An Application of SAT Solvers in Integer Programming Games

Pravesh Koirala
Department of Computer Science, Vanderbilt University, Nashville, TN, USA

Aditya Shrey
Department of Computer Science, Vanderbilt University, Nashville, TN, USA

Forrest Laine
Department of Computer Science, Vanderbilt University, Nashville, TN, USA

Abstract

Integer programming games (IPGs) are a popular game-theoretic tool to model an array of games where each player has a discrete strategy set. These games arise in important domains such as economics, transportation, cybersecurity, etc., but solving them is non-trivial as it is known that checking for the existence of pure Nash equilibria in an IPG is Σ_2^p-complete. Recent works have proposed a class of relaxed solution concepts for IPGs called locally optimal integer solutions (LOIS) and shown it to be an efficient alternative for pure Nash equilibria. While LOIS are significantly simpler to compute, they still do not scale when solved using traditional mathematical solvers, especially when high-quality solutions are desired. In this paper, we apply commercially available SAT solvers to find LOIS in IPGs. We investigate efficient encodings for a cybersecurity game and compare solution times when using SAT solvers vs mathematical program solvers. We also investigate the application of SAT solvers in graph games using a graph interdiction example and compare against the obtained LOI solutions against existing heuristics-based solutions. Our results indicate that with appropriate encodings, large-scale IPGs can be solved much more efficiently using SAT solvers. We also show that SAT solvers can be applied to graph games in conjunction with LOIS for obtaining high-quality solutions. Our results emphasize the potential of SAT solvers combined with LOIS to solve significant game theory problems.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory and mechanism design; Security and privacy → Network security

Keywords and phrases Game Theory, Integer Programming Games, SAT Solvers, Local Solutions, Graph Games

Digital Object Identifier 10.4230/LIPIcs.SAT.2025.19

Supplementary Material *Software*: `https://github.com/PraveshKoirala/SATLois`

1 Introduction

Integer programming games (IPGs) are well-known in game-theoretic literature [7, 26]. They arise when every player is restricted to choosing discrete strategies and are useful for modeling a plethora of scenarios from domains such as transportation, communication, cybersecurity, etc [10, 15, 27]. While they provide substantial modeling flexibility, they are known to be difficult to solve. It is well-known that deciding whether or not an IPG admits a pure Nash equilibria, a popular solution concept for IPGs, is Σ_2^p-complete [8]. This poses severe computational challenges and makes solving for large-scale IPGs intractable.

However, recent works have introduced new class of solution concepts for IPGs called *locally optimal integer solutions* (LOIS) that are more relaxed than PNEs and computationally tractable even for large-scale IPGs [25]. In fact, [25] show that deciding whether IPGs admit LOIS is NP-complete, which means that they can be solved by commercially available mathematical solvers that support integer programming and scale well across large IPG instances. With this relaxed notion of LOIS, many large-scale IPGs that specifically arise in many critical domains, including cybersecurity, become tractable. In particular, in an attacker-defender scenario where there are a large number of cyber infrastructures to be considered with a limited attack/defense budget for both attackers and defenders, the resulting IPG is not readily solvable for large instances using PNE but can indeed be solved using LOIS [25]. Furthermore, a *bilevel* IPG with a leader-follower structure can also be solved by embedding the LOIS optimality conditions of the follower in the leader's program to obtain a single integer program. This allows us to solve an important class of cybersecurity games played on graphs, i.e. *graph interdiction games*, where a defender and an attacker strategically choose nodes to protect and infect while considering the propagation dynamics of the attack on the graph [3]. Despite the scalability of LOIS, they are not without limitations. The most important being that while LOI solutions scale well for lower orders (LOIS-1), higher-order LOI solutions, which are qualitatively better, are time-consuming to calculate due to increased complexity.

SAT solvers that solve the *boolean satisfiability problem* are known to be effective for solving NP-complete problems from various domains including planning, formal verification, cybersecurity, transportation, etc. [2, 20]. Decades of advances in SAT solving techniques, coupled with rigorous optimizations in commercially available SAT solvers make them attractive for solving any large-scale NP-complete problem. Furthermore, the extension of SAT called MaxSAT [28] adds further modeling flexibility to the SAT language, making it possible to perform an *optimization* over the number of true clauses. Despite these powerful modeling and solving capabilities, SAT solvers are underutilized for game-theoretic applications and have not been previously considered for solving IPGs.

In this paper, we show that it is possible to utilize modern SAT solvers to solve for LOI solutions for IPGs using straightforward encodings. In particular, we highlight the fact that modern SAT solvers can scale incredibly well in comparison to traditional mathematical optimizers in certain game-theoretic applications. We apply our methods in two cybersecurity games of interest: a) a critical node game first introduced by [15] where attacker and defender simultaneously choose which infrastructures to attack/defend, and b) a bilevel graph interdiction game with constrained propagation dynamics where defender chooses central nodes to defend followed by the attacker choosing their initial targets to spread infection.

2 Related Works

Local solutions in discrete games. IPGs themself were first investigated by [26] and have found considerable usage since in domains with discrete strategy-space like inventory management, facility location, cybersecurity, etc. [27, 33, 16, 11]. Prevailing methods for solving IPGs focus on finding either pure or mixed Nash equilibrium solutions using a range of algorithms such as branching, cut-and-play, sample generation, zero-regrets etc. [32, 34, 6, 16]. A broad survey of IPGs and related algorithms can be found in [7]. Related to IPGs are also *boolean games* [4, 22], i.e., games where each player's strategy space is composed purely of boolean choices and the players choose assignments with the objective of satisfying their utility (represented by a boolean formula). However, determining the existence of pure Nash equilibria in boolean games is in Σ_2^p as well [4]. The very first work that explored the concept of *local solutions* in discrete strategy games is by [17], where a k-bounded Nash equilibrium is defined for boolean games such that no player can gain utility by unilaterally changing at most k of their strategies. This concept has also been independently explored in a voting game by [29], where they analogously define a *t-local equilibrium* such that no candidate can gain any utility by unilaterally deviating in at most t of their announced positions. A generalization of this boolean local equilibria concept to IPGs was provided more recently by [25], who define a *locally optimal integer solution* to be one where no player can gain by unilaterally deviating in their *integer neighborhood*. Their formulation allows for deriving local optimality conditions for IPGs with shared constraints and is the focus of this work.

SAT solvers in game theory. Usage of SAT solvers in game-theoretic setting is sparse and has generally been restricted to boolean games with boolean utilities. Notably, [12] use *answer set programming* to solve for pure Nash equilibrium of boolean games. [35] apply SAT and MaxSAT solvers in bounded model checking setting to derive Nash equilibrium in a resource allocation problem. An interesting application of satisfiability theory in bimatrix games is proposed in [9], which uses Horn clauses to identify special substructures of a normal form bimatrix game, potentially speeding the process of obtaining Nash equilibrium solutions. An application of SAT on graphical games is by [23] who use it to solve *parity games*, a special type of two-player game played on graphs with the objective of achieving a desired *parity* of minimum node *priority* in an infinite path of play. Similarly, in *reachability games*, SAT solvers have been used to determine and extract winning strategies [18].

Apart from [9], the abovementioned works do not apply to games with non-boolean objectives or constraints, which IPGs specifically allow. And even [9] is an application of SAT to *identify* potential substructures in discrete games that may simplify solutions (and is not a solution method by itself). Therefore, to our knowledge, our work is the first to investigate the application of SAT solvers on IPGs. In particular, we investigate solutions of both simultaneous IPGs (where players make decisions at once) and sequential (or bilevel) IPGs where a leader makes a decision followed by the follower choosing their best response. In summary, our contributions are as follows [1]:

1. We add to the literature of game theory by investigating SAT solvers as an alternative solution method to a class of games of broad interest (i.e., IPGs), and compare their performance against traditional mathematical solvers.
2. We investigate the application of MaxSAT solvers on bilevel graph games and compare the quality of solutions against prevailing heuristics.

Rest of the paper is organized as follows: In section 3 we start with the overview of IPGs, including its definition and related solution concepts. In sections 4 and 5, we introduce the critical node game and the bilevel graph interdiction game and discuss their encoding approaches. In section 6, we discuss timing comparison for mathematical solvers vs SAT solvers for the critical node game and also analyze solution quality of LOIS over heuristics for the bilevel graph game. We briefly recap our work and provide future directions in section 7.

3 Definitions

Throughout the rest of this paper, we follow the conventions and definitions used by [25].

▶ **Definition 1** (IPG). *IPGs with n players is defined as a simultaneous, complete-information, and non-cooperative game in the form of a n-tuple $(P_1, P_2, ...P_n)$, where each P_i is a mathematical program of the form:*

[1] The repository for this paper can be found at `https://github.com/PraveshKoirala/SATLois`

$$P_i := \min_{x^i} \; f^i(x^i; x^{-i})$$
$$\quad s.t. \;\; g^i(x^i) \geq 0, \tag{1}$$

where $x^i \in \mathbb{Z}^{n_i}$ is the strategy of player i, $f^i : \mathbb{Z}^N \mapsto \mathbb{R}$ is their objective function with $N = \sum_i n_i$, and $g^i : \mathbb{Z}^{n_i} \mapsto \mathbb{R}^{m_i}$ are the constraint functions. With $x \in \mathbb{Z}^N$ as the joint strategy space, we use x^{-i} to denote all other decision variables sans x^i.

With definition 1 in place, the pure Nash equilibrium (PNE) of an IPG is defined as:

▶ **Definition 2** (PNE for IPGs). *A joint strategy $x^* \in \mathbb{Z}^N$ is a pure Nash equilibrium for an IPG if and only if $\forall i, g^i(x^*) \geq 0$ and for all players i the following holds:*

$$f^i(x^*) \leq f^i(\tilde{x}^i; (x^*)^{-i}) \quad \forall \tilde{x}^i \in \{x' \in \mathbb{Z}^{n_i} | g^i(x') \geq 0\}$$

Since PNE is computationally prohibitive due to them being in Σ_2^p, [25] introduce the notion of *m-order integer neighborhood* and subsequently *locally optimal integer solution* as follows:

▶ **Definition 3** (m-order integer neighborhood). *For a point $x \in \mathbb{Z}^n$, we define its m-order integer neighborhood as $\mathcal{Z}_m(x) = \{x' \in \mathbb{Z}^n \mid \|x - x'\|_1 \leq m\}$ where the operator $\|\cdot\|_1$ denotes the L1-norm.*

▶ **Definition 4** (Locally optimal integer solution of m-th order). *For a parameterized integer mathematical program $P := \min_{x \in \mathbb{Z}^n} f(x; a)$ s.t. $x \in C$ where the payoff function f is parameterized in $a \in \mathbb{Z}^m$ and C is some feasible set, $\hat{x}^m \in C$ is said to be a locally optimal integer solution of m-th order (LOIS-m) if and only if $f(\hat{x}^m; a) \leq f(x'; a) \;\; \forall x' \in \mathcal{Z}_m(\hat{x}^m) \cap C$.*

By using definition 4, optimality conditions for LOIS can be obtained for a single program as follows:

▶ **Definition 5** (Optimality conditions for LOIS-m). *For a program parameterized in $a \in \mathbb{Z}^p$, where $f : \mathbb{Z}^{n+p} \mapsto \mathbb{R}$ and $\forall j \in 1..J$, $g_j : \mathbb{Z}^n \mapsto \mathbb{R}$, the optimality conditions for $\hat{x}$ to be a LOIS-m can be outlined as the following inequality and implication constraints:*

$$\bigwedge_{j=1}^{J} (g_j(\hat{x}) \geq 0), \tag{2}$$

$$\forall (\hat{x} + \delta \in \mathcal{Z}_m(\hat{x})) \quad \left[f(\hat{x}; a) > f(\hat{x} + \delta; a) \rightarrow \bigvee_{j=1}^{J} (g_j(\hat{x} + \delta) < 0) \right] \tag{3}$$

Finally, using definition 5, the joint optimality condition for a LOIS-m solution for an IPG is defined as:

▶ **Definition 6** (LOIS-m solution for IPGs). *We recall IPGs to be the joint mathematical program $(P_1, ...P_n)$ where each program P_i is parameterized in x^{-i}. For each player i, let $\mathcal{O}_m^i$ be the set of points satisfying their LOIS-m optimality conditions as introduced in definition 5. Then, we define $\hat{x}$ as the LOIS-m solution for the IPG iff:*

$$\hat{x} \in \bigcap_{i=1}^{n} \mathcal{O}_m^i \tag{4}$$

In simpler terms, a *LOIS-m* solution for an IPG is defined such that no single player, by unilaterally deviating in their m-order integer neighborhood, can obtain better payoffs (or lower costs) while remaining feasible. In this way, LOIS-m basically offers a direct adaptation of the concept of *local equilibria* in continuous-strategy games to IPGs. Similarly, when the discrete strategy-set of each player is binary (and the payoff is a logical formula defined over the joint strategy), it is easy to see that LOIS-m for *unconstrained* IPG in such cases translates to the notion of local equilibria for boolean games. In this regard, LOIS-m is actually a generalization of existing local solution concepts from boolean games.

4 Critical Node Game (CNG)

CNG is a cybersecurity game first introduced by [15]. The following description of CNG is derived from the same and we refer readers to the original text for any missing details. The CNG is played over a set of V critical nodes representing sensitive digital infrastructures with the decision variables $x_i \in \{0,1\}, \alpha_i \in \{0,1\}\ \forall i \in V$ indicating respectively the defender and attacker's choice with $x_i = 1$ for defender defending the node i (and zero otherwise), and $\alpha_i = 1$ for the attacker attacking the target (and zero otherwise). Both players have limited *budget* and there are strategic interactions between their choices. The complete 2-player CNG is specified as a simultaneous, non-cooperative, and complete-information game comprised of the programs (P_A, P_D) such that the attacker solves:

$$P_A := \max_{\alpha \in \{0,1\}^{|V|}} \ f^a(\alpha; x) \qquad s.t.\ a^T\alpha \leq A$$

and the defender solves:

$$P_D := \max_{x \in \{0,1\}^{|V|}} \ f^d(x; \alpha) \qquad s.t.\ d^T x \leq D$$

The functions $f^a : \mathbb{Z}^{2|V|} \mapsto \mathbb{R},\ f^d : \mathbb{Z}^{2|V|} \mapsto \mathbb{R}$ are the payoffs of the attacker and the defender respectively. Each attack or defend choice has a cost represented by the vectors $a \in \mathbb{Z}_+^{|V|}, d \in \mathbb{Z}_+^{|V|}$. The total budget for attacker / defender, respectively, is $A \in \mathbb{Z}_+, D \in \mathbb{Z}_+$ and $p_i^d \in \mathbb{Z}_+$, $p_i^a \in \mathbb{Z}_+$ denote the *criticality* of node i for respectively, the defender and the attacker. This formulation varies slightly from the original, as it restricts the costs, criticality, and budget to be integers instead of reals. But as we show later, our encoding scheme can be suitably modified to account for real costs, criticality, and budget as well. The variables $0 \leq \delta, \eta, \epsilon, \gamma \leq 1$ are then chosen as *real-valued* scalar parameters of a CNG with $\delta < \eta < \epsilon$. The full payoff functions for attacker and defender are described in detail in appendix A

4.1 LOIS for CNG

We outline the LOIS-1 optimality conditions for CNG. Optimality conditions for LOIS-m $(m > 1)$ are analogously defined. $(\hat{\alpha}, \hat{x})$ are LOIS-1 solutions for CNG iff the following hold.

For the attacker,

$$\begin{aligned} a^T\hat{\alpha} &\leq A \\ 0 \leq \hat{\alpha}_i &\leq 1, \ \forall i \in V \\ \Delta f_i^a > 0 &\to (\hat{\alpha}_i > 0) \vee (a^T\hat{\alpha} + a_i\hat{\alpha}_i > A), \ \forall i \in V \\ \Delta f_i^a < 0 &\to (\hat{\alpha}_i < 1) \vee (a^T\hat{\alpha} - a_i\hat{\alpha}_i > A), \ \forall i \in V \\ \text{where, } \Delta f_i^a &= p_i^a(1 + \gamma + \hat{x}_i(\gamma - \eta)) \end{aligned} \tag{5}$$

For the defender,

$$\begin{aligned} d^T\hat{x} &\leq D \\ 0 \leq \hat{x}_i &\leq 1, \ \forall i \in V \\ \Delta f_i^d > 0 &\to (\hat{x}_i > 0) \vee (d^T\hat{x} + d_i\hat{x}_i > D), \ \forall i \in V \\ \Delta f_i^d < 0 &\to (\hat{x}_i < 1) \vee (d^T\hat{x} - d_i\hat{x}_i > D), \ \forall i \in V \\ \text{where, } \Delta f_i^d &= p_i^d(\epsilon - 1 + \hat{\alpha}_i(\eta + 1 - \epsilon - \delta)) \end{aligned} \tag{6}$$

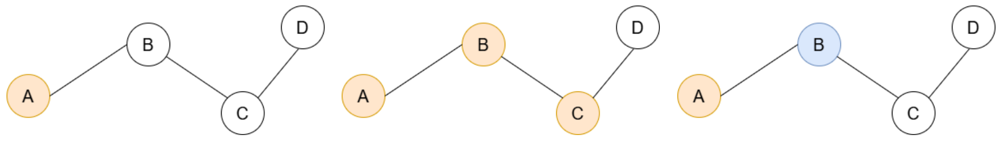
Conditions 5 and 6 when jointly solved, give the LOIS-1 solution $(\hat{\alpha}, \hat{x})$.

4.2 SAT encodings for CNG

The variables for player choices $(\hat{\alpha}, \hat{x})$, attack and defense budgets (A, D), and priority values for each node for attacker and defender (p_i^a, p_i^d) in equations 5 and 6 are integers. For SAT solvers to be able to solve formulas with integer variables, they have to be converted into equivalent fixed-width binary representation, i.e., *bit-vectors* with the process colloquially known as *bit-blasting*. Once equivalent bit-vector representations are obtained, regular arithmetic operations like addition, subtraction, multiplication, and division can be performed using binary circuits. For the parameters $\epsilon, \eta, \delta, \gamma$ that are reals, we encode each of them as ratios a/b where each a, b is a bit-vector of sufficient size. It can be seen that the choice of size needs to be balanced as a larger bit-vector size will result in increased memory and computation, whereas a smaller bit-vector size may result in poor results. Similarly, while we have considered the criticality values, costs, and budgets to be integers, we can substitute them with analogous bit-vector ratios to add support for reals. Once all the variables have been properly encoded, since the operations on them are a combination of simple logical and arithmetic operators (inequality, disjunction, implication, etc.), natively supported by most commercial SAT solvers, these do not require further specialized encoding.

5 Bilevel graph interdiction

We now describe the bilevel graph interdiction game with two players attacker and the defender, with the defender as the leader. The following description of the game is a simplified version of the trilevel game introduced in [31] but with a constrained propagation dynamics. The game is played over an undirected graph $G = (V, E)$ where each node $v \in V$ is a critical digital infrastructure and each edge $(v_1, v_2) \in E$ represents a network connection between v_1, v_2. The objective of the defender (which moves first) is to choose $n_D \in \mathbb{Z}_+$ nodes to *protect*, followed by the attacker that chooses $n_A \in \mathbb{Z}_+$ nodes to *infect* with malware. We assume a constrained propagation model where an attacked node can propagate the malware to *undefended* nodes at a distance of at most r before being detected, making them unsafe, with the distance between a node a and b being defined as the number of edges in the shortest path between a and b. Figure 1 further explains the setup. The full formulation of the problem and its equivalent reduction to a maxSAT form is provided in Appendix B.



(a) Undefended T=0. **(b)** Undefended T=∞. **(c)** Defended T=0/∞.

Figure 1 For a propagation radius $r = 2$. In (a) A is first attacked (orange) and the attack propagates to B and C as shown in (b). D remains safe. In (c), A is first infected but the presence of a defended node B (blue) deters further propagation making B, C, D safe. If there were an edge (A,C) in (c), both C and D would have been eventually infected and B would be the only safe node.

6 Experiments

6.1 CNG

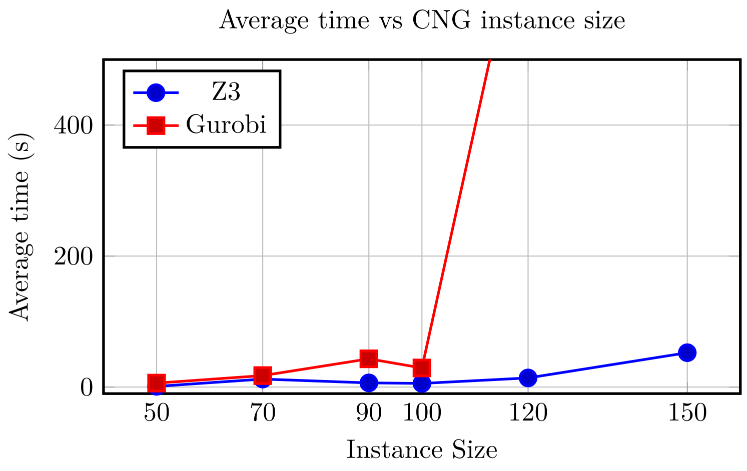


Figure 2 Performance of SAT solver (Z3) and mathematical solver (Gurobi) on CNG instances.

In this experiment, we compare the time taken to obtain higher-order LOIS-2 solutions for increasing CNG instances by popular mathematical program solvers versus a SAT solver. In particular, we choose three mathematical solvers, i.e., HiGHS [24], SCIP [1], and Gurobi [21], to compare the SAT solver of our choice, i.e,. Z3 [13]. We conduct this experiment in two steps. In the preliminary step, we compare the average solution time of HiGHS, SCIP, and Gurobi on 10 random instances of CNG of size 50 generated according to the procedure outlined in [15]. We found the average time for each solver to be HiGHS (991.4s), SCIP(133.2s), and Gurobi(6.05s). As we can see, Gurobi clearly outperforms all other solvers by a large margin. Therefore, in the next step, we exclusively compare Gurobi against Z3 for LOIS-2. We used a bit-vector width of 16 for integer encodings. The results for this step are presented in figure 2. As we can see, Z3 clearly outperforms Gurobi on all instance sizes. Primarily, as the instance size passes a certain threshold, the performance gap becomes starkly visible with Z3 taking an average time of 13.99s for an instance of size 120, whereas Gurobi takes 765.2s for the same.

6.2 Bilevel graph interdiction

For this experiment, we compare the *quality* of results obtained using LOIS-1 solutions against those obtained using prevailing heuristics. In particular, for graphs, metrics like *centrality* measure how important a node is for propagation [14]. Therefore, a heuristic-based method for the bilevel graph interdiction problem is for the defender to defend the most *central* nodes with the highest centrality values, followed by the attacker choosing its best response. We compare the LOIS-1 solution against popular centrality measures in graphs, including degree centrality, eigenvalue centrality, closeness centrality, and betweenness centrality [14]. We also compare with an additional game-theoretic centrality metric known as the *shapley centrality* [30] that is specifically developed for identifying nodes most important for propagation. We consider a modestly sized graph with 30 nodes. We set the defender's budget $n_D = 4$, attacker's budget $n_A = 2$, propagation radius $r = 4$, and evaluate our methods on four kinds of randomly generated graphs i.e. *MST, Line, Edge30*, and *Ring*, where *MST* instances are random minimum spanning tree defined over the nodes, *Edge30* instances contain exactly 30 random edges, *Line* is a path graph [5], and *Ring* is a cycle graph [19]. All solutions are obtained by using the maxSAT solver from Z3 within the time limit of 5 minutes.

Table 1 Number of safe nodes in bilevel graph interdiction game. LOIS-1-based method outperforms all heuristics in every graph variant. Results are averaged over 20 runs for each graph type.

Methods	**Line**	**MST**	**Ring**	**Exact30**
Betweenness	12	13.8	12	14.75
Closeness	12	13.9	12	13.8
Degree	12	17.35	12	16.75
Eigenvalue	12	13.3	12	13.65
Shapley	12	13.85	12	15.25
LOIS-1	**16.55**	**19.5**	**13.7**	**17.65**

As we can see from the results, when central nodes are obtained using the maxSAT solution of the defender with embedded LOIS-1 optimality constraints of the attacker, it outperforms all popular centrality heuristics. This difference is particularly stark when there are symmetries present in the graph, i.e. *line, ring.* We hypothesize that for symmetric graphs, since all nodes have the same centrality, heuristic-based methods fail to identify key nodes to protect.

7 Conclusion

In this study, we showed that by using local solution concepts like the LOIS, it is possible to solve for IPGs using SAT solvers in both simultaneous and sequential discrete-strategy games. In particular, we note that the encodings required for solving these games are straightforward and are supported out-of-the-box by most commercially available SAT solvers. We also showed that these SAT solvers may be more efficient in terms of solution time when compared against state-of-the-art mathematical solvers like Gurobi, and the game theory community should definitely consider these solvers for large-scale discrete games as an alternative to mathematical program solvers. Finally, we also showed that in graph games, maxSAT solvers in conjunction with LOI solutions can provide better quality solutions with respect to heuristic-based solutions, even for tricky problem instances.

We note the following limitations of our study. First and foremost, only local solution concepts like LOIS remain solvable by SAT solvers. Similarly, games with non-linear objectives or constraints that are not encodable by binary circuits may not be possible to solve. In the bilevel graph interdiction example, although we were able to solve for moderate graph instances, we found scaling it to large graph sizes challenging. Future work seeking to expand this line of research should focus on global solution concepts (or higher-order LOI solutions) for IPGs using SAT and finding novel ways to scale graph games when solved using MaxSAT.

References

1 Tobias Achterberg. Scip: solving constraint integer programs. *Mathematical Programming Computation*, 1:1–41, 2009. doi:10.1007/S12532-008-0001-1.

2 Sahel Alouneh, Sa'ed Abed, Mohammad H Al Shayeji, and Raed Mesleh. A comprehensive study and analysis on sat-solvers: advances, usages and achievements. *Artificial Intelligence Review*, 52:2575–2601, 2019. doi:10.1007/S10462-018-9628-0.

3 Andrea Baggio, Margarida Carvalho, Andrea Lodi, and Andrea Tramontani. Multilevel approaches for the critical node problem. *Operations Research*, 69(2):486–508, 2021. doi:10.1287/OPRE.2020.2014.

4 Elise Bonzon, Marie-Christine Lagasquie-Schiex, Jérôme Lang, and Bruno Zanuttini. Boolean games revisited. In *ECAI*, volume 141, pages 265–269, 2006. URL: http://www.booksonline.iospress.nl/Content/View.aspx?piid=1688.

5 Haitze J Broersma and Cornelis Hoede. Path graphs. *Journal of graph theory*, 13(4):427–444, 1989. doi:10.1002/JGT.3190130406.

6 Margarida Carvalho, Gabriele Dragotto, Andrea Lodi, and Sriram Sankaranarayanan. The cut-and-play algorithm: Computing nash equilibria via outer approximations. *arXiv preprint arXiv:2111.05726*, 2021. arXiv:2111.05726.

7 Margarida Carvalho, Gabriele Dragotto, Andrea Lodi, and Sriram Sankaranarayanan. Integer programming games: a gentle computational overview. In *Tutorials in Operations Research: Advancing the Frontiers of OR/MS: From Methodologies to Applications*, pages 31–51. INFORMS, 2023.

8 Margarida Carvalho, Andrea Lodi, and João P Pedroso. Computing equilibria for integer programming games. *European Journal of Operational Research*, 303(3):1057–1070, 2022. doi:10.1016/J.EJOR.2022.03.048.

9 Vincent Conitzer and Tuomas Sandholm. A technique for reducing normal-form games to compute a nash equilibrium. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 537–544, 2006. doi:10.1145/1160633.1160731.

10 Tobias Crönert and Stefan Minner. Location selection for hydrogen fuel stations under emerging provider competition. *Transportation Research Part C: Emerging Technologies*, 133:103426, 2021.

11 Tobias Crönert and Stefan Minner. Equilibrium identification and selection in finite games. *Operations Research*, 72(2):816–831, 2024. doi:10.1287/OPRE.2022.2413.

12 Sofie De Clercq, Kim Bauters, Steven Schockaert, Mihail Mihaylov, Ann Nowé, and Martine De Cock. Exact and heuristic methods for solving boolean games. *Autonomous agents and multi-agent systems*, 31:66–106, 2017. doi:10.1007/S10458-015-9313-5.

13 Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340. Springer, 2008.

14 Paramita Dey, Subhayan Bhattacharya, and Sarbani Roy. A survey on the role of centrality as seed nodes for information propagation in large scale network. *ACM/IMS Transactions on Data Science*, 2(3):1–25, 2021. doi:10.1145/3465374.

15 Gabriele Dragotto, Amine Boukhtouta, Andrea Lodi, and Mehdi Taobane. The critical node game. *Journal of Combinatorial Optimization*, 47(5):74, 2024. doi:10.1007/S10878-024-01173-3.

16 Gabriele Dragotto and Rosario Scatamacchia. The zero regrets algorithm: Optimizing over pure nash equilibria via integer programming. *INFORMS Journal on Computing*, 35(5):1143–1160, 2023. doi:10.1287/IJOC.2022.0282.

17 Paul E Dunne and Michael Wooldridge. Towards tractable boolean games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 939–946, 2012. URL: http://dl.acm.org/citation.cfm?id=2343831.

18 Niklas Eén, Alexander Legg, Nina Narodytska, and Leonid Ryzhyk. Sat-based strategy extraction in reachability games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29(1), 2015.

19 Severino V Gervacio. Cycle graphs. In *Graph Theory Singapore 1983: Proceedings of the First Southeast Asian Graph Theory Colloquium, held in Singapore May 10–28, 1983*, pages 279–293. Springer, 1984.

20 Weiwei Gong and Xu Zhou. A survey of sat solver. In *AIP Conference Proceedings*, volume 1836(1). AIP Publishing, 2017.

21 Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024. URL: https://www.gurobi.com.

22 Paul Harrenstein, Wiebe van der Hoek, John-Jules Meyer, and Cees Witteveen. Boolean games. In *Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 287–298, 2001.

23 Keijo Heljanko, Misa Keinänen, Martin Lange, and Ilkka Niemelä. Solving parity games by a reduction to sat. *Journal of Computer and System Sciences*, 78(2):430–440, 2012. doi:10.1016/J.JCSS.2011.05.004.

24 Qi Huangfu and JA Julian Hall. Parallelizing the dual revised simplex method. *Mathematical Programming Computation*, 10(1):119–142, 2018. doi:10.1007/S12532-017-0130-5.

25 Pravesh Koirala, Mel Krusniak, and Forrest Laine. Locally optimal solutions for integer programming games. *arXiv preprint arXiv:2503.20918*, 2025. doi:10.48550/arXiv.2503.20918.

26 Matthias Köppe, Christopher Thomas Ryan, and Maurice Queyranne. Rational generating functions and integer programming games. *Operations research*, 59(6):1445–1460, 2011. doi:10.1287/OPRE.1110.0964.

27 Alejandro Lamas and Philippe Chevalier. Joint dynamic pricing and lot-sizing under competition. *European Journal of Operational Research*, 266(3):864–876, 2018. doi:10.1016/J.EJOR.2017.10.026.

28 Chu Min Li and Felip Manya. Maxsat, hard and soft constraints. In *Handbook of satisfiability*, pages 903–927. IOS Press, 2021. doi:10.3233/FAIA201007.

29 Javier Maass, Vincent Mousseau, and Anaëlle Wilczynski. A hotelling-downs game for strategic candidacy with binary issues. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 2076–2084, 2023. doi:10.5555/3545946.3598881.

30 Tomasz P Michalak, Karthik V Aadithya, Piotr L Szczepanski, Balaraman Ravindran, and Nicholas R Jennings. Efficient computation of the shapley value for game-theoretic network centrality. *Journal of Artificial Intelligence Research*, 46:607–650, 2013. doi:10.1613/JAIR.3806.

31 Adel Nabli, Margarida Carvalho, and Pierre Hosteins. Complexity of the multilevel critical node problem. *Journal of Computer and System Sciences*, 127:122–145, 2022. doi:10.1016/J.JCSS.2022.02.004.

32 Simone Sagratella. Computing all solutions of nash equilibrium problems with discrete strategy sets. *SIAM Journal on Optimization*, 26(4):2190–2218, 2016. doi:10.1137/15M1052445.

33 Simone Sagratella, Marcel Schmidt, and Nathan Sudermann-Merx. The noncooperative fixed charge transportation problem. *European Journal of Operational Research*, 284(1):373–382, 2020. doi:10.1016/J.EJOR.2019.12.024.

34 Stefan Schwarze and Oliver Stein. A branch-and-prune algorithm for discrete nash equilibrium problems. *Computational Optimization and Applications*, 86(2):491–519, 2023. doi:10.1007/S10589-023-00500-4.

35 Nils Timm, Josua Botha, and Steven Jordaan. Max-sat-based synthesis of optimal and nash equilibrium strategies for multi-agent systems. *Science of Computer Programming*, 228:102946, 2023. doi:10.1016/J.SCICO.2023.102946.

A Payoff description of CNG

The full payoffs for four distinct scenarios are taken as:

1. **Normal operation ($x_i = 0, \alpha_i = 0$):** Defender gets full payoff p_i^d, while attacker incurs opportunity cost γp_i^a.
2. **Successful attack ($x_i = 0, \alpha_i = 1$):** Attacker obtains full payoff p_i^a. Defender obtains reduced δp_i^d.
3. **Successful defense ($x_i = 1, \alpha_i = 0$:** Defender obtains reduced ϵp_i^d. Attacker obtains 0.
4. **Attack and defense ($x_i = 1, \alpha_i = 1$):** Attacker receives $(1-\eta)p_i^a$. Defender receives ηp_i^d.

   The full attacker payoff is then obtained as:

$$f^a(\alpha; x) = \sum_{i \in V} (-\gamma p_i^a (1-x_i)(1-\alpha_i) + p_i^a (1-x_i)\alpha_i + (1-\eta)p_i^a x_i \alpha_i)$$

   Similarly, the full defender payoff is:

$$f^d(x; \alpha) = \sum_{i \in V} \big(p_i^d (1-x_i)(1-\alpha_i) + \delta p_i^d (1-x_i)\alpha_i + \epsilon p_i^d x_i (1-\alpha_i) + \eta p_i^d x_i \alpha_i\big)$$

B Formulation of bilevel interdiction game

For each node $v \in V$, we use the binary variables $a_v \in \{0,1\}^{|V|}, d_v \in \{0,1\}^{|V|}, s_v \in \{0,1\}^{|V|}$ to represent if the node is attacked, defended, or safe. The bilevel program to be solved for by the leader (defender) can then be outlined as follows[2].

[2] For ease of understanding, we use boolean operations in constraints. These can be equivalently translated to integer constraints for an IPG.

$$\max_{d,a} \sum_{i \in V} s_v \tag{7a}$$

$$s.t. \sum_{i \in V} d_v \leq n_D \tag{7b}$$

$$a \in \min_{\tilde{a}} \sum_{i \in V} s_v \tag{7c}$$

$$s.t. \sum_{i \in v} \tilde{a}_v \leq n_A \tag{7d}$$

$$d_i \to \neg \tilde{a}_i \quad \forall i \in V \tag{7e}$$

$$s_i^0 = \neg \tilde{a}_i \quad \forall i \in V \tag{7f}$$

$$s_i^{t+1} = d_i \vee \left(\forall i' \in \mathcal{N}_G(i),\ s_{i'}^t \right) \forall t \in \{0 ... r-1\} \tag{7g}$$

$$s_i = s_i^r \tag{7h}$$

The program on the top level (defender), maximizes the total number of safe nodes by choosing at-most n_D nodes (7b) to defend subject to the fact that the attacker's program chooses at-most n_A nodes to attack (7d). Attacker can not attack a defended node (7e). Initially, only attacked nodes are unsafe (7f) but depending upon the propagation radius r, the final safety status of all nodes are calculated (7g, 7h). The set $\mathcal{N}_G(v)$ in equation 7g denotes all the immediate neighbors of node v in the graph. It's easy to see that the attacker's program is an IPG parameterized in d. Let $\mathcal{A}(d)$ be the set of attacker strategies that satisfy the LOIS-1 optimality of the parameterized program, then the bilevel program can be reformulated as:

$$\max_{d,a} \sum_{i \in V} s_v \tag{8a}$$

$$s.t. \sum_{i \in V} d_v \leq n_D \tag{8b}$$

$$a \in \mathcal{A}(d) \tag{8c}$$

$$\tag{8d}$$

The resulting single-level IPG consists of boolean and integer variables with pseudo-boolean constraints. The objective can also be encoded as a weighted MaxSAT [28] constraint. Therefore, the entire program can be easily solved for using a commercially available MaxSAT solver.