

Numerical solution of dynamic optimization problems with flexible inequality constraints by iterative dynamic programming

Cheng-Liang Chen *, Daim-Yuang Sun, Chia-Yuan Chang

Department of Chemical Engineering, National Taiwan University, Taipei 10617, Taiwan, ROC

Received 20 July 1999; received in revised form 27 April 2001; accepted 3 September 2001

Abstract

A solution strategy for optimizing the dynamic systems with flexible inequality constraints is proposed. To apply fuzzy inference in solution, the flexible portion in the problem is treated as fuzzy constraints. After functional values are bounded in a region, the objective function of this problem can also be fuzzified easily. When the problem is formulated as a fuzzy dynamic optimization problem, the iterative dynamic programming integrated with fuzzy inference is adopted to find the solution. Two examples are employed, demonstrating the facility of the proposed algorithm. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Fuzzy mathematical programming; Fuzzy constraint satisfaction; Iterative dynamic programming

1. Introduction

Since 1960, the optimization of transient behavior for chemical processes has received significant attention [5,10,15,4]. All these investigations share a common assumption that each portion in the problem, including objective function (goal), system dynamics, and constraints are all definite in general. However, real-world situations are not so rigid. For example, the operator may moderately relax the limitation on the use of resource or cost to exchange the improvement of quality. Such flexibility in operation largely depends on the operator's subjective considerations. In addition, engineers also need to make decisions under

the circumstance with uncertain factors. However, the current dynamic optimization techniques becomes futile as the problem contains flexible portion. The aim of this paper is to find a solution strategy for dynamic systems subjected to flexible inequality constraints.

The fuzzy dynamic optimization problems with single or multiple objectives under deterministic or fuzzy environment has been discussed in literature [14]. In [14], both the dynamic equations and the control policy were discretized, the so-called complete parametrization [4], the problem was then solved by the goal programming technique. Despite making the problem easily port on existent softwares, complete parametrizations not only largely expand dimension for properly approximating highly nonlinear dynamics, but also cause convergence difficulties for the system with multimodal nature.

* Corresponding author. Tel.: +886-2-23636194; fax: +886-2-23623040.

E-mail address: ccl@ccms.ntu.edu.tw (C.-L. Chen).

In this study, the fuzzy set concept, initialized by [18], will be used to grade the degree of acceptability for a flexible constraint. Then, the transformation method, proposed by [16,17] for solving linear static optimization problems with flexible constraints, will be adopted to grade the degree of satisfaction for the objective function. By way of such a fuzzification on the problem, the fuzzy dynamic programming, developed by [1], will be applied for further computation. Obviously, when dynamic programming (DP) is used to solve the dynamic optimization problem (DOP), the drastic expansion in problem's dimension, the well-known *curse of dimensionality*, will become unavoidable. To avoid this difficulty, Luss [10] proposed the use of coarse but accessible state grids and region contracting to reduce dimension expansion, and the iterative computation to promote reliability of the solution. The above modification of DP is called the iterative dynamic programming (IDP).

In the rest of this paper, Section 2 introduces the formulation of the problems and the fuzzification of objective function and flexible constraints. Section 3 briefly reviews fuzzy optimization. Section 4 presents the solution algorithms, the fuzzy dynamic programming and the iterative dynamic programming. Section 5 provides numerical illustrations of the proposed algorithm. Conclusions are finally made in Section 6.

2. Problem formulation

Consider the following dynamic optimization problem:

$$\max_{\mathbf{u}(t) \in \tilde{\Omega}} J[\mathbf{x}(t_f)], \quad (1)$$

where $J[\mathbf{x}(t_f)]$ denotes the objective function to be minimized and $\tilde{\Omega}$ represents the feasible space of control policy, in which all control policies $\mathbf{u}(t)$ satisfy system dynamics and flexible inequality constraints, i.e. $\tilde{\Omega} \equiv \{\mathbf{u}(t) \mid \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \mathbf{x}(0) = \mathbf{x}_0; \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \leq \tilde{\mathbf{b}}; \underline{\mathbf{u}}(t) \leq \mathbf{u}(t) \leq \bar{\mathbf{u}}(t)\}$. Here, $\mathbf{x}(t) \in \mathbb{R}^n$ and $\mathbf{u}(t) \in \mathbb{R}^m$ are state and control vectors, respectively, $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \in \mathbb{R}^n$ and $\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \in \mathbb{R}^K$. Elements in $\tilde{\mathbf{b}} = [\tilde{b}_1, \dots, \tilde{b}_K]^T$ are the flexible boundaries for the constraints.

Eq. (1) seems to be a stochastic or probabilistic programming problem. If so, the flexible portions \tilde{b}_k 's will behave randomly during simulation. In practice,

the sources of random variables depend on the nature and the type of the problem. For example, in the design of aircraft the actual loads acting on the plane depends on the atmospheric conditions, which cannot be predicted precisely in advance [12]. In Eq. (1), however, we assume that the operator does not randomly change the values of \tilde{b}_k 's, but changes on his preference.

2.1. Fuzzification of flexible constraints

As previously stated, the system's behavior is affected by a set of flexible inequality constraints, $g_k(\mathbf{x}(t), \mathbf{u}(t)) \leq \tilde{b}_k$, $k = 1, \dots, K$. Here, the so-called flexible inequality constraint means that when the reasonable limiting value b_k and the acceptable maximal tolerance p_k can be preliminarily defined, all those values that are smaller than $b_k + p_k$ can be regarded as satisfying g_k . Therein, all values less than b_k are thoroughly satisfied. For values in between b_k and $b_k + p_k$, however, the extent satisfying g_k decreases with an increase of its value. To quantitatively demonstrate such a linguistic character, a fuzzy set \mathcal{C}_k with $\mu_{\mathcal{C}_k}(\mathbf{x}(t), \mathbf{u}(t))$ denoting the degree of acceptability for the inequality constraint $g_k(\mathbf{x}(t), \mathbf{u}(t))$, is defined.

$$\mu_{\mathcal{C}_k}(\mathbf{x}(t), \mathbf{u}(t)) = \begin{cases} 1 & \text{if } g_k(\mathbf{x}(t), \mathbf{u}(t)) < b_k, \\ \mathbb{F}_{\mathcal{C}_k}(\mathbf{x}(t), \mathbf{u}(t), b_k, p_k) & \text{if } b_k \leq g_k(\mathbf{x}(t), \mathbf{u}(t)) \leq b_k + p_k, \\ 0 & \text{if } g_k(\mathbf{x}(t), \mathbf{u}(t)) > b_k + p_k, \end{cases} \quad (2)$$

where \mathbb{F} denotes any monotonic decreasing function, and is used to describe the membership value for $\mu_{\mathcal{C}_k}$. The general criteria for defining $\mathbb{F}_{\mathcal{C}_k}$ are somehow subjective. Two types of monotonic decreasing membership functions commonly used are listed in the following (also shown in Figs. 1 and 2):

$$\mathbb{F}_{\mathcal{C}_k}(\mathbf{x}(t), \mathbf{u}(t), b_k, p_k) = \begin{cases} \frac{b_k + p_k - g_k(\mathbf{x}, \mathbf{u})}{p_k} & \text{for linear type,} \\ \frac{1 - \exp[-\alpha(b_k + p_k - g_k(\mathbf{x}, \mathbf{u})) / p_k]}{1 - \exp[-\alpha]} & \text{for exponential type,} \end{cases} \quad (3)$$

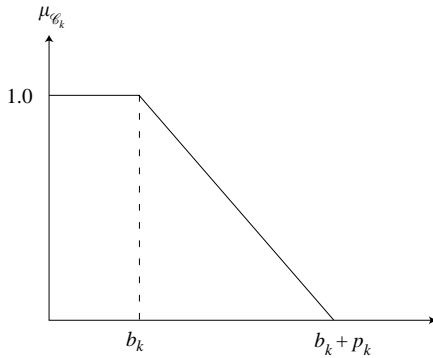


Fig. 1. Linear-type monotonic decreasing membership function for flexible inequality constraints.

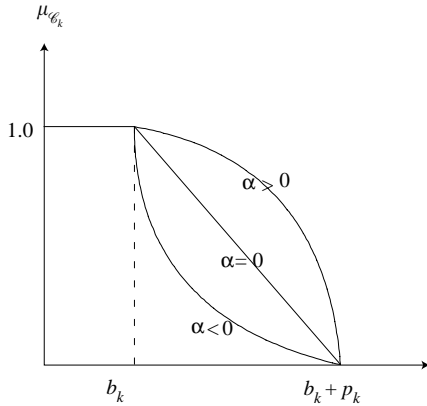


Fig. 2. Exponential-type monotonic decreasing membership function for flexible inequality constraints.

where $\alpha \in \mathbb{R}^1$ is the shape-adjusting factor. For simplicity, we will adopt the linear-type monotonic decreasing membership function \mathcal{C}_k 's. A new fuzzy set can be aggregated as considering all flexible constraints,

$$\mathcal{C} = \mathcal{C}_1 \cap \dots \cap \mathcal{C}_K. \quad (4)$$

The overall degree of acceptability $\mu_{\mathcal{C}}(\mathbf{x}(t), \mathbf{u}(t))$ can be found as

$$\begin{aligned} \mu_{\mathcal{C}}(\mathbf{x}(t), \mathbf{u}(t)) \\ = \mathbb{T}\{\mu_{\mathcal{C}_1}(\mathbf{x}(t), \mathbf{u}(t)), \dots, \mu_{\mathcal{C}_K}(\mathbf{x}(t), \mathbf{u}(t))\}. \end{aligned} \quad (5)$$

Here, $\mu_{\mathcal{C}}(\mathbf{x}(t), \mathbf{u}(t))$ represents the degree of acceptability for the aggregated fuzzy inequality constraints and \mathbb{T} denotes the operation of fuzzy intersection [9].

2.2. Fuzzification of objective function

Eq. (1) can be classified into a parameter sensitivity problem related to the right-hand side bounded on path constraints [12]. For determining the effects of parametric variations, the general way is to solve a series of DOP with each of these parameters changed from large to small values, or vice versa. This procedure, especially in solving a highly nonlinear DOP, is tedious and inefficient. Werners [16,17] has addressed the fuzzy linear programming problems that consists of a crisp objective and fuzzy resources. For solving these problems, Werners [16,17] recommended that the objective should be fuzzified and its resulting optimum with highest satisfaction is bounded by the optimum under the lavish resource and the optimum under the conservative resource. For Eq. (1), we can obtain the similar conclusion by the analysis of *Euler–Lagrange* theory [13]. Therefore, two new subproblems are solved at first: one is restricted by constraints with optimistic boundary $b_k + p_k$, while the other one is confined by the constraints with pessimistic boundary b_k . That is,

$$J^1 = \max_{\mathbf{u}(t) \in \Omega^1} J(\mathbf{x}(t_f)), \quad (6)$$

where $\Omega^1 \equiv \{\mathbf{u}(t) \mid \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \mathbf{x}(0) = \mathbf{x}_0; \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{b} + \mathbf{p}; \underline{\mathbf{u}}(t) \leq \mathbf{u}(t) \leq \bar{\mathbf{u}}(t)\}$ and

$$J^0 = \max_{\mathbf{u}(t) \in \Omega^0} J(\mathbf{x}(t_f)), \quad (7)$$

where $\Omega^0 \equiv \{\mathbf{u}(t) \mid \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \mathbf{x}(0) = \mathbf{x}_0; \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{b}; \underline{\mathbf{u}}(t) \leq \mathbf{u}(t) \leq \bar{\mathbf{u}}(t)\}$. Herein, $\mathbf{b} = [b_1, \dots, b_K]^T$, $\mathbf{p} = [p_1, \dots, p_K]^T$; J^1 and J^0 represent the global optima for the two subproblems, respectively. Once J^1 and J^0 are determined, the optimal value for Eq. (1) will fall in between these two values.

Moreover, owing to various acceptability for g_k changed from b_k to $b_k + p_k$, it will have different satisfaction for J as it changes from J^0 to J^1 . It means that values larger than J^1 have the highest satisfaction and values less than J^0 are unacceptable. For objective values in between J^0 and J^1 , greater J value results in

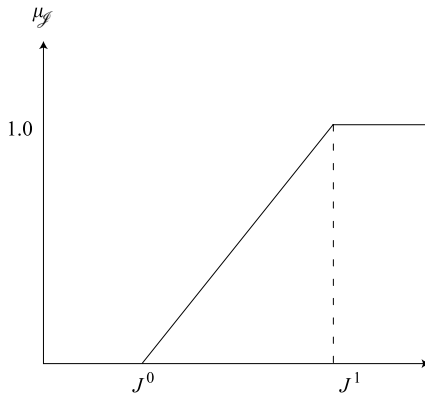


Fig. 3. Linear-type monotonic increasing membership function for the objective.

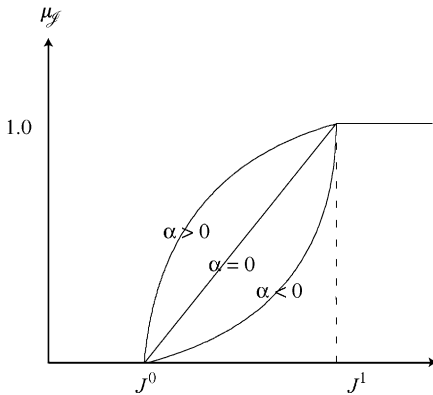


Fig. 4. Exponential-type monotonic increasing membership function for the objective.

increased degree of satisfaction. A fuzzy set \mathcal{J} with degree of satisfaction, $\mu_{\mathcal{J}}[\mathbf{x}(t_f)]$, can thus be defined.

$$\mu_{\mathcal{J}}(\mathbf{x}(t_f)) = \begin{cases} 0 & \text{if } J < J^0, \\ \mathbb{F}_{\mathcal{J}}(J, J^0, J^1) & \text{if } J^0 \leq J \leq J^1, \\ 1 & \text{if } J > J^1. \end{cases} \quad (8)$$

Two frequently used monotonic increasing membership functions, as shown in Figs. 3 and 4 are listed as follows:

$$\mathbb{F}_{\mathcal{J}}(J, J^0, J^1) = \begin{cases} \frac{J - J^0}{J^1 - J^0} & \text{for linear type,} \\ \frac{1 - \exp[-\alpha((J - J^0)/(J^1 - J^0))]}{1 - \exp[-\alpha]} & \text{for exponential type,} \end{cases} \quad (9)$$

where $\alpha \in \mathbb{R}^1$ is the shape-adjusting factor. Similar to the selection for $\mathbb{F}_{\mathcal{C}_k}$, the choice for $\mathbb{F}_{\mathcal{J}}$ is also subjective and dependent on the DM's preference.

3. Fuzzy optimization

As objective and constraints have been fuzzified, the optimization problem in a fuzzy environment can be described as [8]

Attain Fuzzy Objective \mathcal{J}

and Satisfy Fuzzy Constraints \mathcal{C} . (10)

the above statement implies that control policy is viewed as the aggregation of the fuzzy objective \mathcal{J} and the fuzzy inequality constraint \mathcal{C} ,

$$\mathcal{D} = \mathcal{C} \cap \mathcal{J}. \quad (11)$$

The fuzzy decision can be measured by the degree of feasibility, $\mu_{\mathcal{D}}(\mathbf{x}_0|\mathbf{u}(t))$.

$$\begin{aligned} \mu_{\mathcal{D}}(\mathbf{x}_0|\mathbf{u}(t)) &= \mathbb{T}\{\mu_{\mathcal{J}}(\mathbf{x}(t_f)), \mu_{\mathcal{C}}(\mathbf{x}(t), \mathbf{u}(t))\} \\ &= \mathbb{T}\{\mu_{\mathcal{J}}(\mathbf{x}(t_f)), \mu_{\mathcal{C}_1}(\mathbf{x}(t), \mathbf{u}(t)), \dots, \\ &\quad \mu_{\mathcal{C}_K}(\mathbf{x}(t), \mathbf{u}(t))\}, \end{aligned} \quad (12)$$

where $\mu_{\mathcal{D}}(\mathbf{x}_0|\mathbf{u}(t))$ denotes the degree of feasibility for continuous control policy $\mathbf{u}(t)$ starting from initial state \mathbf{x}_0 . For a collection of control actions, the most feasible control policy $\mathbf{u}^*(t)$ is the one having the highest degree of feasibility, $\mu_{\mathcal{D}}^*(\mathbf{x}_0|\mathbf{u}^*(t))$.

$$\begin{aligned} \mu_{\mathcal{D}}^*(\mathbf{x}_0|\mathbf{u}^*(t)) &= \max_{\mathbf{u}(t) \in \Omega} \mu_{\mathcal{D}}(\mathbf{x}_0|\mathbf{u}(t)) \\ &= \max_{\mathbf{u}(t) \in \Omega} \mathbb{T}\{\mu_{\mathcal{J}}(\mathbf{x}(t_f)), \mu_{\mathcal{C}}(\mathbf{x}(t), \mathbf{u}(t))\} \\ &= \max_{\mathbf{u}(t) \in \Omega} \mathbb{T}\{\mu_{\mathcal{J}}(\mathbf{x}(t_f)), \mu_{\mathcal{C}_1}(\mathbf{x}(t), \mathbf{u}(t)), \dots, \\ &\quad \mu_{\mathcal{C}_K}(\mathbf{x}(t), \mathbf{u}(t))\}. \end{aligned} \quad (13)$$

The problem can thus be interpreted as finding an optimal continuous control $\mathbf{u}^*(t)$ over $t \in [t_0, t_f]$ which can simultaneously maintain the degree of satisfaction of objective and the degree of acceptability of fuzzy inequality constraints as much as possible.

4. Solution method

4.1. Fuzzy dynamic programming

The dynamic programming had been applied by Bellman and Zadeh [1] to solve multistage fuzzy optimization problems for a time-invariant discrete system. To extend their algorithm to solve continuous cases, the infinite-dimensional control $\mathbf{u}(t)$ in Eq. (1) can be discretized into a set of finite-dimensional piece-wise constant control actions, such as

$$\max_{\mathbf{u}_0, \dots, \mathbf{u}_{P-1} \in \tilde{\Omega}_d} J(\mathbf{x}_P), \quad (14)$$

where $\tilde{\Omega}_d = \{\mathbf{u}_0, \dots, \mathbf{u}_{P-1} \mid \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}_i), \mathbf{x}(0) = \mathbf{x}_0; \mathbf{g}(\mathbf{x}(t), \mathbf{u}_i) \leq \tilde{\mathbf{b}}; \mathbf{u}(t) \leq \mathbf{u}_i \leq \tilde{\mathbf{u}}(t) \ \forall t \in [t_i, t_{i+1}] \equiv [iT, (i+1)T], i=0, \dots, P-1\}$ represents the feasible space of discretized control policy; $T = t_f/P$ is the control interval in which the control action is kept constant. With the discretized control policy, the degree of feasibility defined in Eq. (12) can be recast as

$$\begin{aligned} \mu_{\mathcal{D}}(\mathbf{x}_0 | \mathbf{u}_0, \dots, \mathbf{u}_{P-1}) \\ = \mathbb{T}\{\mu_{\mathcal{C}}(\mathbf{x}_0, \mathbf{u}_0), \dots, \mu_{\mathcal{C}}(\mathbf{x}_{P-1}, \mathbf{u}_{P-1}), \mu_{\mathcal{J}}(\mathbf{x}_P)\}, \end{aligned} \quad (15)$$

where $\mu_{\mathcal{C}}(\mathbf{x}_i, \mathbf{u}_i) \equiv \mathbb{T}\{\mu_{\mathcal{C}_1}(\mathbf{x}_i, \mathbf{u}_i), \dots, \mu_{\mathcal{C}_K}(\mathbf{x}_i, \mathbf{u}_i)\}$ represents the degree of acceptability for aggregating fuzzy inequality constraints when starting from \mathbf{x}_i and using \mathbf{u}_i as the control action during $t \in [iT, (i+1)T]$, and $\mu_{\mathcal{J}}(\mathbf{x}_P)$ represents the degree of satisfaction for fuzzy objective at the final time end. Now the problem lies in finding a consecutive optimal control series $\mathbf{u}_0^*, \dots, \mathbf{u}_{P-1}^*$ with the highest degree of feasibility, by which, the degree of satisfaction for fuzzy objective and the degree of acceptability for fuzzy constraints can be attained as much as possible, that is,

$$\begin{aligned} \mu_{\mathcal{D}}^*(\mathbf{x}_0 | \mathbf{u}_0^*, \dots, \mathbf{u}_{P-1}^*) \\ = \max_{\mathbf{u}_0, \dots, \mathbf{u}_{P-1} \in \tilde{\Omega}_d} \mu_{\mathcal{D}}(\mathbf{x}_0 | \mathbf{u}_0, \dots, \mathbf{u}_{P-1}) \\ = \max_{\mathbf{u}_0, \dots, \mathbf{u}_{P-1} \in \tilde{\Omega}_d} \mathbb{T}\{\mu_{\mathcal{C}}(\mathbf{x}_0, \mathbf{u}_0), \dots, \\ \mu_{\mathcal{C}}(\mathbf{x}_{P-1}, \mathbf{u}_{P-1}), \mu_{\mathcal{J}}(\mathbf{x}_P)\}. \end{aligned} \quad (16)$$

The dynamic programming can be easily applied to solve Eq. (16). Notably, the last two right-hand side terms depend only on the current state condition \mathbf{x}_{P-1} and the applied control action \mathbf{u}_{P-1} . The fuzzy decision at the last stage can thus be characterized by

$$\mu_{\mathcal{D}}(\mathbf{x}_{P-1} | \mathbf{u}_{P-1}) = \mathbb{T}\{\mu_{\mathcal{C}}(\mathbf{x}_{P-1}, \mathbf{u}_{P-1}), \mu_{\mathcal{J}}(\mathbf{x}_P)\}. \quad (17)$$

During the maximization over the sequence of control policies $\mathbf{u}_0, \dots, \mathbf{u}_{P-1}$ in Eq. (16), these control actions can be divided into two parts:

1. the maximization over the interval not directly related to the last stage: $\mathbf{u}_0, \dots, \mathbf{u}_{P-2}$, and
2. the maximization over \mathbf{u}_{P-1} .

Eq. (16) can thus be rewritten as

$$\begin{aligned} \mu_{\mathcal{D}}^*(\mathbf{x}_0 | \mathbf{u}_0^*, \dots, \mathbf{u}_{P-1}^*) \\ = \max_{\mathbf{u}_0, \dots, \mathbf{u}_{P-2} \in \tilde{\Omega}_d} \mathbb{T}\left\{\mu_{\mathcal{C}}(\mathbf{x}_0, \mathbf{u}_0), \dots, \mu_{\mathcal{C}}(\mathbf{x}_{P-2}, \mathbf{u}_{P-2}), \right. \\ \left. \max_{\mathbf{u}_{P-1} \in \tilde{\Omega}_d} \mathbb{T}[\mu_{\mathcal{C}}(\mathbf{x}_{P-1}, \mathbf{u}_{P-1}), \mu_{\mathcal{J}}(\mathbf{x}_P)]\right\} \\ = \max_{\mathbf{u}_0, \dots, \mathbf{u}_{P-2} \in \tilde{\Omega}_d} \mathbb{T}\{\mu_{\mathcal{C}}(\mathbf{x}_0, \mathbf{u}_0), \dots, \mu_{\mathcal{C}}(\mathbf{x}_{P-2}, \mathbf{u}_{P-2}), \\ \mu_{\mathcal{D}}^*(\mathbf{x}_{P-1} | \mathbf{u}_{P-1}^*)\}, \end{aligned} \quad (18)$$

where $\mu_{\mathcal{D}}^*(\mathbf{x}_{P-1} | \mathbf{u}_{P-1}^*)$ denotes the maximal feasibility at the last time stage caused by the best control policy \mathbf{u}_{P-1}^* . Furthermore, owing to the next decision $\mu_{\mathcal{D}_{P-2}}(\mathbf{x}_{P-2} | \mathbf{u}_{P-2})$ depends only on the current state \mathbf{x}_{P-2} and the applied control action \mathbf{u}_{P-2} , Eq. (18) can be expressed as

$$\begin{aligned} \mu_{\mathcal{D}}^*(\mathbf{x}_0 | \mathbf{u}_0^*, \dots, \mathbf{u}_{P-1}^*) \\ = \max_{\mathbf{u}_0, \dots, \mathbf{u}_{P-3} \in \tilde{\Omega}_d} \mathbb{T}\{\mu_{\mathcal{C}}(\mathbf{x}_0, \mathbf{u}_0), \dots, \mu_{\mathcal{C}}(\mathbf{x}_{P-3}, \mathbf{u}_{P-3}), \\ \max_{\mathbf{u}_{P-2} \in \tilde{\Omega}_d} \mathbb{T}[\mu_{\mathcal{C}}(\mathbf{x}_{P-2}, \mathbf{u}_{P-2}), \mu_{\mathcal{D}}^*(\mathbf{x}_{P-1} | \mathbf{u}_{P-1}^*)]\} \\ = \max_{\mathbf{u}_0, \dots, \mathbf{u}_{P-3} \in \tilde{\Omega}_d} \mathbb{T}\{\mu_{\mathcal{C}}(\mathbf{x}_0, \mathbf{u}_0), \dots, \mu_{\mathcal{C}}(\mathbf{x}_{P-3}, \mathbf{u}_{P-3}), \\ \mu_{\mathcal{D}}^*(\mathbf{x}_{P-2} | \mathbf{u}_{P-2}^*)\}. \end{aligned} \quad (19)$$

Continuing the above iteration in a backward manner, the optimal sequence of control policies \mathbf{u}_{P-3}^* , $\mathbf{u}_{P-4}^*, \dots, \mathbf{u}_0^*$ can be determined by the following recurrent relations:

$$\mu_{\mathcal{D}}^*(\mathbf{x}_i | \mathbf{u}_i^*) = \max_{\mathbf{u}_i \in \hat{\mathcal{Q}}_d} \mathbb{T} \{ \mu_{\mathcal{E}}(\mathbf{x}_i, \mathbf{u}_i), \mu_{\mathcal{D}}^*(\mathbf{x}_{i+1} | \mathbf{u}_{i+1}^*) \},$$

$$i = P - 1, \dots, 0, \quad (20)$$

where $\mu_{\mathcal{D}}^*(\mathbf{x}_P | \mathbf{u}_P^*) \equiv \mu_{\mathcal{F}}^*(\mathbf{x}_P)$.

4.2. Fuzzy iterative dynamic programming

Direct application of dynamic programming to determine the optimal policy for a nonlinear dynamic system is extremely difficult. Two major obstacles discourage the direct use of dynamic programming [10]. The greatest one, commonly referred to as *the curse of dimensionality*, involves the use of a large amount of grid values for the state and the control, respectively, to have reasonable result. Therefore, numerous integrations must be performed at each time stage. Another problem arises when the trajectory calculated for a particular grid point does not meet the grid point at the next time step. Although interpolation can be used, the approximation may still be unreliable. To tackle these drawbacks, Luss [10] proposed the method of iterative dynamic programming (IDP). By using of coarse grid points and region-reduction strategy, IDP not only successfully overcomes the curse and promotes the efficiency of computation but greatly increases numerical accuracy. Since then, extensive researches [6,7,2,3,11] have shown that IDP is one reliable method on obtaining global solution for nonlinear dynamic optimization problems. So, we attempt in this study to integrate the IDP and the fuzzy inference to provide an efficient way on finding the solution of a dynamic optimization problem with flexible inequality constraints. All these works are summarized as follows:

(I) Preliminary preparations

1. Select suitable monotonic decreasing/increasing functions for $\mu_{\mathcal{E}_k}$'s and $\mu_{\mathcal{F}}$, respectively. The linear-type functions will be used in this study.

2. Use b_k and $b_k + p_k$ to define $\mu_{\mathcal{E}_k}$'s,

$$\mu_{\mathcal{E}_k}(\mathbf{x}, \mathbf{u}) = \begin{cases} 1 & \text{if } g_k(\mathbf{x}, \mathbf{u}) < b_k, \\ \frac{b_k + p_k - g_k(\mathbf{x}, \mathbf{u})}{p_k} & \text{if } b_k \leq g_k(\mathbf{x}, \mathbf{u}) \leq b_k + p_k, \\ 0 & \text{if } g_k(\mathbf{x}, \mathbf{u}) > b_k + p_k. \end{cases} \quad (21)$$

3. Solve Eqs. (7) and (6) by any existing optimization method to determine J^0 and J^1 .
4. Use J^0 and J^1 to define $\mu_{\mathcal{J}}$.

$$\mu_{\mathcal{J}}(\mathbf{x}_P) = \begin{cases} 0 & \text{if } J < J^0, \\ \frac{J - J^0}{J^1 - J^0} & \text{if } J^0 \leq J \leq J^1, \\ 1 & \text{if } J > J^1. \end{cases} \quad (22)$$

5. Select operating parameters:

- (a) Set the number of time stages, P . Previous experience shows that a setting of $P = 10$ is enough in general. For problems with high nonlinearities, however, it is necessary to take P as 20–25 to obtain higher numerical accuracy.
- (b) Assign the number of grid points N at each time stage, and the number of control trials M for testing the control performance when starting from a specific grid point. In the following, we use an odd M value and the M control trials are equally spaced in between \underline{u}_j and \bar{u}_j .
- (c) Choose a searching region r for u_j 's, and a contracting factor η . Then η value can be set as 0.8–0.9 in most cases.

(II) Generation of grid points

By using N values of evenly distributed controls inside the allowable region, integrate the system dynamic equations N times to generate the state grids at each time stage, i.e., $\mathbf{x}(t_i, n)$, $i = 1, \dots, P - 1$, $n = 1, \dots, N$, $t_i = iT$, as shown in Fig. 5.

(III) The FIDP

1. Begin at the last stage (i.e., stage P), corresponding to $t \in [t_{P-1}, t_P] = [t_f - T, t_f]$, as

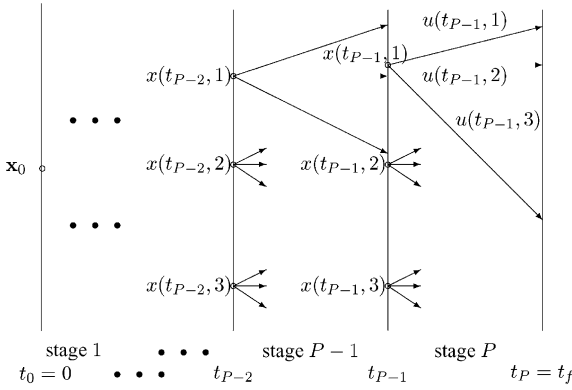


Fig. 5. Illustration of FIDP method ($N=3$, $M=3$).

shown in Fig. 5. Start from $\mathbf{x}_{P-1} = \mathbf{x}(t_{P-1}, n=1)$:

- Integrate the system dynamic equations over $t \in [t_{P-1}, t_P]$ with testing control input $\mathbf{u}_{P-1} = \mathbf{u}(t_{P-1}, m=1)$. Notably, the M testing inputs are equally spaced in the allowable regions centered at the temporal best value.
- Calculate the acceptability, $\mu_{\mathcal{G}_k}(\mathbf{x}_{P-1}, \mathbf{u}_{P-1})$, $k=1, \dots, K$ and the resultant degree of satisfaction, $\mu_{\mathcal{J}}(\mathbf{x}_P)$.
- Aggregate $\mu_{\mathcal{G}_k}(\mathbf{x}_{P-1}, \mathbf{u}_{P-1})$'s and $\mu_{\mathcal{J}}(\mathbf{x}_P)$ to determine the feasibility, $\mu_{\mathcal{D}}(\mathbf{x}_{P-1} | \mathbf{u}_{P-1})$.

$$\begin{aligned} \mu_{\mathcal{D}}(\mathbf{x}_{P-1} | \mathbf{u}_{P-1}) \\ = \mathbb{T}\{\mu_{\mathcal{G}_1}(\mathbf{x}_{P-1}, \mathbf{u}_{P-1}), \dots, \\ \mu_{\mathcal{G}_K}(\mathbf{x}_{P-1}, \mathbf{u}_{P-1}), \mu_{\mathcal{J}}(\mathbf{x}_P)\}. \end{aligned} \quad (23)$$

Here, we use the \min operator for \mathbb{T} . Moreover, numerical experiments show that the minimum acceptability of inequality constraints may occur within the stage. Therefore, extra checkpoints within the stage should be included to detect the point with minimum acceptability.

- Repeat previous steps with $\mathbf{u}_{P-1} = \mathbf{u}(t_{P-1}, m)$, $m=2, \dots, M$.
 - Select the control with the maximized feasibility, $\mu_{\mathcal{D}}^*(\mathbf{x}_{P-1} | \mathbf{u}_{P-1}^*)$.
- Repeat **step 1** starting from other grids of the last stage, i.e., $\mathbf{x}_{P-1} = \mathbf{x}(t_{P-1}, n)$, $n=2, \dots, N$.

- Step back to stage $P-1$, corresponding to $t \in [t_{P-2}, t_{P-1}] = [t_f - 2T, t_f - T]$. Start from $\mathbf{x}_{P-2} = \mathbf{x}(t_{P-2}, n=1)$:

- Integrate the system dynamic equations from time t_{P-2} to t_{P-1} with testing control action $\mathbf{u}_{P-2} = \mathbf{u}(t_{P-2}, m=1)$. For integration over $t \in [t_{P-1}, t_P]$, use the closest grid's temporal best control action determined previously.
- Calculate the acceptability, $\mu_{\mathcal{G}_k}(\mathbf{x}_{P-2}, \mathbf{u}_{P-2})$, $k=1, \dots, K$, and the resulting satisfaction at t_P . Notably, the acceptability, $\mu_{\mathcal{G}_k}(\mathbf{x}_{P-1}, \mathbf{u}_{P-1})$, should be calculated as well since the state trajectory may not exactly meet the state grid at t_{P-1} .
- Aggregate the acceptability and the resulting satisfaction of objective to determine the feasibility for each control value from t_{P-2} to t_P .

$$\begin{aligned} \mu_{\mathcal{D}}(\mathbf{x}_{P-2} | \mathbf{u}_{P-2}) \\ = \mathbb{T}\{\mu_{\mathcal{G}_1}(\mathbf{x}_{P-2}, \mathbf{u}_{P-2}), \dots, \\ \mu_{\mathcal{G}_K}(\mathbf{x}_{P-2}, \mathbf{u}_{P-2}), \mu_{\mathcal{D}}^*(\mathbf{x}_{P-1} | \mathbf{u}_{P-1}^*)\}. \end{aligned} \quad (24)$$

- Repeat previous steps with $\mathbf{u}_{P-2} = \mathbf{u}(t_{P-2}, m)$, $m=2, \dots, M$.
 - Select the control action with maximal feasibility, $\mu_{\mathcal{D}}^*(\mathbf{x}_{P-2} | \mathbf{u}_{P-2}^*)$.
- Repeat **step 3** from other grids of the same stage, i.e., $\mathbf{x}_{P-2} = \mathbf{x}(t_{P-2}, n)$, $n=2, \dots, N$.
 - Repeat **steps 3** and **4** until the first stage.
 - Reduce the region for allowable control actions.

$$r^{(j+1)} = \eta r^{(j)}, \quad (25)$$

where j is the iteration index. Use the optimal state trajectory as the mid-point for state grid at each time stage, and use the optimal controls from previous steps as the central values for the allowable controls.

- Increase the iteration index j by 1 and go back to **step 1**. Continue the iteration for a specific number of iterations such as 20, i.e., one pass, and then examine the results.

5. Numerical illustration

Two numerical examples are used to demonstrate the proposed algorithm, namely: (a) a plug-flow tubular reactor problem, in which the temperature state is restricted by a flexible constraint; and (b) a nonlinear mathematical problem, in which a flexible nonlinear inequality constraints should be satisfied. All of the computations were performed in double precision on Pentium-100 personal computer using DJGPP C++ compiler.

Example 1. A plug-flow tubular reactor problem

Consider a plug-flow tubular reactor adapted from [11].

$$\begin{aligned}\frac{dx_1}{dt} &= (1 - x_1)k_1 - x_1k_2, \\ \frac{dx_2}{dt} &= 300[(1 - x_1)k_1 - x_1k_2] - u(x_2 - 290),\end{aligned}\quad (26)$$

where

$$\begin{aligned}k_1 &= 1.7536 \times 10^5 \exp\left(\frac{-1.1374 \times 10^4}{1.9872x_2}\right), \\ k_2 &= 2.4885 \times 10^{10} \exp\left(\frac{-2.2748 \times 10^4}{1.9872x_2}\right), \\ x_1(0) &= 0, \quad x_1(0) = 380,\end{aligned}\quad (27)$$

where x_1 and x_2 denote the normalized exit concentration of the desired product and the reactor temperature, respectively. The control variable is the normalized coolant flow rate $u(t)$ which is bounded by $0 \leq u(t) \leq 0.5$. The temperature is restricted by a flexible constraint, $x_2(t) \leq 450$. The objective is to maximize the yield given by $x_1(t_f = 5 \text{ min})$, suppose the limiting value and the maximal tolerance for the temperature are set as 450 and 20 K, respectively, and the linear-type function is used for membership. The change of degree of acceptability for temperature can be determined as

$$\mu_c(x_2(t)) = \begin{cases} 1 & \text{if } x_2(t) < 450, \\ \frac{470 - x_2(t)}{470 - 450} & \text{if } 450 \leq x_2(t) \leq 470, \\ 0 & \text{if } x_2(t) > 470. \end{cases}$$

To determine the reasonable objective region, two subproblems are defined, one has the pessimistic restriction,

$$x_2(t) \leq 450 \quad (28)$$

and the other one possesses the optimistic constraint,

$$x_2(t) \leq 470. \quad (29)$$

The optimal values determined by the IDP for the two subproblems are $J^0 = 0.6679$ and $J^1 = 0.6795$, respectively. We employ the linear-type membership function to depict the change of degree of feasibility for the fuzzy objective,

$$\begin{aligned}\mu_{\mathcal{J}}(x_1(t_f)) &= \begin{cases} 0 & \text{if } J(x_1(t_f)) < 0.6679, \\ \frac{J(x_1(t_f)) - 0.6679}{0.6795 - 0.6679} & \text{if } 0.6679 \leq J(x_1(t_f)) \leq 0.6795, \\ 1 & \text{if } J(x_1(t_f)) > 0.6795. \end{cases} \\ &\quad (30)\end{aligned}$$

Now select number of grid points $N = 5$ and number of trial controls $M = 3$. To show computational details, we solve the problem with three stages, i.e., $P = 3$, at first. A more compact solution will be supplied later. Suppose we have obtained temporal optimal control inputs for the three stages, respectively, such as $u_0^* = 0$, $u_1^* = 0.3609$, and $u_2^* = 0.1932$. The contracting factor is $\eta = 0.85$, and the current searching region is $r = 0.0681$. Five control inputs evenly distributed around the temporal optimal control inputs are used for generating five state grids at t_1 and t_2 , which are $u(t_0, n) \in \{0.0681, 0.03405, 0, 0, 0\}$ and $u(t_1, n) \in \{0.4290, 0.3950, 0.3609, 0.32685, 0.2928\}$. Therein, five control series are used sequentially, such as $\{u(t_0, n), u(t_1, n)\}$, $n = 1, \dots, 5$. The states thus generated, $\mathbf{x}(t_1, n)$ and $\mathbf{x}(t_2, n)$, are shown in Table 1. Notably, there are only three state grids at t_1 , since only three numerical values for $u(t_0, n)$ are available.

Starting from $\mathbf{x}(t_2, n = 1) = [0.5434, 461.08]^T$, integrating the system by using three testing control policies in sequence, $u(t_2, m) \in \{0.1251, 0.1932, 0.2613\}$, results in three final states, where $u(t_2, 1) = 0.2613$ gives the maximal feasibility $\mu_{\mathcal{J}}^* = 0.0605$. Similarly,

Table 1
Numerical illustration for the FIDP in Example 1 ($N = 5$, $M = 3$)

| $t_0 = 0$ | t_1 | | | t_2 | | | $t_3 = t_f$ | | | | | | | | | | | | | | | | | |
|-------------------|--------|-------------|----------------------|--------|---------------------|------------------------------|-------------|---------|----------------------|----------------------|---------------------|------------------------------|----------------------|---------|----------------------|------------------------------|---------------------|---------------------|----------------------|--------|--------|---------------------|---------------------|---------------------|
| | | | | | | | | | | $\mathbf{x}(t_2, n)$ | | | $\mathbf{x}(t_3, n)$ | | | | | | | | | | | |
| | | | | | | | | | | x_1 | x_2 | $\mu_{\mathcal{G}}$ | $u(t_2, m)$ | x_1 | x_2 | $\mu_{\mathcal{G}}$ | $\mu_{\mathcal{F}}$ | $\mu_{\mathcal{D}}$ | | | | | | |
| — | | | | | | | | | | 0.5434 | 461.08 | 0.4459 | 0.1251 | 0.6345 | 453.17 | 0.3909 | 0.0000 | 0 | | | | | | |
| | | | | | | | | | | | | | 0.1932 | 0.6549 | 442.11 | 0.4459 | 0.0000 | 0 | | | | | | |
| | | | | | | | | | | | | | 0.2613* | 0.6650 | 429.56 | 0.4459 | 0.0605 | 0.0605* | | | | | | |
| | | | | | | | | | | | | | 0.1251 | 0.6459 | 450.49 | 0.6198 | 0.0000 | 0 | | | | | | |
| | | | | | | | | | | 0.5461 | 454.99 | 0.7504 | 0.1932 | 0.6632 | 438.99 | 0.7471 | 0.0000 | 0 | | | | | | |
| | | | | | | | | | | | | | 0.2613* | 0.6694 | 425.88 | 0.7504 | 0.9548 | 0.7504* | | | | | | |
| | | | | | | | | | | | | | 0.1251 | 0.6538 | 447.90 | 0.8785 | 0.0000 | 0 | | | | | | |
| | | | | | | | | | | 0.5388 | 446.89 | 1.0000 | 0.1932 | 0.6662 | 435.63 | 1.0000 | 0.3136 | 0.3136 | | | | | | |
| | | | | | | | | | | | | | 0.2613* | 0.6666 | 421.57 | 1.0000 | 0.3851 | 0.3851* | | | | | | |
| | | | | | | | | | | | | | 0.1251* | 0.4193 | 399.54 | 1.0000 | 0.0000 | 0* | | | | | | |
| | | | | | | | | | | 0.3251 | 393.49 | 1.0000 | 0.1932 | 0.3996 | 383.85 | 1.0000 | 0.0000 | 0 | | | | | | |
| | | | | | | | | | | | | | 0.2613 | 0.3859 | 371.23 | 1.0000 | 0.0000 | 0 | | | | | | |
| | | | | | | | | | | | | | 0.1251* | 0.2443 | 358.50 | 1.0000 | 0.0000 | 0* | | | | | | |
| | | | | | | | | | | 0.2144 | 364.43 | 1.0000 | 0.1932 | 0.2397 | 350.32 | 1.0000 | 0.0000 | 0 | | | | | | |
| | | | | | | | | | | | | | 0.2613 | 0.2362 | 343.26 | 1.0000 | 0.0000 | 0 | | | | | | |
| | | | | | | | | | | $\mathbf{x}(t_1, n)$ | | | $\mathbf{x}(t_2, n)$ | | | Nearest $\mathbf{x}(t_2, n)$ | | | $\mathbf{x}(t_3, n)$ | | | | | |
| | | | | | | | | | | x_1 | x_2 | $\mu_{\mathcal{G}}$ | $u(t_1, m)$ | x_1 | x_2 | $\mu_{\mathcal{G}}$ | x_1 | x_2 | u_2^* | x_1 | x_2 | $\mu_{\mathcal{G}}$ | $\mu_{\mathcal{F}}$ | $\mu_{\mathcal{D}}$ |
| | | | | | | | | | | 0.1775 | 433.24 | 1.0000 | 0.2928 | 0.5434 | 461.08 | 0.0723 | 0.5434 | 461.08 | 0.2613 | 0.6650 | 429.56 | 0.0723 | 0.0605 | 0.0605 |
| 0.3609* | 0.5388 | 446.89 | 0.8191 | 0.5388 | 446.89 | 0.2613* | 0.6666 | 421.57 | 0.8191 | | | | 0.3851 | 0.3851* | | | | | | | | | | |
| 0.4290 | 0.4887 | 423.20 | 1.0000 | 0.5388 | 446.89 | 0.2613 | 0.6000 | 402.40 | 1.0000 | | | | 0.0000 | 0 | | | | | | | | | | |
| 0.1438 | 417.19 | 1.0000 | 0.2928* | 0.4241 | 434.57 | 1.0000 | 0.5388 | 446.89 | 0.2613* | 0.6085 | 427.06 | 1.0000 | 0.0000 | 0* | | | | | | | | | | |
| | | | 0.3609 | 0.3511 | 404.80 | 1.0000 | 0.3251 | 393.49 | 0.1251 | 0.4972 | 422.90 | 1.0000 | 0.0000 | 0 | | | | | | | | | | |
| | | | 0.4290 | 0.3045 | 384.02 | 1.0000 | 0.3251 | 393.49 | 0.1251 | 0.3679 | 383.46 | 1.0000 | 0.0000 | 0 | | | | | | | | | | |
| 0.1200 | 404.61 | 1.0000 | 0.2928* | 0.2666 | 394.40 | 1.0000 | 0.3251 | 393.49 | 0.1251* | 0.3816 | 406.01 | 1.0000 | 0.0000 | 0* | | | | | | | | | | |
| | | | 0.3609 | 0.2351 | 377.32 | 1.0000 | 0.2144 | 364.43 | 0.1251 | 0.2883 | 375.24 | 1.0000 | 0.0000 | 0 | | | | | | | | | | |
| | | | 0.4290 | 0.2144 | 364.43 | 1.0000 | 0.2144 | 364.43 | 0.1251 | 0.2443 | 358.50 | 1.0000 | 0.0000 | 0 | | | | | | | | | | |
| $\mathbf{x}(t_0)$ | | | $\mathbf{x}(t_1, n)$ | | | Nearest $\mathbf{x}(t_1, n)$ | | | $\mathbf{x}(t_2, n)$ | | | Nearest $\mathbf{x}(t_2, n)$ | | | $\mathbf{x}(t_3, n)$ | | | | | | | | | |
| x_1 | x_2 | $u(t_0, m)$ | x_1 | x_2 | $\mu_{\mathcal{G}}$ | x_1 | x_2 | u_1^* | x_1 | x_2 | $\mu_{\mathcal{G}}$ | x_1 | x_2 | u_2^* | x_1 | x_2 | $\mu_{\mathcal{G}}$ | $\mu_{\mathcal{F}}$ | $\mu_{\mathcal{D}}$ | | | | | |
| 0.00 | 380 | 0.0000* | 0.1775 | 433.24 | 1.0000 | 0.1775 | 433.24 | 0.3609* | 0.5338 | 446.89 | 0.8191 | 0.5388 | 446.89 | 0.2613* | 0.6666 | 421.57 | 0.8191 | 0.3851 | 0.3851* | | | | | |
| | | 0.0681 | 0.1200 | 404.61 | 1.0000 | 0.1200 | 404.61 | 0.2928 | 0.2666 | 394.40 | 1.0000 | 0.3251 | 393.49 | 0.1251 | 0.3816 | 406.01 | 1.0000 | 0.0000 | 0 | | | | | |

we can determine the optimal control inputs for other four state grids at t_2 , such as shown in Table 1.

Stepping back to stage $t_{P-2} = t_1$, we use $u(t_1, m) \in \{0.2928, 0.3609, 0.4290\}$ sequentially as testing control inputs. As shown in Table 1, when starting from $\mathbf{x}(t_1, 1) = [0.1775, 433.24]^T$ and using $u(t_1, 1) = 0.2928$ and $u(t_1, 2) = 0.3609$, the trajectories will exactly reach $[0.5434, 461.08]^T$ and $[0.5388, 446.89]^T$, respectively, at t_2 . Therefore, $u_2^* = 0.2613$ will be used as the subsequent control input and that will attain final states $[0.6650, 429.56]^T$ and $[0.6666, 421.57]^T$, and the resultant feasibilities are $\mu_{\mathcal{Q}} = 0.06050$ and $\mu_{\mathcal{Q}} = 0.3851$, respectively. However, when $u(t_1, 3) = 0.4290$ is used, the trajectory started from the same state grid will be $[0.4887, 423.20]^T$ at t_2 and the closest state grid is $[0.5388, 446.89]^T$. Therefore, the subsequent control input for further integration is also $u_2^* = 0.2613$. Notably, the resultant final state is $[0.6000, 402.40]^T$ with a feasibility of $\mu_{\mathcal{Q}} = 0$. So, when starting from state grid of $\mathbf{x}(t_1, 1) = [0.1775, 433.24]^T$, the control series with the highest feasibility, $\mu_{\mathcal{Q}}^* = 0.3851$, are $u_1^* = 0.3609$ and $u_2^* = 0.2613$. The optimal control series can be determined similarly for other two state grids $\mathbf{x}(t_1, n)$, $n = 2, 3$, such as shown in the table.

Stepping back to $t_{P-3} = t_0$ and using the similar procedure, we can find the optimal control series, $u_0^* = 0.0$, $u_1^* = 0.3609$, and $u_2^* = 0.2613$, with the highest feasibility, $\mu_{\mathcal{Q}}^* = 0.3851$. For such control series, the objective value is 0.6666. The user can repeat the above computational procedures, where the search region for control will be contracted by $\eta = 0.85$.

In order to obtain solution with higher numerical accuracy, the time horizon is divided into $P = 10$ stages. The results after several passes of iterations are shown in Figs. 6–8 for trajectories of concentration, temperature, and the controlled cooling rate, respectively. The final optimal value is 0.6747.

Example 2. A mathematical system with nonlinear inequality constraints

The next example is a modification from [11]. The mathematical system is governed by three differential equations:

$$\frac{dx_1}{dt} = x_2,$$

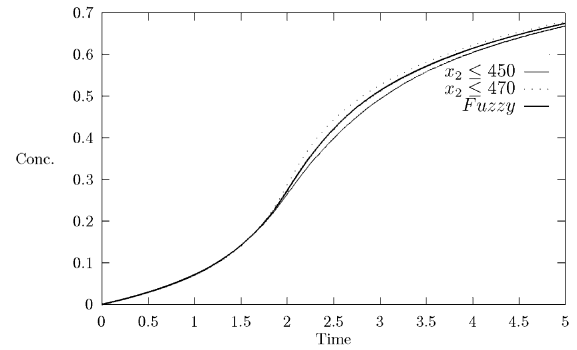


Fig. 6. Concentration trajectory in Example 1.

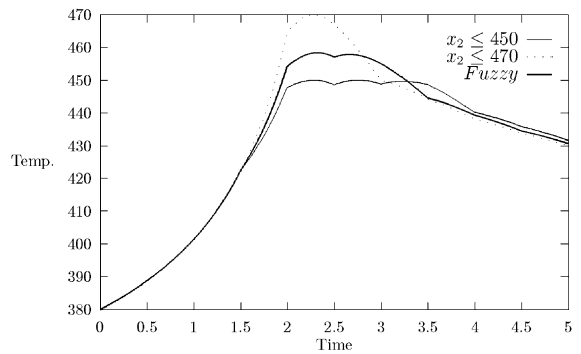


Fig. 7. Temperature trajectory in Example 1.

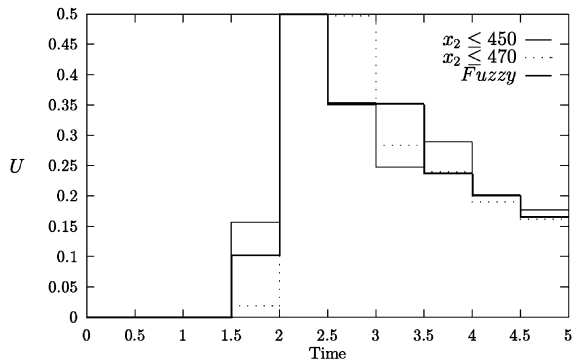


Fig. 8. Control trajectory in Example 1.

$$\frac{dx_2}{dt} = -x_2 + u,$$

$$\frac{dx_3}{dt} = x_1^2 + x_2^2 + 0.005u^2,$$

$$\begin{aligned} x_1(0) = x_3(0) = 0, \quad x_2(0) = -1, \\ -20 \leq u(t) \leq 20. \end{aligned} \quad (31)$$

The objective function to be minimized is $x_3(t_f = 1)$, and the flexible constraint is

$$g(\mathbf{x}) = x_2 + 0.5 - 8(t - 0.5)^2 \leq \tilde{0}. \quad (32)$$

Suppose the limiting value and maximal tolerance for the constraint are 0 and 0.15, respectively. As the linear-type function is used, the change of degree of acceptability can be depicted as follows:

$$\mu_g(\mathbf{x}(t)) = \begin{cases} 1 & \text{if } g(\mathbf{x}) < 0, \\ \frac{0.15 - g(\mathbf{x})}{0.15 - 0} & \text{if } 0.0 \leq g(\mathbf{x}) \leq 0.15, \\ 0 & \text{if } g(\mathbf{x}) > 0.15. \end{cases} \quad (33)$$

The optimal values for the two subproblems are $J^1 = 0.1239$ for optimistic constraints and $J^0 = 0.1818$ for pessimistic constraints. Based on these two values, the linear-type function is applied to characterize the degree of satisfaction.

$$\mu_J(x_3(t_f)) = \begin{cases} 1 & \text{if } J < 0.1239, \\ \frac{0.1818 - J}{0.1818 - 0.1239} & \text{if } 0.1239 \leq J \leq 0.1818, \\ 0 & \text{if } J > 0.1818. \end{cases} \quad (34)$$

Here, we use Zadeh-min to perform the fuzzy intersection. The resulting optimal value is 0.1505. The trajectories of $x_3(t)$ and the control profile are shown in Figs. 9 and 10, respectively.

6. Conclusions

Based on the iterative dynamic programming, a solution strategy for dynamic optimization problems with flexible inequality constraints has been proposed. The problem can be formulated as a fuzzy dynamic optimization problem after its flexible portion is quantified by the fuzzy logic. Compared with the general

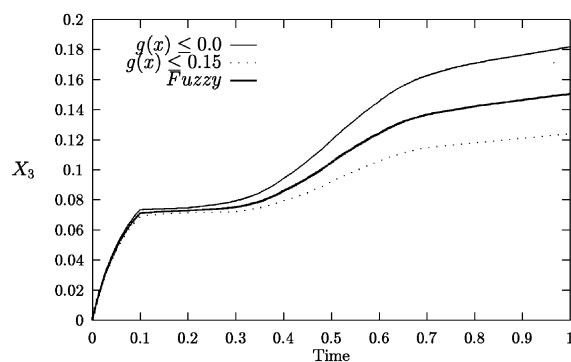


Fig. 9. Trajectory of $x_3(t)$ in Example 2.

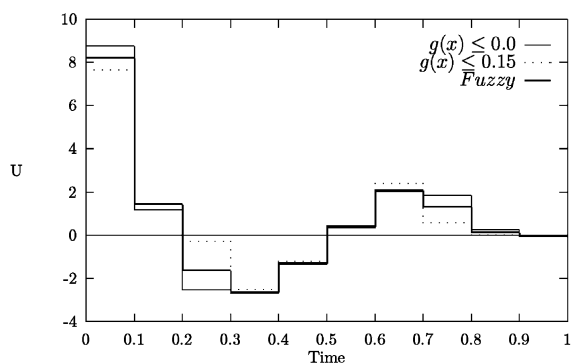


Fig. 10. Trajectory of control action in Example 2.

dynamic optimization problems, the additional requisite of Eq. (14) in computation is to determine the acceptability in every time stage and the satisfaction of objective's value at final time end. This additional computation can be easily fulfilled as IDP is used. Two examples are supplied, demonstrating the feasibility of proposed strategy.

References

- [1] R.E. Bellman, L.A. Zadeh, Decision making in a fuzzy environment, *Manage. Sci.* 17 (1970) 141–164.
- [2] B. Bojkor, R. Luus, Time optimal control of high dimensional systems by iterative dynamic programming, *Can. J. Chem. Eng.* 73 (1995) 380–390.
- [3] B. Bojkor, R. Luus, Optimal control of nonlinear systems with unspecific final times, *Chem. Eng. Sci.* 51 (1996) 905–919.

- [4] E.F. Carrasco, J.R. Banga, Dynamic optimization of batch reactors using adaptive stochastic algorithms, *Ind. Eng. Chem. Res.* 36 (1997) 2252–2261.
- [5] J.J. Cuthrell, L.T. Biegler, On the optimization of differential-algebraic process systems, *A.I.Ch.E.J.* 8 (1987) 1257–1270.
- [6] S.A. Dadebo, K.B. McAuley, Dynamic optimization of constrained chemical engineering problems using dynamic programming, *Comput. Chem. Eng.* 19 (1995) 513–525.
- [7] S.A. Dadebo, K.B. McAuley, Iterative dynamic programming for minimum energy control problems with time delay, *Opt. Control Appl. Meth.* 16 (1995) 217–227.
- [8] J. Kacprzyk, *Multistage Fuzzy Control*, Wiley, New York, 1997.
- [9] G. Klir, B. Yuan, *Fuzzy Set and Fuzzy Logic*, Prentice-Hall, Engelwood Cliffs, NJ, 1995.
- [10] R. Luus, Optimal control by dynamic programming using grids points and region reduction, *Hungarian J. Ind. Chem.* 17 (1989) 523–543.
- [11] W. Mekarapiruk, R. Luus, Optimal control of inequality state constrained systems, *Ind. Eng. Chem. Res.* 36 (1997) 1686–1694.
- [12] S.S. Rao, *Engineering Optimization — Theory and Practice*, Wiley, Inc., New York, 1996.
- [13] W.H. Ray, J. Szekeley, *Process Optimization*, Wiley, Inc., New York, 1973.
- [14] V.N. Sastry, R.N. Tiwari, K.S. Sastri, Dynamic programming approach to multiple objective control problems having deterministic or fuzzy goals, *Fuzzy Sets and Systems* 57 (1993) 195–202.
- [15] V.S. Vassiladis, R.W.H. Sargent, C.C. Pantelides, Solution of a class of multistage dynamic optimization problems. 2, problems with path constraints, *Ind. Eng. Chem. Res.* 33 (1994) 2123–2133.
- [16] B. Werners, Interactive multiple objective programming subjective to flexible constraints, *Eur. J. Oper. Res.* 31 (1987) 342–349.
- [17] B. Werners, An interactive programming system, *Fuzzy Sets and Systems* 23 (1987) 131–147.
- [18] L.A. Zadeh, Fuzzy sets, *Inform. Control* 8 (1965) 338–353.