COMPLEXITY IN COOPERATIVE GAME THEORY

J. M. Bilbao[1], J. R. Fernández[2] and J. J. López[3]

Matemática Aplicada II, Escuela Superior de Ingenieros
Camino de los Descubrimientos s/n, 41092 Sevilla, Spain
http://www.esi.us.es/~mbilbao/

Abstract. We introduce cooperative games (N, v) with a description polynomial in n, where n is the number of players. In order to study the complexity of cooperative game problems, we assume that a cooperative game $v : 2^N \to \mathbb{Q}$ is given by an oracle returning $v(S)$ for each query $S \subseteq N$. Finally, we consider several cooperative game problems and we give a list of complexity results.

Key words. Computational complexity, cooperative game
AMS subject classifications. 91A12

1 Introduction

Let $v : 2^N \to \mathbb{Q}$ be a cooperative game with rational worths. The coalition description of v is given by its 2^n coalitional values $\{v(S) : S \subseteq N\}$. If the complexity is measured in the size of these values, then the input is exponential in n, and thus, the most of the complexity questions would become easy. Therefore, let us start with games (N, v) with a description polynomial in n, where n is the number of players.

Example: A *weighted voting game* $[q;\ w_1, \ldots, w_n]$, where w_i is the number of votes of each player i, is a description polynomial in n of the game v, defined by

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

[1] E-mail: mbilbao@us.es
[2] E-mail: julio@us.es
[3] E-mail: jorge@polimat01.eup.us.es

where $w(S) = \sum_{i \in S} w_i$. We will denote these games by wVG.

Example: Deng and Papadimitriou [1] studied computational complexity of cooperative solution concepts for the following game. If $G = (N, E)$ is a graph with weight $w : E \to \mathbb{Z}$, the *weighted graph game* $v_G : 2^N \to \mathbb{Z}$ is given by $v_G(S) = \sum_{e \in G[S]} w(e)$, and the class of these games is denoted by wGG.

The game v_G evaluates the profit $v_G(S)$ of any coalition S as the profit of the subgraph of G induced by the vertex set S. We can use it for the fair division between n cities of the income from a communication network connecting them. Since $|E| \leq \frac{1}{2} n(n-1)$, this game has a description polynomial in n.

Example: Nagamochi, Zeng, Kabutoya and Ibaraki [14] consider a matroid $(E, \mathcal{M})$ with weight function $w : E \to \mathbb{Q}$. The *minimum base game* $c : 2^E \to \mathbb{Q}$ is given by

$$c(S) = \min \left\{ \sum_{e \in B(S)} w(e) : B(S) \text{ is a basis of } S \right\},$$

and we denoted by MBG the class of these games. If the matroid $(E, \mathcal{M})$ is the forest matroid of a graph $G = (N, E)$, then the *minimum forest game* is

$$c(S) = \min \left\{ \sum_{e \in F_S} w(e) : F_S \text{ is a maximal forest on } S \right\},$$

for all $S \subseteq E$ and the class is *MFG.*

For the last game, we define the encoding length or size of a matroid $(E, \mathcal{M})$ as the number of its independents sets $|\mathcal{M}|$ and we assume that the matroid is given by an *oracle* that testing if a given set $S \subseteq E$ is independent or not. In this context, a matroid on $|E|$ elements can be represented by a string of length $2^{|E|}$, where each character indicates whether a subset $S \subseteq E$ is independent.

In this model, a minimum base $B(S)$ on S can be obtained, in polynomial time, by the greedy algorithm as follows:

Order $S = \{e_1, \ldots, e_s\}$ so that $c(e_1) \leq \cdots \leq c(e_t) < 0 \leq \cdots \leq c(e_s)$;
    $B(S) \leftarrow \emptyset$;
For $i = 1$ to s do
    If $B(S) \cup \{e_i\} \in \mathcal{M}$ then $B(S) \leftarrow B(S) \cup \{e_i\}$.

2 Cooperative game problems

In order to study the complexity of *cooperative game problems*, we assume that a cooperative game $v : 2^N \to \mathbb{Q}$ is given by an oracle returning $v(S)$ for each query $S \subseteq N$. For this oracle we have a polynomial p_v such that for every input of size at most n, the answer of the oracle has size at most $p_v(n)$. Therefore, we know an upper bound $\beta \geq \max \{\text{size}(f(S)) : S \subseteq N\}$.

Definition 1 *Let $v : 2^N \to \mathbb{Q}$ be a game given by an oracle with upper bound β. The size of v is defined by* $\text{size}(v) = |N| + \beta$.

Definition 2 *We say that a problem for a game $v : 2^N \to \mathbb{Q}$ given by an oracle, is solvable in oracle-polynomial time if the number of computational steps, counting each call to the oracle as one step, is polynomially bounded in* $\text{size}(v)$.

Example: A *min-cost spanning tree game* is defined by a set $N = \{1, \ldots, n\}$ of players, a supply vertex 0, a complete graph $G = K_{n+1} = (N \cup \{0\}, E)$ and a nonnegative edge-weight function $w : E \to \mathbb{Q}_+$. The cost $c(S)$ of any coalition $S \subseteq N$, is defined by

$$c(S) = \min \left\{ \sum_{e \in T_S} w(e) : T_S \text{ is a spanning tree of } G[S \cup \{0\}] \right\}.$$

The class of these games is denoted by *MSTG.*

Remark 1 *We know that the greedy algorithm is a polynomial-time algorithm for the min-cost spanning tree problem. Therefore, the oracle is any algorithm that computes $c(S)$ in polynomial time with respect to $|N| + \text{size}(w)$, for all $S \subseteq N$.*

We consider now several cooperative game problems and we give a list of complexity results.

Problem: MEMBER CORE

- Instance: A game $v : 2^N \to \mathbb{Q}$ with $\text{size}(v)$, and $x \in I(v) \cap \mathbb{Q}^N$.
- Question: Is x an element of the core of v?

Good News: For the class *wGG*, Deng and Papadimitriou [1] proved that it is in P if the weight function w is nonnegative. If v is convex (or concave) then it can be solved in oracle-polynomial time by the greedy algorithm (Grötschel, Lovász and Schrijver [9, Section 10.2]).

Bad News: coNP-complete for the classes *wGG* and *MSTG* (Deng and Papadimitriou [1] and Faigle, Kern, Fekete and Hochstättler [2]).

Comments: Deng and Papadimitriou used reduction from MAX-CUT and Faigle *et al.* [2] used reduction from EXACT COVER BY 3-SETS (Papadimitriou [15, page 201]). It is still open the complexity for the classes *MBG* and *MFG.*

Problem: CORE

- Instance: A game $v : 2^N \to \mathbb{Q}$ with size (v).
- Question: Is the core of v empty?

Good News: For *wGG* with nonnegative weights and *MFG* with no all-negative circuit is in P. For *MBG* such that the underlying matroid has no all-negative circuits it is solvable in oracle-polynomial time (Deng and Papadimitriou [1] and Nagamochi *et al.* [14]).

Bad News: NP-complete for *wGG* (Deng and Papadimitriou [1]).

Problem: SUBADDITIVITY

- Instance: A game $c : 2^N \to \mathbb{Q}$ with size (v).
- Question: Is the game c subadditive?

Good News: Nagamochi *et al.* [14] obtained an O(size (v)) time algorithm for the class *MFG* and proved that it is solvable in oracle linear time for *MBG.*

Problem: SUBMODULARITY

- Instance: A game $c : 2^N \to \mathbb{Q}$ with size (v).
- Question: Is the game c submodular?

Good News: It is solvable in oracle-polynomial time for the class *MBG* and in $\mathrm{O}(\mathrm{size}\,(v))$ time for *MFG* (Nagamochi *et al.* [14]).

Problem: SHAPLEY VALUE

- Instance: A game $v : 2^N \to \mathbb{Q}$ with $\mathrm{size}\,(v)$.
- Solution: The Shapley value $\Phi(v)$.

Good News: Megiddo [13] showed that it can be found in $\mathrm{O}(n)$ time for *MSTG,* in the special case where the underlying graph is a tree. For *wGG*, Deng and Papadimitriou [1] proved that it can be computed in $\mathrm{O}(n^2)$.

Bad News: For the class *wVG*, Deng and Papadimitriou [1] proved that is #P-complete, that is, as difficult as any counting problem in NP. Nagamochi *et al.* [14] proved that is #P-complete for the class *MFG* and that there is no oracle-polynomial time algorithm for the class *MBG.*

Comments: Deng and Papadimitriou [1] used the counting problem associated with KNAPSACK (Papadimitriou [15, page 202]). Nagamochi *et al.* [14] used the #P-completeness of GRAPH RELIABILITY (Valiant [18]).

Problem: TIJS VALUE

- Instance: A game $v : 2^N \to \mathbb{Q}$ with $\mathrm{size}\,(v)$.
- Solution: The Tijs value $\tau\,(v)$.

Good News: For the class *MFG*, Nagamochi *et al.* [14] proved that it can be computed in $\mathrm{O}(m^2 + mn\log_2 n)$ time, where $n = |N|$ and $m = |E|$.

Bad News: For the class *MBG*, there is no oracle-polynomial time algorithm.

Comments: Nagamochi *et al.* [14] used the Dijkstra's algorithm implemented by Fredman and Tarjan [5] (see also Lovász *et al.* [12, Theorem 3.3]).

Problem: NUCLEOLUS

- Instance: A game $v : 2^N \to \mathbb{Q}$ with $\mathrm{size}\,(v)$.
- Solution: The nucleolus $\eta\,(v)$.

Good News: Granot and Granot [6] and Granot, Maschler, Owen and Zhu [7] showed that it can be found in strongly polynomial time for a subclass of *MSTG*. Solymosi and Raghavan [16] provides an $\mathrm{O}(n^4)$ algorithm for the class of assignments games. For the class wGG, Deng and Papadimitriou [1] proved that it can be computed in $\mathrm{O}(n^2)$. If v is convex (or concave) then it can be solved in oracle-polynomial time (Kuipers [10] and Faigle, Kern and Kuipers [3]). The $\mathcal{B}$-nucleolus can be calculated in strongly polynomial time for games with size$(\mathcal{B})$ polynomially bounded in n (Granot, Granot and Zhu [8], Kuipers, Solymosi and Aarts [11], Solymosi, Aarts and Driessen [17]).

Bad News: NP-hard, that is, at least as difficult as any optimization problem in NP, for the class *MSTG* (Faigle, Kern and Kuipers [4]).

References

[1] Deng, X. and C. H. Papadimitriou (1994): On the complexity of cooperative solution concepts, *Math. Oper. Res.* 19, 257–266.

[2] Faigle, U., W. Kern, S. P. Fekete and W. Hochstättler (1997): On the complexity of testing membership in the core of min-cost spanning tree games, *Int. J. Game Theory* 26, 361–366.

[3] Faigle, U., W. Kern and J. Kuipers (1998): An efficient algorithm for nucleolus and prekernel computation in some classes of TU-games, *Working paper*, University of Twente, The Netherlands.

[4] Faigle, U., W. Kern and J. Kuipers (1998): Computing the nucleolus of min-cost spanning tree games is NP-hard, *Int. J. Game Theory* 27, 443–450.

[5] Fredman, M. L. and R. E. Tarjan (1987): Fibonacci heaps and their uses in improved network optimization algorithms, *J. Assoc. Comput. Mach.* 34, 596–615.

[6] Granot, D. and F. Granot (1992): Computational complexity of a cost allocation approach to a fixed cost spanning forest problem, *Math. Oper. Res.* 17, 765–780.

[7] Granot, D., M. Maschler, G. Owen and W. R. Zhu (1996): The kernel/nucleolus of a standard tree game, *Int. J. Game Theory* 25, 219–244.

[8] Granot, D., F. Granot and W. R. Zhu (1998): Characterization sets for the nucleolus, *Int. J. Game Theory* 27, 359–374.

[9] Grötschel, M., L. Lovász and A. Schrijver (1993): *Geometric Algorithms and Combinatorial Optimization,* second edition, Springer-Verlag, Berlin.

[10] Kuipers, J. (1996): A polynomial time algorithm for computing the nucleolus of convex games, *Report M 96-12,* University of Maastricht, The Netherlands.

[11] Kuipers, J., T. Solymosi and H. Aarts (1995): Computing the nucleolus of some combinatorially structured games, *Report M 95-08,* University of Maastricht, The Netherlands. To appear in *Math. Programming.*

[12] Lovász, L., D. B. Shmoys and É. Tardos (1995): Combinatorics in Computer Science, in R. Graham, M. Grötschel and L. Lovász, eds., *Handbook of Combinatorics, Vol. II*, Elsevier Science B.V., Amsterdam, 2003–2038.

[13] Megiddo, N. (1978): Computational complexity of the game theory approach to cost allocation for a tree, *Math. Oper. Res.* 3, 189–196.

[14] Nagamochi, H., D.-Z. Zeng, N. Kabutoya and T. Ibaraki (1997): Complexity of the minimum base game on matroids, *Math. Oper. Res.* 22, 146–164.

[15] Papadimitriou, C. H. (1994): *Computational Complexity,* Addison-Wesley, Reading, Massachusetts.

[16] Solymosi, T. and T. E. S. Raghavan (1994): An algorithm for finding the nucleolus of assignment games, *Int. J. Game Theory* 23, 119–143.

[17] Solymosi, T., H. Aarts and T. Driessen (1998): On computing the nucleolus of a balanced connected game, *Math. Oper. Res.* 23, 983–1009.

[18] Valiant, L. G. (1979): The complexity of enumeration and reliability problems, *SIAM J. Comput.* 8, 410–421.