CS490 Project Report: Interdictions And Blotto Games

Anirudh Garg
Roll Number: 22B1005
Undergraduate B.Tech. CSE Student
Department of Computer Science and Engineering
IIT Bombay

May 9, 2024

Project Guide: Prof. Swaprava Nath
External Co-Guides: Prof. Avinash Bhardwaj, Prof. Sriram Sankaranarayanan

1 Phase - I

First meet conducted on Jan 26, 2024.
Discussion on getting introduced to Bilevel Programming

1.1 Papers - Jan 30, 2024

1. A Survey on Mixed-Integer Programming Techniques in Bilevel Optimization
2. The Standard Pessimistic Bilevel Problem
3. Bilevel programming and price setting problems
4. A Review on Bilevel Optimization: From Classical to Evolutionary Approaches and Applications

1.2 Comments

I referred to [4] here before the meeting for personal gain in the various terminologies and formulation of Bilevel problems.
After the meeting, I read [1] and got myself familiar with applications of bi-level programming and the general properties.
Coupling constraints were discussed along with how they may lead to disconnected bilevel feasible region and how they cannot be moved to the lower level without changing the set of optimal solutions, something that wasn't directly obvious to me.
The complexity of it was also discussed here. Then I looked at Stackelberg bimatrix games. But without examples to see where the theory was being applied, I couldn't fully understand beyond this point.
Then I looked at [2] but couldn't get too far into it.
Then, the second meeting took place.

2 Phase - II

Discussion on Other algorithms for Bilevel Programming and looking at papers with more intuitive and exemplified theory which were provided to me over the mail without any formal meeting.

2.1 Papers - Feb 13, 2024

1. The Watermelon Algorithm for The Bilevel Integer Linear Programming Problem
2. Bilevel programming and price setting problems
3. The Mixed Integer Linear Bilevel Programming Problem by J.Moore and J.Bard
4. A Branch-and-cut Algorithm for Integer Bilevel Linear Programs

2.2 Comments

The main focus in this phase was to understand the algorithm given in [1] and what I have gained from it. The paper starts with the usual theory on formulations, bounds we get from relaxation and other prerequisites. This paper exposed me to one of the most prominent examples in this domain as well :-

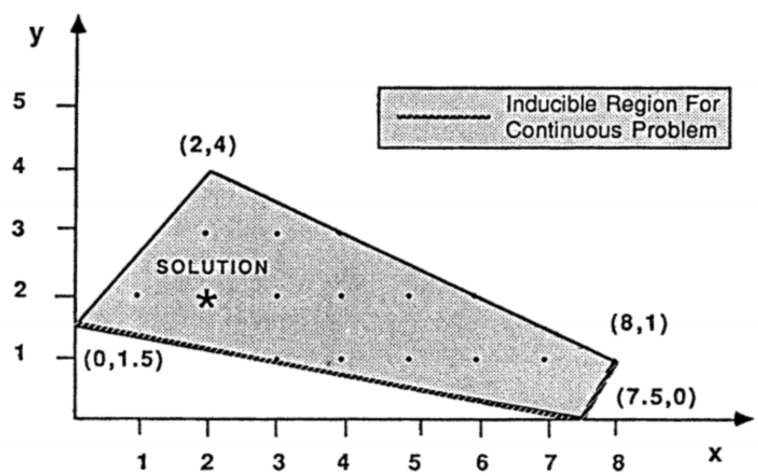
$$\underset{x}{\text{Maximize}}\ F(x, y) = x + 10y$$

where y solves

$$\max_{y} f(x, y) = -y$$

$$\begin{aligned}
\text{subject to}\quad & -25x + 20y \leqslant 30 \\
& x + 2y \leqslant 10 \\
& 2x - y \leqslant 15 \\
& 2x + 10y \geqslant 15 \\
& x, y \geqslant 0 \\
& x, y \text{ integer.}
\end{aligned}$$

(a) Src: [3]



(b) Src: [3]

For learning more about this example, I referred to [3] as well.
In this example, optimizing over the continuous region yields the integer solution (8,1) with the upper level objective value 18. However, the true solution is (2,2) with objective value 22. This example helps in learning 2 things :-

1. The solution of the relaxed Bilevel Problem does not provide a valid bound on the solution of the mixed integer Bilevel Problem
2. Solution to the relaxed Bilevel Problem that are in the inducible region cannot, in general, be fathomed

So what the watermelon algorithm does is solve the relaxation, if the solution is bilevel feasible then it is also bilevel optimal otherwise it must be removed from the feasible set.
The process of removal of it is what makes this algorithm interesting. As this point must be a corner so there will be some constraints that reach equality on that point (termed as the watermelon seed which needs to be scooped out). So, what we do is take all possibilities of changing the one constraint that leads to the elimination of that point and add all these new problems (new set of constraints) to the set of the bilevel problems we are going to solve basically. This is because removing all of them would lead to the resulting feasible region becoming non-convex in some cases.
Even though the blowup would be exponential but looking at the complexity of the initial problem at hand this as it turns out is a favorable trade-off. There are also conditions on when to stop looking for a solution which happens when we are able to conclude that the problem is infeasible or unbounded.
The paper also gives the formulation of the problems to check the existence of these points (Type-2 scoop) where the second formulation gives much better results than the first one as the watermelon seeds are also selected optimally to eliminate as many points as possible.
The proof of finite termination is quite trivial as in each step you eliminate at least one integral point and there are only finitely many points.
In [2], I looked at the price-solving problem which was given as a reference in [1] but only up to the first three sections which included the Bilevel programming background and the following price setting problem :-

In a price setting problem the leader (first level) sets some taxes or prices for some activities, and the followers (second level) select activities from among taxed and untaxed ones to minimize operating costs. We assume that there are n_1 taxed and n_2 untaxed activities. By setting $T \in \mathbb{R}^{n_1}$ as the tax vector, $x \in \mathbb{R}^{n_1}$ and $y \in \mathbb{R}^{n_2}$ as the vectors associated with taxed and untaxed activities respectively, f and g as the objective functions of the leader and the follower respectively, and $\Pi \subset \mathbb{R}^{n_1+n_2}$ as the feasible solution set, the price setting problem can be formulated as:

$$\max_{T} \quad f(T, x, y), \tag{2a}$$

$$\text{s.t.} \quad (x, y) \in \arg\min_{x,y} g(T, x, y), \tag{2b}$$

$$(x, y) \in \Pi. \tag{2c}$$

Figure 2: Src: [2]

[4] was another paper given as a reference in [1] which is also discussed in their results, about how the watermelon algorithm is applicable in more cases and the comparison between the speeds of both the algorithms. Specifically, the branch-and-cut algorithm in [4] can only be applied to Integer Bilevel Linear

Problems whereas the watermelon algorithm in [1] can even be applied to Mixed Integer Bilevel Linear Programs.

The paper discusses a very simple algorithm, they solve the relaxation (As removing the integrality constraints does not result in valid relaxation, they instead remove the lower-level optimality constraint which yields a valid relaxation, proved in another paper) of the current problem and if the solution you get is not bilevel feasible then they branch to more problems that consider a selected variable being more or less than its current value. So the problem forms a tree-like structure for representation.

3 Phase - III

Second meet conducted on Feb 26, 2024.

I presented the watermelon algorithm and my understanding of the bilevel programming problem thus far was evaluated. Based on that, the following papers were given and I was asked which one I wanted to pursue further.

3.1 Papers - Feb 27, 2024

1. Multilevel Approaches for the Critical Node Problem
2. A new exact approach for the Bilevel Knapsack with Interdiction Constraints

I chose [1] for further reading.

3.2 Comments

This paper introduces a 2-player tri-level optimization problem where there is a graph and the defender first vaccinates few nodes according to a budget, the attacker attacks a few nodes, and then the defender again protects a few nodes. The aim is to prevent the attack from spreading as much as possible.

The paper presents a very unintuitive strategy where they optimize the attack strategy for the attacker for all vaccinations over a subset of the protecting strategies. Then if the attack strategy so obtained holds good over the complete set then we are done, else the new protect strategy that does well is added to the previous subset and we start over. In the results, a peculiar pattern we observe is that increasing the budget for the second level slows the computation more than increasing the budget in the first or third level.

$$z_v = \begin{cases} 1 & \text{if } v \text{ is vaccinated} \\ 0 & \text{otherwise} \end{cases}$$

$$x_v = \begin{cases} 1 & \text{if } v \text{ is protected} \\ 0 & \text{otherwise} \end{cases}$$

$$y_v = \begin{cases} 1 & \text{if } v \text{ is attacked} \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha_v = \begin{cases} 1 & \text{if } v \text{ is saved} \\ 0 & \text{otherwise} \end{cases}$$

Note that a node $v \in V \setminus I$ is saved if it has been vaccinated, protected, or all of its adjacent nodes are saved. In this way, it is possible to determine if a node is saved by only restricting it to constraints that take into account the state of adjacent nodes. We can then formulate the MCN problem as the following mixed-integer trilevel program:

$$\max_{\substack{z \in \{0,1\}^V \\ \sum_{v \in V} z_v \leq \Omega}} \; \min_{\substack{y \in \{0,1\}^V \\ \sum_{v \in V} y_v \leq \Phi}} \; \max_{\substack{x \in \{0,1\}^V \\ \alpha \in [0,1]^V}} \sum_{v \in V} \alpha_v$$

$$\sum_{v \in V} x_v \leq \Lambda$$

$$\alpha_v \leq 1 + z_v - y_v \quad \forall v \in V$$

$$\alpha_v \leq \alpha_u + x_v + z_v \, \forall \, (u,v) \in A.$$

Figure 3: Src: [1]

The last constraint is quite interesting here where we implement the logic to check if all of its adjacent nodes are either previously saved, protected, or vaccinated which also makes use of the fact that we are maximizing the sum of α_v. For example, consider a group of 3 interconnected nodes which are surrounded by protected nodes. Here α_v can be 0 or 1 and no constraints will be violated but maximizing leads to the correct solution. Alternatively, you can just as well minimize the number of attacked nodes which will be computationally more efficient based on my knowledge.

4 Phase - IV : Next Step

After going over this paper I started to think of some variant of the MCN (Multilevel Critical Node) Problem which can be explored. I considered some variants like :-

1. Making the budgets uncertain with a probability distribution on them. But this proves very trivial as with as small budgets as discussed in the paper which were in 1-3 range at each level you can just replace "maximising the sum of α_v" with "maximising the expectation of the sum of α_v" and you will get the desired result with the same time complexity.

2. The authors in [1] have used the fence problem in the protection phase where the defender gets a one-time budget which can be used to save that many nodes. This can be replaced by the firefighter problem where the infection propagates through the network from a node to any adjacent node at every time step and at any following time point the defender is allowed to protect some non-infected nodes in order to maximise the saved nodes at the end of the process. The fence problem is just a special case of this one where the defender has non zero budget only at the first time step.

3. Edge protection instead of vertex protection.

But the variant I selected was something that had initially motivated me to work on Bilevel programs. In real life situations the attacker doesn't know where the vaccination has been done while the third level remains the same. So the first two levels are in-effect played simultaneously.
Upon discussion in the meeting I was made aware of the fact that this simultaneous game comes under Blotto Games and what I am looking to do is work on what are essentially bilevel games (interdiction here) where the first level is a blotto game and the next level an optimisation problem.
To gain more knowledge on the subject I have been Provided the folowing papers where [1-5] concern Interdictions whereas [6-8] deal with Blotto Games:-

4.1 Papers - April 1, 2024

1. The Shortest Path Interdiction Problem with Randomized Interdiction Strategies: Complexity and Algorithms — Operations Research

2. A survey of network interdiction models and algorithms - EJOR

3. Interdiction Games and Monotonicity, with Application to Knapsack Problems

4. Bilevel Knapsack with Interdiction Constraints

5. A Branch-and-Cut Algorithm for Submodular Interdiction Games

6. From Duels to Battlefields: Computing Equilibria of Blotto and Other Games — Mathematics of Operations Research

7. Fast and Simple Solutions of Blotto Games — Operations Research

8. An Algorithmic Solution to the Blotto Game Using Multimarginal Couplings — Operations Research

4.2 Comments

Right now, I have only looked at [6], specifically on its section-2. Up to this point, I have only encountered formulations and the background theory required for Blotto games.
To be resumed after End semester exams.