Finding and verifying the nucleolus of cooperative games – technical report

Márton Benedek* Jörg Fliege† Tri-Dung Nguyen‡

Received: date / Accepted: date

This technical report serves as an e-companion for [1]. The notations used in this report is in line with [1].

A Geometric view of the nucleolus

We start by using a simple three-player cooperative game to introduce the main concepts and review the related literature.

Example 1. *Consider the 3-player game v with coalition values $v(\{1\}) = 1$, $v(\{2\}) = 2$, $v(\{3\}) = 5$, $v(\{1,2\}) = 6$, $v(\{1,3\}) = 7$, $v(\{2,3\}) = 8$, and $v(\mathcal{N}) = 12$. The set of all imputations is $\mathbf{I} = \{(x_1, x_2, x_3) \ : \ x_1 + x_2 + x_3 = 12,\ x_1 \geq 1,\ x_2 \geq 2,\ x_3 \geq 5\}$, while the core of the game is*

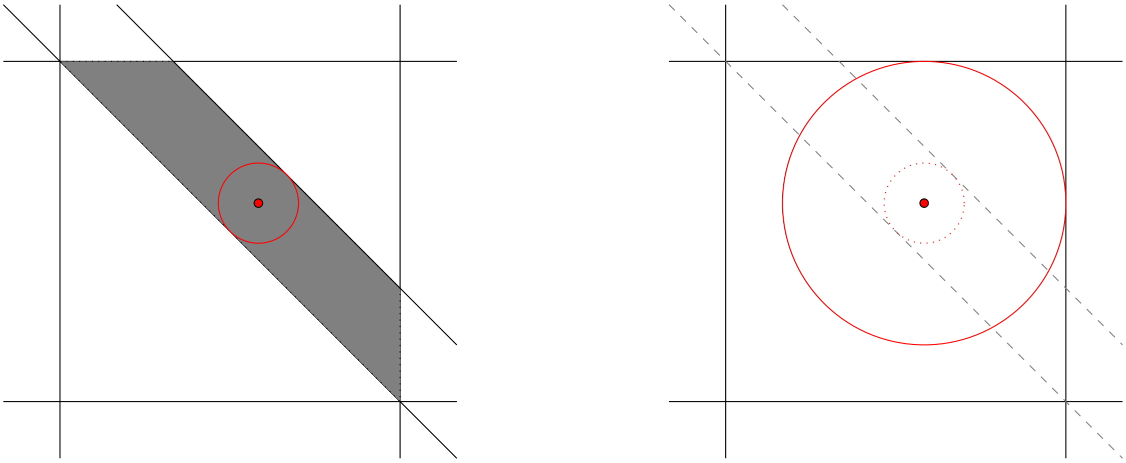
$$\begin{aligned}\mathbf{Co} = \{(x_1, x_2, x_3) : \quad & x_1 + x_2 + x_3 = 12,\ x_1 \geq 1,\ x_2 \geq 2,\ x_3 \geq 5,\\ & x_1 + x_2 \geq 6,\ x_1 + x_3 \geq 7,\ x_2 + x_3 \geq 8\}.\end{aligned}$$

The least core is the line segment connecting $\mathbf{x} = (2, 4.5, 5.5)$ and $\mathbf{y} = (3.5, 3, 5.5)$. The nucleolus is $\nu = (2.75, 3.75, 5.5)$.

*Institute of Economics, Centre for Economic and Regional Studies, Hungarian Academy of Sciences, Tóth Kálmán u. 4., Budapest, 1097, Hungary; Corvinus University of Budapest, Fővám tér 8., Budapest, 1093, Hungary; Budapest University of Technology and Economics, Egry József u. 1., Budapest, 1111, Hungary; ORCID: 0000-0002-7492-1174, benedek.marton@krtk.mta.hu

†Mathematical Sciences, University of Southampton, University Road, Southampton, SO17 1BJ, United Kingdom, J.Fliege@soton.ac.uk.

‡Mathematical Sciences, Business School and CORMSIS, University of Southampton, Southampton, SO17 1BJ, United Kingdom, T.D.Nguyen@soton.ac.uk.



(a) Starting level

(b) Subsequent level

Figure 1: Geometric view of the nucleolus

Following the footsteps of Maschler et al. [2], one can formulate the following geometric property of the nucleolus. For the purpose of convenient visualisation, let us consider the game in Example 1, which has a non-empty core. Then from any point $\mathbf{x}$ within the core of the game, the distances to the hyperplanes forming the core are the levels of satisfactions the corresponding coalitions have over the proposed outcome $\mathbf{x}$. The least core are those points in the core that minimize the maximum level of dissatisfaction, or equivalently, maximize the minimum level of satisfaction. The least core, therefore, contains the centre of the largest balls that we can fit into the core. The corresponding largest radius corresponds to the minimum level of satisfaction.

This geometrical property is captured on Figure 1, where the outcome space $\mathbf{x} \in \mathbb{R}^3$ is projected to the hyperplane $\mathbf{x}(\mathcal{N}) = v(\mathcal{N})$, thereby the image is in a 2-dimensional space. In Figure 1a, the 2D grey trapezoid is the projected core, and we find that we cannot increase the ball fitted into it around the point of the nucleolus because of the two parallel tight constraints. This ball corresponds to the largest ball fitting into the core in the original outcome space, and the tight constraints that avoid the ball from getting larger form a set of 'balanced' coalitions, a concept formally defined in Section 2.1 of [1].

Among the least core outcomes, we want to minimize the second level of maximal dissatisfaction. This is equivalent to finding the largest ball within the new (and larger) polytope that is derived from the core after having removed a balanced set of tight constraints found in the previous step. If we repeat this process, we will finally arrive at the point that lexicographically minimizes the dissatisfactions among all the coalitions. That is precisely the nucleolus.

In the example, if we relax the tight core constraints that are limiting the size of the ball in the first iteration, then we arrive at a new projected polytope as shown in Figure 1b. At this point, if we consider only the balls whose center belongs to the least core, the center of the largest one among these is the nucleolus.

The problem of finding the largest ball within a polytope can be formulated as a linear program. Therefore, the nucleolus can be found by successively solving a nested sequence of linear programs. This is indeed the main underlying framework for the linear programming approach appearing in [2, 3, 4, 5, 6, 7].

To demonstrate the original Kohlberg criterion (Algorithm 1 of [1]), we consider the simple three-player cooperative game of Example 1. At the nucleolus ν, we find $T_0(\nu) = \emptyset$, $T_1(\nu) = \{\{1,2\},\{3\}\}$, $T_2(\nu) = \{\{1,3\},\{2,3\}\}$, $\epsilon_1(\nu) = -0.5$ and $\epsilon_2(\nu) = -1.25$. Here, $T_1(\nu)$ is $T_0(\nu)$-balanced with weight vector $\omega = (1,1)$. Similarly, $(T_1(\nu) \cup T_2(\nu))$ is $T_0(\nu)$-balanced with weight vector $\omega = (3/4, 1/2, 1/4, 1/4)$. This means $(T_0(\nu), T_1(\nu), T_2(\nu))$ has property II and hence ν is verified as the nucleolus. We can also verify that the Kohlberg criterion does not hold for any $\mathbf{x}' \neq \nu$. For example, let $\mathbf{x}' = 1/2(\mathbf{x} + \nu)$. Then $T_1(\mathbf{x}') = \{\{1,2\}\},\{3\}\}$ and $T_2(\mathbf{x}') = \{\{1,3\}\}$. Although $T_1(\mathbf{x}')$ is $T_0(\mathbf{x}')$-balanced, $(T_1(\mathbf{x}') \cup T_2(\mathbf{x}'))$ is not and this verifies that $\mathbf{x}'$ is not the nucleolus.

B Reducing the sizes of the tight sets

When checking the Kohlberg criterion we might end up having to store an exponentially large number of coalitions. The practical efficiency of checking T_0-balancedness depends entirely on the size of the tight sets we encounter. For these purposes, it is of particular interest to find compact representations of large tight sets.

First, we provide a method for reducing the size of H_k to at most $n(n-1)$. This is achieved by replacing tight sets with their compact representations. We start by showing some further balancedness properties.

Proposition 1. *The following results hold:*

(a) If $T \subseteq 2^{\mathcal{N}}$ is a nonempty T_0-balanced set then there exists $R \subseteq T$ with $1 \leq |R| = \operatorname{rank}(R) \leq \operatorname{rank}(T)$ that is T_0-balanced.

(b) Let there be nonempty collections $P, Q \subseteq 2^{\mathcal{N}}$ with $Q \cup P$ being a T_0-balanced set. Then there exists a subset $R \subseteq Q$ with $1 \leq |R| = \operatorname{rank}(R) \leq \operatorname{rank}(Q)$ such that $R \cup P$ is T_0-balanced.

Proof.

(a) Given that T is T_0-balanced, there exists $\gamma \in \mathbb{R}^{|T_0|}_{\geq 0}$ and $\omega \in \mathbb{R}^{|T|}_{>0}$ such that

$$\mathbf{e}(\mathcal{N}) = \sum_{\mathcal{S} \in T_0} \gamma_{\mathcal{S}} \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S} \in T} \omega_{\mathcal{S}} \mathbf{e}(\mathcal{S}).$$

Thus,

$$0 \neq \mathbf{e}(\mathcal{N}) - \sum_{\mathcal{S} \in T_0} \gamma_{\mathcal{S}} \mathbf{e}(\mathcal{S}) = \sum_{\mathcal{S} \in T} \omega_{\mathcal{S}} \mathbf{e}(\mathcal{S}),$$

i.e., $\big(\mathbf{e}(\mathcal{N}) - \sum_{\mathcal{S} \in T_0} \gamma_{\mathcal{S}} \mathbf{e}(\mathcal{S})\big)$ belongs to the convex combination of $\{\mathbf{e}(\mathcal{S})\}_{\mathcal{S} \in T}$. Applying the Caratheodory theorem, there exists a subset $U \subseteq T$ with $\operatorname{rank}(U) = |U| = \operatorname{rank}(T)$ such that $\big(\mathbf{e}(\mathcal{N}) - \sum_{\mathcal{S} \in T_0} \gamma_{\mathcal{S}} \mathbf{e}(\mathcal{S})\big) = \sum_{\mathcal{S} \in U} \beta_{\mathcal{S}} \mathbf{e}(\mathcal{S})$. By removing coalitions with coefficient $\beta_{\mathcal{S}} = 0$, we obtain a subset $R \subseteq U \subseteq T$ with $\operatorname{rank}(R) \leq \operatorname{rank}(U)$ that is T_0-balanced. Note also that, since $\big(\mathbf{e}(\mathcal{N}) - \sum_{\mathcal{S} \in T_0} \gamma_{\mathcal{S}} \mathbf{e}(\mathcal{S})\big) \neq 0$, there exists at least one coalition $\mathcal{S}$ with $\beta_{\mathcal{S}} > 0$. Thus, $1 \leq \operatorname{rank}(R) \leq \operatorname{rank}(T)$ and R is T_0-balanced. In addition, since $\operatorname{rank}(U) = |U|$, we also have $\operatorname{rank}(R) = |R|$.

(b) Since $Q \cup P$ is T_0-balanced, there exist $\gamma \in \mathbb{R}^{|T_0|}_{\geq 0}$, $\alpha \in \mathbb{R}^{|P|}_{>0}$, and $\eta \in \mathbb{R}^{|Q|}_{>0}$ such that $\mathbf{e}(\mathcal{N}) = \sum_{\mathcal{S} \in T_0} \gamma_{\mathcal{S}} \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S} \in P} \alpha_{\mathcal{S}} \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S} \in Q} \eta_{\mathcal{S}} \mathbf{e}(\mathcal{S})$. Thus,

$$\mathbf{e}(\mathcal{N}) - \sum_{\mathcal{S} \in T_0} \gamma_{\mathcal{S}} \mathbf{e}(\mathcal{S}) - \sum_{\mathcal{S} \in P} \alpha_{\mathcal{S}} \mathbf{e}(\mathcal{S}) = \sum_{\mathcal{S} \in Q} \eta_{\mathcal{S}} \mathbf{e}(\mathcal{S}) \neq 0.$$

Using the same arguments as in part (a), there exists a subset $Q' \subseteq Q$ with $\operatorname{rank}(Q') = |Q'| = \operatorname{rank}(Q)$ such that

$$\mathbf{e}(\mathcal{N}) - \sum_{\mathcal{S} \in T_0} \gamma_{\mathcal{S}} \mathbf{e}(\mathcal{S}) - \sum_{\mathcal{S} \in P} \alpha_{\mathcal{S}} \mathbf{e}(\mathcal{S}) = \sum_{\mathcal{S} \in Q'} \beta_{\mathcal{S}} \mathbf{e}(\mathcal{S}).$$

By removing those coalitions $\mathcal{S} \in Q'$ with $\beta_{\mathcal{S}} = 0$, we obtain a non-empty subset $R \subseteq Q'$ such that $R \cup P$ is T_0-balanced and $1 \leq |R| = \operatorname{rank}(R) \leq \operatorname{rank}(Q)$.

□

We denote a subset R as in Proposition 1a as R^a and a subset R as in Proposition 1b as R^b.

Proposition 2. *The collection of tight coalitions in Algorithm 2 of [1], $\cup_{j=1}^{k} T_j$ can be represented by $\cup_{j=1}^{k} R_j^b$ of size at most $n(n-1)$.*

Proof. Since for each T_j, the set R_j^b can be constructed using Proposition 1b as a subset of another full row rank subset, its size is at most n. Since the number of iterations involved checking the Kohlberg criterion is at most $(n-1)$, the total size of $\cup_{j=1}^k R_j^b$ is at most $n(n-1)$. □

Note that, in accordance with Remark 2 of [1], Proposition 1 does not solve the problem of how to find a tight set of exponential size. In particular, it does not even touch on *how* to find a compact representative of such a tight set. To find such a representation, we need to identify among the exponential many vertices of a polytope the few relevant ones.

Algorithm 1: Improved Kohlberg Algorithm with Compact Representation.

Input: Game $(\mathcal{N}, v)$, imputation $\mathbf{x} \in \mathbf{I}$;

Output: Conclude if $\mathbf{x}$ is the nucleolus or not;

1. Initialization: Set $R_0 = \{\mathcal{N}\}$, $T_0 = \{\{i\} : x_i = v(\{i\})\,(i = 1, \ldots, n)\}$, and $k = 1$;

while $|R_{k-1}| < n$ **do**

  2. Find $T_k = \underset{\mathcal{S} \notin \text{span}(R_{k-1})}{\text{argmax}} \{v(\mathcal{S}) - \mathbf{x}(\mathcal{S})\}$;

  if $(\cup_{j=1}^k T_j)$ *is* T_0*-balanced* **then**

    3. Set $R_k = rep(T_k; R_{k-1})$, $k = k+1$ and continue;

  else

    4. Stop the algorithm and conclude that $\mathbf{x}$ is not the nucleolus.

  end

end

5. Conclude that $\mathbf{x}$ is the nucleolus.

Theorem 1. *The while-loop in Algorithm 1 terminates after at most $(n-1)$ iterations and it correctly decides whether an imputation is the nucleolus.*

The proof of Theorem 1 is provided in Appendix F. Note that after Step 2, when we check the condition of T_0-balancedness, while we write $(\cup_{j=1}^k T_j)$ being T_0-balanced, we can exploit Lemma 3b from [1] to use the representative sets R_k instead when performing the balancedness checking.

C Geometric view of the lexicographical descent algorithm of [1]

Example 2. *Let us continue with Example 1 to demonstrate the underlying geometrical intuition behind Algorithm 4 of [1]. For readability, in this example we use superscripts to distinguish between different*

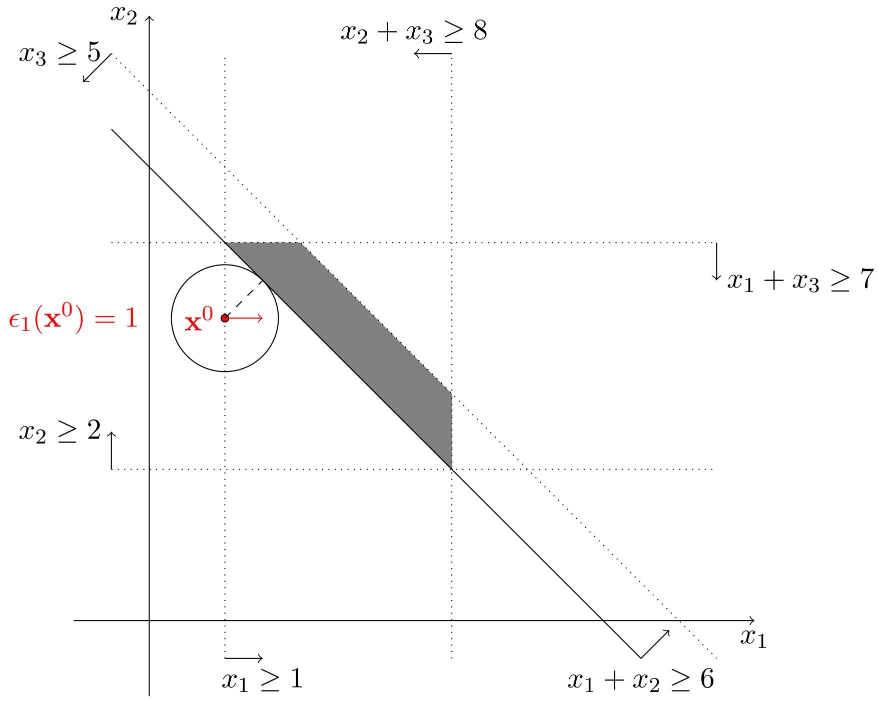


Figure 2: Hyperplane of preimputations: $x_1 + x_2 + x_3 = v(\mathcal{N}) = 12$

imputations, while subscripts of maximum dissatisfaction levels ϵ_k *and tight sets* T_k *are used to keep track of iterations. Figure 2 summarises the game and the solution approach in the* (x_1, x_2) *preimputation space, where* $x_3 = v(\mathcal{N}) - x_1 - x_2$. *The dotted lines show the coalitional rationality constraints and the gray trapezoid is the projection of the core* **Co** *of the game onto the* (x_1, x_2) *space. We are using Algorithm 4 of [1] to find the nucleolus of* ν *from a starting imputation* $\mathbf{x}^0 = [1, 4, 7]$.

The underlying geometrical intuition can be easily followed through by considering Figure 3, which represents a magnified part of Figure 2. First, we find that our distance to the boundary of the core *is 1, hence* $\epsilon_1(\mathbf{x}^0) = 1$. *We also find that currently the largest infeasibility among core inequalities belongs to constraint* $x_1 + x_2 \geq 6$. *Therefore,* $T_1(\mathbf{x}^0) = \{\{1, 2\}\}$, *while* $T_0(\mathbf{x}^0) = \{\{1\}\}$ *because* $x_1^0 = v(\{1\})$. *It is easy to see that the current tight set* $T_1(\mathbf{x}^0)$ *is not* $T_0(\mathbf{x}^0)$*-balanced. In geometrical terms this means there are nearby points having smaller distance to the boundary of the core. In the algorithm we run into an infeasible system when checking the balancedness, so we can find improving directions by solving* $ID(T_1(\mathbf{x}^0); T_0(\mathbf{x}^0); \mathcal{N})$. *As an example,* $\mathbf{y} = [-1, 0, 1]$ *is an improving direction depicted with an arrow from the current point* $\mathbf{x}^0$. *Moving along this direction we can decrease the radius of the Euclidean ball touching the boundary of the core.*

Notice that here we measure the distance from the boundary with a special signed *distance; in the*

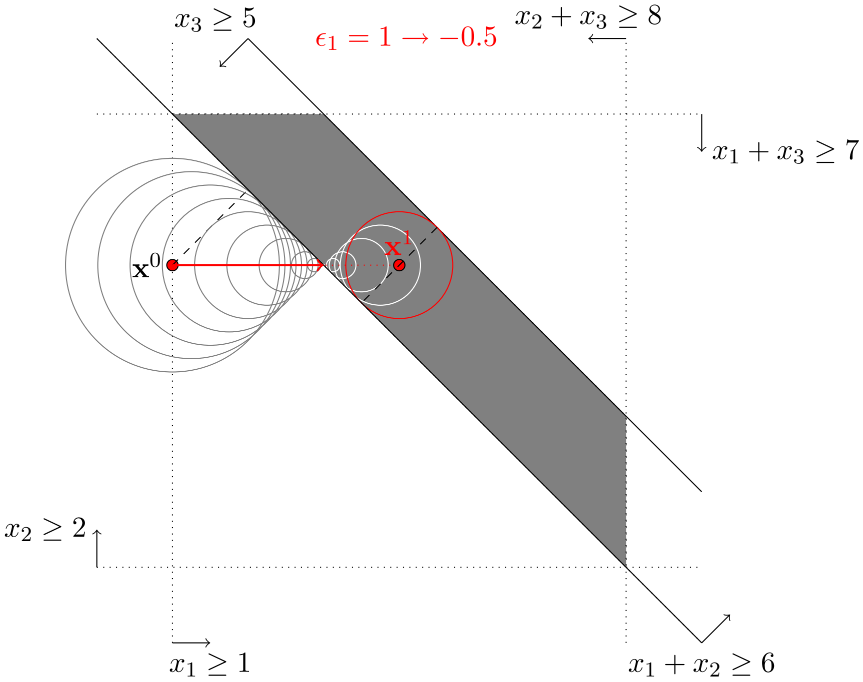


Figure 3: Pivot step

interior of the core the distance from the boundary is understood to be negative. Thus, we can move even further along the direction after reaching the core, until eventually we can not decrease *the radius of the ball touching the boundary of the core any more. This happens precisely when the distances from both constraints* $(x_3 \geq 5)$ *and* $(x_1 + x_2 \geq 6)$ *are* -0.5*, that is when* $\mathbf{x}^1 = [2.5, 4, 5.5]$*,* $\epsilon_1(\mathbf{x}^1) = -0.5$*,* $T_1(\mathbf{x}^1) = \{\{1,2\},\{3\}\}$ *and* $T_0(\mathbf{x}^1) = \emptyset$*. This corresponds to a pivot step in a simplex-like algorithm, where coalition* $\{3\}$ *enters the basis. In accordance with (14) of [1], the step size chosen in direction* $\mathbf{y}$ *is* $\alpha = 1.5$*. After this step, we find that* $T_1(\mathbf{x}^1)$ *is* $T_0(\mathbf{x}^1)$*-balanced, therefore we expand* R_0 *with an arbitrary element of* $T_1(\mathbf{x}^1)$ *as* $\mathrm{rank}(R_0 \cup T_1(\mathbf{x}^1)) = 2$*. By doing so we lift those inequality constraints that made it impossible to decrease the radius of the ball further; that is throughout the remaining execution of the algorithm those constraints remain satisfied with equality at the current largest excess level of* $\epsilon_1(\mathbf{x}^1) = -0.5$*.*

The set of imputations having the coalitions in $T_1(\mathbf{x}^1)$ *being tight at* $\epsilon_1 = -0.5$ *actually form the least core* $\mathbf{LC}$ *of the game, depicted by the thick black line in Figure 4. The dashed line shows that, at the current point* $\mathbf{x}^1$*, among constraints not in* $\mathrm{span}(R_1)$*, we are closest to violating* $x_1 + x_3 \geq 7$ *with* $\epsilon_2(\mathbf{x}^1) = -1$*. Further, we have that* T_0 *remains empty at* $\mathbf{x}^1$ *and the tight set* $T_2(\mathbf{x}^1) = \{\{1,3\}\}$ *is not* T_0*-balanced, so we find an improving direction parallel to the set of least core outcomes. That is, the*

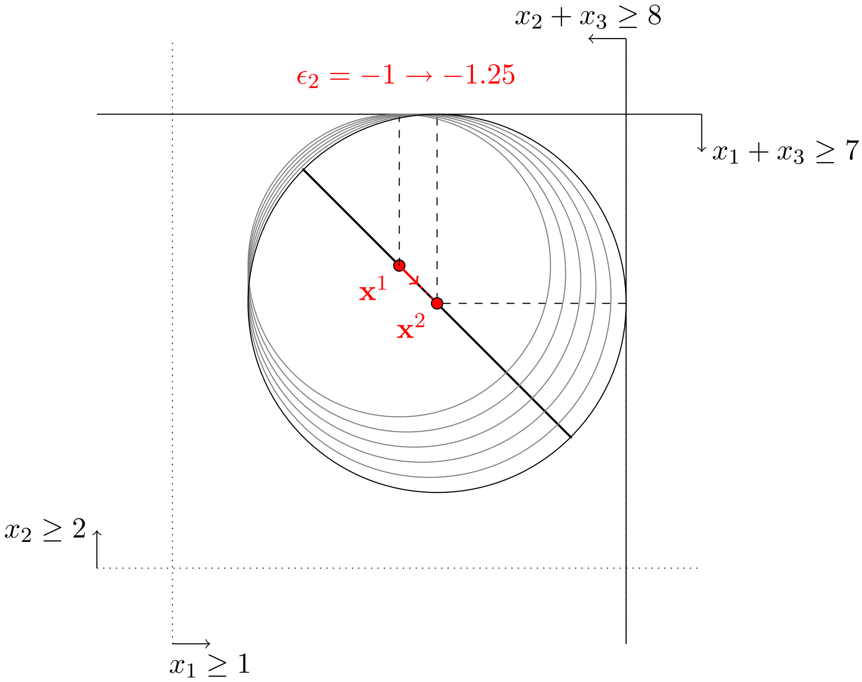


Figure 4: Pivot step

unique solution of $ID(T_2(\mathbf{x}^1); T_0; R_1)$ *is* $\mathbf{y} = [1, -1, 0]$*, and we can take a step size of* $\alpha = 1/4$ *until coalition* $\{2, 3\}$ *becomes tight as well. The resulting point is* $\mathbf{x}^2 = [2.75, 3.75, 5.5]^\top$ *with largest excess* $\epsilon_2(\mathbf{x}^2) = -1.25$*,* $T_0 = \emptyset$ *and tight set* $T_2(\mathbf{x}^2) = \{\{1, 3\}, \{2, 3\}\}$*. Since* $T_1(\mathbf{x}^2) \cup T_2(\mathbf{x}^2)$ *is* T_0*-balanced and* $\operatorname{rank}(R_1 \cup T_2(\mathbf{x}^2)) = n$ *we found the nucleolus* $\nu = \mathbf{x}^2$*.*

D Alternative Proof of Kohlberg Criterion

Let (T_0, T) be two collections of coalitions. For each coalition $\mathcal{C} \in T$, let us the define of the primal

$$\begin{aligned} \max_{\gamma,\omega,\alpha} \quad & \omega_{\mathcal{C}} \\ s.t. \quad & \sum_{\mathcal{S}\in T_0} \gamma_{\mathcal{S}} \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S}\in T} \omega_{\mathcal{S}} \mathbf{e}(\mathcal{S}) - \alpha \mathbf{e}(\mathcal{N}) = 0, \\ & \gamma \geq 0, \omega \geq 0, \end{aligned} \tag{$\tilde{P}(\mathcal{C})$}$$

and the corresponding dual problem

$$\begin{aligned} \min\nolimits_{\mathbf{y}} \quad & 0 \\ s.t. \quad & \mathbf{y}^\top \mathbf{e}(\mathcal{S}) \geq 0, \quad \forall \mathcal{S} \in T_0 \cup T, \\ & \mathbf{y}^\top \mathbf{e}(\mathcal{N}) = 0, \\ & \mathbf{y}^\top \mathbf{e}(\mathcal{C}) \geq 1. \end{aligned} \tag{$\tilde{D}(\mathcal{C})$}$$

We have the following results:

Lemma 1. *For any given pair of subsets (T_0, T) of the power set $2^{\mathcal{N}}$, the followings are equivalent*

(a) T is T_0-balanced if any only if for all $\mathcal{C} \in T$, the primal problem ($\tilde{P}(\mathcal{C})$) is unbounded.

(b) For any $\mathcal{C} \in T$, the primal problem ($\tilde{P}(\mathcal{C})$) is unbounded if any only if the dual ($\tilde{D}(\mathcal{C})$) is infeasible.

(c) The dual problem ($\tilde{D}(\mathcal{C})$) is infeasible for all $\mathcal{C} \in T$ if any only if (T_0, T) has property I.

(d) (T_0, T) has property II if and only if T is T_0-balanced.

Proof. Result in part (d) is what we want to show and this will follows directly if we are able to show (a)-(c). We choose to show both sides of the if and only if statements in part (a)-(c) so that each of these can be viewed as stand-alone results even though the proof of the entire lemma only requires one direction such as (a)⇒(b)⇒(c)⇒(d)⇒(a).

(a) ⇒ If T is T_0-balanced, there exists weight vectors $\gamma \in \mathbb{R}^{|T_0|}_{\geq 0}$,$\omega \in \mathbb{R}^{|T|}_{>0}$ such that

$$\sum_{\mathcal{S} \in T_0} \gamma_{\mathcal{S}} \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S} \in T} \omega_{\mathcal{S}} \mathbf{e}(\mathcal{S}) = \mathbf{e}(\mathcal{N}).$$

For each $\mathcal{C} \in T$, we have $(\gamma, \omega, 1)$ is a feasible solution to ($\tilde{P}(\mathcal{C})$) with an objective value of $\omega_{\mathcal{C}} > 0$. Since the problem is homogeneous on (γ, ω, α), that is for all $\Delta > 0$, we have $(\Delta\gamma, \Delta\omega, \Delta\alpha)$ is also a feasible solution with an optimal value of $\Delta\omega_{\mathcal{C}}$, the primal problem is unbounded and hence the dual problem ($\tilde{P}(\mathcal{C})$) is infeasible.

⇐ For each $\mathcal{C} \in T$, given that the primal problem ($\tilde{P}(\mathcal{C})$) is unbounded, we can pick a corresponding feasible solution $(\gamma, \omega, 1)$ with a positive objective value $\omega_{\mathcal{C}}$. Average out all such feasible solutions $(\gamma, \omega, 1)$, one for each $\mathcal{C} \in T$, we would obtain the average weight $(\bar{\gamma}, \bar{\omega})$ that satisfies $\bar{\gamma} \geq 0$, $\bar{\omega} > 0$, and

$$\sum_{\mathcal{S} \in T_0} \bar{\gamma}_{\mathcal{S}} \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S} \in T} \bar{\omega}_{\mathcal{S}} \mathbf{e}(\mathcal{S}) = \mathbf{e}(\mathcal{N})$$

Thus, T is T_0-balanced.

(b) We can see that the primal problem is always feasible at $(\gamma = 0, \omega = 0, \alpha = 0)$. In addition, the problem is homogeneous on (γ, ω, α) and hence its optimal value is either zero or it is unbounded. The dual problem, on the other hand, is either infeasible or has an optimal value zero. From linear programming duality results, it is easy to show that in this case, the primal is unbounded if and only if the dual is infeasible.

(c) $\Leftarrow$ If (T_0, T) has property I, we have $(\tilde{D}(\mathcal{C}))$ infeasible for all $\mathcal{C} \in T$ by definition of property I.

$\Rightarrow$ If $(\tilde{D}(\mathcal{C}))$ infeasible for all $\mathcal{C} \in T$ then (T_0, T) must have property I since otherwise there exists a $\mathbf{y} \in Y(T_0 \cup T)$ and a coalition $\mathcal{C} \in T$ such that $\mathbf{y}(\mathcal{C}) > 0$. Thus, we can scale up $\mathbf{y}$ by an appropriate factor Δ such that $\Delta\mathbf{y} \in Y(T_0 \cup T)$ and $\Delta\mathbf{y}(\mathcal{C}) \geq 1$. This means the dual problem $(\tilde{D}(\mathcal{C}))$ is feasible. Contradiction!

$\square$

E Proof of Lemma 2 of [1]

Proof. (a) Given that T is T_0-balanced, there exists $\gamma \in \mathbb{R}^{|T_0|}_{\geq 0}$ and $\omega \in \mathbb{R}^{|T|}_{>0}$ such that

$$\mathbf{e}(\mathcal{N}) = \sum_{\mathcal{S} \in T_0} \gamma_{\mathcal{S}} \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S} \in T} \omega_{\mathcal{S}} \mathbf{e}(\mathcal{S}).$$

For any $\mathcal{S}_0 \in \text{span}(T)$, there exists β such that $\mathbf{e}(\mathcal{S}_0) = \sum_{\mathcal{S} \in T} \beta_{\mathcal{S}} \mathbf{e}(\mathcal{S})$. Thus, for any δ, we have

$$\begin{aligned}
\mathbf{e}(\mathcal{N}) &= \sum_{\mathcal{S} \in T_0} \gamma_{\mathcal{S}} \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S} \in T} \omega_{\mathcal{S}} \mathbf{e}(\mathcal{S}) \\
&= \sum_{\mathcal{S} \in T_0} \gamma_{\mathcal{S}} \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S} \in T} \omega_{\mathcal{S}} \mathbf{e}(\mathcal{S}) + \delta(\mathbf{e}(\mathcal{S}_0) - \sum_{\mathcal{S} \in T} \beta_{\mathcal{S}} \mathbf{e}(\mathcal{S})) \\
&= \sum_{\mathcal{S} \in T_0} \gamma_{\mathcal{S}} \mathbf{e}(\mathcal{S}) + \delta \mathbf{e}(\mathcal{S}_0) + \sum_{\mathcal{S} \in T} (\omega_{\mathcal{S}} - \delta\beta_{\mathcal{S}}) \mathbf{e}(\mathcal{S}).
\end{aligned}$$

Since $\omega > 0$, we can choose $\delta > 0$ which is small enough such that $(\omega_{\mathcal{S}} - \delta\beta_{\mathcal{S}}) > 0$, $\forall \mathcal{S} \in T$. Thus, $T \cup \{\mathcal{S}_0\}$ is a T_0-balanced collection. Since this holds for all $\mathcal{S}_0 \in \text{span}(T)$, we can conclude that $\text{span}(T)$ is T_0-balanced.

(b) Given that collections U, V are T_0-balanced, there exists $\gamma, \omega \in \mathbb{R}^{|T_0|}_{\geq 0}$ and $\alpha \in \mathbb{R}^{|U|}_{>0}$, $\beta \in \mathbb{R}^{|V|}_{>0}$ such that

$$\mathbf{e}(\mathcal{N}) = \sum_{\mathcal{S} \in T_0} \gamma_{\mathcal{S}} \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S} \in U} \alpha_{\mathcal{S}} \mathbf{e}(\mathcal{S}) = \sum_{\mathcal{S} \in T_0} \omega_{\mathcal{S}} \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S} \in V} \beta_{\mathcal{S}} \mathbf{e}(\mathcal{S}).$$

This leads to

$$\mathbf{e}(\mathcal{N}) = \sum_{\mathcal{S} \in T_0} (1/2\gamma_{\mathcal{S}} + 1/2\omega_{\mathcal{S}}) \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S} \in U} 1/2\alpha_{\mathcal{S}} \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S} \in V} 1/2\beta_{\mathcal{S}} \mathbf{e}(\mathcal{S}).$$

Thus, $U \cup V$ is also T_0-balanced. We can also prove that $\text{span}(U) \cup \text{span}(V)$ is T_0-balanced in a similar way as shown in the proof of part (a).

(c) The proof is similar to part (a) due to the fact that, for any $\mathcal{S}_0 \in \text{span}(U) \cap V$, we have $\mathcal{S}_0 \in \text{span}(U)$ and hence $U \cup \mathcal{S}_0$ is also T_0-balanced. Thus, $\text{span}(U) \cap V$ is T_0-balanced. $\square$

F Proof of Theorem 1

Proof. After each iteration, we have $T_k \not\subseteq \text{span}(R_{k-1}) = \text{span}(H_{k-1})$ and $\text{rank}(R_k) \geq 1$ by its construction. Therefore $\text{rank}(R_k) = \text{rank}(T_k \cup H_{k-1})$ keeps increasing. Thus, Algorithm 1 terminates after at most $(n-1)$ iterations. We also note that the algorithm terminates at either Step 4 or 5 with complementary conclusions.

Proving that the algorithm correctly decides whether an imputation is the nucleolus is equivalent to showing that (a) if $\mathbf{x}$ is the nucleolus then the algorithm terminates at Step 5, and (b) if the algorithm terminates at Step 5, then the input payoff vector must be the nucleolus.

The proof for part (a) is still the same as that proof for Theorem 2 of [1] since the key property used in that proof was to preserve the T_0-balancedness of H_k. The analogy to Theorem 1 is that R_k is always T_0-balanced throughout the algorithm according to Lemma 3b of [1]. If $\mathbf{x}$ is the nucleolus then T_1 is T_0-balanced and the algorithm gets through to Step 3 at $k = 1$. Suppose, for the purpose of deriving a contradiction, that the algorithm terminates at Step 4 at some index $k > 1$ with $(R_{k-1} \cup T_k)$ not T_0-balanced while R_{k-1} is T_0-balanced by the construction in Step 3 of the previous iteration. Then, by Lemma 1 of [1], there exists $\mathbf{y} \in \mathbb{R}^n$ such that

$$\mathbf{y}(\mathcal{S}) \geq 0, \forall \mathcal{S} \in T_0 \cup R_{k-1} \cup T_k;\ \mathbf{y}(\mathcal{N}) = 0;\ \mathbf{y}(\mathcal{S}') > 0,\ \text{for some } \mathcal{S}' \in T_k.$$

Thus,

$$\Phi^{\text{span}(R_{k-1}) \cup T_k}(\mathbf{x} + \mathbf{y}) <_L \Phi^{\text{span}(R_{k-1}) \cup T_k}(\mathbf{x}).$$

In addition, for all $\mathcal{S} \notin (\text{span}(R_{k-1}) \cup T_k)$ we have $v(\mathcal{S}) - \mathbf{x}(\mathcal{S}) < \epsilon_k$ by the construction in Step 2. Thus, there exist $\delta > 0$, which is small enough such that

$$v(\mathcal{S}) - (\mathbf{x} + \delta \mathbf{y})(\mathcal{S}) < \epsilon_k,\ \forall \mathcal{S} \notin (\text{span}(R_{k-1}) \cup T_k)$$

and

$$\Phi^{\text{span}(R_{k-1}) \cup T_k}(\mathbf{x} + \delta \mathbf{y}) <_L \Phi^{\text{span}(R_{k-1}) \cup T_k}(\mathbf{x}).$$

In other words, the $|\text{span}(R_{k-1}) \cup T_k|$ largest excess value of $\mathbf{x}$ is lexicographically larger than the excess values of $(\mathbf{x} + \delta \mathbf{y})$ on these collections of coalitions. As a result, $\Phi(\mathbf{x})$ is lexicographically larger than $\Phi(\mathbf{x} + \delta \mathbf{y})$ considering all coalitions, which means $\mathbf{x}$ is not the nucleolus. Contradiction!

The proof for part (b) is also the same as that in Theorem 2 of [1] where the key property of retaining $\text{rank}(H_k) = \text{rank}(R_k)$ increased throughout the algorithm is still preserved. Due to the T_0-balancedness of $(R_{k-1} \cup T_k)$, we can use Lemma 1 of [1] to recursively show that $(\mathbf{z} - \mathbf{x})(\mathcal{S}) = 0$ for all $\mathcal{S} \in R_{k-1} \cup T_k$ where $\mathbf{z}$ is the nucleolus as follows.

Let $\mathbf{y} = \mathbf{z} - \mathbf{x}$. We have $\mathbf{y}(\mathcal{N}) = 0$ and $\mathbf{y}(\mathcal{S}) \geq 0, \forall \mathcal{S} \in T_0$ due to the fact that $\mathbf{z}$ is an imputation and also by the definition of T_0. We also have $\mathbf{y}(\mathcal{S}) \geq 0, \forall \mathcal{S} \in R_1$ due to the fact that $\Phi^{R_1}(\mathbf{z}) \leq_L \Phi^{R_1}(\mathbf{x})$.

Applying the result from Lemma 1 of [1] with a note that R_1 is T_0-balanced, we have $\mathbf{y}(\mathcal{S}) = 0, \forall \mathcal{S} \in R_1$ which means $\mathbf{z}$ and $\mathbf{x}$ are lexicographically equivalent on R_1. We will prove by induction that $\mathbf{y}(\mathcal{S}) = 0, \forall \mathcal{S} \in R_k$ for all indices k. Suppose this indeed holds for $(k-1)$, i.e. $\mathbf{y}(\mathcal{S}) = 0, \forall \mathcal{S} \in R_{k-1}$. In other words, $\mathbf{z}$ and $\mathbf{x}$ are lexicographically equivalent on R_{k-1} which spans the collection of coalitions from which $\mathbf{x}$ receives the worst excess values. In order for $\mathbf{z}$ to be lexicographically at least as good as $\mathbf{x}$, the excess values of $\mathbf{z}$ on those coalitions in R_k must be no worse than those from $\mathbf{x}$, i.e., $\mathbf{y}(\mathcal{S}) \geq 0, \forall \mathcal{S} \in R_k$. Applying the result from Lemma 1 [1] with a note that R_k is T_0-balanced, we must also have $\mathbf{y}(\mathcal{S}) = 0, \forall \mathcal{S} \in R_k$; that is $(\mathbf{x} - \mathbf{z})(\mathcal{S}) = 0, \forall \mathcal{S} \in R_k$. Since $\text{rank}(R_{k-1}) = n$, we have $\mathbf{x} = \mathbf{z}$. □

G Proof of Lemma 3 of [1]

Proof. Part (a): if T is T_0-balanced then obviously (7) of [1] is feasible with $\mu = 0$. Now suppose that (7) of [1] is satisfied with some weights $\gamma \in \mathbb{R}^{|T_0|}_{\geq 0}, \omega \in \mathbb{R}^{|T|}_{>0}, \mu \in \mathbb{R}$. First observe that $\mu < 1$ since $\omega > 0$ and $\gamma \geq 0$. Thus,

$$\sum_{\mathcal{S} \in T_0} \frac{\gamma_{\mathcal{S}}}{1-\mu} \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S} \in T} \frac{\omega_{\mathcal{S}}}{1-\mu} \mathbf{e}(\mathcal{S}) = \mathbf{e}(\mathcal{N}),$$

and hence T is T_0-balanced.

Part (b): if T is T_0-balanced then obviously (8) of [1] is feasible with $\mu > 0$. Now suppose $\exists \gamma \in \mathbb{R}^{|T_0|}_{\geq 0}, \omega \in \mathbb{R}^{|T \setminus Q|}_{>0}, \mu \in \mathbb{R}^{|Q|}$ satisfying (8) of [1]. Since Q is T_0-balanced,

$$\sum_{\mathcal{S} \in T_0} \alpha_{\mathcal{S}} \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S} \in Q} \beta_{\mathcal{S}} \mathbf{e}(\mathcal{S}) = \mathbf{e}(\mathcal{N}). \tag{1}$$

with $\alpha \geq \mathbf{0}$ and $\beta > \mathbf{0}$. Choose $\delta > 0$ large enough such that $\mu_{\mathcal{S}} + \delta \beta_{\mathcal{S}} > 0$ for all $\mathcal{S} \in Q$. Then (8) of [1] and (1) lead to

$$\sum_{\mathcal{S} \in T_0} \frac{\gamma_{\mathcal{S}} + \delta \alpha_{\mathcal{S}}}{\delta + 1} \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S} \in T \setminus Q} \frac{\omega_{\mathcal{S}}}{\delta + 1} \mathbf{e}(\mathcal{S}) + \sum_{\mathcal{S} \in Q} \frac{\mu_{\mathcal{S}} + \delta \beta_{\mathcal{S}}}{\delta + 1} \mathbf{e}(\mathcal{S}) = \mathbf{e}(\mathcal{N})$$

showing that T is indeed T_0-balanced. □

H Numerical results for the Kohlberg algorithms

As we mentioned in Section 5 of [1], in the followings we focus our attention to the performance of the various Kohlberg algorithms (*Kohlberg*, *IKA* and *IKAcr*) on payoff vectors randomly chosen from the imputations set, the least core, and the so-called *least-least core* (with $T_1 \cup T_2$ being T_0-balanced). Also as we mentioned, rejecting a random imputation being trivial, and the original Kohlberg algorithm unusable even for moderate sized games ($n \geq 15$), when analysing the results we compare *IKA* and *IKAcr* on the remaining two outcomes. The notation is in line with Section 5 of [1].

H.1 Type I and II games

Table 1: Original and Improved Kohlberg algorithms on type I games and random imputation

n	Time			Iterations			Subroutines			Repr.
	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	IKAcr
5	0.0006	0.0006	**0.0004**	**1**	**1**	**1**	**1**	**1**	**1**	0
10	0.022	0.0011	**0.0009**	**1**	**1**	**1**	**1**	**1**	**1**	0
15	0.024	**0.023**	0.023	**1**	**1**	**1**	**1**	**1**	**1**	0
20	0.94	**0.92**	0.95	**1**	**1**	**1**	**1**	**1**	**1**	0
25	36.6	**36**	36.7	**1**	**1**	**1**	**1**	**1**	**1**	0
26	**75.7**	76	76	**1**	**1**	**1**	**1**	**1**	**1**	0
27	158	**157**	158	**1**	**1**	**1**	**1**	**1**	**1**	0
28	754	755	**746**	**1**	**1**	**1**	**1**	**1**	**1**	0
29	122	110	**107**	**1**	**1**	**1**	**1**	**1**	**1**	0
30	472	454	**446**	**1**	**1**	**1**	**1**	**1**	**1**	0

We observe that other outcomes from the least core greatly resemble what we have noticed in Tables 1 and 2 of [1] with the nucleolus. As the number of iterations are the same for *IKA* and *IKAcr*, since there is only a slight edge in the number of subroutines for *IKAcr* in type I games, we can note that *IKAcr* is still faster in most of the cases (especially for the least core), but the difference is not very significant.

Even though the slightest of difference between *IKA* and *IKAcr* in iterations appears for type II games, we observe practically the same regardless of outcomes from the least or the least-least core. The outlier values of coalitions saved from storage by *IKAcr* at the 30-player type I games are due to a single game producing a tight set magnitudes larger than the average size, but this particular instance does not effect greatly the overall picture.

Table 2: Original and Improved Kohlberg algorithms on type II games and random imputation

n	Time			Iterations			Subroutines			Repr.
	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	IKAcr
5	0.0006	0.0003	**0.0001**	**1**	**1**	**1**	**1**	**1**	**1**	0
10	0.0018	0.0009	**0.0007**	**1**	**1**	**1**	**1**	**1**	**1**	0
15	0.024	0.023	**0.023**	**1**	**1**	**1**	**1**	**1**	**1**	0
20	0.95	**0.92**	0.93	**1**	**1**	**1**	**1**	**1**	**1**	0
25	36.6	**36**	36.9	**1**	**1**	**1**	**1**	**1**	**1**	0
26	**75.4**	76.3	75.8	**1**	**1**	**1**	**1**	**1**	**1**	0
27	158	**157**	158	**1**	**1**	**1**	**1**	**1**	**1**	0
28	**728**	777	752	**1**	**1**	**1**	**1**	**1**	**1**	0
29	113	108	**108**	**1**	**1**	**1**	**1**	**1**	**1**	0
30	**456**	492	464	**1**	**1**	**1**	**1**	**1**	**1**	0

Table 3: Original and Improved Kohlberg algorithms on type I games and least core payoff

n	Time			Iterations			Subroutines			Repr.
	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	IKAcr
5	0.012	0.0013	**0.0011**	6.9	**1.8**	**1.8**	12.2	2.6	**2.0**	1.2
10	1.3	**0.0026**	0.003	385.6	**1.8**	**1.8**	527.8	2.5	**1.8**	1.2
15	OoT	**0.052**	0.052	OoT	**1.6**	**1.6**	OoT	2.2	**1.6**	1
20	-	2.9	**2.9**	-	**1.6**	**1.6**	-	2.3	**1.7**	1.1
25	-	124	**124**	-	**1.6**	**1.6**	-	2.3	**1.8**	1.3
26	-	235	**228**	-	**1.5**	**1.5**	-	1.9	**1.5**	1.2
27	-	500	**487**	-	**1.6**	**1.6**	-	2.1	**1.6**	1.1
28	-	2459	**2289**	-	**1.7**	**1.7**	-	2.2	**1.7**	1.2
29	-	543	**529**	-	**1.1**	**1.1**	-	1.5	**1.4**	1
30	-	**1601**	1652	-	**1.1**	**1.1**	-	1.5	**1.3**	72842

H.2 Type III and IV games

As type III games are rather trivial for primal methods (such as the various Kohlberg algorithms), it is of no surprise to find in Table 5 of [1] that the verification of type III games requires only 1 iteration from *IKA* and *IKAcr*. Bearing this in mind it is evident that the results for type III games completely coincide in the case of least, least-least core and the nucleolus: the least core outcome is already the nucleolus. Therefore, in order to avoid unnecessary duplication, we omitted repeating tables for the least and least-least core payoff vectors with identical content to Table 5 of [1].

Table 4: Original and Improved Kohlberg algorithms on type II games and least core payoff

n	Time			Iterations			Subroutines			Repr.
	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	IKAcr
5	0.0026	**0.0011**	0.0013	4.7	**2.2**	**2.2**	8.2	3.1	**2.2**	0.7
10	1.7	0.0025	**0.0023**	383.2	2	**1.9**	791.6	2.9	**1.9**	1.0
15	OoT	0.081	**0.081**	OoT	**2**	**2**	OoT	3.1	**2.4**	1.4
20	-	3.5	**3.5**	-	**1.9**	**1.9**	-	3	**2.1**	1.2
25	-	111	**110**	-	**1.8**	**1.8**	-	2.9	**2.1**	1.3
26	-	313	**302**	-	**1.9**	**1.9**	-	2.7	**2.1**	1.3
27	-	534	**526**	-	**1.8**	**1.8**	-	2.7	**1.9**	1.1
28	-	2491	**2318**	-	**2**	**2**	-	3.3	**2.4**	1.4
29	-	216	**214**	-	**1**	**1**	-	**2**	**2**	1
30	-	1327	**1299**	-	**1.1**	**1.1**	-	2	**1.9**	1.1

Table 5: Original and Improved Kohlberg algorithms on type I games and least-least core payoff

n	Time			Iterations			Subroutines			Repr.
	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	IKAcr
5	0.01	0.0016	**0.0013**	18.7	**2.5**	**2.5**	30.9	3.2	**2.5**	2.2
10	1.9	**0.003**	0.0038	671.7	**2.2**	**2.2**	909.1	2.9	**2.2**	1.9
15	OoT	**0.062**	0.063	OoT	**1.8**	**1.8**	OoT	2.4	**1.8**	1.7
20	-	3.2	**3.2**	-	**1.7**	**1.7**	-	2	**1.7**	1.7
25	-	147	**146**	-	**1.8**	**1.8**	-	2.3	**1.8**	1.6
26	-	259	**250**	-	**1.6**	**1.6**	-	1.8	**1.6**	1.5
27	-	603	**587**	-	**1.8**	**1.8**	-	2.3	**1.8**	1.7
28	-	2778	**2557**	-	**1.9**	**1.9**	-	2.3	**1.9**	1.8
29	-	**1302**	1317	-	**1.4**	**1.4**	-	1.8	**1.6**	1.4
30	-	**2686**	2750	-	**1.3**	**1.3**	-	1.6	**1.4**	72842.3

The results for type IV games are very similar to type II, with a little more distinguishment in the number of subroutines due to the increased added value of the compact representation of large tight sets in these games.

H.3 UN Security Council game

As we mentioned above in Subsection 5.4 of [1], we introduced weighted voting games based on the UN Security Council voting mechanism with 5 *big* (veto) players, while the rest of the players are *small*. Weights are calibrated such that all 5 veto players' agreement needed to pass a vote, while still approx.

Table 6: Original and Improved Kohlberg algorithms on type II games and least-least core payoff

n	Time			Iterations			Subroutines			Repr.
	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	IKAcr
5	0.009	**0.0015**	**0.0015**	17.1	2.7	**2.6**	31.4	3.8	**2.6**	1.4
10	1.9	**0.003**	**0.003**	483.4	2.4	**2.3**	1005	3.6	**2.5**	1.7
15	OoT	0.1	**0.1**	OoT	**2.4**	**2.4**	OoT	3.4	**2.5**	1.7
20	-	4.7	**4.7**	-	**2.3**	**2.3**	-	3.3	**2.4**	2
25	-	**211**	212	-	**2.7**	**2.7**	-	4.3	**2.9**	2.1
26	-	451	**443**	-	**2.5**	**2.5**	-	3.2	**2.7**	2.4
27	-	817	**733**	-	**2.3**	**2.3**	-	3.2	**2.6**	2.2
28	-	3536	**3307**	-	**2.9**	**2.9**	-	4.6	**3.1**	2.2
29	-	**2312**	2418	-	**2**	**2**	-	3.5	**2.8**	2
30	-	**5714**	5867	-	**2**	**2**	-	3	**2.7**	2

Table 7: Original and Improved Kohlberg algorithms on type III games and random imputation

n	Time			Iterations			Subroutines			Repr.
	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	IKAcr
5	0.0007	**0.0003**	0.0006	**1**	**1**	**1**	**1**	**1**	**1**	0
10	0.61	**0.0011**	0.0014	**1**	**1**	**1**	**1**	**1**	**1**	0
15	0.024	0.023	**0.023**	**1**	**1**	**1**	**1**	**1**	**1**	0
20	0.94	**0.92**	0.93	**1**	**1**	**1**	**1**	**1**	**1**	0
25	36.5	**36**	36.9	**1**	**1**	**1**	**1**	**1**	**1**	0
26	**75.4**	76.2	75.7	**1**	**1**	**1**	**1**	**1**	**1**	0
27	158	**157**	158	**1**	**1**	**1**	**1**	**1**	**1**	0
28	**602**	623	670	**1**	**1**	**1**	**1**	**1**	**1**	0
29	116	**110**	114	**1**	**1**	**1**	**1**	**1**	**1**	0
30	235	**225**	227	**1**	**1**	**1**	**1**	**1**	**1**	0

half of the small players also needed. The weights are for $n \geq 7$: $w_j = \lfloor \frac{n-3}{2} \rfloor$ for $j = 1, \ldots, 5$ and $w_j = 1$ otherwise, the quota is $q = 4w_1 + n - 4$, and the game is

$$v(\mathcal{S}) = \begin{cases} 1, & \text{if } w(\mathcal{S}) \geq q, \\ 0, & \text{otherwise.} \end{cases}$$

As we noted in the introduction of the UN Security Council game in Subsection 5.3 of [1], we expect that these games carry sufficient distinguishing power between *IKA* and *IKAcr* with its very large tight sets, where a compact representation is vital.

Table 8: Original and Improved Kohlberg algorithms on type IV games and random imputation

n	Time			Iterations			Subroutines			Repr.
	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	IKAcr
5	0.0006	0.0004	**0.0003**	**1**	**1**	**1**	**1**	**1**	**1**	0
10	0.0021	0.001	**0.0008**	**1**	**1**	**1**	**1**	**1**	**1**	0
15	0.024	**0.023**	0.023	**1**	**1**	**1**	**1**	**1**	**1**	0
20	0.94	**0.93**	0.93	**1**	**1**	**1**	**1**	**1**	**1**	0
25	36.5	**36**	36.9	**1**	**1**	**1**	**1**	**1**	**1**	0
26	**75.5**	76.2	75.7	**1**	**1**	**1**	**1**	**1**	**1**	0
27	158	**157**	158	**1**	**1**	**1**	**1**	**1**	**1**	0
28	**734**	736	738	**1**	**1**	**1**	**1**	**1**	**1**	0
29	116	110	**108**	**1**	**1**	**1**	**1**	**1**	**1**	0
30	436	448	**429**	**1**	**1**	**1**	**1**	**1**	**1**	0

Table 9: Original and Improved Kohlberg algorithms on type IV games and least core payoff

n	Time			Iterations			Subroutines			Repr.
	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	IKAcr
5	0.0041	0.001	**0.001**	7.7	**1.6**	**1.6**	13.7	2.7	**1.8**	1.9
10	0.98	**0.0024**	0.0025	196.1	**1.8**	**1.8**	413.9	3.2	**2.3**	3.3
15	OoT	**0.056**	0.057	OoT	**1.6**	**1.6**	OoT	3.2	**2.6**	6
20	-	2.4	**2.4**	-	**1.6**	**1.6**	-	**3.6**	3.7	15.3
25	-	129	**129**	-	**2**	**2**	-	4.4	**3.1**	10.9
26	-	283	**275**	-	**2**	**2**	-	4.1	**3.7**	9.6
27	-	642	**635**	-	**2.1**	**2.1**	-	4.6	**3.4**	14.9
28	-	2482	**2403**	-	**2**	**2**	-	4.6	**3.8**	13.3
29	-	**416**	422	-	**1.1**	**1.1**	-	3.4	**3.2**	14.5
30	-	1454	**1405**	-	**1.1**	**1.1**	-	3.3	**3**	16.6

In the case of the nucleolus in Table 13, we can clearly see the effect of the compact representation in *IKAcr*, however, it does not always translate to faster computations. This is a reasonable trade-off between computational time and memory usage, eventually leading to *IKA* running out of memory for $n = 28$, whereas *IKAcr* running out of time instead.

While we find that both methods are unable to verify the nucleolus for $n = 28$, the reason, *IKA* running out of memory while *IKAcr* running out of time, differs very much in quality. It means that with our available computational resources, and some additional time *IKAcr* can complete the task.

[1] As the least-least core element coincides with the nucleolus, we omitted the former to avoid unnecessary duplication.

Table 10: Original and Improved Kohlberg algorithms on type IV games and least-least core payoff

n	Time			Iterations			Subroutines			Repr.
	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	IKAcr
5	0.01	**0.0009**	0.0009	14.1	**1.6**	**1.6**	24.5	2.6	**1.8**	2.8
10	1.9	**0.0025**	0.0027	225.4	**1.9**	**1.9**	478.1	3.3	**2.4**	3.6
15	OoT	0.073	**0.073**	OoT	**1.9**	**1.9**	OoT	3.6	**2.8**	9.3
20	-	3.7	**3.6**	-	**2.1**	**2.1**	-	4.3	**4**	17.6
25	-	**199**	204	-	**2.7**	**2.7**	-	5.7	**3.7**	14.1
26	-	470	**455**	-	**2.8**	**2.8**	-	5.3	**4.4**	14.6
27	-	954	**946**	-	**2.8**	**2.8**	-	5.8	**4**	16
28	-	3569	**3360**	-	**2.8**	**2.8**	-	5.8	**4.3**	15.8
29	-	**2576**	2666	-	**2.1**	**2.1**	-	5.1	**4.5**	23.3
30	-	**5888**	6003	-	**2**	**2**	-	4.9	**4**	24.1

Table 11: Original and Improved Kohlberg algorithms on UN Security Council game and random imputation

n	Time			Iterations			Subroutines			Repr.
	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	IKAcr
7	**0.001**	**0.001**	0.019	**1**	**1**	**1**	**1**	**1**	**1**	0
10	**0.001**	**0.001**	**0.001**	**1**	**1**	**1**	**1**	**1**	**1**	0
15	0.024	0.024	**0.022**	**1**	**1**	**1**	**1**	**1**	**1**	0
20	0.99	0.96	**0.94**	**1**	**1**	**1**	**1**	**1**	**1**	0
25	38.8	**37.3**	37.3	**1**	**1**	**1**	**1**	**1**	**1**	0
26	81.1	88.9	**75.7**	**1**	**1**	**1**	**1**	**1**	**1**	0
27	**156**	157	162	**1**	**1**	**1**	**1**	**1**	**1**	0
28	378	**348**	351	**1**	**1**	**1**	**1**	**1**	**1**	0

For this we find evidence in Table 12, as *IKA* already runs out of memory for an element of the least core, while *IKAcr* is capable of completing the rejection in the second iteration. The difference is of no wonder, as we can note the very high usage of our compact representation of extremely large tight sets.

*Bibliography

[1] M. Benedek, J. Fliege, and T.D. Nguyen. Finding and verifying the nucleolus of cooperative games, 2019.

[2] M. Maschler, B. Peleg, and L.S. Shapley. Geometric properties of the kernel, nucleolus, and related solution concepts. *Mathematics of operations research*, 4(4):303–338, 1979.

Table 12: Original and Improved Kohlberg algorithms on UN Security Council game and least core payoff

n	Time			Iterations			Subroutines			Repr.
	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	IKAcr
7	**0.001**	0.14	0.19	**2**	**2**	**2**	5	4	**3**	3
10	0.003	**0.002**	0.081	**2**	**2**	**2**	7	5	**4**	43
15	**0.060**	0.081	0.13	**2**	**2**	**2**	5	4	**3**	1658
20	2.8	3.3	**2.8**	**2**	**2**	**2**	7	5	**4**	49147
25	**112**	134	130	**2**	**2**	**2**	5	4	**3**	1665238
26	1259	1570	**248**	**2**	**2**	**2**	7	5	**4**	3145723
27	OoT	4760	**2410**	OoT	**2**	**2**	OoT	4	**3**	6644168
28	-	OoM	5967	-	OoM	2	-	OoM	4	12582907

Table 13: Original and Improved Kohlberg algorithms on UN Security Council game and nucleolus payoff[1]

n	Time			Iterations			Subroutines			Repr.
	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	Kohlberg	IKA	IKAcr	IKAcr
7	0.003	**0.001**	**0.001**	6	**2**	**2**	6	**3**	**3**	19
10	0.008	**0.003**	0.004	6	**2**	**2**	7	**4**	**4**	199
15	0.21	0.075	**0.073**	6	**2**	**2**	6	**3**	**3**	6774
20	OoT	3.2	**3.1**	OoT	**2**	**2**	OoT	**4**	**4**	212983
25	-	**134**	143	-	**2**	**2**	-	**3**	**3**	6908114
26	-	847	**555**	-	**2**	**2**	-	**4**	**4**	13631479
27	-	**4381**	5332	-	**2**	**2**	-	**3**	**3**	27615684
28	-	OoM	OoT	-	OoM	OoT	-	OoM	OoT	OoT

[3] J.K. Sankaran. On finding the nucleolus of an n-person cooperative game. *International Journal of Game Theory*, 19(4):329–338, 1991.

[4] T. Solymosi. On computing the nucleolus of cooperative games. *Ph.D. Thesis, University of Illinois*, 1993. doi: 10.13140/RG.2.2.28952.80642.

[5] J.A.M. Potters, J.H. Reijnierse, and M. Ansing. Computing the nucleolus by solving a prolonged simplex algorithm. *Mathematics of operations research*, 21(3):757–768, 1996.

[6] J. Derks and J. Kuipers. Implementing the simplex method for computing the prenucleolus of

transferable utility games. 1997.

[7] T. Nguyen and L. Thomas. Finding the nucleoli of large cooperative games. *European Journal of Operational Research*, 3(248):1078–1092, 2016.