Efficient Warm-Start Strategies for Nash-based Linear Complementarity Problems via Bilinear Approximation

Dominic C. Flocco, Steven A. Gabriel

Abstract. We present an effective warm-starting scheme for solving large linear complementarity problems (LCPs) arising from Nash equilibrium problems. The approach generates high-quality starting points that, when passed to the PATH solver, yield substantial reductions in computational time and variance. Our warm-start routine reformulates each agent's LP using strong duality, leading to a master problem with bilinear constraints that is equivalent to the original LCP. Bilinear terms emerge, for example, from system-level variables in the overall equilibrium problem that are exogenous to individual agent optimization problems. Approximate solutions are obtained using the difference-of-convex function algorithm for bilinear terms (DCA-BL) or a spatial branch-and-bound method (SBB). Unlike conventional bilinear approximation schemes, such as McCormick envelopes, DCA-BL does not rely on tight variable bounds. We test the scheme on a realistic LCP instance derived from a stochastic natural gas equilibrium model with nearly 100,000 variables. Without warm starts, PATH struggles to solve these instances within 24 hours. With DCA-BL or SBB warm starts, solution times drop significantly; the largest instance is solved in about one hour after two hours of warm start. While both warm-start approaches yield faster and less variable computational times, experiments suggest that DCA-BL provides the best starting point, as measured by the resulting PATH runtime. Although demonstrated on a specific LCP, the warm-start method extends to any LCP derived from the KKT conditions of LPs for each agent combined with linear system-level constraints.

1. Introduction

A widely used and flexible framework for modeling systems with multiple, non-cooperative decision-makers, e.g., energy markets, is that of a Nash equilibrium. In this setting, markets are modeled as non-cooperative games with rational agents, where each agent (or player) independently solves an optimization problem while treating the strategies of all other players as fixed. The appropriate solution concept for such games is a Nash equilibrium, at which no individual agent has an incentive to deviate unilaterally. To formulate these models, the first-order Karush–Kuhn–Tucker (KKT) optimality conditions of each player's problem are combined, together with any system-level or shared constraints, to obtain a unified representation of the equilibrium system. This naturally leads to a linear, mixed complementarity problem (MCP) formulation, which has become the standard approach for modeling and solving such equilibrium problems [8, 11, 16, 21]. For example, such systems and associated MCP models are quite common in energy markets but not exclusively to this domain. Additional applications include other infrastructure networks, such as water or transport [1, 2, 5, 6].

Complementarity problems represent a challenging subclass of nonconvex optimization. In the special case of MCPs, the resulting mathematical formulation is a feasibility problem with nonconvexities arising, at least, from bilinear equality constraints of the form $F_i(x)x_i = 0, \forall i$. When $x, F(x) \geq 0$, this can be written equivalently as $F(x)^\top x = 0$ and we express the relationship in shorthand as $0 \leq F(x) \perp x \geq 0$.

Date: February 28, 2026.

Key words and phrases. Complementarity problems; difference-of-convex function programming; Nash equilibrium; bilinear approximation.

Data Statement. The data supporting these findings are not publicly available due to non-disclosure agreements established under the research contract.

Declarations: The authors declare no conflicts of interest. This research was supported by Petrobras under Grant No. GR013290.

Methods for solving MCPs generally fall into two categories: (1) pivoting methods and (2) continuous, iterative methods. Pivoting methods, such as Lemke's algorithm, offer a robust approach for small-to-medium sized problems, but might not scale well for larger instances and often suffer from numerical instability. As such, iterative methods, namely the PATH algorithm [10], are widely regarded as the most successful. The associated PATH solver has become the standard for solving large-scale linear and nonlinear complementarity problems. However, as is the case for many nonconvex optimization solvers, PATH solver performance is highly dependent on the provided starting point.

In this work, we present a new approximation scheme for generating high-quality warm starts for linear complementarity problems (LCPs). The approach is geared towards real-world applications of LCPs, which may be ill-conditioned with large problem dimension, typically on the order of 10^4-10^6 variables. The large number of variables is a function of, for example, the number of agents in the market, length of the temporal planning horizon or the number of stochastic scenarios.

In this setting, we seek a Nash equilibrium of a non-cooperative game in which each player solves a linear program (LP), resulting in a large LCP when all the LPs' optimality conditions are combined. In the warm-starting scheme, rather than concatenating the KKT conditions of each player, we instead construct a master problem that includes primal feasibility, dual feasibility, and strong duality for each player. Because each player's problem is linear, strong duality holds. The resulting formulation is comprised of a set of linear constraints and one bilinear constraint for each player to represent strong duality. These bilinear terms arise from the possible presence of decision variables in the primal objective function or constraints of an individual player that are exogenous to that player, but endogenous to the overall LCP, such as prices. The proposed warm-start routine seeks to approximate these bilinear terms to generate a good starting point to pass to the PATH solver.

The bilinear approximation used in the warm-start routine is based on difference-of-convex (DC) function programming [9], a subfield of nonconvex optimization concerned with finding the extrema of functions expressed as the difference of convex functions. The routine relies on the recasting of bilinear terms as the difference of scalar convex quadratic functions, a reformulation explored in general terms in [12] and referred to as DCA-BL. The concave part of the resulting DC decomposition is iteratively approximated using the standard general DC algorithm (DCA) from [17]. This presolve approach is compared with Gurobi's spatial branch-and-bound algorithm, which uses McCormick envelopes to build out a spatial branch-and-bound tree. Contrary to these conventional approaches, a benefit of the DCA-BL warm-start technique is that it does not explicitly rely on the derivation of tight bounds on variables appearing in the bilinear constraints. Rather, DCA-BL determines the set of supporting hyperplanes for these bilinear terms using a simple first-order Taylor series approximation. In addition to this benefit, we find that DCA-BL generates better starting points than the spatial branch-and-bound algorithm, as measured by the resulting PATH runtime from this starting point. Overall, extensive numerical experiments indicate that both warm-start approaches generate high-quality starting points that result in faster and less variable solution times from the PATH solver.

2. Linear Complementarity Problems

Consider a set of players indexed by the set $P = \{1, \ldots, n_p\}$ competing in a non-cooperative game, where each player $p \in P$ solves a linear program of the form

$$\underset{x^p}{\text{minimize}} \quad c^p(x^{-p}, \nu)^\top x^p \tag{1a}$$

$$\text{s.t.} \quad A^p x^p \geq b^p(x^{-p}, \nu) \qquad (\lambda^p) \tag{1b}$$

$$x^p \geq 0, \tag{1c}$$

where player p's primal decision variables and Lagrange multipliers are denoted by the vectors $x^p \in \mathbb{R}^{N_p}$ and $\lambda^p \in \mathbb{R}^{M_p}$, respectively. Moreover, $x^{-p} = \{x^{p'} : p' \in P, p' \neq p\}$ is the set of decision variables for the other players and $\nu \in \mathbb{R}^\ell$ is a set of exogenously determined Lagrange multipliers determined via ℓ affine system-level constraints, described below. Thus, x^{-p} and ν are exogenous to player p, so that $c^p(x^{-p}, \nu) : \mathbb{R}^{\tilde{N}_p} \times \mathbb{R}^\ell \to \mathbb{R}^{N_p}$ and $b^p(x^{-p}, \nu) : \mathbb{R}^{\tilde{N}_p} \times \mathbb{R}^\ell \to \mathbb{R}^{M_p}$ are vector-valued, affine functions of data that are exogenous to player p, with $\tilde{N}_p = \sum_{p' \neq p} N_p$. This could include other player's variables, or system-level decisions not specific to a single player, such as price. Importantly, the vectors c^p and b^p that appear in player p's optimization problem are functions of data that are endogenous, i.e., decision variables, in the overall LCP.

The necessary and sufficient Karush-Kuhn-Tucker (KKT) conditions for player p's optimization problem are the following: find vectors (x^p, λ^p) such that

$$0 \leq c^p(x^{-p}, \nu) - (A^p)^\top \lambda^p \quad \perp \quad x^p \geq 0 \tag{2a}$$

$$0 \leq A^p x^p - b^p(x^{-p}, \nu) \quad \perp \quad \lambda^p \geq 0 \tag{2b}$$

which can be written more succinctly as:

$$0 \leq \begin{pmatrix} 0 & -(A^p)^\top \\ A^p & 0 \end{pmatrix} \begin{pmatrix} x^p \\ \lambda^p \end{pmatrix} + \begin{pmatrix} c^p(x^{-p}, \nu) \\ -b^p(x^{-p}, \nu) \end{pmatrix} \perp \begin{pmatrix} x^p \\ \lambda^p \end{pmatrix} \geq 0. \tag{3}$$

As a concrete example, in energy market modeling, the players can be producers, consumers or other market agents. Their primal variables represent, for example, production or consumption quantities in a certain time period, and price may be formed by the dual multiplier to a system-level, market-clearing constraint. We do not delve into the details of such models here, but focus on their high-level structure by keeping notation as general as possible.

In addition to the KKT conditions of each player, we consider a set of system-level constraints that link the decisions of each player. In the context of energy market modeling, system-level constraints often describe price formation in the market. The reader is referred to [11, 13] for further detail on price formation in the context of complementarity modeling of energy markets. For this discussion, we consider a general set of system-level constraints to emphasize the flexibility in the proposed warm-start approach. In particular, consider the affine, system-level complementarity constraints

$$0 \leq \sum_{p \in P} (k_i^p)^\top x^p + d_i \quad \perp \quad \nu_i \geq 0 \qquad \forall i = 1, \ldots, \ell, \tag{4}$$

where $k_i^p \in \mathbb{R}^{N_p}$ represents the vector of coefficients for player p's decisions $x^p \in \mathbb{R}^{N_p}$ in system-level constraint i, $d_i \in \mathbb{R}$ is a scalar and $\nu_i \in \mathbb{R}$ is the associated multiplier.

In the context of energy market equilibria, a common instance of a system-level constraint of the form (4) is a market-clearing condition, which balances supply and demand to form the market price. The presence of such a constraint is used to model perfect competition, or price-taking strategies by producers. Given a discrete planning horizon $T = \{1, \ldots, n_t\}$, the market-clearing condition in period $t \in T$ is

$$0 \leq \sum_{p \in P} x_t^p - D_t \quad \perp \quad \pi_t \geq 0, \qquad \forall t = 1, \ldots, n_t, \tag{5}$$

where D_t is the (inelastic) market demand in time period t, x_t^p is the supply from player p in period t and π_t is the market price for period t. In the context of (4), the time index $t = 1, \ldots, n_t$ takes the place of $i = 1, \ldots, \ell$, coefficients $k_i^p = e_t = [0, \ldots, 1, \ldots, 0]^\top$ (i.e., a null vector with 1 in the tth entry), $d^i = -D_t$ and $\nu_i = \pi_t$ is the market price in period t. However, (4) could represent multiple types of linear system-level constraints. Another common example is a shared resource constraint, such as pipeline capacities in gas models. In this setting, the system-level constraint is of the form

$$0 \leq -\sum_{p\in P} x_a^p + \kappa_a \quad \perp \quad \eta_a \geq 0, \qquad \forall a \in \mathcal{A}, \tag{6}$$

where x_a^p is the flow of player p's gas through arc $a \in \mathcal{A}$, and κ_a is the volumetric capacity in arc a. In this example, the associated multiplier η_a is interpreted as the congestion price.

Importantly, the associated dual variable, ν_i in its general form (4), or π_t or η_a in the examples (5) and (6), respectively, must appear elsewhere in the MCP to maintain a square system [11]. In the context of the market-clearing conditions (5) the dual variable π_t typically appears in the objective of each profit-maximizing producer, and similarly for the congestion price η_a. These multipliers are grouped into the vector of exogenous variables $c^p(x^{-p}, \nu)$.

Concatenating the KKT conditions (2) for each each player $p \in P$, in addition to system-level constraints of the form (4), results in a LCP of the form

$$0 \leq Mz + q \qquad \perp \quad z \geq 0, \tag{7}$$

where $z = \{(x^p, \lambda^p)_{p\in P}, \nu\}$ are the decision variables, q is a vector and M the LCP coefficient matrix. The specific structure of the matrix M and vector q in this general setting is derived in Appendix B for interested readers.

2.1. **Energy Market Equilibrium Example.** Now, we consider a concrete example of an energy LCP from [13]. We note that this is a much more simple model than the one used in the numerical experimentation described in Section 5, which has stochastic scenarios and contracts. Nonetheless, the models share similar attributes and application. The purpose of this discussion is to illustrate how the warm-start routine can be employed on a simple LCP model. We begin by providing a concrete example of how an LCP of the form (7) is constructed from a set of linear programs and system-level constraints.

In this example, we consider a set of producers $P = \{1, \dots, n_p\}$ participating in a Cournot game. Each producer p must decide the quantity x_t^p of gas to produce in the market during period t over a discretized planning horizon $T = \{1, \dots, n_t\}$. In this setting, producer p solves the following profit-maximization problem:

$$\underset{x^p}{\text{minimize}} \quad f^p(x,\pi) = \sum_{t\in T} \left[(\gamma_t^p + \tau^{in}) x_t^p - \pi_t x_t^p \right] \tag{8a}$$

$$\text{s.t.} \quad \sum_{t\in T} x_t^p \leq X^{p,\text{total}} \qquad (\lambda^{p,\text{total}}) \tag{8b}$$

$$x_t^p \leq X_t^{p,rate} \qquad (\lambda_t^p) \qquad \forall t \in T. \tag{8c}$$

$$x_t^p \geq 0 \qquad \forall t \in T. \tag{8d}$$

The objective (8a) is to maximize profit (expressed equivalently as minimizing negative profit), where costs are determined by the constant, marginal production cost γ_t^p of each unit of gas plus the entry tariff τ^{in} at a virtual hub, and revenue is given by the market price π_t. Given the upper bounds $X^{p,\text{total}}$ and $X_t^{p,rate}$, constraints (8b) and (8c) enforce volumetric and capacity constraints on production, respectively. The objective function (8a) and all constraints are linear; hence, the KKT conditions are necessary and sufficient for optimality, and are expressed as follows:

$$0 \leq \gamma_t^p + \tau^{in} - \pi_t + \lambda_t^p + \lambda^{p,\text{total}} \quad \perp \quad x_t^p \geq 0 \qquad \forall p \in P, t \in T \tag{9a}$$

$$0 \leq X^{p,\text{total}} - \sum_{t\in T} x_t^p \quad \perp \quad \lambda^{p,\text{total}} \qquad \forall p \in P \tag{9b}$$

$$0 \leq X_t^{p,rate} - x_t^p \quad \perp \quad \lambda_t^p \geq 0 \qquad \forall p \in P, t \in T. \tag{9c}$$

For this example, we assume producers are price-takers, as opposed to price-makers, i.e., producers cannot directly influence the price in the market. As such, the market price π_t is formed by the market-clearing conditions

$$0 \leq \sum_{p \in P} x_t^p - D_t \quad \perp \quad \pi_t \geq 0 \qquad \forall t \in T, \tag{10}$$

where D_t is the demand in the market during period t. The market-clearing conditions (10) are an example of a general system-level constraint of the form (4), with $\ell = |T|$ system-level constraints. Together, the KKT conditions (9) for each player $p \in P$, along with the market-clearing conditions (10), define the overall LCP.

3. Warm-Start Master Problem

We now return to the general setting to outline a linear program with bilinear constraints that is equivalent to the LCP (7). The resulting quadratically constrained linear program (QCLP) will serve as the warm-start problem. Rather than concatenating the KKT conditions of each player, we instead construct a master problem that includes primal feasibility, dual feasibility and strong duality for each player. In the warm-starting schemes, we will compute an approximation to this master linear program to use as a starting point for PATH. Consider the corresponding dual LP to the primal LP (1) for player p

$$\begin{aligned} \underset{\lambda^p}{\text{maximize}} \quad & b^p(x^{-p}, \nu)^\top \lambda^p && \text{(11a)}\\ \text{s.t.} \quad & (A^p)^\top \lambda^p \leq c^p(x^{-p}, \nu) \qquad (x^p) && \text{(11b)}\\ & \lambda^p \geq 0, && \text{(11c)} \end{aligned}$$

where $c^p(x^{-p}, \nu)$ and $b^p(x^{-p}, \nu)$ are again vector-valued, affine functions of x^{-p} and ν. LP theory (see, e.g., [3]) indicates that the KKT conditions are necessary and sufficient for optimality but so are: primal feasibility, dual feasibility, and strong duality.

For the primal, dual pair (1), (11), strong duality states that at optimality $c^p(x^{-p,*}, \nu^*)^\top x^{p,*} = b^p(x^{-p,*}, \nu^*)^\top \lambda^{p,*}$ for $(x^{p,*}, \lambda^{p,*})$ optimal to their respective primal and dual problems. Weak duality, by contrast, states that for all primal feasible x^p and all dual feasible λ^p, we must have $c^p(x^{-p}, \nu)^\top x^p \geq b^p(x^{-p}, \nu)^\top \lambda^p$. The resulting new problem to solve includes the primal and dual feasibility for each producer and consumer LP as well as either "reverse weak duality" or strong duality. Reverse weak duality here is $c(x^{-p}, \nu)^\top x^p \leq b(x^{-p}, \nu)^\top \lambda^p$, which, when combined with weak duality (automatically satisfied for feasible primal and dual solutions to an LP), will result in strong duality. The computational advantage of using reverse weak duality is that it admits a larger feasible region than strong duality for the resulting QCLP.

We form the resulting optimization problem with an arbitrary objective function $f(x, \lambda, \nu)$ as:

$$\begin{aligned} \underset{x,\nu,\lambda}{\text{minimize}} \quad & f(x, \lambda, \nu) && && \text{(12a)}\\ \text{s.t.} \quad & A^p x^p \geq b^p(x^{-p}, \nu),\ x^p \geq 0 && \forall p \in P && \text{(12b)}\\ & (A^p)^\top \lambda^p \leq c^p(x^{-p}, \nu),\ \lambda^p \geq 0 && \forall p \in P && \text{(12c)}\\ & c^p(x^{-p}, \nu)^\top x^p \leq b^p(x^{-p}, \nu)^\top \lambda^p && \forall p \in P. && \text{(12d)} \end{aligned}$$

Here, (12b) and (12c) represent primal and dual feasibility for player $p \in P$, respectively, and (12d) is the reverse weak duality constraint for player p. The objective $f(x, \lambda, \nu)$ could be any (linear) function of x, λ and ν, including just $f(x, \lambda, \nu) = 0$. For example, $f(x, \lambda, \nu)$ could be to maximize social welfare or minimize total emissions, see, e.g., [13], or a slack penalty term as described in Section 3.1. Note that here we use $x = \{x^p\}_{p \in P}$, $\lambda = \{\lambda_p\}_{p \in P}$ and $\nu = \{\nu_i\}_{i=1}^{\ell}$ to denote the vector of all primal and dual decision variables for all players, and all system-level constraint multipliers.

For brevity, define $A = \operatorname{diag}(A^1, \ldots, A^{n_p})$, $b = \{b^p(x^{-p})\}_{p \in P}$ and $c = \{c^p(x^{-p})\}_{p \in P}$ so that we can express (12) as

$$\begin{aligned}
&\underset{x,\nu,\lambda}{\text{minimize}} && f(x,\lambda,\nu) && && (13a)\\
&\text{s.t.} && Ax \geq b, x \geq 0 && && (13b)\\
& && A^\top \lambda \leq c, \lambda \geq 0 && && (13c)\\
& && c^p(x^{-p},\nu)^\top x^p \leq b^p(x^{-p},\nu)^\top \lambda^p && \forall p \in P, && (13d)
\end{aligned}$$

where (13b) and (13c) represent primal and dual feasibility for all players. We denote by $N = \sum_{p\in P}(N_p + M_p) + \ell$ the dimension of the decision vector (x, λ, ν) for the master problem (13).

The question now is how to deal with the system-level constraints (4). Generally, the disjunctive nature of the complementarity condition (4) can be modeling using big-M constraints via the introduction of binary variable $b_i \in \{0, 1\}$ and a suitably large constant M_i for $i = 1, \dots, \ell$. The system-level constraints (4) are then equivalent to:

$$\begin{aligned}
&0 \leq \sum_{p\in P}(k_i^p)^\top x^p + d_i \leq M_i(1 - b_i) && i = 1\dots,\ell && (14a)\\
&0 \leq \nu_i \leq M_i b_i && i = 1\dots,\ell && (14b)\\
&b_i \in \{0,1\} && i = 1\dots,\ell. && (14c)
\end{aligned}$$

By adding (14) to the master problem (13), we obtain a mixed integer program (MIP). Let $\overline{x}^p$ and $\underline{x}^p$ denote the upper and lower bounds on player p's primal decision vector x^p. Then, based on (14a), a lower bound on the sufficient value of M_i is given by

$$M_i \geq \sum_{p\in P}\left(\sum_{j:k_{i,j}^p>0} k_{i,j}^p \overline{x}_j^p + \sum_{j:k_{i,j}^p<0} k_{i,j}^p \underline{x}_j^p\right) + \max(0, d_i) \qquad i = 1,\dots,\ell,$$

where $k_{i,j}^p$ is the jth element of vector k_i^p. Often times, the upper and lower bounds $\overline{x}^p$ and $\underline{x}^p$ of player p's primal variables are easily derivable from problem-specific information. For example, in the energy market equilibrium problem described in Section 2.1, player p's primal decision x_t^p is upper bounded by the maximum production $X_t^{p,rate}$. Therefore, when applying the reformulation (14) to the market-clearing conditions (10), the lower bound on M_t is $M_t \leq \sum_{p\in P} X_t^{p,rate}$ for each $t \in T$.

Depending on the nature of the overall MCP, this MIP may be at least as computationally challenging to solve as the original MCP. Since our goal is to efficiently compute an approximate solution to (13) that can be used as a starting point, we do not need to solve (13)-(14) exactly. As such, we propose the following options for dealing with (4) and producing an efficient warm start scheme:

1) Relax the integrality constraint (14c) in the big-M formulation (i.e., replace (14c) by $0 \leq b_i \leq 1$ for all $i = 1, \dots, \ell$).
2) Fix a subset, or all, of the binary variables b_i based on problem-specific information to reduce, or remove, the set of binary constraints. For example, if (4) is comprised of market-clearing constraints of the form (5), assuming the price $\pi_t > 0$ then we may fix the corresponding binary variable $b_i = 1$ and we are left with a set of linear constraints for the system-level conditions.

The choice of how to deal with system-level constraints may be specific to the problem formulation, computational resources available and quality of warm start required.

3.1. **Slack-Minimization Reformulation.** Noting that $c^p(x^{-p}, \nu)$ and $b^p(x^{-p}, \nu)$ are vector-valued, affine functions of x^{-p} and ν, the challenge in solving (13) is in the presence of bilinear terms in the reverse weak-duality constraint (13d). To initially address this nonconvex constraint set, we first relax the inequality constraint by introducing the slack variable $t \geq 0$, which is minimized in the objective:

$$\begin{aligned}
\underset{x,\lambda,\nu,t}{\text{minimize}} \quad & t && \text{(15a)}\\
\text{s.t.} \quad & Ax \geq b, x \geq 0 && \text{(15b)}\\
& A^\top \lambda \leq c, \lambda \geq 0 && \text{(15c)}\\
& c^p(x^{-p},\nu)^\top x^p - b^p(x^{-p},\nu)^\top \lambda^p \leq t \qquad \forall p \in P. && \text{(15d)}\\
& t \geq 0. && \text{(15e)}
\end{aligned}$$

Hence, when $t = 0$, (15) is equivalent to (13) with $f(x, \lambda, \nu) = 0$. In this sense, (15) is a relaxed version of (13). The benefit of this approach is that we've now expanded the original feasible region, widening the search space.

Lastly, we rewrite (15) to isolate the bilinear terms for clarity. The warm-starting routine for solving the associated LCP (7) relies on finding approximate solutions to the QCLP

$$\begin{aligned}
\underset{x,\lambda,\nu,t}{\text{minimize}} \quad & t && \text{(16a)}\\
\text{s.t.} \quad & Ax \geq b, x \geq 0 && \text{(16b)}\\
& A^\top \lambda \leq c, \lambda \geq 0 && \text{(16c)}\\
& \phi^p(x,\lambda,\nu) + \sum_{i \in \mathcal{I}^p} q_i^p x_i \nu_i \leq t \qquad \forall p \in P, && \text{(16d)}\\
& t \geq 0, && \text{(16e)}
\end{aligned}$$

where $\phi^p(x, \lambda, \nu)$ is a affine function of x, λ and ν, and q_i^p is a scalar. The bilinear terms to be approximated are of the form $x_i \nu_i$, where $i \in \mathcal{I}^p$ is the index of player p's variables that require bilinear terms. Note that some of the bilinear terms may be of the form $x_i^p x_i^{-p}$. However, for consistency of notation and brevity, we express this term generally as $x_i \nu_i$. We note that in (16) there is only one bilinear constraint (16d) for each player, and typically the number of players is significantly smaller than the overall dimension of the LCP (i.e., $|P| \ll N$).

3.2. **Energy Market Equilibrium Example.** Next, we continue with the the price-taker energy equilibrium example from Section 2.1 and derive the dual and master warm-start problem. The dual to (8) can be written as follows:

$$\begin{aligned}
\underset{\lambda}{\text{maximize}} \quad & X^{p,\text{total}} \lambda^{p,\text{total}} + \sum_{t \in T} X_t^{p,rate} \lambda_t^p && \text{(17a)}\\
\text{s.t.} \quad & \gamma_t^p + \tau^{in} - \pi_t + \lambda_t^p + \lambda^{p,\text{total}} \geq 0 \qquad (x_t^p) \qquad \forall t \in T && \text{(17b)}\\
& 0 \leq \lambda_t^p \qquad \forall t \in T && \text{(17c)}\\
& 0 \leq \lambda^{p,\text{total}}. && \text{(17d)}
\end{aligned}$$

Weak duality for producer p implies that

$$X^{p,\text{total}} \lambda^{p,\text{total}} + \sum_{t \in T} X_t^{p,rate} \lambda_t^p \leq \sum_{t \in T} (\gamma_t^p + \tau^{in}) x_t^p - \pi_t x_t^p, \tag{18}$$

which must hold for all feasible solutions to the primal problem (8) and dual problem (17). If the relationship (18) holds at equality, we have strong duality. Therefore, following the logic outlined in

Section 3.1, we introduce slack parameter $t \geq 0$ and arrive at the following warm-start problem:

$$\begin{aligned}
&\underset{x,\lambda,\pi,t}{\text{minimize}} \quad t && && \text{(19a)}\\
&\text{s.t.} \quad \sum_{t\in T} x_t^p \leq X^{p,\text{total}} && \forall p \in P && \text{(19b)}\\
&x_t^p \leq X_t^{p,rate} && \forall t \in T, p \in P && \text{(19c)}\\
&\gamma_t^p + \tau^{in} - \pi_t + \lambda_t^p + \lambda^{p,\text{total}} \geq 0 && \forall t \in T, p \in P && \text{(19d)}\\
&\left[\sum_{t\in T} (\gamma_t^p + \tau^{in}) x_t^p - \pi_t x_t^p\right] - \left[X^{p,\text{total}}\lambda^{p,\text{total}} + \sum_{t\in T} X_t^{p,rate}\lambda_t^p\right] \leq t && \forall p \in P && \text{(19e)}\\
&x_t^p, \lambda_t^p \geq 0 && \forall t \in T, p \in P && \text{(19f)}\\
&\lambda^{p,\text{total}} \geq 0 && \forall p \in P && \text{(19g)}\\
&t \geq 0. && && \text{(19h)}
\end{aligned}$$

The bilinear term appears in the relaxed, reverse weak duality constraint (19e) as price times quantity $\pi_t \cdot x_t^p$. While the market price π_t is exogenous data to each producer solving (8), it is endogenous to the overall LCP and formed by the system-level, market-clearing constraint (10). We note that for more complex models, the strong duality constraint (19e) may include addition of bilinear terms. Regardless, (19e) is the only bilinear constraint, and there are just $|P|$ of them. Without (19e), (19) is a LP.

4. Approximating Bilinear Terms

In this section, we present a difference-of-convex function based method to approximate the bilinear term $x_i\nu_i$ that appears in (16).

4.1. **Difference-of-Convex Functions Approach.** As introduced in [12], we can model the bilinear relationship $x_i\nu_i$ with difference-of-convex functions (DC). Introduce the auxiliary variables u_i, v_i for each $i \in \mathcal{I}$, where $\mathcal{I} = \bigcup_{p\in P} \mathcal{I}^p$ is the set of all bilinear indices in (16d). These auxiliary variables are related to x_i, ν_i via the following linear transformation:

$$\begin{pmatrix} \nu_i \\ x_i \end{pmatrix} = N \begin{pmatrix} u_i \\ v_i \end{pmatrix} \qquad \text{with} \qquad N = \begin{bmatrix} a & b \\ c & d \end{bmatrix}. \tag{20}$$

If N satisfies (i) $ad - bc \neq 0$, (ii) $ad + bc = 0$ and (iii) $ac \geq 0$, then N is invertible and there exist constants $\alpha_i, \beta_i > 0$ such that $x_i\nu_i = \alpha_i u_i^2 - \beta_i v_i^2$, where $\alpha_i = ac$ and $\beta_i = -bd$ (Theorem 1, Gabriel et al. [12]). Denote by $z = (x, \lambda, \nu, v, u)$ the new vector of decision variables. Under this transformation, we can express the bilinear terms $x_i\nu_i$ as the difference of scalar quadratic functions. In particular, we can express (16d) equivalently as

$$F^p(z) := \phi^p(x,\nu) + \sum_{i\in\mathcal{I}^p} q_i^p x_i \nu_i = \phi^p(x,\lambda,\nu) + \sum_{i\in\mathcal{I}^p} q_i^p(\alpha_i u_i^2 - \beta_i v_i^2), \tag{21}$$

where $\phi^p(x,\lambda,\nu)$ is again an affine function of x, λ and ν, and q_i^p is a scalar. Noting that $\alpha_i, \beta_i > 0$ for all $i \in \mathcal{I}$, (21) admits the following DC decomposition:

$$F^p(z) = \underbrace{\phi^p(x,\lambda,\nu) + \left[\sum_{i:q_i^p>0} q_i^p\alpha_i u_i^2 - \sum_{i:q_i^p<0} q_i^p\beta_i v_i^2\right]}_{G^p(z)} - \underbrace{\left[\sum_{i:q_i^p>0} q_i^p\beta_i v_i^2 - \sum_{i:q_i^p<0} q_i^p\alpha_i u_i^2\right]}_{H^p(z)}. \tag{22}$$

The basic idea of DC function programming is to linearly approximate second DC part (i.e., the concave part H^p) of (22) via it's first-order Taylor series. Generally, for a DC function $F(z) = G(z) - H(z)$, where G, H are convex, we approximate F at the current iterate z^k as $G(z) - [H(z^k) + \nabla H(z^k)^\top(z - z^k)]$.

More specifically, given the current iterate $z^k = (x^k, \lambda^k, \nu^k, v^k, u^k)$, the convex approximation of (22) is

$$F^p(z) = G^p(z) - H^p(z) \approx G^p(z) - \left[H^p(z^k) + \nabla H^p(z^k)^\top (z - z^k)\right], \tag{23}$$

where the the gradient $\nabla H^p(z^k)$ is comprised of the partial derivatives:

$$\frac{\partial H^p(z^k)}{\partial u_i} = -2q_i^p \alpha_i u_i^k \quad \text{and} \quad \frac{\partial H^p(z^k)}{\partial v_i} = 2q_i^p \beta_i v_i^k.$$

The result is a QCLP that approximates the solution to (16). This QCLP is given by

$$\begin{aligned}
&\underset{x,\lambda,\nu,v,u,t}{\text{minimize}} && \rho_k t && && \text{(24a)}\\
&\text{s.t.} && Ax \geq b, x \geq 0 && && \text{(24b)}\\
& && A^\top \lambda \leq c, \lambda \geq 0 && && \text{(24c)}\\
& && \begin{pmatrix} \nu_i \\ x_i \end{pmatrix} = N^i \begin{pmatrix} u_i \\ v_i \end{pmatrix}, && \forall i \in \mathcal{I}^p, p \in P && \text{(24d)}\\
& && G^p(z) - \left[H^p(z^k) + \nabla H^p(z^k)^\top (z - z^k)\right] \leq t && \forall p \in P, && \text{(24e)}
\end{aligned}$$

where $N^i \in \mathbb{R}^{2\times 2}$ is a suitably defined matrix according to (20), with elements a, b, c and d satisfying (i) $ad - bc \neq 0$, (ii) $ad + bc = 0$ and (iii) $ac \geq 0$, and $\rho_k > 0$ is a penalty parameter. We note that (24) has only $|P|$ convex, quadratic constraints in (24e), one for each player. Moreover, as noted in the original work [12], the variables ν and x may be substituted away with the auxiliary variables u and v via the linear transformation (24d) in implementation (as the matrix N^i is invertible). This way, the DC reformulation does not introduce additional decision variables compared to (16). Moreover, the slack parameter t is introduced so that (24) is always feasible, as noted by Le Thi et al. [17]. Hence, assuming the LCP (7) is solvable, (24) is feasible.

The resulting algorithm is presented in Algorithm 1 of [12] with (24) as the convex subproblem solved at each iteration. The DC algorithm is also restated in Appendix A for completeness. This approach, referred to as DCA-BL (DCA for bilinear terms), builds on the GDCA2 scheme from [17] using slack variables and adaptive penalty parameter updates. Theorem 2 of [12] provides the main convergence result for the algorithm presented in Appendix A, which states that if the algorithm stops, it is at a solution (16). Convergence in the classical sense does not hold due to the lack of the Mangasarian-Fromowitz constraint qualification of the LCP. Hence, DCA-BL should be viewed as a heuristic that computes approximate solutions to the warm-start problem.

Another way to handle such bilinear terms is to approximate them with McCormick envelopes, which form the basis of many spatial branch-and-bound algorithms and other relaxation techniques. The general idea of McCormick envelopes is to approximate the bilinear function $w_i = x_i\nu_i$ using the upper and lower bounds of x_i and ν_i [19]. However, a shortfall in this approach is that the accuracy of this approximation–and consequently the speed of algorithmic convergence–depends on the tightness of variable bounds. While primal (physical) variables often have intuitive, or easily derived bounds based on problem data, tight upper- and lower-bounds may be difficult to derive on the dual variables ν and λ^p appearing in (4) and (11), respectively. This difficulty is particularly amplified if the dual variable is free. Notably, the DC reformulation (24) does not directly rely on tight upper- or lower-bounds of the variables that appear in product terms. Rather, DCA determines the set of supporting hyperplanes for the bilinear constraints (22) using a simple first-order Taylor series approximation.

4.2. **Energy Market Equilibrium Example.** Now, we complete the energy equilibrium example from Sections 2.1 and 3.2 by applying the DCA-BL scheme to (19). Introduce auxiliary variables u_t^p, v_t^p for all $t \in T$ and $p \in P$, and scalar $\rho_k > 0$, which is a penalty parameter that is updated each iteration. Then, the convex subproblem solved at each DCA iteration k is

$$\underset{x,\lambda,\pi,t,u,v}{\text{minimize}} \quad \rho_k t \tag{25a}$$

$$\text{s.t.} \quad \sum_{t\in T} x_t^p \le X^{p,total} \qquad \forall p \in P \tag{25b}$$

$$x_t^p \le X_t^{p,rate} \qquad \forall t \in T, p \in P \tag{25c}$$

$$\gamma_t^p + \tau^{in} - \pi_t + \lambda_t^p + \lambda^{p,total} \ge 0 \qquad \forall t \in T, p \in P \tag{25d}$$

$$\begin{aligned} &\left[\sum_{t\in T} (\gamma_t^p + \tau^{in}) x_t^p + \left[\beta_t^p (v_t^p)^2 - \alpha_t^p (u_t^{p,k})^2 - 2\alpha_t^p u_t^{p,k}(u_t^p - u_t^{p,k})\right]\right] \\ &\qquad - \left[X^{p,total}\lambda^{p,total} + \sum_{t\in T} X_t^{p,rate}\lambda_t^p\right] \le t \qquad \forall p \in P \end{aligned} \tag{25e}$$

$$\begin{pmatrix} \pi_t \\ x_t^p \end{pmatrix} = N_t^p \begin{pmatrix} u_t^p \\ v_t^p \end{pmatrix} \qquad \forall t \in T, p \in P \tag{25f}$$

$$x_t^p \ge 0 \qquad \forall t \in T, p \in P \tag{25g}$$

$$\lambda_t^p \ge 0 \qquad \forall t \in T, p \in P \tag{25h}$$

$$\lambda^{p,total} \ge 0 \qquad \forall p \in P, \tag{25i}$$

$$t \ge 0, \tag{25j}$$

where N_t^p, α_t^p and β_t^p are defined according to (20) and $u_t^{p,k}$ and $v_t^{p,k}$ are the values of u_t^p, v_t^p at the current iteration k. The penalty parameter $\rho_k > 0$ is updated at each iteration according to Algorithm 1 in Appendix A, see, e.g., [12, 17, 20].

5. Numerical Results

We test the warm-start scheme on a set of large-scale LCPs arising from real-world application to natural gas market modeling. The model encompasses a stochastic Nash-Cournot equilibrium problem for computing equilibrium production quantities, consumption quantities and prices in the Brazilian natural gas network. The model is comprised of a set of producers and consumers, each solving a linear two-stage stochastic recourse problem (see, e.g., Birge and Louveaux [4] for more details). In the first stage, producers and consumers must decide on the quantity of gas to consume in a bilateral contract market, the prices of which are a function of a stochastic process. Then, their recourse decision is the quantity of gas to produce/consume in the the spot market versus the contract markets while respecting their contractual obligation decided upon in the first stage. The spot market price is determined via a market-clearing constraint, similar to (5) and in the general form (4), which is formed exogenously from any one agent's optimization problem. Concatenating the necessary and sufficient KKT conditions of each agent's linear program, in addition to system-level constraints of the form (4), results in a LCP of the form (7). The coefficient matrix for this LCP has the block structure shown in (30) in Appendix B. For brevity and data-sharing restrictions, we do not go into extensive detail on the specific formulation of the model but refer to [13] for details on a similar, deterministic model.

However, important to our numerical discussion is the role of time periods and scenarios in the model. The recourse problem of each agent grows as a function of the number of periods and stochastic scenarios considered in the problem instance. Specifically, each recourse decision variable and constraint is indexed by time period and scenario. As such, the number of variables in the overall LCP grows with the number of scenarios and periods. By modifying the number of periods and scenarios, we are able to generate a large number of different-sized problem instances to test the proposed warm-start routine. In particular, let n denote the number of LCP decision variables in the model formulation for one period and one scenario. Then, the total number of variables in the same model formulation with n_ω scenarios and n_t periods is approximately $N = n_\omega \cdot n_t \cdot n$. In our case, $n \approx 650$, and the model with $n_t = 12$ periods and $n_\omega = 12$ scenarios has 92,919 total variables.

The goal of the numerical experiments is to explore the computational gain from the proposed warm-start scheme for LCPs of varying size. We test two bilinear approximation methods for solving the warm-start problem (13): DCA-BL from [12] and Gurobi's off-the-shelf spatial branch-and-bound algorithm, referred to as SBB [14]. The SBB algorithm relies on constructing convex relaxations of the problem at each node of the search tree using McCormick envelopes [19]. Throughout the discussion, we refer to the warm-start routine as Phase 1, and the PATH solve as Phase 2. Using these two approaches for Phase 1, we compare the resulting Phase 2 runtime (i.e., PATH solve time) with no warm-start scheme applied. As we will see, the most significant computational improvements from warm starting come for large problem instances.

A natural question is whether using an intelligently devised, or guessed, starting point based on problem-specific knowledge is useful. For example, if there is prior knowledge of the market price, or distribution of a producer's production between different markets, one could use this to generate an adequate starting point. Such a starting point could be more accurate than using a trivial starting point (i.e., a vector of zeros) and less computationally expensive than the proposed warm-start approaches. However, in real-world applications, it is unlikely that such prior knowledge is known about all primal (physical) decisions, and even less likely that a reasonable guess at dual multipliers can be easily derived. We tested with this approach based on problem-specific knowledge of the natural gas equilibrium problem used in experimentation, and the resulting PATH runtimes were almost identical to those obtained using the baseline option (1) as a starting point. Since the derivation of an intelligent guess is problem-specific, and not reproducible, we omit this comparison in the proceeding analysis. Rather, we argue that the using even a poorly approximated solution to (15) as a starting point yields better PATH performance, since *all* LCP variables can be approximated and no problem-specific knowledge is required.

5.1. **Experimental Design.** We generate 144 instances of the LCP by varying the number of periods $n_t \in \{1, \ldots, 12\}$ and number of scenarios $n_\omega \in \{1, \ldots, 12\}$. The largest instance instance (with $n_t = 12$ periods and $n_\omega = 12$ scenarios) has 92,919 total variables. Each instance is solved with PATH using three different starting points, which are computed in Phase 1:

(1) The zero vector (i.e., no warm start), referred to as No-WS;
(2) Approximate solution to (15) computed by Gurobi's spatial branch-and-bound algorithm, referred to as SBB; and,
(3) Approximate solution to (15) computed by the DCA-BL scheme in [12], referred to as DCA-BL.

For each problem instance, we record the computational time required to solve the LCP with PATH (Phase 2) from each starting point. If PATH converges within the prescribed time limit from any of the starting points, a validation routine is run to ensure the computed solution is indeed a Nash equilibrium. Specifically, after the equilibrium solution is computed, each individual agent's LP (1) is solved with all exogenous variables, i.e., x^{-p} and ν, fixed to ensure no agent is incentivized to deviate from their current position. The system-level constraints in the model are market-clearing conditions of the form (5) that form prices π. In implementation of the warm-start problem, we make the assumption that $\pi > 0$ and remove these additional disjunctive constraints, i.e., we fix $b_i = 1$ in the implementation of big-M constraints (14). This is option 2 for dealing with system-level constraints from Section 3.

The time limit for Phase 2 is 6 hrs (21,600 seconds) and the time limit for Phase 1 (i.e., the time allowed to compute an approximate solution to (15)) is set to 0.25 hrs (900 seconds). All experiments were run on an Apple Silicon M2 Max processor with 64 GB RAM and 12 CPU cores running up to 3.49 GHz. The LCP and accompanying warm-start schemes are implemented using the Pyomo 6.9.2 [7, 15]

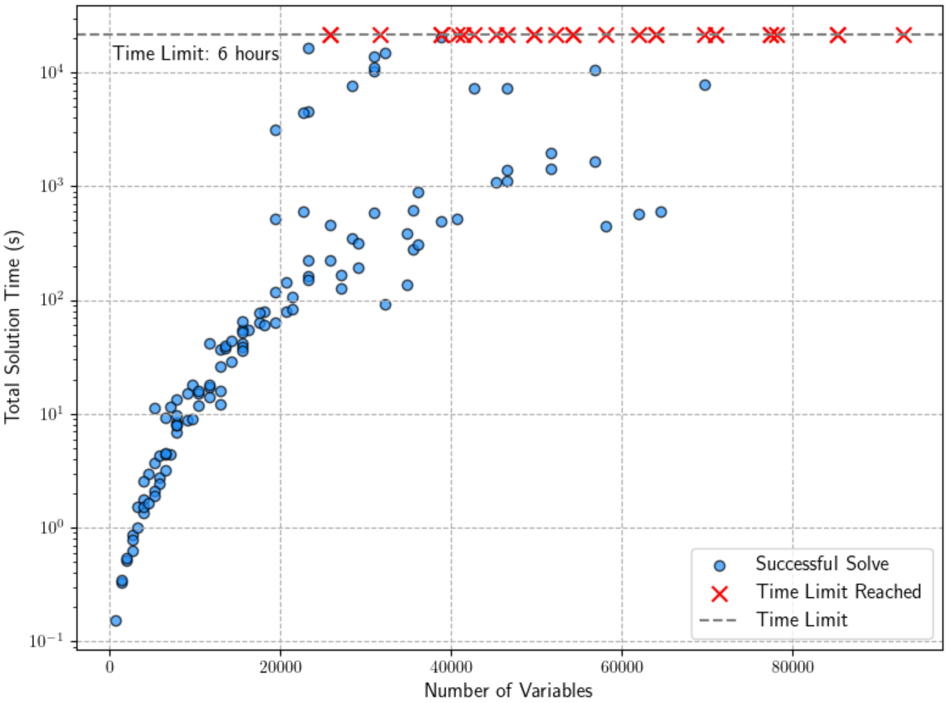

FIGURE 1. Computational time (log scale) of PATH (Phase 2) with no warm start (baseline) for varying number of LCP variables.

modeling package in Python 3.9.22. We use PATH 5.0.05 [10] to solve all LCPs and Gurobi 11.0.0 to solve DCA subproblems. The off-the-shelf spatial branch-and-bound algorithm, i.e., SBB warm start, also uses Gurobi 11.0.0. In the DCA-BL implementation, we use

$$N^i = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \forall i \in \mathcal{I}^p, p \in P$$

for the linear transformation in (24d). Based on limited testing performed in [12], the choice of linear transformation appears to have little impact on algorithm performance.

5.2. **Baseline Results.** Figure 1 shows the solve time required by PATH (i.e., Phase 2) using the zero vector as a starting point. These times will serve as the baseline for our warm-start experiment. With no warm starting applied, PATH solved 80.6% of instances within the prescribed time limit of 6 hours. An interesting observation is that the solution times show a consistent upward trend for small instances with minimal deviation. However, for problem instances greater than 20,000 variables, the solution times are much more variable. In particular, PATH reaches the time limit for some moderately sized instances, shown by the red “x”s in Figure 1, but solves slightly larger instances in a reasonable amount of time. This variability is exacerbated when analyzing instances larger than 40,000 variables, where PATH only solved 14 of 37 instances (37.8%) with a standard deviation among successful solves of 3488.59 seconds (about 1 hr). This variability is precisely what we would like to address with the warm-start approaches.

5.3. **Warm-start Results.** We analyze the computational times of PATH using the two proposed warm-start schemes: SBB and DCA-BL. We compare the Phase 2 computational time using starting points generated by these warm-start algorithms in Phase 1 with the baseline results shown in Section 5.2, referred to as No-WS. We note that except for small instances, specifically those with $n \leq 5000$, the SBB and DCA-BL warm-start algorithms did not solve (15) exactly in the prescribed presolve (Phase 1) time limit of 15 minutes, which is expected due to the size and complexity of the problem. Hence, Phase 1 should be viewed as computing an approximate LCP solution to (15) that is used as a starting point in

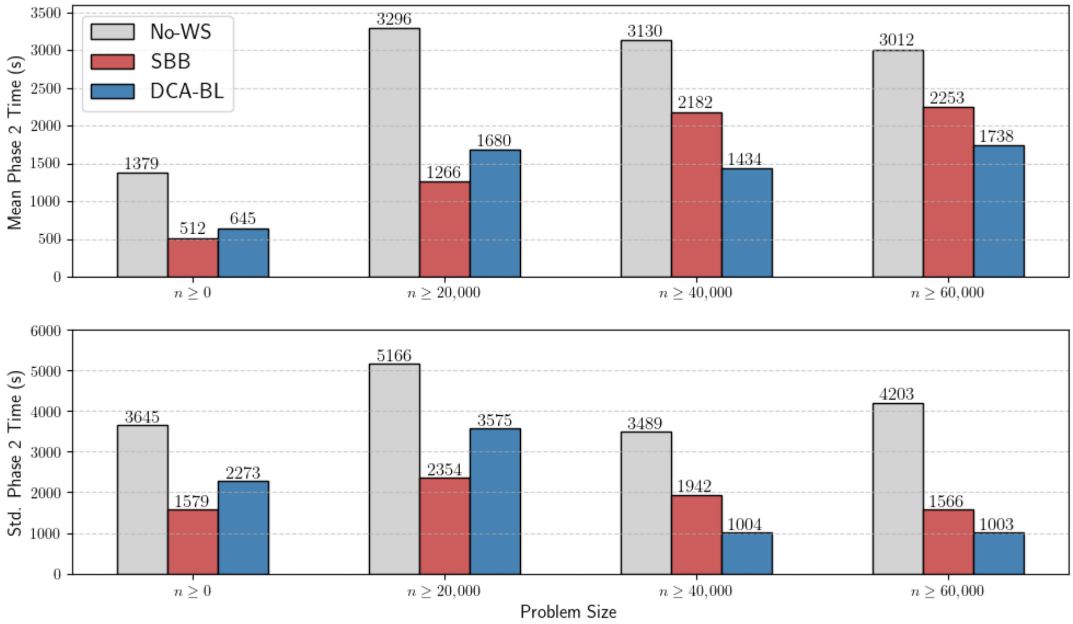

FIGURE 2. Mean (top) and standard deviation (bottom) of PATH computational times (Phase 2) using starting points generated by different warm-start routines (Phase 1).

Phase 2 by PATH to solve (7).

Figure 2 shows the geometric mean and standard deviation of the Phase 2 computational times for the instances successfully solved by PATH within 6 hours, divided into bins based on problem size. First, looking at the mean times shown in the top frame, observe that the mean time for No-WS is greater than both warm-start schemes for all problem dimensions. This indicates that, as expected, providing a high-quality starting point can be used to significantly improve the solution time of the PATH solver. Similarly, the standard deviation of solution time for the PATH solver is highest in the baseline case for all problem dimensions, as shown in Figure 2 (bottom). This further indicates that providing a high-quality starting point can decrease the variability in PATH computational times.

Next, we compare the Phase 2 computational times for each warm-start scheme for varying problem sizes, also using Figure 2. We first note that the bins used in Figure 2 represent a cumulative distribution of instances (e.g., the instances summarized in the $n \geq 20,000$ bin are also in the $n \geq 0$ bin). Hence, each bin does not have an equal number of problem instances. However, this is done intentionally for two reasons: (1) to understand the benefit of warm-starting as the problem size grows and (2) for data-visualization purposes. As shown in Figure 2, the standard deviation of Phase 2 runtime using No-WS is 3645.0 seconds (1.01 hours) for $n \geq 0$ and increases to 4203.3 seconds (1.17 hours) for $n \geq 60,000$. This suggests that not only does the PATH solve time exhibit high variability when no warm start is used, but this variability increases as the problem dimension grows. Conversely, when using SBB and DCA-BL warm start for $n \geq 0$, the standard deviation in Phase 2 solve time is 1579.4 seconds (0.44 hours) and 2273.1 (0.63 hours), respectively. Moreover, this variability decreases to 1566.4 seconds (0.43 hours) and 1003.2 seconds (0.28 hours) for $n \geq 60,000$ with SBB and DCA-BL warm start, respectively. This indicates (for these examples) that the warm-start schemes help in reducing the variability in PATH's computational time, and this performance gain increases as the problem size grows. Overall, DCA-BL and SBB warm-start approaches provide a percent improvement of 53.2% (2.13 speed-up) and 62.9% (2.69 speed-up) in average Phase 2 computational time across all instances (i.e., $n \geq 0$), respectively.[1]

[1] The *percent improvement* of warm-start method $s \in \{\texttt{DCA-BL, SBB}\}$ is computed using the formula $(t_0 - t_s)/t_0$. For $n \geq 0$, $t_0 = 1379.1$ seconds is the mean Phase 2 computational time with No-WS, and t_s is the mean Phase 2 time when using warm-start scheme s, where $t_s = 645.2$ for DCA-BL and $t_s = 511.8$ for SBB. The *speed-up* of warm-start scheme s is given

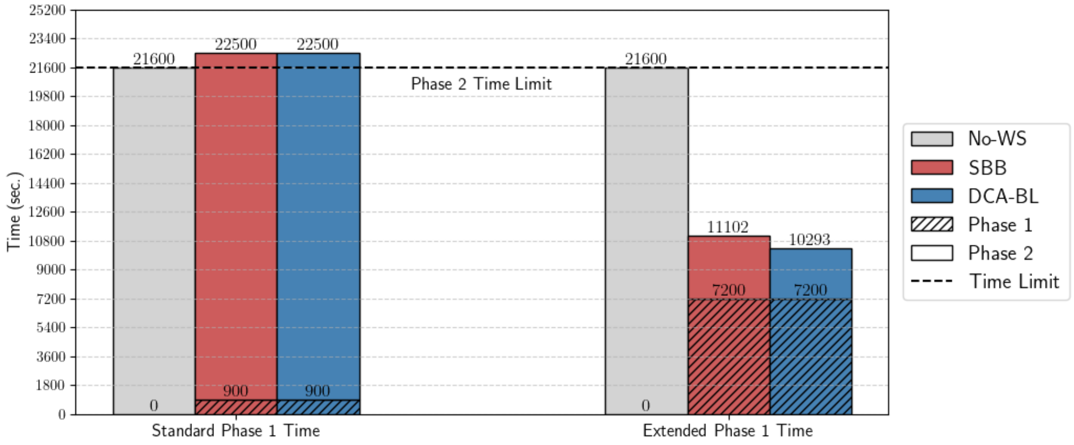

FIGURE 3. Computational time with standard (15 minutes) and extended (2 hours) Phase 1 time limits with different warm-start routines. Phase 1 is the warm-start routine and Phase 2 is the solving the LCP with PATH from the starting point computed in Phase 1.

When considering larger problems with size $n \geq 40,000$, DCA-BL provides the best percent improvement in average computational time: 54.2% (2.18 speed-up) compared to No-WS and 43.3% (1.53 speed-up) compared to SBB.

5.4. **Impact of Extended Warm Start.** Although the DCA-BL and SBB warm starts result in faster and less variable Phase 2 solution times, PATH was still unable to solve some of the larger instances within the 6-hour time limit even with the warm start applied. Therefore, to analyze the impact of a longer warm-start period, we extend the Phase 1 time for the largest instance with 92,919 variables. Specifically, we extend the time limit of Phase 1 to 2 hours, compared to 15 minutes used previously, allowing SBB and DCA-BL more time to compute a warm-start solution. As shown in Figure 3, the resulting PATH (Phase 2) runtime is 3093.2 seconds (0.85 hours) and 3902.5 seconds (1.1 hours) using the DCA-BL and SBB warm starts, respectively, compared to the baseline (No-WS) which timed out after 6 hours. As expected, this suggests that the warm-start approaches are able to compute a more accurate approximation during Phase 1 when more time is allowed, resulting in a higher-quality starting point. Moreover, as noted in Section 5.3, the DCA-BL warm start leads to a slightly faster Phase 2 time compared to the SBB warm start, suggesting that the DC-based approach is a more effective warm-start strategy.

6. CONCLUSIONS

We present an effective warm-starting scheme for generating high-quality starting points for large linear complementarity problems arising from Nash equilibrium problems. The warm-start strategy yields faster and more consistent solution times with the PATH solver. The proposed warm-start quadratically constrained linear program leverages a strong duality formulation of each agent's LP, yielding a master problem with bilinear constraints that is equivalent to the original LCP. To approximate solutions to this nonconvex bilinear program, we use two methods: (1) Gurobi's spatial branch-and-bound algorithm (SBB), and (2) the DCA-BL approach from [12]. Contrary to conventional bilinear approximation schemes, e.g., McCormick envelopes, a benefit of the DCA-BL warm start is that it does not explicitly rely on tight variable bounds. This benefit is particularly amplified in the Nash equilibria setting, where tight bounds on agents' dual variables may not be easily obtained.

We test the warm-start schemes on a realistic LCP instance with varying size, derived from a stochastic natural gas equilibrium model with nearly 100,000 variables. Experimentation shows that without warm

by the ratio t_0/t_s.

starting PATH struggles significantly in solving large-scale, realistic LCP instances, even with a 24-hour time limit. Using approximate solutions from these warm-start routines as starting points, the PATH solver performance improves substantially, particularly for large instances. In particular, both the variance and the average computational time of PATH decrease significantly (as the problem size increases), leading to a more efficient, consistent, and reliable solution process. Using the proposed DCA-BL warm-start scheme, we are able to solve the largest LCP instance of nearly 100,000 variables in about 1 hour with 2 hours of presolve time, compared to more than 24 hours with no warm start applied. While both warm starts yield improved PATH performance, experimentation suggests that DCA-BL produces a better starting point than SBB, as measured by the resulting PATH runtime and distance to the true solution. While our warm-start strategy is demonstrated on a specific LCP, it is broadly applicable to any LCP formed from the KKT conditions of agent LPs combined with linear system-level constraints.

References

[1] Agdeppa, R. P., Yamashita, N., and Fukushima, M. “The traffic equilibrium problem with nonadditive costs and its monotone mixed complementarity problem formulation.” In: *Transportation Research Part B: Methodological* 41.8 (2007), pp. 862–874.

[2] Bahrami, S., Aashtiani, H. Z., Nourinejad, M., and Roorda, M. J. “A complementarity equilibrium model for electric vehicles with charging.” In: *International Journal of Transportation Science and Technology* 6.4 (2017), pp. 255–271.

[3] Bertsimas, D. and Tsitsiklis, J. N. *Introduction to linear optimization.* Vol. 6. Athena scientific Belmont, MA, 1997.

[4] Birge, J. R. and Louveaux, F. *Introduction to stochastic programming.* Springer Science & Business Media, 2011.

[5] Boyd, N. T. and Gabriel, S. A. “Understanding the role of technological complexity in sustainability transitions using stochastic, bi-level optimization.” In: *Computational Management Science* 22.1 (2025), pp. 1–36.

[6] Boyd, N. T., Gabriel, S. A., Rest, G., and Dumm, T. “Generalized Nash equilibrium models for asymmetric, non-cooperative games on line graphs: Application to water resource systems.” In: *Computers & Operations Research* 154 (2023), p. 106194.

[7] Bynum, M. L., Hackebeil, G. A., Hart, W. E., Laird, C. D., Nicholson, B. L., Siirola, J. D., Watson, J.-P., and Woodruff, D. L. *Pyomo–optimization modeling in python.* Third. Vol. 67. Springer Science & Business Media, 2021.

[8] Conejo, A. J. and Ruiz, C. “Complementarity, not optimization, is the language of markets.” In: *IEEE Open Access Journal of Power and Energy* 7 (2020), pp. 344–353.

[9] Dinh Tao, P. and Le An, T. H. “Convex Analysis Approach to D.C. Programming: Theory, Algorithms and Applications.” In: *Acta Mathematica Vietnamica* 22.1 (1997), pp. 289–355.

[10] Ferris, M. C. and Munson, T. S. “Interfaces to PATH 3.0: Design, Implementation and Usage.” In: *Computational Optimization and Applications* 12.1 (1999), pp. 207–227.

[11] Gabriel, S. A., Conejo, A. J., Fuller, J. D., Hobbs, B. F., and Ruiz, C. *Complementarity modeling in energy markets.* Vol. 180. Springer Science & Business Media, 2012.

[12] Gabriel, S. A., Flocco, D., Boomsma, T. K., Schmidt, M., and Lejeune, M. A. “A Heuristic for Complementarity Problems Using Difference of Convex Functions.” In: *INFORMS Journal on Computing* (2025).

[13] Gabriel, S. A., Flocco, D. C., Mazzini, F. F., Sotelo, D., Castor, K. d. S., and Levorato, M. “Theoretical results for gas market equilibrium modeling with application to Brazil.” In: *Computational Management Science* 22.1 (2025), p. 2.

[14] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual.* 2024.

[15] Hart, W. E., Watson, J.-P., and Woodruff, D. L. "Pyomo: modeling and solving mathematical programs in Python." In: *Mathematical Programming Computation* 3.3 (2011), pp. 219–260.

[16] Hobbs, B. "Linear complementarity models of Nash-Cournot competition in bilateral and POOLCO power markets." In: *IEEE Transactions on power systems* 16.2 (2001), pp. 194–202.

[17] Le Thi, H. A., Huynh, V. N., and Dinh, T. P. "DC Programming and DCA for General DC Programs." In: *Advances in Intelligent Systems and Computing*. Springer International Publishing, 2014, pp. 15–35.

[18] Le Thi, H. A., Huynh, V. N., and Dinh, T. P. "DC programming and DCA for general DC programs." In: *Advanced Computational Methods for Knowledge Engineering: Proceedings of the 2nd International Conference on Computer Science, Applied Mathematics and Applications (ICCSAMA 2014)*. Springer. 2014, pp. 15–35.

[19] McCormick, G. P. "Computability of global solutions to factorable nonconvex programs: Part I convex underestimating problems." In: *Mathematical Programming* 10.1 (1976), pp. 147–175.

[20] Solodov, M. V. "On the Sequential Quadratically Constrained Quadratic Programming Methods." In: *Mathematics of Operations Research* 29 (2004), pp. 64–79.

[21] Zhuang, J. and Gabriel, S. A. "A complementarity model for solving stochastic natural gas market equilibria." In: *Energy Economics* 30.1 (2008), pp. 113–147.

Appendix A. DCA-BL Pseudocode

Here, we restate the DCA-BL algorithm from the original work [12], which applies the GDCA2 framework from Le Thi et al. [18]. For the proposed DCA-BL warm start scheme, Algorithm 1 is used to solve the QCLP (16).

Algorithm 1 DCA for Bilinear Terms (DCA-BL) [12]

```
1: Initialization. Find an initial point $z^1$. Choose parameters $\delta_1, \delta_2 > 0$, an initial penalty parameter
   value $\rho_1 > 0$, and set $k = 1$.
2: Subproblem. Solve the convex subproblem (24) to obtain an optimal solution $(z^{k+1}, t^{k+1})$ and
   corresponding Lagrange multipliers $\lambda^{k+1} \in \mathbb{R}^{n+1}$.
3: Convergence check. If $z^{k+1} = z^k$ and $t^{k+1} = 0$, then STOP and return $z^{k+1}$ as an approximate
   solution to (16). Otherwise, go to Step 4.
4: Penalty parameter update. Let
       $$r_k := \min\left\{||z^{k+1} - z^k||_2^{-1}, ||\lambda^{k+1}||_1 + \delta_1\right\},$$
   where $\|\cdot\|_1$ is the $\ell_1$-norm and $\|\cdot\|_2$ is the $\ell_2$-norm. Set
       $$\rho_{k+1} = \begin{cases} \rho_k, & \text{if } \rho_k \geq r_k, \\ \rho_k + \delta_2, & \text{if } \rho_k < r_k. \end{cases}$$
5: Iterate. Set $k \leftarrow k + 1$ and go to Step 2.
```

Appendix B. LCP System Structure Example

In this section, we outline the overall structure of the LCP for the general case. To clarify this structure, we first derive the LCP structure for the energy market equilibrium example introduced in Section 2.1.

B.1. **Example of LCP Matrix Structure.** Consider the LCP defined by the concatenation of KKT conditions (9) plus market-clearing constraint (10) with two players $P = \{1, 2\}$ and two time periods $T = \{1, 2\}$. The objective coefficients for player p's objective are

$$c^p(x^{-p}, \nu) = \begin{pmatrix} \gamma_1^p + \tau^{in} \\ \gamma_2^p + \tau^{in} \end{pmatrix} + \begin{pmatrix} -\pi_1 \\ -\pi_2 \end{pmatrix},$$

which we write as the sum of two vectors to separate problem data from prices π_1, π_2, which are exogenous to player p but endogenous to the overall LCP. These market prices play the role of the system-level multiplier ν in this example. The constraints (8b) and (8c) can be expressed in matrix-vector form consistent with (1) as:

$$A^p = \begin{pmatrix} -1 & -1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \text{and} \quad b^p(x^{-p}, \nu) = \begin{pmatrix} -X^{p,\text{total}} \\ -X_1^{p,rate} \\ -X_2^{p,rate} \end{pmatrix}.$$

Putting this together, the related KKT conditions for player p includes the two primal variables x_1^p, x_2^p and the three Lagrange multipliers $\lambda_1^p, \lambda_2^p, \lambda_3^p$ for the 3 constraints (not including nonnegativity).

Together, the LCP (7) in 2 players and 2 time periods involves 12 variables: $x_1^1, x_2^1, \lambda_1^1, \lambda_2^1$ and λ_3^1 for player 1; $x_1^2, x_2^2, \lambda_1^2, \lambda_2^2$ and λ_3^2 for player 2; and, prices π_1 and π_2, the latter two constituting the vector ν. Thus, the vector $z \in \mathbb{R}^{12}$ is then given as:

$$z = \left(x_1^1, x_2^1, \lambda_1^1, \lambda_2^1, \lambda_3^1, x_1^2, x_2^2, \lambda_1^2, \lambda_2^2, \lambda_3^2, \pi_1, \pi_2\right)^\top$$

In this case, the complementarity dimension of the LCP is 12, and the linear system $0 \leq Mz + q$ in (7) is expressed as:

$$0 \leq \underbrace{\left(\begin{array}{ccccc|ccccc|cc} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right)}_{M} \underbrace{\left(\begin{array}{c} x_1^1 \\ x_2^1 \\ \lambda^{1,\text{total}} \\ \lambda_1^1 \\ \lambda_2^1 \\ \hline x_1^2 \\ x_2^2 \\ \lambda^{1,\text{total}} \\ \lambda_1^2 \\ \lambda_2^2 \\ \hline \pi_1 \\ \pi_2 \end{array}\right)}_{z} + \underbrace{\left(\begin{array}{c} \gamma_1^1 + \tau^{in} \\ \gamma_2^1 + \tau^{in} \\ X_1^{1,\text{total}} \\ X_1^{1,\text{rate}} \\ X_2^{2,\text{rate}} \\ \hline \gamma_1^2 + \tau^{in} \\ \gamma_2^2 + \tau^{in} \\ X_1^{2,\text{total}} \\ X_1^{2,\text{rate}} \\ X_2^{2,\text{rate}} \\ \hline -D_1 \\ -D_2 \end{array}\right)}_{q}$$

The resulting formulation is an LCP that is block separable by player, with linking variables and constraints induced by the system-level complementarity conditions. In [13], the authors show that this structure extends naturally to an arbitrary number of time periods and players, and under various assumptions on price formation (e.g., price-taker and price-maker settings). Notably, in this example the LCP matrix M is positive semidefinite, a property with important implications for the existence and uniqueness of Nash equilibria in the overall system, which are also examined in [13]. Next, we extend this example to derive the LCP structure of the general equilibrium model considered in the manuscript.

B.2. **Generalizing the LCP Structure.** To generalize the LCP structure, we first re-express player p's objective in (1) to be more concrete. Specifically, define the optimization problem for player p as

$$\begin{aligned}
\underset{x^p}{\text{minimize}} \quad & (x^p)^\top u^p + (x^p)^\top G^p \nu + (x^p)^\top H^p x^{-p} && \text{(26a)}\\
\text{s.t.} \quad & A^p x^p \geq e^p + Q^p \nu + R^p x^{-p} \qquad (\lambda^p) && \text{(26b)}\\
& x^p \geq 0, && \text{(26c)}
\end{aligned}$$

where $u^p \in \mathbb{R}^{N_p}$ and $e^p \in \mathbb{R}^{M_p}$ are vectors, $G^p \in \mathbb{R}^{N_p \times \ell}$ and $Q^p \in \mathbb{R}^{M_p \times \ell}$ are matrices representing the coefficients of system-level multipliers ν, and $H^p \in \mathbb{R}^{N_p \times \tilde{N}_p}$ and $R^p \in \mathbb{R}^{M_p \times \tilde{M}_p}$ are matrices representing the coefficients of other players' decisions x^{-p}, with column dimension $\tilde{N}_p = \sum_{\bar{p} \neq p} N_{\bar{p}}$ and $\tilde{M}_p = \sum_{\bar{p} \neq p} M_{\bar{p}}$. In the context of (1), we have re-expressed the vectors $c^p(x^{-p}, \nu)$ and $b^p(x^{-p}, \nu)$ to be more concrete. Specifically, we express them as three separate affine maps to isolate exogenous variables in p's optimization problem:

$$c^p(x^{-p}, \nu) = u^p + G^p \nu + H^p x^{-p} \quad \text{and} \quad b^p(x^{-p}, \nu) = e^p + Q^p \nu + R^p x^{-p}$$

In the context of the example in Appendix B.1, the components of player p's objective are:

$$u^p = \begin{pmatrix} \gamma_1^p + \tau^{in} \\ \gamma_2^p + \tau^{in} \end{pmatrix}, \quad G^p \nu = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \pi_1 \\ \pi_2 \end{pmatrix} = \begin{pmatrix} -\pi_1 \\ -\pi_2 \end{pmatrix}, \quad \text{and} \quad H^p x^{-p} = 0.$$

Note that $H^p = 0$ since other player's decision variables x^{-p} do not appear in player p's objective for this example. Also, the prices π_1 and π_2 play the role of system-level constraint multipliers ν here. The components of the right-hand-side vector are:

$$e^p = \begin{pmatrix} -X^{p,\text{total}} \\ -X_1^{p,rate} \\ -X_2^{p,rate} \end{pmatrix}, \quad \text{and} \quad Q^p = R^p = 0.$$

Here, $Q^p = R^p = 0$ since neither system-level multipliers ν nor other players' decisions x^{-p} appear in the right-hand-side of player p's constraints.

The necessary and sufficient KKT conditions for player p's optimization problem are the following: find vectors (x^p, λ^p) such that

$$\begin{aligned}
0 \leq u^p + G^p \nu + H^p x^{-p} - (A^p)^\top \lambda^p \quad &\perp \quad x^p \geq 0 && \text{(27a)}\\
0 \leq A^p x^p - e^p - Q^p \nu - R^p x^{-p} \quad &\perp \quad \lambda^p \geq 0 && \text{(27b)}
\end{aligned}$$

which can be written more succinctly as:

$$0 \leq \begin{pmatrix} 0 & -(A^p)^\top \\ A^p & 0 \end{pmatrix} \begin{pmatrix} x^p \\ \lambda^p \end{pmatrix} + \begin{pmatrix} u^p + G^p \nu + H^p x^{-p} \\ -e^p - Q^p \nu - R^p x^{-p} \end{pmatrix} \perp \begin{pmatrix} x^p \\ \lambda^p \end{pmatrix} \geq 0. \tag{28}$$

Concatenating the KKT conditions (27) for each each player $p \in P$, in addition to system-level constraints of the form (4), results in a linear complementarity problem (LCP) of the form

$$0 \leq Mz + q \qquad \perp \qquad z \geq 0, \tag{29}$$

where $z = \{(x^p, \lambda^p)_{p \in P}, \nu\}$ are the decision variables, comprised of player primal and dual decisions x^p and λ^p, and system-level multipliers ν. Also, q is a vector and M the LCP coefficient matrix. In this general setting, the linear system $Mz + q$ has the form

$$\begin{bmatrix} \begin{pmatrix} 0 & -(A^1)^\top \\ A^1 & 0 \end{pmatrix} & \begin{pmatrix} H_2^1 & 0 \\ -R_2^1 & 0 \end{pmatrix} & \begin{pmatrix} H_3^1 & 0 \\ -R_3^1 & 0 \end{pmatrix} & \cdots & \begin{pmatrix} H_{n_p}^1 & 0 \\ -R_{n_p}^1 & 0 \end{pmatrix} & \begin{pmatrix} G^1 \\ -Q^1 \end{pmatrix} \\ \begin{pmatrix} H_1^2 & 0 \\ -R_1^2 & 0 \end{pmatrix} & \begin{pmatrix} 0 & -(A^2)^\top \\ A^2 & 0 \end{pmatrix} & \begin{pmatrix} H_3^2 & 0 \\ -R_3^2 & 0 \end{pmatrix} & \cdots & \begin{pmatrix} H_{n_p}^2 & 0 \\ -R_{n_p}^2 & 0 \end{pmatrix} & \begin{pmatrix} G^2 \\ -Q^2 \end{pmatrix} \\ \vdots & & & \ddots & & \vdots \\ \begin{pmatrix} H_1^{n_p} & 0 \\ -R_1^{n_p} & 0 \end{pmatrix} & \begin{pmatrix} H_2^{n_p} & 0 \\ -R_2^{n_p} & 0 \end{pmatrix} & \begin{pmatrix} H_3^{n_p} & 0 \\ -R_3^{n_p} & 0 \end{pmatrix} & \cdots & \begin{pmatrix} 0 & -(A^{n_p})^\top \\ A^{n_p} & 0 \end{pmatrix} & \begin{pmatrix} G^{n_p} \\ -Q^{n_p} \end{pmatrix} \\ \begin{pmatrix} K^1 & 0 \end{pmatrix} & \begin{pmatrix} K^2 & 0 \end{pmatrix} & \begin{pmatrix} K^3 & 0 \end{pmatrix} & \cdots & \begin{pmatrix} K^{n_p} & 0 \end{pmatrix} & 0 \end{bmatrix} \begin{bmatrix} \begin{pmatrix} x^1 \\ \lambda^1 \end{pmatrix} \\ \begin{pmatrix} x^2 \\ \lambda^2 \end{pmatrix} \\ \begin{pmatrix} x^3 \\ \lambda^3 \end{pmatrix} \\ \vdots \\ \begin{pmatrix} x^{n_p} \\ \lambda^{n_p} \end{pmatrix} \\ \nu \end{bmatrix} + \begin{bmatrix} \begin{pmatrix} u^1 \\ -e^1 \end{pmatrix} \\ \begin{pmatrix} u^2 \\ -e^2 \end{pmatrix} \\ \begin{pmatrix} u^3 \\ -e^3 \end{pmatrix} \\ \vdots \\ \begin{pmatrix} u^{n_p} \\ -e^{n_p} \end{pmatrix} \\ d \end{bmatrix} \tag{30}$$

Here, we adopt the convention that capital letters denote the concatenation of vectors corresponding to the vectors or functions denoted by lowercase letters. First, K^p is a $\ell \times N_p$ matrix whose ith row is $(k_i^p)^\top$ for all $p \in P$ and $d = (d_1, \ldots, d_\ell)^\top$. Specifically, in the case of the general system-level constraint (4), entries of the matrix K^p are given by:

$$K^p = \begin{bmatrix} (k_1^p)^\top \\ \vdots \\ (k_\ell^p)^\top \end{bmatrix}. \tag{31}$$

For the example in Appendix B.1, whose system-level constraints are given by the market-clearing conditions (10), $K^p = I_{n_t}$. Similarly, as shown above, $G^p = -I_{n_t}$ for this example, where $n_t = 2$ is the number of time periods. The matrices $H_{\bar{p}}^p \in \mathbb{R}^{N_{\bar{p}} \times N_p}$ and $R_{\bar{p}}^p \in \mathbb{R}^{M_{\bar{p}} \times M_p}$ are blocks of larger matrices H^p and R^p, respectively. Each block captures the coefficients associated with a *single* other player's variables $\bar{p}$ in player p's optimization problem. Accordingly, the full matrices appearing in player p's optimization problem (26) may be written as $H^p = \operatorname{diag}(H_{\bar{p}}^p : \bar{p} \neq p)$ and $R^p = \operatorname{diag}(R_{\bar{p}}^p : \bar{p} \neq p)$. As noted earlier, $H_{\bar{p}}^p = R_{\bar{p}}^p = 0$ for all $p, \bar{p} \in P$ in the example in Appendix B.1.

(D. C. Flocco) University of Maryland, Department of Mathematics, College Park, Maryland 20742, USA
Email address: dflocco@umd.edu

(S. A. Gabriel) (a) University of Maryland, College Park, Maryland 20742, USA, (b) Norwegian University of Science and Technology, Trondheim, Norway, (c) Aalto University, Espoo, Finland
Email address: sgabriel@umd.edu