Simultaneous Optimal Control and Discrete Stochastic Sensor Selection[⋆]

D. Bernardini[1], D. Muñoz de la Peña[2], A. Bemporad[1,⋆⋆], and E. Frazzoli[3]

[1] Department of Information Engineering, University of Siena, Italy
{bernardini,bemporad}@dii.unisi.it
[2] Dep. de Ingeniería de Sistemas y Automática, University of Seville, Spain
davidmps@cartuja.us.es
[3] Massachusetts Institute of Technology, MA, USA
frazzoli@mit.edu

Abstract. In this paper we present the problem of combining optimal control with efficient information gathering in an uncertain environment. We assume that the decision maker has the ability to choose among a discrete set of sources of information, where the outcome of each source is stochastic. Different sources and outcomes determine a reduction of uncertainty, expressed in terms of constraints on system variables and set-points, in different directions. This paper proposes an optimization-based decision making algorithm that simultaneously determines the best source to query and the optimal sequence of control moves, according to the minimization of the expected value of an index that weights both dynamic performance and the cost of querying. The problem is formulated using stochastic programming ideas with decision-dependent scenario trees, and a solution based on mixed-integer linear programming is presented. The results are demonstrated on a simple supply-chain management example with uncertain market demand.

1 Introduction

A large number of problems in production planning and scheduling, location, transportation, finance, and engineering design require taking optimal decisions in the presence of uncertainty. Uncertainty, for instance, governs the prices of fuels, the availability of electricity, and the demand for chemicals. In general, these uncertainties affect the constraints of the corresponding optimization problem. A standard approach to deal with uncertain constraints is to guarantee constraint satisfaction for all possible cases. In order to reduce the conservativeness of this solution, additional information about the uncertainties may be gathered, for example by carrying a demand field study to better estimate the value of future demand of a certain product in a production planning problem. With this additional information, the optimization problem is updated in a less conservative

[⋆] This work was partially supported by the European Commission under the HYCON Network of Excellence, contract number FP6-IST-511368, and under the WIDE project, contract number FP7-IST-224168.

[⋆⋆] Corresponding author.

R. Majumdar and P. Tabuada (Eds.): HSCC 2009, LNCS 5469, pp. 61–75, 2009.

way and an improved solution is obtained. In addition, with the current developments in networked control systems (NCS) [18,7,12], efficient information gathering has become a very relevant problem in modern industrial automation. Possible examples of this framework are given by control over wireless networks, where communication is subject to strong energy constraints, and more in general by any kind of NCS in which measurement acquisition is expensive. For such process control problems a selection criterion for the kind of information that is convenient to retrieve is recommended.

In general, however, the outcome of these information queries is not known a priori. Moreover, queries have fixed costs that do not depend on the quantitative outcome of the information gathered, i.e., costs associated with the querying process per se. This poses a difficult problem of whether a query would be profitable or not. The difficulty increases when there are several possible queries at hand and, even more difficult, when a whole sequence of queries must be planned. There are different ways of approaching the problem. It can be cast as a Markov decision problem (MDP) [13], but the cardinality of the state space of this representation grows exponentially with the number of events, due to the number of possible combinations of events which could take place. Hence, the exact solution of such a problem becomes computationally intractable very quickly, even for relatively small problems. The approach taken in [6] for a similar problem (the bridge problem) is based on reinforcement learning, which is a set of techniques aimed at approximating the MDP value function. We refer the reader to the literature on the subject for further details [4,17].

In this paper we take a different route and propose a stochastic recursive optimization scheme in which we have to decide not only an optimal sequence of future control actions, but also which measurements/queries are worth to be carried out. Each query is defined by its own fixed cost and a series of possible outcomes described by a discrete probability function. The constraints on the sequence of future actions and performance indices depend on such outcomes. Consequently, the optimal control problem becomes stochastic as well, for which we employ a stochastic programming formulation to minimize expected values under stochastic constraint sets. Stochastic programming is a special class of mathematical programming that involves optimization under uncertainty (see [5,9,14]). The first applications of stochastic programming date back to the 50's and nowadays it is becoming a mature theory that is successfully applied in several domains [15]. A stochastic programming problem is defined by a sequence of random events and recourse decisions. Each decision is a different stage and stages are divided by random events. In the proposed formulation, there are two stages, that is, two sets of decision variables separated by a random event: First the query has to be chosen without knowing the outcome of the response, then the outcome of the query is obtained (the random event takes place) and the second stage decision (the dynamic optimization variables) is made based on this information.

For long optimization horizons, we advocate the use of recursive shorter-horizon optimization to obtain suboptimal solutions within a manageable computational burden (see for example [3] for the application of recursive stochastic hybrid optimal control in the management of power distribution networks). The proposed scheme is demonstrated on a supply-chain management example in which the future demand is uncertain, but additional information can be obtained from market studies.

2 Stochastic Querying Model

Consider the generic problem of linear programming (LP)

$$\begin{aligned} \min_{z} \ & c'z \\ \text{s.t.} \ & z \in \mathcal{Z}, \end{aligned} \tag{1}$$

in which $\mathcal{Z}$ is a polyhedron that defines the region of feasibility. As in general by expanding $\mathcal{Z}$ one improves the optimum achieved in (1), we consider the case in which we can perform a query Q in order to obtain additional information that allows us to enlarge the size of $\mathcal{Z}$. The main idea is that in the presence of uncertainty, if a robust approach is taken, the feasible set takes into account all possible values of uncertain parameters. Hence, by obtaining additional information that reduces the set of possible values of the uncertain parameters, the size of $\mathcal{Z}$ increases, and hence the optimal cost is improved. We define a query Q as follows:

Definition 1. *A* query $Q(q) \in \mathcal{Q}$ *is defined as*

$$Q(q) = \{C(q), \mathcal{V}_q\}, \ q \in \{0, 1, \ldots, n_q\}, \tag{2}$$

where q is the query index*, $C(q) \geq 0$ is the* querying cost*, and $\mathcal{V}_q = \{V_1^q, V_2^q, \ldots, V_{m_q}^q\}$ is the set of the m_q possible* outcomes*.*

Definition 2. *A* query outcome V_v^q *for the query q is defined as*

$$V_v^q = \{\mathcal{Z}(q, v)\}, \tag{3}$$

where $v \in \{1, 2, \ldots, m_q\}$ is the outcome index*, and $\mathcal{Z}(q, v)$ is the* updated feasibility set*.*

Note that, in general, the number m_q of possible outcomes depends on the query q. For compactness of notation, in the sequel we will often refer to a "query" $Q(q)$ directly by its corresponding query index q, and to an "outcome" V_v^q directly by its corresponding outcome index v. To model the case where the source of information is not queried at all, we introduce the *null query*, indexed by $q = 0$ and defined below.

Definition 3. *The* null query *is defined as*

$$Q(0) = \{0, \mathcal{V}_0\}, \qquad \mathcal{V}_0 = \left\{V_1^0 = \{\mathcal{Z}\}\right\}, \tag{4}$$

The query $Q(q)$ can be chosen among the finite set $\mathcal{Q}$ of different queries, however the information obtained from each query is stochastic. Each query is defined by a cost $C(q)$ and a set of possible outcomes $\mathcal{V}_q$ with a given probability, which we assume to be available.

Definition 4. *For every query $q \in \{0, 1, \ldots, n_q\}$, the* outcome probability distribution *is a discrete distribution given by*

$$\mathcal{P}_i = \left\{ p_{ij} : p_{ij} = Pr[v = V_j^i | q = i],\ j = 1, 2, \ldots, m_i,\ \sum_{j=1}^{m_i} p_{ij} = 1 \right\}. \tag{5}$$

The objective is to choose the query $Q(q) \in \mathcal{Q}$ such that J_q is minimized, where J_q is the expected value of the cost function with respect to the possible outcomes of the query plus the cost of the query itself

$$\min_{q \in \mathcal{Q}}\ \mathbb{E}_v\left[J_{qv} | q\right] + C(q), \tag{6}$$

with J_{qv} the optimal cost corresponding to the outcome v of the query q defined as

$$J_{qv} = \min_{z}\ c'z \tag{7a}$$

$$\text{s.t. } z \in \mathcal{Z}(q, v). \tag{7b}$$

Problem 7 can be posed as a two-stage stochastic optimization problem [5,9,14]. As observed earlier, here the query q is the first-stage variable, and z is the second-stage variable which is decided after the random outcome event v takes place. In the following section we propose to apply this general framework to the the problem of combining optimal control with efficient information gathering in an uncertain environment.

3 Simultaneous Optimal Control and Sensor Selection Problem

Consider the discrete-time linear model of the process

$$x(t+1) = Ax(t) + Bu(t), \tag{8}$$

where the input $u \in \mathbb{R}^{n_u}$ and the input rate $\Delta u(t) = u(t) - u(t-1)$ are subject to known component-wise constraints[1] $u_{\min} \le u \le u_{\max}$, and $\Delta u_{\min} \le \Delta u \le \Delta u_{\max}$. The state $x \in \mathbb{R}^{n_x}$ is subject to uncertain constraints. The only available a priori information on the admissible state set is given by the set-membership relation $x \in \mathcal{X}$, where $\mathcal{X}$ is a conservative estimate of the admissible state set that guarantees robust constraint satisfaction for all possible values of queries

[1] Here component-wise constraints are considered for simplicity, but it is straightforward to extend the approach to the more general case of polytopic constraints.

and outcomes. The goal of the control action is to make the state $x(t)$ and the input $u(t)$ track an uncertain reference value $r_x(t)$, $r_u(t)$, respectively, where $r(t) = \left[\begin{smallmatrix} r_x(t) \\ r_u(t) \end{smallmatrix}\right] \in \mathcal{R}$. The set $\mathcal{R}$ is a conservative estimate of all the possible values that the reference can take[2].

Without additional information, a recursive optimal control problem formulation based on model (8), the conservative estimates $\mathcal{X}$ and $\mathcal{R}$, and a min-max cost function can be formulated using standard min-max model predictive control ideas [16,1,11]. We refer to this problem as the standard min-max problem. However, in this paper, we assume that every T time steps the decision maker is allowed to reduce the conservativeness by querying additional sources of information at a certain cost. This additional information in general may provide a reduced conservativeness on the admissible state sets (i.e., a larger domain $\mathcal{X}$), and/or a better estimate of the reference $r(t)$ (i.e., a smaller domain $\mathcal{R}$). In both cases, the obtained solution is less conservative, with consequent improvement of the overall performance of the process. By following the problem formulation of Section 2, the outcome v related to the query q is denoted by

$$V_v^q = \{\mathcal{X}(q, v|t), \mathcal{R}(q, v|t)\}, \tag{9}$$

where $\mathcal{X}(q, v|t)$, $\mathcal{R}(q, v|t)$ are the updated state constraints and reference sets, such that $\mathcal{X}(q, v|t) \supseteq \mathcal{X}(t)$, $\mathcal{R}(q, v|t) \subseteq \mathcal{R}(t)$. Moreover, the outcome set for the *null* query is

$$V_1^0 = \{\mathcal{X}(t), \mathcal{R}(t)\}, \tag{10}$$

where $\mathcal{X}(t)$, $\mathcal{R}(t)$ is the available information at time t on state constraints and reference set. Note that we consider here problems in which the estimates of the uncertain sets are time varying. This may be the case for instance in which the outcomes obtained after each query accumulate.

The querying mechanism can be modeled in different ways, for example by introducing delays between the query transmission and the availability of the outcome. For simplicity, in the following sections we restrict ourselves to the following assumption.

Assumption 1. *The outcome V_v^q of a query $Q(q)$ performed at time step t is immediately available, and the provided information is supposed to be significant only for time step $t+1$.*

We aim at defining a stochastic optimal control setup that, at each time step t, provides at the same time a sequence of optimal input values $u(t)$, $u(t+1)$, ..., $u(t+N-1)$, $N \geq 1$, and the most profitable query $q(t)$, by taking into account model (8), the set of possible outcomes (9), and the corresponding probability distributions (5). As mentioned before, we assume that a query can be done every T time steps, where T is constant and such that $T \geq 1$. We also assume that a query is done at time step $t = 0$. This implies that a query will be done

[2] Note that this setup can be easily extended to the case of bounded additive disturbances, as they can be modeled without loss of generality by means of more conservative state or input constraints.

at time steps $t = kT$, $k \in \mathbb{Z}$, $k \geq 0$. Given a generic time t, the next query will be henceforth carried out at time $t + H$, where

$$H = \left\lceil \frac{t}{T} \right\rceil T - t, \tag{11}$$

and where $\lceil a \rceil$ denotes the smallest integer greater than or equal to a. When H is smaller than the optimal control horizon N, the future query has to be decided by the optimal decision mechanism, otherwise a standard min-max problem (no query) is solved.

Note that in general, instead of choosing off-line a constant value for T, any time-varying, state-dependent interval $T(t)$ could be considered, as long as the condition $0 < H(t) < N \Rightarrow H(t+1) \leq H(t) - 1$ is enforced to preserve the consistency of the receding horizon control.

Based on the above description, at time step $t \in \mathbb{N}$ the simultaneous optimal control and sensor selection problem is defined as

$$\min_{q} \; \mathbb{E}_v \left[J_{qv} | q \right] + cC(q) \tag{12a}$$

$$\text{s.t.} \; \begin{cases} q \in \{0, 1, 2, \ldots, n_q\} & \text{if } H < N, \\ q = 0 & \text{otherwise,} \end{cases} \tag{12b}$$

with

$$\begin{aligned} J_{ij} \;=\; \min_{\Delta u} \Bigg\{ \max_{r} \sum_{k=0}^{N-1} \ell(&x(t+k,i,j|t) - r_x(t+k,i,j|t), \\ &u(t+k,i,j|t) - r_u(t+k,i,j|t), \Delta u(t+k,i,j|t)) \Bigg\} \end{aligned} \tag{13a}$$

$$\begin{aligned}
\text{s.t. } & x(t+k+1,i,j|t) = Ax(t+k,i,j|t) + Bu(t+k,i,j|t), && (13b) \\
& u(t+k,i,j|t) = u(t+k-1,i,j|t) + \Delta u(t+k,i,j|t), && (13c) \\
& u_{\min} \leq u(t+k,i,j|t) \leq u_{\max}, && (13d) \\
& \Delta u_{\min} \leq \Delta u(t+k,i,j|t) \leq \Delta u_{\max}, && (13e) \\
& x(t+k,i,j|t) \in \begin{cases} \mathcal{X}(i,j|t) & \text{if } k = H+1, \\ \mathcal{X}(t) & \text{otherwise,} \end{cases} && (13f) \\
& r(t+k,i,j|t) \in \begin{cases} \mathcal{R}(i,j|t) & \text{if } k = H+1, \\ \mathcal{R}(t) & \text{otherwise,} \end{cases} && (13g) \\
& x(t,i,j|t) = x(t|t), && (13h) \\
& \Delta u(t+h,i,j|t) = \Delta u(t+h,w,z|t), \; \forall w \neq i, \; \forall z \neq j, && (13i) \\
& h = 0, 1, \ldots, \min(H, N) - 1, \\
& k = 0, 1, \ldots, N-1,
\end{aligned}$$

for $i = 0, 1, \ldots, n_q$, $j = 1, 2, \ldots, m_i$, where $x(t|t) = x(t)$ is the current state, used as the initial condition for the optimal control problem, $x(t+k,i,j|t)$, $r(t+k,i,j|t)$, $u(t+k,i,j|t)$, $\Delta u(t+k,i,j|t)$ are the predicted state, the input, the

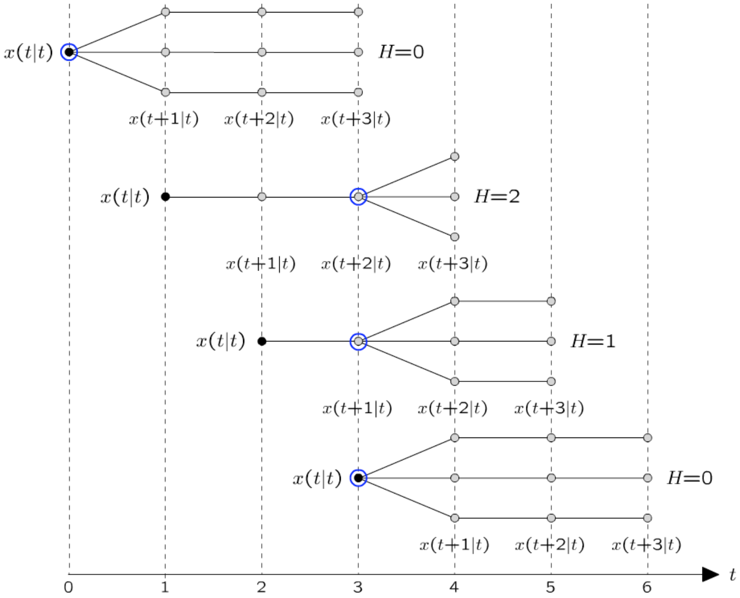

Fig. 1. Optimization tree structure for different time steps t, where to enforce causality the inputs are not branched until the query is performed ($N = T = 3$, circled dots denote decisions taken on q)

input rate and uncertain reference at time step $t + k$ corresponding to making a query i with the outcome j at time $t+H$, $r = \left[\begin{smallmatrix} r_x \\ r_u \end{smallmatrix}\right]$, $c \geq 0$ is the tradeoff coefficient between performance and querying costs, N is the prediction horizon, T is the time period between two consecutive queries, and the stage cost $\ell : \mathbb{R}^{n_x+2n_u} \to \mathbb{R}$ is a nonnegative function.

The positive scalar H defined in (11) represents the time step at which a query decision will take place. Until that time, the causality constraint (13i) enforces the same input sequence for all the possible sequences of states, regardless of the dependence on future decision on q. Note that the optimization problem has the time-varying structure depicted in Figure 1, as the imposed constraints at time t depend on the current value of $H = \left\lceil \frac{t}{T} \right\rceil T - t$.

In principle, Problem 13 is an infinite dimensional optimization problem, due to the maximization part that involves an infinite number of realizations of the reference r. However, it is well known that when the process is linear and the constraints and the cost function are convex, the max problem can be solved by considering only the "extreme" realizations, namely the vertices of the reference set $\mathcal{R}$ (see, e.g., [16]). In the next section we will exploit this property to reformulate Problem 13 as a stochastic mixed-integer linear programming (MILP) problem.

According to the aforementioned stochastic optimization nomenclature [5], Problem 12 is a two-stage optimization problem in which the second-stage variables are Δu's. Since only one decision on q is modeled in the problem, the proposed formulation is exact with respect to the system behavior only for $N \leq T$. For $N > T$ a more complex *multi-stage* stochastic programming formulation would be necessary. By using the two-stage formulation (12) also when $N > T$, i.e., by modeling just the first decision on q, one gets a conservative solution which does not exploit all the available information, but nonetheless is computationally more viable.

The following Algorithm 1 summarizes the proposed recursive stochastic simultaneous optimal control and sensor selection decision mechanism.

Algorithm 1. Recursive stochastic simultaneous optimal control and sensor selection.

```
For all $t \geq 0$:
    1. get $x(t)$ and compute $H$ as in (11);
    2. solve Problem 12 and get the optimal solution $q^*(t)$, $u^*(t,i,j|t)$, $\forall i,j$;
    3. if $H = 0$
        3.1. perform the query $q^*(t)$ and get the query outcome $v^*(t)$;
        3.2. set $u(t) = u(t, q^*, v^*|t)$ in (8);
    4. else
        4.1. set $u(t) = u(t, 0, 1|t)$ in (8);
    5. end.
```

Next section focuses on computational methods for solving Problem 12. A closed-loop stability analysis of the receding horizon control scheme proposed by Algorithm 1 is beyond the scope of this paper and will addressed in future works, based on an adaptation of convergence properties existing for deterministic min-max model predictive control schemes [16] to the present stochastic min-max setting.

4 Solution Methods

Let the stage cost ℓ be based on infinity norms

$$\ell\left(x - r_x, u - r_u, \Delta u\right) = ||Q_x(x - r_x)||_\infty + ||Q_u(u - r_u)||_\infty + ||Q_{\Delta u}\Delta u||_\infty, \quad (14)$$

and, for the sake of generality, assume that a terminal cost

$$\ell\left(x(t+N, i, j|t) - r_x\right) = \|Q_N(x(t+N, i, j|t) - r_x)\|_\infty \quad (15)$$

is added in the cost function (13a), where Q_x, Q_u, $Q_{\Delta u}$, Q_N are full row-rank matrices, and $\|Qx\|_\infty = \max_{i=1,\ldots,n_x} |Q_i x|$ with Q_i the ith row of Q[3]. In this case Problem 12 can be solved using an MILP problem by following the so-called "scenario enumeration" approach of stochastic programming [5], as detailed below, where we exploit the convexity of (14), (15), to get rid of the max problem

[3] The results of this paper extend to any convex piecewise affine function ℓ.

in (12) through enumeration of vertices, introduce slack variables that upper bound each stage term of the stage cost [1], and use big-M techniques to transform a multiplication between a binary variable and a continuous variable into a set of linear constraints [19]. The case of quadratic cost in (14) can also be handled similarly, by using mixed-integer quadratic programming (MIQP). Then, Problem 12 can be formulated as the following MILP

$$\min_{\delta,\Delta u,F,\gamma} \sum_{i=0}^{n_q} F_i \tag{16a}$$

$$\begin{aligned}
\text{s.t. } & x(t+k+1,i,j|t) = Ax(t+k,i,j|t) + Bu(t+k,i,j|t), \\
& u(t+k,i,j|t) = u(t+k-1,i,j|t) + \Delta u(t+k,i,j|t), \\
& u_{\min} \le u(t+k,i,j|t) \le u_{\max}, \\
& \Delta u_{\min} \le \Delta u(t+k,i,j|t) \le \Delta u_{\max}, \\
& x(t+k,i,j|t) \in \begin{cases} \mathcal{X}(i,j|t) & \text{if } k = H+1, \\ \mathcal{X}(t) & \text{otherwise}, \end{cases} \\
& x(t,i,j|t) = x(t|t), \\
& \Delta u(t+h,i,j|t) = \Delta u(t+h,w,z|t),\ \forall w \neq i,\ \forall z \neq j, \\
& \underbrace{\begin{aligned} & \gamma_{ij}^{kx} \ge \|Q_x(x(t+k,i,j|t) - r_x(t+k,i,j|t))\|_\infty, \\ & \quad k = 0,\dots,N-1, \\ & \gamma_{ij}^{ku} \ge \|Q_u(u(t+k,i,j|t) - r_u(t+k,i,j|t))\|_\infty, \\ & \quad k = 0,\dots,N-1, \\ & \gamma_{ij}^{k\Delta u} \ge \|Q_{\Delta u}\Delta u(t+k,i,j|t)\|_\infty, \\ & \quad k = 0,\dots,N-1, \\ & \gamma_{ij}^{Nx} \ge \|Q_N(x(t+N,i,j|t) - r_x(t+N,i,j|t))\|_\infty, \end{aligned}}_{\forall r(t+k,i,j|t) \in \begin{cases} \mathcal{R}_v(i,j|t) & \text{if } k = H+1, \\ \mathcal{R}_v(t) & \text{otherwise}, \end{cases}} \qquad (16b) \\
& -M\delta_i \le F_i \le \sum_{j=1}^{m_i} p_{ij}(t) \left(\gamma_{ij}^{Nx} + \sum_{k=0}^{N-1} \gamma_{ij}^{kx} + \gamma_{ij}^{ku} + \gamma_{ij}^{k\Delta u} \right) \\
& \quad + cC(i) + M(1-\delta_i), \qquad (16c) \\
& M\delta_i \ge F_i \ge \sum_{j=1}^{m_i} p_{ij}(t) \left(\gamma_{ij}^{Nx} + \sum_{k=0}^{N-1} \gamma_{ij}^{kx} + \gamma_{ij}^{ku} + \gamma_{ij}^{k\Delta u} \right) \\
& \quad + cC(i) - M(1-\delta_i), \qquad (16d) \\
& \sum_{i=0}^{n_q} \delta_i = 1,\ \delta_i \in \{0,1\},\ i = 0,\dots,n_q, \\
& h = 0,1,\dots,\min(H,N)-1, \\
& i = 0,1,\dots,n_q,\ j = 1,2,\dots,m_i,\ k = 0,\dots,N-1.
\end{aligned}$$

where the array of binary variables $\delta = \{\delta_0, \delta_1, \dots, \delta_{n_q}\}$, $\delta_i \in \{0,1\}$, $i = 0,\dots,n_q$, one for every possible query choice, is used to choose the query to be done among all possibilities; that is, if query i is chosen, then $\delta_i = 1$ and the rest are equal to zero. The slack variables γ_{ij}^{kx}, γ_{ij}^{ku}, $\gamma_{ij}^{k\Delta u}$, γ_{ij}^{Nx} in (16b) define the value of the min-max problem (13) for every couple (i, j), where $\mathcal{R}_v(t)$, $\mathcal{R}_v(i, j|t)$ are the sets of the vertices of $\mathcal{R}(t)$ and $\mathcal{R}(i, j|t)$, respectively. Note that (16b) are linear constraints, since in general $\gamma \geq \|z\|_\infty$ can be rewritten as $\gamma \geq \pm z_i$, $\forall i$. By means of the big-M constraints (16c)-(16d), all the continuous variables F_i take zero value, except for the one referred to the chosen query. Then, the cost function (16a) is equivalent to (12a). M is a large enough positive scalar, satisfying the condition

$$\begin{aligned} M \geq \textstyle\sum_{j=1}^{m_i} p_{ij}(t)\big(&\|Q_N(x(t+N,i,j|t) - r_x(t+N,i,j|t))\|_\infty \\ &+ \sum_{k=1}^{N-1} \|Q_x(x(t+k,i,j|t) - r_x(t+k,i,j|t))\|_\infty + \|Q_u(u(t+k,i,j|t) \\ &- r_u(t+k,i,j|t))\|_\infty + \|Q_{\Delta u}\Delta u(t+k,i,j|t)\|_\infty\big) + cC(i), \end{aligned}$$

for all $i = 0, 1, \ldots, n_q$. Note that it is not strictly necessary to model the input sequences for all the possible pairs (q, v), since only $\max_q (m_q)$ scenarios are evaluated simultaneously. However, in this case a number of additional constraints would be needed, resulting in a higher computational burden. Reducing the number of the inputs can be desirable if some or all of them are integer variables.

5 Illustrative Example

The use of receding horizon control policies in supply chain management have been investigated in [10,8], and approached by hybrid techniques in [2]. In this paper we consider the supply chain shown in Figure 2, where a single product is processed through a network of four nodes. A product is distributed, stored, and sold to the customer. The goal of the control problem is to minimize a performance index, mainly given by the satisfaction of customer demand and production costs, while fulfilling constraints on production, storage and transport capacities. The process is modeled as

$$F_1(t+1) = F_1(t) + P_1(t) - T_{11}(t) - T_{12}(t), \tag{17a}$$

$$F_2(t+1) = F_2(t) + P_2(t) - T_{21}(t) - T_{22}(t), \tag{17b}$$

$$R_1(t+1) = R_1(t) + T_{11}(t) + T_{21}(t) - D_1(t), \tag{17c}$$

$$R_2(t+1) = R_2(t) + T_{12}(t) + T_{22}(t) - D_2(t), \tag{17d}$$

where, at time t, $P_i(t)$ is the number of products which enter the supply chain and are stored in Factory i, $T_{ij}(t)$ is the number of transported products from Factory i to Retailer j, and $D_j(t)$ is the number of products sold by Retailer j.

We define the state vector $x = [\,F_1\ F_2\ R_1\ R_2\,]' \in \mathbb{R}^4$ and the input vector $u = [\,P_1\ P_2\ T_{11}\ T_{12}\ T_{21}\ T_{22}\ \bar{D}_1\ \bar{D}_2\,]' \in \mathbb{R}^8$, where $\bar{D}_i$ is the nominal value for the

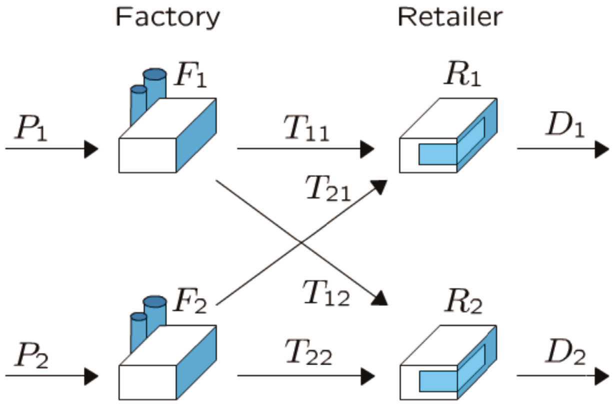

Fig. 2. Supply chain scheme

demand D_i. Then, the dynamics (8) of the supply chain model is described by the matrices

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 \end{bmatrix}. \tag{18}$$

The bounds on states and inputs are

$$x_{max} = \begin{bmatrix}100\ 100\ 100\ 100\end{bmatrix}',\ u_{max} = \begin{bmatrix}100\ 100\ 50\ 50\ 50\ 50\ 100\ 100\end{bmatrix}', \tag{19a}$$
$$x_{min} = \begin{bmatrix}0\ 0\ 20\ 20\end{bmatrix}',\ u_{min} = \begin{bmatrix}0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\end{bmatrix}', \tag{19b}$$

respectively, and input increments are considered unbounded. In addition, the model is subject to the following constraints on product availability:

$$T_{11}(t) + T_{12}(t) \leq F_1(t), \tag{20a}$$
$$T_{21}(t) + T_{22}(t) \leq F_2(t), \tag{20b}$$
$$D_1(t) \leq R_1(t), \tag{20c}$$
$$D_2(t) \leq R_2(t). \tag{20d}$$

In this example the state constraints are fully known, but customer demand is uncertain. In particular, we assume that at every time step t customer demand can be described by two probability distributions, called *low mode* and *high mode*, respectively. They are essentially modeled as a mixture of Gaussians, normalized in the demand space:

$$\begin{bmatrix} D_1(t) \\ D_2(t) \end{bmatrix} \sim \begin{cases} \frac{\mathcal{N}(\mu_0,\Sigma_0)}{\int_0^{80}\int_0^{80}\mathcal{N}(\mu_0,\Sigma_0)dD_1dD_2} & \text{with 70\% prob. \textit{(low mode)}} \\ \frac{\mathcal{N}(\mu_0,\Sigma_0)+0.75\mathcal{N}(\mu_1,\Sigma_1)}{\int_0^{80}\int_0^{80}(\mathcal{N}(\mu_0,\Sigma_0)+0.75\mathcal{N}(\mu_1,\Sigma_1))dD_1dD_2} & \text{with 30\% prob. \textit{(high mode)}} \end{cases} \tag{21}$$

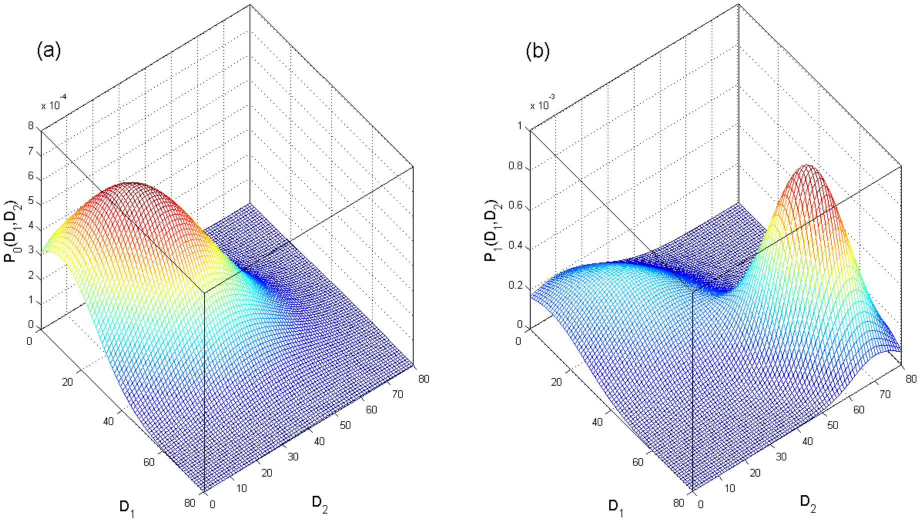

Fig. 3. Probabilistic models for customer demand in low mode (a) and high mode (b)

where $\mu_0 = \begin{bmatrix} 20 \\ 20 \end{bmatrix}$, $\mu_1 = \begin{bmatrix} 60 \\ 60 \end{bmatrix}$, $\Sigma_0 = \begin{bmatrix} 400 & 200 \\ 200 & 400 \end{bmatrix}$, $\Sigma_1 = \begin{bmatrix} 100 & 50 \\ 50 & 100 \end{bmatrix}$. The probability distributions of the customer demand associated to each of the two modes are shown in Figure 3. The decision maker is supposed to know the current demand mode at each time step by freely available market polls, but to ignore the exact value of the demand. Henceforth the reference values are time-varying and uncertain. Let $\mathcal{R}_D$ denote the subset of the reference set related to the demand D_1, D_2. We assume to be able to perform two different queries to get an approximated description of the actual demand probability distribution: The first is simpler and cheaper, the second is more accurate but more expensive. The numerical values for available queries and their outcomes are given in Table 1. Note that by Assumption 1 we consider a query outcome to be reliable only for the time step following the time at which the query was sent.

We consider four different decision-making policies:

(1) A *deterministic* policy obtained by setting $q(t) = 0$, $\forall t$, corresponding to a standard min-max problem where no additional information is retrieved by the querying mechanism (LP);
(2) A *stochastic random* policy in which $q(t)$ is picked up randomly in $\{1, \dots, n_q\}$, $\forall t$, and, therefore, $q(t)$ does not depend on the current state $x(t)$ of the model (LP);
(3) A *stochastic heuristics-based* policy in which $q(t)$ is selected according to deterministic conditions on the market state (LP). The following rule is applied: at time t, if the market is in *high mode*, select $q(t) = 2$, else select $q(t) = 1$;
(4) The *stochastic optimized* policy of Problem 12 (MILP).

Table 1. Queries and outcomes definition

(q,v)	$C(q)$	p_{qv} in Low Mode	p_{qv} in High Mode	$\mathcal{R}_D(q,v\|t)$
$(0,1)$	0	$p_{01,L}=1$	$p_{01,H}=1$	$r_{min}=[\ 0\ \ 0]$, $r_{max}=[80\ 80]$
$(1,1)$	1	$p_{11,L}=0.804$	$p_{11,H}=0.423$	$r_{min}^{11}=[\ 0\ \ 0]$, $r_{max}^{11}=[40\ 80]$
$(1,2)$	1	$p_{12,L}=0.196$	$p_{12,H}=0.577$	$r_{min}^{12}=[40\ \ 0]$, $r_{max}^{12}=[80\ 80]$
$(2,1)$	15	$p_{21,L}=0.684$	$p_{21,H}=0.351$	$r_{min}^{21}=[\ 0\ \ 0]$, $r_{max}^{21}=[40\ 40]$
$(2,2)$	15	$p_{22,L}=0.120$	$p_{22,H}=0.072$	$r_{min}^{22}=[\ 0\ 40]$, $r_{max}^{22}=[40\ 80]$
$(2,3)$	15	$p_{23,L}=0.120$	$p_{23,H}=0.072$	$r_{min}^{23}=[40\ \ 0]$, $r_{max}^{23}=[80\ 40]$
$(2,4)$	15	$p_{24,L}=0.076$	$p_{24,H}=0.505$	$r_{min}^{24}=[40\ 40]$, $r_{max}^{24}=[80\ 80]$

Table 2. Simulation results

Control policy	Performance J_{exp}	Avg. CPU time
Deterministic min-max with *null* query	401.90	16.0 ms
Stochastic min-max with random query selection	360.84	16.9 ms
Stochastic min-max with heuristic query selection	345.76	16.7 ms
Stochastic min-max with optimized query selection	319.74	36.3 ms

We run $N_s = 10$ simulations of $T_{sim} = 10$ time steps each, using parameters $T = 1$, $N = 4$, $c = 1$, $Q_{\Delta u} = 0$, $Q_x = Q_N = \text{Diag}([0.1,\ 0.1,\ 0.2,\ 0.2])$, $Q_u = \text{Diag}([10,\ 10,\ 0.1,\ 0.2,\ 0.2,\ 0.1,\ 10,\ 10])$. The initial state is $x(0) = [40\ 40\ 60\ 60]'$. Table 2 shows the obtained results in terms of the achieved average performance evaluated as

$$\begin{aligned} J_{exp} = \frac{1}{N_s T_{sim}} \sum_{i=1}^{N_s} \sum_{t=1}^{T_{sim}} \big(& \|Q_x(x^i(t) - r_x^i(t))\|_\infty \\ & + \|Q_u(u^i(t) - r_u^i(t))\|_\infty + \|Q_{\Delta u} u^i(t)\|_\infty + cC(q^i(t))\big), \end{aligned} \tag{22}$$

where $i = 1, \ldots, N_s$ indexes the state, input, references, and query values related to the i-th simulation. The table also reports the average CPU time for solving Problem 12 on a Macbook 2.4GHz running Matlab 7.6 and Cplex 9.0.

As one can see from the results reported in Table 2, the proposed stochastic min-max policy achieves the best average performance, with an improvement of 20.4% with respect to the deterministic min-max policy, an additional 11.4% with respect to the stochastic min-max policy with random query, and a further 7.5% with respect to the stochastic min-max policy with heuristics-based query. Moreover, the computation times for all the policies are of the same order of magnitude for this particular application, which demonstrates the viability of the methodology from a computational viewpoint.

6 Conclusions

In this paper we proposed a stochastic programming approach to the problem of simultaneous optimal information gathering and decision making in an uncertain environment. In particular, we dealt with linear optimization problems in which the feasibility set can be enlarged via a set of possible queries with stochastic outcomes. This class of problems was posed as a two-stage mixed integer stochastic optimization problems with endogenous uncertainty, that can be solved recursively in time for optimal performance of systems subject to uncertain constraints and uncertain references, where it is possible to reduce uncertainty bounds through queries. The proposed scheme minimizes the expected optimal cost with respect to the chosen query, while still guaranteeing robust constraint satisfaction. The results are demonstrated using a supply-chain example, which also shows the viability of the methodology from a computational viewpoint.

References

1. Bemporad, A., Borrelli, F., Morari, M.: Min-max control of constrained uncertain discrete-time linear systems. IEEE Transactions On Automatic Control 48(9), 1600–1606 (2003)
2. Bemporad, A., Giorgetti, N.: Logic-based methods for optimal control of hybrid systems. IEEE Transactions On Automatic Control 51(6), 963–976 (2006)
3. Bemporad, A., Muñoz de la Peña, D., Piazzesi, P.: Optimal control of investments for quality of supply improvement in electrical energy distribution networks. Automatica 42(8), 1331–1336 (2006)
4. Bertsekas, D.P., Tsitsiklis, J.T.: Neuro-Dynamic Programming. Athena Scientific, Belmont (1996)
5. Birge, J.R., Louveaux, F.V.: Introduction to Stochastic Programming. Springer, New York (1997)
6. Blei, D.M., Kaelbling, L.P.: Shortest paths in a dynamic uncertain environment. In: IJCAI Workshop on Adaptive Spatial Representations of Dynamic Environments (1999)
7. Chong, C.Y., Kumar, S.P.: Sensor networks: Evolution, opportunities, and challenges. Proceedings of the IEEE 91, 1247–1256 (2003)
8. Perea-López, E., Ydstie, B.E., Grossmann, I.E.: A model predictive control strategy for supply chain optimization. Computers and Chemical Engineering 27(8-9), 1201–1218 (2003)
9. Kall, P., Wallace, S.W.: Stochastic Programming. Wiley, Chichester (1994)
10. Brauna, M.W., Rivera, D.E., Floresa, M.E., Carlyleb, W.M., Kempf, K.G.: A Model Predictive Control framework for robust management of multi-product, multi-echelon demand networks. Annual Reviews in Control 27(2), 229–245 (2003)
11. Muñoz de la Peña, D., Alamo, T., Bemporad, A., Camacho, E.F.: A decomposition algorithm for feedback min-max model predictive control. IEEE Transactions On Automatic Control 51(10), 1688–1692 (2006)
12. Neumann, P.: Communication in industrial automation - what is going on? Control Engineering Practice 15, 1332–1347 (2007)
13. Puterman, M.L.: Markov Decision Processes. Wiley and Sons, Chichester (1994)
14. Ross, S.: Introduction to Stochastic Dynamic Programming. Academic Press, London (1983)
15. Sahinidis, N.V.: Optimization under uncertainty: State-of-the-art and opportunities. Computers & Chemical Engineering 28(6-7), 971–983 (2004)

16. Scokaert, P.O.M., Mayne, D.Q.: Min-max feedback model predictive control for constrained linear systems. IEEE Transactions On Automatic Control 43, 1136–1142 (1998)
17. Sutton, R.S., Barto, A.T.: Reinforcement Learning. MIT Press, Cambridge (1998)
18. Tatikonda, S., Mitter, S.: Control under communication constraints. IEEE Transactions on Automatic Control 49, 1056–1068 (2004)
19. Williams, H.P.: Model Building in Mathematical Programming, 3rd edn. John Wiley & Sons, Chichester (1993)