New Results for Network Pollution Games*

Eleftherios Anastasiadis[1], Xiaotie Deng[2], Piotr Krysta[1],
Minming Li[3], Han Qiao[4], and Jinshan Zhang[1]

[1] Dept. of Computer Science, University of Liverpool, UK, {e.anastasiadis, p.krysta, jinshan.zhang}@liverpool.ac.uk
[2] Dept. of Computer Science & Engineering, Shanghai Jiao Tong University, China deng-xt@cs.sjtu.edu.cn
[3] Dept. of Computer Science, City University of Hong Kong, China minming.li@cityu.edu.hk
[4] School of Management, University of Chinese Academy of Sciences, China qiaohan@ucas.ac.cn

Abstract. We study a newly introduced network model of the pollution control and design approximation algorithms and truthful mechanisms with objective to maximize the social welfare. On a high level, we are given a graph whose nodes represent the agents (sources of pollution), and edges between agents represent the effect of pollution spread. The government is responsible to maximize the social welfare while setting bounds on the levels of emitted pollution both locally and globally. We obtain a truthful in expectation FPTAS when the network is a tree (modelling water pollution) and a deterministic truthful 3-approximation mechanism. On planar networks (modelling air pollution) the previous result was a huge constant approximation algorithm. We design a PTAS with a small violation of local pollution constraints. We also design approximation algorithms for general networks with bounded degree. Our approximations are near best possible under appropriate complexity assumptions.

Keywords: Algorithmic mechanism design, approximation algorithms, planar and tree networks

1 Introduction

Environmental degradation accompanies the advance in technology, resulting in global water and air pollution. As an example, in 2012, China discharged 68.5 billion tons of industrial wastewater, and the SO_2 emissions reached 21.2 million tons (National Bureau of Statistics of China, 2013). The recent annual State of the Air report of the American Lung Association finds 47% of Americans live in counties with frequently unhealthy levels of either ozone or particulate pollution [2]. The latest assessment of

* X. Deng was supported by the National Science Foundation of China (Grant No. 61173011) and a Project 985 grant of Shanghai Jiaotong University. P. Krysta and J. Zhang were supported by the Engineering and Physical Sciences Research Council under grant EP/K01000X/1. M. Li was partly supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 117913). H. Qiao was supported by the National Science Foundation of China (Grant No. 71373262).

air quality, by the European Environment Agency, finds that around 90% of city inhabitants in the European Union are exposed to one of the most damaging air pollutants at harmful levels [1]. Environmental research suggests that water pollution is on of the very significant factor affecting water security worldwide [19]. It is the role of regulatory authorities to make efficient environmental pollution control policies in balancing economic growth and environment protection.

We give new algorithmic results on the pollution control model called a Pollution Game (PG), introduced in [3], and inspired by [15,6]. We briefly describe applications of PG to air pollution control presented in [3]; for precise definition of PG see Section 2. In the first application, the graph's vertices represent pollution sources (agents) and edges are routes of pollution transition from one source to another. The government as the regulator can decide to either shut down or keep open a pollution source (by selling licences to agents) taking into account the diffusion nature of pollution (emission at one source affects the neighbors at diminishing level). It sets bounds on global and local levels of pollution (called global and local constraint(s), resp.), aiming to optimize the social welfare. The emissions exceeding licences, if any, must be cleaned-up (hence, agent's clean-up cost). In the second application [3], vertices represent mayors of cities and edges the roads between cities. The percentage of cars moving from one city to another is represented by the weight of the corresponding edge. The model allows the regulator to auction pollution licences for cars to mayors. The pollution level of an agent (mayor), i.e., the number of allocated licences and their prices, is set by the regulator.

Here we also consider an application of PG to water pollution in rivers, modelled by tree networks. In water pollution the government decides which pollution sources should be shut down so that the effluent level in water is as low as possible. Water pollution cost sharing was introduced in [17] and the network is a path (single river). This model was extended to tree networks (a system of rivers) in [10]. We also model a system of rivers as a tree, but study a different pollution control model, i.e., [3].

Our Results. We present best possible algorithmic results for trees and planar graphs when we allow a small violation of the constraints on local pollution of every agent (called a local constraint). Suppose first that the objective function is linear. Then, for PG on trees we obtain an FPTAS and this is the best we can achieve as PG is weakly NP-hard [3] on stars. For planar graphs the best known result was a big constant approximation algorithm [3]. We design a PTAS with $(1+\delta)$-violation of the local pollution constraints for any $\delta > 0$, and this is tight as we prove that the problem is strongly NP-hard on planar graphs even with $(1+\delta)$-violations. By using a Lavi-Swamy technique [16] we prove that our FPTAS for trees leads to a randomized truthful in expectation mechanism. In addition, we also design a deterministic truthful mechanism on trees with an approximation ratio $3+\epsilon$. Suppose now that the objective function is 2-piecewise linear or general and monotone. Then for graphs with degree at most Δ we obtain $O(\Delta)$-approximation algorithms and a Unique Games-hardness within $\Delta/\log^2 \Delta$.

Technical Contributions/Approaches. Suppose that the objective functions are linear. When the network is a directed tree, a somehow non-standard two level dynamic programming approach is designed to obtain an FPTAS for PG with binary variables. This approach is crucial to deal with the global constraint. For that we design an FPTAS for a special multiple choice, multi-dimensional knapsack problem where coefficients of all

	General objective function	Linear objective function		
	Bounded Degree Δ	Trees		Planar
Lower bound	$\Omega(\frac{\Delta}{\log \Delta^2})$	NP-hard		strongly NP-hard (δ violation)
PG(poly)	$O(\Delta)^a$	FPTAS TiE	$O(1)$ DT	PTAS (δ violation)
PG(general)	$O(\Delta)$ TiE [b]	FPTAS TiE [c]		$O(1)$ TiE [3]

[a] Monotone increasing obj. function. [b] Piece-wise linear obj. function with one shift and an additional mild assumption. [c] Running time is polynomial in q.

Table 1: Our results. TiE/DT: truthful in expectation/deterministic truthful mechanism. PG(poly) is PG with poly-size integer variables, PG(general) without this assumption.

constraints except one are bounded by a polynomial of the input size; this generalizes the results in [7]. A similar idea is applied to design deterministic truthful mechanisms on trees and a PTAS for PG on planar graphs with $(1 + \delta)$-violations.

To obtain our PTAS for planar PG with $(1+\delta)$-violations, we first use known rounding techniques (e.g., [14,8]) to make all the coefficients polynomially bounded. Then, we design a dynamic programming approach to solve PG on graphs with bounded treewidth tree decomposition. Finally, we combine a special (called nice) tree decomposition of k-outerplanar graphs, Baker's shifting technique and our two-level dynamic programming approach for dealing with the global constraint, obtaining our PTAS.

Even when polluters' cost functions are linear with a single parameter, simple monotonicity is not sufficient to turn our algorithms into truthful mechanisms. This is because polluters' utility functions have externalities – they are affected by their neighbours. Thus, we need to use general techniques to obtain truthful mechanisms: maximal in range mechanisms (for deterministic truthfulness) and maximal in distributional range mechanisms (for truthfulness in expectation). The deterministic truthful mechanism for trees uses a maximum in range technique (Chapter 11 and 12 in [18]).

For piece-wise linear objective functions on bounded degree graphs we prove that PG is Δ column sparse so a randomized algorithm of [5] is applicable. For general monotone objective functions on bounded degree graphs we prove that the objective function is submodular and use randomized rounding with alterations.

Organization. Section 2 contains definitions and preliminaries, and our results on trees are in Section 3. Section 4 presents our results on planar graphs, and, finally, Section 5 discusses general objective functions. All missing details and proofs will appear in the full version.

2 Preliminaries

Model and Applications. We describe the model and mention two applications following [3] to gain an intuition. Consider an area of pollution sources (e.g. factories) each owned by an agent. The government's goal as a regulator is to optimize the social welfare, restricting levels of emitted pollution. Thus, given a weighted digraph $G = (V, E)$, where V is the set of n pollution sources (players, agents) and edge $(u, v) \in E$ means u and v are geographic neighbours, i.e., $(u, v) \in E$ if the pollution emitted by u affects v. For each $(u, v) \in E$ weight $w_{(u,v)} = w_{uv}$ is a discount factor of the pollution discharged by player u affecting its neighbour v. W.l.o.g., $w_{uv} \in (0, 1]$, $\forall (u, v) \in E$.

The government sets the total pollution quota discharged to the environment (by the number of pollution sources that remain open) to be $p \geq \sum_{v \in V} x_v$, where $x_v \in \{0, 1\}$ denotes if pollution source $v \in V$ will be shut down or not. Each agent v has a non-decreasing benefit function $b_v : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}_{\geq 0}$, where $b_v(x_v)$ is a concave increasing function with $b_v(0) = 0$, representing v's benefit. Each v has a non-decreasing damage function $d_v : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}_{\geq 0}$, and b_v is concave increasing, $b_v(0) = 0$ and d_v is convex increasing [5]. Player v's total welfare r_v is v's benefit minus damage cost: $b_v(x_v) - d_v\left(x_v + \sum_{u \in \delta^-_G(v)} w_{uv} x_u\right)$, where, $\delta^-_G(v) = \{u \in V : (u, v) \in E\}$, $\delta^+_G(v) = \{u \in V : (v, u) \in E\}$. Thus, v is affected via the damage function by his own pollution if $x_v \neq 0$ and by the total discounted pollution neighbours. This models that pollution spreads along the edges of G. The government decides on the allowable local level of pollution p_v, for every $v \in V$, which imposes the following constraints for every $v \in V$: $x_v \leq q_v, \;\; x_v + \sum_{u \in \delta^-_G(v)} w_{uv} x_u \leq p_v$. The first application assumes $x_v \in \{0, 1\}$ and $q_v = 1, \forall v \in V$ and the second $x_v \in \{0, 1, \ldots, q_v\}$ and $q_v \in \mathbb{N}$.

The problem of social welfare maximization is the following convex integer program (1)-(4), called a pollution game (PG) on G, where (2) is called global constraint, (3) are local constraints, and $x_v + \sum_{u \in \delta^-_G(v)} w_{uv} x_u$ is the local level of pollution of v. Value q_v is decided by the government and for this application $q_v = 1$. We call (1)-(4), PG with integer variables (if $x_v \in \mathbb{Z}$) or with binary variables (if $x_v \in \{0, 1\}$). For an instance I of PG, $|I|$ is the number of bits to encode I, and if $q \in poly(|I|)$, $q = \max_{v \in V}\{q_v\}$+1, we call (1)-(4), PG with polynomial size integer variables.

$$\max \;\; R(x) = \sum_{v \in V} (b_v(x_v) - d_v(x_v + \sum_{u \in \delta^-_G(v)} w_{uv} x_u)) \quad (1)$$

$$\text{s.t.} \;\; \sum_{v \in V} x_v \leq p \quad (2)$$

$$x_v + \sum_{u \in \delta^-_G(v)} w_{uv} x_u \leq p_v, \;\; \forall v \in V \quad (3)$$

$$x_v \in \{0, 1, \ldots, q_v\}, \;\; \forall v \in V \quad (4)$$

Basic Definitions. Let $I = (G, \mathbf{b}, \mathbf{d}, \mathbf{p}, \mathbf{q})$ be an instance of PG, $\mathbf{b} = (b_v)_{v \in V}$, $\mathbf{d} = (d_v)_{v \in V}$, $\mathbf{p} = (p_v)_{v \in V}$ and $\mathbf{q} = (q_v)_{v \in V}$ (b_v is private information of v and other parameters are public). Let $\mathcal{I}$ be the set of all instances, and $\mathcal{X}$ the set of feasible allocations. Given a digraph $G = (V, E)$, $G^{un} = (V, E^{un})$, where $E^{un} = \{(u, v) : (u, v) \in E$ or $(v, u) \in E\}$. A mechanism $\phi = (X, P)$ consists of an allocation $X : \mathcal{I} \to \mathcal{X}$ and payment function $P : \mathcal{I} \to \mathbb{R}^{|V|}_{\geq 0}$ ($X(I)$ satisfies (2)–(4)). For any vector x, x_{-u} denotes vector x without its u-th component. Note, $r_v(X(I)) = b_v(X_v(I)) - d_v(X_v(I) + \sum_{u \in \delta^-_G(v)} w_{uv} X_u(I))$ is the welfare of player v under $X(I)$. A mechanism $\phi = (X, P)$ is truthful, if for any b_{-v}, b_v and b'_v, $r_v(X(b_v, b_{-v})) - P_v(b_v, b_{-v}) \geq r_v(X(b'_v, b_{-v})) - P_v(b'_v, b_{-v})$. A randomized mechanism is truthful in expectation if for any b_{-v}, b_v and b'_v, $\mathbb{E}(r_v(X(b_v, b_{-v})) - P_v(b_v, b_{-v})) \geq \mathbb{E}(r_v(X(b'_v, b_{-v})) - P_v(b'_v, b_{-v}))$, where $\mathbb{E}(\cdot)$ is over the algorithm's random bits. $OPT^{fr}_G(PG)$ ($OPT^{in}_G(PG)$, resp.) denotes the value of the optimal fractional (integral, resp.) solution of PG on G. A mechanism is in-

[5] [15] uses cost function rather than benefit function, viewed as $M_v - b_v(x_v)$, with M_v a large constant for any $v \in V$. The cost function is convex decreasing and it is equivalent to $b_v(x_v)$ being a concave increasing function. We use benefit function rather than cost function.

dividually rational if each agent v has non-negative utility when he declares b_v, regardless of the other agents' declarations. The approximation ratio of an algorithm $\mathcal{A}$ w.r.t. $OPT_G^{in}(PG)$ (resp. $OPT_G^{fr}(PG)$) is $\eta^{in}(\mathcal{A}) = \frac{OPT_G^{in}(PG)}{R(\mathcal{A})}$ ($\eta^{fr}(\mathcal{A}) = \frac{OPT_G^{fr}(PG)}{R(\mathcal{A})}$), where $R(\mathcal{A})$ is the objective value of the $\mathcal{A}$'s solution. If unspecified, the approximation ratio refers to η^{in}. An FPTAS (PTAS, resp.) for a problem $\mathcal{P}$ is an algorithm $\mathcal{A}$ that for any $\epsilon > 0$ and any instance I of $\mathcal{P}$, outputs a solution with the objective value at least $(1-\epsilon)OPT_I^{in}(\mathcal{P})$ and terminates in time $poly(\frac{1}{\epsilon}, |I|)$ ($(\frac{1}{\epsilon}|I|)^{g(\frac{1}{\epsilon})}$, resp.), where g is a function independent from I. Let $\gamma_k = \min\{2k^2+2, 8k, \frac{k}{(1-\frac{1}{k}(1+(\frac{2}{k})^{\frac{1}{3}}))^k}\} = (e+o(1))k = O(k)$, and $[n] = \{1,\ldots,n\}$. We use 'vertex' to denote the vertex in a graph and 'node' to denote a vertex of the tree obtained from a tree decomposition of a graph. An undirected graph is an outerplanar if it can be drawn in the plane without crossings in such a way that all of the vertices belong to the unbounded face of the drawing. An undirected graph G is k-outerplanar if for $k = 1$, G is outerplanar and for $k > 1$, G has a planar embedding such that if all vertices on the exterior face are deleted, the connected components of the remaining graph are all $(k-1)$-outerplanar. An planar graph is k outerplanar where k can be equal to $+\infty$. A digraph is called a planar graph if its undirected version is planar. We consider some standard embedding of a planar graph and define level k vertices in a planar embedding E of a planar graph G. A vertex is at level 1 if it is on the exterior face. Call a cycle of level i vertices a level i face if it is an interior face in the subgraph induced by the level i vertices. For each level i face f, let G_f be the subgraph induced by all vertices placed inside f in this embedding. Then the vertices on the exterior face of G_t are at level $i+1$.

In Sections 3 and 4 we assume that b_v and d_v are both linear with slopes s_v^0 and s_v^1 respectively, i.e., $b_v(x) = s_v^0 x$ and $d_v(y) = s_v^1 y$, for any $v \in V$. The social welfare function is $R(x) = \sum_{v\in V}\omega_v x_v$, where $\omega_v = s_v^0 - s_v^1 - \sum_{u\in\delta_G^+(v)} s_u^1 w_{vu}$ ($R(x) = \sum_{v\in V} b_v(x_v) - d_v(x_v + \sum_{u\in\delta_G^-(v)} w_{uv}x_u) = \sum_{v\in V} s_v^0 x_v - s_v^1(x_v + \sum_{u\in\delta_G^-(v)} w_{uv}x_u) = \sum_{v\in V}\omega_v x_v$).

3 Directed Trees

Truthful in Expectation Mechanisms. A digraph G is called a *directed tree* if the undirected graph G^{un} is a tree. We consider trees where arcs are directed towards the leaves. We obtain our truthful in expectation FPTAS for PG with binary variables on any directed trees by a two-level dynamic programming (DP) approach (used also in Section 4). The first bottom-up level is based on a careful application of the standard single-dimensional knapsack FPTAS. The second level is by an interesting generalization of an FPTAS of [7] for a special multi-dimensional knapsack problem, see (IP_2) below, with a constant number of constraints most of which have $poly(|I|)$ size of coefficients. This FPTAS generalizes the results in [7], where the authors consider the one dimensional knapsack problem with cardinality constraint; it will appear in our paper's full version.

We will also need the following tool from mechanism design for packing problems. An integer linear packing problem with binary variables is a problem of maximising a linear objective function over a set of linear packing constraints, i.e., constraints of form $a \cdot x \le b$ where $x \in \{0,1\}^n$ is a vector of binary variables, and $a, b \in \mathbb{R}^n_{\ge 0}$.

Proposition 1 ([11]). *Given an FPTAS for an integer linear packing problem with binary variables, there is a truthful in expectation mechanism that is an FPTAS.*

We first present an FPTAS without constraint (2) which captures our main technique.

Warmup (without global constraint). The algorithm uses a DP and FPTAS for knapsack as a subroutine. Note, on a star, any instance of knapsack can be reduced to PG without global constraint. Thus FPTAS is the best possible for such PG unless $P = NP$.

$$\begin{aligned} \max\ & \textstyle\sum_{i\in[n_v]}(M^{v\text{in}}_{u_i\text{in}}x_{u_i} + M^{v\text{in}}_{u_i\text{out}}(1-x_{u_i})) + \omega_v \\ \text{s.t.}\ & 1 + w_{v'v} + \textstyle\sum_{i\in[n_v]} w_{u_i v}x_{u_i} \le p_v, \qquad (IP_1) \\ & x_{u_i} \in \{0,1\},\ \ \forall i \in [n_v] \end{aligned}$$

We keep four values for each $v \in V$. Suppose v's father is v', let $M^{v'\text{in}}_{v\text{in}}$ denote the optimal value of PG on subtree rooted at v when both v' and v are selected in the solution. Similarly, we have $M^{v'\text{in}}_{v\text{out}}$, $M^{v'\text{out}}_{v\text{in}}$ and $M^{v'\text{out}}_{v\text{out}}$. Let u_i, $i = 1, 2, ..., n_v$ denote children of v. Suppose $M^{v\text{in}}_{u_i\text{in}}$, $M^{v\text{in}}_{u_i\text{out}}$, $M^{v\text{out}}_{u_i\text{in}}$ and $M^{v\text{out}}_{u_i\text{out}}$ have been calculated, for any $i = 1, ..., n_v$. Some of them are undefined due to infeasibility. Now, calculate $M^{v'\text{in}}_{v\text{in}}$. Observe, $M^{v'\text{in}}_{v\text{in}}$ is equal to the optimal value of the knapsack (IP_1), where $M^{v\text{in}}_{u_i\text{in}}$ and $M^{v\text{in}}_{u_i\text{out}}$ have finite values (otherwise remove them). If this knapsack problem has a feasible solution, we get value $M^{v'\text{in}}_{v\text{in}}$, otherwise set $M^{v'\text{in}}_{v\text{in}}$ undefined. Similarly, calculate $M^{v'\text{in}}_{v\text{out}}$, $M^{v'\text{out}}_{v\text{in}}$ and $M^{v'\text{out}}_{v\text{out}}$. Thus, at each step if we calculate an optimal solution, it will be obtained by above DP approach. For knapsack with n_v variables, there is an FPTAS. Hence, at each step we get approximate value $\bar{M}^{v'\text{in}}_{v\text{in}} \ge (1-\epsilon)M^{v'\text{in}}_{v\text{in}}$ in poly-time in n_v and $\frac{1}{\epsilon}$ by knapsack's FPTAS; similarly for other three values. Thus, in the final solution, $\bar{M}_{root} \ge (1-\epsilon)^k M_{root}$, where k is the number of levels of the tree and M_{root} is PG's optimal value without global constraint, terminating in $poly(|I|, \frac{1}{\epsilon})$ time; $|I|$ is the input size. Set $1-\epsilon' = (1-\epsilon)^k$, then $\epsilon = \Theta(\frac{\epsilon'}{k})$. The run time is $poly(|I|, \frac{k}{\epsilon'}) = poly(|I|, \frac{1}{\epsilon'})$ due to $k \le |I|$, giving FPTAS for PG without global constraint.

W.l.o.g., suppose $p \le n$, otherwise let $p = n$. For each vertex v, we keep $4p$ values. Let v's father be v', and let $M^{v'\text{in}}_{v\text{in}}(s)$ be the optimal value of PG on the subtree rooted at v when both v' and v are selected in the solution, and the total pollution level allocated to the subtree rooted at v is $\le s$, $s = 0, 1, ..., p$. Similarly, we have $M^{v'\text{in}}_{v\text{out}}(s)$, $M^{v'\text{out}}_{v\text{in}}(s)$ and $M^{v'\text{out}}_{v\text{out}}(s)$. Let u_i, $i \in [n_v]$ denote the children of v. Suppose $M^{v\text{in}}_{u_i\text{in}}(s)$, $M^{v\text{in}}_{u_i\text{out}}(s)$, $M^{v\text{out}}_{u_i\text{in}}(s)$ and $M^{v\text{out}}_{u_i\text{out}}(s)$ have been calculated, for any $i \in [n_v]$ and $s = 0, 1, ..., p$. Some of them are undefined due to infeasibility. Note, $M^{v\text{in}}_{u_i\text{in}}(0)$, $M^{v\text{out}}_{u_i\text{in}}(0)$ are undefined and $M^{v\text{out}}_{u_i\text{out}}(0) = M^{v\text{in}}_{u_i\text{out}}(0) = 0$. Now, calculate $M^{v'\text{in}}_{v\text{in}}(\ell)$. Observe, $M^{v'\text{in}}_{v\text{in}}(\ell)$ is equal to the optimal value of the knapsack problem (IP_2) (called $\text{KNAPSACK}_v(\ell)$) plus ω_v. If $M^{v\text{out}}_{u_i\text{in}}(s)$ and $M^{v\text{out}}_{u_i\text{out}}(s)$ are undefined, they are removed from $\text{KNAPSACK}_v(\ell)$. Note, $x_{i0} \equiv 0$, for any $i \in [d]$. If $\text{KNAPSACK}_v(\ell)$ has a feasible solution, we get the value $M^{v'\text{in}}_{v\text{in}}(\ell)$, otherwise set $M^{v'\text{in}}_{v\text{in}}(\ell)$ undefined. Similarly, calculate $M^{v'\text{in}}_{v\text{out}}(\ell)$, $M^{v'\text{out}}_{v\text{in}}(\ell)$, $M^{v'\text{out}}_{v\text{out}}(\ell)$, $\ell = 0, 1, ..., p$. From the analysis of DP without global constraint, if there is an FPTAS for $\text{KNAPSACK}_v(\ell)$, then there is one for $\text{KNAPSACK}_{root}(p)$, and so an FPTAS for PG with binary variables on directed trees. Note, the second constraint in (IP_2) can be replaced by $\sum_{s=1}^{p}(x_{is} + y_{is}) \le 1, \forall i \in [n_v]$. Then, by Proposition 1:

$$\begin{aligned} \max\ & \textstyle\sum_{i\in[n_v]}\sum_{s\in[p]}(M^{v\text{in}}_{u_i\text{in}}(s)x_{is} + M^{v\text{in}}_{u_i\text{out}}(s)y_{is}) \\ \text{s.t.}\ & \textstyle\sum_{i\in[n_v]}\sum_{s\in[p]} s(x_{is} + y_{is}) \le \ell - 1, \\ & \textstyle\sum_{s=0}^{p}(x_{is} + y_{is}) = 1,\ \forall i \in [n_v] \qquad (IP_2) \\ & 1 + w_{v'v} + \textstyle\sum_{i\in[n_v]}[w_{u_i v}(\sum_{s=0}^{p} x_{is})] \le p_v, \\ & x_{is}, y_{is} \in \{0,1\},\ \forall i \in [n_v], s \in [p] \end{aligned}$$

Theorem 1. *There is a truthful in expectation mechanism for PG with binary variables on directed trees, which is an FPTAS.*

For $x_v \in \mathbb{Z}$, we can replace each x_v by q_v duplicated variables $x_{vj}, j = 1, \cdots, q_v$, i.e., $\{x_v \in \{0, 1, \ldots, q_v\}\} = \{\sum_{j \in [q_v]} j x_{vj} \mid \sum_{j \in [q_v]} x_{vj} \leq 1, x_{vj} \in \{0, 1\}\}$. This transforms a poly-size integer constraint into a multiple choice, one dimensional knapsack constraint. Hence, for directed trees, by a DP, we can construct a pseudo poly-time algorithm to compute the exact optimal value of PG with integer variables, in time $poly(|V|, q, OPT^{in}(PG))$. And, we can remove $OPT^{in}(PG)$ from the running time losing an ϵ by scaling techniques, implying a $(1-\epsilon)$-approximation algorithm for PG with integer variables with time $poly(|V|, q, 1/\epsilon)$. By Proposition 1:

Theorem 2. *There is a truthful in expectation mechanism for PG with polynomial size integer variables on directed trees, which is an FPTAS.*

Deterministic Truthful Mechanisms. We use a maximal in range (MIR) mechanism for PG with polynomial size integer variables on directed trees. By transformation from integer constraint into multiple choice and one dimensional knapsack constraint, we know we only need to show such approximation algorithm for binary variables. Based on recent deterministic truthful PTAS for 2 dimensional knapsack[6][9,14,8] we obtain:

Theorem 3. *There is a deterministic ($\eta^{in} = 3 + \epsilon$)- approximation truthful mechanism for PG with polynomial size integer variables on directed trees, which for binary variables terminates in $O(|V|^2 \Delta^{6+\frac{1}{\epsilon}})$ time.*

4 Planar Graphs

A PTAS with δ-violation: Our approach to obtain a PTAS has three main steps:

1. Round PG to an equivalent problem $\bar{PG}_2$ with polynomial size integer variables.
2. Using the nice tree decomposition, we present a dynamic programming approach to solve $\bar{PG}_2$ optimally on an k-outerplanar graph.
3. By a shifting technique similar to [4], we obtain a PTAS with $1 + \delta$ violation.

Step 1: Rounding Procedure. Recall that PG is equivalent to maximizing $\sum_{v \in V} \omega_v x_v$ subject to constraints (1) - (3) where $\omega_v = \max\{0, s_v^0 - s_v^1 - \sum_{u \in \delta_G^+(v)} s_u^1 w_{vu}\}$ and $w_{v,v} = 1 \; \forall v \in V$, and b_v and d_v are both linear with slopes s_v^0 and s_v^1. For each $v \in V$, suppose $q_v \in [2^{o_v - 1} - 1, 2^{o_v} - 1)$. Let $o_v = \lfloor \log_2(q_v) \rfloor + 1$ if $q_v \neq 2^{o_v - 1} - 1$ and $o_v = \lfloor \log_2(q_v) \rfloor + 2$ otherwise; $c_v^i = 2^{i-1}$, $i \in [o_v - 1]$ and $c_v^{o_v} = q_v - 2^{o_v - 1} + 1$. Notice, $\{x_v \mid x_v \in \mathbb{Z}, 0 \leq x_v \leq q_v\} = \{\sum_{i=1}^{o_v} c_v^i y_v^i \mid y_v^i \in \{0, 1\}, i \in [o_v]\}$, for any $v \in V$. Thus, PG is equivalent to the following integer program (denoted as PG'):

$$\begin{array}{lll|lll}
\max & \sum_{v \in V} \sum_{i=1}^{o_v} \omega_v c_v^i y_v^i & (PG') & \max & \sum_{v \in V} \sum_{i=1}^{o_v} \omega_v b_v^i y_v^i & (\bar{PG}_1) \\
\text{s.t.} & \sum_{v \in V} \sum_{i=1}^{o_v} c_v^i y_v^i \leq p, & & \text{s.t.} & \sum_{v \in V} \sum_{i=1}^{o_v} c_v^i y_v^i \leq p & \\
& \forall v \in V: \; \sum_{i=1}^{o_v} w_{vv} c_v^i y_v^i + & & & \forall v \in V: \; \sum_{i=1}^{o_v} \bar{w}_{vv}^i y_v^i + & \\
& \quad + \sum_{u \in \delta_G^-(v)} \sum_{i=1}^{s_u} w_{uv} c_u^i y_u^i \leq p_v, & & & \quad + \sum_{u \in \delta_G^-(v)} \sum_{i=1}^{s_u} \bar{w}_{uv}^i y_v^i \leq \bar{p}_v & \\
& \forall v \in V, i \in [o_v]: \; y_v^i \in \{0, 1\}, & & & \forall v \in V, i \in [o_v]: \; y_v^i \in \{0, 1\} &
\end{array}$$

[6] This PTAS also works for multiple choice and constant dimensional knapsack problem, which will be used for PG with polynomial size integer variables.

Let $o^* = \max_{v\in V} o_v$ and $\rho = o^*|V|$. Recall that $q = \max_{v\in V}\{q_v\} + 1$. For any $\delta > 0$, let $\bar{w}^i_{uv} = \lfloor \frac{2w_{uv}c^i_v\rho}{p_v\delta} \rfloor$ and $\bar{p}_v = \lceil \frac{2p_v\rho}{p_v\delta} \rceil = \lceil \frac{2\rho}{\delta} \rceil$, for any $u, v \in V$. Then we have the following modified PG' (denoted as $\bar{P}G_1$ – see above.)

Lemma 1. *Any feasible solution of PG' is feasible in $\bar{P}G_1$, and any feasible solution of $\bar{P}G_1$ is feasible for PG except violating each local constraint by a factor of $1+\delta$.*

Proof. We only prove local constraints for each direction since the proof of the global constraint is similar. Let $\{y^i_v\}_{v\in V,\, i\in[o_v]}$ be a feasible solution of PG'. We know that $\sum_{i=1}^{o_v} w_{vv}c^i_v y^i_v + \sum_{u\in\delta^-_G(v)}\sum_{i=1}^{s_u} w_{uv}c^i_u y^i_v \le p_v,\ \forall v \in V$. Then $\sum_{i=1}^{o_v}\bar{w}^i_{vv}y^i_v + \sum_{u\in\delta^-_G(v)}\sum_{i=1}^{s_u}\bar{w}^i_{uv}y^i_v \le \frac{2\rho}{p_v\delta}(\sum_{i=1}^{o_v} w_{vv}c^i_v y^i_v + \sum_{u\in\delta^-_G(v)}\sum_{i=1}^{s_u} w_{uv}c^i_u y^i_v) \le \frac{2\rho}{p_v\delta}p_v \le \bar{p}_v$ as desired. On the other hand, suppose $\{y^i_v\}_{v\in V,\, i\in[o_v]}$ is a feasible solution of $\bar{P}G_1$. We know $\sum_{i=1}^{o_v}\bar{w}^i_{vv}y^i_v + \sum_{u\in\delta^-_G(v)}\sum_{i=1}^{s_u}\bar{w}^i_{uv}y^i_v \le \bar{p}_v, \forall v \in V$. Then $\sum_{i=1}^{o_v} w_{vv}c^i_v y^i_v + \sum_{u\in\delta^-_G(v)}\sum_{i=1}^{s_u} w_{uv}c^i_u y^i_v \le \frac{p_v\delta}{2\rho}[\sum_{i=1}^{o_v}(\bar{w}^i_{vv}+1)y^i_v + \sum_{u\in\delta^-_G(v)}\sum_{i=1}^{s_u}(\bar{w}^i_{uv}+1)y^i_v] \le \frac{p_v\delta}{2\rho}\bar{w}^i_{vv}y^i_v + \sum_{u\in\delta^-_G(v)}\sum_{i=1}^{s_u}\bar{w}^i_{uv}y^i_v + \frac{p_v\delta\rho}{2\rho} \le \frac{p_v\delta\bar{p}_v}{2\rho} + \frac{p_v\delta}{2} \le \frac{p_v\delta}{2\rho}(\frac{2\rho}{\delta}+1) + \frac{p_v\delta}{2} \le p_v(1+\delta), \forall v \in V$. □

$$\begin{array}{lll} \max & \sum_{v\in V}\omega_v x_v & (\bar{P}G_2) \\ \text{s.t.} & \sum_{v\in V} x_v \le p & \\ \multicolumn{3}{l}{\forall v \in V : \bar{w}_{vv}(x_v) + \sum_{u\in\delta^-_G(v)}\bar{w}_{uv}(x_u) \le \bar{p}_v} \\ \multicolumn{3}{l}{\forall v \in V : x_v \in \Lambda_v} \end{array}$$

Note, for each $\ell \in [q_v]$, there is a solution $\{y^i_v\}_{i\in[o_v]}$ s.t. $\sum_{i=1}^{o_v} c^i_v y^i_v = \ell$. We use the following solution: If $\ell \le 2^{o_v-1}-1$, set $y^{o_v}_v = 0$ and there is a unique solution $\sum_{i=1}^{o_v} c^i_v y^i_v = \ell$; If $2^{o_v-1}-1 < \ell \le q_v$, set $y^{o_v}_v = q_v - 2^{o_v-1}+1$ and there is also a unique solution s.t. $\sum_{i=1}^{o_v} c^i_v y^i_v = \ell$. Hence, there is one-to-one correspondence from x_v to $\{y^i_v\}_{i\in[o_v]}$. Notice that for a given x_v, the above defined solution $\{y^i_v\}_{i\in[o_v]}$ is the one such that $\sum_{i=1}^{o_v}\bar{w}^i_{vv}y^i_v + \sum_{u\in\delta^-_G(v)}\sum_{i=1}^{s_u}\bar{w}^i_{uv}y^i_v$ is minimized. Now let $\bar{w}_{vu}(x_v) = \sum_{i=1}^{o_v}\bar{w}^i_{vu}y^i_v$, for any $v, u \in V$, where $\{y^i_v\}_{i\in[o_v]}$ is according to the above solution corresponding to x_v. Let $\Lambda_v = [q_v] \cup \{0\}$. Thus, $\bar{P}G_1$ (also PG) is equivalent to the integer program (denoted as $\bar{P}G_2$, see above).

Step 2: Preliminaries of Tree Decompositions on k-outerplanar Graphs

Definition 1. *A tree decomposition of an undirected graph $G = (V, E)$ is a pair $(\{X_i|i \in I\}, T = (I, F))$, with $\{X_i|i \in I\}$ a family of subsets of V, one for each node of T, and T a tree such that: 1) $\bigcup_{i\in I} X_i = V$, 2) for all edges $(v, w) \in E$, there exists an $i \in I$ with $v \in X_i$ and $w \in X_i$, 3) for all i, j, k $\in I$: if j is on the path from i to k in T, then $X_i \cap X_k \subseteq X_j$. The width of a tree decomposition $(\{X_i|i \in I\}, T = (I, F))$ is $\max_{i\in I}|X_i|-1$. The minimum width of all tree decompositions of G is called treewidth.*

Definition 2. *A tree decomposition $(\{X_i|i \in I\}, T = (I, F))$ of $G = (V, E)$ is called a nice tree decomposition if T is a rooted binary tree and 1) if a node $i \in I$ has two children j and k, then $X_i = X_j = X_k$ (joint node), 2) if a node $i \in I$ has one child j, then either $X_i \subset X_j$, and $|X_i| = |X_j| - 1$ (forget node), or $X_j \subset X_i$ and $|X_j| = |X_i| - 1$ (introduce node), 3) if node $i \in I$ is a leaf of T, then $|X_i| = 1$ (leaf node).*

Lemma 2 ([12]). *For any k-outerplanar graph $G = (V, E)$, there is an algorithm to compute a tree decomposition $(\{X_i | i \in I\}, T = (I, F))$ of G with treewidth at most $3k - 1 = O(k)$, and $I = O(|V|)$ in $O(k|V|)$ time.*

Given a tree decomposition $(\{X_i | i \in I\}, T = (I, F))$ for $G = (V, E)$ with treewidth k and $I = O(|V|)$, we can obtain a nice tree decomposition with the same treewidth k and the number of nodes $O(k|V|)$ in $O(k^2|V|)$ time [13]. Thus, for any k-outerplanar graph $G = (V, E)$, we can compute a nice tree decomposition $(\{X_i | i \in I\}, T = (I, F))$ of G with treewidth at most $3k - 1 = O(k)$, and $I = O(k|V|)$ in $O(k^2|V|)$ time. In the following, we suppose there is a nice tree decomposition for any k-outerplanar graph.

Dynamic Programming (DP). A DP to solve $\bar{PG}_2$ on a k-outerplanar digraph is presented by using a nice tree decomposition of its undirected version. Note, a nice tree decomposition of an undirected version of digraph is also a nice tree decomposition of itself. Given nice tree decomposition $(\{X_i | i \in I\}, T = (I, F))$ of a k-outerplanar digraph $G = (V, E)$, using a bottom-up approach, DP for $\bar{PG}_2$ works as follows.

For any node $i \in I$, suppose $X_i = \{v_1^i, v_2^i, \cdots, v_t^i\}$, where $t \leq 3k$. We also say vertex v_1^i belongs to node X_i, similarly we can say a vertex belongs to a subtree of T, meaning this vertex belongs to some node of this subtree. Given any emission amount $\{x_v\}_{v \in V}$, recall $\bar{w}_{vv}(x_v) + \sum_{u \in \delta_G^-(v)} \bar{w}_{uv}(x_u)$ is the local level of pollution of vertex v. We use $\boldsymbol{a^i} = (a_1^i, a_2^i, \cdots, a_t^i)$ to denote the emission amount allocated to vertices in X_i, i.e., a_s^i denotes the emission amount allocated to the vertex v_s^i, $s \in [t]$. Similarly $\boldsymbol{\ell^i}$ denotes the local levels of pollution of vertices in X_i. Let G_i denote the subgraph generated by all the vertices belonging to the subtree (node X_i) rooted at X_i. We use Q^i to denote the total emission quota allocated to G_i. Let $\Omega_i(\boldsymbol{a^i}, \boldsymbol{\ell^i}, Q^i)$ denote the optimal objective value of $\bar{PG}_2$ restricted on the subgraph G_i, when the emission amount and local level of pollution of v_s^i are exactly a_s^i and ℓ_s^i, $s \in [t]$, and the total emission amount allocated to G_i is exactly Q^i. If there is no feasible solution for $\Omega_i(\boldsymbol{a^i}, \boldsymbol{\ell^i}, Q^i)$, we will see that our DP approach will automatically set $\Omega_i(\boldsymbol{a^i}, \boldsymbol{\ell^i}, Q^i)$ to be $-\infty$. Let $\bar{w}_{uv}(x_v) \equiv 0$ if (u, v) is not an edge in G. Note that the range of a_s^i we need to compute is in Λ_v, and ℓ_s^i is from 0 to $\bar{p}_{v_s^i}$, $s \in [t]$, Q^i is from 0 to p. We present the DP approach

- X_i is a leaf node or a start node, where $t = 1$. $\Omega_i(a_1^i, \ell_1^i, Q^i) = \omega_{v_1^i} a_1^i$ if the triple (a^i, ℓ^i, Q^i) is feasible, which can be verified easily e.g. $Q^i = a_1^i$ and $\ell_1^i = \bar{w}_{v_1^i v_1^i}(a_1^i)$. Let $\Omega_i(a_1^i, \ell_1^i, Q^i) = -\infty$ if the triple (a^i, ℓ^i, Q^i) is not feasible.
- X_i is a forget node, and suppose its child is $X_j = X_i \cup \{v_{t+1}^j\}$.
  $\Omega_i(\boldsymbol{a^i}, \boldsymbol{\ell^i}, Q^i) = \max_{a_{t+1}^j, \ell_{t+1}^j} \Omega_j(\boldsymbol{a^i}, a_{t+1}^j, \boldsymbol{\ell^i}, \ell_{t+1}^j, Q^i)$
- X_i is an introduce node, and suppose its child is $X_j = X_i \backslash \{v_t^i\}$. Let $a_s^j = a_s^i$ and $\ell_s^j = \ell_s^i - \bar{w}_{v_t^i v_s^i}(a_t^i), \forall s \in [t-1]$. $\Omega_i(\boldsymbol{a^i}, \boldsymbol{\ell^i}, Q^i) = \Omega_j(\boldsymbol{a^j}, \boldsymbol{\ell^j}, Q^i - a_t^i) + \omega_{v_t^i} a_t^i$ if $\sum_{s \in [t]} \bar{w}_{v_s^i v_t^i}(a_s^i) = \ell_t^i$, and $\Omega_i(\boldsymbol{a^i}, \boldsymbol{\ell^i}, Q^i) = -\infty$ otherwise.
- X_i is a joint node, and suppose its two children are $X_j = X_k = X_i$. $\Omega_i(\boldsymbol{a^i}, \boldsymbol{\ell^i}, Q^i) = \max_A \{\Omega_j(\boldsymbol{a^j}, \boldsymbol{\ell^j}, Q^j) + \Omega_k(\boldsymbol{a^k}, \boldsymbol{\ell^k}, Q^k)\}$, where the condition
  $A = \{(\boldsymbol{a^j}, \boldsymbol{\ell^j}, Q^j), (\boldsymbol{a^k}, \boldsymbol{\ell^k}, Q^k) \mid \boldsymbol{a^j} + \boldsymbol{a^k} = \boldsymbol{a^i}, \boldsymbol{\ell^j} + \boldsymbol{\ell^k} = \boldsymbol{\ell^i}, Q^j + Q^k = Q^i\}$.
- X_i is the root of T, $OPT(Q^i) = \max_{\boldsymbol{a^i}, \boldsymbol{\ell^i}} \{\Omega_i(\boldsymbol{a^i}, \boldsymbol{\ell^i}, Q^i)\}$ is the optimal value (social welfare) of $\bar{PG}_2$ when total scaled emission amount is exactly Q^i, i.e., the global constraint satisfies $\sum_{v \in V} b_v x_v = Q^i$.

Analysis of Running Time of DP. It is not difficult to see that the above DP approach gives the correct solution of $\bar{PG}_2$ on k-outerplanar graphs. For each node X_i, we need to keep $O(pq^{3k}\lceil\frac{2\rho}{\delta}\rceil^{3k}) = O(|V|q^{3k+1}\lceil\frac{2\rho}{\delta}\rceil^{3k})$ number of Ω_i values. Each Ω_i can be computed in $O(|V|q^{3k+1}\lceil\frac{2\rho}{\delta}\rceil^{3k})$ time (this is the worst case running time when X_i is a joint node). There are $O(k|V|)$ nodes in T. Therefore, the total running time of the DP approach (by multiplying above three numbers) is $O(k|V|^3q^{6k+2}\lceil\frac{2\rho}{\delta}\rceil^{6k})$.

Based on the above DP approach, we can solve $\bar{PG}_2$ on any k-outerplanar graph optimally for any fixed k (which includes any directed tree whose treewidth is 2). Therefore, for any $\delta > 0$ and fixed k, we can use VCG (see, e.g., Chapter 9 in [18]) to get an optimal deterministic truthful mechanism for PG on any directed k-outerplanar graph that violates each local constraint by a factor of δ and runs in $O(k|V|^3q^{6k+2}\lceil\frac{2\rho}{\delta}\rceil^{6k})$ time (note that Theorem 4 also works for bounded treewidth graphs).

Theorem 4. *For any $\delta > 0$ and fixed k, there is an optimal deterministic truthful mechanism for PG on any directed k-outerplanar graph $G = (V, E)$ that violates each local constraint by a factor of $1 + \delta$ and runs in $O(k|V|^3q^{6k+2}\lceil\frac{2\rho}{\delta}\rceil^{6k})$ time, where $\rho = |V|(\lfloor\log_2(q)\rfloor + 2)$.*

Step 3: PTAS for Planar Graphs Observe that when there are some boundary conditions on k-outerplanar, the above DP approach still works. For example, if the emission amount of any vertex in any first and last face (level 1 and level k face) of the k-outerplanar graph is zero, we just modify the dynamic programming approach in a bottom-up manner to set $\Omega_i = -\infty$ if any vertex v in any first and last face is a parameter of Ω_i and its emission amount $a_v^i > 0$. Then the modified DP approach is the desired algorithm for $\bar{PG}_2$ on the k-outerplanar graph under this boundary condition.

Proposition 2. *PG is strongly NP-hard on planar graphs with degree at most 3 when we allow a $(1 + \delta)$-violation of local constraints.*

Theorem 5. *For any fixed k and $\delta > 0$, there is an $O(k^2|V|^3q^{6k+2}\lceil\frac{2\rho}{\delta}\rceil^{6k})$ algorithm for PG with integer variables on directed planar graph $G = (V, E)$ that achieves $(\eta^{in} = \frac{k}{k-2})$-approximation and violates each local constraint by a factor of $1 + \delta$, where $\rho = |V|(\lfloor\log_2(q)\rfloor + 2)$.*

Proof. We use $OPT(\bar{PG}_2)$ to denote $OPT_G^{in}(\bar{PG}_2)$ and omit the superscript and subscript. By Lemma 1, we know $OPT = OPT(PG) \le OPT(\bar{PG}_2)$. Let $\bar{PG}_2(i)$ denote the $\bar{PG}_2$ restricted on G by setting $x_v = 0$ for each v who belongs to any face $f \equiv i$ or $i + 1$ $(mod\, k)$. Let $\{x_v^*\}_{v\in V}$ be an optimal solution for $\bar{PG}_2$. Then we know $\sum_{i\in[k]}\sum_{v\in f: f\equiv i \text{ or } i+1 (mod\, k)} x_v^* = 2OPT(\bar{PG}_2)$. As a consequence, there exists $i \in [k]$ such that $\sum_{v\in f: f\equiv i \text{ or } i+1(mod\, k)} x_v^* \le \frac{2OPT(\bar{PG}_2)}{k}$. Observe that $\{x_v\}_{v\in V}$ is a feasible solution for $\bar{PG}_2(i)$, where $x_v = 0$ if v belongs to any face $f \equiv i$ or $i+1$ $(mod\, k)$ and $x_v = x_v^*$ otherwise. Thus, $OPT(\bar{PG}_2(i)) \ge (1 - \frac{2}{k})OPT(\bar{PG}_2) \ge (1 - \frac{2}{k})OPT$. Solving each $\bar{PG}_2(i)$, $i \in [k]$, then choosing $\max_{i\in[k]}\{OPT(\bar{PG}_2(i))\}$ (which is at least $(1 - \frac{2}{k})OPT$) gives the desired result. Now let us see how to solve $\bar{PG}_2(i)$. Note that for $\bar{PG}_2(i)$, $x_v = 0$ for any v who belongs to any face $f \equiv i$ or $i + 1$ $(mod\, k)$. $\bar{PG}_2(i)$ consists of independent $k'-$outerplanar graphs, each of which has some boundary condition i.e. the emission amount of any vertex in any first and last face is zero and $k' \leq k$. Suppose the number of these independent k'-outerplanar graphs is L^i. W.l.o.g. suppose these k'-outerplanar graphs are ordered from exterior to interior as $G_s = (V_s, E_s)$, $s \in [L^i]$ (e.g. G_s is the subgraph of G constructed by all the vertices of levels from $(s-2)k+i+1$ to $(s-1)k+i$, $s = 2, \cdots, L^i - 1$, with boundary $x_v = 0$ if v is of level $(s-2)k+i+1$ or $(s-1)k+i$).

Let $\Omega_s(Q^s)$ denote the optimal value if there is a solution such that the total allocated scaled emission amount to G_s is exactly Q^s with boundary condition and $\Omega_s(Q^s) = 0$ otherwise, which can be solved by the above DP approach on k'-outerplanar graphs with boundary conditions. Then, it is not difficult to see the optimal solution for $\bar{PG}_2(i)$ is the optimal solution of the following integer linear program (denoted SUB):

$$\begin{aligned} \max\ & \textstyle\sum_{s\in[L^i]} \sum_{Q^s=0}^{p} \Omega_s(Q^s) y_{sQ^s} \\ \text{s.t.}\ & \textstyle\sum_{s\in[L^i]} \sum_{Q^s=0}^{p} Q^s y_{sQ^s} \leq p \\ & \textstyle\sum_{Q^s=0}^{p} y_{sQ^s} = 1 \\ & y_{sQ^s} \in \{0,1\} \forall s \in [L^i],\ Q^s \in [p] \end{aligned}$$

Let $g_t(Q)$ denote the optimal integer value of SUB when only G_s, $s \in [t]$ is considered and the total emission amount allocated to these graphs is exactly Q. Then we have the following recursion function: $g_t(Q) = \max_{Q^t=0,1,\cdots,Q}\{g_{t-1}(Q - Q^t) + \Omega_t(Q^t)\}$. The optimal value of SUB is $\max_{Q=0,1,\cdots,p}\{g_{L^i}(Q)\}$, which gives the optimal solution of $\bar{PG}_2(i)$ by tracking the optimal value of this dynamic programming approach. The running time of this approach is $O(|L^i|p^2)$. Hence, the total running time for obtaining and solving $\bar{PG}_2(i)$ is $O(|L^i|p^2) + \sum_{s\in[L^i]} O(k|V_s|^3 q^{6k+2} \lceil \frac{2\rho}{\delta} \rceil^{6k}) = O(k|V|^3 q^{6k+2} \lceil \frac{2\rho}{\delta} \rceil^{6k})$. We need to solve $\bar{PG}_2(i)$, for each $i \in [k]$ and then get $\max_{i\in[k]}\{OPT(\bar{PG}_2(i))\}$. Therefore, the overall running time is $O(k^2|V|^3 q^{6k+2} \lceil \frac{2\rho}{\delta} \rceil^{6k})$, and Theorem 5 is proved. □

Let $\frac{2}{k} = \epsilon$ in Theorem 5. Also note that $\rho = |V|(\lfloor \log_2(q) \rfloor + 2)$. We have:

Theorem 6. *There is* $O\left(\frac{1}{\epsilon^2}|V|^{12/\epsilon+3} q^2 \lceil \frac{2(\lfloor \log_2 q \rfloor + 2)q}{\delta} \rceil^{12/\epsilon+1}\right) = \left(\frac{|V|q(\log_2 q + 2)}{\delta}\right)^{O(\frac{1}{\epsilon})}$ *time algorithm for PG for fixed* $\delta, \epsilon > 0$ *on directed planar graph* $G = (V, E)$ *that achieves social welfare* $(1-\epsilon)OPT^{in}(PG)$ *and violates each local constraint by a factor of* $1+\delta$. *This is a PTAS for PG with polynomial size integer variables.*

5 General Objective Function for Bounded Degree Graphs

Full details of our results for general objective functions will appear in the full version of the paper. Our most general algorithmic result is given in Theorem 7.

Theorem 7. *Let* $x_v \in \{0,1\}$ *for any* $v \in V$. *Assume that* $R(x)$ *is monotone increasing as set function on sets* $S \subseteq V$ *s.t.* $v \in S$ *iff* $x_v = 1$. *Then there is an* $(\eta^{fr} = \frac{e\gamma_{\Delta+2}}{e-1} + 1)$*-approximation algorithm for PG with integer variables on graphs with degree* $\leq \Delta$.

Our hardness results for general objective functions are Theorems 8 and 9. By a reduction from independent set we get the following:

Theorem 8. *PG is Unique Games-hard to approximate within* $n^{1-\epsilon}$ *and within* $\frac{\Delta}{\log^2 \Delta}$ *for G with degree* Δ *when* p_v *is any constant number* ≥ 1, $b_v(x_v)$ *is linear and* $d_v(y)$ *is piecewise linear (with 2 pieces)* $\forall v \in V$ *and* w_{vu} *is positive constant* $\forall (v,u) \in E$.

Theorem 9. *It is strongly NP-hard to find an optimal solution to Pollution Game (PG) when* p_v *is any constant number* ≥ 1 *,* $b_v(x_v)$ *is linear and* $d_v(y)$ *is piecewise linear (with two pieces)* $\forall v \in V$ *and* w_{vu} *is positive constant for any* $(v, u) \in E$*.*

References

1. Air quality in Europe - 2014 Report. European Environment Agency Report No. 5/2014 http://www.eea.europa.eu/publications/air-quality-in-europe-2014.
2. State of the Air 2014 Report. *American Lung Association*, 30 April, 2014. http://www.stateoftheair.org/2014/key-findings/.
3. Eleftherios Anastasiadis, Xiaotie Deng, Piotr Krysta, Minming Li, Han Qiao, and Jinshan Zhang. Network pollution games. *AAMAS*, 2016.
4. B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. of the ACM*, 41(1):153–180, 1994.
5. N. Bansal, N. Korula, V. Nagarajan, and A. Srinivasan. Solving packing integer programs via randomized rounding with alterations. *Theory of Computing*, 8(1):533–565, 2012.
6. A.V. Belitskaya. Network Game of Pollution Cost Reduction. *Contributions to Game Theory and Management*, 6(0):24–34, 2013.
7. A. Caprara, H. Kellerer, U. Pferschy, and D. Pisinger. Approximation algorithms for knapsack problems with cardinality constraints. *European Journal of Operational Research*, 123(2):333–345, 2000.
8. C. Chau, K. Elbassioni, and M. Khonji. Truthful mechanisms for combinatorial ac electric power allocation. In *Proc. of the 13th AAMAS*, pages 1005–1012, 2014.
9. S. Dobzinski and N. Nisan. Mechanisms for multi-unit auctions. In *Proc. of the 8th ACM conference on Electronic commerce*, pages 346–351. ACM, 2007.
10. B. Dong, D. Ni, and Y. Wang. Sharing a polluted river network. *Environmental and Resource Economics*, 53(3), 2012.
11. S. Dughmi and T. Roughgarden. Black-box randomized reductions in algorithmic mechanism design. *SIAM Journal on Computing*, 43(1):312–336, 2014.
12. I. Katsikarelis. Computing bounded-width tree and branch decompositions of k-outerplanar graphs. *arXiv preprint arXiv:1301.5896*, 2013.
13. T. Kloks. *Treewidth: computations and approximations*, volume 842. Springer, 1994.
14. P. Krysta, O. Telelis, and C. Ventre. Mechanisms for multi-unit combinatorial auctions with a few distinct goods. In *Proc. 12th AAMAS*, pages 691–698, 2013.
15. E. Kwerel. To tell the truth: Imperfect information and optimal pollution control. *The Review of Economic Studies*, pages 595–601, 1977.
16. R. Lavi and C. Swamy. Truthful and near-optimal mechanism design via linear programming. *Journal of the ACM (JACM)*, 58(6):25, 2011.
17. Debing Ni and Yuntong Wang. Sharing a polluted river. *Games and Economic Behavior*, 60(1):176–186, 2007.
18. Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic game theory*, volume 1. Cambridge University Press Cambridge, 2007.
19. C.J. Vorosmarty, P.B. McIntyre, M.O. Gessner, D. Dudgeon, A. Prusevich, P. Green, s. Glidden, S.E. Bunn, C.A. Sullivan, C. Reidy Liermann, and P.M. Davies. Global threats to human water security and river biodiversity. *Nature*, 467:555–561, 2010.