Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

MARC C. STEINBACH

Tree-Sparse Convex Programs

ZIB-Report 01-08 (April 2001)

TREE-SPARSE CONVEX PROGRAMS

MARC C. STEINBACH

ABSTRACT. Dynamic stochastic programs are prototypical for optimization problems with an inherent tree structure inducing characteristic sparsity patterns in the KKT systems of interior methods. We propose an integrated modeling and solution approach for such tree-sparse programs. Three closely related natural formulations are theoretically analyzed from a control-theoretic perspective and compared to each other. Associated KKT system solution algorithms with linear complexity are developed and comparisons to other interior approaches and related problem formulations are discussed.

0. INTRODUCTION

The current paper studies convex programs with an underlying tree topology, such as discrete-time stochastic control problems. We propose an integrated natural modeling and solution framework for this class of large, *tree-sparse* optimization problems.

Dynamic control problems are characterized by an inherent recourse structure. The proposed modeling is natural in the sense that constraints are categorized according to their control-theoretic interpretation, namely as dynamic equations (in which the recourse structure manifests itself), local constraints, and global constraints; the latter two categories have subcategories covering all kinds of boundary conditions. Furthermore, natural regularity assumptions are associated with each (sub)category of constraints.

Our principal interest here is in the *algebraic* structure of the KKT systems arising in standard interior methods. (Stochastic aspects will be discussed elsewhere in full detail.) Extending earlier work [28, 30, 31, 32] with promising computational results, we develop a thorough theoretical understanding of these tree-sparse linear indefinite KKT systems, with accent on the hierarchical constraint structure of three principal variants differing in the formulation of dynamics. The theoretical analysis leads to natural solution algorithms which combine a dynamic recursion with local projections for local constraints and a Schur complement approach for global constraints, giving linear complexity in the tree size.

Other interior approaches for stochastic programs include [2, 6, 9, 11, 13, 20, 23] (two-stage LP case) and [5, 12, 19, 27] (linear or convex multistage case). We compare our framework with these approaches and with the generalized linear-quadratic control formulations developed by Rockafellar [24, 25] and Rockafellar and Wets [26].

The material is organized as follows. After recalling basic facts on convex programs, trees, and interior methods, we present in Section 2 the tree-sparse problem classes along with regularity conditions and selected references to examples and applications. A detailed technical comparison is provided in Section 3, and the KKT solution algorithms are discussed in Section 4. Sections 5 and 6 investigate the respective relations of our approach to other interior methods and to generalized linear-quadratic control problems. Final remarks and indications of future research in Section 7 conclude the paper.

2000 *Mathematics Subject Classification.* 90C15, 90C06, 90C25, 65F50, 15A23 .

Key words and phrases. Convex program, tree, discrete-time optimal control, dynamic stochastic program, recursive factorization, local projection.

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) under grant STE 566/1-1.

1. PRELIMINARIES

Recalling basic facts on convex programs, trees, and interior methods, sections 1.1–1.3 introduce the general setting and notation adopted in this paper. Further details are given in earlier work of the author, particularly [32], and in the cited literature. For background material on convex optimization see, e.g., Stoer and Witzgall [35]; we also make frequent use of empty matrices and vectors in the sense of [35]: $A \in \mathbb{R}^{0\times n}$, $A \in \mathbb{R}^{n\times 0}$ or $x \in \mathbb{R}^0$.

1.1. **Convex Programs.** Consider a smooth convex program (CP) with polyhedral constraints specified as equalities and lower and upper bound and range inequalities,

$$\min_y \phi(y) \qquad \text{s.t.} \qquad Ay + a = 0, \quad By \in [r_l, r_u], \quad y \in [b_l, b_u], \tag{1}$$

where $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$ with $\nabla^2\phi(y) \ge 0\ \forall y \in \mathbb{R}^n$, and $A \in \mathbb{R}^{l\times n}$, $B \in \mathbb{R}^{k\times n}$, $l \le n$.

Notational convention. As described in [32], the values $b_l^\nu, r_l^\kappa = -\infty$ and $b_u^\nu, r_u^\kappa = +\infty$ are formally allowed to indicate absent limit components. Rigorously, we maintain index sets $\mathcal{B}_l, \mathcal{B}_u \subseteq \{1,\dots,n\}$ and $\mathcal{R}_l, \mathcal{R}_u \subseteq \{1,\dots,k\}$ that indicate which limits are present. Associated with a set $\mathcal{B}$ are gather and scatter matrices $P_\mathcal{B} \in \mathbb{R}^{|\mathcal{B}|\times n}$ and $P_\mathcal{B}^* \in \mathbb{R}^{n\times|\mathcal{B}|}$ such that $I - P_\mathcal{B}^* P_\mathcal{B}$ is the projection onto the null space $N(P_\mathcal{B})$, and $P_\mathcal{B} P_\mathcal{B}^* = I$ on $\mathbb{R}^{|\mathcal{B}|}$.

Regularity assumptions. Consider the affine subspace $\mathcal{F}_{\text{eq}} := \{y \in \mathbb{R}^n : Ay + a = 0\}$ and the polyhedron $\mathcal{F}_{\text{lim}} := \{y \in [b_l, b_u] : By \in [r_l, r_u]\}$, and denote by $\mathcal{F} := \mathcal{F}_{\text{eq}} \cap \mathcal{F}_{\text{lim}}$ the feasible set of (1). Throughout the paper we make the following assumptions.

(A0) $\mathcal{F}$ has nonempty relative interior with respect to $\mathcal{F}_{\text{eq}}$, that is, $\operatorname{int}(\mathcal{F}_{\text{lim}}) \cap \mathcal{F}_{\text{eq}} \ne \emptyset$.
(A1) A has full rank. (Equivalently $N(A^*) = \{0\}$ since $l \le n$.)
(A2) $\nabla^2\phi(y)|\mathcal{N} \ge \epsilon I > 0$ for all $y \in \mathcal{F}$, where $\mathcal{N}$ is the null space

$$\mathcal{N} := N(A) \cap N(P_{\mathcal{B}_l\cup\mathcal{B}_u}) \cap N(P_{\mathcal{R}_l\cup\mathcal{R}_u} B),$$

and $\nabla^2\phi(y)|\mathcal{N} : \mathcal{N} \to \mathcal{N}$ denotes the *projected Hessian*.

These conditions are tailored toward the barrier problems arising in an interior point framework; cf. [32]. They do not imply existence or uniqueness of solutions of (1), since strong convexity (A2) is only required on the *lineality space* of $\mathcal{F}$, i.e., the largest linear subspace *inside* the recession cone $\operatorname{rec}(\mathcal{F})$. The CP itself may have multiple solutions ($\min_{y\ge 0} 0$) or a finite infimum that is not attained ($\min_{y\ge 0} e^{-y}$), or it may be unbounded ($\min_{y\ge 0} -y$). By standard results in convex optimization, each solution of the CP is a global minimum, and the set $\mathcal{S}$ of all such solutions is convex. (More generally, every level set $N_c := \{y \in \mathcal{F} : \phi(y) \le c\}$ is convex.) We are primarily interested in the case where $\mathcal{S}$ is nonempty and bounded (hence compact) which is guaranteed, e.g., under an additional growth condition,

(A3) $\phi(y^{(k)}) \to \infty$ for every sequence $y^{(k)} \in \mathcal{F}$ with $\|y^{(k)}\| \to \infty$.

This holds, for instance, if $\mathcal{F}$ is bounded or if $\mathcal{N}$ in (A2) is replaced by the linear hull of the recession cone, $\mathcal{N}' := N(A) \cap N(P_{\mathcal{B}_l\cap\mathcal{B}_u}) \cap N(P_{\mathcal{R}_l\cap\mathcal{R}_u} B) \supseteq \operatorname{rec}(\mathcal{F})$.

1.2. **Trees.** The problem classes studied in this paper are characterized by the presence of an underlying tree topology, such as a *scenario tree* in stochastic optimization. Let V denote the set of nodes (or *vertices*) of the tree, $L_t \subseteq V$ the level set of nodes at depth t, and L the set of leaves; further $0 \in L_0$ the root, $j \in L_t$ the "current" node, $S(j)$ its set of successors, $i \equiv \pi(j)$ its unique predecessor (if $t > 0$), and $\Pi(j) = \{0,\dots,i,j\}$ the unique path from the root to j. Finally define $V^* := V \setminus \{0\}$. The subtree rooted in j has respective vertex set, level sets and leaves $V(j)$, $L_t(j)$, and $L(j)$. Below, the vertex set is often taken to be $V = \{0, 1, \dots, N\}$ where nodes are numbered in any ascending order; cf. Fig. 1.

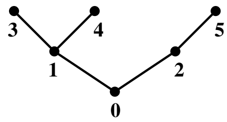

FIGURE 1. A small tree.

1.3. **Interior Methods.** Interior methods are well known and extensively covered in the literature (see [37, 39] and references therein). The basic concept consists in converting inequality constraints to barrier terms and solving a sequence of barrier subproblems whose solutions converge to a solution of the original problem. Barrier subproblems involve the calculation of a step direction from a reduced KKT system (or *augmented system*)

$$\begin{bmatrix} H+\Phi & A^* & B^* \\ A & & \\ B & & -\Psi^{-1} \end{bmatrix} \begin{bmatrix} \Delta y \\ -\Delta z \\ -\Delta \tilde{v} \end{bmatrix} = \begin{bmatrix} \tilde{f} \\ \tilde{a} \\ \tilde{r} \end{bmatrix} \iff \begin{bmatrix} \bar{H} & A^* \\ A & \end{bmatrix} \begin{bmatrix} \Delta y \\ -\Delta z \end{bmatrix} = \begin{bmatrix} \bar{f} \\ \bar{a} \end{bmatrix}. \tag{2}$$

Here $\bar{H} := H + \Phi + B^*\Psi B$ is composed from the current Hessian $H := \nabla^2\phi(y)$ and nonnegative diagonal matrices Φ, Ψ whose entries depend on the type of interior method and on primal and/or dual slacks of the current iterate. Details for the convex program (1) are provided in [32, §3]. In particular, it is proved that under conditions (A0)–(A2) the reduced KKT matrix (2) is nonsingular and each barrier subproblem has a unique solution.

The early literature treats the LP case $\min_y\{f^*y\colon Ay = a, y \geq 0\}$ (where $\bar{H} \equiv \Phi > 0$) and solves (2) by a standard *Schur complement* approach: after eliminating Δy formally, the dual step Δz is determined by the positive definite system of *normal equations*

$$(A\bar{H}^{-1}A^*)\Delta z = (\bar{a} - A\bar{H}^{-1}\bar{f}), \tag{3}$$

which is solved by a Cholesky factorization of the Schur complement $A\bar{H}^{-1}A^*$.

2. PROBLEM CLASSES

We consider three variants of tree-sparse programs differing in the form of dynamic equations: *explicit* dynamics with *outgoing* and *incoming* control, and *implicit* dynamics. Given a tree, there are local decision variables $y_j \in \mathcal{Y}_j$ in all nodes $j \in V$. In both explicit formulations, y_j consists of a state $x_j \in \mathcal{X}_j$ and a control $u_j \in \mathcal{U}_j$; the implicit variant lacks such a partitioning. Simplified basic versions of the following problems have been introduced in [30, 31]; detailed investigations of certain special cases are found in [28, 32]. Here we present the complete formulations in full generality.

2.1. **Outgoing Control.** The general tree-sparse CP with outgoing control reads

$$\min_{x,u} \quad \sum_{j\in V} \phi_j(x_j, u_j) \tag{4}$$

$$\text{s.t.} \quad x_j = G_j x_i + E_j u_i + h_j \qquad \forall j \in V, \tag{5}$$

$$F_j^x x_j + e_j^x = 0 \qquad \forall j \in V^*, \tag{6}$$

$$D_j^u u_j + e_j^u = 0 \qquad \forall j \in V, \tag{7}$$

$$F_j^c x_j + D_j^c u_j + e_j^c = 0 \qquad \forall j \in V^*, \tag{8}$$

$$F_j^r x_j + D_j^r u_j \in [r_{lj}, r_{uj}] \qquad \forall j \in V, \tag{9}$$

$$x_j \in [b_{lj}^x, b_{uj}^x] \qquad \forall j \in V^*, \tag{10}$$

$$u_j \in [b_{lj}^u, b_{uj}^u] \qquad \forall j \in V, \tag{11}$$

$$\sum_{j\in V} (F_j x_j + D_j u_j + e_j) = 0. \tag{12}$$

Apart from the node-wise separable objective (4) and dynamic equations (5), the problem includes *state* constraints (6), *control* constraints (7), *mixed* (or *coupled*) constraints (8), *range* constraints (9), *bound* constraints (10), (11), and *global* constraints (12). Except for global constraints (which potentially couple all nodes of the tree) and dynamic equations, all those constraints are *local* in the sense that they involve only variables of a single node. In the stochastic case, (4) and (12) represent *expectations*.

Let us collect node variables $y := (x_0, u_0, \dots, x_N, u_N)$ and local blocks

$$F_j^l := \begin{pmatrix} F_j^x & \\ & D_j^u \\ F_j^c & D_j^c \end{pmatrix}, \qquad B_j := \begin{pmatrix} F_j^r & D_j^r \end{pmatrix}. \tag{13}$$

The constraint matrices of (1) can then be written

$$A = \begin{bmatrix} G \\ F^l \\ F \end{bmatrix}, \qquad B = \operatorname{Diag}(B_0, \dots, B_N) \equiv \begin{bmatrix} B_0 & & \\ & \ddots & \\ & & B_N \end{bmatrix}, \tag{14}$$

where $F^l := \operatorname{Diag}(F_0^l, \dots, F_N^l)$, $F := \begin{pmatrix} F_0 & D_0 & \dots & F_N & D_N \end{pmatrix}$, and the structure of G reflects the topology of the particular tree. For instance, the example tree in Fig. 1 yields

$$G = \begin{bmatrix} -I & 0 & & & & & & & & & & \\ G_1 & E_1 & -I & 0 & & & & & & & & \\ G_2 & E_2 & & & -I & 0 & & & & & & \\ & & G_3 & E_3 & & & -I & 0 & & & & \\ & & G_4 & E_4 & & & & & -I & 0 & & \\ & & & & G_5 & E_5 & & & & & -I & 0 \end{bmatrix}. \tag{15}$$

Dynamics. By (5), every non-root state x_j depends explicitly on the preceding state and control, $x_i \equiv x_{\pi(j)}$ and $u_i \equiv u_{\pi(j)}$. This is the common formulation in the special case of deterministic control problems (where the tree reduces to a chain). We refer to it as *outgoing* control since all siblings $k \in S(j)$ are influenced by the predecessor's control u_j. In the stochastic case, the natural interpretation is that decision u_t is based on complete information at time t but becomes effective at time $t+1$, possibly due to a delay in observing the system, in implementing the decision, or in the dynamic system itself.

The explicit nature of dynamics implies that the range of G is the global state space and the null space of G is isomorphic to the global control space. More precisely,

$$R(G) = R(G|(\mathcal{X} \times \{0\})) = \mathcal{X} := \prod_{j \in V} \mathcal{X}_j, \qquad N(G) \cong \mathcal{U} := \prod_{j \in V} \mathcal{U}_j. \tag{16}$$

Thus, the (independent) control u represents all degrees of freedom in the system, and the (dependent) state x is uniquely determined by the control[1].

The recursive dynamic structure has an important consequence: If any *state* components are fixed by *local* constraints (6) or (8), this implies restrictions of the form (8) in the *preceding* node. For instance, $x_j = \hat{x}_j$ yields $G_j x_i + E_j u_i + (h_j - \hat{x}_j) = 0$. Ultimately, all conditions in node j have to be met by appropriate choice of controls u_i, $i \in \Pi(j)$. Our recursive solution algorithm makes use of precisely this backward effect in the handling of local constraints; details are given in §4, or in [28] for the deterministic case.

Dimensions. Let n_j^x, n_j^u denote the dimensions of $\mathcal{X}_j, \mathcal{U}_j$, respectively, and $l_j^x, l_j^u, l_j^c, l_j^r$ and m the dimensions of constraints (6)–(9) and (12). Each of them is allowed to be zero. In view of the specific matrix structure (14), the simple global restriction $l \le n$ is refined

[1] The root plays a special role since dynamics act as an *initial condition* here, $x_0 = h_0$, where formally $x_{\pi(0)}, u_{\pi(0)} \in \mathbb{R}^0$. Free components of x_0 may be modeled by prepending an artificial node with empty states and suitable controls. Alternatively one could drop dynamics (5) in the root and include (6), (8), and (10) instead. In the latter case x_0 has the nature of a control variable; therefore we prefer the given formulation.

hierarchically by imposing suitable local and global restrictions on the above dimensions of equality constraints,

$$l_j^x \le n_j^x, \qquad l_j^u \le n_j^u, \qquad l_j^x + l_j^u + \tilde{l}_j^c \le n_j^x + n_j^u, \tag{17}$$

$$m \le n^u - (l^x + l^u + l^c) := \sum_{j \in V} (n_j^u - l_j^x - l_j^u - l_j^c). \tag{18}$$

Here $\tilde{l}_j^c$ counts mixed constraints plus the *minimal* number of implied constraints from all successor nodes in the subtree $V(j)$; it is recursively defined as

$$\tilde{l}_j^c := l_j^c + \sum_{k \in S(j)} [l_k^x + \max(l_k^u + \tilde{l}_k^c - n_k^u, 0)].$$

In *stochastic* programs it is typical that backward effects do *not* occur: otherwise the number of implied conditions in each node might depend on the number of successors, $|S(j)|$, and hence a finer discretization of the probability space might render the problem infeasible due to a lack of local degrees of freedom (unless n_j^u increases with $|S(j)|$ as well). If implied constraints are to be excluded, we must be able to satisfy all local conditions by local control variables in the same node; then (17), (18) simplify to the stronger restrictions

$$l_j^x = 0, \qquad l_j^u + l_j^c \le n_j^u, \qquad m \le n^u - (l^u + l^c). \tag{19}$$

Note finally that the effective dimension $n_0^x = 0$ must be used in (17) since $x_0 \equiv h_0$ is always fixed. This explains why conditions (6), (8), (10) are absent in the root: (6) and (10) are meaningless, and (8) is subsumed under (7).

Regularity assumptions. Assumption (A0) from the general CP is kept literally but (A1) is refined hierarchically and (A2) is slightly strengthened:

($\mathrm{A1.1_{out}}$) $\forall j \in V$: F_j^x has full rank (= l_j^x).
($\mathrm{A1.2_{out}}$) $\forall j \in V$: D_j^u has full rank (= l_j^u).
($\mathrm{A1.3_{out}}$) $\forall j \in V$: $(F_j^c\ D_j^c)|(N(F_j^x) \times N(D_j^u))$ has full rank (= l_j^c).
($\mathrm{A1.4_{out}}$) $G|N(F^l)$ has full rank (= n^x).
($\mathrm{A1.5_{out}}$) $F|(N(F^l) \cap N(G))$ has full rank (= m).
($\mathrm{A2_{out}}$) $\forall y \in \mathcal{F}$: $\nabla^2\phi(y)|\mathcal{N}^* \ge \epsilon I > 0$ where

$$\mathcal{N}^* := N(P_{\mathcal{B}_l \cup \mathcal{B}_u}) \cap N(P_{\mathcal{R}_l \cup \mathcal{R}_u} B) \cap N(F^l) \cap N(G).$$

Lemma 1. *The following properties hold.*

(a) *Conditions* ($\mathrm{A1.1_{out}}$)–($\mathrm{A1.3_{out}}$) *are equivalent to full rank of* F^l.
(b) *Conditions* ($\mathrm{A1.1_{out}}$)–($\mathrm{A1.5_{out}}$) *are equivalent to full rank of* A.
(c) *Condition* ($\mathrm{A2_{out}}$) *implies* (A2).
(d) *Conditions* ($\mathrm{A1.1_{out}}$)–($\mathrm{A2_{out}}$) *are equivalent with* (A0)–(A2) *if and only if global constraints are absent, that is, if* $m = 0$.

Proof. Counting the relevant degrees of freedom in ($\mathrm{A1.1_{out}}$)–($\mathrm{A1.5_{out}}$) shows that full rank is indeed always equivalent to full *row* rank (as indicated). Statement (a) is now trivial and leads readily to statement (b). Statement (c) holds since $\mathcal{N} = \mathcal{N}^* \cap N(F)$. Now (d) is an immediate consequence of (a)–(c). ☐

Remarks. The conditions above are specified in terms of node quantities where possible; the combination of local and global conditions reflects the hierarchical problem structure. The strengthened convexity condition ($\mathrm{A2_{out}}$) ensures that (A2) still holds with global constraints dropped, which enables our recursive solution algorithm to employ a natural sparsity-preserving pivot order. That condition can be replaced by a considerably stronger set of *local* conditions on the control parts of individual Hessian blocks: $\forall y \in \mathcal{F}$: $\forall j \in V$: $\nabla^2_{u_j u_j}\phi_j(x_j, u_j)|\mathcal{N}_j^u \ge \epsilon_j I > 0$ where

$$\mathcal{N}_j^u := N(P_{\mathcal{B}_{lj}^u \cup \mathcal{B}_{uj}^u}) \cap N(P_{\mathcal{R}_{lj} \cup \mathcal{R}_{uj}} D_j^r) \cap N(D_j^u) \cap N(D_j^c).$$

Furthermore, ($A1.4_{\text{out}}$)–($A2_{\text{out}}$) can be expressed by *equivalent* local conditions which, however, involve intermediate results of the recursive factorization. Those equivalent conditions are of course checked by the algorithm. Backward effects will not occur if the necessary condition (19) holds and ($A1.3_{\text{out}}$) is replaced by ($A1.3'_{\text{out}}$): $\forall j \in V$: $D^c_j|N(D^u_j)$ has full rank ($= l^c_j$); cf. Theorem 1.

Examples. Outgoing control formulations are rarely used in stochastic optimization. Two exceptions are the financial model in [12] and the process engineering problem in [18].

2.2. **Incoming Control.** The general tree-sparse CP with incoming control reads

$$\min_{u,x} \quad \sum_{j \in V} \phi_{ij}(x_i, u_j) + \phi_j(x_j) \tag{20}$$

$$\text{s.t.} \quad x_j = G_j x_i + E_j u_j + h_j \qquad \forall j \in V, \tag{21}$$

$$D^u_j u_j + e^u_j = 0 \qquad \forall j \in V, \tag{22}$$

$$F^x_j x_j + e^x_j = 0 \qquad \forall j \in V, \tag{23}$$

$$F^c_{ij} x_i + D^c_j u_j + e^c_j = 0 \qquad \forall j \in V, \tag{24}$$

$$F^r_{ij} x_i + D^r_j u_j \in [r^u_{lj}, r^u_{uj}] \qquad \forall j \in V, \tag{25}$$

$$F^r_j x_j \in [r^x_{lj}, r^x_{uj}] \qquad \forall j \in V, \tag{26}$$

$$u_j \in [b^u_{lj}, b^u_{uj}] \qquad \forall j \in V, \tag{27}$$

$$x_j \in [b^x_{lj}, b^x_{uj}] \qquad \forall j \in V, \tag{28}$$

$$\sum_{j \in V} (D_j u_j + F_j x_j + e_j) = 0. \tag{29}$$

Here we rearrange node variables as $y := (u_0, x_0, \dots, u_N, x_N)$ and local blocks as

$$F^l_j := \begin{pmatrix} D^u_j & \\ D^c_j & \\ & F^x_j \end{pmatrix}, \qquad B_j := \begin{pmatrix} D^r_j & \\ & F^r_j \end{pmatrix}.$$

Due to the coupling of u_j with x_i in (20), (21), (24), (25), the tree structure appears now in all constraint matrices: B has blocks B_j along the diagonal, but an additional block F^r_{jk} below F^r_j and left of D^r_k for each $k \in S(j)$. Likewise, F^l has blocks F^l_j along the diagonal and additional blocks F^c_{jk} below F^x_j and left of D^c_k. The representation (14) for A remains valid, where $F := \begin{pmatrix} D_0 & F_0 & \dots & D_N & F_N \end{pmatrix}$ and, for the example tree in Fig. 1,

$$G = \begin{bmatrix} E_0 & -I & & & & & & & & & & \\ & G_1 & E_1 & -I & & & & & & & & \\ & G_2 & & & E_2 & -I & & & & & & \\ & & & G_3 & & & E_3 & -I & & & & \\ & & & G_4 & & & & & E_4 & -I & & \\ & & & & & G_5 & & & & & E_5 & -I \end{bmatrix}.$$

Dynamics. States x_j depend explicitly on the parent state x_i but on the current control u_j, hence we speak of *incoming* control. In the stochastic case, the natural interpretation is that decision u_t is based on complete information at time t and takes effect immediately. As before, the explicit dependence implies $R(G) = R(G|(\{0\} \times \mathcal{X})) = \mathcal{X}$ and $N(G) \cong \mathcal{U}$. Note that, since each state has "its own" control now, the root state is not necessarily fixed, and local state constraints do not necessarily cause backward effects. Mixed local constraints (24) involve x_i, u_j rather than u_j, x_j since state constraints (23) generate implied constraints of the form (24) in the *current* node. In turn, coupled constraints (24) may imply additional state constraints (23) in the *preceding* node. Details will be given in §4. When controls u_k, $k \in S(j)$ are seen as one outgoing control in j, the problem is effectively converted to outgoing control form. This becomes apparent in the regularity conditions; details will be provided in Theorem 3. In the deterministic case, both variants are identical up to an index shift.

Dimensions. The respective dimensions of variables and constraints are denoted n_j^u, n_j^x and $l_j^u, l_j^x, l_j^c, l_j^{ru}, l_j^{rx}, m$. Dimension restrictions now read

$$l_j^u \le n_j^u, \qquad \tilde{l}_j^x \le n_j^x, \qquad l_j^u + \tilde{l}_j^c \le n_j^u + n_j^x, \qquad m \le n^u - (l^u + l^x + l^c),$$

where

$$\tilde{l}_j^x := l_j^x + \sum_{k \in S(j)} \max(l_k^u + \tilde{l}_k^c - n_k^u, 0), \qquad \tilde{l}_j^c := l_j^c + \tilde{l}_j^x.$$

The simplified case without backward effects (i.e., $\tilde{l}_j^x = l_j^x$) requires restrictions

$$l_j^x \le n_j^x, \qquad l_j^u + \tilde{l}_j^c \le n_j^u, \qquad m \le n^u - (l^u + l^x + l^c). \tag{30}$$

A further simplification with no implied constraints at all (not even in the same node, i.e., $\tilde{l}_j^c = l_j^c$) requires

$$l_j^x = 0, \qquad l_j^u + l_j^c \le n_j^u, \qquad m \le n^u - (l^u + l^c).$$

Regularity assumptions. For $j \in V$ and $i \equiv \pi(j)$ let $S(i) = \{j_1, \dots, j_s\}$ (= $\{0\}$ if $j = 0$),

$$F_{i,S(i)}^c := \begin{pmatrix} F_{ij_1}^c \\ \vdots \\ F_{ij_s}^c \end{pmatrix}, \quad D_{S(i)}^c := \begin{pmatrix} D_{j_1}^c & & \\ & \ddots & \\ & & D_{j_s}^c \end{pmatrix}, \quad l_{S(i)}^c := \sum_{j \in S(i)} l_j^c,$$

and define $D_{S(i)}^u$ like $D_{S(i)}^c$. As in the previous case, assumption (A0) is now kept literally, (A1) is refined hierarchically, and (A2) is slightly strengthened:

(A1.1$_{\text{in}}$) $\forall j \in V$: D_j^u has full rank (= l_j^u).
(A1.2$_{\text{in}}$) $\forall j \in V$: F_j^x has full rank (= l_j^x).
(A1.3$_{\text{in}}$) $\forall j \in V$: $\begin{pmatrix} F_{i,S(i)}^c & D_{S(i)}^c \end{pmatrix} \mid \big(N(F_i^x) \cap N(D_{S(i)}^u)\big)$ has full rank (= $l_{S(i)}^c$).
(A1.4$_{\text{in}}$) $G|N(F^l)$ has full rank (= n^x).
(A1.5$_{\text{in}}$) $F|(N(F^l) \cap N(G))$ has full rank (= m).
(A2$_{\text{in}}$) $\forall y \in \mathcal{F}$: $\nabla^2\phi(y)|\mathcal{N}^* \ge \epsilon I > 0$ where

$$\mathcal{N}^* := N(P_{\mathcal{B}_l \cup \mathcal{B}_u}) \cap N(P_{\mathcal{R}_l \cup \mathcal{R}_u} B) \cap N(F^l) \cap N(G).$$

Lemma 2. *The following properties hold.*

(a) *Conditions* (A1.1$_{\text{in}}$)–(A1.3$_{\text{in}}$) *are equivalent to full rank of F^l.*
(b) *Condition* (A1.3$_{\text{in}}$) *implies full row rank of* $\begin{pmatrix} F_{ij}^c & D_j^c \end{pmatrix} | (N(F_i^x) \times N(D_j^u))$ $\forall j \in V$. *The reverse implication does not hold in general.*
(c) *Conditions* (A1.1$_{\text{in}}$)–(A1.5$_{\text{in}}$) *are equivalent to full rank of A.*
(d) *Condition* (A2$_{\text{in}}$) *implies* (A2).
(e) *Conditions* (A1.1$_{\text{in}}$)–(A2$_{\text{in}}$) *are equivalent with* (A0)–(A2) *if and only if global constraints are absent, that is, if $m = 0$.*

Proof. Statements (a), (c), (d), (e) are proved as in Lemma 1, where $F_{i,S(i)}^c, D_{S(i)}^c$ now play the roles of F_j^c, D_j^c. The implication stated in (b) is obvious. A simple counterexample shows that the reverse implication is generally false: let $D_j^u = I$, $F_{ij}^c = I$, and $D_j^c = 0$ for all $j \in S(i)$ (where $|S(i)| > 1$ and $l_j^u = l_j^c = n_i^x$). □

Remarks. Condition (A1.3$_{\text{in}}$) plays precisely the same role as (A1.3$_{\text{out}}$) but is more complicated due to the (x_i, u_j)-coupling. Again, backward effects will not occur if (30) holds and (A1.3$_{\text{in}}$) is replaced by (A1.3$'_{\text{in}}$): $\forall j \in V$: $D_j^c|N(D_j^u)$ has full rank (= l_j^c).

Examples. The incoming control form appears to be the most common one in stochastic programming. Only two problems in the collection of King [22] are *not* posed in this form. The widest application area is probably mathematical finance. The current work started with the dynamic mean-variance approach [16, 17]; a simplified version is discussed in

[30, 31], for a theoretical investigation see [33]. Another important field concerns logistics; a particularly difficult recent application is described by Dempster et al. [15].

2.3. **Implicit Dynamics.** The general implicit tree-sparse CP in variables $y_j \in \mathcal{Y}_j$ reads

$$\min_y \quad \sum_{j \in V} \phi_j(y_j) \tag{31}$$

$$\text{s.t.} \quad P_j y_j = G_j y_i + h_j \qquad \forall j \in V, \tag{32}$$

$$F_j^y y_j + e_j^y = 0 \qquad \forall j \in V, \tag{33}$$

$$F_j^r y_j \in [r_{lj}, r_{uj}] \qquad \forall j \in V, \tag{34}$$

$$y_j \in [b_{lj}, b_{uj}] \qquad \forall j \in V, \tag{35}$$

$$\sum_{j \in V} (F_j y_j + e_j) = 0. \tag{36}$$

The representation of A, B in (14) remains valid if we define $y := (y_0, \dots, y_N)$, $B_j := F_j^r$, $F^l \equiv F^y := \mathrm{Diag}(F_0^y, \dots, F_N^y)$, $F := \begin{pmatrix} F_0 & \dots & F_N \end{pmatrix}$, and (for the example tree)

$$G = \begin{bmatrix} -P_0 & & & & & \\ G_1 & -P_1 & & & & \\ G_2 & & -P_2 & & & \\ & G_3 & & -P_3 & & \\ & G_4 & & & -P_4 & \\ & & G_5 & & & -P_5 \end{bmatrix}. \tag{37}$$

A detailed investigation of this formulation is provided in [32]. For the sake of completeness we summarize the most important aspects here and give additional comments comparing implicit and explicit variants. The most obvious difference is that only one type of decision variables appears, which may be thought of as combined states and controls. Correspondingly, the dynamic equations are given in implicit form. Their dimension will typically be smaller than the number of variables to leave some degrees of freedom for optimization ("hidden controls").

Dimensions. The respective dimensions of variables, dynamics, and constraints are n_j, l_j^d, and l_j^y, l_j^r, m, with restrictions

$$l_j^d + l_j^y \le n_j^y, \qquad m \le n^y - (l^d + l^y) \equiv \sum_{j \in V} (n_j^y - l_j^d - l_j^y).$$

In this setting it does not make sense to consider implied constraints: their treatment would entail a reformulation of the problem in one of the explicit variants (as is discussed in §3).

Regularity assumptions. Here we keep only (A0) and refine assumptions (A1) and (A2):

(A1.1$_{\text{impl}}$) $\forall j \in V$: F_j^y has full rank (= l_j^y).

(A1.2$_{\text{impl}}$) $\forall j \in V$: $P_j|N(F_j^y)$ has full rank (= l_j^d).

(A1.3$_{\text{impl}}$) $F|(N(F^y) \cap N(G))$ has full rank (= m).

(A2$_{\text{impl}}$) $\forall y \in \mathcal{F}$: $\forall j \in V$: $\nabla^2 \phi_j(y_j)|\mathcal{N}_j \ge \epsilon_j I > 0$ where

$$\mathcal{N}_j := N(P_{\mathcal{B}_{lj} \cup \mathcal{B}_{uj}}) \cap N(P_{\mathcal{R}_{lj} \cup \mathcal{R}_{uj}} F_j^r) \cap N(F_j^y) \cap \bigcap_{k \in S(j)} N(G_k).$$

Lemma 3. *The following properties hold.*

(a) *Condition* (A1.1$_{\text{impl}}$) *is equivalent to full rank of* F^y.

(b) *Condition* (A1.2$_{\text{impl}}$) *implies full rank of* $G|N(F^y)$. *The converse is not true in general.*

(c) *Conditions* (A1.1$_{\text{impl}}$)–(A1.3$_{\text{impl}}$) *imply full rank of* A.

(d) *Conditions* (A1.1$_{\text{impl}}$)–(A2$_{\text{impl}}$) *imply* $\nabla^2 \phi(y)|\mathcal{N}^* \ge \epsilon I > 0$ *where*

$$\mathcal{N}^* := N(P_{\mathcal{B}_l \cup \mathcal{B}_u}) \cap N(P_{\mathcal{R}_l \cup \mathcal{R}_u} B) \cap N(F^y) \cap N(G),$$

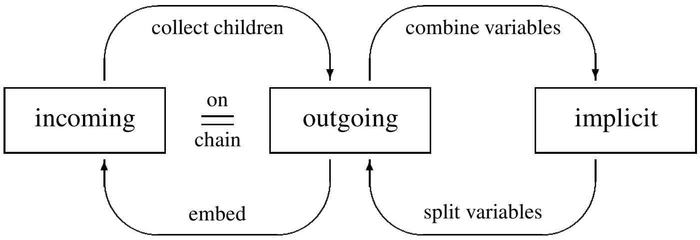

FIGURE 2. Relations between incoming, outgoing, and implicit control.

but $\mathcal{N}^* \subseteq \prod_{j\in V} \mathcal{N}_j$ *does not necessarily hold.*

Proof. Statement (a) and the implication stated in (b) are obvious. (The latter holds even if $G_j = 0\ \forall j \in V$.) To see that the reverse implication is generally false, let $V = \{0, 1\}$, $P_0 = P_1 = F_1^y = (1\ 0)$, and $G_1 = (0\ 1)$. Statement (c) follows immediately. The first part of statement (d) follows from Lemma 4 in [32]. Letting $V = \{0, 1, 2\}$, $P_0 = G_1 = (1\ 0)$, $P_1 = P_2 = (1)$, and $G_2 = (0\ 1)$ yields

$$\mathcal{N}^* = N(G) = \operatorname{span}(\{(0,1,0,1)\}), \qquad \prod_{j\in V} \mathcal{N}_j = \{(0,0)\} \times \mathbb{R} \times \mathbb{R}.$$

(For $j \in L$ we always have $\mathcal{N}_j = \mathcal{Y}_j$.) This completes the proof. $\square$

Remarks. Actually (A1.2$_{\text{impl}}$) imposes a specific structure of local constraints that rules out the possibility of backward effects (see Theorem 1). This is required by the solution algorithm, which does not distinguish states and controls and consequently exploits the tree-sparse structure to a lesser extent than the explicit variants. Assumption (A2$_{\text{impl}}$) is specified in local quantities, and it is stronger than in the two explicit cases by statemet (d).

Note also that local constraints (33) may always be modeled as dynamics (32), and that both are interchangeable in the root. The main difference of these conditions lies in the algorithmic treatment, so it is up to the modeler to choose the appropriate one.

Examples. The standard form for multistage stochastic linear programs (see, e.g., [8, 10, 21]) is actually implicit according to our classification, except that the various categories of constraints are not distinguished. A nice example is the long-term asset management model developed in [1]; we discuss it in [32] to illustrate the constraint structure. The early problem collection of King [22] includes two implicit examples on pages 548 and 560.

3. COMPARISON

Apparently there exist close relations between the three types of tree-sparse problems. It turns out that the problems are almost algebraically equivalent in the sense that they can be transformed into each other. In this section we study the precise relations; Fig. 2 gives a schematic overview. The technical comparison highlights similarities and differences in the details of the three formulations, in particular regarding the regularity requirements.

3.1. Implicit vs. Outgoing Control.

The simplest transformation is the implicit reformulation of outgoing control: one simply combines variables and matrix blocks appropriately. Conversely, a partitioning of decision variables into states and controls is possible, but this requires a partial solution of the system since otherwise the control components determined by local constraints would appear on the wrong side of dynamic equations.

Theorem 1. *In the notation of* §2.1*, let* $y_j := (x_j, u_j)$, $b_{lj} := (b^x_{lj}, b^u_{lj})$, $b_{uj} := (b^x_{uj}, b^u_{uj})$. *Then the CP with outgoing control* (4)–(12) *is equivalent to the implicit tree-sparse CP*

$$\begin{aligned} \min_y \quad & \sum_{j \in V} \phi_j(y_j) \\ \text{s.t.} \quad & \begin{pmatrix} I & 0 \end{pmatrix} y_j = \begin{pmatrix} G_j & E_j \end{pmatrix} y_i + h_j && \forall j \in V, \\ & F^l_j y_j + e^l_j = 0 && \forall j \in V, \\ & B_j y_j \in [r_{lj}, r_{uj}] && \forall j \in V, \\ & y_j \in [b_{lj}, b_{uj}] && \forall j \in V, \\ & \sum_{j \in V} \left[\begin{pmatrix} F_j & D_j \end{pmatrix} y_j + e_j \right] = 0. \end{aligned}$$

Assumptions (A1.1$_{\text{impl}}$), (A1.2$_{\text{impl}}$) *are satisfied if and only if no state constraints* (6) *occur* ($l^x_j = 0$) *and* (A1.2$_{\text{out}}$), (A1.3$'_{\text{out}}$) *hold;* (A1.4$_{\text{out}}$) *then follows, and* (A1.3$_{\text{impl}}$) *is equivalent to* (A1.5$_{\text{out}}$) *in any case. Condition* (A2$_{\text{impl}}$) *is not necessarily satisfied if* (A2$_{\text{out}}$) *holds.*

Proof. Equivalence of the problems and of condition (A1.3$_{\text{impl}}$) with (A1.5$_{\text{out}}$) are obvious. Condition (A1.1$_{\text{impl}}$) is equivalent to (A1.1$_{\text{out}}$)–(A1.3$_{\text{out}}$) by statement (a) of Lemma 1. Since $N(P_j) = N\begin{pmatrix} I & 0 \end{pmatrix} = \{0\} \times \mathcal{U}_j$, (A1.2$_{\text{impl}}$) holds if and only if for every $x_j \in \mathcal{X}_j$ there exists $u_j \in \mathcal{U}_j$ such that $F^l_j y_j = 0$, which in turn is equivalent to $N(F^x_j) = \{0\}$ and $R(F^c_j) \subseteq R(D^c_j | N(D^u_j))$. These two conditions are trivial consequences of $l^x_j = 0$ and (A1.3$'_{\text{out}}$), which also imply (A1.1$_{\text{out}}$) and (A1.3$_{\text{out}}$). This proves that the stated conditions imply (A1.1$_{\text{impl}}$), (A1.2$_{\text{impl}}$). Conversely, $l^x_j = 0$ and (A1.3$'_{\text{out}}$) follow from $N(F^x_j) = \{0\}$ and $R(F^c_j) \subseteq R(D^c_j | N(D^u_j))$ whenever (A1.1$_{\text{out}}$) and (A1.3$_{\text{out}}$) hold. Observe finally that (A1.2$_{\text{impl}}$) implies (A1.3$_{\text{out}}$), and that (A2$_{\text{impl}}$) may require $\nabla^2 \phi_j(y_j) > 0$ in the leaves, which is not guaranteed by (A2$_{\text{out}}$). □

Remark. Conditions $l^x_j = 0$ and (A1.3$'_{\text{out}}$) are precisely the restrictions that enable the simpler recursion without backward effects for the implicit CP. The initial projection steps of the recursive solution procedure for the CP with outgoing control eliminate all local constraints (6), (7), and (8). Thus, if (A1.1$_{\text{out}}$)–(A1.5$_{\text{out}}$) hold, then (A1.1$_{\text{impl}}$)–(A1.3$_{\text{impl}}$) will be satisfied after the partial solve.

Theorem 2. *Assume that* (A1.1$_{\text{impl}}$) *and* (A1.2$_{\text{impl}}$) *hold in the implicit CP of* §2.3*, and let*

$$\begin{aligned} \Pi_{j1} &:= \begin{pmatrix} I & 0 \end{pmatrix} \in \mathbb{R}^{l^y_j \times n^y_j}, & \Pi_{jx} &:= \begin{pmatrix} I & 0 \end{pmatrix} \in \mathbb{R}^{l_j \times (n^y_j - l^y_j)}, \\ \Pi_{j2} &:= \begin{pmatrix} 0 & I \end{pmatrix} \in \mathbb{R}^{(n^y_j - l^y_j) \times n^y_j}, & \Pi_{ju} &:= \begin{pmatrix} 0 & I \end{pmatrix} \in \mathbb{R}^{(n^y_j - l^y_j - l_j) \times (n^y_j - l^y_j)}. \end{aligned}$$

Then y_j can be partitioned into state and control variables x_j, u_j and a constant vector y_{j1}, and non-singular matrices L_j, U_j, U^y_j exist so that with abbreviations

$$\bar{y}_{j1} := (U^y_j)^{-1} \Pi^*_{j1} y_{j1}, \qquad \bar{U}_j := (U^y_j)^{-1} \Pi^*_{j2} U_j^{-1},$$

(31)–(36) *is equivalent to the CP with outgoing control defined by* $\bar{\phi}_j(x_j, u_j) := \phi_j(y_j)$ *and the following problem data:*

$$\begin{aligned} \begin{pmatrix} \bar{G}_j & \bar{E}_j \end{pmatrix} &:= L_j^{-1} G_j \bar{U}_i, & \bar{h}_j &:= L_j^{-1} [h_j + G_j \bar{y}_{i1} - P_j \bar{y}_{j1}], \\ \begin{pmatrix} \bar{F}^r_j & \bar{D}^r_j \end{pmatrix} &:= \begin{pmatrix} F^r_j \\ I \end{pmatrix} \bar{U}_j, & \bar{r}_{*j} &:= \begin{pmatrix} r_{*j} \\ b_{*j} \end{pmatrix} - \begin{pmatrix} F^r_j \\ I \end{pmatrix} \bar{y}_{j1}, \quad * = l, u, \\ \begin{pmatrix} \bar{F}_j & \bar{D}_j \end{pmatrix} &:= F_j \bar{U}_j, & \bar{e}_j &:= e_j + F_j \bar{y}_{j1}. \end{aligned}$$

Conditions (A1.1$_{\text{out}}$)–(A1.4$_{\text{out}}$) *hold without further requirements, and* (A1.5$_{\text{out}}$), (A2$_{\text{out}}$) *hold if* (A1.3$_{\text{impl}}$) *respectively* (A2$_{\text{impl}}$) *are satisfied.*

Proof. By full rank of F^y_j and of $P_j|N(F^y_j)$ we have factorizations $F^y_j = L^y_j \Pi_{j1} U^y_j$ and $P_{j2} = L_j \Pi_{jx} U_j$ where $(P_{j1}\ P_{j2})U^y_j = P_j$, that is, $P_{j\nu} = P_j(U^y_j)^{-1}\Pi^*_{j\nu}$. Now partition

$$U^y_j y_j =: \begin{pmatrix} y_{j1} \\ y_{j2} \end{pmatrix}, \qquad U_j y_{j2} =: \begin{pmatrix} x_j \\ u_j \end{pmatrix},$$

that is, $y_{j\nu} := \Pi_{j\nu} U^y_j y_j$, $\nu = 1, 2$, and $x_j := \Pi_{jx} U_j y_{j2}$, $u_j := \Pi_{ju} U_j y_{j2}$. (Here y_{j1} might be regarded as a control variable, but we must fix its actual value $y_{j1} = -(L^y_j)^{-1} e^y_j$ determined by (33), since otherwise the dynamics reformulation would remain implicit.) Proving the equivalence with the original problem is now straightforward; we exercise this only for the most involved part, the dynamics. By definition,

$$\bar{G}_j x_i + \bar{E}_j u_i = [L_j^{-1} G_j (U^y_i)^{-1} \Pi^*_{i2} U_i^{-1}](U_i y_{i2}) = L_j^{-1} G_j (U^y_i)^{-1} \Pi^*_{i2} y_{i2}.$$

Thus, since $\Pi^*_{i2} y_{i2} + \Pi^*_{i1} y_{i1} = U^y_i y_i$, $G_j y_i + h_j = P_j y_j$, and $P_j \bar{y}_{j1} = P_{j1} y_{j1}$,

$$\begin{aligned} \bar{G}_j x_i + \bar{E}_j u_i + \bar{h}_j &= L_j^{-1}[G_j (U^y_i)^{-1}(\Pi^*_{i2} y_{i2} + \Pi^*_{i1} y_{i1}) + h_j - P_j \bar{y}_{j1}] \\ &= L_j^{-1}[P_j y_j - P_{j1} y_{j1}] = L_j^{-1} P_{j2} y_{j2} = L_j^{-1} L_j \Pi_{jx} U_j y_{j2} = x_j. \end{aligned}$$

Observe that the reformulation is the restriction of the CP to $\bar{y}_{j1} + N(F^l)$, and $\bar{F}^l$ is empty. Thus (A1.1$_{\text{out}}$)–(A1.3$_{\text{out}}$) hold trivially, and (A1.4$_{\text{out}}$) follows from the stronger condition (A1.2$_{\text{impl}}$) by Lemma 3 (b) since $\bar{G}$ corresponds to $G|N(F^l)$ by construction. Likewise, (A1.3$_{\text{impl}}$) implies (A1.5$_{\text{out}}$) and (A2$_{\text{impl}}$) implies (A2$_{\text{out}}$) by Lemma 3 (c) and (d). □

3.2. **Incoming vs. Outgoing Control.** In the deterministic case (where the tree reduces to a chain), the two explicit variants are clearly identical except for the numbering of controls and obvious differences at the head and tail of the chain.

On a general tree the collection of incoming controls of all siblings can be defined as outgoing control of their *parent* node (after prepending a new node before the root). Conversely, each outgoing control can be reinterpreted as an incoming control of the *current* node if a copy is appended to its associated state (and thus passed on to the successors).

Theorem 3. *In the notation of* §2.2*, let* $\bar{u}_j := (u_{k_1}, \ldots, u_{k_s})$ *where* $S(j) = \{k_1, \ldots, k_s\}$*, and similarly* $\bar{e}^u_j := (e^u_{k_1}, \ldots, e^u_{k_s})$, $\bar{e}^c_j := (e^c_{k_1}, \ldots, e^c_{k_s})$*. Define*

$$\bar{\phi}_j(x_j, \bar{u}_j) := \phi_j(x_j) + \sum_{k \in S(j)} \phi_{jk}(x_j, u_k), \qquad \bar{E}_j := \big(0\ \ldots\ \underbrace{E_j}_{\text{block column } j}\ \ldots\ 0\big),$$

and

$$\bar{F}^c_j := \begin{pmatrix} F^c_{jk_1} \\ \vdots \\ F^c_{jk_s} \end{pmatrix}, \qquad \bar{F}^r_j := \begin{pmatrix} F^r_j \\ F^r_{jk_1} \\ \vdots \\ F^r_{jk_s} \end{pmatrix}, \qquad \bar{r}_{lj} := \begin{pmatrix} r^x_{lj} \\ r^u_{lk_1} \\ \vdots \\ r^u_{lk_s} \end{pmatrix}, \quad \bar{r}_{uj} := \begin{pmatrix} r^x_{uj} \\ r^u_{uk_1} \\ \vdots \\ r^u_{uk_s} \end{pmatrix}.$$

Finally let

$$\begin{aligned} \bar{D}^u_j &:= \operatorname{Diag}(D^u_{k_1}, \ldots, D^u_{k_s}), & \bar{D}^r_j &:= \begin{pmatrix} 0 \\ \operatorname{Diag}(D^r_{k_1}, \ldots, D^r_{k_s}) \end{pmatrix}, \\ \bar{D}^c_j &:= \operatorname{Diag}(D^c_{k_1}, \ldots, D^c_{k_s}), & \bar{D}_j &:= (D_{k_1}, \ldots, D_{k_s}). \end{aligned}$$

All remaining quantities—including the states—remain unchanged, $\bar{x}_j := x_j$, $\bar{G}_j := G_j$, $\bar{h}_j := h_j$, $\bar{F}^x_j := F^x_j$, $\bar{e}^x_j := e^x_j$ $\bar{F}_j := F_j$*, and* $\bar{e}_j := e_j$*. These data then define an equivalent CP with outgoing control. If assumptions* (A1.1$_{\text{in}}$)–(A2$_{\text{in}}$) *are satisfied, then* (A1.1$_{\text{out}}$)–(A2$_{\text{out}}$) *hold in the transformed CP.*

Proof. We have tacitly prepended a new node $j = -1$ before the root to carry the outgoing control $\bar{u}_{-1} \equiv u_0$, an empty state $x_{-1} \in \mathbb{R}^0$, and the associated local CP data. The leaves have empty outgoing controls now, $\bar{u}_j \in \mathbb{R}^0$, $j \in L$. Otherwise the original problem has just been rearranged and the verification of CP equivalence is straightforward. Each of the conditions (A1.1$_{\text{out}}$)–(A2$_{\text{out}}$) follows immediately from the corresponding condition in (A1.1$_{\text{in}}$)–(A2$_{\text{in}}$), where (A1.1$_{\text{in}}$) corresponds to (A1.2$_{\text{out}}$) and (A1.2$_{\text{in}}$) to (A1.1$_{\text{out}}$). □

Theorem 4 (Embedding). *In the notation of* §2.1*, let* $\bar{x}_j := (x_j, u_j)$ *and* $\bar{b}^x_{lj} := (b^x_{lj}, b^u_{lj})$*,* $\bar{b}^x_{uj} := (b^x_{uj}, b^u_{uj})$*. Then* (4)–(12) *is equivalent to the following CP with incoming control:*

$$
\begin{aligned}
\min_{u,\bar{x}} \quad & \sum_{j\in V} \phi_j(\bar{x}_j) \\
\textit{s.t.} \quad & \bar{x}_j = \begin{pmatrix} G_j & E_j \\ 0 & 0 \end{pmatrix} \bar{x}_i + \begin{pmatrix} 0 \\ I \end{pmatrix} u_j + \begin{pmatrix} h_j \\ 0 \end{pmatrix} && \forall j \in V, \\
& F^l_j \bar{x}_j + e^l_j = 0 && \forall j \in V, \\
& B_j \bar{x}_j \in [r_{lj}, r_{uj}] && \forall j \in V, \\
& \bar{x}_j \in [\bar{b}^x_{lj}, \bar{b}^x_{uj}] && \forall j \in V, \\
& \sum_{j\in V} \left[\begin{pmatrix} F_j & D_j \end{pmatrix} \bar{x}_j + e_j \right] = 0.
\end{aligned}
$$

If (A1.1$_{\text{out}}$)–(A2$_{\text{out}}$) *are satisfied, then* (A1.1$_{\text{in}}$)–(A2$_{\text{in}}$) *hold in the transformed CP.*

Proof. We have only modified the dynamic equations, where all duplicated control variables are determined by an identical number of conditions. Otherwise the CP has merely been rewritten, and equivalence is easily verified. Conditions (A1.1$_{\text{in}}$), (A1.3$_{\text{in}}$) hold trivially since the associated constraints are empty. (A1.1$_{\text{out}}$)–(A1.3$_{\text{out}}$) clearly imply (A1.2$_{\text{in}}$). If $G|N(F^l)$ has full rank, then this is also true for the original (upper) part of the transformed dynamics. The new (lower) part is unaffected by the restriction, so (A1.4$_{\text{in}}$) follows. A similar argument shows that (A1.5$_{\text{out}}$), (A2$_{\text{out}}$) imply (A1.5$_{\text{in}}$), (A2$_{\text{in}}$), since $\bar{x}_{j2} = u_j$ for all $j \in V$. □

3.3. **Implicit vs. Incoming Control.** The previous results are now combined to establish the relation between implicit and incoming control formulations.

Theorem 5. *Given a CP with incoming control, the composed transformations of Theorem 3 and Theorem 1 yield an equivalent CP in implicit formulation. Conditions* (A1.1$_{\text{impl}}$)*,* (A1.2$_{\text{impl}}$) *are satisfied if and only if no state constraints* (23) *occur (*$l^x_j = 0$*) and* (A1.1$_{\text{in}}$)*,* (A1.3$^l_{\text{in}}$) *hold;* (A1.4$_{\text{in}}$) *then follows, and* (A1.3$_{\text{impl}}$) *is equivalent to* (A1.5$_{\text{in}}$) *in any case. Condition* (A2$_{\text{impl}}$) *is not necessarily satisfied if* (A2$_{\text{in}}$) *holds.*

Proof. This follows directly from Theorem 3 and Theorem 1. □

Theorem 6. *Assume that* (A1.1$_{\text{impl}}$) *and* (A1.2$_{\text{impl}}$) *hold in* §2.3*. Then the combined transformations of Theorem 2 and Theorem 4 yield an equivalent CP with incoming control. Conditions* (A1.1$_{\text{in}}$)–(A1.4$_{\text{in}}$) *hold without further requirements, and* (A1.5$_{\text{in}}$)*,* (A2$_{\text{in}}$) *hold if* (A1.3$_{\text{impl}}$) *respectively* (A2$_{\text{impl}}$) *are satisfied.*

Proof. This follows directly from Theorem 2 and Theorem 4. □

3.4. **Discussion.** The previous results show that the explicit CP forms and associated regularity conditions are always equivalent, with a slightly finer structure in the incoming control form. Although straightforward (re)formulations in implicit form always exist, efficient solution requires stronger regularity conditions than in the explicit variants. If they are satisfied, the problem can always be recast in one of the explicit forms.

As indicated in [28], a potential substructure of global constraints can be exploited in the recursions. More precisely, each row can—and should—be eliminated in the root of the unique smallest subtree with nonzero entries in that particular row of F_j, D_j. Thus our framework offers the choice of modeling local constraints as such (handled by projections) or as global constraints (handled by Schur complement calculations). This is similar to viewing local constraints as dynamic equations in §2.3.

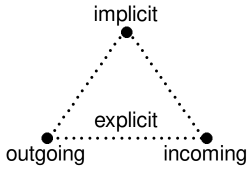

FIGURE 3. Range of possible dynamics formulations

One might also consider mixed forms of dynamics, where each node has an incoming and an outgoing control, or where some state components are determined by explicit equations and other components by implicit ones. This entire range of possibilities (which we do not wish to explore) is covered by suitable "convex" combinations of the three pure variants; see Fig. 3.

4. KKT SOLUTION

We now address the key task of solving the reduced KKT systems (2) in interior methods for tree-sparse programs. Range constraints in all variants have the property that $B^*\Psi B$ and $H = \nabla^2\phi$ have identical block structures. Thus we may consider the second form,

$$\begin{bmatrix} \bar{H} & A^* \\ A & \end{bmatrix} \begin{bmatrix} \Delta y \\ -\Delta z \end{bmatrix} = \begin{bmatrix} \bar{f} \\ \bar{a} \end{bmatrix}, \tag{38}$$

which is equivalent to an equality-constrained convex quadratic tree-sparse program,

$$\min_{\Delta y} \frac{1}{2}\Delta y^* \bar{H} \Delta y - \bar{f}^* \Delta y \qquad \text{s.t.} \qquad A\Delta y = \bar{a}. \tag{39}$$

This problem is addressed in the following. To simplify notation, we drop the overbars and write y for the step direction Δy. (Note also that f, a now have the opposite sign.)

4.1. **Outgoing Control.** A complete investigation of the *deterministic* problem has already been provided in [28]. Moreover, by our earlier results, the outgoing control case is rather similar to the (slightly more involved) incoming control case. Therefore we present only the latter variant here; the reader may easily derive the necessary modifications.

4.2. **Incoming Control.** Suppose that (A1.1_{in})–(A2_{in}) hold. By Lemma 2 (e) in this paper and Lemma 2 in [32], problems (38) and (39) have a unique solution even if $m = 0$. Therefore consider the CP (20)–(24) without global constraints, and with objective

$$\min_{x,u} \sum_{j\in V} \left[\frac{1}{2} \begin{pmatrix} x_i \\ u_j \end{pmatrix}^* \begin{pmatrix} 0 & J_j^* \\ J_j & K_j \end{pmatrix} \begin{pmatrix} x_i \\ u_j \end{pmatrix} + \frac{1}{2} x_j^* H_j x_j - \begin{pmatrix} d_j \\ f_j \end{pmatrix}^* \begin{pmatrix} u_j \\ x_j \end{pmatrix} \right].$$

This problem has a separable Lagrangian

$$L(u, x, \lambda, \mu^u, \mu^x, \mu^c) = \sum_{j\in V} L_j(u_j, x_i, x_j, \lambda_j, \mu_j^u, \mu_j^x, \mu_j^c)$$

with node contributions

$$\begin{aligned} L_j = {} & u_j^* J_j x_i + \frac{1}{2} u_j^* K_j u_j + \frac{1}{2} x_j^* H_j x_j - d_j^* u_j - f_j^* x_j \\ & - \lambda_j^* (G_j x_i + E_j u_j - x_j - h_j) - \mu_j^{u*} (D_j^u u_j - e_j^u) \\ & - \mu_j^{x*} (F_j^x x_j - e_j^x) - \mu_j^{c*} (F_{ij} x_i + D_j^c u_j - e_j^c). \end{aligned}$$

The KKT system (38) decomposes into local conditions in every node $j \in V$,

$$
\begin{aligned}
(L_{u_j}):&\quad J_j x_i + K_j u_j + E_j^*(-\lambda_j) + D_j^{u*}(-\mu_j^u) + D_j^{c*}(-\mu_j^c) = d_j,\\
(L_{x_j}):&\quad H_j x_j - (-\lambda_j) + F_j^{x*}(-\mu_j^x) + \sum_{k\in S(j)} [J_k^* u_k + G_k^*(-\lambda_k) + F_{jk}^{c*}(-\mu_k^c)] = f_j,\\
(L_{\lambda_j}):&\quad G_j x_i + E_j u_j - x_j = h_j,\\
(L_{\mu_j^u}):&\quad D_j^u u_j = e_j^u,\\
(L_{\mu_j^x}):&\quad F_j^x x_j = e_j^x,\\
(L_{\mu_j^c}):&\quad F_{ij}^c x_i + D_j^c u_j = e_j^c.
\end{aligned}
$$

The recursive solution algorithm starts with a subset of leaves $S(i) \subseteq L$ having the same predecessor i. The following operations are performed for each $j \in S(i)$.

(1) The initial step is a projection onto the null space of local control constraints (22). By (A1.1$_{\text{in}}$) the full-rank matrix D_j^u admits a rectangular LU factorization yielding

$$D_j^u u_j = \begin{pmatrix} L_j & 0 \end{pmatrix} U_j u_j = \begin{pmatrix} L_j & 0 \end{pmatrix} \begin{pmatrix} u_j^1 \\ u_j^2 \end{pmatrix} = L_j u_j^1 = e_j^u.$$

Thus we have $u_j^1 = L_j^{-1} e_j^u$. Substitution into (L_{u_j}) with appropriate partitioning gives

$$
\begin{aligned}
(L_{u_j^1}):&\quad J_j^1 x_i + K_j^{12} u_j^2 + E_j^{1*}(-\lambda_j) + L_j^*(-\mu_j^u) + D_j^{c1*}(-\mu_j^c) = (d_j^1 - K_j^{11} u_j^1),\\
(L_{u_j^2}):&\quad J_j^2 x_i + K_j^{22} u_j^2 + E_j^{2*}(-\lambda_j) \qquad\qquad + D_j^{c2*}(-\mu_j^c) = (d_j^2 - K_j^{21} u_j^1).
\end{aligned}
$$

The upper part determines the multiplier μ_j^u associated with (22); the lower part has the same form as (L_{u_j}) but without the D_j^u term. Analogous partitionings and substitutions are performed with all remaining conditions that involve u_j, where contributions from u_j^1 are absorbed into the right hand sides. (Note that in (L_{x_i}), each sibling $j \in S(i)$ adds a term $-J_j^{1*} u_j^1$ to f_i.) The resulting set of conditions has the same form as if no control constraints had ever been present in $S(i)$. We drop component indices of the partitioned data and proceed with this simplified case in the original notation.

(2) The second step is an analogous projection onto the null space of local state constraints (23). By (A1.2$_{\text{in}}$) the full-rank matrix F_j^x admits a rectangular LU factorization yielding (with different L_j, U_j, of course)

$$F_j^x x_j = \begin{pmatrix} L_j & 0 \end{pmatrix} U_j x_j = \begin{pmatrix} L_j & 0 \end{pmatrix} \begin{pmatrix} x_j^1 \\ x_j^2 \end{pmatrix} = L_j x_j^1 = e_j^x.$$

We obtain $x_j^1 = L_j^{-1} e_j^x$ and, since the sum over $k \in S(j) = \emptyset$ in (L_{x_j}) vanishes,

$$
\begin{aligned}
(L_{x_j^1}):&\quad H_j^{12} x_j^2 - (-\lambda_j^1) + L_j^*(-\mu_j^x) = (f_j^1 - H_j^{11} x_j^1),\\
(L_{x_j^2}):&\quad H_j^{22} x_j^2 - (-\lambda_j^2) \qquad\qquad = (f_j^2 - H_j^{21} x_j^1).
\end{aligned}
$$

Conditions (L_{u_j}) and $(L_{\mu_j^c})$ remain unaffected, and the partitioning of (L_{λ_j}) yields

$$
\begin{aligned}
(L_{\lambda_j^1}):&\quad G_j^1 x_i + E_j^1 u_j \qquad = (h_j^1 + x_j^1),\\
(L_{\lambda_j^2}):&\quad G_j^2 x_i + E_j^2 u_j - x_j^2 = h_j^2.
\end{aligned}
$$

The lower part is of course the dynamic equation for the remaining component x_j^2, but the upper part has been converted to an additional (*implied*) mixed constraint of type (24). Therefore we combine the relevant matrices and vectors,

$$C_j := \begin{pmatrix} G_j^1 \\ F_{ij}^c \end{pmatrix}, \quad A_j := \begin{pmatrix} E_j^1 \\ D_j^c \end{pmatrix}, \quad c_j := \begin{pmatrix} h_j^1 + x_j^1 \\ e_j^c \end{pmatrix}, \qquad \nu_j := \begin{pmatrix} \lambda_j^1 \\ \mu_j^c \end{pmatrix}.$$

Using otherwise the original notation, this yields the same set of conditions as if neither (22) nor (23) had ever been present in $S(i)$. But now we also have to monitor (L_{x_i}),

$$\begin{aligned}
&(L_{x_i}): && H_i x_i - (-\lambda_i) + F_i^{x*}(-\mu_i^x) + \sum_{j\in S(i)} [J_j^* u_j + G_j^*(-\lambda_j) + C_j^*(-\nu_j)] = f_i,\\
&(L_{u_j}): && J_j x_i + K_j u_j + E_j^*(-\lambda_j) + A_j^*(-\nu_j) = d_j\\
&(L_{x_j}): && H_j x_j - (-\lambda_j) = f_j\\
&(L_{\lambda_j}): && G_j x_i + E_j u_j - x_j = h_j\\
&(L_{\mu_j^c}): && C_j x_i + A_j u_j = c_j.
\end{aligned}$$

(3) The next step is a more complicated projection. We wish to determine as many control components as possible from condition $(L_{\mu_j^c})$ to avoid unnecessary backward effects. But A_j may be rank-deficient, so we perform a (partial) factorization with row pivoting and rank decision, and partition C_j, c_j accordingly,

$$P_j A_j = \begin{pmatrix} L_j & \\ A_j^x & Z_j \end{pmatrix} U_j, \qquad P_j C_j = \begin{pmatrix} C_j^u \\ C_j^x \end{pmatrix}, \quad P_j c_j = \begin{pmatrix} c_j^u \\ c_j^x \end{pmatrix}. \tag{40}$$

Here the partitioning is chosen such that L_j is well-conditioned and Z_j can be neglected. (In exact arithmetic we have $\operatorname{rank}(L_j) = \operatorname{rank}(A_j)$ and $Z_j = 0$.) Dropping Z_j gives

$$\begin{aligned}
&(L_{\mu_j^{cu}}): && u_j^1 = L_j^{-1}(c_j^u - C_j^u x_i),\\
&(L_{\mu_j^{cx}}): && (C_j^x - A_j^x L_j^{-1} C_j^u) x_i = (c_j^x - A_j^x L_j^{-1} c_j^u).
\end{aligned}$$

Thus, the upper part yields a *local feedback law* for u_j^1 whereas the lower part yields a local state constraint (23) in the *preceding* node $i = \pi(j)$. (Such a backward effect may potentially occur in *every* $j \in S(i)$.) The corresponding partitioning of (L_{u_j}) gives

$$\begin{aligned}
&(L_{u_j^1}): && -\nu_j^u = L_j^{-*}[d_j^1 - J_j^1 x_i - K_j^{11} u_j^1 - K_j^{12} u_j^2 - E_j^{1*}(-\lambda_j) - A_j^{x*}(-\nu_j^x)],\\
&(L_{u_j^2}): && (J_j^2 - K_j^{21} L_j^{-1} C_j^u) x_i + K_j^{22} u_j^2 + E_j^{2*}(-\lambda_j) = (d_j^2 - K_j^{21} L_j^{-1} c_j^u).
\end{aligned}$$

Here the upper part is a local feedback law for $-\nu_j^u$ (which is evaluated after u_j^1, on which it depends). The lower part has the original form but without any local constraints; similarly for (L_{x_j}) which is unaffected by the transformation, and (L_{λ_j}) which now reads

$$(L_{\lambda_j}): \qquad (G_j - E_j^1 L_j^{-1} C_j^u) x_i + E_j^2 u_j^2 - x_j = (h_j - E_j^1 L_j^{-1} c_j^u).$$

Condition (L_{x_i}) undergoes the most complicated transformation since we have to substitute expressions for both u_j^1 and $-\nu_j^u$, with u_j^1 substituted in turn,

$$\begin{aligned}
(L_{x_i}): \quad & \big[H_i + \textstyle\sum_{j\in S(i)} (C_j^{u*} L_j^{-*} K_j^{11} L_j^{-1} C_j^u - J_j^{1*} L_j^{-1} C_j^u - C_j^{u*} L_j^{-*} J_j^1)\big] x_i\\
&+ \textstyle\sum_{j\in S(i)} (J_j^2 - K_j^{21} L_j^{-1} C_j^u)^* u_j^2 - (-\lambda_i)\\
&+ \textstyle\sum_{j\in S(i)} (G_j - E_j^1 L_j^{-1} C_j^u)^* (-\lambda_j) + F_i^{x*}(-\mu_i^x)\\
&+ \textstyle\sum_{j\in S(i)} (C_j^x - A_j^x L_j^{-1} C_j^u)^* (-\nu_j^x) =\\
& \big[f_i + \textstyle\sum_{j\in S(i)} (C_j^{u*} L_j^{-*} \bar d_j^1 - J_j^{1*} L_j^{-1} c_j^u)\big], \qquad \bar d_j^1 := d_j^1 - K_j^{11} L_j^{-1} c_j^u.
\end{aligned}$$

This completes the projection part of the algorithm in nodes $j \in S(i)$. With appropriate (re)definitions, the remaining optimality conditions now read

$$\begin{aligned}
&(L_{u_j}): && J_j x_i + K_j u_j + E_j^*(-\lambda_j) = d_j,\\
&(L_{x_j}): && H_j x_j - (-\lambda_j) = f_j,\\
&(L_{\lambda_j}): && G_j x_i + E_j u_j - x_j = h_j.
\end{aligned}$$

Relevant conditions in the preceding node (with $\bar{C}_j^x := C_j^x - A_j^x L_j^{-1} C_j^u$) are

$$
\begin{aligned}
(L_{x_i})\colon \quad & H_i x_i - (-\lambda_i) + F_i^{x*}(-\mu_j^x) + \sum_{j\in S(i)} [J_j^* u_j + G_j^*(-\lambda_j) + \bar{C}_j^{x*}(-\nu_j^x)] = f_i, \\
(L_{\mu_i^x})\colon \quad & F_i^x x_i = e_i^x, \\
& \forall j \in S(i): \quad \bar{C}_j^x x_i = \bar{c}_j^x.
\end{aligned}
$$

At this point we augment F_i^x with all $\bar{C}_j^x$, e_i^x with all $\bar{c}_j^x$, and μ_j^x with all ν_j^x, obtaining $\bar{F}_i^x$, $\bar{e}_i^x$, and $\bar{\mu}_i^x$. Thusly combining original state constraints with implied ones restores the original form of (L_{x_i}) and $(L_{\mu_i^x})$.

Lemma 4. *In exact arithmetic, the augmented matrix $\bar{F}_i^x$ has full row rank.*

Proof. Assume that $\bar{F}_i^x$ does not have full row rank. Then, by (A1.2$_{\text{in}}$), there exists $\hat{x}_i \in N(F_i^x) \cap N(\bar{C}_j^x)$ for some $j \in S(i)$. Letting $\hat{u}_j^1 := -L_j^{-1} C_j^u \hat{x}_i$, the factorization (40) with $Z_j = 0$ (exact arithmetic) shows that $C_j \hat{x}_i + A_j \hat{u}_j = 0$ for any choice of $\hat{u}_j^2$. But then $(\hat{x}_i, \hat{u}_j)$ lies in the null space of at least one row of $(G_j^1 \ E_j^1)$ or $(F_{ij}^c \ D_j^c)$. Observing that $\hat{u}_j \in N(D_j^u)$ by construction, this contradicts either (A1.3$_{\text{in}}$) or (A1.4$_{\text{in}}$). □

Remark. In case of ill-conditioning (that is, $\operatorname{rank}(L_j) < \operatorname{rank}(A_j)$ for some j), F_i^x may of course still have full row rank; it is just not guaranteed.

(4) After the local projections **(1)–(3)**, all local constraints are eliminated and the solution algorithm proceeds with the basic recursion described in [31]. This basic part includes the handling of (projected) global constraints and is not repeated here. Its node operations can be performed immediately after the projection in the same node, or in an independent traversal of the tree after completing the projections in all nodes. As explained in [31, 32], the entire algorithm defines a direct sparse factorization of the KKT matrix together with the associated forward and backward substitutions.

4.3. **Implicit Dynamics.** A complete description of the general implicit formulation and associated solution algorithm has already been given in [32]. As in the explicit variants, local constraints are first eliminated by local projections, and an independent basic recursion solves the projected dynamic equations and global constraints.

5. Other Interior Approaches

5.1. **Two-Stage Linear Stochastic Programs.** The classical linear two-stage model with recourse originates in the well-known work of Beale [3] and Dantzig [14]. In our notation, the standard formulation with N scenarios yields the *block-angular* linear program

$$\min_y \quad f_0^* y_0 + \sum_{j=1}^{N} f_j^* y_j \tag{41}$$

$$\text{s.t.} \quad P_0 y_0 = h_0, \quad y_0 \geq 0, \tag{42}$$

$$P_j y_j = h_j + G_j y_0, \quad y_j \geq 0, \qquad j = 1, \ldots, N. \tag{43}$$

This obviously fits the implicit dynamics form as the special case where

(a) the problem has two stages, $V = \{0\} \cup S(0) \equiv \{0\} \cup L$;
(b) the objective is linear;
(c) all constraints are formulated as dynamics or nonnegativity constraints.

Under the full-rank condition (A1.2$_{\text{impl}}$) on P_j, assumptions (A1.1$_{\text{impl}}$)–(A2$_{\text{impl}}$) will hold. Several interior methods have been developed for this problem class [2, 6, 9, 11, 13, 20, 23]; most of them turn out to be encompassed within our framework.

TABLE 1. Corresponding matrix blocks in Birge and Holmes [9, §3.4] and Steinbach [32, §4.2] (index 2 indicates blocks after projection).

	given					generated		
[9]	A_0	W_l	T_l	$D_0^{-2} + A_0^* A_0$	D_l^{-2}	S_l	G_1	G_2
[32]	$-P_{02}$	$-P_{l2}$	G_{l2}	$H_{022} = \Phi_{022} + P_{02}^* P_{02}$	$H_{l22} = \Phi_{l22}$	$\hat{Y}_l$	$\tilde{H}_0$	$\hat{Y}_0$

Theorem 7. *The modification* [9] *of the block-angular factorization of Birge and Qi* [11] *for* (41)–(43) *is equivalent to the tree-sparse factorization applied to the (equivalent) problem with an additional quadratic penalty term in the root,*

$$\min_y \frac{1}{2}\|P_0 y_0 - h_0\|_2^2 + f_0^* y_0 + \sum_{j \in S(0)} f_j^* y_j .$$

Proof. A comparison of Lemma 4 and Table 1 in Steinbach [32] with Theorem 1 and procedure `finddy` in Birge and Holmes [9] (or with the proof of Theorem 3.2 in [11]) reveals a one-to-one correspondence of the matrices in Table 1. The order of block calculations is $S_l, S_l^{-1}, l = 1, \ldots l$ and then $G_1, G_1^{-1}, G_2, G_2^{-1}$ in [9], and $\hat{Y}_l, \hat{Y}_l^{-1}, \tilde{H}_0, \tilde{H}_0^{-1}, \hat{Y}_0, \hat{Y}_0^{-1}$ in [32]. The stabilizing term $A_0^* A_0$ in [9] corresponds precisely to the penalty term. □

Remark. Although Birge, Qi, and Holmes think in terms of the dense normal equations (3), their factorization actually recovers the full block-sparse structure of the augmented system (and adds the implicit stabilization). Only their order of evaluating solution components differs slightly from our symmetric algorithm.

Other approaches can now be characterized as follows (see Birge [7, §3.2–3.6] for a detailed earlier overview).

Lustig, Mulvey, and Carpenter [23] use a standard interior point code (OB1) working with a Cholesky factorization of the normal equations. To reduce the density of (3), they reformulate the model in “split-variable” form (with replicated root variables and explicit nonanticipativity constraints). They also study a partial splitting where certain variables are *not* replicated (our *controls*), thus reducing the size of (3) without destroying sparsity.

Jessup, Yang, and Zenios [20] investigate a parallel implementation of the Birge and Qi factorization. Yang and Zenios [38] pursue the same direction and introduce the obvious extension to quadratic objectives in the context of “robust optimization”.

Bahn et al. [2] apply an analytic center cutting plane method to a modified standard formulation (with dynamics $P_0 y_0 \le h_0$, $G_j y_0 - P_j y_j \le h_j$), but do not specialize the Cholesky factorization in calculating the step direction of the dual normal form $A^* \Phi A$.

Czyzyk, Fourer and Mehrotra [13] compare the splitting approach of [23] with their own augmented system approach in a computational study. They employ the same pivoting order as Birge and Qi (calling it “natural”) but drop the root system stabilization, thus arriving at the implicit dynamics recursion for the unmodified problem.

Berkelaar et al. [6] use a homogeneous self-dual path-following algorithm. The special factorization developed for the KKT system is again equivalent to ours. An interesting aspect in this context is that primal and dual objectives are combined to a *global constraint*.

5.2. **Multi-Stage Convex Stochastic Programs.** Berger et al. [5] treat convex multistage programs in non-Markovian standard form using a full split-variable formulation. They use Vanderbei's code LOQO with a special pivoting strategy called *tree dissection*. Further details are discussed in [32, §4.2].

Schweitzer [27] develops a recursive block factorization for $A\Phi^{-1}A^*$ in the standard LP form. This approach has linear complexity but requires stronger regularity conditions than ours and is less efficient; see “global Schur complement approach” in [32, §4.2].

In [12], Blomvall and Lindberg have recently developed a primal interior method using the outgoing control formulation. They work under stronger regularity assumptions (modifying D_j^c if necessary) to treat mixed constraints (8) by Lagrangian relaxation. This avoids local projections, obviates a distinction of state and control constraints (6), (7), and leads to Schur complement calculations for both (8) and (12) on top of the basic recursion.

Gondzio and Kouwenberg [19] pursue a completely different approach for multistage stochastic LP (cast in two-stage form by aggregating stages): they employ interior methods as master problem and subproblem solvers in a Benders decomposition framework [4, 36].

5.3. **Discussion.** For *two-stage* problems, all authors develop either a suitable form of the normal equations or a special factorization. In the latter group, all algorithms are essentially equivalent to the Birge and Qi approach, thus exploiting the block structure of the standard form (41)–(43). Our implicit two-stage formulation, although it is *not* more general, has the advantage of a finer, natural constraint structure, which would be replaced by an *artificial substructure* of P_j, G_j if converted to standard form. The preferred explicit variants offer even higher potential for exploiting natural sparsity.

The few alternative *multistage* approaches are either high-level adaptations of standard methodology ([5, 19]) or similar to our approach ([12, 27]). Again, our framework offers a finer structure allowing more flexible pivoting strategies with local projections.

6. GENERALIZED LINEAR-QUADRATIC CONTROL

In [26], Rockafellar and Wets study a class of deterministic and stochastic discrete-time control problems based on the concept of generalized linear-quadratic programming as introduced by Rockafellar [24]. This concept circles around a quadratic convex-concave function defined on a polyhedral set $\mathcal{U} \times \mathcal{V} \subseteq \mathbb{R}^k \times \mathbb{R}^l$,

$$J(u,v) = \frac{1}{2}\begin{pmatrix} u \\ v \end{pmatrix}^* \begin{pmatrix} P & -D^* \\ -D & -Q \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} p \\ q \end{pmatrix}^* \begin{pmatrix} u \\ v \end{pmatrix},$$

where P and Q are symmetric and positive semidefinite, yielding the primal problem

$$(\mathcal{P}) \qquad \min_{u \in \mathcal{U}} f(u), \qquad f(u) := \sup_{v \in \mathcal{V}} J(u,v),$$

and the completely symmetric dual problem

$$(\mathcal{D}) \qquad \max_{v \in \mathcal{V}} g(v), \qquad g(v) := \inf_{u \in \mathcal{U}} J(u,v).$$

Here it is understood that feasible solutions must satisfy $f(u) < \infty$ and $g(v) > -\infty$, respectively; thus the $\sup$ and $\inf$ formulations may hide implicit constraints. The supremum in $(\mathcal{P})$ turns out to be finite if $q - Du \in \mathcal{L} := [N(Q) \cap \operatorname{rec}(\mathcal{V})]^\circ$, where $\mathcal{L}$ is the polar cone of the intersection of the null space of Q with the recession cone of $\mathcal{V}$ [24, Prop. 2.4]. The primal problem can thus be written

$$\min_{u \in \mathcal{U}, q - Du \in \mathcal{L}} \left\{ \frac{1}{2} u^* P u + p^* u + \max_{v \in \mathcal{V}} \left\{ (q - Du)^* v - \frac{1}{2} v^* Q v \right\} \right\}.$$

We are interested in the smooth quadratic case obtained with $Q := 0$ and $\mathcal{V} := \mathbb{R}^{l^1} \times \mathbb{R}_+^{l^2}$ [24, Example 3.2]. In this case we get $\mathcal{L} = \mathcal{V}^\circ = \{0\} \times \mathbb{R}_-^{l^2}$, and the inner maximum over the dual variable v is always zero, yielding

$$\min_{u \in \mathcal{U}} \frac{1}{2} u^* P u + p^* u \qquad \text{s.t.} \qquad D^1 u = q^1, \quad D^2 u \geq q^2.$$

The associated deterministic discrete-time control problem $(\mathcal{P}_{\text{det}})$ [26, §3] reads

$$\min_{u\in\mathcal{U}} \quad \sum_{t=0}^{T}\Big(\frac{1}{2}u_t^* P_t u_t + p_t^* u_t - c_{t+1}^* x_t\Big) \tag{44}$$

$$\text{s.t.} \quad x_t = A_t x_{t-1} + B_t u_t + b_t \qquad \forall t = 0, \dots, T, \tag{45}$$

$$C_t^1 x_{t-1} + D_t^1 u_t = q_t^1 \qquad \forall t = 1, \dots, T+1, \tag{46}$$

$$C_t^2 x_{t-1} + D_t^2 u_t \geq q_t^2 \qquad \forall t = 1, \dots, T+1. \tag{47}$$

Here primal quantities are defined for $t = 0, \dots, T$ (in particular, $\mathcal{U} = \mathcal{U}_0 \times \cdots \times \mathcal{U}_T$) and dual quantities are defined for $t = 1, \dots, T+1$. Hence the matrices A_0 and D_{T+1}^ν are empty, but not $C_{T+1}^\nu, q_{T+1}^\nu, c_{T+1}$. Given a scenario tree of depth $T+1$, we just replace random elements X_t by their realizations X_j, $j \in L_t$, to obtain the corresponding stochastic problem $(\mathcal{P}_{\text{sto}})$ [26, §4][2]. Introducing empty vectors $c_0 \in \mathbb{R}^0$ and $u_j, x_j \in \mathbb{R}^0$, $j \in L_{T+1}$ (with associated empty matrices) we finally arrive at the formulation

$$\min_{u,x} \quad \sum_{j\in V}\Big(\frac{1}{2}u_j^* P_j u_j + p_j^* u_j - c_j^* x_i\Big) \tag{48}$$

$$\text{s.t.} \quad x_j = A_j x_i + B_j u_j + b_j \qquad \forall j \in V, \tag{49}$$

$$C_j^1 x_i + D_j^1 u_j = q_j^1 \qquad \forall j \in V, \tag{50}$$

$$C_j^2 x_i + D_j^2 u_j \geq q_j^2 \qquad \forall j \in V, \tag{51}$$

$$u_j \in \mathcal{U}_j \qquad \forall j \in V. \tag{52}$$

Theorem 8. *The linear-quadratic problem* (48)–(52) *can be reframed as a tree-sparse linear-quadratic program with incoming control.*

Proof. The objective and dynamics obviously fit into the tree-sparse framework, and constraints (50), (51) have the respective forms (24), (25). We conclude the proof by observing that each polyhedron $\mathcal{U}_j$ can be represented by constraints (22), (25) (with $F_{ij}^r = 0$), and (27). □

Theorem 9. *The tree-sparse convex program* (20)–(28) *with linear or quadratic objective (and without global constraints) can be reformulated as* (48)–(52).

Proof. If $H_j \neq 0$ or $J_j \neq 0$ in some node j, replace u_j by $\bar{u}_j := (x_i, u_j)$ and use the x_i part of $\bar{u}_j$ in quadratic objective terms involving x_i. Accordingly, augment (24) by the condition $Ix_i + (0\ {-I})\bar{u}_j = 0$ and extend other matrices that are multiplied by $\bar{u}_j$. Now define $\mathcal{U}_j := \{u_j \in [b_{lj}^u, b_{uj}^u] \colon D_j^u u_j + e_j^u = 0\}$ and

$$\begin{pmatrix} C_j^1 & D_j^1 & q_j^1 \end{pmatrix} := \begin{pmatrix} F_i^x & 0 & -e_i^x \\ F_{ij}^c & D_j^c & -e_j^c \end{pmatrix}, \qquad \begin{pmatrix} C_j^2 & D_j^2 & q_j^2 \end{pmatrix} := \begin{pmatrix} I & 0 & b_{li}^x \\ -I & 0 & -b_{ui}^x \\ F_i^r & 0 & r_{li}^x \\ -F_i^r & 0 & -r_{ui}^x \\ F_{ij}^r & D_j^r & r_{li}^u \\ -F_{ij}^r & D_j^r & -r_{ui}^u \end{pmatrix}.$$

Here it suffices to specify the first block row of $\begin{pmatrix} C_j^1 & D_j^1 & q_j^1 \end{pmatrix}$ in just one node $j \in S(i)$, and similarly for the first four rows of $\begin{pmatrix} C_j^2 & D_j^2 & q_j^2 \end{pmatrix}$. In the root, these five rows are all empty and the remaining three rows are instead specified as further restrictions of $\mathcal{U}_0$. (Recall that the q_j constraints *must* be empty in the root.) For $j \in L$, all restrictions on x_j are specified as q_j constraints in stage $T+1$ (which contains nothing else, as required). □

[2]Here we make the common assumption of complete information for simplicity, that is, $\mathcal{G}_t = \mathcal{F}_t$ in the notation of [26].

6.1. **Global Constraints.** If global constraints are present in the incoming control problem, its reformulation as a generalized linear-quadratic control problem requires additional state variables to pass partial sums $\sum \gamma_j := \sum (D_j u_j + F_j x_j + e_j)$ up or down the tree.

6.1.1. *Chains.* In the special case of a chain, $t = 0, \dots, T$, we define additional states $x'_t \in \mathbb{R}^m$ representing $\sum_{\tau<t} \gamma_\tau + D_t u_t + e_t$. The dynamic equations (49) are now

$$\begin{aligned} x_t &= G_t x_{t-1} + 0\, x'_{t-1} + E_t u_t + h_t \qquad && \forall t = 0, \dots, T, \\ x'_t &= F_{t-1} x_{t-1} + x'_{t-1} + D_t u_t + e_t \qquad && \forall t = 0, \dots, T, \end{aligned}$$

and the global constraint turns into a single terminal state constraint (50),

$$F_T x_T + x'_T = 0.$$

6.1.2. *Trees.* On a tree one must effectively accumulate partial sums backward in time. In every node $j \in V$ we add states $x'_j \in \mathbb{R}^m$ representing $F_j x_j + \sum_{k \in V(j)^*} \gamma_k$, and for each child $k \in S(j)$ an additional control-state pair $(u'_{jk}, x'_{jk}) \in \mathbb{R}^{2m}$ representing $\sum_{l \in V(k)} \gamma_l$. The dynamic equations (49) now read

$$\begin{aligned} x_j &= G_j x_i + E_j u_j + h_j \qquad && \forall j \in V, \\ x'_j &= x'_{ij} - D_j u_j - e_j \qquad && \forall j \in V, \\ x'_{jk} &= u'_{jk} \qquad && \forall j \in V,\ \forall k \in S(j), \end{aligned}$$

with the *root equation* $x'_0 = -D_0 u_0 - e_0$ replacing the original global constraint, and in each node we have a local state constraint (50)

$$x'_j - F_j x_j - \sum_{k \in S(j)} x'_{jk} = 0.$$

Although equivalent with our problem, this formulation is clearly much more complicated and requires lots of extra variables.

6.2. **Discussion.** We have seen that generalized linear-quadratic control problems share a central aspect with tree-sparse incoming control problems: the formulation of dynamic equations. The important subclass of smooth quadratic problems is equivalent to our class.

The generalized control approach has its origin in a deterministic continuous-time setting [24]; accent is placed on *convexity* and a completely symmetric *duality* framework, yielding deep theoretical insight. The full problem class is considerably more general than ours, including non-smooth saddlepoint problems with the possibility of constraints on dual variables. (In [25] Rockafellar also studies a scenario tree formulation with outgoing control but a similar duality framework.)

Our tree-sparse problems also originate in (discretized) nonlinear optimal control [28, 29, 34]. Here accent is placed on numerical exploitation of *differentiability* and *sparsity*, whereas convexity is not required (and rarely present in applications). The approach is probably more attractive from a practitioner's viewpoint since it allows the direct and efficient handling of arbitrary (primal) constraints, including coupled multipoint boundary conditions in the deterministic case and conditions on expectations in the stochastic case.

7. Conclusions

We have proposed and analyzed a flexible modeling and solution framework for dynamic stochastic programs and similarly structured optimization problems, thus developing a thorough theoretical understanding of the interaction between the inherent recourse structure and the hierarchy of constraints into which it is embedded. We have also developed natural KKT solution algorithms reflecting this hierarchical structure, so that all operations have direct control-theoretic interpretations in terms of the original formulation.

Although the presentation was restricted to the convex case for simplicity of exposition, the algorithmic approach extends directly to non-convex problems through appropriate interior methods or SQP methods using convex QP subproblems.

The proposed algorithmic concept is particularly efficient when the tree complexity dominates, that is, on large trees with dense blocks or blocks of moderate size. Moreover, by specializing the node operations it can be adapted to exploit a problem-specific substructure; an illustrative case study has been given in [32]. Considering general application problems, the real challenge consists in finding a practical way to handle large and sparse blocks efficiently. This is not entirely hopeless since the natural block elimination scheme confines fill-in to the given blocks and provides strong guidelines for the sparsity analysis. First investigations toward this direction are currently being conducted.

REFERENCES

[1] K. AINASSAARI, M. KALLIO, AND A. RANNE, *An asset management model for a pension insurance company*, technical report, Helsinki School of Economics, 1997.

[2] O. BAHN, O. DU MERLE, J.-L. GOFFIN, AND J. P. VIAL, *A cutting plane method from analytic centers for stochastic programming*, Math. Programming, 69 (1995), pp. 45–73.

[3] E. M. L. BEALE, *On minimizing a convex function subject to linear inequalities*, J. Royal Statist. Soc., Ser. B 17 (1955), pp. 173–184.

[4] J. F. BENDERS, *Partitioning procedures for solving mixed-variables programming problems*, Numer. Math., 4 (1962), pp. 238–252.

[5] A. J. BERGER, J. M. MULVEY, E. ROTHBERG, AND R. J. VANDERBEI, *Solving multistage stochastic programs using tree dissection*, Technical Report SOR-95-07, Princeton University, 1995.

[6] A. BERKELAAR, C. DERT, B. OLDENKAMP, AND S. ZHANG, *A primal-dual decomposition-based interior point approach to two-stage stochastic linear programming*, EI-Report 9918, Econometric Institute, Erasmus University, Rotterdam, The Netherlands, Mar. 1999.

[7] J. R. BIRGE, *Stochastic programming computations and applications*, INFORMS J. Comput., 9 (1997), pp. 111–133.

[8] J. R. BIRGE, M. A. H. DEMPSTER, H. I. GASSMANN, E. A. GUNN, A. J. KING, AND S. W. WALLACE, *A standard input format for multiperiod stochastic linear programs*, COAL newsletter, (1987), pp. 1–19.

[9] J. R. BIRGE AND D. F. HOLMES, *Efficient solution of two-stage stochastic linear programs with interior point methods*, Comput. Optim. Appl., 1 (1992), pp. 245–276.

[10] J. R. BIRGE AND F. LOUVEAUX, *Introduction to Stochastic Programming*, Springer-Verlag, New York, 1997.

[11] J. R. BIRGE AND L. QI, *Computing block-angular Karmarkar projections with applications to stochastic programming*, Management Sci., 34 (1988), pp. 1472–1479.

[12] J. BLOMVALL, *Optimization of Financial Decisions Using a New Stochastic Programming Method*, PhD thesis, Linköpings universitet, Linköping, Sweden, 2001.

[13] J. CZYZYK, R. FOURER, AND S. MEHROTRA, *A study of the augmented system and column-splitting approaches for solving two-stage stochastic linear programs by interior-point methods*, ORSA J. Comput., 7 (1995), pp. 474–490.

[14] G. B. DANTZIG, *Linear programming under uncertainty*, Management Sci., 1 (1955), pp. 197–206.

[15] M. A. H. DEMPSTER, N. H. PEDRÓN, E. A. MEDOVA, J. E. SCOTT, AND A. SEMBOS, *Planning logistics operations in the oil industry*, J. Operational Res. Society, (2000).

[16] K. FRAUENDORFER, *The stochastic programming extension of the Markowitz approach*, Int. J. Neural Mass-Parallel Comput. Inform. Syst., 5 (1995), pp. 449–460.

[17] K. FRAUENDORFER AND H. SIEDE, *Portfolio selection using multi-stage stochastic programming*, Central European J. Oper. Res., 7 (2000), pp. 277–290.

[18] I. GARRIDO AND M. C. STEINBACH, *A multistage stochastic programming approach in real-time process control*, Technical Report ZR-01-05, ZIB, Mar. 2001. Submitted for publication.

[19] J. GONDZIO AND R. KOUWENBERG, *High performance computing for asset liability management*, Oper. Res., (To appear).

[20] E. R. JESSUP, D. YANG, AND S. A. ZENIOS, *Parallel factorization of structured matrices arising in stochastic programming*, SIAM J. Optim., 4 (1994), pp. 833–846.

[21] P. KALL AND S. W. WALLACE, *Stochastic Programming*, Wiley, New York, 1994.

[22] A. J. KING, *Stochastic programming problems: Examples from the literature*, in Numerical Techniques for Stochastic Optimization, Y. Ermoliev and R. J.-B. Wets, eds., Springer-Verlag, New York, 1988, pp. 543–567.

[23] I. J. LUSTIG, J. M. MULVEY, AND T. J. CARPENTER, *Formulating two-stage stochastic programs for interior point methods*, Oper. Res., 39 (1991), pp. 757–770.

[24] R. T. ROCKAFELLAR, *Linear-quadratic programming and optimal control*, SIAM J. Control Optim., 25 (1987), pp. 781–814.

[25] ———, *Duality and optimality in multistage stochastic programming*, Ann. Oper. Res., 85 (1999), pp. 1–19.

[26] R. T. ROCKAFELLAR AND R. J.-B. WETS, *Generalized linear-quadratic problems of deterministic and stochastic optimal control in discrete time*, SIAM J. Control Optim., 28 (1990), pp. 810–822.

[27] E. SCHWEITZER, *An interior random vector algorithm for multistage stochastic linear programs*, SIAM J. Optim., 8 (1998), pp. 956–972.

[28] M. C. STEINBACH, *Fast Recursive SQP Methods for Large-Scale Optimal Control Problems*, Ph. D. dissertation, University of Heidelberg, Germany, 1995.

[29] ———, *Structured interior point SQP methods in optimal control*, Z. Angew. Math. Mech., 76 (1996), pp. S59–S62.

[30] ———, *Recursive direct optimization and successive refinement in multistage stochastic programs*, Preprint SC-98-27, ZIB, 1998.

[31] ———, *Recursive direct algorithms for multistage stochastic programs in financial engineering*, in Operations Research Proceedings 1998, P. Kall and H.-J. Lüthi, eds., New York, 1999, Springer-Verlag, pp. 241–250.

[32] ———, *Hierarchical sparsity in multistage convex stochastic programs*, in Stochastic Optimization: Algorithms and Applications, S. P. Uryasev and P. M. Pardalos, eds., Kluwer Academic Publishers, 2001, Dordrecht, The Netherlands, pp. 363–388.

[33] ———, *Markowitz revisited: Mean-variance models in financial portfolio analysis*, SIAM Rev., 43 (2001), pp. 31–85.

[34] M. C. STEINBACH, H. G. BOCK, G. V. KOSTIN, AND R. W. LONGMAN, *Mathematical optimization in robotics: Towards automated high speed motion planning*, Surveys Math. Indust., 7 (1998), pp. 303–340.

[35] J. STOER AND C. WITZGALL, *Convexity and Optimization in Finite Dimensions I*, Springer-Verlag, 1970.

[36] J. VAN SLYKE AND R. J. WETS, *L-shaped linear programs with applications to optimal control and stochastic programming*, SIAM J. Appl. Math., 17 (1969), pp. 638–663.

[37] S. J. WRIGHT, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, PA, 1997.

[38] D. YANG AND S. A. ZENIOS, *A scalable parallel interior point algorithm for stochastic linear programming and robust optimization*, Comput. Optim. Appl., 7 (1997), pp. 143–158.

[39] Y. YE, *Interior Point Algorithms. Theory and Analysis*, Wiley, New York, 1997.

MARC C. STEINBACH, KONRAD-ZUSE-ZENTRUM FÜR INFORMATIONSTECHNIK BERLIN, DEPARTMENT OPTIMIZATION, TAKUSTR. 7, 14195 BERLIN-DAHLEM, GERMANY

E-mail address: `steinbach@zib.de`

URL: `http://www.zib.de/steinbach`