Les Cahiers du GERAD ISSN: 0711–2440

Linear-Quadratic Games Played Over Event Trees

P.V. Reddy
G. Zaccour

G–2013–37

May 2013

Linear-Quadratic Games Played Over Event Trees

Puduru Viswanadha Reddy
Georges Zaccour

GERAD & HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada, H3T 2A7

puduru.reddy@gerad.ca
georges.zaccour@gerad.ca

May 2013

Les Cahiers du GERAD

G–2013–37

Abstract: In this paper, we study N-player finite-horizon discrete-time dynamic stochastic games where the uncertainty is described by an event tree. We consider linear-state dynamics, with a one-period lag structure, and quadratic costs. We derive necessary and sufficient conditions for the existence of S-adapted Nash equilibria with open-loop and closed-loop (no-memory) information structures. We observe that the existence of these equilibria is related to the solvability of a generalized backward Riccati recursion defined on the event tree. Next, we consider these games with (node-specific) linear constraints. We show that the S-adapted Nash equilibria (both open-loop and closed-loop) can be obtained by solving a parametric linear-complementarity problem defined on the entire event tree.

Key Words: Linear-quadratic dynamic games; Event tree; Stochastic games; Open-loop Nash equilibria; Closed-loop Nash equilibria.

Résumé : On étudie dans cet article un jeu stochastique et dynamique à temps discret à N joueurs et où l'incertitude est décrite par un arbre d'événements. On retient une dynamique linéaire dans l'état impliquant un retard d'une période et des coûts quadratiques. On dérive les conditions nécessaires et suffisantes d'existence d'un équilibre de Nash en boucle ouverte et en boucle fermée. On trouve que l'existence de ces équilibres est reliée à la solvabilité d'équations de Riccati définies sur l'arbre d'événements. On analyse aussi le cas où les joueurs font face à des contraintes linéaires. On montre que les équilibres en boucle ouverte et en boucle fermée peuvent être obtenus en résolvant un problème de complémentarité linéaire défini sur l'arbre au complet.

Mots clés : jeux dynamiques linéaires-quadratiques; arbre d'événements; jeu stochastiques; équilibres de Nash en boucle ouverte; équilibres de Nash en boucle fermée.

1 Introduction

Many decision problems in operations research and management science, economics and engineering share the following features: (i) They involve few decision makers or players (firms, countries, etc.) having interdependent payoffs, i.e., the action of one player affects the rewards of the others. (ii) They are inherently dynamic, i.e., the players compete repeatedly over time and their actions have an impact on the evolution of the state of the system. (iii) They involve some uncertain parameters. And, (iv) They involve lags in the realization of actions. An illustrative example is a set of power utilities competing in a given geographical market; the state variables are the available production capacities (of different technologies) and the control variables are the quantities put on the market at each period, as well as the investment in production capacities; the uncertainty can be in the demand or in the cost parameters, or both. A natural methodological framework to deal with this family of problems is the theory of (discrete) dynamic games played over uncontrolled event trees, that is, games where the transition from one node to another is an act of nature and cannot be influenced by the players' actions. This class of games was initially introduced in Zaccour (1987) and Haurie et al. (1990) to study noncooperative equilibria in the European natural gas market, which is characterized by the presence of four suppliers competing over a long-term planning horizon in nine markets described by stochastic demand laws. The solution concept was termed an S-adapted equilibrium, where the S stands for *sample* of realizations of the random process (see Haurie et al. (2012) for details). Gürken et al. (1999) and Haurie and Moresino (2002) showed that this concept is related to the concept of stochastic variational inequality. Pineau and Murto (2003), Genc et al. (2007), Genc and Sen (2008) and Pineau et al. (2011) modeled different energy markets as dynamic games played over an event tree and computed the resulting S-adapted equilibria. Recently, Reddy et al. (2013) assumed that players can cooperate and proposed a time-consistent Shapley value to share the proceeds of cooperation among the participating players. In all these papers, the information structure is open loop, that is, decisions are a function of time and of the initial conditions of the state variables.

The optimization of dynamic economic systems for the single agent case, in the presence of uncertainty, has been traditionally approached using stochastic dynamic programming and stochastic programming. In the former approach one solves the problem by looking for a fixed point of an operator defined in the space of value functions whereas stochastic programming uses mathematical programming techniques. Stochastic programming approach can be implemented when the agent actions do not influence the probability measure of the random process describing the uncertainty. It has an advantage over the dynamic programming approach in terms of complexity bounds, as it theoretically permits the solution of convex problems in polynomial time, at least when the uncertainty is represented as an event tree.

In this paper we study a class of games where the random disturbance process has a probability measure that is not affected by players actions. Using this framework, we aim at extending the stochastic programming approach to a dynamic game context. More specifically, we study a framework for linear-quadratic games played over event trees, that is, games where each player's objective is quadratic in the state and control variables, where the dynamics are described by linear functions of the state variables, where the uncertainty is described by a scenario tree, and which also account for linear inequalities that jointly involve state and control variables. Deterministic linear-quadratic dynamic games have a very long tradition of application, in both continuous and discrete time. The main reasons for their popularity lies in the availability of theorems characterizing the existence and uniqueness of equilibria, and in their tractability. For a complete coverage of the theory of linear-quadratic games, see Başar and Olsder (1998), Dockner et al. (2000) and Engwerda (2005). A comprehensive survey of linear-quadratic differential games in economics and management science, covering papers published during the period from 2000 to 2005, is provided in Jørgensen and Zaccour (2006).

We believe that the extension of deterministic linear-quadratic games to a stochastic programming setting is greatly needed to deal with realistic decision problems characterized by the above-stated features. More specifically, we characterize S-adapted Nash equilibria for this class of games under open-loop and closed-loop (no-memory) information structures. To the best of our knowledge, closed-loop equilibria for these games have not yet been studied in the literature. The focus on linear-quadratic structure is motivated by the computational possibilities, and also by setting up a general framework for large-scale finite dimensional linear-quadratic game problems on the event tree that reflect a close resemblance to linear-quadratic difference games and stochastic equilibrium programming. In addition to studying multistage (sequential) games under uncertainty, we provide a formulation that makes it possible to model lags between the time of decision and its realization, commonly known as time-to-build. Besides these features, the algorithms developed in this paper can serve as a tool to compare the performance of open-loop and closed-loop equilibria for these games, called price of information (see Başar and Zhu (2011)).

The rest of this paper is organized as follows: Section 2 introduces the game model, and Section 3, the S-adapted open-loop and closed-loop (no memory) Nash equilibria in the unconstrained case. We observe that due to the one-period lag structure in the dynamics, the computation of Nash equilibria involves solving backward recursive Riccati-type equations defined on the the event tree. Section 4 extends these results to the setting where the players face linear constraints jointly in the state and control variables. Here, we restrict the information structure as constrained open-loop and closed-loop structures. We observe that the necessary conditions lead to a weakly coupled system of backward equations defined on the event tree and node-specific parametrized linear-complementarity problems, which after a suitable reformulation, allows for computing the Nash equilibria, by solving a linear-complementarity problem defined on the entire event tree. We illustrate the implementation of the theory in two examples in Sections 5.1 and 5.2, and briefly conclude in Section 6.

Notation: In the sequel, we shall use the following notation. M' denotes the transpose of a matrix $M \in \mathbb{R}^{n\times n}$. $\texttt{Col}(A_1, A_2)$ represents the column vector/matrix obtained by appending rows of A_2 to A_1. $\texttt{Col}(A_k)_{k=1}^n$ represents the column matrix/vector obtained by appending the rows of $A_1,\ A_2,\ \cdots, A_n$ in this sequence. Similarly, $\texttt{Row}(A_k)_{k=1}^n$ represents the row matrix/vector obtained by appending the rows of $A_1,\ A_2,\ \cdots, A_n$ in this sequence. $\texttt{BDmat}(A_1, A_2)$ represents the block diagonal matrix obtained by taking matrices A_1 and A_2 as the diagonal elements. $\texttt{BDmat}(A_k)_{k=1}^n$ represents the block diagonal matrix obtained by taking the matrices $A_1,\ A_2, \cdots, A_n$ as diagonal elements in this sequence. $A_1 \otimes A_2$ represents the Kronecker product of the two vectors/matrices A_1 and A_2. The null matrix is represented by 0 or $\mathbf{0}$. The identity matrix is represented by I or $\mathbf{I}$. We suppress writing the dimensions of matrices and vectors and assume that they are defined appropriately depending upon the context.

2 Preliminaries

In this section we introduce the multistage game model. Firstly, we briefly describe the event tree which captures the underlying model of uncertainty. Let $\mathcal{T} = \{0, 1, \ldots, T\}$ be the set of periods. Denote by $\{\xi(t) : t \in \mathcal{T}\backslash T\}$ the exogenous stochastic process represented by an event tree. This tree has a root node n_0 in period 0 and has a finite set of nodes $\mathbf{n}^t$ in period $t \in \mathcal{T}$. Each node $n^t \in \mathbf{n}^t$ represents a possible sample value of the history of the $\xi(\cdot)$ process up to time t. The tree graph structure represents the nesting of information as one one-time period succeeds another. We denote by $a(n^t) \in \mathbf{n}^{t-1}$ the unique predecessor of node $n^t \in \mathbf{n}^t$, and by ν a successor of the node n^t. We denote by $\mathcal{S}(n^t) \in \mathbf{n}^{t+1}$ the set of all possible direct successors of node n^t. We illustrate a branch of the event tree in Figure 1. A path from the root node n_0 to a terminal node n^T is called a *scenario*. Each scenario has a probability and the probabilities of all scenarios sum up to 1. We denote by $\pi(n^t)$ the probability of passing through node n^t, which corresponds to the sum of the probabilities of all scenarios that contain this node. In particular, $\pi(n_0) = 1$ and $\pi(n^T)$ is equal to the probability of the single scenario that terminates in (leaf) node $n^T \in \mathbf{n}^T$. We denote by $\pi_{n^t}^{\nu}$ the transition probability from node n^t to a particular node $\nu \in \mathcal{S}(n^t)$. We denote by $\boldsymbol{\pi}_{n^t}^{\nu}$ the row vector of transition probabilities, that is,

$$\boldsymbol{\pi}_{n^t}^{\mathcal{S}(n^t)} = \begin{bmatrix} \pi_{n^t}^{\nu^1} & \pi_{n^t}^{\nu^2} & \cdots & \pi_{n^t}^{\nu^{|\mathcal{S}(n^t)|}} \end{bmatrix},$$

where $\nu_1, \nu_2, \cdots, \nu_{|\mathcal{S}(n^t)|}$ are the successors of node n^t.

Next, denote by $\bar{N} = \{1, 2, \ldots, N\}$ the set of players. Let $x(n^t) \in X \subset \mathbb{R}^n$, with n a given positive integer, be a state vector at node n^t. The control/action/decision sets of player j at node n^t, $U_j^{n^t}$ are taken as measurable subsets of $\mathbb{R}^{m_j^{n^t}}$, with $m_j^{n^t}$ being a given positive integer. Denote by $u_j(n^t) \in U_j^{n^t}$ the decision

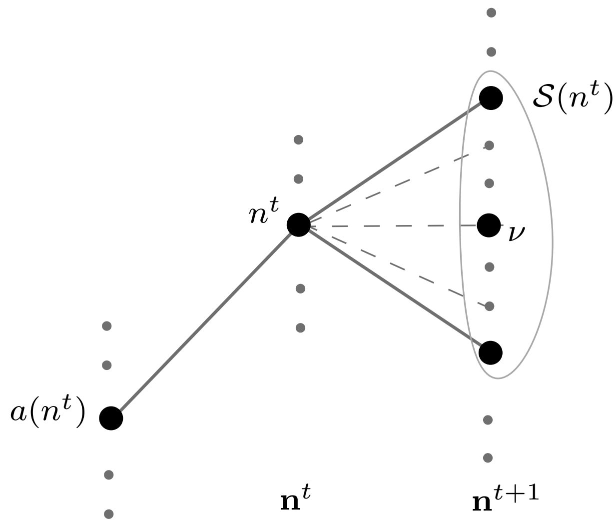

Figure 1: Branch of the event tree

variable of player j at node $n^t \in \mathbf{n}^t$. The state equation defined over the event tree is given as follows:

$$x(n^t) = A(a(n^t))x(a(n^t)) + \sum_{l \in \bar{N}} B_l(a(n^t))u_l(a(n^t)), \ n^t \in \mathbf{n}^t, \ t \in \mathcal{T} \backslash 0, \ x(n^0) = x_0 \in X, \tag{1}$$

where $A(n^t) \in \mathbb{R}^{n \times n}$ and $B_i(n^t) \in \mathbb{R}^{n \times m_j^{n^t}}$. According to the state equation (1), at each node $n^t \in \mathbf{n}^t$, the players, using the actions $u_i(n^t), \ i \in \bar{N}$, influence the state $x(\nu)$ for all successor nodes $\nu \in \mathcal{S}(n^t)$. The state variable at node n^t in (1) depends on the state variable and the controls chosen at the ancestor node $a(n^t)$. Here, the decisions are taken by players before the realization of uncertainty. So, (1) reflects a one-period lag[1] in the dynamics, and as a result, we have $x(\nu_1) = x(\nu_2), \ \forall \nu_1, \nu_2 \in \mathcal{S}(n^t)$ such that $\nu_1 \neq \nu_2$.

We assume that each player i is endowed with an additional decision variable $v_i(n^t) \in V_i^{n^t}$ at each node of the event tree. The feasible action sets $V_i^{n^t}$ are measurable subsets of $\mathbb{R}_+^{s_i^{n^t}}$, with $s_i^{n^t}$ a given positive integer. These additional decision variables allow players to influence the game after the realization of uncertainty. We assume that the decision variables $v_i(n^t)$ do not enter the dynamics directly[2] but appear only in the form of linear constraints, jointly with the state variable, at every node n^t as follows

$$M_i(n^t)x(n^t) + N_i(n^t)v_i(n^t) + r_i(n^t) \geq 0, \ v_i(n^t) \geq 0, \ n^t \in \mathbf{n}^t, \ t \in \mathcal{T}, \ i \in \bar{N}, \tag{2}$$

where $M_i(n^t) \in \mathbb{R}^{c_i^{n^t} \times n}$, $N_i(n^t) \in \mathbb{R}^{c_i^{n^t} \times s_i^{n^t}}$ and $r_i(t) \in \mathbb{R}^{c_i^{n^t}}$. We denote by $\mathbf{U}^{n^t} = U_1^{n^t} \times \ldots \times U_j^{n^t} \ldots \times U_N^{n^t}$ and $\mathbf{V}^{n^t} = V_1^{n^t} \times V_2^{n^t} \times \cdots \times V_N^{n^t}$ the joint decision sets of the players. The joint actions of players, at every node n^t, are denoted by $\mathbf{u}(n^t) = (u_1(n^t), \ldots, u_N(n^t))$ and $\mathbf{v}(n^t) = (v_1(n^t), \cdots, v_N(n^t))$. We denote by $\tilde{u}_i = \{u_i(n^t), \ n^t \in \mathbf{n}^t, \ t \in \mathcal{T} \backslash T\}$ and by $\tilde{v}_i = \{v_i(n^t), \ n^t \in \mathbf{n}^t, \ t \in \mathcal{T}\}$ the strategies, that is, a complete specification of actions defined for every node of the event tree, for player i. Note that the decision variables are indexed over the set of nodes in the event tree, with each node being an exhaustive summary of the history of the $\xi(\cdot)$ process. Making the decision variables depend on the nodes in the event tree is therefore equivalent to saying that the decisions are adapted to the history of the $\xi(\cdot)$ process. Here, the players are permitted to adapt their decisions to the sample path of the stochastic process but are not observing what the other players do when time unfolds. So, the strategies of players are sample path adapted, denoted by S-adapted, and defined formally as follows.

[1] It is possible to model longer lags, but we restrict our attention to one period. We will see in Section 3 that this assumption facilitates the recursive formulations for the computation of Nash equilibria.

[2] As an illustrative example, the u variables can be investments in production capacities to be engaged before knowing the realization of the random process at the next period, and the v variables are the quantities put on the market at current node, and are subject to capacity constraints.

Definition 1 *An admissible S-adapted strategy for player i is defined by $\tilde{u}_i = \{u_i(n^t),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T\}$ and $\tilde{v}_i = \{v_i(n^t),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\}$. It defines the plan of actions adapted to the history of the random process $\xi(.)$ represented by the event tree.*

We denote the admissible strategy space for player i as $(\tilde{U}_i, \tilde{V}_i)$, the Cartesian product of the action spaces over all the nodes of the event tree. We denote by $(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})$ the joint admissible S-adapted strategies of players, where $\tilde{\mathbf{u}} = (\tilde{u}_1, \ldots, \tilde{u}_N)$ and $\tilde{\mathbf{v}} = (\tilde{v}_1, \ldots, \tilde{v}_N)$. In this state equation formalism, the players' decisions are the actions $u_i(n^t)$ and $v_i(n^t)$, which are chosen independently. The state variables $x(n^t)$ are determined once the actions have been chosen. The state variables are shared by all the players and they enter the definition of players' objective functions. In the following we define two types of games; one without constraints denoted by LQGET (linear-quadratic game played over event tree) and one with constraints denoted by Con-LQGET (constrained linear-quadratic game played over event tree). The rationale behind this distinction is as follows. The unconstrained game has a close resemblance to deterministic linear-quadratic difference games, see Basar and Olsder (1998), and results in special recursive formulations towards the computation of Nash equilibria. Next, using these results we shall compute the Nash equilibria of the constrained game.

The unconstrained game is defined as follows:

$$\begin{aligned}
&\textbf{LQGET} \quad \min_{\tilde{u}_i} J_i(x_0, \tilde{\mathbf{u}}) \\
&J_i(x_0, \tilde{\mathbf{u}}) = \sum_{t=0}^{T-1} \sum_{n^t \in \mathbf{n}^t} \pi(n^t) \Big(\frac{1}{2} x'(n^t) Q_i(n^t) x(n^t) + p_i'(n^t) x(n^t) + \frac{1}{2} \sum_{j \in \bar{N}} u_j'(n^t) R_{ij}(n^t) u_j(n^t) \Big) \\
&\qquad + \sum_{n^T \in \mathbf{n}^T} \pi(n^T) \Big(\frac{1}{2} x'(n^T) Q_i(n^T) x(n^T) + p'(n^T) x(n^T) \Big) \qquad (3) \\
&\text{subject to:} \quad x(n^t) = A(a(n^t)) x(a(n^t)) + \sum_{l \in \bar{N}} B_l(a(n^t)) u_l(a(n^t)),\ x(n^0) = x_0.
\end{aligned}$$

At each node $n^t,\ t \in \mathcal{T}\backslash T$, the cost to player i is a quadratic function of the state and of the controls of all players. At a terminal node n^T the cost to player i is a quadratic function of the terminal state $x(n^T)$. $Q_i(n^t) \in \mathbb{R}^{n \times n}$ is a symmetric matrix and $R_{ii}(n^t) \in \mathbb{R}^{m_i^{n^t} \times m_i^{n^t}}$ is a positive definite matrix, and $p_i(n^t) \in \mathbb{R}^n$ for all $n^t \in \mathbf{n}^t,\ t \in \mathcal{T}$. The optimizing behavior of a player i depends on the actions of players $\bar{N}\backslash i$ and the interaction environment is captured by the dynamics (1). Hence, the above LQGET model defines a (multistage) linear-quadratic game defined on event tree. In the next section, we derive necessary conditions for a particular outcome of LQGET when players play non cooperatively.

The constrained game is defined as follows:

$$\begin{aligned}
&\textbf{Con-LQGET} \quad \min_{\tilde{u}_i, \tilde{v}_i} J_i(x_0, \tilde{\mathbf{u}}, \tilde{\mathbf{v}}) \\
&Jc_i(x_0, \tilde{\mathbf{u}}, \tilde{\mathbf{v}}) = J_i(x_0, \tilde{\mathbf{u}}) + \sum_{t=0}^{T} \sum_{n^t \in \mathbf{n}^t} \frac{\pi(n^t)}{2} \Big(\mathbf{v}'(n^t) T^i(n^t) \mathbf{v}(n^t) + 2 t^{i\prime}(n^t) \mathbf{v}(n^t) + 2 x'(n^t) L^i(n^t) \mathbf{v}(n^t) \Big) \quad (4) \\
&\text{subject to:} \quad x(n^t) = A(a(n^t)) x(a(n^t)) + \sum_{l \in \bar{N}} B_l(a(n^t)) u_l(a(n^t)),\ x(n^0) = x_0 \\
&\qquad M_i(n^t) x(n^t) + N_i(n^t) v_i(n^t) + r_i(n^t) \geq 0,\ v_i(n^t) \geq 0,\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T},\ i \in \bar{N},
\end{aligned}$$

where $T^i(t) = T^{i\prime}(t) \in \mathbb{R}^{s^{n^t} \times s^{n^t}}$, $t^i(n^t) \in \mathbb{R}^{s^{n^t}}$, $L^i(n^t) \in \mathbb{R}^{n \times s^{n^t}}$ and $c_i^{n^t}$ is a positive integer. We assume $T^i(n^t)$ as symmetric and denote by $v_i'(n^t) T_{ij}^i(n^t) v_j(n^t)$ the cross term between controls of players i and j. We denote the remaining coefficients of $v_i(n^t)$ by $x'(n^t) L_i^i(n^t)$ and $t_i^i(n^t)$. The objective function in the above model does not include cross terms between the controls $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{v}}$, and hence it has a separable structure. We shall see in Section 4 that this assumption allows to take stock on the results obtained for the LQGET model.

Remark 1 *The above game models have a different structure compared to the discrete-time linear-quadratic stochastic game (see Section 6.7 of Başar and Olsder (1998)), where the uncertain parameter enters the state* *dynamics additively. The main difference is that the uncertainty lies in nature choosing the scenarios, and the evolution of state is indexed over these scenarios. In that sense, the above game models are qualitatively closer to problems studied in multistage stochastic programming (see Rockafellar and Wets (1990)), see also Remark 3.*

3 LQGET model

In this section we provide necessary conditions for an S-adapted equilibrium in the LQGET model. We have the following definition.

Definition 2 *An S-adpated equilibrium is an admissible S-adapted strategy* $\tilde{\mathbf{u}}^*$ *if the following relations hold:*

$$J_i(x_0,(\tilde{u}_i^*,\tilde{u}_{-i}^*)) \le J_i(x_0,(\tilde{u}_i,\tilde{u}_{-i}^*)),\ \forall \tilde{u}_i \in \tilde{U}_i,$$

for all $i \in \bar{N}$*, subject to* (1)*.*

We can formulate the necessary conditions for an S-adapted equilibrium by defining the Lagrangian for each player i as follows:

$$\begin{aligned}
&\mathcal{L}_i(\tilde{x},(u_i,\tilde{u}_{-i}^*),\tilde{\lambda}_i)\\
&= \frac{1}{2}\Big(x'(n^0)Q_i(n^0)x(n^0) + 2p_i'(n^0)x(n^0) + u_i{}'(n^0)R_{ii}(n^t)u_i(n^0) + \sum_{j\in\bar{N}\setminus i} u_j^{*\prime}(n^0)R_{ij}(n^0)u_j^*(n^0)\Big)\\
&+ \sum_{t=1}^{T-1}\sum_{n^t\in\mathbf{n}^t}\frac{\pi(n^t)}{2}\left(x'(n^t)Q_i(n^t)x(n^t) + 2p_i'(n^t)x(n^t) + u_i'(n^t)R_{ii}(n^t)u_i(n^t) + \sum_{l\in\bar{N}} u_j^{*\prime}(n^t)R_{ij}(n^t)u_j^*(n^t)\right)\\
&+ \sum_{n^T}\frac{\pi(n^T)}{2}\left(x'(n^T)Q_i(n^T)x(n^T) + 2p_i'(n^T)x(n^T)\right) + \lambda_i'(n^0)\left(x_0 - x(n^0)\right)\\
&+ \sum_{t=1}^{T}\sum_{n^t\in\mathbf{n}^t}\pi(n^t)\lambda_i'(n^t)\left(A(a(n^t))x(a(n^t)) + B_i(a(n^t))u_i(a(n^t)) + \sum_{j\in\bar{N}\setminus i} B_j(a(n^t))u_j^*(a(n^t)) - x(n^t)\right).
\end{aligned} \tag{5}$$

In the above expression, we have introduced, for each player i, a Lagrange multiplier $\lambda_i(n^t)$, also indexed over the set of nodes and having the same dimension as $x(n^t)$. We observe the following: Due to the one-period lag in the dynamics, the state variable at all successor nodes $\nu \in \mathcal{S}(n^t)$, of the node n^t, has the same value. The last expression in the above Lagrangian reflects this feature, that is, the costate variable evaluated at $\nu \in \mathcal{S}(n^t)$ is multiplied with terms evaluated at the ancestor node n^t. This allows us to aggregate the costate variables as a conditional sum denoted by $\lambda_i(\mathcal{S}(n^t)) = \sum_{\nu\in\mathcal{S}(n^t)}\pi_{n^t}^{\nu}\lambda_i(\nu)$ (from here on, we denote by $z(\mathcal{S}(n^t))$ the conditional sum $\sum_{\nu\in\mathcal{S}(n^t)}\pi_{n^t}^{\nu}z(\nu)$ evaluated at the successors ν of $\mathcal{S}(n^t)$.) This observation allows us to define, for each player $i \in \bar{N}$ and each node $n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\setminus T$ the following pre-Hamiltonian function:

$$\begin{aligned}
\tilde{\mathcal{H}}_i(x(n^t),u_i(n^t),u_{-i}^*(n^t),\lambda_i(\mathcal{S}(n^t))) &= \lambda_i'(\mathcal{S}(n^t))\Big(A(n^t)x(n^t) + B_i(n^t)u_i(n^t) + \sum_{j\in\bar{N}\setminus i} B_j(n^t)u_j^*(n^t)\Big)\\
&+ \frac{1}{2}\Big(x'(n^t)Q_i(n^t)x(n^t) + 2p_i'(n^t)x(n^t) + \sum_{j\in\bar{N}\setminus i} u_j^{*\prime}(n^t)R_{ij}(n^t)u_j^*(n^t)\Big),\\
&= \mathcal{H}_i(x(n^t),u_i(n^t),u_{-i}^*(n^t),\lambda_i(\mathcal{S}(n^t))) + p_i'(n^t)x(n^t).
\end{aligned} \tag{6}$$

The necessary condition for a strategy profile $\tilde{\mathbf{u}}^*$ to be an S-adapted equilibrium is given as follows:

Theorem 3.1 *Assume that* $\tilde{\mathbf{u}}^*$ *is an S-adapted equilibrium at* x_0*, generating the state trajectory* $\tilde{x}^*$ *over the event tree. Then there exists, for each player* i*, a costate trajectory* $\tilde{\lambda}_i$ *such that the following conditions hold for* $i \in \bar{N}$*:*

$$\frac{\partial \mathcal{H}_i}{\partial u_i(n^t)} = 0 \to u_i^*(n^t) = -R_{ii}^{-1}(n^t)B_i'(n^t)\lambda_i(\mathcal{S}(n^t)), \tag{7}$$

$$\lambda_i(n^t) = \frac{\partial \mathcal{H}_i(.)}{\partial x_i(n^t)} + p_i(n^t),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T, \tag{8}$$

$$\lambda_i(n^T) = Q_i(n^T)x^*(n^T) + p_i(n^T),\ n^T \in \mathbf{n}^T. \tag{9}$$

Proof. In the expression of the Lagrangian (5), using (6), we group together the terms that contain $x(n^t)$, to obtain

$$\mathcal{L}_i\left(\tilde{x},(\tilde{u}_i,\tilde{u}_{-i}^*),\tilde{\lambda}_i\right) = \sum_{t=0}^{T-1}\sum_{n^t\in\mathbf{n}^t}\pi(n^t)\Big(\mathcal{H}_i(x(n^t),u_i(n^t),u_{-i}^*(n^t),\lambda_i(\mathcal{S}(n^t))) + p_i(n^t)x(n^t) - \lambda_i'(n^t)x(n^t)\Big)$$
$$+\sum_{n^T\in\mathbf{n}^T}\pi(n^T)\Big(\frac{1}{2}x'(n^T)Q_i(n^T)x(n^T) + p_i'(n^T)x(n^T) - \lambda_i'(n^T)x(n^T)\Big) + \lambda_i'(n^0)x_0$$

Equations (7)–(9) are obtained by taking the partial derivatives of the Lagrangian with respect to $u_i(n^t)$ at $u_i^*(n^t)$ and $x(n^t)$ at $x^*(n^t)$. □

From the above necessary conditions, it is clear that the S-adapted equilibrium depends upon the information that players use during the decision-making process. More precisely, if $\Gamma_i^{n^t}$ denotes the information set available to player i at node n^t, then the admissible actions available to the player are governed by $\Gamma_i^{n^t}$. We will consider two widely used information sets.[3] Firstly, players can design their equilibrium strategy, indexed over the nodes of the event tree, at the beginning of the planning period, and then stick to that strategy throughout the entire planning period. This is called S-adapted open-loop strategy. Secondly, players can design a rule for choosing actions at each node of the event tree, based on the observations of the state allowing for continuous revision of the strategy. This is called an S-adapted closed-loop strategy. We define these behavioral notions formally in the following definition:

Definition 3 *In an N-person LQGET of prespecified fixed duration, we say that player i's information structure is an*

1. *S-adapted open-loop pattern if* $\Gamma_i^{\mathsf{o}n^t} = \{n^t; x_0\}$*, with the initial state* x_0 *being fixed and a known parameter of the game.*
2. *S-adapted closed-loop no-memory pattern if* $\Gamma_i^{\mathsf{c}n^t} = \{x(n^t)\}$.

In the first case, the players use $u_i(n^t) = \gamma_i^{\mathsf{o}}(n^t; x_0)$, where $\gamma_i^{\mathsf{o}} \in U_i^{n^t}$, and as a result, the term $\frac{\partial u_i(n^t)}{\partial x(n^t)} = 0$ in (8). Whereas, in the latter case, players use $u_i(n^t) = \gamma_i^{\mathsf{c}}(x(n^t))$, where $\gamma_i^{\mathsf{c}} : X \to U_i^{n^t}$ is a measurable mapping, and, as a result, the term $\frac{\partial u_i(n^t)}{\partial x(n^t)} \neq 0$ in (8). In the remaining discussion, we derive necessary conditions for the S-adapted Nash equilibrium with these behavioral assumptions.

3.1 S-adapted open-loop Nash equilibrium

In the computation of an open-loop S-adapted equilibrium, the control action depends only on the node n^t in the event tree. Therefore, (8) leads to

$$\begin{aligned}\lambda_i(n^t) =& A'(n^t)\lambda_i(\mathcal{S}(n^t)) + Q_i(n^t)x^*(n^t) + p_i(n^t),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T,\\ \lambda_i(n^T) =& Q_i(n^T)x^*(n^T) + p_i(n^T),\ n^T \in \mathbf{n}^T,\ i \in \bar{N}.\end{aligned} \tag{10}$$

[3] It is possible to introduce more general information structures, for instance, those that involve memory. However, we restrict our analysis to open-loop and closed-loop no-memory structures to avoid informational non-uniqueness of the Nash equilibria (see Başar and Olsder (1998) for more details).

Similarly, writing the same for the remaining players and letting $S_i(n^t) = B_i(n^t)R_{ii}^{-1}(n^t)B_i'(n^t)$, and using (7) in the state equation (1) gives

$$x^*(\nu) = A(n^t)x^*(n^t) - \sum_{l\in\bar{N}} S_l(n^t)\lambda_l(\mathcal{S}(n^t)),\ \nu \in \mathcal{S}(n^t),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash 0,\ x^*(n^0) = x_0. \tag{11}$$

Note that the right-hand side of the equilibrium state dynamics contains terms that are averaged over all the nodes $\nu \in \mathcal{S}(n^t)$. Consequently, we have $x^*(\nu^1) = x^*(\nu^2),\ \forall \nu^1, \nu^2 \in \mathcal{S}(n^t)$. The above equations provide a set of necessary conditions for the open-loop S-adapted Nash equilibrium in the form of a two-point boundary-value problem (10)–(11). There may or may not exist a solution for these problems. Further, if the above equations are solvable, there can be more than one solution (see Başar and Olsder (1998) and Jank and Abou-Kandil (2003) in a deterministic difference game context). The following theorem provides sufficient conditions for the existence of a unique S-adapted open-loop Nash equilibrium.

Theorem 3.2 *For an N-person LQGET, let $\Lambda_\circ(\mathcal{S}(n^t), n^t)$ and $K_i^\circ(n^t)$ $(n^t \in \mathbf{n}^t,\ i \in \bar{N})$ be matrices of appropriate dimensions, defined by*

$$\Lambda_\circ(\mathcal{S}(n^t), n^t) = I + \sum_{l\in\bar{N}} S_l(n^t)K_l^\circ(\mathcal{S}(n^t)), \tag{12}$$

$$K_i^\circ(n^t) = Q_i(n^t) + A'(n^t)K_i^\circ(\mathcal{S}(n^t))\Lambda_\circ^{-1}(\mathcal{S}(n^t), n^t)A(n^t),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T,\ K_i^\circ(n^T) = Q_i(n^T). \tag{13}$$

If the matrices $\Lambda_\circ(\mathcal{S}(n^t), n^t),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T$, thus recursively defined, are invertible, then the game admits a unique S adapted open-loop Nash equilibrium solution for player $i \in \bar{N}$ given by

$$\begin{aligned} u_i^*(n^t) \equiv \gamma_i^\circ(x_0, n^t) = &- R_{ii}^{-1}(n^t)B_i'(n^t)\left(K_i^\circ(\mathcal{S}(n^t))\Lambda_\circ^{-1}(\mathcal{S}(n^t), n^t)A(n^t)x^*(n^t)\right) \\ &- R_{ii}^{-1}(n^t)B_i'(n^t)\Big(\beta_i^\circ(\mathcal{S}(n^t)) - K_i^\circ(\mathcal{S}(n^t))\Lambda_\circ^{-1}(\mathcal{S}(n^t), n^t)\sum_{l\in\bar{N}} S_l(n^t)\beta_l^\circ(\mathcal{S}(n^t))\Big), \end{aligned} \tag{14}$$

where $\tilde{x}^*$ is the associated state trajectory determined from

$$x^*(\nu) = \Lambda_\circ^{-1}(\mathcal{S}(n^t), n^t)\left(A(n^t)x^*(n^t) - \sum_{l\in\bar{N}} S_l(n^t)\beta_l^\circ(\mathcal{S}(n^t))\right),\ \forall \nu \in \mathcal{S}(n^t),\ x(n^0) = x_0, \tag{15}$$

with $\beta_i^\circ(n^t)$ recursively defined by

$$\begin{aligned} \beta_i^\circ(n^t) = p_i(n^t) + A'(n^t)\Big(\beta_i^\circ(\mathcal{S}(n^t)) - K_i^\circ(\mathcal{S}(n^t))\Lambda_\circ^{-1}(\mathcal{S}(n^t), n^t)\sum_{l\in\bar{N}} S_l(n^t)\beta_l^\circ(\mathcal{S}(n^t))\Big), \\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T,\ \beta_i(n^T) = p_i(n^T),\ n^T \in \mathbf{n}^T. \end{aligned} \tag{16}$$

Proof. See Appendix. □

We show in the following corollary that the invertibility of the matrices $\{\Lambda_\circ(\mathcal{S}(n^t), n^t),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T\}$ is a sufficient condition for unique solvability of the two-point boundary-value problem (10)–(11) defined on the event tree.

Corollary 3.3 *If the set of backward equations* (12), (13) *and* (16) *admit a solution then the two-point boundary-value problem* (10)–(11) *has a unique solution.*

Proof. See Appendix. □

Remark 2 *We emphasize that, though $u_i^*(n^t)$ appears affine in the state $x^*(n^t)$, the players actually implement $u^*(n^t) = \gamma^\circ(n^t; x_0)$ (to be consistent with Definition 3). This is achieved by representing $x^*(n^t)$ as a function of x_0^* and n^t using* (15) *recursively.*

Remark 3 *Note that (13) is a generalized coupled Riccati-type recursive equation defined on the event tree. We use the term generalized to emphasize that the computation of $K_i^{\mathsf{o}}(n^t)$ involves computing the expected sum of $K_j^{\mathsf{o}}(\nu),\ j \in \bar{N}$, over all the successor nodes, $\nu \in \mathcal{S}(n^t)$, of n^t. This feature is due to the one-period-lag assumption in the state dynamics (1). So, starting at the leaf nodes, matrices $K_i^{\mathsf{o}}(n^t)$ are computed recursively for the entire event tree using (13). Riccati equations defined over event/scenario trees have been studied earlier in the context of multistate stochastic programming. Salinger and Rockafellar (1999) apply operator splitting methods to solve a class of multistage stochastic programs and arrive at a linear-quadratic control problem defined over event trees as one of the subproblems. They show that this subproblem can be solved using linear feedback solution, which includes a backward Riccati recursion defined on the event tree. Blomvall and Lindberg (2002a, 2002b) show that certain linear and nonlinear multistage stochastic optimization problems can be solved by combining primal interior point methods with a linear-quadratic control problem over the scenario tree, which again leads to a backward Riccati equation defined on the event tree. Our paper is concerned with linear-quadratic multistage games defined over event trees, and in this regard we obtain a coupled Riccati recursion over the event tree.*

Next, (16) is a linear-backward recursive equation starting at the leaf nodes of the event tree. The parameters entering this equation depend[4] upon the solution of (13). We denote $\boldsymbol{\beta}^{\mathsf{o}}(n^t) = \texttt{Col}(\beta_i^{\mathsf{o}}(n^t))_{i=1}^N$, and represent (16) in vector form as follows:

$$\boldsymbol{\beta}^{\mathsf{o}}(n^t) = \mathbf{p}(n^t) + \mathbf{G}_{\mathsf{o}}(\mathcal{S}(n^t), n^t)\boldsymbol{\beta}^{\mathsf{o}}(\mathcal{S}(n^t)),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T,\ \boldsymbol{\beta}^{\mathsf{o}}(n^T) = \mathbf{p}(n^T), \tag{17}$$

where $\mathbf{p}(n^t)$ is again a vector representation of $p_i(n^t),\ i \in \bar{N}$, and the open-loop gain matrix $\mathbf{G}_o(\mathcal{S}(n^t), n^t)$ is given by

$$\Big[\mathbf{G}_{\mathsf{o}}(\mathcal{S}(n^t), n^t)\Big]_{ij} = \begin{cases} -A'(n^t)(K_i^{\mathsf{o}}(\mathcal{S}(n^t))\Lambda_{\mathsf{o}}^{-1}(\mathcal{S}(n^t), n^t)S_j(n^t)), & i \neq j \\ A'(n^t) - A'(n^t)K_i^{\mathsf{o}}(\mathcal{S}(n^t))\Lambda_{\mathsf{o}}^{-1}(\mathcal{S}(n^t), n^t)S_i(n^t), & i = j. \end{cases}$$

3.2 S-adapted closed-loop (no-memory) Nash equilibrium

We assume that the players use strategies of the type $u_i(n^t) = \gamma_i^{\mathsf{c}}(x(n^t))$. The adjoint equation (8) has the following form:

$$\lambda_i(n^t) = Q_i(n^t)x^*(n^t) + p_i(n^t) + \left(A(n^t) + \sum_{j\in\bar{N}\backslash i} B_j(n^t)\frac{\partial u_j^*(n^t)}{\partial x^*(n^t)}\right)'\lambda_i(\mathcal{S}(n^t)) + \sum_{j\in\bar{N}\backslash i}\left(\frac{\partial u_j^*(n^t)}{\partial x^*(n^t)}\right)' R_{ij}(n^t)u_j^*(n^t),$$
$$n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T,\ \lambda_i(n^T) = Q_i(n^T)x^*(n^T) + p_i(n^T). \tag{18}$$

Coupled with (1) and (7), the above equation results in a two-point boundary-value problem, and provides necessary conditions for the existence of an S-adapted closed-loop Nash equilibrium. Like before, we provide the sufficient conditions for the existence of a unique S-adapted closed-loop no-memory Nash equilibrium in the following theorem:

Theorem 3.4 *For an N-person LQGET, let $\Lambda_{\mathsf{c}}(\mathcal{S}(n^t), n^t)$ and $K_i^{\mathsf{c}}(n^t)$ $(n^t \in \mathbf{n}^t,\ i \in \bar{N})$ be matrices of the appropriate dimensions, defined by*

$$\Lambda_{\mathsf{c}}(\mathcal{S}(n^t), n^t) = I + \sum_{l\in\bar{N}} S_l(n^t)K_l^{\mathsf{c}}(\mathcal{S}(n^t)), \tag{19}$$

$$K_i^{\mathsf{c}}(n^t) = Q_i(n^t) + A'(n^t)\sum_{l\in\bar{N}}\Theta_{il}'(\mathcal{S}(n^t), n^t)K_l^{\mathsf{c}}(\mathcal{S}(n^t))\Lambda_{\mathsf{c}}^{-1}(\mathcal{S}(n^t), n^t)A(n^t), \tag{20}$$
$$n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T,\ K_i^{\mathsf{o}}(n^T) = Q_i(n^T),$$

where

$$\Theta_{ij}(\mathcal{S}(n^t), n^t) = \begin{cases} I - \sum_{j\in\bar{N}\backslash i} S_j(n^t)K_j^{\mathsf{c}}(\mathcal{S}(n^t))\Lambda_{\mathsf{c}}^{-1}(\mathcal{S}(n^t), n^t), & i = j, \\ S_{ij}(n^t)K_j^{\mathsf{c}}(\mathcal{S}(n^t))\Lambda_{\mathsf{c}}^{-1}(\mathcal{S}(n^t), n^t), & i \neq j, \end{cases}$$

[4] Note that the solution of (13) does not depend on (16). We will see in Section 4 that this feature facilitates solving the constrained LQGET model.

and $S_{ij}(n^t) = B_j(n^t)R_{jj}^{-1}(n^t)R_{ij}(n^t)R_{jj}^{-1}(n^t)B_j'(n^t),\ i \neq j$. *If the matrices* $\Lambda_{\mathsf{c}}(\mathcal{S}(n^t), n^t),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T$, *thus recursively defined, are invertible, then the game admits an S-adapted closed-loop no-memory Nash equilibrium solution for player* $i \in \bar{N}$, *which is given by*

$$\begin{aligned} u_i^*(n^t) \equiv \gamma_i^{\mathsf{c}}(n^t) = &- R_{ii}^{-1}(n^t)B_i'(n^t)\left(K_i^{\mathsf{c}}(\mathcal{S}(n^t))\Lambda_{\mathsf{c}}^{-1}(\mathcal{S}(n^t), n^t)A(n^t)x^*(n^t)\right) \\ &- R_{ii}^{-1}(n^t)B_i'(n^t)\Big(\beta_i^{\mathsf{c}}(\mathcal{S}(n^t)) - K_i^{\mathsf{c}}(\mathcal{S}(n^t))\Lambda_{\mathsf{c}}^{-1}(\mathcal{S}(n^t), n^t)\sum_{l\in\bar{N}} S_l(n^t)\beta_l^{\mathsf{c}}(\mathcal{S}(n^t))\Big), \end{aligned} \tag{21}$$

where $\tilde{x}^*$ *is the associated state trajectory determined from*

$$x^*(\nu) = \Lambda_{\mathsf{c}}^{-1}(\mathcal{S}(n^t), n^t)\Bigg(A(n^t)x^*(n^t) - \sum_{l\in\bar{N}} S_l(n^t)\beta_l^{\mathsf{c}}(\mathcal{S}(n^t))\Bigg),\ \nu \in \mathcal{S}(n^t),\ x(n^0) = x_0,$$

with $\beta_i^{\mathsf{c}}(n^t)$ *recursively defined by*

$$\begin{aligned} \beta_i^{\mathsf{c}}(n^t) = p_i(n^t) + A'(n^t)\sum_{j\in\bar{N}}\Theta_{ij}'(\mathcal{S}(n^t), n^t)\Big(\beta_j^{\mathsf{c}}(\mathcal{S}(n^t)) - K_j^{\mathsf{c}}(\mathcal{S}(n^t))\Lambda_{\mathsf{c}}^{-1}(\mathcal{S}(n^t), n^t)\sum_{l\in\bar{N}} S_l(n^t)\beta_l^{\mathsf{c}}(\mathcal{S}(n^t))\Big), \\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T,\ \beta_i^{\mathsf{c}}(n^T) = p_i(n^T),\ n^T \in \mathbf{n}^T. \end{aligned} \tag{22}$$

Proof. See Appendix. □

Remark 4 *The above approach for finding the closed-loop (no-memory) Nash equilibrium suffers with the problem of informational non uniqueness, see Başar and Olsder (1998). To observe this, firstly we imposed a linear structure for the costate variable that leads to an affine structure of control* (44)*, which again influences the costate variable* (18) *through* (45)*; see also Footnote 5. One way to circumvent the problem is to assume that players are restricted at the outset to memoryless strategies (affine in state variable here) and solve the minimization problem, given in Definition 4. This approach, however, leads to a different problem formulation and system of equations, see Lukes (1971).*

Again, we observe that (20) is a generalized coupled Riccati-type recursive equation defined on the event tree, due to the one-period lag in the dynamics (1). Next, (22) is a linear-backward recursive equation starting at the leaf nodes of the event tree. Note that the above equations are structurally similar[5] to those obtained for the open-loop case. Like before, (22) can be represented in the following vector form:

$$\boldsymbol{\beta}^{\mathsf{c}}(n^t) = \mathbf{p}(n^t) + \mathbf{G}_{\mathsf{c}}(\mathcal{S}(n^t), n^t)\boldsymbol{\beta}^{\mathsf{c}}(\mathcal{S}(n^t)),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T,\ \boldsymbol{\beta}^{\mathsf{c}}(n^T) = \mathbf{p}(n^T), \tag{23}$$

where the closed-loop gain is given by

$$\Big[\mathbf{G}_{\mathsf{c}}(\mathcal{S}(n^t), n^t)\Big]_{ij} = A'(n^t)\Theta_{ij}'(\mathcal{S}(n^t), n^t) - A'(n^t)\Bigg(\sum_{l\in\bar{N}}\Theta_{il}'(\mathcal{S}(n^t), n^t)K_l^{\mathsf{c}}(\mathcal{S}(n^t))\Lambda_{\mathsf{c}}^{-1}(\mathcal{S}(n^t), n^t)\Bigg)S_j(n^t).$$

So far, we have discussed a dynamic game model where the players make decisions before the realization of uncertainty, and without constraints. In the next section we shall discuss a particular class of stochastic dynamic games where players can influence the game after the realization of uncertainty. Further, these decision variables enter the game description in the form of constraints jointly with the state variable.

4 Constrained LQGET model

In this section we provide necessary conditions for an S-adapted equilibrium in the Con-LQGET model. We have the following definition:

[5] We can recover the open-loop S-adapted equilibrium from (19)–(22) by replacing Θ with the identity matrix.

Definition 4 *A constrained S-adpated equilibrium is an admissible S-adapted strategy* $(\tilde{\mathbf{u}}^*, \tilde{\mathbf{v}}^*)$ *if the following relations hold:*

$$Jc_i(x_0, (\tilde{u}_i^*, \tilde{u}_{i-}^*), (\tilde{v}_i^*, \tilde{v}_{i-}^*)) \leq Jc_i(x_0, (\tilde{u}_i, \tilde{u}_{i-}^*), (\tilde{v}_i, \tilde{v}_{i-}^*)),\ \forall \tilde{u}_i \in \tilde{U}_i,\ \tilde{v}_i \in \tilde{V}_i,$$

for all $i \in \bar{N}$*, subject to* (1) *and* (2)*.*

We introduce the Lagrangian connected with player i's minimization problem as follows:

$$\begin{aligned}
\mathcal{L}c_i(\tilde{x}, (u_i, \tilde{u}_{-i}^*), (v_i, \tilde{v}_{-i}^*), \tilde{\lambda}_i, \tilde{\mu}_i) &= \mathcal{L}_i(\tilde{x}, (u_i, \tilde{u}_{-i}^*), \tilde{\lambda}_i) \\
&+ \sum_{t=0}^{T} \sum_{n^t \in \mathbf{n}^t} \pi(n^t) \Big(t_i^{i\prime}(n^t) v_i(n^t) + x'(n^t) L_i^i(n^t) v_i(n^t) + t_{-i}^{i}{}'(n^t) v_{-i}^*(n^t) + x'(n^t) L_{-i}^i(n^t) v_{-i}^*(n^t) \Big) \\
&+ \sum_{t=0}^{T} \sum_{n^t \in \mathbf{n}^t} \frac{\pi(n^t)}{2} \Big(v_i'(n^t) T_{ii}^i(n^t) v_i(n^t) + 2 v_i'(n^t) T_{i,-i}^i v_{-i}^*(n^t) + v_{-i}^*{}'(n^t) T_{-i,-i}^i v_{-i}^*(n^t) \Big) \\
&- \sum_{t=0}^{T} \sum_{n^t \in \mathbf{n}^t} \pi(n^t) \mu_i'(n^t) \Big(M_i(n^t) x(n^t) + N_i(n^t) v_i(n^t) + r_i(n^t) \Big),
\end{aligned} \tag{24}$$

where $\tilde{\mu}_i$ is the multiplier associated with constraints (2). In the LQGET model, we gave necessary conditions in Theorem 3.1 for an S-adapted equilibrium strategy, without making a prior distinction about whether the players will base their decision on their location on the event tree or on the state variable. Later, while computing the adjoint variable, such a distinction was made to arrive at the different behavioral strategies. In the constrained case, the adjoint variable $\lambda_i(n^t)$, evaluated by taking the partial derivative of the above Lagrangian with respect to the state variable, is a function of the terms $\frac{\partial u_i(n^t)}{\partial x(n^t)}$ and $\frac{\partial v_i(n^t)}{\partial x(n^t)}$. Further, we observe that the state variable inherently affects the actions $\mathbf{v}(n^t)$ as a parameter entering the node specific constraint. The information structure depends on how players view the state variable available at n^t: as a parameter entering the constraint or as an additional instrument on which to base their action. Therefore, the constrained model needs a deeper analysis and we elaborate on a specific information structure in the next subsection.

4.1 Constrained information structure and S-adapted Nash equilibrium

As before, players can condition their actions $u_i(n^t)$ on the position of the node on the event tree as $u_i(n^t) = \gamma_i^{\mathsf{o}}(n^t; x_0)$, or on the state variable as $u_i(n^t) = \gamma_i^{\mathsf{c}}(x(n^t))$. For the actions $v_i(n^t)$ there are again two possibilities: players treat the state variable as a parameter entering the constraint as $v_i(n^t) = \delta_i(n^t; x(n^t))$, or as a function of the state variable as $v_i(n^t) = \delta_i(x(n^t))$.[6] In summary, all of these four situations can occur. We restrict our attention to the case $v(n^t) = \delta_i(n^t; x(n^t)) \in \Delta_i(x(n^t))$, where $\Delta_i(x(n^t)) \subset V_i^{n^t}$ is the action set that ensures feasibility of (2). We will see that this assumption uses a natural extension of LQGET model towards computing the S-adapted Nash equilibrium. Further, this assumption has the following interpretation: the players make decisions based on the location in the event tree and the state variable enters the feasibility constraints. Next, the latter case $v(n^t) = \delta_i(x(n^t))$ leads to pure state constraints at each node n^t of the event tree. It is not clear at the moment which functional forms of $\delta(.)$ would facilitate a closed-form analysis akin to the previous case. So we consider the following two information structures $\mathtt{I} = \{\mathtt{Co}, \mathtt{Cc}\}$ in the remaining analysis, where:

- `Co` - constrained S-adapted open-loop pattern if player i uses $u_i(n^t) = \gamma_i^{\mathsf{o}}(n^t; x_0)$ and $v_i(n^t) = \delta_i(n^t; x(n^t))$,
- `Cc` - constrained S-adapted closed-loop pattern if player i uses $u_i(x(n^t)) = \gamma_i^{\mathsf{c}}(x(n^t))$ and $v_i(n^t) = \delta_i(n^t; x(n^t))$.

Next, the necessary conditions for $(\tilde{\mathbf{u}}^*, \tilde{\mathbf{v}}^*)$ to be an admissible S-adapted Nash equilibrium for Con-LQGET, with information structure I, is given in the following theorem.

[6] The distinction surfaces clearly when the dimension of $x(n^t)$ is greater than one.

Theorem 4.1 *Assume that* $(\mathbf{u}^*, \mathbf{v}^*)$ *is an S-adapted equilibrium, with an information pattern I, generating the state trajectory* $\tilde{x}^*$ *over the event tree. Then there exists, for each player i, a costate profile* $\tilde{\lambda}_i$ *and a multiplier profile* $\tilde{\mu}_i$ *such that the following conditions hold for* $i \in \bar{N}$*:*

$$u_i^*(n^t) = -R_{ii}^{-1}(n^t)B_i'(n^t)\lambda_i(\mathcal{S}(n^t)),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T, \tag{25}$$

$$x^*(n^t) = A(a(n^t))x^*(a(n^t)) + \sum_{l\in\bar{N}} B_l(a(n^t))u_l^*(a(n^t)),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash 0,\ x(n^0) = x_0, \tag{26}$$

$$\lambda_i(n^t) = \frac{\partial \mathcal{H}_i(.)}{\partial x^*(n^t)} + p_i(n^t) - M_i'(n^t)\mu_i(n^t) + L^i(n^t)\mathbf{v}^*(n^t),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T, \tag{27}$$
$$\lambda_i(n^T) = Q_i(n^T)x^*(n^T) + p_i(n^T) - M_i'(n^T)\mu_i(n^T) + L^i(n^T)\mathbf{v}^*(n^T),$$

$$T_{ii}^i v_i^*(n^t) + \sum_{j\in\bar{N}\backslash i} T_{ij}^i v_j^*(n^t) + L_i^{i\prime}(n^t)x^*(n^t) - N_i'(n^t)\mu_i(n^t) + t_i^i(n^t) \perp v_i^*(n^t), \tag{28}$$

$$M_i(n^t)x^*(n^t) + N_i(n^t)v_i^*(n^t) + r_i(n^t) \perp \mu_i(n^t), \quad n^t \in \mathbf{n}^t,\ t \in \mathcal{T}. \tag{29}$$

Proof. See Appendix. □

A few comments on the role of the one-period lag in the dynamics and separable cost structure are in order. In the LQGET model, we noticed that the one-period lag in the dynamics resulted in recursive formulations towards computing the equilibria. In the Con-LQGET, we obtain similar recursive formulations due to the one-period lag and the restriction of the information structure to either `Co` or `Cc`. This is clear from the three necessary conditions (25)–(27), as they are similar to those of a LQGET model with the vector $p_i(t)$ replaced with $p_i(t) - M_i'(n^t)\mu_i(n^t) + L^i(n^t)\mathbf{v}^*(n^t)$. So we represent this set of equations as LQGET$(\tilde{\boldsymbol{\mu}}, \tilde{\mathbf{v}}^*)$. The last two equations represent the necessary conditions for the solution of a linear-complementarity problem, usually associated with solving for Nash equilibria in static games, computed at each node of the event tree and parametrized with the equilibrium state trajectory $\tilde{x}^*$. This is a consequence of the separable cost structure. We represent them as LCP$(\tilde{x}^*)$. In summary, the S-adapted equilibrium in the Con-LQGET model is obtained by solving these (weakly) coupled systems of equations.

Remark 5 *We emphasize that the necessary conditions given by* (25)*–*(29) *are not sufficient. So, solving them result in candidates for constrained Nash equilibria. A few comments related to this aspect are given in Remark 6.*

4.2 Reformulation as a linear-complementarity problem

In this subsection, we illustrate a procedure for solving the system of equations (25)–(29). To start, note that LQGET$(\tilde{\boldsymbol{\mu}}, \tilde{\mathbf{v}}^*)$ also results in recursive equations similar to (17) and (23), which are structurally the same, except for the gain terms, which differ for the open-loop and closed-loop cases. So, from now on we suppress the superscripts. We replace $p_i(n^t)$ in (16) or (22) with $p_i(t) - M_i'(n^t)\mu_i(n^t) + L^i(n^t)\mathbf{v}^*(n^t)$ for all $i \in \bar{N}$ to obtain the following linear recursion in vector form:

$$\boldsymbol{\beta}(n^t) = \mathbf{p}(n^t) + \begin{bmatrix}\mathbf{L}_1(n^t) - \mathbf{M}_1'(n^t)\end{bmatrix}\begin{bmatrix}\mathbf{v}^*(n^t)\\ \boldsymbol{\mu}(n^t)\end{bmatrix} + \mathbf{G}(\mathcal{S}(n^t), n^t)\ \boldsymbol{\beta}(\mathcal{S}(n^t)),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T,$$
$$\boldsymbol{\beta}(n^T) = \mathbf{p}(n^T) + \begin{bmatrix}\mathbf{L}_1(n^T) & -\mathbf{M}_1'(n^T)\end{bmatrix}\begin{bmatrix}\mathbf{v}^*(n^T)\\ \boldsymbol{\mu}(n^T)\end{bmatrix}, \tag{30}$$

where $\mathbf{L}_1(n^t) = \texttt{Col}(L^i(n^t))_{i=1}^N$, $\mathbf{M}_1(n^t) = \texttt{BDmat}(M_i(n^t))_{i=1}^N$ and $\boldsymbol{\mu}(n^t) = \texttt{Col}(\mu_i(n^t))_{i=1}^N$. Denoting $\mathbf{S}(n^t) = \texttt{Row}(S_i(n^t))_{i=1}^N$, the state equation is represented as follows:

$$x^*(\mathcal{S}(n^t)) = \Lambda^{-1}(\mathcal{S}(n^t), n^t)\left(A(n^t)x^*(n^t) - \mathbf{S}(n^t)\boldsymbol{\beta}(\mathcal{S}(n^t))\right),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T,\ x(n^0) = x_0. \tag{31}$$

Next, the vector representation of LCP$(x^*(n^t))$ is given by

$$\begin{bmatrix}\mathbf{T}(n^t) & -\mathbf{N}'(n^t)\\ \mathbf{N}(n^t) & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{v}^*(n^t)\\ \boldsymbol{\mu}(n^t)\end{bmatrix}+\begin{bmatrix}\mathbf{L}_2'(n^t)\\ \mathbf{M}_2(n^t)\end{bmatrix}x^*(n^t)+\begin{bmatrix}\mathbf{t}(n^t)\\ \mathbf{r}(n^t)\end{bmatrix}\perp\begin{bmatrix}\mathbf{v}^*(n^t)\\ \boldsymbol{\mu}(n^t)\end{bmatrix}, \tag{32}$$

where $[\mathbf{T}(n^t)]_{ij}=T^i_{ij}(n^t)$, $\mathbf{N}(n^t)=\mathtt{BDmat}(N_i(n^t))_{i=1}^N$, $\mathbf{L}_2(n^t)=\mathtt{Row}(L^i_i(n^t))_{i=1}^N$, $\mathbf{M}_2(n^t)=\mathtt{Col}(M_i(n^t))_{i=1}^N$, $\mathbf{t}(n^t)=\mathtt{Col}(t^i_i(n^t))_{i=1}^N$ and $\mathbf{r}(n^t)=\mathtt{Col}(r_i(n^t))_{i=1}^N$. We introduce some additional notation before presenting the main results. First, we enumerate the event tree as $\mathfrak{n}=\{n^0,\mathbf{n}\}$, where $\mathbf{n}=\{\mathbf{n}^1,\mathbf{n}^2\cdots,\mathbf{n}^T\}$ and $\mathbf{n}^t=\{n^t_1,n^t_2,\cdots,n^t_{|\mathbf{n}^t|}\}$. Observe that (30) is a backward-linear recursion defined at each node n^t of the event tree relating to its successors $\mathcal{S}(n^t)$. We denote $\boldsymbol{\beta}(\mathbf{n}^t)=\mathtt{Col}(\boldsymbol{\beta}(n^t_i))_{i=1}^{|\mathbf{n}^t|}$ and provide a relation with $\boldsymbol{\beta}(\mathbf{n}^{t+1})$ in Lemma 4.2. Again denoting $\boldsymbol{\beta}(\mathbf{n})=\mathtt{Col}(\boldsymbol{\beta}(\mathbf{n}^t_i))_{i=1}^T$, in the same lemma, we provide a relation of $\boldsymbol{\beta}(\mathbf{n})$ with the parameters $\tilde{\mathbf{v}}^*$ and $\tilde{\boldsymbol{\mu}}$. Then using these ideas in Lemma 4.3, we provide a relation between $x^*(\mathbf{n})$, the state information on the entire event tree, excepting the root node, and the parameters $\tilde{\mathbf{v}}^*$ and $\tilde{\boldsymbol{\mu}}$. Finally, we collect these results to transform (32) as a parametrized linear-complementarity problem, with known x_0^* and $\mathbf{p}(\mathbf{n})$, which is defined on the enumerated event tree $\mathfrak{n}$. We present these steps in the following lemmas.

Lemma 4.2 *For the enumerated event tree $\mathfrak{n}=\{n^0,\mathbf{n}\}$, the following relation holds:*

$$\boldsymbol{\beta}(\mathbf{n})=\mathcal{G}^{-1}\left(\mathbf{p}(\mathbf{n})+\begin{bmatrix}\mathbf{L}_1(\mathbf{n}) & -\mathbf{M}_1'(\mathbf{n})\end{bmatrix}\begin{bmatrix}\mathbf{v}^*(\mathbf{n})\\ \boldsymbol{\mu}(\mathbf{n})\end{bmatrix}\right). \tag{33}$$

Proof. Let the set of successors of the node n^t_i be $\mathcal{S}(n^t_i)=\left\{\nu^1_i,\nu^2_i,\cdots,\nu_i^{|\mathcal{S}(n^t_i)|}\right\}$. Then equation (30) is rewritten as

$$\boldsymbol{\beta}(n^t_i)=\mathbf{p}(n^t_i)+\begin{bmatrix}\mathbf{L}_1(n^t_i) & -\mathbf{M}_1'(n^t_i)\end{bmatrix}\begin{bmatrix}\mathbf{v}^*(n^t_i)\\ \boldsymbol{\mu}(n^t_i)\end{bmatrix}+\left(\boldsymbol{\pi}_{n^t_i}^{\mathcal{S}(n^t_i)}\otimes\mathbf{G}(\mathcal{S}(n^t_i),n^t_i)\right)\begin{bmatrix}\boldsymbol{\beta}(\nu^1_i)\\ \boldsymbol{\beta}(\nu^2_i)\\ \vdots\\ \boldsymbol{\beta}(\nu_i^{|\mathcal{S}(n^t_i)|})\end{bmatrix}.$$

We introduce the block diagonal matrices $\mathbf{L}_1(\mathbf{n}^t)=\mathtt{BDmat}(\mathbf{L}_1(n^t_k))_{k=1}^{|\mathbf{n}^t|}$, $\mathbf{M}_1(\mathbf{n}^t)=\mathtt{BDmat}(\mathbf{M}_1(n^t_k))_{k=1}^{|\mathbf{n}^t|}$ and $\mathbf{G}_\pi(\mathbf{n}^{t+1},\mathbf{n}^t)=\mathtt{BDmat}\left(\boldsymbol{\pi}_{n^t_k}^{\mathcal{S}(n^t_k)}\otimes\mathbf{G}(\mathcal{S}(n^t_k),n^t_k)\right)_{k=1}^{|\mathbf{n}^t|}$. The linear recursion defined on the set of nodes at t and $t+1$ is given by

$$\boldsymbol{\beta}(\mathbf{n}^t)=\mathbf{p}(\mathbf{n}^t)+\begin{bmatrix}\mathbf{L}_1(\mathbf{n}^t) & -\mathbf{M}_1'(\mathbf{n}^t)\end{bmatrix}\begin{bmatrix}\mathbf{v}^*(\mathbf{n}^t)\\ \boldsymbol{\mu}(\mathbf{n}^t)\end{bmatrix}+\mathbf{G}_\pi(\mathbf{n}^{t+1},\mathbf{n}^t)\ \boldsymbol{\beta}(\mathbf{n}^{t+1}).$$

Now, defining the block matrices $\mathbf{L}_1(\mathbf{n})=\mathtt{BDmat}(\mathbf{L}_1(\mathbf{n}^t))_{t=1}^T$, $\mathbf{M}_1(\mathbf{n})=-\mathtt{BDmat}(\mathbf{M}_1(\mathbf{n}^t))_{t=1}^T$ and $\mathbf{G}_\pi(\mathbf{n})=\mathtt{BDmat}(\mathbf{G}_\pi(\mathbf{n}^t))_{t=1}^T$, we have

$$\boldsymbol{\beta}(\mathbf{n})=\mathbf{p}(\mathbf{n})+\begin{bmatrix}\mathbf{L}_1(\mathbf{n}) & -\mathbf{M}_1'(\mathbf{n})\end{bmatrix}\begin{bmatrix}\mathbf{v}^*(\mathbf{n})\\ \boldsymbol{\mu}(\mathbf{n})\end{bmatrix}+\begin{bmatrix}\mathbf{0} & \mathbf{G}_\pi(\mathbf{n})\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\boldsymbol{\beta}(\mathbf{n}).$$

Since, $\mathbf{G}_\pi(\mathbf{n})$ is block diagonal, the matrix $\mathcal{G}=\mathbf{I}-\left[\begin{smallmatrix}\mathbf{0} & \mathbf{G}_\pi(\mathbf{n})\\ \mathbf{0} & \mathbf{0}\end{smallmatrix}\right]$ is invertible and the relation (33) follows. □

Lemma 4.3 *The optimal-state trajectory evaluated on the enumerated event tree $\mathfrak{n}=\{n^0,\mathbf{n}\}$, represented by $x^*(\mathbf{n})$, satisfies the following relation:*

$$x^*(\mathbf{n})=\boldsymbol{\psi}_0(\mathbf{n})x_0^*+\boldsymbol{\psi}_1(\mathbf{n})\left(\mathbf{p}(\mathbf{n})+\begin{bmatrix}\mathbf{L}_1(\mathbf{n}) & -\mathbf{M}_1'(\mathbf{n})\end{bmatrix}\begin{bmatrix}\mathbf{v}^*(\mathbf{n})\\ \boldsymbol{\mu}(\mathbf{n})\end{bmatrix}\right). \tag{34}$$

Proof. As before, we list the state variables computed at the successors of a node $n^t_i\in\mathbf{n}^t$ as follows:

$$\begin{bmatrix}x^*(\nu^1_i)\\ x^*(\nu^2_i)\\ \vdots\\ x^*(\nu_i^{|\mathcal{S}(n^t_i)|})\end{bmatrix}=\boldsymbol{\Phi}_1(\mathcal{S}(n^t_i),n^t_i)x^*(n^t_i)+\boldsymbol{\Phi}_2(\mathcal{S}(n^t_i),n^t_i)\begin{bmatrix}\boldsymbol{\beta}(\nu^1_i)\\ \boldsymbol{\beta}(\nu^2_i)\\ \vdots\\ \boldsymbol{\beta}(\nu_i^{|\mathcal{S}(n^t_i)|})\end{bmatrix},$$

where

$$\mathbf{\Phi}_1(\mathcal{S}(n_i^t), n_i^t) = \Big(\mathbf{1}_{|\mathcal{S}(n_i^t)|\times 1} \otimes \Lambda^{-1}(\mathcal{S}(n_i^t), n_i^t)\Big) A(n_i^t),$$
$$\mathbf{\Phi}_2(\mathcal{S}(n_i^t), n_i^t) = \Big(\mathbf{1}_{|\mathcal{S}(n_i^t)|\times 1} \otimes \Lambda^{-1}(\mathcal{S}(n_i^t), n_i^t)\Big)\Big(\boldsymbol{\pi}_{n_i^t}^{\mathcal{S}(n_i^t)} \otimes \mathbf{S}(n_i^t)\Big).$$

Here, notice that the Kronecker product with the vector of ones reflects the one-period lag structure in the state dynamics. Next, we define, as before, the block diagonal matrices, which relate the set of nodes at times t and $t+1$ as $\mathbf{\Phi}_1(\mathbf{n}^{t+1}, \mathbf{n}^t) = \texttt{BDmat}\left(\mathbf{\Phi}_1(\mathcal{S}(n_k^t), n_k^t)\right)_{k=1}^{|\mathbf{n}^t|}$ and $\mathbf{\Phi}_2(\mathbf{n}^{t+1}, \mathbf{n}^t) = \texttt{BDmat}\left(\mathbf{\Phi}_2(\mathcal{S}(n_k^t), n_k^t)\right)_{k=1}^{|\mathbf{n}^t|}$. So, the state vector computed at the nodes $\mathbf{n}^{t+1}$ is then given by

$$x^*(\mathbf{n}^{t+1}) = \mathbf{\Phi}_1(\mathbf{n}^{t+1}, \mathbf{n}^t) x^*(\mathbf{n}^t) + \mathbf{\Phi}_2(\mathbf{n}^{t+1}, \mathbf{n}^t)\boldsymbol{\beta}(\mathbf{n}^{t+1}).$$

We define $\mathbf{\Phi}_1(\mathbf{n}) = \texttt{BDmat}\left(\mathbf{\Phi}_1(\mathbf{n}^{t+1}, \mathbf{n}^t)\right)_{t=1}^{T-1}$ and $\mathbf{\Phi}_2(\mathbf{n}) = \texttt{BDmat}\left(\mathbf{\Phi}_2(\mathbf{n}^{t+1}, \mathbf{n}^t)\right)_{t=1}^{T-1}$. The state vector computed on the entire event tree is given as follows:

$$\begin{bmatrix} x_0^* \\ \hline x^*(\mathbf{n}^1) \\ \vdots \\ x^*(\mathbf{n}^T) \end{bmatrix} = \left[\begin{array}{c|ccc} I & \mathbf{0} & \cdots & \mathbf{0} \\ \hline \mathbf{\Phi}_1(\mathbf{n}^1, n^0) & \mathbf{0} & & \mathbf{0} \\ \vdots & & & \\ \mathbf{0} & \mathbf{\Phi}_1(\mathbf{n}) & & \mathbf{0} \end{array}\right] \begin{bmatrix} x_0^* \\ \hline x^*(\mathbf{n}^1) \\ \vdots \\ x^*(\mathbf{n}^T) \end{bmatrix} + \left[\begin{array}{c|ccc} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \hline \mathbf{0} & & & \\ \vdots & & \mathbf{\Phi}_2(\mathbf{n}) & \\ \mathbf{0} & & & \end{array}\right] \begin{bmatrix} \boldsymbol{\beta}(n^0) \\ \hline \boldsymbol{\beta}(\mathbf{n}^1) \\ \vdots \\ \boldsymbol{\beta}(\mathbf{n}^T) \end{bmatrix},$$

$$\left[\begin{array}{c|ccc} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \hline -\mathbf{\Phi}_1(\mathbf{n}^1, n^0) & & & \\ \vdots & & \mathbf{I} - \left[\begin{smallmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{\Phi}_1(\mathbf{n}) & \mathbf{0} \end{smallmatrix}\right] & \\ \mathbf{0} & & & \end{array}\right] \begin{bmatrix} x_0^* \\ \hline x^*(\mathbf{n}^1) \\ \vdots \\ x^*(\mathbf{n}^T) \end{bmatrix} = \left[\begin{array}{c|ccc} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \hline \mathbf{0} & & & \\ \vdots & & \mathbf{\Phi}_2(\mathbf{n}) & \\ \mathbf{0} & & & \end{array}\right] \begin{bmatrix} \boldsymbol{\beta}(n^0) \\ \hline \boldsymbol{\beta}(\mathbf{n}^1) \\ \vdots \\ \boldsymbol{\beta}(\mathbf{n}^T) \end{bmatrix}. \tag{35}$$

Here, $\mathbf{\Phi}_1(\mathbf{n})$ and $\mathbf{\Phi}_2(\mathbf{n})$ are block diagonal matrices, so the matrix $\mathbf{I} - \left[\begin{smallmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{\Phi}_1(\mathbf{n}) & \mathbf{0} \end{smallmatrix}\right]$ is invertible. Finally, using (33) in (35) and by identifying $\boldsymbol{\psi}_0(\mathbf{n}) = \left(\mathbf{I} - \left[\begin{smallmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{\Phi}_1(\mathbf{n}) & \mathbf{0} \end{smallmatrix}\right]\right)^{-1} \left[\begin{smallmatrix} \mathbf{\Phi}_1(\mathbf{n}^1, n^0) \\ \mathbf{0} \end{smallmatrix}\right]$ and $\boldsymbol{\psi}_1(\mathbf{n}) = \left(\mathbf{I} - \left[\begin{smallmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{\Phi}_1(\mathbf{n}) & \mathbf{0} \end{smallmatrix}\right]\right)^{-1} \Phi_2(\mathbf{n}) \left(\mathbf{I} - \left[\begin{smallmatrix} \mathbf{0} & \mathbf{G}_\pi(\mathbf{n}) \\ \mathbf{0} & \mathbf{0} \end{smallmatrix}\right]\right)^{-1}$, (34) follows. □

Next, linear-complementarity problems (32), defined at node n^t, is written as a single complementarity problem defined for all the nodes in $\mathbf{n}^t$ as follows:

$$\begin{bmatrix} \mathbf{T}(\mathbf{n}^t) & -\mathbf{N}'(\mathbf{n}^t) \\ \mathbf{N}(\mathbf{n}^t) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}^*(\mathbf{n}^t) \\ \boldsymbol{\mu}(\mathbf{n}^t) \end{bmatrix} + \begin{bmatrix} \mathbf{L}_2'(\mathbf{n}^t) \\ \mathbf{M}_2(\mathbf{n}^t) \end{bmatrix} x^*(\mathbf{n}^t) + \begin{bmatrix} \mathbf{t}(\mathbf{n}^t) \\ \mathbf{r}(\mathbf{n}^t) \end{bmatrix} \perp \begin{bmatrix} \mathbf{v}^*(\mathbf{n}^t) \\ \boldsymbol{\mu}(\mathbf{n}^t) \end{bmatrix},$$

where $\mathbf{T}(\mathbf{n}^t) = \texttt{BDmat}(\mathbf{T}(n_k^t))_{k=1}^{|\mathbf{n}^t|}$, $\mathbf{N}(\mathbf{n}^t) = \texttt{BDmat}(\mathbf{N}(n_k^t))_{k=1}^{|\mathbf{n}^t|}$, $\mathbf{L}_2(\mathbf{n}^t) = \texttt{BDmat}(\mathbf{N}_2(n_k^t))_{k=1}^{|\mathbf{n}^t|}$, $\mathbf{t}(\mathbf{n}^t) = \texttt{Col}(\mathbf{t}(n_k^t))_{k=1}^{|\mathbf{n}^t|}$, $\mathbf{v}(\mathbf{n}^t) = \texttt{Col}(\mathbf{v}(n_k^t))_{k=1}^{|\mathbf{n}^t|}$, $\boldsymbol{\mu}(\mathbf{n}^t) = \texttt{Col}(\boldsymbol{\mu}(n_k^t))_{k=1}^{|\mathbf{n}^t|}$. Again, using the same procedure, we write the above problems defined on $\mathbf{n}^t$ for all $t \in \mathcal{T}\backslash 0$ as a single problem defined on nodes in $\mathbf{n}$ as follows:

$$\begin{bmatrix} \mathbf{T}(\mathbf{n}) & -\mathbf{N}'(\mathbf{n}) \\ \mathbf{N}(\mathbf{n}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}^*(\mathbf{n}) \\ \boldsymbol{\mu}(\mathbf{n}) \end{bmatrix} + \begin{bmatrix} \mathbf{L}_2'(\mathbf{n}) \\ \mathbf{M}_2(\mathbf{n}) \end{bmatrix} x^*(\mathbf{n}) + \begin{bmatrix} \mathbf{t}(\mathbf{n}) \\ \mathbf{r}(\mathbf{n}) \end{bmatrix} \perp \begin{bmatrix} \mathbf{v}^*(\mathbf{n}) \\ \boldsymbol{\mu}(\mathbf{n}) \end{bmatrix},$$

where $\mathbf{T}(\mathbf{n}) = \texttt{BDmat}(\mathbf{T}(\mathbf{n}^t))_{t=1}^{T}$, $\mathbf{N}(\mathbf{n}) = \texttt{BDmat}(\mathbf{N}(\mathbf{n}^t))_{t=1}^{T}$, $\mathbf{L}_2(\mathbf{n}) = \texttt{BDmat}(\mathbf{N}_2(\mathbf{n}^t))_{t=1}^{T}$, $\mathbf{t}(\mathbf{n}) = \texttt{Col}(\mathbf{t}(\mathbf{n}^t))_{t=1}^{T}$, $\mathbf{v}(\mathbf{n}) = \texttt{Col}(\mathbf{v}(\mathbf{n}^t))_{t=1}^{T}$ and $\boldsymbol{\mu}(\mathbf{n}) = \texttt{Col}(\boldsymbol{\mu}(\mathbf{n}^t))_{t=1}^{T}$. Using (34), the above problem can be written as a parametrized linear complementarity problem defined for all the nodes in $\mathbf{n}$ as follows

$$\begin{bmatrix} \mathbf{A}_1(\mathbf{n}) & \mathbf{A}_2(\mathbf{n}) \\ \mathbf{A}_3(\mathbf{n}) & \mathbf{A}_4(\mathbf{n}) \end{bmatrix} \begin{bmatrix} \mathbf{v}^*(\mathbf{n}) \\ \boldsymbol{\mu}(\mathbf{n}) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1(\mathbf{n}) \\ \mathbf{B}_2(\mathbf{n}) \end{bmatrix} x_0^* + \begin{bmatrix} \mathbf{t}_p(\mathbf{n}) \\ \mathbf{r}_p(\mathbf{n}) \end{bmatrix} \perp \begin{bmatrix} \mathbf{v}^*(\mathbf{n}) \\ \boldsymbol{\mu}(\mathbf{n}) \end{bmatrix},$$

where

$$\begin{bmatrix} \mathbf{A}_1(\mathbf{n}) & \mathbf{A}_2(\mathbf{n}) \\ \mathbf{A}_3(\mathbf{n}) & \mathbf{A}_4(\mathbf{n}) \end{bmatrix} = \left(\begin{bmatrix} \mathbf{L}_2'(\mathbf{n}) \\ \mathbf{M}_2(\mathbf{n}) \end{bmatrix} \boldsymbol{\psi}_1(\mathbf{n}) \begin{bmatrix} \mathbf{L}_1(\mathbf{n}) & -\mathbf{M}_1'(\mathbf{n}) \end{bmatrix} + \begin{bmatrix} \mathbf{T}(\mathbf{n}) & -\mathbf{N}'(\mathbf{n}) \\ \mathbf{N}(\mathbf{n}) & \mathbf{0} \end{bmatrix} \right),$$

$$\begin{bmatrix}\mathbf{B}_1(\mathbf{n})\\ \mathbf{B}_2(\mathbf{n})\end{bmatrix} = \begin{bmatrix}\mathbf{L}_2'(\mathbf{n})\\ \mathbf{M}_2(\mathbf{n})\end{bmatrix}\boldsymbol{\psi}_0(\mathbf{n}), \quad \begin{bmatrix}\mathbf{t}_p(\mathbf{n})\\ \mathbf{r}_p(\mathbf{n})\end{bmatrix} = \begin{bmatrix}\mathbf{L}_2'(\mathbf{n})\\ \mathbf{M}_2(\mathbf{n})\end{bmatrix}\boldsymbol{\psi}_1(\mathbf{n})\mathbf{p}(\mathbf{n}) + \begin{bmatrix}\mathbf{t}(\mathbf{n})\\ \mathbf{r}(\mathbf{n})\end{bmatrix}.$$

Finally, defining

$$\begin{aligned}
&[\mathbf{M}(\mathfrak{n})]_{11} = \texttt{BDmat}(\mathbf{T}(n^0), \mathbf{A}_1(\mathbf{n})), \quad [\mathbf{M}(\mathfrak{n})]_{12} = \texttt{BDmat}(-\mathbf{N}'(n^0), \mathbf{A}_2(\mathbf{n})),\\
&[\mathbf{M}(\mathfrak{n})]_{21} = \texttt{BDmat}(\mathbf{N}(n^0), \mathbf{A}_3(\mathbf{n})), \quad [\mathbf{M}(\mathfrak{n})]_{22} = \texttt{BDmat}(\mathbf{0}, \mathbf{A}_4(\mathbf{n})), \text{ and}\\
&\mathbf{q}(\mathfrak{n}; x_0, \mathbf{p}(\mathbf{n})) = \texttt{Col}\Big(\mathbf{L}_2'(n^0), \mathbf{B}_1(\mathbf{n}), \mathbf{M}_2(n^0), \mathbf{B}_2(\mathbf{n})\Big)x_0^* + \texttt{Col}\Big(\mathbf{t}(n^0), \mathbf{t}_p(\mathbf{n}), \mathbf{r}(n^0), \mathbf{r}_p(\mathbf{n})\Big),
\end{aligned}$$

the linear complementarity problems defined at each node $n^t \in \mathbf{n}^t,\ t \in \mathcal{T}$ of the event tree are written as a large-scale linear complementarity problem (parametrized by x_0 and $\mathbf{p}(\mathbf{n})$) defined on the enumerated event tree $\mathfrak{n}$ as follows

$$\mathbf{M}(\mathfrak{n})\mathbf{z}(\mathfrak{n}) + \mathbf{q}(\mathfrak{n}; x_0, \mathbf{p}(\mathbf{n})) \perp \mathbf{z}(\mathfrak{n}), \quad \mathbf{z}(\mathfrak{n}) = \begin{bmatrix}\mathbf{v}^*(\mathfrak{n})\\ \boldsymbol{\mu}(\mathfrak{n})\end{bmatrix}. \tag{36}$$

A few remarks on the above complementarity problem are in order. Firstly, the matrix $\mathbf{M}(\mathfrak{n})$ may not be symmetric; a feature generally true with quadratic programming problems. Here, the loss of symmetry is attributed to the transformations involved in the derivation of (36). Next, the solvability depends on the choice of the initial state and the parameters $\mathbf{p}(\mathbf{n})$. In general, if the above problem is solvable, it may have more than one solution (see Cottle et al. (1992) for more details on the existence and uniqueness of solutions). Finally, we summarize the main results of this section as an algorithm for computing the candidate Nash equilibrium strategies.

Algorithm 1 Candidate Nash equilibrium strategies with information structure $\mathtt{I} = \{\mathtt{Co}, \mathtt{Cc}\}$

1. Using the problem parameters check if the matrices $\Lambda_{\mathtt{Co}}(\mathcal{S}(n^t), t)$ ($\Lambda_{\mathtt{Cc}}(\mathcal{S}(n^t), t)$) recursively defined by (12)–(13) ((19)–(20)) are invertible for $t = T-1, \cdots, 1$. If step 1 is true, then go to step 2 else go to step 4.
2. Check if the linear complementarity problem (36) solvable. If step 2 is true then go to step 3 else go to step 4.
3. The Nash equilibrium strategies $(\tilde{\mathbf{u}}^{*\mathtt{I}}, \tilde{\mathbf{v}}^{*\mathtt{I}})$ are given as follows:
   (a) The strategies $\tilde{u}_i^*$ for each $i \in N$
   $$u_i^*(n^t) = -R_{ii}^{-1}(n^t)B_i'(n^t)\left(K_i^{\mathtt{I}}(t)x^*(n^t) + \beta_i^{\mathtt{I}}(n^t)\right),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T$$
   where $K_i^{\mathtt{I}}(n^t)$ is given by (13) ((19)), $\beta_i^{\mathtt{I}}(n^t)$ is given by (30), and the state trajectory is given by (31).
   (b) The strategies $\tilde{v}_i^*,\ i \in \bar{N}$ are obtained directly from the solution of the problem (36).
4. Stop.

We have the following remark for sufficiency of the conditions (25)–(29).

Remark 6 *We know that every dynamic game defined over an event tree can be represented in normal form (see Haurie et al. (2012)), and the Con-LQGET can be written as*

$$\min_{\tilde{u}_i, \tilde{v}_i} J_i(x_0, \tilde{\mathbf{u}}, \tilde{\mathbf{v}}) \text{ subject to } f_i(x_0, \tilde{\mathbf{u}}, \tilde{\mathbf{v}}) \geq 0.$$

The functions $f_i(x_0, \tilde{\mathbf{u}}, \tilde{\mathbf{v}})$ represent the constraints (2) after eliminating the state variable; an equality constraint described by (1). The strategies of player i in this normal form game are $(\tilde{u}_i, \tilde{v}_i) \in \tilde{U}_i \times \tilde{V}_i$. It is shown that for convex games, see Rosen (1965), i.e., when every joint strategy space $(\tilde{U}_1 \times \tilde{V}_1) \times (\tilde{U}_2 \times \tilde{V}_2) \times \cdots \times (\tilde{U}_N \times \tilde{V}_N)$ is convex and bounded, and that each player's payoff function $J_i(x_0, \tilde{\mathbf{u}}, \tilde{\mathbf{v}})$ is convex in his own strategies $(\tilde{u}_i, \tilde{v}_i)$, then the game admits a Nash equilibrium. Further, it was shown that if the payoff functions satisfy an additional requirement called diagonal convexity, then the convex game has a unique equilibrium. So, if the normal form representation of Con-LQGET is convex, then the KKT conditions given *by* (25)–(29) *are also sufficient and the candidates computed from Algorithm 1 are indeed the Nash equilibria for Con-LQGET.*

In the remaining part of the paper, we demonstrate the application of Theorem 3.2, Theorem 3.4 and Algorithm 1 with two examples.

5 Numerical illustration

5.1 Dynamic duopoly with stochastic demand

We consider a duopoly model with a stochastic demand. Let $k_i(n^t) \in \mathbb{R}$ denote the production capability of firm i, $I_i(n^t)$ its investment control instrument, and δ_i the depreciation rate of its capital, all at node n^t of the event tree. The state equation for firm i is then

$$k_i(n^t) = (1-\delta_i)k_i(a(n^t)) + I_i(a(n^t)),\ k_i(n^0) = k_i^0,\ i = 1, 2.$$

We assume that each firm has a quadratic maintenance cost $e_i k_i^2(n^t)$ and investment cost $d_i I_i^2(n^t)$, where e_i and d_i are positive parameters. We assume that competition is à la Cournot, and that the price is given by the following stochastic inverse-demand law:

$$D(k_1(n^t), k_2(n^t)) = \alpha(n^t) - \beta(n^t)\left(k_1(n^t) + k_2(n^t)\right).$$

Player i minimizes the following cost function:[7]

$$\begin{aligned} J_i(x_0, \tilde{I}) = &\sum_{t=0}^{T-1} \sum_{n^t \in \mathcal{N}^t} \pi(n^t)\ \rho^t \left(e_i k_i^2(n^t) + d_i I_i^2(n^t) - k_i(n^t) D(k_1(n^t), k_2(n^t))\right) \\ &+ \sum_{n^T \in \mathcal{N}^T} \pi(n^T)\ \rho^T \left(v_i k_i^2(n^T) - k_i(n^T) D(k_1(n^T), k_2(n^T))\right), \end{aligned} \tag{37}$$

where ρ is the common discount factor. We recognize that the above game belongs to the LQGET class with

$$\begin{aligned} x(n^t) &= \begin{bmatrix} k_1(n^t) \\ k_2(n^t) \end{bmatrix},\ u_1(n^t) = I_1(n^t),\ u_2(n^t) = I_2(n^t), \\ A(n^t) &= \begin{bmatrix} 1-\delta_1 & 0 \\ 0 & 1-\delta_2 \end{bmatrix},\ B_1(n^t) = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ B_2(n^t) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\ Q_1(n^t) &= 2\rho^t \begin{bmatrix} e_1 + \beta(n^t) & \frac{1}{2}\beta(n^t) \\ \frac{1}{2}\beta(n^t) & 0 \end{bmatrix},\ Q_2(n^t) = 2\rho^t \begin{bmatrix} 0 & \frac{1}{2}\beta(n^t) \\ \frac{1}{2}\beta(n^t) & e_2 + \beta(n^t) \end{bmatrix},\ t \in \mathcal{T}\backslash T, \\ R_{11}(n^t) &= 2\rho^t d_1,\ R_{12}(n^t) = 0,\ R_{21}(n^t) = 0,\ R_{22}(n^t) = 2\rho^t d_2,\ t \in \mathcal{T}\backslash T, \\ p_1(n^t) &= \begin{bmatrix} -\rho^t \alpha(n^t) \\ 0 \end{bmatrix},\ p_2(n^t) = \begin{bmatrix} 0 \\ -\rho^t \alpha(n^t) \end{bmatrix},\ t \in \mathcal{T}, \\ Q_1(n^T) &= 2\rho^T \begin{bmatrix} v_1 + \beta(n^T) & \frac{1}{2}\beta(n^T) \\ \frac{1}{2}\beta(n^T) & 0 \end{bmatrix},\ Q_2(n^T) = 2\rho^T \begin{bmatrix} 0 & \frac{1}{2}\beta(n^T) \\ \frac{1}{2}\beta(n^T) & v_2 + \beta(n^T) \end{bmatrix}. \end{aligned}$$

For simplicity, we assume symmetric players and focus on the comparison of open-loop and closed-loop S-adapted equilibrium strategies. We choose the following values for the parameters: depreciation rates $\delta_1 = \delta_2 = 0.1$, maintenance costs $e_1 = e_2 = 0.25$, investment costs $d_1 = d_2 = 1$, terminal costs $v_1 = v_2 = 0.1$ and the discount factor is taken as $\rho = 0.9$ for both firms. The initial values of firms' production levels are taken as $k_1^0 = k_2^0 = 3$.

[7] Here, the LQGET and Con-LQGET problems are posed as minimization problems.

Event tree and demand function

We consider a planning horizon $T = 13$. Each node, excepting the terminal nodes, has two successors. At each node in the event tree, the demand is either at high state - H where the price is given by

$$D^H(k_1(n^t), k_2(n^t)) = 60 - 2(k_1(n^t) + k_2(n^t)),$$

or in a low state - L and the price is then given by

$$D^L(k_1(n^t), k_2(n^t)) = 40 - 2(k_1(n^t) + k_2(n^t)).$$

Consequently, the event tree consists of a total of 16,383 nodes. The conditional probabilities are generated using a Bernoulli distribution (due to the two possible fixed-market conditions). We indicate the first successor to represent the low-demand market condition and the second successor to correspond to the high-demand market condition. The S-adapted open-loop and closed-loop (no-memory) investment strategies are obtained using (14) and (21) from Theorem 3.2 and Theorem 3.4, respectively. Table 1 illustrates the S-adapted open-loop and closed-loop (no-memory) investment and the production capability trajectories for the following particular scenario:[8]

$$\{0, 2, 5, 12, 26, 54, 110, 222, 445, 892, 1785, 3571, 7143, 14287\}.$$

We observe that firms invest more, in almost all periods, when they implement closed-loop strategies than when they play with open-loop strategies. Consequently, the production capabilities of firms are higher under closed-loop strategies. This result is similar to what has been found in a number of applications of dynamic games, where feedback (so not precisely closed-loop) strategies increase competition, that is, where feedback equilibrium investments in, e.g., production capacity, R&D efforts or advertising budgets, exceed the open-loop ones (see for instance Reynolds, Dockner (1992), Driskill and McCafferty (1989), Long et al. (1999), and Driskill (2001)).

The optimized cost of firms under S-adapted open-loop and closed-loop strategies for the above scenario are calculated as -1110.7 and -1094.2 respectively. We observe that firms gain more (but less than 2%) using the S-adapted open-loop strategy, which can be explained by the above result that open-loop strategies soften competition with respect to closed-loop ones.

Table 1: S-adapted open-loop and closed-loop investment strategies for a particular scenario.

	Scenario					
Period (t)	Node label (n^t)	Demand	$I_1^{\circ}(n^t)$	$I_1^{c}(n^t)$	$k_1^{\circ}(n^t)$	$k_1^{c}(n^t)$
0	0	H	4.0139	4.1873	3.0000	3.0000
1	2	H	1.4084	1.4628	6.7139	6.8873
2	5	L	0.8934	0.9222	7.4509	7.6613
3	12	H	0.8014	0.8243	7.5992	7.8173
4	26	H	0.7104	0.7327	7.6407	7.8599
5	54	H	0.7604	0.7823	7.5871	7.8066
6	110	H	0.8149	0.8370	7.5888	7.8082
7	222	H	0.7707	0.7933	7.6448	7.8643
8	445	L	0.6746	0.6987	7.6510	7.8712
9	892	H	0.8436	0.8662	7.5605	7.7828
10	1785	L	0.9623	0.9841	7.6480	7.8707
11	3571	L	0.4881	0.4834	7.8455	8.0677
12	7143	L	0.5394	0.4100	7.5490	7.7443
13	14287	L	0	0	7.3335	7.3799

[8] The event tree is enumerated using a breadth-first search approach.

5.2 Constrained dynamic duopoly with stochastic demand

In the above example, production capacity was synonymous to output. In this example, we distinguish between these two variables. More specifically, we introduce as an additional decision variable the quantity $q_i(n^t)$ put on the market by player i at each node n^t. Like before, $k_i(n^t) \in \mathbb{R}$ denotes the production capacity of firm i, $I_i(n^t)$ its investment-control instrument, and δ_i the depreciation rate of its capital, all at node n^t of the event tree. The state equation for firm i is then

$$k_i(n^t) = \ (1-\delta_i)k_i(a(n^t)) + I_i(a(n^t)),\ k_i(n^0) = k_i^0,\ t \in \mathcal{T}\backslash 0. \tag{38}$$

At each node, player i faces the capacity constraint

$$q_i(n^t) \leq \ k_i(n^t),\ i = 1,2,\ t \in \mathcal{T}.$$

We assume that each firm has a quadratic investment cost $(d_i I_i^2(n^t))$ and a production cost proportional to its capacity $(c_i k_i(n^t) q_i(n^t))$. The stochastic inverse-demand law is linear in the total output and is given by

$$D(q_1(n^t), q_2(n^t)) = \alpha(n^t) - \beta(n^t)\left(q_1(n^t) + q_2(n^t)\right).$$

Further, we assume that the firms pay a tax, specified exogenously, proportional to their total production capacities as $\tau(n^t)(k_1(n^t)+k_2(n^t))^2$. Player i minimizes the following cost function:

$$J_i(x_0, \tilde{I}, \tilde{q}) = \sum_{t=0}^{T-1} \sum_{n^t \in \mathcal{N}^t} \pi(n^t)\ \rho^t \Big(d_i I_i^2(n^t) + c_i k_i(n^t) q_i(n^t) + \tau(n^t)(k_1(n^t)+k_2(n^t))^2 - q_i(n^t) D(q_1(n^t), q_2(n^t)) \Big)$$
$$+ \sum_{n^T \in \mathcal{N}^T} \pi(n^T)\ \rho^T \Big(v_i k_i^2(n^T) + c_i k_i(n^T) q_i(n^T) + \tau(n^T)(k_1(n^T)+k_2(n^T))^2 - q_i(n^T) D(q_1(n^T), q_2(n^T)) \Big). \tag{39}$$

It is easy to verify that this game belongs to the Con-LQGET class, with variables and parameters defined as follows:

$$x(n^t) = \begin{bmatrix} k_1(n^t) \\ k_2(n^t) \end{bmatrix},\ u_1(n^t) = I_1(n^t),\ u_2(n^t) = I_2(n^t),$$
$$A(n^t) = \begin{bmatrix} 1-\delta_1 & 0 \\ 0 & 1-\delta_2 \end{bmatrix},\ B_1(n^t) = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ B_2(n^t) = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$
$$Q_1(n^t) = 2\rho^t \tau(n^t) \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},\ Q_2(n^t) = 2\rho^t \tau(n^t) \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},\ t \in \mathcal{T}\backslash T,$$
$$R_{11}(n^t) = 2\rho^t d_1,\ R_{12}(n^t) = 0,\ R_{21}(n^t) = 0,\ R_{22}(n^t) = 2\rho^t d_2,\ t \in \mathcal{T}\backslash T,$$
$$p_1(n^t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\ p_2(n^t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\ t \in \mathcal{T},$$
$$Q_1(n^T) = 2\rho^T \begin{bmatrix} v_1 + \tau(n^T) & \tau(n^T) \\ \tau(n^T) & \tau(n^T) \end{bmatrix},\ Q_2(n^T) = 2\rho^T \begin{bmatrix} \tau(n^T) & \tau(n^T) \\ \tau(n^T) & v_2 + \tau(n^T) \end{bmatrix},$$
$$T^1(n^t) = 2\rho^t \begin{bmatrix} \beta(n^t) & \frac{1}{2}\beta(n^t) \\ \frac{1}{2}\beta(n^t) & 0 \end{bmatrix},\ T^2(n^t) = 2\rho^t \begin{bmatrix} 0 & \frac{1}{2}\beta(n^t) \\ \frac{1}{2}\beta(n^t) & \beta(n^t) \end{bmatrix},$$
$$t^1(n^t) = \rho^t \begin{bmatrix} -\alpha(n^t) \\ 0 \end{bmatrix},\ t^2(n^t) = \rho^t \begin{bmatrix} 0 \\ -\alpha(n^t) \end{bmatrix},$$
$$L^1(n^t) = \rho^t \begin{bmatrix} c_1 & 0 \\ 0 & 0 \end{bmatrix},\ L^2(n^t) = \rho^t \begin{bmatrix} 0 & 0 \\ 0 & c_2 \end{bmatrix},$$
$$M_1(n^t) = \begin{bmatrix} 1 & 0 \end{bmatrix},\ M_2(n^t) = \begin{bmatrix} 0 & 1 \end{bmatrix},\ N_1(n^t) = -1,\ N_2(n^t) = -1,\ r_1(n^t) = 0,\ r_2(n^t) = 0.$$

Again, we assume symmetric players, as the main objective is to compare the S-adapted constrained[9] open-loop and constrained closed-loop equilibrium strategies. We choose the following values for the parameters: depreciation rates $\delta_1 = \delta_2 = 0.1$, production costs $c_1 = c_2 = 0.25$, investment costs $d_1 = d_2 = 1$,

[9] Here, we recall that in the Con-LQGET model, the information structures are restricted to `Co`, where $I_i(n^t) = \gamma_i(n^t, x_0)$, $q_i(n^t) = \delta(n^t; x(n^t))$ and `Cc`, where $I_i(n^t) = \gamma_i(x(n^t); x_0)$, $q_i(n^t) = \delta_i(n^t; x(n^t))$. In both cases, the quantities are set based on the calendar date and uncertainty, i.e., node n^t of the event tree; and the feasibility (capacity) constraints are influenced by the state $x(n^t) = [k_1(n^t)\ k_2(n^t)]'$.

terminal costs $v_1 = v_2 = 0.1$, and the discount factor is taken as $\rho = 0.9$ for both firms. The initial values of firms' production levels are set at $k_1^0 = k_2^0 = 10$. The event-tree description is the same as in the previous example, that is, at any given node, demand can be in a high state - H or in a low state - L. To reduce the computational burden, we set $T = 8$. Therefore, the event tree consists of a total of 511 nodes. The high and low inverse demand functions are given by

$$D^H(q_1(n^t), q_2(n^t)) = 60 - 2(q_1(n^t) + q_2(n^t)),$$
$$D^L(q_1(n^t), q_2(n^t)) = 40 - 2(q_1(n^t) + q_2(n^t)).$$

Table 2 illustrates the S-adapted open-loop and closed-loop (no-memory) investment and production strategies for a a specific but qualitatively representative scenario $\{0, 2, 6, 14, 29, 60, 121, 244, 490\}$. As in the previous example, we note that firms invest more initially with a closed-loop strategy when compared with the open-loop strategy. Further, we observe that the production capacities of firms are higher with closed-loop strategies. However, we observe that production quantity is lower with a constrained closed-loop information structure. The optimized cost of firms under constrained S-adapted open-loop and closed-loop strategies for the above scenario are calculated as -1364.3136 and -1361.4659, respectively. Here also, we observe that firms gain (marginally) more using the S-adapted open-loop strategy.

Table 2: Constrained S-adapted open-loop and closed-loop investment strategies for a particular scenario, with the tax parameter set to $\tau(n^t) = 0.01$.

	Scenario							
Period (t)	Node label (n^t)	Demand	$I_1^{\mathsf{Co}}(n^t)$	$I_1^{\mathsf{Cc}}(n^t)$	$k_1^{\mathsf{Co}}(n^t)$	$k_1^{\mathsf{Cc}}(n^t)$	$q_1^{\mathsf{Co}}(n^t)$	$q_2^{\mathsf{Cc}}(n^t)$
0	0	H	3.5725	3.6286	10	10	9.5833	9.5833
1	2	H	3.4878	3.5314	12.5725	12.6286	9.4179	9.4158
2	6	H	3.3467	3.3773	14.8031	14.8972	9.2385	9.2346
3	14	H	3.0518	3.0698	16.6695	16.7847	9.0472	9.0421
4	29	L	2.7530	2.7601	18.0544	18.1760	5.5201	5.5143
5	60	H	2.1666	2.1649	19.0019	19.1185	8.6592	8.6532
6	121	L	1.2131	1.2058	19.2683	19.3716	5.1560	5.1503
7	244	H	0.1113	0.1021	18.5546	18.6402	8.3836	8.3785
8	490	H	0	0	16.8105	16.8783	8.3728	8.3683

A few comments on the implementation details are in order. We used the Matlab PATH solver `pathlcp`[10] for solving complementarity problem (36). The initial conditions and the parameter values used in the above examples were set to ensure the positivity of the investment variable and a unique solution of (36).

6 Conclusions

In this paper, we study linear quadratic games played over uncontrolled event trees, that is, games where the transition from one node to another is nature's decision and cannot be influenced by the players' actions. Firstly, we considered the unconstrained case (LQGET) and obtained necessary conditions for the existence of S-adapted open-loop and closed-loop Nash equilibrium strategies. We observed that these conditions are related to solvability of generalized Riccati-type backward-recursive equations defined on the event tree. Next, we consider the LQGET model with node-specific linear constraints. By restricting the information pattern as constrained S-adapted open-loop and closed-loop structure, we derived necessary conditions for the existence of the associated Nash equilibrium strategies. We observed that these necessary conditions result in a weakly coupled system of backward-recursive equations and node-specific parametric linear complementarity problems defined on the event tree. Next, using suitable reformulations, the Nash equilibrium strategies can be obtained by solving a single large-scale parametric linear complementarity problem defined on the entire event tree. We illustrate the applicability of proposed algorithms with numerical examples.

[10] Freely downloadable from http://pages.cs.wisc.edu/~ferris/path/.

A Appendix

Proof of Theorem 3.2. Since $R_{ii}(n^t) > 0$, $\mathcal{H}_i(.)$ is a strictly convex function of $u_i(n^t)$ for all $u_i(n^t) \in U_i^{n^t}$. Therefore, minimization of this Hamiltonian over $u_i(n^t) \in U_i^{n^t}$ yields the following unique relation for all $t \in \mathcal{T} \backslash T$

$$u_i^*(n^t) = -R_{ii}^{-1}(n^t)B_i'(n^t)\lambda_i(\mathcal{S}(n^t)). \tag{40}$$

We define[11] $\lambda_i(n^t) = K_i^\circ(n^t)x^*(n^t) + \beta_i^\circ(n^t),\ n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T$ due to the linear structure. Then the state equation reduces to

$$\begin{aligned} x^*(\nu) &= A(n^t)x^*(n^t) - \sum_{l \in \bar{N}} S_l(n^t) \sum_{\nu \in \mathcal{S}(n^t)} \pi_{n^t}^{\nu} \Big(K_l^\circ(\nu)x^*(\nu) + \beta_l^\circ(\nu) \Big) \\ &= A(n^t)x^*(n^t) - \sum_{\nu \in \mathcal{S}(n^t)} \pi_{n^t}^{\nu} \sum_{l \in \bar{N}} S_l(n^t)K_l^\circ(\nu)x^*(\nu) - \sum_{\nu \in \mathcal{S}(n^t)} \pi_{n^t}^{\nu} \sum_{l \in \bar{N}} S_l(n^t)\beta_l^\circ(\nu). \end{aligned} \tag{41}$$

The right-hand side of the above equation contains the expected value of the terms evaluated at the successor nodes $\nu \in \mathcal{S}(n^t)$. So, the state variable takes the same value for all the successor nodes ν of the node n^t along the optimal trajectory, that is, $x^*(\nu^1) = x^*(\nu^2),\ \forall \nu^1, \nu^2 \in \mathcal{S}(n^t)$. This observation is consistent with the one-period lag assumption in model (1). Taking the expectation on both sides of (41) with respect to the distribution $\boldsymbol{\pi}_{n^t}^{\nu}$ results in

$$x^*(\mathcal{S}(n^t)) = \sum_{\nu \in \mathcal{S}(n^t)} \pi_{n^t}^{\nu} x^*(\nu) = x^*(\nu).$$

Replacing $x^*(\nu)$ in (41) results in

$$x^*(\mathcal{S}(n^t)) = A(n^t)x^*(n^t) - \sum_{\nu \in \mathcal{S}(n^t)} \pi_{n^t}^{\nu} \sum_{l \in \bar{N}} S_l(n^t)K_l^\circ(\nu)x^*(\mathcal{S}(n^t)) - \sum_{\nu \in \mathcal{S}(n^t)} \pi_{n^t}^{\nu} \sum_{l \in \bar{N}} S_l(n^t)\beta_l^\circ(\nu).$$

Since the matrix $\Lambda_\circ(\mathcal{S}(n^t), n^t) = I + \sum_{l \in \bar{N}} S_l(n^t)K_l^\circ(\mathcal{S}(n^t))$ is assumed to be invertible, the state variable $x^*(\nu)$ is then given by

$$x^*(\nu) = x^*(\mathcal{S}(n^t)) = \Lambda_\circ^{-1}(\mathcal{S}(n^t), n^t)\Big(A(n^t)x^*(n^t) - \sum_{l \in \bar{N}} S_l(n^t)\beta_l^\circ(\mathcal{S}(n^t)) \Big). \tag{42}$$

Using the above, the costate variable at ν is written as follows:

$$\begin{aligned} \lambda_i(\nu) &= K_i^\circ(\nu)x^*(\nu) + \beta_i^\circ(\nu) \\ &= K_i^\circ(\nu)\Lambda_\circ^{-1}(\mathcal{S}(n^t), n^t)\Big(A(n^t)x^*(n^t) - \sum_{l \in \bar{N}} S_l(n^t)\beta_l^\circ(\mathcal{S}(n^t)) \Big) + \beta_i^\circ(\nu). \end{aligned} \tag{43}$$

Using the above in equation (10), we have for $n^t \in \mathbf{n}^t,\ t \in \mathcal{T}\backslash T$,

$$\begin{aligned} K_i^\circ(n^t)x^*(n^t) + \beta_i^\circ(n^t) =& \Big(Q_i(n^t) + A'(n^t)K_i^\circ(\mathcal{S}(n^t))\Lambda_\circ^{-1}(\mathcal{S}(n^t), n^t)A(n^t) \Big) x^*(n^t) \\ &+ p_i(n^t) + A'(n^t)\Big(\beta_i^\circ(\mathcal{S}(n^t)) - K_i^\circ(\mathcal{S}(n^t))\Lambda_\circ^{-1}(\mathcal{S}(n^t), n^t) \sum_{l \in \bar{N}} S_l(n^t)\beta_l^\circ(\mathcal{S}(n^t)) \Big). \end{aligned}$$

Collecting the coefficients of $x^*(n^t)$ and by the assumption (13), the relation (16) follows. The remaining statements follow from the terminal conditions (9), and using (43) in (40). □

[11] This is a standard procedure in solving linear quadratic difference games (see Başar and Olsder (1998), Engwerda (2005), Pindyck (1977)). Here, we follow the approach of (Pindyck (1977)).

Proof of Corollary 3.3. Let $\bar{\lambda}_i(n^t) = \lambda_i(n^t) - (K_i^{\mathsf{o}}(n^t)x^*(n^t) + \beta_i^{\mathsf{o}}(n^t))$ be any other solution to the two point boundary value problem. Firstly, we have

$$\begin{aligned}\bar{\lambda}_i(\mathcal{S}(n^t) &= \sum_{\nu\in\mathcal{S}(n^t)} \pi_{n^t}^{\nu}\left(\lambda_i(\nu) - (K_i^{\mathsf{o}}(\nu)x^*(\nu) + \beta_i^{\mathsf{o}}(\nu))\right) \\ &= \lambda_i(\mathcal{S}(n^t)) - \left(K_i^{\mathsf{o}}(\mathcal{S}(n^t))x^*(\mathcal{S}(n^t)) + \beta_i^{\mathsf{o}}(\mathcal{S}(n^t))\right) \text{ (follows as } x^*(\nu) = x^*(\mathcal{S}(n^t)))\end{aligned}$$

Then substituting this in (10)–(11) we write the two point boundary value problem in $(x^*, \bar{\lambda})$ coordinates as:

$$\begin{aligned}x^*(\nu) =& \Lambda_{\mathsf{o}}^{-1}(\mathcal{S}(n^t), n^t)\left(A(n^t)x^*(n^t) - \sum_{l\in\bar{N}} S_l(n^t)\bar{\lambda}_i(\mathcal{S}(n^t) - \sum_{l\in\bar{N}} S_l(n^t)\beta_i^{\mathsf{o}}(\mathcal{S}(n^t))\right),\ \nu\in\mathcal{S}(n^t) \\ \bar{\lambda}_i(n^t) =& A'(t)\bar{\lambda}_i(\mathcal{S}(n^t)) - A'(n^t)K_i^{\mathsf{o}}(\mathcal{S}(n^t))\Lambda_{\mathsf{o}}^{-1}(\mathcal{S}(n^t), n^t)\sum_{l\in\bar{N}} S_l(n^t)\bar{\lambda}_i(\mathcal{S}(n^t))\end{aligned}$$

Notice, that the above system of equations is decoupled. The terminal conditions lead to $\bar{\lambda}_i(n^T) = 0$ for all $n^T \in \mathbf{n}^T$, and as a result we have $\bar{\lambda}_i(n^t) = 0$ for all $n^t \in \mathbf{n}^t,\ t\in\mathcal{T}$. □

Proof of Theorem 3.4. The procedure for the closed-loop (no-memory) case is similar to the open-loop case, see Pindyck (1977). Here also, we assume $\lambda_i(n^t) = K_i^c(n^t)x^*(n^t) + \beta_i^c(n^t)$ due to linear structure. Since the matrix $\Lambda_{\mathsf{c}}(\mathcal{S}(n^t), n^t) = I + \sum_{l\in\bar{N}} S_l(n^t)K_l^c(\mathcal{S}(n^t))$ is invertible, with straightforward algebraic manipulations, we can show that

$$x^*(\nu) = x^*(\mathcal{S}(n^t)) = \Lambda_{\mathsf{c}}^{-1}(\mathcal{S}(n^t), n^t)\Big(A(n^t)x^*(n^t) - \sum_{l\in\bar{N}} S_l(n^t)\beta_l^{\mathsf{c}}(\mathcal{S}(n^t))\Big),$$

$$u_j^*(n^t) = -R_{jj}^{-1}(n^t)B_j'(n^t)\lambda_j(\mathcal{S}(n^t)), \tag{44}$$

$$\frac{\partial u_j^*(n^t)}{\partial x^*(n^t)} = -R_{jj}^{-1}(n^t)B_j'(n^t)K_j^{\mathsf{c}}(\mathcal{S}(n^t))\Lambda_{\mathsf{c}}^{-1}(\mathcal{S}(n^t), n^t)A(n^t). \tag{45}$$

Next define $S_{ij}(n^t) = B_j(n^t)R_{jj}^{-1}(n^t)R_{ij}(n^t)R_{jj}^{-1}(n^t)B_j'(n^t),\ i\neq j$ and a matrix $\Theta(\mathcal{S}(n^t), n^t)$ as

$$\Theta_{ij}(\mathcal{S}(n^t), n^t) = \begin{cases} I - \sum_{j\in\bar{N}\setminus i} S_j(n^t)K_j^{\mathsf{c}}(\mathcal{S}(n^t))\Lambda_{\mathsf{c}}^{-1}(\mathcal{S}(n^t), n^t) & i=j \\ S_{ij}(n^t)K_j^{\mathsf{c}}(\mathcal{S}(n^t))\Lambda_{\mathsf{c}}^{-1}(\mathcal{S}(n^t), n^t) & i\neq j \end{cases}.$$

Next, substituting the above relations in (18) and from assumption (20), the relation (22) follows.[12] The remaining statements follow from the terminal conditions (9). □

Proof of Theorem 4.1. The Lagrangian given by (24) has a special form (relating to the Lagrangian (5)) due to the separable cost structure of the objective. Like before, we use Hamiltonian (6) to express the term $\mathcal{L}_i(\tilde{x}, (u_i, \tilde{u}_{-i}^*), \tilde{\lambda}_i)$. We have $\frac{\partial v_i(n^t)}{\partial x(n^t)} = 0$ due to the assumed information structure $\mathtt{I} = \{\mathtt{Co}, \mathtt{Cc}\}$. Then, we obtain the necessary conditions (25)–(27) by equating to zero the partial derivatives of the Lagrangian (24), with respect to $u_i(n^t)$ evaluated at $u^*(n^t)$, and $x_i(n^t)$ evaluated at $x^*(n^t)$. Again from information structure $\mathtt{I}$ and the separable cost structure, players at node n^t minimize a parametrized quadratic programming problem. The Karush-Kuhn-Tucker conditions for $v_i^*(n^t)$ to be the minimizer, at each node $n^t\in\mathbf{n}^t,\ t\in\mathcal{T}$, are given by

$$\begin{aligned}&\alpha_i(n^t) = \sum_{l\in\bar{N}} T_{il}^i(n^t)v_l^*(n^t) + L_i^{i\prime}(n^t)x^*(n^t) - N_i'(n^t)\mu_i(n^t) + t_i^i(n^t) \geq 0,\ v_i^*(n^t)\geq 0,\ \alpha_i'(n^t)v_i^*(n^t) = 0, \\ &\beta_i(n^t) = M_i(n^t)x^*(n^t) + N_i(n^t)v_i^*(n^t) + r_i(n^t) \geq 0,\ \mu_i(n^t)\geq 0,\ \beta_i'(n^t)\mu_i(n^t) = 0.\end{aligned}$$

The necessary conditions (28)–(29) correspond to the linear complementarity formulation of the above necessary conditions. □

[12] We skip the lengthy straightforward calculations as they are similar to the open-loop case.

References

[1] T. Başar and G.J. Olsder, (1998). *Dynamic Noncooperative Game Theory*, Academic Press, New York.

[2] T. Başar and Q. Zhu, (2011). Prices of anarchy, information, and cooperation in differential games, *Dynamic Games and Applications*, 1, 1, 50–73.

[3] J. Blomvall and P.O. Lindberg, (2002a). A Riccati-based primal interior point solver for multistage stochastic programming, *European Journal of Operational Research* 143, 2, 452–461.

[4] J. Blomvall and P.O. Lindberg, (2002b). A Riccati-based primal interior point solver for multistage stochastic programming – extensions, *Optimization Methods and Software* 17, 3, 383–407.

[5] R.W. Cottle, J.S. Pang and R.E. Stone, (1992). *The Linear Complementarity Problem*, SIAM Classics in Applied Mathematics.

[6] E.J. Dockner, (1992). A dynamic theory of conjectural variations, *The Journal of Industrial Economics* 40, 4, 377–395.

[7] R.A. Driskill, (2001). Durable goods oligopoly, *International Journal of Industrial Organization* 19, 391–413.

[8] R.A. Driskill, S. McCafferty, (1989). Dynamic duopoly with adjustment costs: A differential game approach, *Journal of Economic Theory* 49, 324–338.

[9] J.C. Engwerda, (2005). *LQ Dynamic Optimization and Differential Games*, John Wiley & Sons.

[10] T.S. Genc, S.S. Reynolds and S. Sen, (2007). Dynamic oligopolistic games under uncertainty: A stochastic programming approach, *Journal of Economic Dynamics and Control*, 31, 1, 55–80.

[11] T. Genc and S. Sen (2008). An Analysis of Capacity and Price Trajectories for the Ontario Electricity Market Using Dynamic Nash Equilibrium under Uncertainty. Energy Economics, 30, 1, 173–191.

[12] A. Haurie, J.B. Krawczyk and G. Zaccour, (2012). *Games And Dynamic Games*, World Scientific Publishing.

[13] A. Haurie, G. Zaccour and Y. Smeers, (1990). Stochastic equilibrium programming for dynamic oligopolistic markets, *Journal of Optimization Theory and Applications*, 66, 243–253.

[14] G. Jank and H. Abou-Kandil, (2003). Existence and Uniqueness of Open-Loop Nash Equilibria in Linear-Quadratic Discrete Time Games. *IEEE Transactions on Automatic Control*, 48, 2, 267–271.

[15] N.V. Long, K. Shimomura, H. Takahashi, (1999). Comparing open-loop with Markov equilibria in a class of differential games, *Japanese Economic Review* 50, 4, 457–469.

[16] D.L. Lukes, (1971). Equilibrium Feedback Control in Linear Games with Quadratic Costs, *SIAM Journal on Control* 9, 2, 234–252.

[17] R. Pindyck, (1977). Optimal economic stabilization policies under decentralized control and conflicting objectives, *IEEE Transactions on Automatic Control*, 22, 517–530.

[18] P. Pineau and P. Murto, (2003). An oligopolistic investment model of the Finnish electricity market, *Annals of Operations Research*, 121, 123–148.

[19] P.V. Reddy, E. Shevkoplyas and G. Zaccour, (2013). Time-consistent Shapley value for games played over event trees, *Automatica* 49, 6, 1521–1527.

[20] S. Reynolds, (1987). Capacity Investment, Preemption and Commitment in an Infinite Horizon Model, *International Economic Review* 28, 69–88.

[21] R.T. Rockafellar and J.B. Wets, (1990). Generalized linear-quadratic problems of deterministic and stochastic optimal control in discrete time, *SIAM Journal on Control Optimization*, 28, 810–822.

[22] J.B. Rosen, (1965). Existence and uniqueness of equilibrium points for concave N person games, *Econometrica*, 33, 3, 520–534.

[23] D.H. Salinger and R.T. Rockafellar, (1999). Dynamic Splitting: An Algorithm for Deterministic and Stochastic Multiperiod Optimization. Department of Mathematics, University of Washington, Seattle, *working paper*.

[24] G. Zaccour, (1987). Théorie des jeux et marchés énergétiques : marché européen de gaz naturel et échanges d'électricité, Ph.D. Thesis, HEC, Montréal, Canada.