Optimal Pricing in Markets with Non-Convex Costs

Navid Azizan, California Institute of Technology
Yu Su, California Institute of Technology
Krishnamurthy Dvijotham, Google DeepMind
Adam Wierman, California Institute of Technology

We consider a market run by an operator, who seeks to satisfy a given consumer demand for a commodity by purchasing the needed amount from a group of competing suppliers with non-convex cost functions. The operator knows the suppliers' cost functions and announces a price/payment function for each supplier, which determines the payment to that supplier for producing different quantities. Each supplier then makes an individual decision about how much to produce, in order to maximize its own profit. The key question is how to design the price functions. To that end, we propose a new pricing scheme, which is applicable to general non-convex costs, and allows using general parametric pricing functions. Optimizing for the quantities and the price parameters simultaneously, and the ability to use general parametric pricing functions allows our scheme to find prices that are typically economically more efficient and less discriminatory than those of the existing schemes. In addition, we supplement the proposed method with a polynomial-time approximation algorithm, which can be used to approximate the optimal quantities and prices. Our framework extends to the case of networked markets, which, to the best of our knowledge, has not been considered in previous work.

1 INTRODUCTION

While there has been a long history of studying markets under convexity assumptions in economic theory, *non-convexities* are ubiquitous in most real-world markets. They arise due to start-up or shut-down costs, indivisibilities, avoidable costs, or simply economies of scale.

It has been widely noted in the literature that in the presence of non-convexities, there may be no linear prices (constant per quantity) that support a competitive market equilibrium, e.g., [Brown, 1991, Guesnerie, 1975], and it was suggested as early as 1980s that in these markets one needs to consider using *price functions*, as opposed to the conventional prices [Wolsey, 1981]. Following the work of [Scarf, 1990, 1994], there have been many pricing schemes proposed in the literature. In particular, during the past decade, motivated by the deregulation of the electricity markets in the US and around the world, the problem of pricing in non-convex markets has attracted renewed interest from researchers, and there has been considerable work on this problem [Liberopoulos and Andrianesis, 2016]. These schemes are deployed in practice, and the operation and efficiency of our electricity markets relies on them [Azizan Ruhi et al., 2017].

Formally, the non-convex pricing problem is that, given an inelastic demand for a commodity from a number of consumers, a market operator seeks to satisfy the demand by purchasing the required amount from a group of competing suppliers with non-convex cost functions. The operator knows the suppliers' cost functions, and it announces a price/payment function for each supplier, which determines the payment to that supplier for producing different quantities. Each supplier then makes an individual decision about how much to produce in order to maximize its own profit. The question of interest is then how to design the price functions, which is quite different from mechanism design since the cost functions of the suppliers are known to the market operator. However, even though the cost functions are known, the design of the price functions in these markets is difficult.

An important early approach to the pricing problem was the work of [O'Neill et al., 2005], sometimes referred to as integer programming (IP) pricing, which considered the class of non-convexities that arise from the start-up costs of the suppliers (with linear marginal costs). The

Preprint.

paper proposes a clever pricing rule, based on solving a mixed-integer linear program and forcing the integral variables to their optimal values as a constraint. The scheme is economically efficient and has a nice dual interpretation. Modified versions of IP pricing have been proposed by [Bjørndal and Jörnsten, 2008, 2010] and others. Another approach, proposed for the more general class of non-convex cost functions that are in the form of a start-up plus a convex (rather than linear) cost, is the minimum-uplift (MU) pricing of [Hogan and Ring, 2003], and its closely related refinement of [Gribik et al., 2007], known as convex hull (CH) pricing. These schemes provide discriminatory uplifts to different suppliers to incentivize production, and the uplifts are minimal in a specific sense [Schiro et al., 2016]. The possibility of having both positive and negative uplifts was also considered by [Galiana et al., 2003, Motto and Galiana, 2002]. Other pricing schemes include the semi-Lagrangian relaxation (SLR) approach of [Araoz and Jörnsten, 2011], and the primal-dual (PD) approach of [Ruiz et al., 2012]. These schemes seek to find uniform linear prices that are revenue-adequate (but not supporting of the equilibrium). A survey on all the above pricing schemes was recently written by Liberopoulos and Andrianesis [2016]. However, the overall desired properties, as well as the properties that each of the schemes satisfy, were not examined there. We formalize the desired properties considered in the literature in Section 2, and discuss the properties of the existing schemes in Section 3. Table 1 summarizes the common schemes and their properties.

Despite the large body of work on the pricing problem, the existing schemes have several shortcomings. For example, most of the existing schemes mentioned above are proposed for specific classes of non-convex cost functions, and cannot handle more general non-convexities. Furthermore, even the ones that are applicable for general cost function fail to satisfy some of the key desired properties of the market, such as economic efficiency or supporting a competitive equilibrium. In addition, none of the existing schemes is accompanied by a computationally tractable algorithm for general non-convexities, and they typically rely on off-the-shelf heuristic solvers for mixed-integer programs that are known to be NP-hard.

In this paper, we propose a pricing scheme for markets with general non-convex costs that designs general parametric price functions and addresses all the issues mentioned above. Our approach seeks to find the optimal schedule (allocation) and the optimal pricing rule simultaneously, which generally allows for finding *economically more efficient* solutions. The ability to use general price functions (e.g. piece-wise linear, quadratic, etc.) enables our approach to design price functions that are *less discriminatory*, while still *supporting a competitive equilibrium*. Furthermore, our pricing scheme is accompanied by a *computationally efficient* (polynomial-time) approximation algorithm which allows one to find the approximately-optimal schedule and prices for general non-convex cost functions. Lastly, our proposed pricing rule can be extended to *networked* markets, which, to the best of our knowledge, are not considered in any of the existing work.

Specifically, this paper makes the following contributions.

(1) We survey the common pricing schemes proposed in the literature for markets with non-convex costs and provide a compact summary of their properties (Section 3).
(2) We propose a framework for pricing in markets with *general* non-convex costs, using *general* price functions (Section 4.1). Our scheme seeks to find the optimal price functions and allocations simultaneously, while imposing the equilibrium conditions as constraints. For this reason, our approach is generally economically more efficient than the existing methods, while satisfying the equilibrium conditions. Moreover, the ability to use general price forms allows one to obtain more uniform prices (smaller "uplifts").
(3) We supplement our pricing scheme with a computationally efficient (polynomial-time) approximation algorithm for finding the allocations and prices (Section 4.2).
(4) We extend our framework to networked markets (Section 5), and also propose an approximation algorithm that can compute the solution efficiently for acyclic networks, a common scenario in electric distribution networks.

2 MARKET DESCRIPTION AND PRICING OBJECTIVES

While our goal in this paper is to design an economically and computationally efficient pricing scheme for non-convex *networked* markets, we begin with the problem of designing one for a *single* market, which is difficult in its own right. We return to the case of networked markets in Section 5. When the cost functions are non-convex, even this seemingly simple problem has proven to be challenging, and a wide variety of pricing schemes have been proposed for it in the literature. In the following, we describe the market model and survey the desired market properties.

2.1 Market Model

We consider a single market consisting of n competing suppliers (often referred to as firms or generators). The market is run by a market operator that seeks to satisfy a deterministic and inelastic demand d for a commodity in a single period. Each supplier i has a cost function $c_i(q_i)$ for producing quantity q_i, which may be non-convex.

The suppliers' cost functions are known by the operator, and the operator uses them to determine the prices. In particular, the operator announces price/payment functions $p_i(q_i)$, which determine the payment to supplier i when producing q_i. Note that, in general, the price functions can be different for different suppliers, but it is often desired to have close-to-uniform prices.

Upon the announcement of the price functions, each supplier i makes an individual decision, based on the price function $p_i(\cdot)$ and the cost function $c_i(\cdot)$, about how much to produce (and whether to participate in the market), in order to maximize its own profit, i.e., $p_i(q_i) - c_i(q_i)$. The suppliers are then paid for their production according to the payment function, and the demand (consumers) is charged for the total payment.

This model is classical, and has been studied in a wide variety of contexts, initially under the assumption of convex cost functions for production and linear pricing functions, but more recently under non-convex cost functions. Non-convex cost functions are particularly important in the context of electricity markets. As a result, there is a large literature focusing on non-convex pricing in electricity markets, see [Liberopoulos and Andrianesis, 2016] for a recent survey. Often this literature assumes specific forms of non-convexities (e.g., startup/fixed costs), and specific forms of payment functions (e.g., linear plus uplift). The results from this literature have guided the design and operation of electricity markets across the world today.

2.2 Pricing Objectives

The key design question in the market described above is how to devise the payment functions. The goal of the operator is to (1) find the optimal quantities q_i^*, and (2) design the payment functions $p_i(\cdot)$ that ensure that the suppliers produce the optimal quantities q_i^*.

There is a huge design space for such payment functions, and there is a large literature evaluating proposals in the context of non-convex cost functions, e.g., [Araoz and Jörnsten, 2011, Bjørndal and Jörnsten, 2008, Gribik et al., 2007, Hogan and Ring, 2003, Hua and Baldick, 2017, Liberopoulos and Andrianesis, 2016, O'Neill et al., 2005, Ruiz et al., 2012, Schiro et al., 2016].

From this literature has emerged a variety of desirable properties which pricing rules attempt to satisfy. The following is a summary of the most sought-after properties in this literature. Note that no existing rules satisfy all of these properties for general non-convex markets.

(1) **Market Clearing** (a.k.a. **Load Balancing**): The total supply is equal to the demand, i.e., $\sum_{i=1}^{n} q_i^* = d$.
(2) **Economic Efficiency**
   (a) **Minimal Production Cost** *(Suppliers' Total Cost)*: The total production cost of the suppliers, i.e. $\sum_{i=1}^{n} c_i(q_i^*)$, is minimal.
   (b) **Minimal Payment** *(Total Paid Cost)*: The total cost that is paid to the suppliers for the commodity, i.e. $\sum_{i=1}^{n} p_i(q_i^*)$, is minimal.

(3) **Incentivizing**
   (a) **Revenue Adequacy**: For every supplier, the net profit at the optimum is nonnegative, i.e., $p_i(q_i^*) - c_i(q_i^*) \geq 0$, for $i = 1, \ldots, n$.
   (b) **Support a Competitive Equilibrium**: The optimum production quantity for each supplier is a maximizer of its individual profit, i.e., $q_i^* \in \arg\max_{q_i} \; p_i(q_i) - c_i(q_i)$, or equivalently $p_i(q_i^*) - c_i(q_i^*) \geq \max_{q_i \neq q_i^*} \; p_i(q_i) - c_i(q_i)$, for $i = 1, \ldots, n$.
(4) **Simplicity and Uniformity**: The price functions are simple and interpretable (ideally linear: $p_i(q_i) = \lambda_i q_i$) and non-discriminatory (ideally uniform across suppliers: $p_i(q_i) = p(q_i)$).
(5) **Computational Tractability**: The optimal quantities and price functions can be computed/approximated in time that is polynomial in n.

A few remarks about these properties are warranted. Property 1 ensures that the demand is met. Property 2 is somewhat more elaborate and concerns the economic efficiency of the scheme, in terms of total expenditure. Even though in certain cases (e.g. in IP pricing of [O'Neill et al., 2005] for startup-plus-linear costs), the suppliers' total cost $\sum_{i=1}^n c_i(q_i)$ and the total paid cost $\sum_{i=1}^n p_i(q_i)$ match and are both minimal, there is an inevitable gap between the two in general. Ultimately, the quantity which determines the cost of satisfying the demand is the total payment to the suppliers $\sum_{i=1}^n p_i(q_i)$, and therefore Property 2b is arguably more crucial than Property 2a. However, ostensibly, because the price functions are not directly available while computing the optimal quantities, many pricing schemes have resorted to minimizing the total suppliers' cost $\sum_{i=1}^n c_i(q_i)$ as a surrogate for the paid cost. In this paper, we advocate a direct approach for minimizing the total payment.

Property 3 incentivizes the suppliers to follow the dispatch and produce the socially-optimal quantities. More specifically, Property 3a ensures that the suppliers do not lose by producing q_i^*, and further, Property 3b assures that it is in each supplier's best interest to follow the dispatch. Property 3b is generally a stronger condition than Property 3a, and if $p_i(0) = c_i(0) = 0 \; \forall i$, then (3b) implies (3a).

Property 4 concerns having price forms that are "close to linear" (simple) and "close to uniform" (non-discriminatory), in some sense. One common approach to this is to use uniform linear prices plus a generator-dependent "uplift," i.e. $p_i(q_i) = \lambda q_i + u_i \mathbb{1}_{q_i = q_i^*}$, and try to minimize the uplifts u_i. As Property 4 is subjective by its nature, we allow arbitrary parametrized price functions in our scheme. However, we also examine our scheme when applied to the popular minimal-uplift approach. Note that Property 4 also rules out the use of "dictatorial" prices, in which the operator pays the cost (plus an ϵ) only at the desired amount, and pays nothing for any other amount produced.

The final property, Computational Tractability, is particularly challenging to address in the context of non-convex markets. Nearly all standard approaches work by computing the optimal production quantities and then deriving the prices from these quantities in some way. Under general non-convex cost functions, this first step is already computationally intractable. Thus, it is important to consider relaxations (approximations) of other properties if the goal is to enforce computational tractability. To that end, we consider approximate versions of the Incentivizing and Economic Efficiency conditions, which we discuss in Section 4.2. We propose an algorithm that satisfies these approximate versions, while being computationally tractable.

3 EXISTING PRICING SCHEMES

To this point, no pricing design for general non-convex markets satisfies all the properties discussed above. However, it is possible to achieve all the properties in the case when the cost functions are convex via a classical approach: *shadow pricing*. In this section, we briefly illustrate how shadow

pricing works for the convex case, and then survey some prominent approaches in the literature that seek to extend the properties of shadow pricing to the non-convex case.

3.1 Pricing in Convex Markets

When the cost functions $c_i(.)$ are convex, a simple and uniform pricing rule, often referred to as *shadow pricing* or *marginal-cost pricing* [Beato and Mas-Colell, 1985, Turvey, 1969], can achieve all the above-mentioned properties. The pricing scheme works as follows. The operator first solves the convex program

$$\min_{q_1,\dots,q_n} \quad \sum_{i=1}^{n} c_i(q_i) \tag{1a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d \qquad (\lambda) \tag{1b}$$

where λ is the dual variable corresponding to the load-balance constraint. Let $q_1^*, \dots, q_n^*$ and λ^* denote an optimal primal-dual pair of this problem (if there are multiple dual solutions, take λ^* to be the smallest). A payment function of the form

$$p_i(q_i) = \lambda^* q_i \quad i = 1, \dots, n \tag{2}$$

satisfies all the properties outlined in Section 2.2, and it is relatively straightforward to see that.

For simplicity assume that $c_i(.)$ are differentiable. The optimal solution of (1) satisfies the following (KKT) conditions (which does not require convexity):

$$\begin{cases} \sum_{i=1}^{n} q_i^* = d \\ \frac{dc_i}{dq_i}(q_i^*) = \lambda^*, \quad i = 1, \dots, n \end{cases}$$

Next, note that supplier i's profit-maximization problem is

$$\max_{q_i} \quad \lambda^* q_i - c_i(q_i).$$

Since $c_i(.)$ is convex, the objective is concave and any point at which the derivative is zero, is a global maximizer. In particular, the derivative at q_i^* is zero, because of the KKT conditions, and therefore that is a solution to the supplier i's profit-maximization problem. As a result, the scheme supports a competitive equilibrium that clears the market and minimizes the production cost, while using a price form that is simple and uniform.

Note that the total payment of this scheme is $\sum_{i=1}^{n} p_i(q_i^*) = \lambda^* d$, which can be generally higher than $\sum_{i=1}^{n} c_i(q_i^*)$. One can always opt for a *non-uniform* affine price function as $p_i(q_i) = \lambda^* q_i + b_i$, with $b_i = c_i(q_i^*) - \lambda^* q_i^*$, which has lower payments, and makes $\sum_{i=1}^{n} p_i(q_i^*)$ exactly equal to $\sum_{i=1}^{n} c_i(q_i^*)$. However, if one requires a *uniform and linear* price function, it can be shown that $p_i(q_i) = \lambda^* q_i$ has the *lowest total payment* among all such functions.

3.2 Pricing in Non-Convex Markets

If the cost functions are non-convex, the approach of shadow pricing, described above, fails. This is because the net profit of each supplier is no longer a concave function, and its stationary points do not necessarily correspond to the maximum. In other words, there may not be a subderivative at q_i^* supporting the cost function $c_i(.)$.

There have been several schemes proposed in the literature that attempt to address this issue and design pricing rules that satisfy the properties discussed above in the context of non-convex cost functions. We review the most promising ones here. Some of the schemes maintain a uniform pricing rule with additional discriminatory side-payments called "uplifts" for incentivizing the

Table 1. Summary of common pricing schemes and their properties. IP: Integer Programming, MU: Minimum Uplift, CH: Convex Hull, SLR: Semi-Lagrangean Relaxation, PD: Primal-Dual, EC: Equilibrium-Constrained

Scheme \ Property	Proposed for $c_i(q_i) =$	Price form $p_i(q_i) =$	Market Clearing	Revenue Adequate	Supports Competitive Equilibrium	Economically Efficient	Applicable to general costs	Computationally Efficient in general
Shadow Pricing	Convex	λq_i	✓	✓	✓	✓	×	N/A
IP	Startup+linear	$\lambda q_i + u_i \mathbb{1}_{q_i>0}$	✓	✓	✓	✓	×	N/A
MU/CH	Startup+convex	$\lambda q_i + u_i \mathbb{1}_{q_i=q_i^*}$	✓	✓	✓	×	✓	×
SLR	Startup+linear	λq_i	✓	✓	×	×	×	N/A
PD	Startup+linear	λq_i	✓	✓	×	×	×	N/A
EC (proposed)	General	General	✓	✓	✓	✓	✓	✓

suppliers to follow the dispatch, while others raise the uniform price so that it is revenue-adequate. A summary of the pricing schemes, along with their properties, is provided in Table 1.

Integer Programming (IP). A pricing scheme for non-convex cost functions that are in the form of a fixed (start-up) cost plus a linear marginal cost, sometimes referred to as "IP pricing" was proposed in [O'Neill et al., 2005]. This scheme uses uniform marginal pricing for the commodity and discriminatory pricing for the integral activity of the suppliers. It is based on (i) formulating an optimization similar to (1), as a mixed integer linear program (MILP) and solving it for optimal allocations, (ii) reformulating the original MILP as an LP by replacing the integral constraints with forcing commitment choices equal to their optimal values, and (iii) solving the LP problem and using the dual variable λ of Market Clearing constraint as the uniform price and the dual variables $\{u_i^*\}$ of the forced equality constraints as discriminatory uplifts: $p_i(q_i) = \lambda^* q_i + u_i^* \mathbb{1}\{q_i > 0\}$.

IP pricing uses a uniform price plus a discriminatory uplift to clear the market efficiently such that every supplier's net profit is zero. As a result, both total payments and total production costs are minimized at the same time. It was shown in [O'Neill et al., 2005] that the optimal solutions generated by IP pricing are optimal to the decentralized profit maximization problems for every supplier and thus they support a competitive equilibrium. However, IP pricing assumes knowledge of the optimal solutions to the unit commitment problem and thus is not intended as a practical approach to find the optimal allocation. It is pointed out in [Hogan and Ring, 2003] that uniform price generated under IP pricing can be volatile (i.e. a small change in demand could lead to a big change in the uniform price) and uplifts could be generally very large.

Minimum Uplift (MU) / Convex Hull (CH). To avoid the unwanted properties of IP pricing (i.e. volatility and instability), a pricing scheme, proposed in [Hogan and Ring, 2003] for the (non-convex) class of startup-plus-convex cost functions, offers minimum uplifts that incentivize each supplier to follow the dispatch rather than maximize their own profits in the absence of uplifts. The scheme is based on solving the mixed-integer program minimizing the total production cost and minimizing total uplifts. Given a fixed uniform price λ, each supplier chooses between following the dispatch to receive the uplifts or not. The uplifts can be viewed as the extra potential profit that the suppliers can make by self-scheduling and maximizing their own profit. The MU pricing was refined by [Gribik et al., 2007] and they proposed the concept of Convex Hull pricing, which is based on (i) replacing the non-convex cost of the original program with its convex hull to formulate a new LP, (ii) solving the new LP and using the dual variable of Market Clearing constraint as the marginal price and deriving the lost opportunity cost (LOC) as the minimum uplifts to incentivize suppliers'

compliance. The final payment $p_i(q_i, z_i)$ as a function of quantity q_i and commitment choice z_i is in the form of a uniform price λ^* and a discriminatory uplift u_i^* as $p_i(q_i) = \lambda^* q_i + u_i^* \mathbb{1}\left\{q_i = q_i^*\right\}$.

Even though MU/CH pricing minimizes total uplifts, the generated marginal price might end up being high, and the payments can be much higher than those of the other schemes. In general, the total payments under this scheme might end up being much higher than the total production costs, which defeats the purpose of minimizing the costs. Even for the class of startup-plus-linear cost functions, where IP pricing is optimal (the total payment is equal to the total production cost, and they are both minimal), MU pricing is not economically efficient, as it fails to minimize the payments.

On the computational side, although a polynomially-solvable primal formulation for the Lagrangian dual problem by explicitly describing the convex hull for piecewise linear or quadratic cost functions was proposed in [Hua and Baldick, 2017], describing the convex hull of cost functions could be very challenging in general and thus makes the problem computationally intractable.

Semi-Lagrangean Relaxation (SLR). A semi-Lagrangean relaxation approach to find a uniform price that is revenue-adequate at the same solution for quantity and commitment choices as the original optimization problem, for cost functions of startup-plus-linear form, was proposed in [Araoz and Jörnsten, 2011]. The scheme is based on formulating and solving the SLR of mixed-integer program (MIP) by semi-relaxing the Market Clearing constraint with standard Lagrange multiplier λ. The solution under SLR satisfies the constraints in the original MIP and makes the duality gap between MILP and SLR zero. Though the payment function $p_i(q_i) = \lambda^* q_i$ under SLR pricing is high enough to avoid negative profits for suppliers, it incentivizes the suppliers to deviate and operate at full capacity and total payments usually end up being much higher than total costs of production.

Primal-Dual (PD). Another revenue-adequate pricing scheme, proposed by [Ruiz et al., 2012], exploits a primal-dual approach to derive a uniform price to guarantee that dispatched suppliers are willing to remain in the market (revenue adequacy). The scheme works for cost functions with the form of start-up cost plus linear cost, and the prices have shown not to deviate much from that of [O'Neill et al., 2005]. The approach is based on (i) relaxing the integral constraint of the original MILP to formulate a primal LP problem, (ii) deriving the dual LP problem of the primal LP problem, (iii) formulating a new LP problem that seeks to minimize the duality gap between the primal and dual problems subject to both primal and dual constraints and (iv) adding back the integral constraints as well as nonlinear constraints to ensure that no supplier incurs loss and solving the new problem for optimal solutions q_i^*, z_i^* and λ^*.

Though this scheme may be implemented using standard branch-and-cut solvers, it is computationally intractable in general. The prices $p_i(q_i) = \lambda^* q_i$ and profits produced under PD do not significantly deviate from dual prices if integral constraints are relaxed and thus are always bounded. However, as a revenue-adequate pricing scheme, PD fails to form a competitive equilibrium as suppliers are incentivized to operate at full capacity. In general, total payments are much higher than total production costs.

4 EQUILIBRIUM-CONSTRAINED (EC) PRICING

As mentioned in the previous section, most existing schemes in the literature are proposed for specific classes of non-convexities, and are not applicable for more general non-convex costs. Furthermore, even the ones that are applicable for more general cost functions either already lack some of the key properties (such as economic efficiency) or they lose those properties for more general costs. Additionally, the existing schemes are not accompanied by a computationally tractable algorithm for general non-convexities, and they typically rely on off-the-shelf heuristic

solvers for mixed-integer programs that are NP-hard. This serves to emphasize that no existing pricing scheme satisfies the desired properties described in Section 2.2.

The main contribution of this paper is the introduction of a new pricing scheme, *Equilibrium-Constrained (EC) pricing*, which is applicable to general non-convex costs, allows using general parametric price functions, and satisfies all the desired properties outlined before, as long as the price class is general enough. The name of this scheme stems from the fact that we directly impose all the equilibrium conditions as constraints in the optimization problem for finding the best allocations, as opposed to adjusting the prices later to make the allocations an equilibrium. The optimization problem is, of course, non-convex, and non-convex problems are intractable in general. However, we also present a tractable approximation algorithm for approximately solving the proposed optimization.

We present the formulation of the optimization at the core of Equilibrium-Constrained pricing in Section 4.1, and then develop an efficient algorithm for solving the optimization problem approximately in Section 4.2.

4.1 Pricing Formulation

In this section, we propose a systematic approach for determining a pricing rule under generic non-convex costs that minimizes payments and satisfies the properties outlined in Section 2.2, while allowing flexibility in the choice of the form of price functions.

Specifically, consider a class of desired price functions, denoted by $\mathcal{P}$, which can be an arbitrary class such as linear, linear plus uplift, piece-wise linear, etc. This choice can be due to interpretability/uniformity reasons or other practical considerations. The core of Equilibrium-Constrained pricing is an optimization problem for finding the best price functions in $\mathcal{P}$ and the best allocations, at the same time. The operator is buying the commodity from the suppliers, on behalf of the consumers, and therefore its objective is to minimize the total cost incurred (total payment), subject to the equilibrium constraints. The optimization problem can be expressed as follows.

Equilibrium-Constrained (EC) Pricing:

$$p^* = \min_{\substack{q_1,\dots,q_n \\ p_1,\dots,p_n \in \mathcal{P}}} \quad \sum_{i=1}^{n} p_i(q_i) \tag{3a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d \tag{3b}$$

$$p_i(q_i) - c_i(q_i) \geq 0, \quad i = 1, \dots, n \tag{3c}$$

$$p_i(q_i) - c_i(q_i) \geq \max_{q_i' \neq q_i} \; p_i(q_i') - c_i(q_i'), \quad i = 1, \dots, n \tag{3d}$$

Constraints (3b), (3c) and (3d) are the Market Clearing, Revenue Adequacy, and Competitive Equilibrium conditions, respectively. Constraint (3d) can also be equivalently expressed as

$$p_i(q_i) - c_i(q_i) \geq p_i(q_i') - c_i(q_i') \quad \forall q_i' \neq q_i, \quad i = 1, \dots, n. \tag{4}$$

The key difference between EC pricing and the existing methods for pricing in non-convex markets is that it directly minimizes the total paid cost and seeks to find both the optimal allocations q_i^* and the optimal price functions $p_i^*(.)$ simultaneously. The scheme enforces the desired economic properties as constraints, while allowing the use of any class of price functions, rather than imposing a fixed form for the price.

Remark 4.1. *It is easy to see, by relaxing the last constraint and using constraint* (3c)*, that the optimal value of the above optimization problem p^* is bounded below by the minimum total production*

cost $c^* = \sum_{i=1}^n c_i(q_i^0)$*, where*

$$(q_i^0, \ldots, q_i^n) = \underset{q_1,\ldots,q_n}{\arg\min} \quad \sum_{i=1}^n c_i(q_i) \tag{5a}$$

$$\text{s.t.} \quad \sum_{i=1}^n q_i = d \tag{5b}$$

is the "minimal production cost" solution. Mathematically, we have

$$p^* \geq \min_{\substack{q_1,\ldots,q_n \\ p_1,\ldots,p_n}} \sum_{i=1}^n p_i(q_i) \qquad \geq \min_{q_1,\ldots,q_n} \sum_{i=1}^n c_i(q_i) \qquad = c^*$$
$$\text{s.t.} \quad \sum_{i=1}^n q_i = d \qquad\qquad \text{s.t.} \quad \sum_{i=1}^n q_i = d$$
$$p_i(q_i) \geq c_i(q_i), \quad i = 1, \ldots, n$$

This emphasizes that in any scheme that satisfies Revenue Adequacy, the total production cost is upper-bounded by the total payment. Therefore, minimizing the total payment minimizes the total production cost as well, while the opposite is not true in general (minimizing the total production cost can result in very high payments).

Remark 4.2. *We have imposed nearly all the desired properties as constraints in the optimization problem* (3)*, and it might not be clear whether this optimization problem has a solution at all. Indeed, there always exists a class of price functions for which problem* (3) *has a solution, and further the bound mentioned in Remark 4.1 is achieved.*

A naive choice of price function is enough to prove this claim. In fact, one can check that for any price function of the form

$$p_i(q_i) \begin{cases} = c_i(q_i) & \text{for } q_i = q_i^0 \\ \leq c_i(q_i) & \text{for } q_i \neq q_i^0 \end{cases}$$

problem (3) *has an optimal solution* $q^* = q^0$*, and achieves the bound* $p^* = c^*$*.*

While Remark 4.2 asserts the existence of an optimal price function, of course for specific classes of price functions, problem (3) may not have a solution. The key point is that problem (3) always allows using more sophisticated price forms (e.g. piece-wise linear) for which it will have a solution; and for any given choice of price form, it finds the best one, along with the optimal quantities.

Remark 4.3. *While in most scenarios the operator is buying the commodity from the suppliers on behalf of the consumers, and it makes sense to minimize the total payments* $\sum_{i=1}^n p_i(q_i)$*, in general one may seek to balance between the consumers' and the suppliers' costs. In other words, one can take the objective to be a linear combination of the consumers' cost* $\sum_{i=1}^n p_i(q_i)$ *and the suppliers' net cost (negative profit)* $\sum_{i=1}^n (c_i(q_i) - p_i(q_i))$*. Without loss of generality, the weighted some can be normalized to an affine (i.e. convex) combination* $(1-\theta)\sum_{i=1}^n p_i(q_i) + \theta \sum_{i=1}^n (c_i(q_i) - p_i(q_i))$ *with parameter* θ*. The optimization can be expressed as follows.*

$$p^* = \min_{\substack{q_1,\ldots,q_n \\ p_1,\ldots,p_n \in \mathcal{P}}} \quad (1-2\theta)\sum_{i=1}^{n} p_i(q_i) + \theta \sum_{i=1}^{n} c_i(q_i) \tag{6a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d \tag{6b}$$

$$p_i(q_i) - c_i(q_i) \geq 0, \quad i = 1,\ldots,n \tag{6c}$$

$$p_i(q_i) - c_i(q_i) \geq \max_{q_i' \neq q_i} \; p_i(q_i') - c_i(q_i'), \quad i = 1,\ldots,n \tag{6d}$$

For the cases when the total payment $\sum_{i=1}^{n} p_i(q_i^*)$ *from the optimization problem* (3) *is equal to* $\sum_{i=1}^{n} c_i(q_i^*)$*, the solution from* (6) *is the same as that of* (3)*. It is worth mentioning that our algorithm proposed in Section 4.2 for solving* (3) *is also capable of handling the weighted objective* (6a)*. However, for the sake of simplicity, we focus on the case of* $\theta = 0$*.*

To be more explicit about the class of price functions, we consider a general parametric form for $\mathcal{P}$, specified by $p_i(q_i) := p(q_i; \alpha, \beta_i)$ with two types of parameters $\alpha \in \mathbb{R}^{l_1}$, and $\beta_i \in \mathbb{R}^{l_2}$ for $i = 1,\ldots,n$, where parameter α is shared among all the suppliers, and it constitutes the uniform component of the price, while parameter β_i is specific to supplier i. The parameters are in general constrained to be in some bounded sets $\mathcal{A} \subseteq \mathbb{R}^{l_1}$ and $\mathcal{B} \subseteq \mathbb{R}^{l_2}$, i.e., $\alpha \in \mathcal{A}$, and $\beta_i \in \mathcal{B}$ for all $i = 1,\ldots,n$. This parametric form is general enough that it encompasses all the assumed price forms in the literature. In particular, the linear-plus-uplift form ($p_i(q_i) = \lambda q_i + u_i \mathbb{1}_{q_i = \hat{q}_i}$) is a special case of this form, where the shared parameter is the uniform price λ, and the individual parameters are the amount and location of the uplifts $u_i, \hat{q}_i$. Using the general parametric form, the optimization problem (3) can be re-expressed as follows.

Parameterized Equilibrium-Constrained (EC) Pricing :

$$p^* = \min_{\substack{q_1,\ldots,q_n \\ \alpha \in \mathcal{A} \\ \beta_1,\ldots,\beta_n \in \mathcal{B}}} \quad \sum_{i=1}^{n} p(q_i; \alpha, \beta_i) \tag{7a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d \tag{7b}$$

$$p(q_i; \alpha, \beta_i) - c_i(q_i) \geq 0, \quad i = 1,\ldots,n \tag{7c}$$

$$p(q_i; \alpha, \beta_i) - c_i(q_i) \geq \max_{q_i' \neq q_i} \; p(q_i'; \alpha, \beta_i) - c_i(q_i'), \quad i = 1,\ldots,n \tag{7d}$$

To show a concrete application of this general pricing scheme, we apply our framework to the popular class of linear-plus-uplift price functions, which has been a standard form considered in the electricity markets literature (e.g. in [Gribik et al., 2007, Hogan and Ring, 2003]), and minimize the uplifts. We derive closed-form solutions for the optimal quantities and prices.

4.1.1 ***Linear+Uplift Pricing.*** As mentioned earlier, using a linear uniform price plus an uplift term is a common choice of class of price functions, in practice. For this class, we have $p(q_i; \lambda, u_i, \hat{q}_i) = \lambda q_i + u_i \mathbb{1}_{q_i = \hat{q}_i}$, where $\lambda, u_1, \ldots, u_n \geq 0$. Without loss of generality, we can assume $\hat{q}_i^* = q_i^*$, i.e., the optimal location of uplift coincides with the desired production level, which is

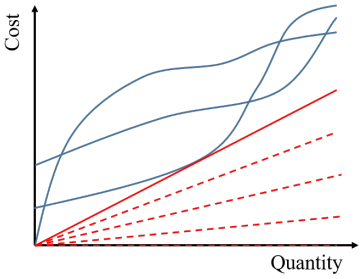


Fig. 1. An illustration of the set Λ for an example with 3 non-convex cost functions. The three blue curves are the cost functions. The (dashed and solid) red lines lie below all the cost functions and their slopes are in Λ. The (slope of the) solid red line corresponds to the largest element of Λ.

intuitive (See the appendix for proof). The optimization problem (7) can then be reduced to

$$p^*_{\text{uplift}} = \min_{\substack{q_1,\ldots,q_n \\ \lambda \geq 0 \\ u_1,\ldots,u_n \geq 0}} \quad \sum_{i=1}^{n} (\lambda q_i + u_i) \tag{8a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d \tag{8b}$$

$$\lambda q_i + u_i - c_i(q_i) \geq 0, \quad i = 1, \ldots, n \tag{8c}$$

$$\lambda q_i + u_i - c_i(q_i) \geq \max_{q_i' \neq q_i} \lambda q_i' - c_i(q_i'), \quad i = 1, \ldots, n \tag{8d}$$

Remark 4.4. *From Remark 4.1, we know that $p^*_{uplift} \geq c^*$. On the other hand, plugging the feasible point $\left(q_i = q_i^0 \ \forall i,\ \lambda = 0,\ u_i = c_i(q_i^0)\ \forall i\right)$ into (8) results in $p^*_{uplift} \leq c^*$. Therefore $p^*_{uplift} = c^*$.*

Problem (8) has potentially many solutions, and the solution $q_i = q_i^0\ \forall i,\ \lambda = 0,\ u_i = c_i(q_i^0)\ \forall i$ corresponds to the naive pay-as-bid scheme, which is equivalent to having no uniform price and paying each supplier for its own cost. To obtain price functions that are close to uniform, it is desirable to pick a solution for which the uplifts are minimum (in ℓ_1 sense, for example). That is equivalent to adding a layer on top of the optimization problem (8) to pick the minimal-uplift solution among all the solutions, i.e.

$$\min_{\boldsymbol{q},\lambda,\boldsymbol{u}} \quad \sum_{i=1}^{n} u_i \tag{9a}$$

$$\text{s.t.} \quad (\boldsymbol{q}, \lambda, \boldsymbol{u}) \in \operatorname*{arg\,min}_{\boldsymbol{q},\lambda,\boldsymbol{u}} \text{(8a)} \tag{9b}$$

$$\text{s.t. (8b), (8c), (8d)} \tag{9c}$$

where $\boldsymbol{q}$ and $\boldsymbol{u}$ denote $(q_1, \ldots, q_n)$ and $(u_1, \ldots, u_n)$, respectively.

Let us define Λ as the set of all λ's for which the linear price λq lies below all the cost functions, i.e.

$$\Lambda = \{\lambda \geq 0 \mid \lambda q \leq c_i(q),\ \forall q, \forall i\}\,. \tag{10}$$

Figure 1 illustrates this set for an example with three non-convex costs.

The solutions to problems (8) and (9) can be found in closed-form, and the following summarizes the results.

Proposition 4.5. *The set of optimal solutions of problem* (8) *is given by*

$$\begin{cases} q_i^* = q_i^0, \ \forall i \\ \lambda^* = \Lambda \\ u_i^* = c_i(q_i^*) - \lambda^* q_i^*, \ \forall i \end{cases}.$$

Proposition 4.6. *Problem* (9) *has a unique optimal solution as*

$$\begin{cases} q_i^* = q_i^0, \ \forall i \\ \lambda^* = \max \ \Lambda \\ u_i^* = c_i(q_i^*) - \lambda^* q_i^*, \ \forall i \end{cases}.$$

See the appendix for proofs.

4.2 An Efficient Approximation Algorithm

The optimization problem (7) defines a pricing rule that satisfies the desired properties in any non-convex market. For specific classes of cost functions, similar to the existing approaches, one may be able to solve this optimization problem using off-the-shelf solvers. For generic non-convex cost functions, however, there is no existing algorithm that can solve the optimization problem (7) to optimality. Furthermore, even finding an approximate solution, e.g., by discretizating the variables, requires a brute-force search, which quickly becomes intractable. In this section, we design a computationally efficient algorithm for solving the problem (7) approximately, based on decomposing it into smaller pieces, which works for general non-convex cost functions. This approximation algorithm can also be used to provide tractable calculations of some of the other non-convex pricing rules such as IP pricing.

Before going through the details of the algorithm, let us define the notion of an approximate solution to (7), which we consider. One could define an approximate solution as a value that is close enough, in a certain sense, to the optimal solution $(q_1^*, \ldots, q_n^*, \alpha^*, \beta_1^*, \ldots, \beta_n^*)$. However, no matter how close is that approximation to the optimal solution, that per se does not guarantee anything about the properties that the scheme will satisfy. Instead, we define an approximate solution to (7) as a set of quantities $q_1, \ldots, q_n$ and price parameters $\alpha, \beta_1, \ldots, \beta_n$ for which the Market Clearing condition holds exactly, the Revenue Adequacy and Competitive Equilibrium conditions are relaxed by an ϵ, and the total payment is at most $n\epsilon$ away from the optimal. More formally, it is defined as follows.

Definition 4.7. $(q_1, \ldots, q_n, \alpha, \beta_1, \ldots, \beta_n)$ is called an *ϵ-approximate solution* to (7) if it satisfies

$$\sum_{i=1}^{n} q_i = d, \qquad \text{(Market Clearing)}$$

$$p(q_i; \alpha, \beta_i) - c_i(q_i) + \epsilon \geq 0, \quad i = 1, \ldots, n, \qquad (\epsilon\text{-Revenue Adequacy})$$

$$p(q_i; \alpha, \beta_i) - c_i(q_i) + \epsilon \geq p(q_i'; \alpha, \beta_i) - c_i(q_i'), \quad \forall q_i' \neq q_i, \quad i = 1, \ldots, n, \qquad (\epsilon\text{-Competitive Equilibrium})$$

and

$$\sum_{i=1}^{n} p(q_i; \alpha, \beta_i) \leq p^* + n\epsilon. \qquad (\epsilon\text{-Economic Efficiency})$$

Given this notion of an approximate solution, we can move towards designing the algorithm. The optimization problem (7) looks highly coupled, at first, since the constraints share a lot of common variables. However, one can see that, for a fixed value of α, the objective becomes additively

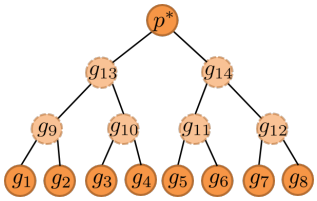


Fig. 2. An example of the binary tree defined by Algorithm 1 for $n = 8$. The faded circles correspond to the added dummy nodes.

separable in (q_i, β_i). Furthermore (again for fixed α), constraints (7c),(7d) involve only the i-th variables (q_i, β_i) for each i. Although the Market Clearing condition still couples the variables together, this observation allows us to reformulate (7) as

$$p^* = \min_{\substack{q_1,\ldots,q_n \\ \alpha \in \mathcal{A}}} \quad \sum_{i=1}^{n} g_i(q_i; \alpha) \tag{11a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d, \tag{11b}$$

where

$$g_i(q; \alpha) = \min_{\beta_i \in \mathcal{B}} \quad p(q; \alpha, \beta_i) \tag{12a}$$

$$\text{s.t.} \quad p(q; \alpha, \beta_i) - c_i(q) \geq 0, \tag{12b}$$

$$p(q; \alpha, \beta_i) - c_i(q) \geq p(q'; \alpha, \beta_i) - c_i(q'), \quad \forall q' \neq q, \tag{12c}$$

for all $i = 1, \ldots, n$.

Therefore, for any fixed value of α and q_i, the optimization over β_i can be done individually, as in (12). What remains to address, however, is the coupling of the variables as a result of the Market Clearing constraint. One naive approach would be to simply try all possible choices of $(q_1, \ldots, q_n)$, and pick the one that has the minimum objective value. This is very inefficient. Instead, we take a dynamic programming approach, and group pairs of variables together, defining a new variable as their *parent*. We then group the parents together, and continue this process until we reach the *root*, i.e., where there is only one node. During this procedure, at each new node i, we need to solve the following (small) problem

$$\begin{aligned} g_i(q; \alpha) = \min_{q_j, q_k} \quad & g_j(q_j; \alpha) + g_k(q_k; \alpha) \\ \text{s.t.} \quad & q_j + q_k = q, \end{aligned} \tag{13}$$

for every q, where j and k are the *children* of i. At the root of the tree we will be able to compute $g_{\text{root}}(d; \alpha)$. Figure 2 shows an example of the created binary tree for this procedure for $n = 8$. This procedure can be repeated for different values of α, and the optimal value p^* can be computed as $\min_\alpha g_{\text{root}}(d; \alpha)$.

The problem with recursion (13) is that it requires an infinite-dimensional computation at every step, since the values of $g_i(q; \alpha)$ need to be computed for every q. To get around this issue, we note that the variables q_i live in the bounded set $[0, d]$, and hence can be discretized to lie in a finite set Q, such that every possible q_i is at most $\delta(\epsilon)$ away from some point in Q. Similarly, if the α and β_i's are continuous variables, we can discretize the bounded sets $\mathcal{A}$ and $\mathcal{B}$ into some finite sets $\mathcal{A}'$ and $\mathcal{B}'$, such that every point in $\mathcal{A}$ (or $\mathcal{B}$) is at most $\delta(\epsilon)$ away, in infinity-norm sense, from some point in $\mathcal{A}'$ (or $\mathcal{B}'$). See the appendix for details.

For finding an ϵ-approximate solution, (12) is relaxed to

$$g_i(q;\alpha) = \min_{\beta_i \in \mathcal{B}'} \quad p(q;\alpha,\beta_i) \tag{14a}$$

$$\text{s.t.} \quad p(q;\alpha,\beta_i) - c_i(q) + \epsilon \geq 0, \tag{14b}$$

$$p(q;\alpha,\beta_i) - c_i(q) + \epsilon \geq p(q';\alpha,\beta_i) - c_i(q'), \quad \forall q' \neq q, \tag{14c}$$

for all $i = 1, \ldots, n$, and (13) remains the same, except the variables (q_j, q_k) take values in Q, i.e.

$$\begin{aligned} g_i(q;\alpha) = \min_{q_j, q_k \in Q} \quad & g_j(q_j;\alpha) + g_k(q_k;\alpha) \\ \text{s.t.} \quad & q_j + q_k = q, \end{aligned} \tag{15}$$

for all $i > n$. We denote the optimizer of (14) by $b_i(q;\alpha)$, and the optimizer of (15), which is a pair of quantities (q_j, q_k), by $x_i(q;\alpha)$. The full procedure is summarized in psuedocode in Algorithm 1 in the appendix.

While not immediately clear, the proposed approximation algorithm can be shown to run in time that is polynomial in both n and $1/\epsilon$ (in fact, linear in n). Further, the solution it provides is ϵ-accurate under a mild smoothness assumption on the cost and price functions, which holds true for almost any function considered in the literature. These two results are summarized in the following theorem, which is proven in the appendix.

Theorem 4.8. *Consider $c_i(.)$ and $p(.;.)$ that have at most a finite number of discontinuities and are Lipschitz on each continuous piece of their domain. Algorithm 1 finds an ϵ-approximate solution to the optimal pricing problem* (7) *with running time $O\left(n(1/\epsilon)^{l_1+l_2+2}\right)$, where n is the number of suppliers, and l_1 and l_2 are the number of shared and individual parameters in the price, respectively.*

It is worth emphasizing that l_1 and l_2 do not scale with n, and are typically very small constants. For example, for the so-called linear-plus-uplift price functions $l_1 = l_2 = 1$. Therefore, the algorithm is very efficient.

We should also remark that if one requires the total payment in Definition 4.7 to be at most ϵ (rather than $n\epsilon$) away from the optimal p^*, the running time of our algorithm will still be polynomial in both n and $1/\epsilon$, i.e., $O\left(n^3(\frac{1}{\epsilon})^{l_1+l_2+2}\right)$. See the appendix for details.

5 EQUILIBRIUM-CONSTRAINED PRICING FOR NETWORKED MARKETS

We now consider the more general problem of finding an efficient pricing scheme in a networked market. The networked market we consider has n suppliers, located at the nodes (vertices) $V = \{1, \ldots, n\}$ of a network, and connected through lines (edges) E, where, without loss of generality, the edges are defined to be from the smaller node to the larger node (i.e. $\forall (i,j) \in E,\ i < j$). The i-th supplier has a cost function $c_i(q_i)$ for producing quantity q_i, which may be non-convex, as before, and there is an inelastic demand d_i at each node i. The lines connecting the nodes can possibly have certain capacities for the flows they can carry. We denote the flow of any line $e = (i,j)$, from i to j, by f_e, and its limits (capacity) by $\underline{f_e}$ and $\overline{f_e}$ (the flow from j to i is $-f_e$).

Note that if there are multiple suppliers co-located in a market, we can simply assign them each their own vertex, and connect them through paths with infinite capacities. In other words, a node with multiple suppliers can be simply replaced with a "line graph" composed of those suppliers and infinite-capacity edges.

5.1 Pricing Formulation

A key benefit of EC pricing is the ease of generalization to the networked setting. There are no current pricing rules that can be readily applied to the networked case. Assuming a parametric form $p_i(q_i) := p(q_i;\alpha,\beta_i)$ for $\mathcal{P}$, with shared parameters α and individual parameters β_i as before, the Equilibrium-Constrained pricing can be formulated as the following optimization problem.

Parameterized Networked Equilibrium-Constrained (EC) Pricing:

$$p^* = \min_{\substack{q_1,\ldots,q_n \\ \{f_e\}_{e\in E} \\ \alpha\in\mathcal{A} \\ \beta_1,\ldots,\beta_n\in\mathcal{B}}} \quad \sum_{i=1}^{n} p(q_i;\alpha,\beta_i) \tag{16a}$$

$$\text{s.t.} \quad q_i - d_i = \sum_{\substack{j \\ (i,j)\in E}} f_{(i,j)} - \sum_{\substack{j \\ (j,i)\in E}} f_{(j,i)}, \quad i = 1,\ldots,n \tag{16b}$$

$$\underline{f_e} \le f_e \le \overline{f_e}, \quad e \in E \tag{16c}$$

$$p(q_i;\alpha,\beta_i) - c_i(q_i) \ge 0, \quad i = 1,\ldots,n \tag{16d}$$

$$p(q_i;\alpha,\beta_i) - c_i(q_i) \ge \max_{q_i'\neq q_i} p(q_i';\alpha,\beta_i) - c_i(q_i'), \quad i = 1,\ldots,n \tag{16e}$$

The objective is the total payment, as discussed before, and the optimization is over quantities q_i, line flows f_e, and price functions $p_i \in \mathcal{P}$. Constraint (16b) is the Market Clearing condition (or Flow Conservation) for each individual node, i.e., the net production at each node should be equal to its outgoing flow. Constraint (16c) enforces the line limits (Capacity Constraints). Constraints (16d) and (16e) are Revenue Adequacy and Competitive Equilibrium, respectively, as before. The key difference between the networked setting and the single-market one is that here the Market Clearing condition is spread across the network, and we have to solve the problem for the flows as well.

Remark 5.1. *When the capacity constraints* (16c) *are relaxed* ($\underline{f_e} = -\infty, \overline{f_e} = \infty,\ \forall e \in E$)*, the networked problem reduces to the single-market one. In this case, the solution to the optimization problem* (16) *reduces to that of* (7)*. That is because the only constraint involving the flows would be* (16b)*, and we can always finds flows that satisfy it, as long as* $\sum_{i=1}^{n} q_i - \sum_{i=1}^{n} d_i = 0$*, which is the conventional Market Clearing condition.*

5.2 An Efficient Approximation Algorithm

For certain classes of non-convexities, the optimization problem (16) can still be solved using off-the-shelf solvers, similar to those used in the other methods for the no-network case. However, those algorithms cannot handle more general classes of non-convexities. In this section, we develop a computationally efficient approximation algorithm for general non-convex costs, for a special class of networks.

A special yet important class of networks are *acyclic* networks, which are a typical topology in many markets, including *electricity distribution networks*. Acyclic networks have a *tree* topology (they do not have cycles), which allows us to devise an efficient algorithm for them. In the remainder of this section, we limit our attention to these networks. The main ideas extend directly to more general networks, as long as there are not "too many cycles" in the network in some sense (i.e. bounded tree-width networks). We have focused on the acyclic case due to space constraints.

Without loss of generality, let us denote the first node as the *root* of the tree, and nodes with only one neighbor as the *leaves*. Every node (except the root) has a unique *parent*, defined as the first node on the unique path connecting it to the root node. The set of nodes that have a given node i as their parent is said to be node i's *children*. It can be shown that any tree with arbitrary degree can be transformed into a *binary tree*, i.e., a tree where each node has a unique parent and at most 2 children, with $O(n)$ nodes (See the appendix). Thus, we can focus on binary trees.

For a node i, let $\mathrm{ch}_1(i), \mathrm{ch}_2(i)$ denote its children ($\mathrm{ch}_1(i) = \emptyset$ and/or $\mathrm{ch}_2(i) = \emptyset$ when i has less than two children). The problem can then be written as

$$p^* = \min_{\substack{q_1,\ldots,q_n \\ f_1,\ldots,f_n \\ \alpha\in\mathcal{A} \\ \beta_1,\ldots,\beta_n\in\mathcal{B}}} \quad \sum_{i=1}^{n} p(q_i;\alpha,\beta_i) \tag{17a}$$

$$\text{s.t.} \quad q_i - d_i = f_{\mathrm{ch}_1(i)} + f_{\mathrm{ch}_2(i)} - f_i, \quad i = 1,\ldots,n \tag{17b}$$

$$\underline{f_i} \le f_i \le \overline{f_i}, \quad i = 1,\ldots,n \tag{17c}$$

$$p(q_i;\alpha,\beta_i) - c_i(q_i) \ge 0, \quad i = 1,\ldots,n \tag{17d}$$

$$p(q_i;\alpha,\beta_i) - c_i(q_i) \ge \max_{q_i'\neq q_i} \; p(q_i';\alpha,\beta_i) - c_i(q_i'), \quad i = 1,\ldots,n \tag{17e}$$

where f_i represents the incoming flow to each node i from its parent, and $\underline{f_{\mathrm{root}}} = \overline{f_{\mathrm{root}}} = 0$.

Similarly as in the single-market case, we define an ϵ-approximate solution to this problem.

Definition 5.2. $(q_1,\ldots,q_n,f_1,\ldots,f_n,\alpha,\beta_1,\ldots,\beta_n)$ is called an *ϵ-approximate solution* to (17) if it satisfies

$$|q_i - d_i - f_{\mathrm{ch}_1(i)} - f_{\mathrm{ch}_2(i)} + f_i| \le \epsilon, \quad i = 1,\ldots,n, \qquad (\epsilon\text{-Load Balancing})$$

$$\underline{f_i} \le f_i \le \overline{f_i}, \quad i = 1,\ldots,n, \qquad (\text{Flow Limit})$$

$$p(q_i;\alpha,\beta_i) - c_i(q_i) + \epsilon \ge 0, \quad i = 1,\ldots,n, \qquad (\epsilon\text{-Revenue Adequacy})$$

$$p(q_i;\alpha,\beta_i) - c_i(q_i) + \epsilon \ge p(q_i';\alpha,\beta_i) - c_i(q_i'), \quad \forall q_i' \neq q_i, \quad i = 1,\ldots,n, \qquad (\epsilon\text{-Competitive Equilibrium})$$

$$\sum_{i=1}^{n} p(q_i;\alpha,\beta_i) \le p^* + n\epsilon. \qquad (\epsilon\text{-Economic Efficiency})$$

The main difference from the definition in the single-market case is that the Market Clearing condition has been replaced with ϵ-Load Balancing and exact Flow Limit conditions here.

Note that the minimization over the variables β_i in problem (17) can be done "internally," and the problem can be re-expressed as

$$p^* = \min_{\substack{q_1,\ldots,q_n \\ f_1,\ldots,f_n \\ \alpha\in A}} \quad \sum_{i=1}^{n} g_i(q_i;\alpha) \tag{18a}$$

$$\text{s.t.} \quad q_i - d_i = f_{\mathrm{ch}_1(i)} + f_{\mathrm{ch}_2(i)} - f_i, \quad i = 1,\ldots,n \tag{18b}$$

$$\underline{f_i} \le f_i \le \overline{f_i}, \quad i = 1,\ldots,n \tag{18c}$$

where

$$g_i(q;\alpha) = \min_{\beta_i\in\mathcal{B}} \quad p(q;\alpha,\beta_i) \tag{19a}$$

$$\text{s.t.} \quad p(q;\alpha,\beta_i) - c_i(q) \ge 0, \tag{19b}$$

$$p(q;\alpha,\beta_i) - c_i(q) \ge p(q';\alpha,\beta_i) - c_i(q'), \quad \forall q' \neq q, \tag{19c}$$

for all $i = 1,\ldots,n$.

The key insight is that the tree structure of the constraints (18b), allows us to write the optimization problem in a recursive form as follows.

$$p^* = \min_{\alpha} \; h_{\mathrm{root}}(0;\alpha) \tag{20}$$

where

$$h_i(f_i;\alpha) = \min_{q_i, f_{\mathrm{ch}_1(i)}, f_{\mathrm{ch}_2(i)}} \quad g_i(q_i;\alpha) + h_{\mathrm{ch}_1(i)}(f_{\mathrm{ch}_1(i)};\alpha) + h_{\mathrm{ch}_2(i)}(f_{\mathrm{ch}_2(i)};\alpha) \tag{21a}$$

$$\text{s.t.} \quad q_i - d_i = f_{\mathrm{ch}_1(i)} + f_{\mathrm{ch}_2(i)} - f_i \tag{21b}$$

$$\underline{f_{\mathrm{ch}_1(i)}} \le f_{\mathrm{ch}_1(i)} \le \overline{f_{\mathrm{ch}_1(i)}} \tag{21c}$$

$$\underline{f_{\mathrm{ch}_2(i)}} \le f_{\mathrm{ch}_2(i)} \le \overline{f_{\mathrm{ch}_2(i)}} \tag{21d}$$

for all $i = 1, \dots, n$.

Now, this recursive form is amenable to dynamic programming. However, since the variables are continuous, each step still requires an infinite-dimensional search. In order to tackle this issue, we can discretize the variables and solve the following approximate versions.

$$h_i(f_i;\alpha) = \min_{\substack{q_i \in Q_i \\ f_{\mathrm{ch}_1(i)} \in F_{\mathrm{ch}_1(i)} \\ f_{\mathrm{ch}_2(i)} \in F_{\mathrm{ch}_2(i)}}} \quad g_i(q_i;\alpha) + h_{\mathrm{ch}_1(i)}(f_{\mathrm{ch}_1(i)};\alpha) + h_{\mathrm{ch}_2(i)}(f_{\mathrm{ch}_2(i)};\alpha) \tag{22a}$$

$$\text{s.t.} \quad |q_i - d_i - f_{\mathrm{ch}_1(i)} - f_{\mathrm{ch}_2(i)} + f_i| \le \epsilon \tag{22b}$$

for all $i = 1, \dots, n$, where $Q_1, \dots, Q_n$ and $F_1, \dots, F_n$ are properly-defined discrete sets (See the appendix for details). We denote the optimizer (triple) of (22) by $y_i(f_i;\alpha)$.

$$g_i(q;\alpha) = \min_{\beta_i \in \mathcal{B}'} \quad p(q;\alpha,\beta_i) \tag{23a}$$

$$\text{s.t.} \quad p(q;\alpha,\beta_i) - c_i(q) + \epsilon \ge 0, \tag{23b}$$

$$p(q;\alpha,\beta_i) - c_i(q) + \epsilon \ge p(q';\alpha,\beta_i) - c_i(q'), \quad \forall q' \ne q, \tag{23c}$$

for all $i = 1, \dots, n$. The optimizer of (23) is denoted by $b_i(q;\alpha)$.

The steps of the procedure are summarized in psuedocode in Algorithm 2 in the appendix, and the following result summarizes the theoretical guarantee of the algorithm.

THEOREM 5.3. *Consider $c_i(.)$ and $p(.;.)$ that have at most a finite number of discontinuities and are Lipschitz on each continuous piece of their domain. Algorithm 2 finds an ϵ-approximate solution to the optimal networked pricing problem* (17)*, with running time $O\left(n(1/\epsilon)^{l_1+\max\{l_2,1\}+2}\right)$, where n is the number of suppliers, and l_1 and l_2 are the number of shared and individual parameters in the price, respectively.*

It is worth mentioning that the network algorithm developed in this section suggests another way of solving the no-network case as well, by replacing the single market with a line graph with infinite capacities. This algorithm will in turn have time complexity $O\left(n(\frac{1}{\epsilon})^{l_1+l_2+2}\right)$, which is the same as that of the one developed in Section 4.2.

6 CONCLUDING REMARKS

We study the problem of pricing in single and networked markets with non-convex costs. Our key contribution is the proposal of a novel scheme, Equilibrium-Constrained (EC) pricing, which optimizes for the allocations and the price parameters at the same time, while imposing the equilibrium conditions as constraints. Our pricing framework is general in the sense that: (i) it can be used for pricing general non-convex cost functions, (ii) it allows for using general price classes, (iii) can be computed in polynomial-time regardless of the source of the non-convexities, and (iv) it extends easily to networked markets.

This paper opens up a variety of important directions for future work. First, as this framework enables one to use general price classes, it would be interesting to apply it to specific classes of price

functions (e.g. quadratic plus uplift, piece-wise, etc.) and characterize the solution theoretically and/or numerically. One can then investigate the potential trade-offs between the complexity of the class and the economic efficiency or the uniformity of the price. Second, since electricity markets are an important application of the pricing problem studied here, it would be interesting to evaluate the proposed scheme in practical settings for electricity markets. Our preliminary exploration (Section B of the appendix) shows that we can achieve more efficient (lower total payments) and less discriminatory (lower uplifts) prices with, for instance, piece-wise linear functions. More evaluations in large-scale, practical settings should be carried out in order to evaluate the potential of deployment. Another important direction to pursue is the extension of our results to networked markets with more general network structures. Our algorithm currently applies to networks with bounded tree-width; however beyond such networks new ideas are needed. Finally, our proposed pricing scheme has broader implications for non-convex optimization problems as well. In the convex setting, dual prices are crucial for the development of distributed optimization algorithms, but such approaches have not been possible in non-convex settings due to the lack of pricing rules with the desirable properties laid out in Section 2.2. It is now possible to explore whether EC prices can be used as the basis for distributed optimization algorithms in the non-convex setting.

REFERENCES

Veronica Araoz and Kurt Jörnsten. 2011. Semi-Lagrangean approach for price discovery in markets with non-convexities. *European Journal of Operational Research* 214, 2 (2011), 411–417.

Navid Azizan Ruhi, Krishnamurthy Dvijotham, Niangjun Chen, and Adam Wierman. 2017. Opportunities for price manipulation by aggregators in electricity markets. *IEEE Transactions on Smart Grid* (2017).

Paulina Beato and Andreu Mas-Colell. 1985. On marginal cost pricing with given tax-subsidy rules. *Journal of Economic Theory* 37, 2 (1985), 356–365.

Mette Bjørndal and Kurt Jörnsten. 2008. Equilibrium prices supported by dual price functions in markets with non-convexities. *European Journal of Operational Research* 190, 3 (2008), 768–789.

Mette Bjørndal and Kurt Jörnsten. 2010. A Partitioning Method that Generates Interpretable Prices for Integer Programming Problems. *Handbook of Power Systems II* (2010), 337–350.

Donald J Brown. 1991. Equilibrium analysis with non-convex technologies. *Handbook of mathematical economics* 4 (1991), 1963–1995.

Francisco D Galiana, Alexis L Motto, and François Bouffard. 2003. Reconciling social welfare, agent profits, and consumer payments in electricity pools. *IEEE Transactions on Power Systems* 18, 2 (2003), 452–459.

Paul R Gribik, William W Hogan, and Susan L Pope. 2007. Market-clearing electricity prices and energy uplift. (2007).

Roger Guesnerie. 1975. Pareto optimality in non-convex economies. *Econometrica: Journal of the Econometric Society* (1975), 1–29.

William W Hogan and Brendan J Ring. 2003. On minimum-uplift pricing for electricity markets. *Electricity Policy Group* (2003).

Bowen Hua and Ross Baldick. 2017. A convex primal formulation for convex hull pricing. *IEEE Transactions on Power Systems* 32, 5 (2017), 3814–3823.

George Liberopoulos and Panagiotis Andrianesis. 2016. Critical review of pricing schemes in markets with non-convex costs. *Operations Research* 64, 1 (2016), 17–31.

Alexis L Motto and Francisco D Galiana. 2002. Equilibrium of auction markets with unit commitment: The need for augmented pricing. *IEEE Transactions on Power Systems* 17, 3 (2002), 798–805.

Richard P O'Neill, Paul M Sotkiewicz, Benjamin F Hobbs, Michael H Rothkopf, and William R Stewart. 2005. Efficient market-clearing prices in markets with nonconvexities. *European journal of operational research* 164, 1 (2005), 269–285.

Carlos Ruiz, Antonio J Conejo, and Steven A Gabriel. 2012. Pricing non-convexities in an electricity pool. *IEEE Transactions on Power Systems* 27, 3 (2012), 1334–1342.

Herbert E Scarf. 1990. Mathematical programming and economic theory. *Operations Research* 38, 3 (1990), 377–385.

Herbert E Scarf. 1994. The allocation of resources in the presence of indivisibilities. *The Journal of Economic Perspectives* 8, 4 (1994), 111–128.

Dane A Schiro, Tongxin Zheng, Feng Zhao, and Eugene Litvinov. 2016. Convex Hull Pricing in Electricity Markets: Formulation, Analysis, and Implementation Challenges. *IEEE Transactions on Power Systems* 31, 5 (2016), 4068–4075.

Ralph Turvey. 1969. Marginal cost. *The Economic Journal* 79, 314 (1969), 282–299.

Laurence A Wolsey. 1981. Integer programming duality: Price functions and sensitivity analysis. *Mathematical Programming* 20, 1 (1981), 173–195.

A PSEUDOCODE

Algorithm 1 Find an ϵ-approximate solution to the optimal pricing problem (7)

Input: $n, c_1(.), \ldots, c_n(.), p(.;.), \epsilon$
for α in $\mathcal{A}'$ **do**
  $S = 1 : n$
  for i in S **do** ▷ for the leaves
    compute $g_i(q;\alpha)$ **for all** q in Q, using (14)
  end for
  while $|S| > 2$ **do** ▷ while not reached the root
    $S_{\text{new}} = S(\text{end}) + 1 : S(\text{end}) + \lceil \frac{|S|}{2} \rceil$
    for i in S_{new} **do** ▷ for the intermediate nodes
      $[j,k]$ = indices of children of i
      if $k = \emptyset$ **then** $g_i(.;\alpha) = g_j(.;\alpha)$
      else, compute $g_i(q;\alpha)$ **for all** q in Q, using (15) ▷ it has two children
      end if
    end for
    $S = S_{\text{new}}$
  end while
  $[j,k] = S$
  compute $g_{\text{root}}(d;\alpha)$, using (15) ▷ at the root
end for
$\alpha^* = \underset{\alpha \in \mathcal{A}'}{\arg\min} \quad g_{\text{root}}(d;\alpha)$
$q^*_{\text{root}} = d$
for $i = \text{root} : -1 : n+1$ **do**
  $[q^*_j, q^*_k] = x_i(q^*_i;\alpha^*)$, where $[j,k]$ = indices of children of i
end for
for $i = n : -1 : 1$ **do**
  $\beta^*_i = b_i(q^*_i;\alpha^*)$
end for
return $(q^*_1, \ldots, q^*_n, \alpha^*, \beta^*_1, \ldots, \beta^*_n)$

Algorithm 2 Find an ϵ-approximate solution to the optimal networked pricing problem (17)

Input: G=(V,E), $c_1(.), \ldots, c_n(.), p(.;.), \epsilon$
for α in $\mathcal{A}'$ **do**
  for all nodes i **do**
    compute $g_i(q_i;\alpha)$ **for all** q_i in Q_i, using (23)
  end for
  for all nodes $i \neq$root (in bottom-up order) **do**
    compute $h_i(f;\alpha)$ **for all** f in F_i, using (22)
  end for
  compute $h_{\text{root}}(0;\alpha)$, using (22)
end for
$\alpha^* = \underset{\alpha\in\mathcal{A}'}{\arg\min} \quad h_{\text{root}}(0;\alpha)$
$f^*_{\text{root}} = 0$
for all nodes i (in top-down order) **do**
  $[q_i^*, f^*_{\text{ch}_1(i)}, f^*_{\text{ch}_2(i)}] = y_i(f_i^*;\alpha^*)$
  $\beta_i^* = b_i(q_i^*;\alpha^*)$
end for
return $(q_1^*, \ldots, q_n^*, f_1^*, \ldots, f_n^*, \alpha^*, \beta_1^*, \ldots, \beta_n^*)$

B EXPERIMENTAL RESULTS

To illustrate the contrasts between EC Pricing and the more traditional approaches, we look at the prices derived in some simple numerical examples in this section. Specifically, we compare the payments and uplifts generated from different pricing schemes, including IP, CH, SLR, PD and EC. Among all these schemes, only EC allows flexibility of payment forms. As a result, we further divide EC into one with a payment function in the form of linear marginal price plus uplifts and another Pricing with a payment form of piecewise linear marginal prices plus uplifts. In practice, specific limits on number of sections and maximum slope among all sections can be used to further restrict EC. For convenience, we name these variations of EC in terms of number of piecewise sections of its payment form, i.e. EC2 refers to EC with a payment function in the form of 2 piecewise sections plus uplifts.

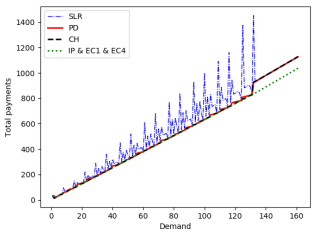
First, we apply all these pricing schemes to a *single* market example from [Hogan and Ring, 2003], which is a modification of Scarf's example developed in [Scarf, 1994]. Second, we adapt cost functions in the modified Scarf's example to be quadratic plus startup cost in order to further explore how these schemes generalize to different cost functions. Finally, we consider a further generalization to a simple 2-node networked market.

B.1 Case 1: Linear plus startup cost

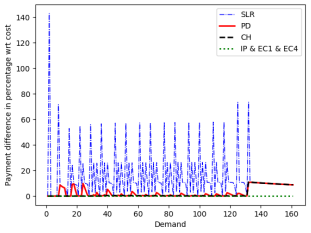
We consider a modified Scarf's example, as proposed in [Hogan and Ring, 2003]. The parameters are listed in Table 2. We assume that demand is inelastic with a maximum capacity of 161 units. We restrict the payment function of EC4 to have four sections and impose that marginal price of any section cannot exceed the maximum marginal price for any supplier operating at full capacity. Figure 3a shows total payments for different demand levels while Figure 3c shows the corresponding uplifts of the pricing schemes that apply, i.e. CH, EC1 and EC4. Payments of two revenue-adequate pricing schemes, including SLR and PD, are higher than total costs in general. IP, EC1 and EC4 achieve the minimum payments equal to total costs. CH achieves the minimum payments at low demand levels and its total payments surpass total costs as demand gets high. As for uplifts, EC4 achieves the smallest among the three pricing schemes. Total uplifts of CH and EC1 are close to

Table 2. Summary of the production characteristics in the modified Scarf's example.

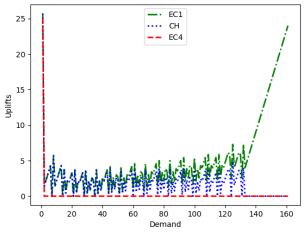
Type	Smokestack	High Tech	Med Tech
Capacity	16	7	6
Minimum output	0	0	2
Startup cost	53	30	0
Marginal cost	3	2	7
Quantity	6	5	5



(a) Total payments as a function of demand

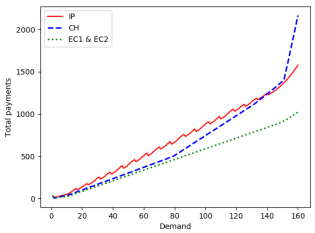


(b) Payment difference in percentage w.r.t cost as a function of demand

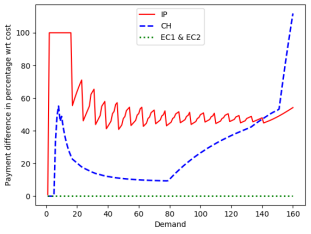


(c) Total uplifts as a function of demand

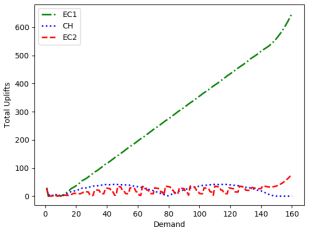
Fig. 3. An example with cost functions of the form of linear plus startup cost



(a) Total payments as a function of demand



(b) Payment difference in percentage w.r.t cost as a function of demand



(c) Total uplifts as a function of demand

Fig. 4. An example with cost functions of the form of quadratic plus startup cost

each other at a low demand level and that of EC1 increases significantly when demand approaches capacity. This is not surprising as total payments of CH go above total costs at a high demand, making it possible for relatively smaller total uplifts. It is worth noting that startup prices and marginal prices for IP are volatile and unstable. Overall, EC4 outperforms other pricing schemes in terms of total payments and total uplifts.

B.2 Case 2: Quadratic plus startup cost

To further explore how these pricing schemes generalize to different cost functions, we modify the cost functions of the example above. Table 3 describes the new cost functions for each supplier. Since it is not clear how to generalize SLR and PD, we focus on a comparison among IP, CH, EC1

Table 3. Summary of the new cost functions in the modified Scarf's example.

Type	Smokestack	High Tech	Med Tech
Cost function	$\frac{3}{16}q^2 + 53 * \mathbb{1}\{q > 0\}$	$\frac{2}{7}q^2 + 30 * \mathbb{1}\{q > 0\}$	$\frac{7}{6}q^2$

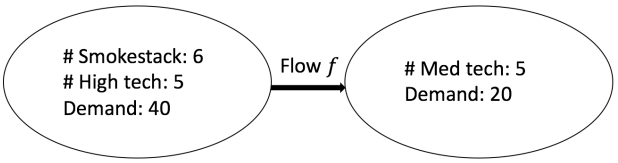
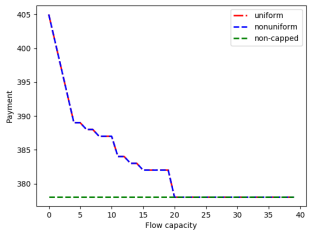


Fig. 5. A schematic drawing for two connected markets with a constraint on flow capacity

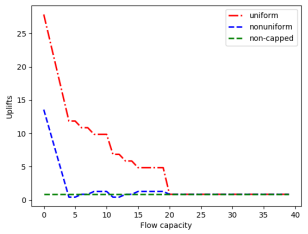
and EC2. We restrict the payment function of EC2 to have two sections with the marginal price of any section bounded by the maximum of marginal price for any supplier operating at full capacity. As can be seen in Figure 4a, EC1 and EC2 achieve the possible minimum total payments equal to total costs. Total payments of IP and CH are both above total costs and the gap between total payments and costs grows as demand increases. Observe that the demand here ranges from 1 to 160 because marginal price of CH increases dramatically at the capacity level and the plot over the interval $(1, 160)$ would be a flat line if the whole range were covered. Figure 4c shows that total uplifts of EC1 are much larger than that of CH and EC2. At a low demand level, uplifts of EC1 and EC2 are close to each other. As demand increases, uplifts of EC2 are a little larger than those of CH, in order to maintain a smaller overall payment. There is a trade-off between minimizing total payments and minimizing total costs. Allowing the flexibility of payment function form enables EC2 to perform better than either CH or EC1 in terms of total payments and uplifts.

B.3 A Networked Market with Capacity Constraints

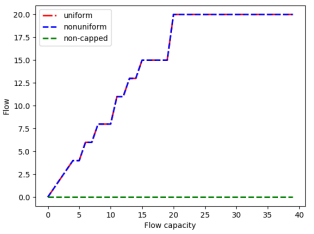
One advantage EC has over all the other pricing schemes is its generality. Specifically, EC can be applied to networked markets. In this section, we divide a single market with a fixed total demand 60 as described earlier into one market with only med tech suppliers and the other one with the smokestack and high tech suppliers. The cost functions of the suppliers are the same as defined earlier, i.e. linear plus startup cost. As pictured in Figure 5, these two markets are connected via a flow capacity constraint. We consider two different cases of non-uniform marginal pricing and uniform marginal pricing for these two markets. Figure 6 shows how total payments, total uplifts and flow between these two connected markets vary as flow capacity increases for nonuniform and uniform marginal pricing settings. The results show that the total payments and total uplifts decrease as more flow is allowed between these two markets until it reaches the demand of one market, which means one market alone meets the total demand. Allowing non-uniform pricing does not further reduce total payments as total payments are minimal and equal the total costs. However, it helps reduce total uplifts, as we can see in Figure 6b.



(a) Total payments as a function of flow capacity



(b) Total uplifts as a function of flow capacity



(c) Flow between two markets as a function of flow capacity

Fig. 6. An example of two connected markets with a constraint on the flow capacity

C SUPPLEMENT TO SECTION 4.1

In this section, we formally prove the reduction of the optimization problem for the class of linear-plus-uplift functions to (8), and then show Propositions 4.5 and 4.6.

C.1 Reduction

Here we show that for the class of linear-plus-uplift price functions $p(q_i; \lambda, u_i, \hat{q}_i) = \lambda q_i + u_i \mathbb{1}_{q_i=\hat{q}_i}$, one can assume $\hat{q}_i^* = q_i^*$ without loss of generality, and therefore the optimization problem (7) reduces to (8) for this class. The optimization problem (7) for price function $p(q_i; \lambda, u_i, \hat{q}_i) = \lambda q_i + u_i \mathbb{1}_{q_i=\hat{q}_i}$, $\lambda, u_1, \dots, u_n \geq 0$, is as follows

$$p^*_{\text{uplift}} = \min_{\substack{q_1,\dots,q_n \\ \lambda \geq 0 \\ u_1,\dots,u_n \geq 0 \\ \hat{q}_1,\dots,\hat{q}_n}} \quad \sum_{i=1}^{n} (\lambda q_i + u_i \mathbb{1}_{q_i=\hat{q}_i}) \tag{24a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d \tag{24b}$$

$$\lambda q_i + u_i \mathbb{1}_{q_i=\hat{q}_i} - c_i(q_i) \geq 0, \quad i = 1, \dots, n \tag{24c}$$

$$\lambda q_i + u_i \mathbb{1}_{q_i=\hat{q}_i} - c_i(q_i) \geq \max_{q_i' \neq q_i} \lambda q_i' + u_i \mathbb{1}_{q_i'=\hat{q}_i} - c_i(q_i'), \quad i = 1, \dots, n \tag{24d}$$

The following lemma shows that this optimization problem can be reduced to (8), and the optimal uplifts of (8) are no larger than those of (24).

Lemma C.1. *Given any solution* $(\boldsymbol{q}^*, \lambda^*, \boldsymbol{u}^*, \hat{\boldsymbol{q}}^*)$ *to the optimization problem* (24), $(\boldsymbol{q}^*, \lambda^*, \underline{\boldsymbol{u}}, \boldsymbol{q}^*)$ *is also a solution, where*

$$\underline{u}_i = \begin{cases} u_i^*, & \text{if } \hat{q}_i^* = q_i^* \\ 0, & \text{o.w.} \end{cases}.$$

Proof of Lemma C.1. Let us first show the feasibility of $(\boldsymbol{q}^*, \lambda^*, \underline{\boldsymbol{u}}, \boldsymbol{q}^*)$. For any i such that $\hat{q}_i^* \neq q_i^*$, we have that

$$\lambda^* q_i^* - c_i(q_i^*) \geq 0$$
$$\lambda^* q_i^* - c_i(q_i^*) \geq \max_{q_i' \neq q_i^*} \lambda^* q_i' + u_i^* \mathbb{1}_{q_i'=\hat{q}_i^*} - c_i(q_i') \geq \max_{q_i' \neq q_i^*} \lambda^* q_i' - c_i(q_i'),$$

which implies

$$\lambda^* q_i^* + \underline{u}_i^* \mathbb{1}_{q_i^* = q_i^*} - c_i(q_i^*) \geq 0$$
$$\lambda^* q_i^* + \underline{u}_i^* \mathbb{1}_{q_i^* = q_i^*} - c_i(q_i^*) \geq \max_{q_i' \neq q_i^*} \lambda^* q_i' + \underline{u}_i^* \mathbb{1}_{q_i' = \hat{q}_i^*} - c_i(q_i'),$$

because $\underline{u}_i^* = 0$. Therefore $(\boldsymbol{q^*}, \lambda^*, \underline{\boldsymbol{u}}, \boldsymbol{q^*})$ is feasible.

The objective value of $(\boldsymbol{q^*}, \lambda^*, \underline{\boldsymbol{u}}, \boldsymbol{q^*})$ is

$$\sum_{i=1}^n (\lambda^* q_i^* + \underline{u}_i) = \sum_{i:\hat{q}_i^* = q_i^*} (\lambda^* q_i^* + u_i^*) + \sum_{i:\hat{q}_i^* \neq q_i^*} \lambda^* q_i^*$$
$$= \sum_{i=1}^n (\lambda^* q_i^* + u_i^* \mathbb{1}_{q_i^* = \hat{q}_i^*}),$$

which is the same as that of $(\boldsymbol{q^*}, \lambda^*, \boldsymbol{u^*}, \boldsymbol{\hat{q}^*})$, and is therefore optimal. □

Based on this lemma, the optimization problem (24) can be reduced to (8).

C.2 Closed-Form Solutions

Proof of Proposition 4.5. In the optimization problem (8), the order of variables in the minimizations does not matter, and further, for every fixed $q_1, \ldots, q_n$ and λ, the minimization over each u_i can be done separately. Therefore this program can be massaged into the following form

$$p^*_{\text{uplift}} = \min_{q_1, \ldots, q_n} \quad \left(\min_{\lambda \geq 0} \sum_{i=1}^n g_i(q_i; \lambda) \right) \tag{25a}$$
$$\text{s.t.} \quad \sum_{i=1}^n q_i = d, \tag{25b}$$

where

$$g_i(q_i; \lambda) = \min_{u_i \geq 0} \quad \lambda q_i + u_i \tag{26a}$$
$$\text{s.t.} \quad \lambda q_i + u_i - c_i(q_i) \geq 0, \tag{26b}$$
$$\lambda q_i + u_i - c_i(q_i) \geq \max_{q_i' \neq q_i} \lambda q_i' - c_i(q_i'). \tag{26c}$$

for all $i = 1, \ldots, n$. Constraints (26b) and (26c) can be expressed as

$$\lambda q_i + u_i \geq c_i(q_i),$$
$$\lambda q_i + u_i \geq c_i(q_i) + \max_{q_i' \neq q_i} \lambda q_i' - c_i(q_i').$$

It follows that

$$g_i(q_i; \lambda) = \lambda q_i + u_i^* = c_i(q_i) + \max \left[0, \max_{q_i' \neq q_i} \lambda q_i' - c_i(q_i') \right].$$

which is, of course, a function of λ and q_i. Therefore we have

$$\min_{\lambda \geq 0} \sum_{i=1}^n g_i(q_i; \lambda) = \sum_{i=1}^n c_i(q_i)$$

and the minimizers λ^* are all values λ for which $\max_{q_i' \neq q_i} \lambda q_i' - c_i(q_i') \leq 0$, which are exactly the elements of $\Lambda = \{\lambda \geq 0 \mid \lambda q \leq c_i(q), \ \forall q, \forall i\}$ (Figure 1 provides a pictorial description of these

values). Finally we have the last minimization, which is

$$\min_{q_1,\ldots,q_n} \quad \sum_{i=1}^{n} c_i(q_i) \tag{27a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d \tag{27b}$$

and therefore has $q_i^* = q_i^0 \ \forall i$ as its optimizer. We also have $u_i^* = c_i(q_i^*) - \lambda^* q_i^*, \ \forall i$. □

Proof of Proposition 4.6. The steps of the proof are exactly the same as in the previous one, except that the additional minimizer picks the λ with the smallest total uplift $\sum_{i=1}^{n} u_i(\lambda)$, which corresponds to the largest element of Λ. □

D SUPPLEMENT TO SECTION 4.2

In this section, we prove Theorem 4.8, in two parts. First, we show that there exist finite sets $Q, \mathcal{A}', \mathcal{B}'$ for which Algorithm 1 finds an ϵ-approximate solution, and we quantify the sizes of these sets as a function of ϵ. In the second part, we analyze the running time of Algorithm 1.

D.1 ϵ-Accuracy

Let us first state a simple but useful lemma.

Lemma D.1 (δ-discretization). *Given a set $C \subseteq [\underline{L_1}, \overline{L_1}] \times \cdots \times [\underline{L_k}, \overline{L_k}]$, for any $\delta > 0$, there exists a finite set C' such that*

$$\forall z \in C, \ \exists z' \in C' \text{ s.t. } \|z - z'\|_\infty \leq \delta,$$

and further C' contains at most V/δ^k points, where $V = \prod_{i=1}^{k}(\overline{L_i} - \underline{L_i})$ is a constant (the volume of the box). C' is said to be a δ-discretization of C.

Let Q, $\mathcal{A}'$ and $\mathcal{B}'$ denote some δ-discretizations of sets $[0, d]$, $\mathcal{A}$ and $\mathcal{B}$, respectively. In other words, for every $q \in [0, d]$, $\alpha \in \mathcal{A}$, and $\beta \in \mathcal{B}$, there exist $q' \in Q$, $\alpha' \in \mathcal{A}'$, and $\beta' \in \mathcal{B}'$, such that $|q - q'| \leq \delta$, $\|\alpha - \alpha'\|_\infty \leq \delta$, and $\|\beta - \beta'\|_\infty \leq \delta$. We can combine all these inequalities as

$$\|(q, \alpha, \beta) - (q', \alpha', \beta')\|_\infty \leq \delta.$$

On the other hand, given that the cost function $c_i(.)$ for each i is Lipschitz on each continuous piece of its domain, there exists a positive constant K_i such that $|c_i(q) - c_i(q')| \leq K_i |q - q'|$, which implies

$$|c_i(q) - c_i(q')| \leq K_i \delta. \tag{28}$$

Similarly, Lipschitz continuity of $p(.;.)$ implies existence of a positive constant K such that $|p(q, \alpha, \beta) - p(q', \alpha', \beta')| \leq K \|(q, \alpha, \beta) - (q', \alpha', \beta')\|_\infty$, which yields

$$|p(q, \alpha, \beta) - p(q', \alpha', \beta')| \leq K\delta. \tag{29}$$

Using Eqs. (28),(29), we can see that for any solution $q_1^*, \ldots, q_n^*, \alpha^*, \beta_1^*, \ldots, \beta_n^*$ to optimization (7), there exists a point $q_1, \ldots, q_n, \alpha, \beta_1, \ldots, \beta_n$ with $q_1, \ldots, q_n \in Q$, $\alpha \in \mathcal{A}'$ and $\beta \in \mathcal{B}'$, for which constraints (7c) and (7d) are violated at most by $(K + K_i)\delta$ and $(2K + 2K_i)\delta$, respectively, and the objective is larger than p^* at most by $nK\delta$. As a result, this point will be an ϵ-approximate solution if

$$(K + K_i)\delta \leq \epsilon \quad \forall i, \tag{30}$$

$$2(K + K_i)\delta \leq \epsilon \quad \forall i, \tag{31}$$

$$nK\delta \leq n\epsilon. \tag{32}$$

These constraints altogether enforce an upper bound on the value of δ as

$$\delta \leq C\epsilon,$$

for some constant C. Therefore if we pick

$$\delta = \frac{d}{\left\lceil \frac{d}{C\epsilon} \right\rceil}, \tag{33}$$

our algorithm is guaranteed to encounter an ϵ-approximate solution while enumerating the points, and $Q = \{0, \delta, 2\delta, \ldots, d\}$ is a valid δ-discretization for $[0, d]$, which has $N_q = \left\lceil \frac{d}{C\epsilon} \right\rceil + 1 = O\left(\frac{1}{\epsilon}\right)$ points. The nice thing about this particular choice of δ is that now d can be written as a sum of n elements in Q (because all the elements, including d, are multiples of δ), which allows us to satisfy the Market Clearing condition exactly. Based on Lemma (D.1), $\mathcal{A}'$ and $\mathcal{B}'$ contain $N_\alpha = O\left(\frac{1}{\delta^{l_1}}\right) = O\left(\frac{1}{\epsilon^{l_1}}\right)$ and $N_\beta = O\left(\frac{1}{\delta^{l_2}}\right) = O\left(\frac{1}{\epsilon^{l_2}}\right)$ points.

Finally, if there are any discontinuities in the cost or price functions, we can simply add them to our discrete sets Q, $\mathcal{A}'$ and $\mathcal{B}'$, and since there are at most a finite number of them, the sizes of the sets remain in the same order, i.e., $N_q = O\left(\frac{1}{\epsilon}\right)$, $N_\alpha = O\left(\frac{1}{\epsilon^{l_1}}\right)$ and $N_\beta = O\left(\frac{1}{\epsilon^{l_2}}\right)$. Next, we calculate the time complexity of Algorithm 1 running on these discrete sets.

D.2 Run-Time Analysis

In this section, we show that Algorithm 1 has a time complexity of $O\left(n(\frac{1}{\epsilon})^{l_1+l_2+2}\right)$. For every fixed α, we have the following computations

(1) The leaves: We need to compute $g_i(q; \alpha)$ for every i and every $q \in Q$. Computing each $g_i(q; \alpha)$ (i.e. for fixed i, q, α) takes $O(N_\beta N_q)$. The reason for that is we have to search over all $\beta_i \in B'$, and for each one there are $N_q + 1$ constraints to check. More explicitly, we need to (a) check $O(N_\beta N_q)$ constraints, (b) compute N_β objectives, and (c) find the minimum among those N_β values. All these steps together take $O(N_\beta N_q)$, and repeating for every i and q makes it $O(nN_\beta N_q^2)$.

(2) The intermediate nodes: In each new level, there are at most half as many (+1) nodes as in the previous level. For each node i in this level, we need to compute $g_i(q; \alpha)$ for every $q \in Q$. For every fixed q, there are $O(N_q)$ possible pairs of (q_j, q_k) that add up to q, and therefore we need to (a) sum $O(N_q)$ pairs of objective values, and (b) find the minimum among them, which take $O(N_q)$. Hence, the computation for each node takes $O(N_q^2)$. There are $O(\frac{n}{2} + \frac{n}{4} + \cdots + 2) = O(n)$ intermediate nodes in total, and therefore the total complexity of this part is $O(nN_q^2)$.

(3) The root: Finally at the root, we need to compute $g_{\text{root}}(d; \alpha)$. There are N_q possible pairs of (q_j, q_k) that add up to d. Therefore, we need to compute N_q sums, and find the minimum among the resulting N_q values, which takes $O(N_q)$.

Putting the pieces together, the computation for all values of α takes $N_\alpha \times \Big(O(nN_\beta N_q^2) + O(nN_q^2) + O(N_q)\Big)$, which in turn is $O(nN_\alpha N_\beta N_q^2)$. Finally, finding the minimum among the N_α values simply takes $O(N_\alpha)$.

The backward procedure, which finds the quantities q_i and the parameters β_i, takes just $O(n)$, since it is just a substitution for every node. As a result, the total running time is $O(nN_\alpha N_\beta N_q^2)$, which based on the first part (Section D.1) is $O\left(n(\frac{1}{\epsilon})^{l_1+l_2+2}\right)$. □

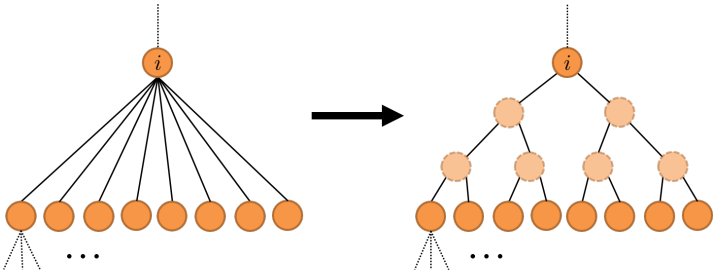


Fig. 7. The transformation of an arbitrary-degree tree to a binary tree

D.3 Remark on the ϵ-Approximation

As mentioned at the end of Section 4.2, if one requires the total payment in Definition 4.7 to be at most ϵ (rather than $n\epsilon$) away from the optimal p^*, the running time of our algorithm will still be polynomial in both n and $1/\epsilon$, i.e., $O\left(n^3(\frac{1}{\epsilon})^{l_1+l_2+2}\right)$. To see that, notice in this case (30) and (31) remain the same, and (32) changes to $nK\delta \leq \epsilon$. Therefore, the upper bound enforced by the constraints will be $\delta \leq \frac{C\epsilon}{n}$, for some constant C. In this case, our choice of δ would be $\delta = \frac{d}{\lceil \frac{dn}{C\epsilon} \rceil}$, and hence $N_q = O\left(\frac{n}{\epsilon}\right)$. N_α and N_β remain the same as before. The running time is $O(nN_\alpha N_\beta N_q^2)$, as computed previously, which in this case would be $O\left(n^3(\frac{1}{\epsilon})^{l_1+l_2+2}\right)$.

E SUPPLEMENT TO SECTION 5

In this section, we first show the transformation of the problem on a tree to one on a binary tree, and then prove Theorem 5.3.

E.1 Transformation into Binary Tree

Lemma E.1. *Given any tree with n nodes (suppliers), there exists a binary tree with additional nodes which has the same solution* $(q_i^*, \ldots, q_n^*, \alpha^*, \beta_1, \ldots, \beta_n)$ *for those nodes as the original network. The binary tree has* $O(n)$ *nodes.*

Proof. Take any node i that has $k_i > 2$ children. For any two children introduce a dummy parent node. For any two dummy parent nodes introduce a new level of dummy parent nodes. Continue this process until there are 2 or less nodes in the uppermost layer, and then connect them to node i (See Fig. 7). The capacities of the lines immediately connected to the children are the same as those in the original graph. The capacities of the new lines are infinite.

The total number of introduced dummy nodes by this procedure is

$$O(\frac{k_i}{2} + \frac{k_i}{4} + \cdots + 2) = O(k_i).$$

Since there are $1 + k_1 + k_2 + \cdots + k_n = n$ nodes in total in the original tree, the number of introduced additional nodes is $O(k_1 + \cdots + k_n) = O(n)$. Therefore the total number of nodes in the new (binary) tree is $O(n)$. □

E.2 Proof of Theorem 5.3

Most of the proof is similar to the one presented in Section D. For this reason, we only highlight the main points. The proof consists of ϵ-accuracy and run-time, as before.

E.2.1 ϵ-Accuracy.

Let $Q_1, \ldots, Q_n, F_1, \ldots, F_n, \mathcal{A}', \mathcal{B}'$ denote some δ-discretizations of sets $[0, d_1 + \overline{f_{\mathrm{ch}_1(1)}} + \overline{f_{\mathrm{ch}_2(1)}} - \underline{f_1}], \ldots, [0, d_n + \overline{f_{\mathrm{ch}_1(\mathrm{n})}} + \overline{f_{\mathrm{ch}_2(\mathrm{n})}} - \underline{f_n}]$, $[\underline{f_1}, \overline{f_1}], \ldots, [\underline{f_n}, \overline{f_n}]$, $\mathcal{A}$, $\mathcal{B}$, respectively. Note that if any line capacities are infinite, the intervals can be replaced by $[0, \sum_{i=1}^{n} d_i]$ instead. Similar as in Section D, the constraints enforce an upper bound on the value of δ as $\delta \leq C\epsilon$, for some constant C. Based on Lemma (D.1), the sizes of the sets will be $N_{q_i} = O\left(\frac{1}{\epsilon}\right)\ \forall i$, $N_{f_i} = O\left(\frac{1}{\epsilon}\right)\ \forall i$, $N_\alpha = O\left(\frac{1}{\epsilon^{l_1}}\right)$ and $N_\beta = O\left(\frac{1}{\epsilon^{l_2}}\right)$

E.2.2 Run-Time Analysis.

For every fixed α, the run-time of the required computations is as follows.

(1) The time complexity of computing $g_i(q_i; \alpha)$ for each node i and each fixed value of q_i is $O(N_\beta N_{q_i})$. Therefore, computing it for all nodes and all values takes $O(nN_\beta N_q^2)$.

(2) Computing $h_i(f_i; \alpha)$ for each node i and each fixed value of f_i takes $O(N_f^2)$, because there are $O(N_f) \times O(N_f)$ pairs of values for $(f_{\mathrm{ch}_1(i)}, f_{\mathrm{ch}_2(i)})$ (q_i is automatically determined as the closest point in Q_i to $d_i + f_{\mathrm{ch}_1(i)} + f_{\mathrm{ch}_2(i)} - f_i$). Therefore, its overall computation for all nodes and all values takes $O(nN_f^3)$.

As a result, the overall computation takes $N_\alpha \times \left(O\left(nN_\beta N_q^2\right) + O\left(nN_f^3\right)\right)$, which is $O\left(n(\frac{1}{\epsilon})^{l_1+l_2+2}\right) + O\left(n(\frac{1}{\epsilon})^{l_1+3}\right)$, or equivalently $O\left(n(\frac{1}{\epsilon})^{l_1+\max\{l_2,1\}+2}\right)$. □