# Computing Stable Solutions in Threshold Network Flow Games With Bounded Treewidth

Extended Abstract

Aldo Pacchiano
UC Berkeley
Berkeley, CA
pacchiano@berkeley.edu

Yoram Bachrach
Deepmind
London, U.K
yorambac@google.com

## ABSTRACT

Network flow games are a prominent model for team formation, where a commodity can flow through a network whose edges are controlled by selfish agents. In Threshold Network Flow Games (TNFGs), an agent team is successful if the flow it can achieve between a source and target vertices meets or exceeds a certain threshold. Cooperative game theory allows predicting how agents are likely to share the the joint reward in such settings, by applying solution concepts such as the core, which characterizes stable reward distributions. When TNFGs have empty cores, every reward distribution is somewhat unstable, which requires using a relaxed solution such as the least-core to find the most stable distribution. Earlier work showed that computing the least-core in TNFGs is computationally hard, but tractable for very restricted graphs, such as layer graphs. We extend these results, presenting polynomial algorithms for the much larger class of bounded-treewidth graphs.

## KEYWORDS

Network Flow Games; Least Core

**ACM Reference Format:**

Aldo Pacchiano and Yoram Bachrach. 2019. Computing Stable Solutions in Threshold Network Flow Games With Bounded Treewidth. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

## 1 INTRODUCTION

When multiple agents inhabit the same environment, their actions affect one another. Game theory characterizes the possible outcomes in such settings, examining many domains including security [25, 30, 31], trade [8, 10, 17, 32], mechanism design [21, 24, 28], voting [2, 7, 33] and negotiation [1, 11, 12]. When no individual agent can accomplish a task on their own, agents must form a team to achieve the goal and may interact trying to negotiate the reward allocation [14, 18, 27]. An agent who gets a low share of the reward in one team may be tempted to join another team offering them a higher reward [9, 16, 22]. Thus, a stable coalition can only be formed if the resulting gains are distributed appropriately. Game theory suggests ways to distribute gains, formalized in *cooperative solution concepts* such as the core [15], and least-core [23].

A transferable utility **coalitional game** is composed of a set of $n$ agents, $I = \{1, 2, \ldots, n\}$, and a **characteristic function** mapping any subset (coalition) of the agents to a rational value $v : 2^I \rightarrow \mathbb{Q}$, indicating the total utility these agents achieve together.The characteristic function only defines the gains a *coalition* achieves, but does not indicate how they are to be distributed among the coalition's agents. An **imputation** $(p_1, \ldots, p_n)$ is a division of the gains of the grand coalition $I$ among the agents, where $p_i \geq 0$, such that $\sum_{i=1}^{n} p_i = v(I)$. We call $p_i$ the payoff of agent $i$, and denote the payoff of a coalition $C$ as $p(C) = \sum_{i \in C} p_i$. We say a coalition $B$ **blocks** the payoff vector $(p_1, \ldots, p_n)$ if $p(B) < v(B)$, since $B$'s members can split from the original coalition, derive the gains of $v(B)$ in the game, give each member $i \in B$ its previous gains $p_i$, and still some utility remains, so each member can get more utility. If a blocked payoff vector is chosen, the coalition is unstable. A prominent solution focusing on stability is the core [15]. The **core** of a game is the set of all imputations $(p_1, \ldots, p_n)$ that are not blocked by any coalition, so that for any coalition $C$, we have: $p(C) \geq v(C)$. The $\epsilon$-core is the set of all imputations $(p_1, \ldots, p_n)$ such that for any coalition $C \subseteq I$, $p(C) \geq v(C) - \epsilon$. Given an imputation $p$, the **excess** of $C$ is the difference between $C$'s value and payoff: $e(C) = v(C) - p(C)$. A natural question is finding the smallest $\epsilon$ such that the $\epsilon$-core is non-empty, known as the **least-core**.

## 2 THE LEAST-CORE IN TNFGS

Network Flow Games (**NFGs**) model team formation among selfish agents trying to maximize the flow between a source and a target in a network [4, 19, 20]. A flow network consists of a directed graph $G = \langle V, E \rangle$ with capacities on the edges $c : E \rightarrow \mathbb{Q}+$, a distinguished source vertex $s \in V$ and a target (sink) vertex $t \in V$. A network flow is a function $f : E \rightarrow \mathbb{Q}+$ which obeys the capacity constraints and conserves total zero flow at each vertex except $s, t$.

A **threshold network flow domain** consists of a network flow graph $G = \langle V, E \rangle$, with capacities on the edges $c : E \rightarrow \mathbb{Q}$, a source vertex $s$, a target vertex $t$, and a set $I$ of agents, where agent $i$ controls the edge $e_i \in E$, and flow threshold $k$. A coalition $C \subseteq I$, controls the edges $E_C = \{e_i | i \in C\}$. Given a coalition $C$ we denote by $G_C = \langle V, E_C \rangle$ the induced graph where the only edges are those that belong to the agents in $C$. We denote by $f_C$ the maximum flow between $s$ and $t$ in $G_C$. In a **threshold network flow game (TNFG)**, a coalition $C$ wins if it can achieve a $k$-flow between $s$ and $t$ and loses otherwise. The characteristic function of the game, $v(C)$ is 1 if $f_C \geq k$, so $E_C$ allows a flow of $k$ from $s$ to $t$ and 0 otherwise.

**Maximal Excess in TNFGS:** One key question given a TNFG is determining the degree of stability a certain imputation $p = (p_1, \ldots, p_n)$ achieves, as measured by the excess, the maximal incentive any sub-coalition has to deviate and call the problem of

*Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.* 

computing it **ME** (Maximal Excess). While core related problems can be tractably solved for *Cardinal* NFGs [19, 20], the threshold version of TNFGs is computationally more difficult.

Despite hardness results for computing the least-core in TNFGs, polynomial algorithms have been proposed for two very restricted classes of TNFGs: where all the edges have the same capacity, and where the network topology is of a layer graph with bounded degrees and bounded integer capacities [3]. We extend these results to graphs which have a bounded treewidth. The treewidth of a graph is a parameter which describes the "locality" of interaction in a graph [5, 6]. Our method is a "fixed-parameter traclable" approach [13], meaning that it works on *any* graph but its runtime is exponential in the treewidth of the network, so it only has a polynomial runtime for graphs whose treewidth is bounded by a low constant. In particular, as series-parallel graphs, outerplanar graphs and Halin graphs all have a treewidth of at most 3, our approach works for these types of networks in polynomial time.

Earlier work has uncovered an relation between finding a maximal excess coalition in a TNFG and a variant of the min-cost flow problem. In the **All-Or-Nothing Min-Cost Flow** problem we are given a network flow domain and a target flow $k$, and edge cost vector $X \in \mathbb{R}^E$ with $X = \{x_e\}_{e \in E}$. The All-Or-Nothing cost (AON-cost) of an edge subset $C$ is the total cost of edges with a non-zero flow. In the AON Min-Cost Flow problem we are asked to find a $k$-flow of a minimal AON cost (i.e. a flow $f$ of minimal AON cost whose magnitude is at least $k$). Earlier work shows that the maximal excess problem **ME** is equivalent to AON Min-Cost Flow [3].

## 3 AON MIN-COST FLOW IN BOUNDED TREEWIDTH GRAPHS

We provide an algorithm for the computation of the $\epsilon-$core of TNFGs on bounded treewidth and bounded integer capacities networks. Our method belongs to the class of Fixed Parameter Tractable algorithms. Bounded **treewidth** is a common structural assumption in the analysis of graph based algorithms that allows to prove tractable running times for special classes of graphs.

*Definition 3.1.* A *Flow Tree decomposition* of a graph $G = (V, E)$ - directed or indirected- is a pair $(\mathcal{T}, \mathcal{X})$ where $\mathcal{T}$ is a tree and $X = \{X_i | X_i \subseteq V, i \text{ vertex } \in \mathcal{T}\}$ is a family of subsets of $V$ for which the Tree Decomposition conditions hold (see [26]), and:

(1) $T$ is a rooted binary tree
(2) There are four types of nodes: *introduce nodes*, *forget nodes*, *leaf nodes* and *join nodes*.
  (a) $i$ is an introduce node if it has only one child $j$ and $X_i = X_j \cup \{v\}$ for some $v \in V \backslash X_j$.
  (b) $i$ is a forget node if it has only one child $j$ and $X_i \cup \{v\} = X_j$.
  (c) $i$ is called a join node if it has two children $j, h$ such that $X_i = X_j = X_h$.
  (d) $i$ is a leaf node if it does not have any child and $X_i$ consists of a node $u$ together with a subset of its neighborhood.
(3) Each of the sets corresponding to leaves of $(T, \mathcal{X})$ is an edge of $G$. No two leaves have the same associated edge.

The width of a decomposition is the maximum size of a subset of $\mathcal{X}$. The treewidth of $G$ is the minimum width of all decompositions.

### 3.1 An Algorithm for AON Min-Cost Flow

Our first step is defining an auxiliary problem, the AON -Min-Cost-Circulation and show AON-Min-Cost-Flow can be reduced to it. A circulation is a flow vector such that for all vertices in the network the inflow equals the outflow. Our algorithm first builds a Flow Tree Decomposition $(\mathcal{T}, \mathcal{X})$ of the input graph and constructs tables of partial solutions $\{\mathcal{P}\}_{i \in \mathcal{T}}$ via upward propagation through $\mathcal{T}$. It pastes these partial solutions into a global one. The way in which it does this depends on the type of the node in the Flow Tree Decomposition. We get a polynomial algorithm for bounded treewidth graphs for the AON min flow problem. If exists, a solution can be recovered by backtracking through reconstruction maps.

**Algorithm 2**. AON-Min-Cost-Circulation Algorithm for bounded treewidth graphs with bounded integer capacities.

**Inputs:** Graph $G$, flow parameter $k$, special nodes, $s, t$.

(1) Construct a Flow Tree Decomposition $(T, X)$ for $G$.
(2) Set the flow from $s$ to $t$ equal to $k$.
(3) Construct tables of partial solutions for each of the nodes $i \in T$ starting by the leaves and propagating the solutions upwards via the procedures outlined above depending on the type of node (Forget, Introduce or Join).
(4) If $r$ is the root of $T$, the AON min cost $k$-circulation over $G$ equals the min cost among all valid partial solutions.
(5) Reconstruct the min AON cost $k-$circulation by backtracking through the reconstruction maps.

### 3.2 Algorithm Runtime

Let $w$ be the treewidth of the input graph. For any edge $e$ let $C_e$ be its maximum allowed flow. We call $C_G = \max_{e \in E} C_e$ the maximum capacity over all the edges in $E$, and $d_G$ the maximum degree of $G$.

If it exists, a Flow Tree Decomposition $(\mathcal{T}, \mathcal{X})$ of $G$ can be built in linear time [29] ensuring $|\mathcal{T}|$ is linear in $|G|$. Step 3 is the intensive part of the Algorithm 2 and has a runtime of $O(|\mathcal{T}| C_G^{w^2} * (C_G * d_G)^{2w})$ with a memory use of $O(C_G^{w^2} * (C_G * d_G)^w |\mathcal{T}|)$ for the partial solutions tables and reconstruction maps of each node of $\mathcal{T}$. Whenever $\omega$ is a constant, the runtime of our algorithm is polynomial on $|G|$ and $C_G$. Since there is an equivalence between the **AON Min-Cost Flow** problem and the **ME** problem in TNFGs [3]:

Corollary 3.2. *ME is in P for TNFGs over bounded treewidth graphs with bounded integer capacities.*

## 4 CONCLUSIONS

We extend existing positive results regarding the least-core in TNFGs to a much larger class of graphs. Our approach relates to a parameter of the graph called the treewidth, relating to the "locality" of interaction between nodes [5]. Our method can compute the least core of TNFGs with bounded treewidth in polynomial time. It also works for general graphs, in which case its runtime is exponential in the treewidth of the graph. We also provide simulation results for geometric graphs and duplication divergence graphs, which model realistic traffic networks with spatially local interaction, showing that our approach scales well to such networks eventhough their have unbounded treewidth.

## REFERENCES


[1] Samir Aknine, Suzanne Pinson, and Melvin F Shakun. 2004. An extended multi-agent negotiation protocol. *Autonomous Agents and Multi-Agent Systems* 8, 1 (2004), 5–45.

[2] Kenneth J Arrow, Amartya Sen, and Kotaro Suzumura. 2010. *Handbook of social choice and welfare*. Vol. 2. Elsevier.

[3] Yoram Bachrach. 2011. The least-core of threshold network flow games. In *International Symposium on Mathematical Foundations of Computer Science*. Springer, 36–47.

[4] Yoram Bachrach and Jeffrey S. Rosenschein. 2009. Power in threshold network flow games. *Autonomous Agents and Multi-Agent Systems* (2009).

[5] Hans L Bodlaender. 1994. A tourist guide through treewidth. *Acta cybernetica* 11, 1-2 (1994), 1.

[6] Hans L. Bodlaender. 1997. Treewidth: Algorithmic Techniques and Results. *Proceedings 22nd International Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science* 1295 (1997), 29–36.

[7] Felix Brandt, Vincent Conitzer, and Ulle Endriss. 2012. Computational social choice. *Multiagent systems* (2012), 213–283.

[8] Mukun Cao, Xudong Luo, Xin Robert Luo, and Xiaopei Dai. 2015. Automated negotiation for e-commerce decision making: a goal deliberated agent architecture for multi-strategy selection. *Decision Support Systems* 73 (2015), 1–14.

[9] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. 2011. Computational aspects of cooperative game theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 5, 6 (2011), 1–168.

[10] Georgios Chalkiadakis, Valentin Robu, Ramachandra Kota, Alex Rogers, and Nicholas R Jennings. 2011. Cooperatives of distributed energy resources for efficient virtual power plants. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 787–794.

[11] Yiling Chen, John K Lai, David C Parkes, and Ariel D Procaccia. 2013. Truth, justice, and cake cutting. *Games and Economic Behavior* 77, 1 (2013), 284–297.

[12] Yann Chevaleyre, Paul E Dunne, Ulle Endriss, Jérôme Lang, Michel Lemaitre, Nicolas Maudet, Julian Padget, Steven Phelps, Juan A Rodriguez-Aguilar, and Paulo Sousa. 2006. Issues in multiagent resource allocation. (2006).

[13] Rodney G Downey and Michael Ralph Fellows. 2012. *Parameterized complexity*. Springer Science & Business Media.

[14] Joan Feigenbaum, Christos H Papadimitriou, and Scott Shenker. 2001. Sharing the cost of multicast transmissions. *J. Comput. System Sci.* 63, 1 (2001), 21–41.

[15] Donald Bruce Gillies. 1953. *Some theorems on n-person games*. Ph.D. Dissertation.

[16] Joseph Greenberg. 1994. Coalition structures. *Handbook of game theory with economic applications* 2 (1994), 1305–1337.

[17] Robert H Guttman, Alexandros G Moukas, and Pattie Maes. 1998. Agent-mediated electronic commerce: A survey. *The Knowledge Engineering Review* 13, 2 (1998), 147–159.

[18] Nicholas R Jennings, Peyman Faratin, Alessio R Lomuscio, Simon Parsons, Michael J Wooldridge, and Carles Sierra. 2001. Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation* 10, 2 (2001), 199–215.

[19] Ehud Kalai and Eitan Zemel. 1982. Generalized Network Problems Yielding Totally Balanced Games. *Operations Research* 30 (September 1982), 998–1008.

[20] Ehud Kalai and Eitan Zemel. 1982. Totally balanced games and games of flow. *Mathematics of Operations Research* 7, 3 (1982), 476–478.

[21] Sven Koenig, Pinar Keskinocak, and Craig A Tovey. 2010. Progress on Agent Coordination with Cooperative Auctions.. In *AAAI*, Vol. 10. 1713–1717.

[22] Hideo Konishi and Debraj Ray. 2003. Coalition formation as a dynamic process. *Journal of Economic theory* 110, 1 (2003), 1–41.

[23] M. Maschler, B. Peleg, and L. S. Shapley. 1979. Geometric Properties of the Kernel, Nucleolus, and Related Solution Concepts. *Math. Oper. Res.* (1979).

[24] David C Parkes and Michael P Wellman. 2015. Economic reasoning and artificial intelligence. *Science* 349, 6245 (2015), 267–272.

[25] James Pita, Manish Jain, Janusz Marecki, Fernando Ordóñez, Christopher Portway, Milind Tambe, Craig Western, Praveen Paruchuri, and Sarit Kraus. 2008. Deployed ARMOR protection: the application of a game theoretic model for security at the Los Angeles International Airport. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track*. International Foundation for Autonomous Agents and Multiagent Systems, 125–132.

[26] Neil Robertson and Paul D. Seymour. 1986. Graph minors. II Algorithmic aspects of Tree-width. *J.Algorithms* 7(3) (1986), 309–322.

[27] Tuomas W Sandhlom and Victor RT Lesser. 1997. Coalitions among computationally bounded agents. *Artificial intelligence* 94, 1-2 (1997), 99–137.

[28] Tuomas Sandholm. 2002. Algorithm for optimal winner determination in combinatorial auctions. *Artificial intelligence* 135, 1-2 (2002), 1–54.

[29] P. Scheffler. 1994. A practical linear time algorithm for disjoint paths in graphs with bounded tree–width. *Technical Report 396 Berlin, Fachbereich 3 Mathematik* 396 (1994).

[30] Eric Shieh, Bo An, Rong Yang, Milind Tambe, Craig Baldwin, Joseph DiRenzo, Ben Maule, and Garrett Meyer. 2012. Protect: A deployed game theoretic system to protect the ports of the united states. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 13–20.

[31] Milind Tambe. 2011. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge University Press.

[32] Maksim Tsvetovat and Katia Sycara. 2000. Customer coalitions in the electronic marketplace. In *Proceedings of the fourth international conference on Autonomous agents*. ACM, 263–264.

[33] Lirong Xia. 2013. Designing social choice mechanisms using machine learning. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 471–474.