The Geometry of Truthfulness.

Angelina Vidali*

September 30, 2009

Abstract

We study the geometrical shape of the partitions of the input space created by the allocation rule of a truthful mechanism for multi-unit auctions with multidimensional types and additive quasilinear utilities. We introduce a new method for describing the allocation graph and the geometry of truthful mechanisms for an arbitrary number of items(/tasks). Applying this method we characterize all possible mechanisms for the case of three items.

Previous work shows that Monotonicity is a necessary and sufficient condition for truthfulness in convex domains. If there is only one item, monotonicity is the most practical description of truthfulness we could hope for, however for the case of more than two items and additive valuations (like in the scheduling domain) we would need a global and more intuitive description, hopefully also practical for proving lower bounds. We replace Monotonicity by a geometrical and global characterization of truthfulness.

Our results apply directly to the scheduling unrelated machines problem. Until now such a characterization was only known for the case of two tasks. It was one of the tools used for proving a lower bound of $1+\sqrt{2}$ for the case of 3 players. This makes our work potentially useful for obtaining improved lower bounds for this very important problem.

Finally we show lower bounds of $1+\sqrt{n}$ and n respectively for two special classes of scheduling mechanisms, defined in terms of their geometry, demonstrating how geometrical considerations can lead to lower bound proofs.

1 Introduction

Mechanism design is the branch of game theory that tries to implement social goals taking into account the selfish nature of the individuals involved. Mechanism design constructs allocation algorithms that together with appropriate payments elicit from the players their secret values or preferences. In this paper we give a characterization result that reveals the exact geometry of truthful mechanisms. The goal of this paper is to understand and visualize truthful mechanisms better. We realized the need for such a result while trying to improve the lower bound for the scheduling selfish unrelated machines problem [15, 6, 11], however the result is more broadly applicable and interesting from itself continuing a line of research attempting to grasp truthfulness better [18, 10, 13, 1, 4]. What differentiates our work from this line of research is that we fully exploit the linearity in the geometry of additive valuations.

There exists a simple necessary and sufficient condition for truthfulness in convex domains and a finite number of outcomes, the Monotonicity Property. In single parameter domains, like for example in an auction where there is only one item, monotonicity is exactly the monotonicity we know from calculus and the most practical description of truthfulness we could hope for. The allocation should be a monotone (for the case of auctions an increasing,

*Department of Informatics, University of Athens, Email: avidali@di.uoa.gr

while for the case of scheduling a decreasing) function of the player's valuation for the item. However for the case of two or more items Monotonicity is a local condition that should be satisfied by any pair of instances of the problem and does not give us any clue about the global picture of the mechanism, when considering the whole space of inputs together. We would instead need a global and more intuitive description, hopefully also practical for proving lower bounds. We replace Monotonicity by a geometrical and global characterization of truthfulness, for the case when the valuations are additive.

Until now such a characterization was known in the context of the scheduling unrelated machines problem only for the easy case of two tasks [8] and it turned out to be a a quintessential element of the characterization proof in [7] and the lower bound in [8]. We believe that our result here can be used for obtaining new lower bounds. The only discouraging fact is that even for the case of 3 tasks the different mechanisms are too many and geometrically complicated.

No matter how many are the players participating in a mechanism, determining whether a mechanism is truthful boils down to a single-player case. Truthfulness requires that for fixed values of the other players, a player should not be able to increase his utility by lying. Studying the mechanism for fixed values of the other players is like studying a single-player case. Consequently in our setting there is a single player and m different indivisible items (or tasks). The player's type is denoted by the vector $t = (t_1, \ldots, t_m)$, where t_i is the valuation of the bidder for the i-th item/task and the allocation is denoted by $a = (a_1, \ldots, a_m)$ where $a_i \in \{0, 1\}$.

We assume that the bidder has additive valuations and hence the bidder's valuation function when his type is t and his allocation a is $v_t(a) = a \cdot t$. In fact it is easy to see that our results also apply if the valuations are of the form $v_t(a) = \lambda(a \cdot t) + \gamma_a$ for some constants λ, γ_a (we can have one different γ_a for each different allocation a). The reason for this is simple namely these valuations also satisfy the Monotonicity Property and moreover the possible truthful mechanisms for such valuations are like in Figure 1 (this would not be the case for valuations with $v_t(11) = t_1 \cdot t_2$ or $v_t(11) = 2t_1 + t_2$ as the sloped hyperplane would not be $45°$). A mechanism consists of an allocation algorithm a and a payment algorithm p. We make the standard assumption that the utilities are quasilinear, that is the utility of the player is $u(a, t) = v_t(a) - p(a)$.

The allocation part of the mechanism gives a partition of the space $\mathbb{R}^m$ of possible values of a player to 2^m different regions, one for each possible different allocation a of the player. But which are exactly the possible partitions of the space the mechanism creates? This is exactly the question we address in this paper.

We know [18] that a mechanism is truthful if and only if its allocation part satisfies the monotonicity property.

Definition 1 (Monotonicity Property). *An allocation algorithm is called monotone if it satisfies the following property: for every two sets of tasks t and t' the associated allocations a and a' satisfy $(a - a') \cdot (t - t') \leq 0$, where $\cdot$ denotes the dot product of the vectors, that is, $\sum_{j=1}^{m} (a_j - a'_j)(t_j - t'_j) \leq 0$.*

Notice that the Monotonicity Property is a necessary and sufficient condition for truthfulness and that it only involves the allocation part of the mechanism. Consequently by determining the possible partitions of the input space created by the allocation part of the mechanism we will eventually give a characterization of truthfulness.

Our problem can be reformulated as an interesting, simple and fun geometrical problem (forgetting everything about mechanisms and game theory) as follows:

Definition 2 (Geometrical statement of the problem)**.** *Suppose you have the m-dimensional cube $[0,1]^m$. The vector a, formed by the coordinates of each one of the vertices of the cube, is the "allocation"(/label) at this point. Consequently there are 2^m different possible "allocations". We want to give to each one of the points in the interior of the cube one of the possible "allocations" so that the monotonicity condition $(t-t')(a-a') \leq 0$ is satisfied for each pair of different points t, t' in the cube and their corresponding "allocations" a, a'. Which are the partitions of $\mathbb{R}^m$ that satisfy this property?*

As it has already been noticed in [10] in the case of additive valuations the boundaries of the mechanism are hyperplanes of a very specific form, every region created by this partition is a convex polyhedron. In this paper we show exactly which (rather few) polytopes are involved in such a partition. For proving our results we reduce the problem to that of determining the allocation graph of the mechanism, i.e. which of the regions share a common boundary. We can then determine the exact geometrical shape of the mechanism because the hyperplane that separates two regions can be easily derived from the monotonicity property.

Our results apply directly to the scheduling unrelated machines problem giving lower bounds for two very interesting special cases of the problem.

Definition 3 (The scheduling unrelated machines problem)**.** *The input to the scheduling problem is a nonnegative matrix t of n rows, one for each machine-player, and m columns, one for each task. The entry t_{ij} (of the i-th row and j-th column) is the time it takes for machine i to execute task j. Let t_i denote the times for machine i, which is the vector of the i-th row. The output is an allocation $a = a(t)$, which partitions the tasks into the n machines. We describe the partition using indicator values $a_{ij} \in \{0,1\}$: $a_{ij} = 1$ iff task j is allocated to machine i. We should allocate each task to exactly one machine, or more formally $\sum_{j=1}^{n} a_{ij} = 1$. The goal is to minimize the makespan, i.e. to minimize the total processing time of the player that finishes last.*

1.1 Our Tools

Besides the potential applications of our characterization, we believe that also the method we introduce for studying the allocation graph is of particular interest as it provides a very simple way to handle a very complicated partition of the space.

We propose a new, practical, method for determining all possible allocation graphs and the geometrical shapes of the mechanism: For each region R_a of the mechanism instead of considering its complicated geometrical shape we define a box that contains the region. The signs of distances between parallel to each other boundaries of the mechanism determine whether two of these boxes intersect. If two boxes intersect then the corresponding regions share a common boundary. Alternatively if two boxes intersect then there is an edge between the corresponding edges in the allocation graph. These distances however are not independent from each other. Applying cycle-monotonicity for appropriately chosen zero-length cycles allows us to determine how these constants relate. As boundaries between regions that differ only in one allocation always exist we will concentrate on the subgraph of the allocation graph that consists of the edges corresponding to Hamming distance-1 boundaries. For m tasks it is practical to consider this graph as an m-dimensional hypercube.

1.2 Related work

Myerson [14] gave a characterization of truthful algorithms for one-parameter problems, in terms of a monotonicity condition, which was rediscovered by Archer and Tardos [2]. For the case of multidimensional types Bikchandani et al. [5] prove that a simple necessary monotonicity property of the allocations of different inputs (and without any reference to payments) is also sufficient for truthful mechanisms, while Gui, Müller, and Vohra [10] extend this to a greater variety of domains (this work is rather close to ours as it also follows a geometrical approach). Saks and Yu [18] generalize this result to cover all convex domains of finitely many outcomes. Monderer [13] showed that this result cannot be essentially extended to a larger class of domains. Both these results concern domains of finitely many outcomes. There are however cases, like the fractional version of the scheduling problem, when the set of all possible allocations is infinite. For these, Archer and Kleinberg [1] provided a necessary and sufficient condition for truthfulness. Very recently Berger et al. [4] generalize all these results for the case of convex valuations.

Nisan and Ronen introduced the mechanism-design version of the scheduling problem on unrelated machines in the paper that founded the algorithmic theory of Mechanism Design [15, 16]. They showed that the well-known VCG mechanism, which is a polynomial-time algorithm and truthful, has approximation ratio n. They conjectured that there is no deterministic mechanism with approximation ratio less than n. They also showed that no mechanism (polynomial-time or not) can achieve approximation ratio better than 2. This was improved to $1+\sqrt{2}$ [8], and further to $1+\varphi$ in [11]. For the case of two machines [9] Dobzinski and Sundararajan characterized all mechanisms with finite approximation ratio, while [7] gave a characterization of all (regardless of approximation ratio) decisive truthful mechanisms in terms of affine minimizers and threshold mechanisms.

In a very recent paper [3] Ashlagi, Dobizinski and Lavi prove a lower bound of n for a special class of mechanisms, which they call "anonymous". Lavi and Swamy [12] considered another special case of the same problem when the processing times have only two possible values low or high, and devised a deterministic 2-approximation truthful mechanism; the use of cycle monotonicity played a central role in this work as well.

2 Preliminaries

For any two different assignments a, b for player i we define $f_{a:b} := \sup\{(a-b)\cdot t \mid t \in R_a\}$.

Definition 4. *We denote by R_a the closure of the subset of $\mathbb{R}^m$ where the mechanism gives assignment a and we will call it a* region *of the mechanism.*

We define the Hamming Distance $\mathrm{Hd}(a,b)$ between two vectors a, b, as the number of positions in which the two vectors are different. The *Minkowski sum* of two sets $A, B \subseteq \mathbb{R}^m$ is $A \oplus B = \{a+b \mid a \in A, b \in B\}$. Let also $B_a := \{t \mid (-1)^{a_j} t_j \geq 0,\ j = 1, \dots, m\}$. For $m = 2$ each B_a is a quadrant of $\mathbb{R}^2$.

Lemma 1. *a) If a point b belongs to region R_a of a truthful mechanism, then also $b \oplus B_a \subseteq R_a$.*

b) Regions R_a and $R_{a'}$ are separated by the hyperplane $(a-a')\cdot t = f_{a:a'}$ and each region is bounded by a convex polytope.

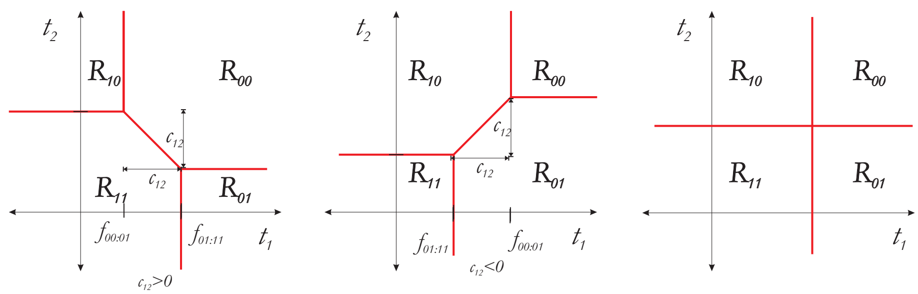

Figure 1: The two possible ways to partition the positive orthant for the case of 2 tasks and the threshold mechanism as a degenerate case of both.

c) Every region R_a satisfies $R_a \subseteq F_a \oplus B_a$ where $F_a := (f_{a:a_{-1},1-a_1}, \ldots, f_{a:a_{-m},1-a_m})$. In other words region R_a is included in the box we get by shifting the box B_a so that it that has its vertex at the point F_a.

This means that every region R_a is included in a box defined by the boundaries of R_a with all regions R_b such that $\mathrm{Hd}(a, b) = 1$. The proof is immediate by the monotonicity property and the definition of $f_{a:b}$.

2.1 The allocation graph of each player

We define an edge-weighted directed graph G, *the allocation graph*, whose vertex set are all possible allocations of the player. For each two allocations a, b the weight of the edge from a to b is $f_{a:b}$.

The following property is necessary and sufficient for truthfulness [17].

Definition 5 (Cycle monotonicity). *An allocation algorithm satisfies cycle monotonicity if for every integer K and cycle $a_1, \ldots, a_K, a_{K+1} = a_1$ on the allocation graph $\sum_{k=1}^{K} f_{a_k:a_{k+1}} \leq 0$.*

The following Lemma is an essential tool for our proofs. (We omit its proof, which is an extension of [[18], Proposition 5].)

Lemma 2. *Two regions $R_a, R_{a'}$ that share at least one common boundary point satisfy $p(a) - p(a') = f_{a:a'} = -f_{a':a}$.*

Lemma 3. *Any cycle on the allocation graph in which each pair of consecutive nodes corresponds to a pair of regions sharing a common boundary point has length zero.*

Remark 1. *An alternative definition for $f_{a:a'}$ would be to define it as $p(a) - p(a')$. The two definitions coincide when the regions $R_a, R_{a'}$ have a common boundary point, and these are also the pairs of allocations for which we will need this definition.*

3 New tools for the case of m items

The mechanism consists of sloped hyperplanes, as well as hyperplanes vertical to some axis, which we will call Hd-1 boundaries (because they separate regions that have Hamming distance equal to 1, i.e. differ in only one task). The trouble with the sloped hyperplanes is that they do appear as boundaries in all possible shapes of the mechanism, so we have to take cases. Luckily the hyperplanes vertical to some axis appear in all possible shapes. We will use the distance between these hyperplanes in order to describe the allocation graph of the mechanism. The sign of these distances determines exactly which of the sloped lines appear in the geometrical picture of the mechanism.

3.1 The example of two tasks demonstrates the idea

The idea of our approach is best depicted if we apply it for the easy case of two tasks (for which we already know that the two possible mechanisms are depicted in Figure 1). We observe that the two lines ($t_1 = f_{11:01}$ and $t_1 = -f_{00:10}$) that are vertical to the axis t_1 and the two lines ($t_2 = f_{11:10}$ and $t_2 = -f_{01:00}$) that are vertical to the axis t_2 have the same distance (otherwise the sloped line would not be 45°).

Another, purely algebraic and more straightforward way, to obtain this fact is to just to apply once cycle monotonicity. Taking the cycle $00 \to 01 \to 11 \to 10 \to 00$ we get $f_{00:10} + f_{10:11} + f_{11:01} + f_{01:00} = 0$ or equivalently $f_{11:01} + f_{00:10} = f_{11:10} + f_{00:01}$. If we define $c_{12} := f_{11:01} + f_{00:10}$ (This is the distance between the two lines vertical to the axis t_1.) then by the previous cycle it turns out that the distance between the two lines vertical to the axis t_2, which can be expressed as $f_{11:01} + f_{00:10}$ is also equal to c_{12}. Notice that we did not take two cases and did no drawing.

We are now ready to describe the allocation graph of the mechanism: Region R_{11} is contained in the box $t_1 \leq f_{11:01}, t_2 \leq f_{11:10}$. Region R_{00} is contained in the box $t_1 \geq f_{10:00}, t_2 \geq f_{01:00}$. Regions R_{11} and R_{00} share a common boundary line if and only if the boxes that contain them intersect i.e. if and only if $c_{12} > 0$. (Similarly regions R_{01} and R_{10} share a common boundary line if and only if the boxes that contain them intersect i.e. if and only if $c_{12} < 0$.) That is the sign of c_{12} determines which of the two possible shapes has the mechanism.

Knowing the allocation graph we can then very easily draw the picture of the mechanism. In what follows we generalize this idea to describe the allocation graph and the geometry of a truthful mechanism.

3.2 Expressing the distances between regions

We proceed to define some constants that generalize this idea we demonstrated for the case of two tasks. The constant $c_{ij|a_{-\{i,j\}}}$ measures the distance between the separating hyperplanes of the mechanism $t_i = f_{11a_{-\{i,j\}}:01a_{-\{i,j\}}}$ and $t_i = f_{10a_{-\{i,j\}}:00a_{-\{i,j\}}}$, which are two parallel hyperplanes corresponding to Hamming distance 1 boundaries. This constant fully describes the geometry of the mechanism if the allocation of all tasks, except for tasks i, j, is fixed to $a_{-\{i,j\}}$. To provide some intuition why we choose these consider that in a decisive mechanism this would give an asymptotic picture of the mechanism: If the values of only two tasks i, j are allowed to be variables, while the remaining tasks with allocation 1 are fixed to the biggest possible value ($+\infty$) and the tasks with allocation 0 are fixed to the smallest possible value, this constant describes the geometry of the mechanism that allocates tasks i, j.

Definition 6. *For all i, j and all possible $m-2$-tuples (/allocations) $a_{-\{i,j\}}$ we define*

$$c_{ij|a_{-\{i,j\}}} := f_{11a_{-\{i,j\}}:01a_{-\{i,j\}}} + f_{00a_{-\{i,j\}}:10a_{-\{i,j\}}} = f_{11a_{-\{i,j\}}:10a_{-\{i,j\}}} + f_{00a_{-\{i,j\}}:01a_{-\{i,j\}}} \quad (1)$$

But are these constants independent from each other? As the following Lemma shows, the answer is no and the relation between these constants is derived from Cycle Monotonicity.

Lemma 4. *If a mechanism is truthful then the constants $c_{ij|a_{-\{i,j\}}}$ satisfy the following equation:*

$$c_{ij|1a_{-\{i,j,k\}}} - c_{ij|0a_{-\{i,j,k\}}} = c_{ik|1a_{-\{i,j,k\}}} - c_{ik|0a_{-\{i,j,k\}}} \quad (2)$$

Proof. (Sketch) We get this from the following cycle (also depicted in Figure 3) $f_{111:011} + f_{011:010} + f_{010:000} + f_{000:001} + f_{001:101} + f_{101:100} + f_{100:110} + f_{110:111} = 0$. □

By Lemma 1 each region R_a of the mechanism is contained in a box formed by the separating hyperplanes between R_a and all regions with assignment in Hamming distance 1 from a. If we concentrate on a pair of intersecting regions, then the boxes that contain them have a non-empty intersection. But it is also the other way round (we include the proof of the next Lemma in the Appendix):

Lemma 5. *If the boxes corresponding to two regions intersect then the regions share a common boundary hyperplane.*

We proceed to define $d^i_{a:b}$ as the difference of the Hd-1 boundaries on axis i corresponding to two distinct regions R_a, R_b. We have $d^i_{a:1-a} > 0$ for all $i = 1, \dots, m$ if and only if regions R_a and R_{1-a} intersect.

Even though the geometry of the mechanism is complicated it turns out that we can derive a general formula for the $d^i_{a:b}$s using now a more complicated zero-length cycle on the allocation graph.

Definition 7. *We define the distance $d^i_{a:b} := f_{a:1-a_i,a_{-i}} + f_{b:1-b_i,b_{-i}}$.*

Lemma 6. *We have $d^i_{a:b} = d^i_{b:a}$ (symmetry) and $d^i_{a:b} = -d^i_{1-a_i,a_{-i}:1-b_i,b_{-i}}$.*

Lemma 7. *The distance $d^i_{a:1-a}$ can be expressed as the following sum of constants:*

$$d^i_{a:1-a} := \sum_{j \neq i, j \in \{1,\dots,m\}} (-1)^{a_i + a_j} c_{ij|b_{-\{i,j\}}},$$

where the k-th coordinate of the allocation b_k is $b_k = \begin{cases} 1 - a_k & \text{if } k < j \\ a_k & \text{if } k > j. \end{cases}$

For example we have

$$d^1_{a:1-a} := (-1)^{a_1+a_2} c_{12|a_{-\{1,2\}}} + (-1)^{a_1+a_3} c_{13|1-a_2 a_{-\{1,2,3\}}} + (-1)^{a_1+a_4} c_{14|1-a_2 1-a_3 a_{-\{1,2,3,4\}}} + \dots + (-1)^{a_1+a_m} c_{1m|1-a_2 1-a_3 \dots 1-a_{m-1}}.$$

Note that $d^1_{11:00} = c_{1,2} = c$ so Definition 7 is just an extension of Definition 6.

Proof. For simplicity we will give the details of the proof for $i = 1$ and m tasks, that is if we want to compute d^1. We take the path that at each step changes the allocation of a single task in the following order: $(1, 2, 3\dots, m, 1, m, m-1, \dots, 2)$. The path is

$$
\begin{aligned}
f_{a_1a_2\dots a_m:1-a_1a_2\dots a_m} &+ f_{1-a_1a_2a_3\dots a_m:1-a_11-a_2a_3\dots a_m} + \cdots \\
&+ f_{1-a_1\dots 1-a_{m-1}a_m:1-a_1\dots 1-a_m} \\
+ f_{1-a_11-a_2\dots 1-a_m:a_11-a_2\dots 1-a_m} &+ f_{a_11-a_2\dots 1-a_{m-1}1-a_m:a_11-a_2\dots 1-a_{m-1}a_m} + \cdots \\
&+ f_{a_11-a_2a_3\dots a_m:a_1a_2a_3\dots a_m} = 0.
\end{aligned}
$$

We then put together two pairs of allocations if the allocations in both pairs differ on the same position (for example the pairs $111 : 011$ and $001 : 101$ differ both on the first task so $f_{111:011}$ and $f_{001:101}$ would make a pair) and applying the rule $f_{a:b} = -f_{b:a}$ for a, b that differ only in one position, we get

$$
\begin{aligned}
d^1_{a:1-a} &= f_{1-a_11-a_2a_3\dots a_m:1-a_1a_2a_3\dots a_m} + f_{a_1a_2a_3\dots a_m:a_11-a_2a_3\dots a_m} + \cdots \\
&\quad + f_{1-a_1\dots 1-a_m:1-a_1\dots 1-a_{m-1}a_m} + f_{a_11-a_2\dots 1-a_{m-1}a_m:a_11-a_2\dots 1-a_{m-1}1-a_m} \\
&= (-1)^{a_1+a_2}c_{1,2|a_{-\{1,2\}}} + (-1)^{a_1+a_3}c_{1,3|1-a_2a_{-\{1,2,3\}}} + \dots + (-1)^{a_1+a_m}c_{1,m|1-a_{-\{1,m\}}}.
\end{aligned}
$$

The technique for computing $d^i_{a:1-a}$ is analogous. We use the following cycle: We first change the allocation of task i, we then change in $m-1$ steps the allocation of the remaining $m-1$ tasks in ascending order (that is the sequence of indices of the tasks we change is increasing) until we have changed the allocation of all tasks. After this it it time to change the allocation of task i for the second time. The last step is to reverse the sequence of the $m-1$ remaining indices and change the allocation of the corresponding tasks successively according to the reversed sequence. □

4 Characterization of 3-Dimensional mechanisms

4.1 Calculating the distances

We believe that the tools we have developed in the preceding section are useful for the study of the allocation graph for an arbitrary number of tasks m. We demonstrate this by using them in order to determine the allocation graphs and the corresponding geometrical shapes a truthful mechanism can take for the case $m = 3$.

For the case of 3 tasks we will apply Lemmas 7 and 6 in order to compute the distances $d^i_{a:1-a}$ with respect to the constants $c_{i,j|a_{-\{i,j\}}}$. For simplicity of notation we will write d^j instead of $d^j_{111:000}$, for $j = 1, 2, 3$ and it turns out that all other distances $d^j_{a:b}$, between regions R_a and R_b, can be expressed using the three distances d^1, d^2, d^3 between regions R_{111} and R_{000}. We define the constant e as

$$e = c_{12|0} - c_{12|1} = c_{13|0} - c_{13|1} = c_{23|0} - c_{23|1}. \tag{3}$$

then $c_{12|0} = c_{12|1} + e$ and we can rewrite the equalities in the following way:

$$\begin{aligned}
d^1_{111:000} &= c_{13|1} + c_{12|0} = c_{13|1} + c_{12|1} + e = c_{13|0} + c_{12|0} - e = d^1 \\
d^2_{111:000} &= c_{12|1} + c_{23|0} = c_{12|1} + c_{23|1} + e = c_{12|0} + c_{23|0} - e = d^2 \\
d^3_{111:000} &= c_{13|1} + c_{23|0} = c_{13|1} + c_{23|1} + e = c_{13|0} + c_{23|0} - e = d^3
\end{aligned}$$

$$\begin{aligned}
d^1_{011:100} &= -d^1 \\
d^2_{011:100} &= -c_{12|1} + c_{23|1} = d^3 - d^1 \\
d^3_{011:100} &= -c_{13|1} + c_{23|1} = d^2 - d^1
\end{aligned}$$

$$\begin{aligned}
d^1_{101:010} &= -c_{12|1} + c_{13|1} = d^3 - d^2 \\
d^2_{101:010} &= -d^2 \\
d^3_{101:010} &= = -(d^2 - d^1)
\end{aligned}$$

$$\begin{aligned}
d^1_{110:001} &= -(d^3 - d^2) \\
d^2_{110:001} &= -(d^3 - d^1) \\
d^3_{110:001} &= -d^3
\end{aligned}$$

4.2 Properties satisfied by the allocation graph

Lemma 8. *There always exist two regions R_a, R_b in $\mathrm{Hd} = 3$ such that $d^i_{a:b} \geq 0$ for $i = 1, 2, 3$.*

Proof. Suppose towards a contradiction that the statement of the lemma is not true. Then R_{111}, R_{000} do not share a common boundary. There are three cases: either d^1, d^2, d^3 are all negative, or two of them are negative or one of them is negative. Suppose that $d^1 \leq d^2 \leq d^3$ (the proof for any other relative ranking of the three distances is similar we would just find a different pair of intersecting boxes). Then the three cases are: $d^1 \leq d^2 \leq d^3 < 0$ or $d^1 \leq d^2 < 0 \leq d^3$ or $d^1 < 0 \leq d^2 \leq d^3$. In any of the cases we have $d^1_{011:100} \geq 0, d^2_{011:100} \geq 0, d^3_{011:100} \geq 0$ and consequently regions R_{011}, R_{100} intersect. □

Remark 2. *In what follows we will make the assumption that this pair of regions R_a, R_b in $\mathrm{Hd} = 3$ such that $d^i_{a:b} \geq 0$ for $i = 1, 2, 3$, guaranteed to exist by Lemma 8 are R_{111} and R_{000}.*

For any mechanism we present here you can get another truthful mechanism by applying the following rotations: Think of the mechanism as a partition of the cube, if you rotate one of the possible partitions so that the faces of the cube go to faces of the cube after the rotation (and the center of axes goes to another vertex of the cube), you also get a truthful mechanism. The reason is that the slope of the separating hyperplane between two regions only depends on their Hamming Distance, i.e. on the number of tasks on which they differ. The characteristic of the rotation we described is that it respects the Hamming distances.

Lemma 9. *If R_{111} and R_{000} intersect then*

a) if $e < 0$ then at least two of the constants $c_{12|1}, c_{13|1}, c_{23|1}$ are strictly positive,

b) if $e > 0$ then at least two of the constants $c_{12|0}, c_{13|0}, c_{23|0}$ are strictly positive.

Proof. We will deal with the case $e < 0$ (the other case is very similar). Observe the second expression for d^i. Since R_{111} and R_{000} share a common boundary we should have $d^i > 0$ for $i = 1, 2, 3$. Each one of the constants $c_{12|1}, c_{13|1}, c_{23|1}$ appears exactly in two of the three distances and $e < 0$. Suppose towards a contradiction that two of the constants were negative, then at least one of the distances d^i would be negative, contradiction. □

Lemma 10. *If a pair of regions R_a, R_{1-a} share a common Hd-3 boundary then no other pair R_b, R_{1-b} of regions share a common Hd-3 boundary.*

Proof. Suppose for example that R_{111}, R_{000} share a common Hd-3 boundary then this boundary is a hyperplane of the form $t_{11}+t_{12}+t_{13}$ =constant. Consequently the boxes that contain the regions intersect and more specifically all three distances d^1, d^2, d^3 are positive. But as by Lemma 6 $d^1_{011:100} = -d^1$, we have $d^1_{011:100} < 0$ and consequently R_{011}, R_{100} do not intersect on axis t_1 and thus cannot share a common Hd-3 boundary. Similarly as $d^2_{101:010} = -d^2$ regions R_{101}, R_{010} do not intersect on axis t_2 and thus cannot share a common Hd-3 boundary and so on. □

Lemma 11. *If R_{111} and R_{000} share a common boundary and $c_{12|1} > 0, c_{13|1} < 0, c_{23|1} > 0, e < 0$ then we also have $c_{12|0} > 0, c_{13|0} < 0, c_{23|0} > 0$.*

The proof is very easy and we include it in appendix A.

4.3 All possible Mechanisms

Definition 8. *A degenerate version of a mechanism M is a mechanism for which some of the constants $c_{ij|0}, c_{ij|1}, d^k_{a:b}$, for some $i, j, k \in \{1, 2, 3\}$ and some allocations a, b, become 0, while all other such constants retain the same sign as in the non-degenerate mechanism.*

We will describe the possible shapes of the mechanism when a Hd-3 boundary exists and thanks to Lemma 8 any other mechanism is a degenerate version of a mechanism with a Hd-3 boundary. Summarizing all restrictions to the shape of the mechanism we obtained in the previous section we get the following characterization:

Theorem 1. *The possible truthful mechanisms are the following five possible partitions of the space and all their rotations. (In Figure 2 you can see their geometrical shapes.)*

As for any mechanism we give here we also include in our characterization all its rotations, we suppose without loss of generality that R_{111}, R_{000} share a common boundary, that $e < 0$ and that the two constants guaranteed to be positive by Lemma 9 are $c_{12|1} > 0, c_{23|1} > 0$.

1. $c_{12|1} > 0, c_{13|1} > 0, c_{23|1} > 0, \quad c_{12|0} > 0, c_{13|0} > 0, c_{23|0} > 0$
2. $c_{12|1} > 0, c_{13|1} > 0, c_{23|1} > 0, \quad c_{12|0} < 0, c_{13|0} < 0, c_{23|0} < 0$
3. $c_{12|1} > 0, c_{13|1} > 0, c_{23|1} > 0, \quad c_{12|0} < 0, c_{13|0} < 0, c_{23|0} > 0$
4. $c_{12|1} > 0, c_{13|1} > 0, c_{23|1} > 0, \quad c_{12|0} > 0, c_{13|0} < 0, c_{23|0} > 0$
5. $c_{12|1} > 0, c_{13|1} < 0, c_{23|1} > 0, \quad c_{12|0} > 0, c_{13|0} < 0, c_{23|0} > 0.$

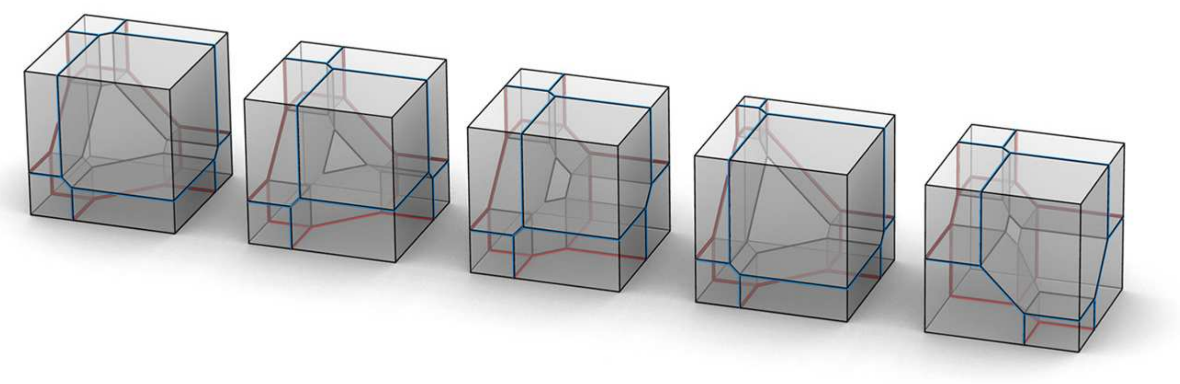

Figure 2: 3D models of the possible partitions (up to rotation). Looking just at the blue projections you can determine the constants $c_{ij|0}$ and from the red projections the constants $c_{ij|1}$.

4.4 Knowing a few distances we can draw the whole mechanism

Now we can fully describe the boxes that contain different regions of the mechanism. Let for simplicity in notation $f_1 := f_{111:011}, f_2 := f_{111:101}, f_3 := f_{111:110}$. In fact given four of the constants $c_{ij|a}$ as input it is easy to find the exact shape of the mechanism thanks to the following relations

$$\begin{aligned}
F_{111} &:= (f_{111:011}, f_{111:101}, f_{111:110}) = (f_1, f_2, f_3) \\
F_{000} &:= (f_1 - d^1, f_2 - d^2, f_3 - d^3) \\
F_{011} &:= (f_1, f_2 - c_{12|1}, f_3 - c_{13|1}) \\
F_{101} &:= (f_1 - c_{12|1}, f_2, f_3 - c_{23|1}) \\
F_{110} &:= (f_1 - c_{13|1}, f_2 - c_{23|1}, f_3) \\
F_{100} &:= (f_1 - d^1, f_2 - c_{23|1}, f_3 - c_{23|1}) \\
F_{010} &:= (f_1 - c_{13|1}, f_2 - d^2, f_3 - c_{13|1}) \\
F_{001} &:= (f_1 - c_{12|1}, f_2 - c_{12|1}, f_3 - d^3).
\end{aligned}$$

Having found a general formula for each one of these boxes we can also describe an algorithm for constructing the geometrical shape of the mechanism. The input is four of the constants $c_{ij|a}$ (knowing these we can compute the other two constants) and the output is the exact shape of the mechanism. We first construct all boxes. Wherever two or more boxes intersect we have to divide the points in the intersection between the regions corresponding to the intersecting boxes. At this point we have to be cautious, because if we are given two intersecting boxes in Hd $= 3$ then there are infinitely many possible separating hyperplanes between them. However if we first consider the intersection between Hd $= 2$ boxes there is a single hyperplane that can separate the two regions. Then consider the intersecting boxes with allocations in Hd $= 3$ (there is at most one such pair as we will see in lemma 10). As we have already constructed the Hd ≤ 2 boundaries there is a single possible Hd-3 separating hyperplane. It is not hard to implement this algorithm and given four of the constants $c_{ij|a}$

of the mechanisms we can fully determine the allocation graph and the geometrical shape of the mechanism. We next proceed to explore which are the possible geometrical shapes of the mechanism, which depend on the signs of the distances.

5 Lower bounds for some Scheduling Mechanisms

Observing the figures we got from our characterization we see that many of the regions have the shape of a box, for some of these cases the region that has the shape of the box is $R_{1\ldots1}$. Threshold (/additive) mechanisms [7, 15] are the special case of these mechanisms, when all regions are boxes. Even though these mechanisms are much more general, we can still show the same lower bound of $1+\sqrt{n}$ using an argument very simmilar to the one used in [7]. For these cases we can prove a lower bound of $1+\sqrt{n}$ (while the lower bound for the general case is still a small constant). (We include the proof in the appendix.)

Theorem 2. *Every mechanism for which $R_{1\ldots1}$ is a box has approximation ratio at least $1+\sqrt{n}$.*

Finally there is a non-trivial geometrically defined class of mechanisms for which we can provide an n lower bound.

Definition 9. *We will say that a mechanism is* non-penalizing *if in the allocation graph no pair of regions of the form R_{a10}, R_{b01}, where a, b are $(m-2)$-dimensional allocation vectors, share a common boundary.*

The first mechanism in Figure 2 is an example of such a mechanism. The intuition behind these mechanisms is that, if for fixed values of the other players, a player lowers one of his values he only gets more tasks (regardless of his initial allocation for the tasks he lowers), in other words a machine never loses a job, just because it becomes faster for another job.

Theorem 3. *Every non-penalizing mechanism has approximation ratio at least n.*

6 Concluding Remarks and open problems

Our characterization is only for the case of 3 tasks, the tools we have developed to obtain this characterization are however for the general case of m tasks. Can we find a succinct way to describe all possible allocation graphs for the general case?

We would like to stress the connection of our results with the scheduling unrelated machines problem. The lower bounds in the last section show that many mechanisms have bad approximation ratio just because of the geometrical shape of their projections. Finally we believe that the characterization for the case of three tasks can be used to improve the existing [11] lower bound of 2.465 for the case of 4 machines to a better constant.

Acknowledgements

I would like to thank Christos Athanasiadis, Ioannis Emiris and Elias Koutsoupias for helpful discussions and my brother Aris Vidalis for making the nice shaded 3D models I include in this paper (and also for bothering to imagine these complicated partitions of the space!).

References

[1] Aaron Archer and Robert Kleinberg. Truthful germs are contagious: A local to global characterization of truthfulness. In *ACM Conference on Electronic Commerce (EC)*, 2008.

[2] Aaron Archer and Éva Tardos. Truthful mechanisms for one-parameter agents. In *42nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 482–491, 2001.

[3] Itai Ashlagi, Shahar Dobzinski, and Ron Lavi. An optimal lower bound for anonymous scheduling mechanisms. In *ACM Conference on Electronic Commerce (EC)*, 2009.

[4] André Berger, Rudolf Müller, and Seyed Hossein Naeemi. Characterizing incentive compatibility for convex valuations. In *to appear in SAGT*, 2009.

[5] Sushil Bikhchandani, Shurojit Chatterji, Ron Lavi, Ahuva Mu'alem, Noam Nisan, and Arunava Sen. Weak monotonicity characterizes deterministic dominant-strategy implementation. *Econometrica*, 74(4):1109–1132, 2006.

[6] George Christodoulou, Elias Koutsoupias, and Angelina Vidali. A lower bound for scheduling mechanisms. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1163–1169, 2007.

[7] George Christodoulou, Elias Koutsoupias, and Angelina Vidali. A characterization of 2-player mechanisms for scheduling. In *Algorithms - ESA, 16th Annual European Symposium*, pages 297–307, 2008.

[8] George Christodoulou, Elias Koutsoupias, and Angelina Vidali. A lower bound for scheduling mechanisms. *Algorithmica*, 2008.

[9] Shahar Dobzinski and Mukund Sundararajan. On characterizations of truthful mechanisms for combinatorial auctions and scheduling. In *EC*, 2008.

[10] Hongwei Gui, Rudolf Müller, and Rakesh V. Vohra. Dominant strategy mechanisms with multidimensional types. In *Computing and Markets*, 2005.

[11] Elias Koutsoupias and Angelina Vidali. A lower bound of $1+\phi$ for truthful scheduling mechanisms. In *MFCS*, pages 454–464, 2007.

[12] Ron Lavi and Chaitanya Swamy. Truthful mechanism design for multi-dimensional scheduling via cycle monotonicity. In *ACM Conference on Electronic Commerce (EC)*, pages 252–261, 2007.

[13] Dov Monderer. Monotonicity and implementability. In *ACM Conference on Electronic Commerce (EC)*, 2008.

[14] Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[15] Noam Nisan and Amir Ronen. Algorithmic mechanism design (extended abstract). In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing (STOC)*, pages 129–140, 1999.

[16] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[17] Jean-Charles Rochet. A necessary and sufficient condition for rationalizability in a quasi-linear context. *Journal of Mathematical Economics*, 16:191–200, 1987.

[18] Michael E. Saks and Lan Yu. Weak monotonicity suffices for truthfulness on convex domains. In *EC*, pages 286–293, 2005.

A Missing proofs

Proof of Lemma 11. Since R_{111} and R_{000} share a common boundary $d^1 = c_{13|1} + c_{12|0} > 0$ consequently $c_{12|0} > -c_{13|1} > 0$. As $e = c_{13|0} - c_{13|1} < 0$ we have $c_{13|0} < c_{13|1} < 0$. Finally $d^2 = c_{13|1} + c_{23|0} > 0$ and consequently $c_{23|0} > -c_{13|1} > 0$. □

Proof of Lemma 7. For simplicity we will give the details of the proof for $i = 1$ and m tasks, that is if we want to compute d^1. We take the path that at each step changes the allocation of a single task in the following order: $(1, 2, 3 \ldots, m, 1, m, m-1, \ldots, 2)$. The path is

$$\begin{aligned} & f_{a_1 a_2 \ldots a_m : 1-a_1 a_2 \ldots a_m} + f_{1-a_1 a_2 a_3 \ldots a_m : 1-a_1 1-a_2 a_3 \ldots a_m} + \cdots \\ & \qquad + f_{1-a_1 \ldots 1-a_{m-1} a_m : 1-a_1 \ldots 1-a_m} \\ & + f_{1-a_1 1-a_2 \ldots 1-a_m : a_1 1-a_2 \ldots 1-a_m} + f_{a_1 1-a_2 \ldots 1-a_{m-1} 1-a_m : a_1 1-a_2 \ldots 1-a_{m-1} a_m} + \cdots \\ & \qquad + f_{a_1 1-a_2 a_3 \ldots a_m : a_1 a_2 a_3 \ldots a_m} = 0. \end{aligned}$$

We then put together two pairs of allocations if the allocations in both pairs differ on the same position (for example the pairs 111 : 011 and 001 : 101 differ both on the first task so $f_{111:011}$ and $f_{001:101}$ would make a pair) and applying the rule $f_{a:b} = -f_{b:a}$ for a, b that differ only in one position, we get

$$\begin{aligned} d^1_{a:1-a} &= f_{1-a_1 1-a_2 a_3 \ldots a_m : 1-a_1 a_2 a_3 \ldots a_m} + f_{a_1 a_2 a_3 \ldots a_m : a_1 1-a_2 a_3 \ldots a_m} + \cdots \\ & \quad + f_{1-a_1 \ldots 1-a_m : 1-a_1 \ldots 1-a_{m-1} a_m} + f_{a_1 1-a_2 \ldots 1-a_{m-1} a_m : a_1 1-a_2 \ldots 1-a_{m-1} 1-a_m} \\ &= (-1)^{a_1+a_2} c_{1,2|a_{-\{1,2\}}} + (-1)^{a_1+a_3} c_{1,3|1-a_2 a_{-\{1,2,3\}}} + \cdots + (-1)^{a_1+a_m} c_{1,m|1-a_{-\{1,m\}}}. \end{aligned}$$

The technique for computing $d^i_{a:1-a}$ is analogous. We use the following cycle: We first change the allocation of task i, we then change in $m-1$ steps the allocation of the remaining $m-1$ tasks in ascending order (that is the sequence of indices of the tasks we change is increasing) until we have changed the allocation of all tasks. After this it it time to change the allocation of task i for the second time. The last step is to reverse the sequence of the $m-1$ remaining indices and change the allocation of the corresponding tasks successively according to the reversed sequence. □

Proof of Lemma 6. Directly from the definition we have $d^i_{a:b} = f_{1-a_i, a_{-i}:a} + f_{1-b_i, b_{-i}:b}$ and $d^i_{1-a_i, a_{-i}:1-b_i, b_{-i}} = f_{a:1-a_i, a_{-i}} + f_{b:1-b_i, b_{-i}}$. Consequently $d^i_{a:b} = -d^i_{1-a_i, a_{-i}:1-b_i, b_{-i}}$. □

Proof of Lemma 5. Consider the projections of the mechanism for two tasks i, j when the processing times for the rest of the tasks are fixed. Then all other terms of the monotonicity property vanish except for the terms corresponding to tasks i, j. Consequently for fixed values of the other players the mechanism should have one of the two shapes in Figure 1.

Suppose that the boxes B_a, B_b corresponding to regions R_a, R_b intersect. Say that a, b differ in the allocation of tasks i, j (possibly also in the allocation of other tasks). Then taking the projection for fixed $t_{-\{i,j\}}$ it is obvious that the two regions share a common boundary. $\square$

Proof of Theorem 2. We will just show that it has ratio at least $\sqrt{n}$. By adding some additional dummy tasks (like in the proof of the $1+\sqrt{2}$ lower bound [6]) it is an easy exercise to improve this to a lower bound of $1+\sqrt{n}$.

Consider the following two $n \times n$ matrices of processing times

$$\begin{pmatrix} b\star & \dots & b\star \\ 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{pmatrix} \qquad \begin{pmatrix} \epsilon\star & \dots & b+\epsilon & \dots & \epsilon\star \\ 1 & \dots & 1\star & \dots & 1 \\ \vdots & \dots & 1 & \ddots & \vdots \\ 1 & \dots & 1 & \dots & 1 \end{pmatrix}.$$

There exists some b such that the first player gets all tasks, so that the allocation in the first instance is the one indicated by the stars. If there exists some $b \geq 1$ then the mechanism has approximation n and we are done. Suppose that $0 < b < 1$ and choose b to be the supremum of all these values, the approximation ratio is $n \cdot b$.

There exists a task j such that $f_{1\dots1:a} = b$ where a differs from $(1, \dots, 1)$ only in position j. Now consider the second matrix of processing times, since $f_{1\dots1:a} = b$ and since $R_{1\dots1}$ has the shape of a box, the allocation is the one indicated by the stars and the approximation ratio is $\frac{1}{b}$. The solution of the equation $n \cdot b = \frac{1}{b}$ is $\sqrt{n}$. $\square$

Proof. For a better understanding we will give the proof of a lower bound of 3 for the case of 3 tasks. Exactly the same technique gives a lower bound of n for $n^2 - n + 1$ tasks and n players as this number of tasks guarantees that one of the players will get at least n tasks.

We start with the instance

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

where we can assume w.l.o.g. that player 1 gets at least 3 tasks. The idea is to lower all values of player 1 to some small $\epsilon > 0$ except for 3 (and in the general case n) values. If we set a task that gets allocated to player 1 to ϵ, then by the Monotonicity Property the allocation of player 1 remains the same. We then lower one by one all of the tasks that are not assigned to player 1. Player 1 doesn't loose any of the tasks initially assigned to him (he might however get more tasks than those initially assigned to him), because the mechanism is non-penalizing mechanism. We continue until all tasks of player 1 except for the 3 that were initially assigned to him are zero. The approximation ratio is 3.

Example 1. *For example if the original assignment is the one marked by the stars we get:*

$$\begin{pmatrix} 1\star & 1\star & 1\star & 1 & 1 & 1 & 1\star \\ 1 & 1 & 1 & 1\star & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1\star & 1\star & 1 \end{pmatrix} \to \begin{pmatrix} 1\star & 1\star & 1\star & 0 & 0 & 0 & 0\star \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

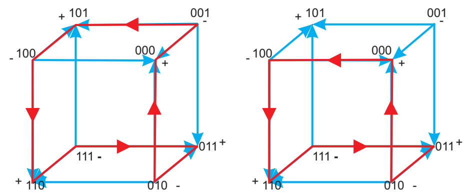

Figure 3: The first path gives the equations: $c_{1,2|1} - c_{1,2|0} = c_{1,3|1} - c_{1,3|0}$ and $c_{1,2|1} - c_{1,2|0} = c_{2,3|1} - c_{2,3|0}$. The second path gives the expression for $d^1_{111:000}$.

The tasked whose allocation we do not indicate can be allocated to anyone of the players, but the approximation ratio we get is at least n for any of these allocations.

Exactly the same technique gives a lower bound of n for $n^2 - n + 1$ tasks and n players as this number of tasks guarantees that one of the players will get at least n tasks. □

B Some more figures

We have already presented all possible mechanisms, but in this section we will present some of their degenerate versions just in order to make more plausible the notion degeneracy.

Remark 3. *Actually in the degenerate cases the allocation graph has some additional edges, which are not depicted in Figure 5, edges between regions that do not share a full-dimensional boundary. It is very easy to figure out from the geometrical shapes which are these edges but we do not depict them in order to keep the figure easier to understand and more relevant to the geometrical shape.*

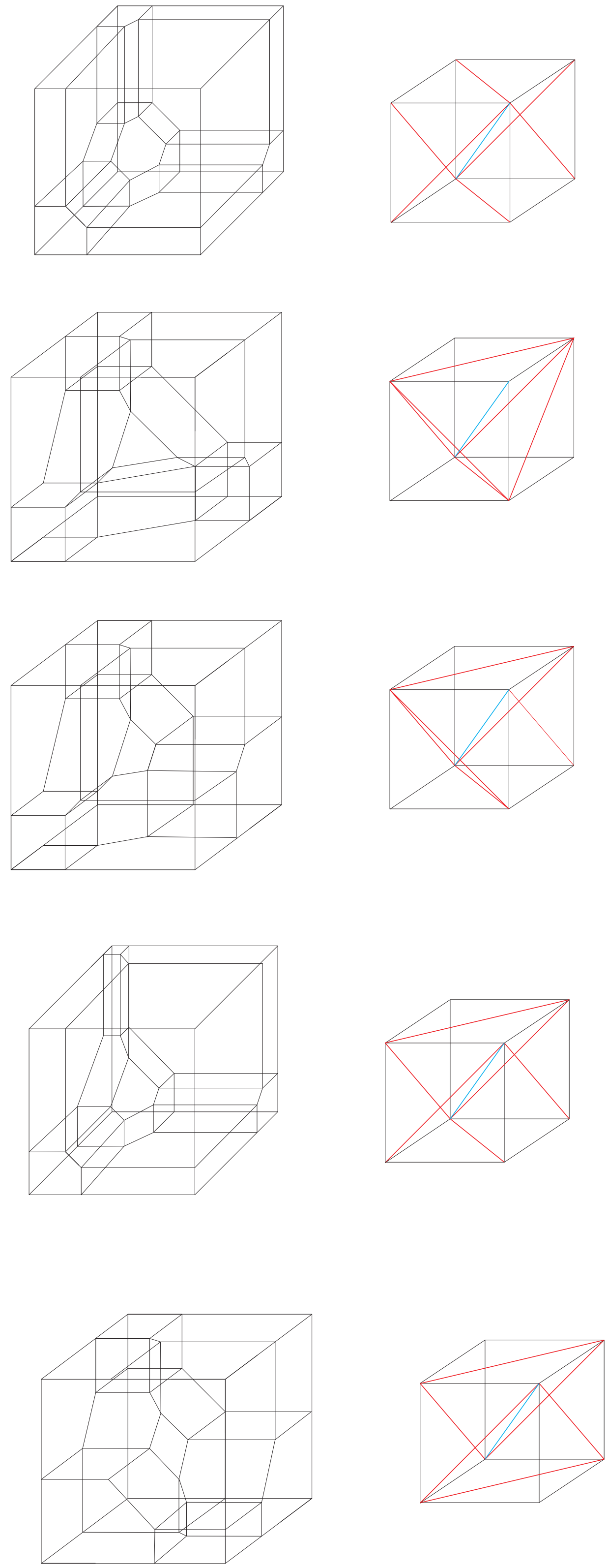

Figure 4: Possible (up to rotation) mechanisms with a Hd-3 boundary and all constants $c_{ij|0}, c_{ij|1}, d^k_{a:b} \neq 0$ and the corresponding allocation graph.

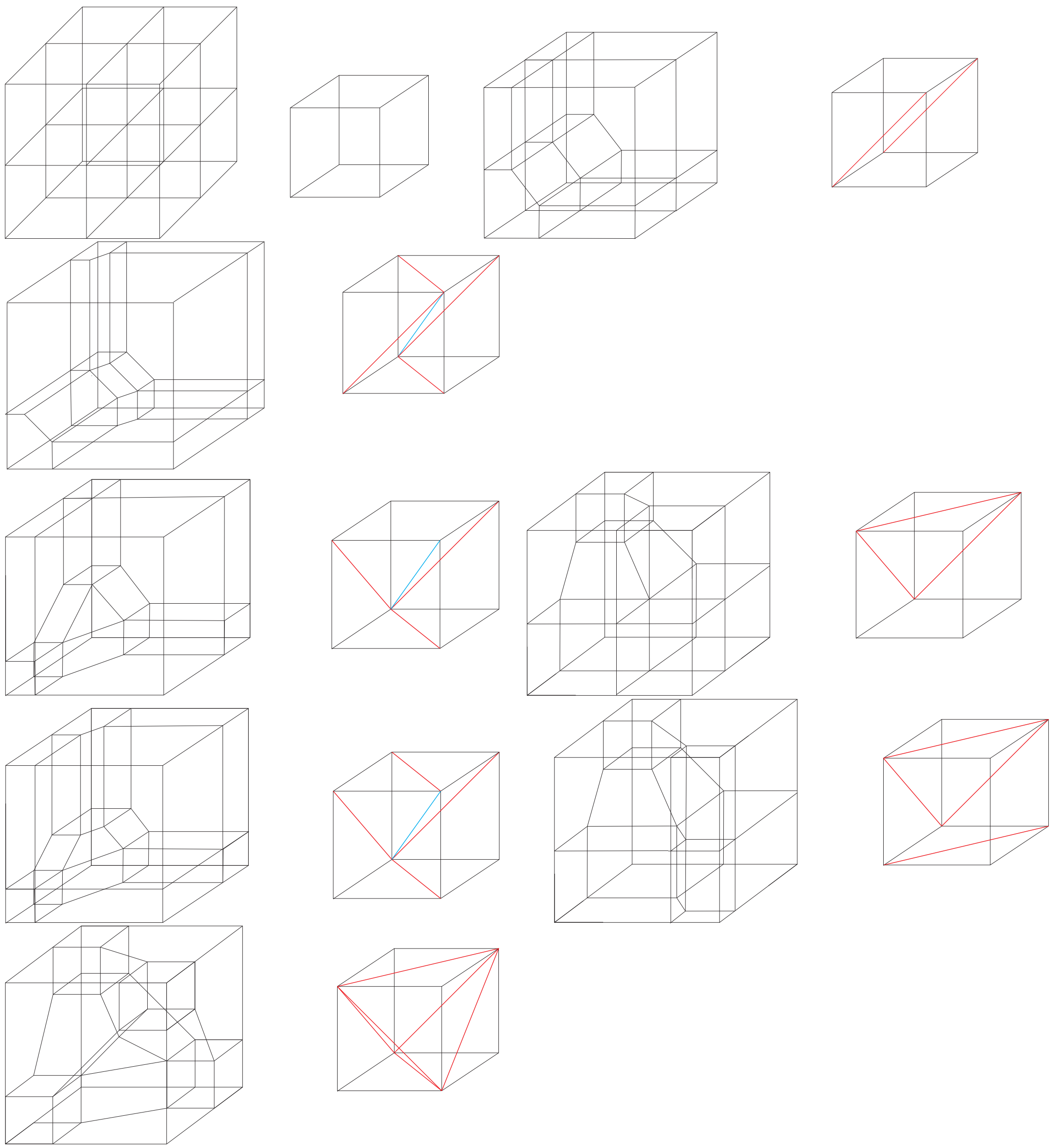

Figure 5: Some degenerate mechanisms and their corresponding allocation graph. The Hd-1 boundaries are the edges of the cube. (The edges on the allocation graph corresponding to Hd-2 boundaries are red and the ones corresponding to Hd-3 boundaries blue.)