & -E & A_2^T \end{bmatrix} \in \mathbb{R}^{2sp \times (2sp + mp)}$$

with

$$A_1 = \begin{bmatrix} W_1^1 & W_1^m \\ W_2^1 & \ldots & W_2^m \\ \vdots & \ddots & \vdots \\ W_p^1 & \ldots & W_p^m \end{bmatrix} \in \mathbb{R}^{sp \times mp},$$

where $E \in \mathbb{R}^{sp \times sp}$ is the unit matrix and $0 \in \mathbb{R}^{sp \times sp}$ the 0-matrix. Further, we use

$$\tilde{A}_2 = \begin{bmatrix} 0 & 0 & A_2 \end{bmatrix} \in \mathbb{R}^{1 \times (2sp + mp)}.$$

The new right-hand side vector is $\tilde{b}' = (\tilde{b}, 1)^T \in \mathbb{R}^{2sp + n}$ with $1 = (1, \ldots, 1)^T \in \mathbb{R}^n$ and $\tilde{b} = (B_1^+, \ldots, B_p^+, B_1^-, \ldots, B_p^-)^T$ listing the upper and lower bound vectors on the cluster sizes.

Again, we use $\tilde{l} = 0 \in \mathbb{R}^N$ as lower bounds. The upper bound vector $\tilde{u} = (\tilde{u}^1, \ldots, \tilde{u}^n)^T \in \mathbb{R}^N$ contains $\tilde{u}^1 = (\tilde{\nu}, \ldots, \tilde{\nu}, u_1^1)^T \in \mathbb{R}^{2sp + mp}$ which begins with $2p$ vectors $\tilde{\nu} = \sum_{i=1}^p \sum_{j=1}^n |W_i^j| \in \mathbb{R}^s$ (where $|W_i^j|$ refers to the vector listing the absolutes in $W_i^j$ componentwisely), and $\tilde{u}_i^j = (0, \ldots, 0, u_i^j)^T \in \mathbb{R}^{2sp + mp}$ begins with $2sp$ entries 0 for all $i > 1$. The $u_i^j = (u_i^1, \ldots, u_i^m)^T$ are defined by setting $u_i^j = 1 \in \mathbb{Z}^p$ if item $j$ has weight matrix $W_i^j$ and $u_i^j = 0 \in \mathbb{Z}^p$ otherwise.

Finally, as in the construction for Theorem 2, we have to ignore the parts that correspond to slack variables for the objective function value. We do so by means of the matrix $\tilde{C} = (\tilde{C}^1, \ldots, \tilde{C}^n) \in \mathbb{R}^{dp \times N}$ that consists of $n$ building-blocks $\tilde{C}^j = [0 \; \ldots \; (C^j_1) \ldots (C^j_r)] \in \mathbb{R}^{dp \times 2sp + mp}$, where $0 \in \mathbb{R}^{dp \times 2sp}$ is a 0-matrix. Then $f(\tilde{C} \tilde{x}) = f(\tilde{C} \tilde{x}) = f(Cx)$. 

It remains to explain why this formulation represents the constraints of \((P_3)\). By definition of \(\tilde{l}\) and \(\tilde{u}\), the ‘\(x\)-part’ of \(\tilde{x}\) still is a 0,1-solution. These 0,1-entries play the same role as in the original construction by definition of the \(A_2\). The slack variables are non-negative, and by the upper bounds only the slack variables in the first block of the construction can be greater than zero. The components of \(\nu\) are sufficiently large to not impose a restriction. Again it suffices to allow integral values for the slack variables, as both the \(W^j\) and the \(B_i^\pm\) are integral. The rows of the \(A_1\)-blocks then guarantee that the cluster sizes plus the slack variables add up to the given lower and upper bounds for each component of the weight vectors of the clusters, which implies that all of the clusters’ weight categories lie within the range given by the \(B_i^\pm\).

4 An application in land consolidation

Let us exhibit an application that uses the full expressive power of our model. In many agricultural regions of Bavaria and Middle Germany, farmers own a large number of small lots that are scattered over the region. In such a situation, a land consolidation process can be initiated to improve on the cost-effective structure of the region. Voluntary land exchanges (or lend-lease agreements) are an increasingly popular method for such a process: The existing lots are kept without changes and the rights of cultivation are redistributed among the farmers of the region.

Here a natural constraint is that - after the redistribution - each farmer should have lots of the same characteristics as before: the same total value, total size, quality of soil, or subsidies attached. For this, such a redistribution can be modeled as a clustering problem under certain balancing constraints. See [6,7,8,9].

4.1 Weight-balancing constraints

Let us connect the general definitions in our model with the application. We here extend the model in [6]. The items are the \(n\) lots which have to be divided among the \(p\) farmers \(\pi_1, \ldots, \pi_p\) (the clusters). First, we turn to the balancing constraints.

The lots differ in \(s\) features, for example size, value (which is impacted by the quality of soil), attached subsidies, and so on. While some of these measures are independent of which farmer the lot is assigned to - for example the size and quality of soil are just fixed numbers - others may not be. The subsidies a farmer gets for cultivating a lot depend on several factors that differ between farmers - for example depending whether the farmer represents a small local family or a large agricultural business working in multiple regions.

For each lot, we set up a matrix \(W^j \in \mathbb{R}^{sp}\) that has a vector \(W^j_i\) of the \(s\) features that farmer \(\pi_i\) gets if lot \(j\) is assigned to him. Let \(B_i \in \mathbb{R}^s\) list the summed-up total features of farmer \(\pi_i\). In the redistribution process, the farmers do not accept a large deviation with respect to any of these measures. For an accepted change of 3\%, we obtain \(B_i^- = 0.97 \cdot B_i\) and \(B_i^+ = 1.03 \cdot B_i\). The accepted deviations can also be defined differently for the different measures.

4.2 Objective function

In the following, in a generalization of the model in [6], we present a family of objective functions for land consolidation that use the expressive power of our model. The lots are represented by their centers \(x_j \in \mathbb{R}^2\). In many regions, the farmers \(\pi_i\) specify the location \(v_i \in \mathbb{R}^2\) of their farmstead. We measure the distance of a lot and a farmstead by the square of their Euclidean distance \(\|v_i - x_j\|^2\).

In our objective function, we will use the sizes of the lots of each farmer. The information on the size of lots is represented in \(W^j \in \mathbb{R}^{np}\); in the following we give it an explicit name \(\omega_j\) for lot \(j\).

We begin with the classical least-squares assignment for a given, single location \(v_i\) of a farmstead (as a basic building block of our model). The lots are assigned to the farmers such that

\[
\sum_{i=1}^{p} \sum_{j \in \pi_i} \omega_j \|v_i - x_j\|^2
\]  
\((f_1)\)
is minimized. The sizes $\omega_j$ of the lots are used as scaling factors for the distances $\|v_i - x_j\|^2$ to have a fair treatment of the assignment of one large lot or of many small lots.

It remains to explain why this objective function fits our framework. This means we have to show that it can be represented by a function $f : \mathbb{R}^{dp} \to \mathbb{R}$ that is passed the $p$ arguments $\sum_{j \in \pi_i} C^j_i$ and is convex on each of these sums [18].

For each item $j$, we use $C^j_i = -\omega_j \|v_i - x_j\|^2$, so that $\sum_{j \in \pi_i} C^j_i = -\sum_{j \in \pi_i} \omega_j \|v_i - x_j\|^2$. It then suffices use the linear function $f$ that sums up its arguments. Formally $f(y) = 1^T y$ with $1 = (1, \ldots, 1)^T \in \mathbb{R}^p$.

Such a least-squares assignment favors a good assignment of lots of a farmer with larger total size over a good assignment of a smaller farmer. Thus one may want to ‘normalize’ the different parts of this sum. We introduce this new approach as normed least-squares assignment.

For this, we define $C^j_i = (-\omega_j \|v_i - x_j\|^2) \in \mathbb{R}$ and use a function $f : \mathbb{R}^{2p} \to \mathbb{R}$ that sends $y = \left(\left(y_1^{(1)}, \ldots, y_{p+1}^{(1)}\right), \ldots, \left(y_1^{(p)}, \ldots, y_{p+1}^{(p)}\right)\right)$ to $f(y) = \sum_{i=1}^p \frac{1}{\omega_i} \sum_{j \in \pi_i} \omega_j \|v_i - x_j\|^2$. As $\sum_{j \in \pi_i} C^j_i = \left(-\sum_{j \in \pi_i} \omega_j \|v_i - x_j\|^2\right)$, we then obtain a minimization of

$$\sum_{i=1}^p \frac{1}{\omega_i} \sum_{j \in \pi_i} \omega_j \|v_i - x_j\|^2.$$  

By scaling by the inverse of each farmer’s total assigned land, each farmer’s quality of assignment contributes evenly to the final objective function value.

Unfortunately, $f(y) = \sum_{i=1}^p \frac{1}{\omega_i} y_{i+1}^{(i)}$ is not convex, even when restricted to a strictly positive domain, so we have to resort to a (provably good) approximation for $f_2$. For this, we denote the original total size of each farmers’ lots by $\kappa_i$ and the accepted lower and upper bounds on the sizes as $\kappa_i^\pm$. We then use these apriori $\kappa_i$ to estimate the total size of lots of each farmer after the redistribution. (Compare this to the use of an approximate center of gravity in [11].)

This results in a minimization of

$$\sum_{i=1}^p \frac{1}{\kappa_i} \sum_{j \in \pi_i} \omega_j \|v_i - x_j\|^2,$$  

which is a linear transform of a least-squares assignment and thus fits our framework, too. We just have to use $C^j_i = -\frac{\omega_j}{\kappa_i} \|v_i - x_j\|^2$.

Let us discuss the approximation error we do by optimizing $f_3$ instead of $f_2$.

**Lemma 1.** Let $\pi$ be an optimal partition for $f_3$ and $\pi'$ be optimal for $f_2$. Then $\pi$ is a $(\max_{i \leq p} \frac{\kappa_i^-}{\kappa_i})(\max_{i \leq p} \frac{\kappa_i^+}{\kappa_i})$-approximation with respect to $f_2$.

**Proof.** For a simple notation, we refer to the corresponding objective function values as $f_3(\pi)$ and $f_2(\pi')$. As $\kappa_i^- \geq \sum_{j \in \pi_i} \omega_j \leq \kappa_i^+$, we get both $f_2(\pi) \leq (\max_{i \leq p} \frac{\kappa_i^-}{\kappa_i}) f_3(\pi)$ and $f_3(\pi) \leq f_2(\pi') \leq (\max_{i \leq p} \frac{\kappa_i^+}{\kappa_i}) f_2(\pi')$, which combines to $f_2(\pi) \leq (\max_{i \leq p} \frac{\kappa_i^-}{\kappa_i}) \cdot (\max_{i \leq p} \frac{\kappa_i^+}{\kappa_i}) \cdot f_2(\pi')$. Thus we obtain a $(\max_{i \leq p} \frac{\kappa_i^-}{\kappa_i})(\max_{i \leq p} \frac{\kappa_i^+}{\kappa_i})$-approximation. \hfill $\square$

For an accepted upper and lower deviation of 3%, $f_3$ would yield a 1.0609-approximation error for $f_2$.

**4.3 Other objective functions**

Note that $f_3$ is linear and is defined with $d = 1$, but uses the fact that $C^j_i$ and $C^j_{i_2}$ can differ for $i_1 \neq i_2$. In the literature there are several examples for objective functions that use a larger $d$, but that only fit our framework for $\kappa_i^- = \kappa_i^+$ for all $i \leq p$. Examples include finding a partition of minimal variance [13,18] and pushing apart the centers of gravity of the partition parts [7,10].
Both of them can then be interpreted as norm-maximization over a gravity polytope or a shaped partition polytope.

In general, the convex functions \( f : \mathbb{R}^{dp} \to \mathbb{R} \) of the form \( f(\sum_{j \in \pi_1} C^j_1, \ldots, \sum_{j \in \pi_p} C^j_p) \) cover a wide range of objective functions commonly used in clustering. In particular, they directly represent the aggregation of utility values. Each of the \( p \) clusters \( \pi_i \) contributes a summed-up utility vector \( \sum_{j \in \pi_i} C^j_i \) of its items and these vectors then are aggregated to a value in \( \mathbb{R} \) by the convex function \( f \). A simplest case for \( f \) is to just add up the components of all these vectors possibly (scaled by a factor), as we did in \( f_3 \). If the components of all \( C^j_i \) are non-negative, such an objective function corresponds to a linear transform of the \( l_1 \)-norm. Recall that all norms are convex functions. By choosing \( l_s \) with \( s > 1 \), one values single components in the utility vectors of the clusters higher in relation to many equally large values. For example, for \( l_\infty \), only the largest absolute values among all components counts.

One can represent even more general ways of aggregating the utility by turning to the so-called clustering bodies [10]: Here one combines two norms, one \( \| \cdot \| \) for \( \mathbb{R}^d \) on the parts \( \sum_{j \in \pi_i} C^j_i \) and a monotone norm \( \| \cdot \|_\ast \) for \( \mathbb{R}^p \) aggregating these values. The objective function then takes the form

\[
\| (\sum_{j \in \pi_1} C^j_1, \ldots, \sum_{j \in \pi_p} C^j_p) \|_\ast = \|(\sum_{j \in \pi_1} \| C^j_1 \|, \ldots, \sum_{j \in \pi_p} \| C^j_p \|) \|_\ast.
\]

The level set for value at most one is a convex body, a clustering body, and serves as the unit ball for a semi-norm.

### 4.4 Polynomial-time solvability

We now have all of the input for our clustering framework. Theorem 3 tells us that for a fixed number \( p \) of farmers, and if there is but a fixed set of vectors of lot features, this model is solvable in polynomial time in the size of the input, in particular \( n \).

The fixed set \( \Omega \) of vectors of lot features may come from some expert knowledge: For example, if especially large slots (\( > 5 \) hectare) are not traded at all and one does not distinguish between lot sizes that differ by less than a tenth of a hectare, one obtains a finite domain of lot sizes. The quality of soil is typically measured with a number in between 1 and 100, and the value of a lot comes from multiplying this number with its size. Then we have a finite domain of values, too. The same is possible for the subsidies and other measures.

Let us sum up our information.

**Theorem 4.** Suppose \( d \) and \( p \) are fixed and suppose there is a fixed set \( \Omega \subset \mathbb{Z}^{dp} \) such that all \( W^j \in \Omega \). Then the above model for land consolidation can be solved exactly in polynomial time for objective functions \( f_1 \) and \( f_2 \). For \( f_3 \) a \( (\max_{i \leq p} \frac{\kappa}{\kappa_i}, \max_{i \leq p} \frac{\kappa^+}{\kappa_i}) \)-approximation can be computed in polynomial time.

### Acknowledgements

The first author gratefully acknowledges support from the Alexander-von-Humboldt Foundation. The second author was partially supported by a VPR Grant at the Technion any by the Fund for the Promotion of Research at the Technion.

### References