OPTIMAL CONTROL APPLICATIONS AND METHODS
Optim. Control Appl. Meth., 2003; **24**:279–296 (DOI: 10.1002/oca.732)

Solving multi-objective dynamic optimization problems with fuzzy satisfying method

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

Department of Chemical Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C.

SUMMARY

This article proposes a novel algorithm integrating iterative dynamic programming and fuzzy aggregation to solve multi-objective optimal control problems. First, the optimal control policies involving these objectives are sequentially determined. A payoff table is then established by applying each optimal policy in series to evaluate these multiple objectives. Considering the imprecise nature of decision-maker's judgment, these multiple objectives are viewed as fuzzy variables. Simple monotonic increasing or decreasing membership functions are then defined for degrees of satisfaction for these linguistic objective functions. The optimal control policy is finally searched by maximizing the aggregated fuzzy decision values. The proposed method is rather easy to implement. Two chemical processes, Nylon 6 batch polymerization and Penicillin G fed-batch fermentation, are used to demonstrate that the method has a significant potential to solve real industrial problems. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: optimal control; multi-objective optimization; fuzzy set; iterative dynamic programming

1. INTRODUCTION

In recent years, a considerable number of studies have been made on the optimization of dynamic systems with single objective [1–7]. However, multiple aims are usually desired in practice. For instance, when operating a batch reactor, maintaining undesirable byproducts at the lowest possible levels and attaining the desired fractional conversion in the shortest amount of time are both important. Furthermore, the operators can simultaneously consider other important factors such as economic efficiency, safety, reliability, or the impact on the environment. All these objectives are usually non-commensurable. Operators thus need a multi-objective decision-making technique to help them look for a satisfying solution from those conflicting objectives. Recently, application of multi-objective approach on dynamic optimization problems has been addressed by many researchers [8].

Optimization for a multi-objective problem is a procedure looking for a compromise policy. The result, called a Pareto optimal or non-inferior solution, consists of an infinite number of

*Correspondence to: Professor Cheng-Liang Chen, Department of Chemical Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C
† E-mail: ccl@ccms.ntu.edu.tw

Received 22 January 2001
Revised 20 May 2003

alternatives. There is a large variety of methods for treating the multi-objective optimization problem. Those methods can be classified in many ways according to different criteria [9, 10]. For example, Cohon [11] categorized methods into two relatively distinct subsets: generating methods and preference-based methods. The generating methods produce a set of Pareto optima and then the decision maker (DM) selects one of them on a basis of subjective value judgment. Among them the weighting-sum method and the ε-constraint method are well-known. As the algorithm cannot converge to a suitable solution, or the DM does not agree with the result, the DM can adjust the related parameters used in the algorithm, such as the weighting factors in the weighting-sum method. The computation can be repeated until a satisfactory solution is obtained. The preference-based methods, on the other hand, contain DM's preference as the solution process goes on, and the solution that best fulfills DM's preference is selected. Thus, all these multi-objective optimization methods for finding a Pareto optimal solution are filled with subjective and fuzzy properties [10].

In order to overcome the difficulty of describing a fuzzy attribute, Zadeh [12] proposed the fuzzy set concept. By using multi-valued logic to replace the traditional Boolean logic, people can quantitatively elucidate unclear information or knowledge. Afterward, Bellman and Zadeh [13] further extended the fuzzy concept to the decision making under fuzzy environment. Tanaka *et al*. [14] brought in the concept of fuzzy mathematical programming and proved that fuzzy mathematical programming can be reduced into conventional non-linear programming problem. Zimmermann [15] introduced fuzzy set theory into conventional linear programming problems, considering linear programming problems with a fuzzy goal and some fuzzy constraints. Following the fuzzy decision, together with linear membership functions, he proved that there exists an equivalent linear programming problem. Recently, Sakawa *et al*. [16] proposed the fuzzy satisfying method to find solution for multi-objective linear problems by applying the payoff table. Wang *et al*. [17] proposed a fuzzy decision-making procedure to determine the feed profile of a fermentation process for fuel ethanol production, using the fuzzy min–max method. In this paper, we attempt to extend the fuzzy inference on solving the dynamic optimization problem with multiple objectives. The augmented min–max approach, proposed by Sakawa *et al*. [16], is applied instead of the conventional min–max approach used by Wang and Shieh [18] since the uniqueness of the optimal solution is not guaranteed in non-linear systems. By mapping each objective value into a normalized domain referenced from the payoff table and further aggregating these normalized values, the vector objective problem can be grouped into a single objective problem. Then, the iterative dynamic programming (IDP), developed by Luus and his co-workers [4], is utilized as the platform to determine the solution for such problems. By using appropriate mapping functions and aggregates, the solution found by the proposed algorithm can be proved to be (local) Pareto optimal. Notably, by applying the fuzzy aggregation and payoff table approach, one can obtain single Pareto optimal solution that best satisfies the decision maker with least subjective knowledge. Furthermore, the grouped single objective dynamic optimization problem can also be solved by using other searching methods, such as the integrated controlled random search (ICRS) [7, 19].

In the rest of this article, the formulation of the problem is set out in Section 2. The procedure for grouping the vector objectives into a scalar one using the fuzzy set concept is given in Section 3. Some related mathematical properties required to guarantee the optimality of the solution are given in Section 4. In Section 5, a review and modification for IDP is presented. Therein, limitations of proposed method on local Pareto optimum is also discussed. Two numerical examples are supplied in Section 6, demonstrating the usefulness of the proposed method. Some conclusions and discussions are made in Section 7.

2. PROBLEM STATEMENT

Consider the following multi-objective dynamic optimization problem (MODOP) with a specified final time t_f,

$$\begin{aligned} &\min_{\mathbf{u}(t)\in\Omega} J_1(\mathbf{x}(t_f)) \\ \text{and} \quad &\min_{\mathbf{u}(t)\in\Omega} J_2(\mathbf{x}(t_f)) \\ &\vdots \\ \text{and} \quad &\min_{\mathbf{u}(t)\in\Omega} J_I(\mathbf{x}(t_f)) \end{aligned} \tag{1}$$

or in a more compact form

$$\min_{\mathbf{u}(t)\in\Omega} \mathbf{J}(\mathbf{x}(t_f)) = [J_1(\mathbf{x}(t_f)), \ldots, J_I(\mathbf{x}(t_f))]^{\mathrm{T}} \tag{2}$$

Here, $\mathbf{x}(t)$ denotes the $n \times 1$ state vector with initial condition $\mathbf{x}(0) = \mathbf{x}_0$; $\mathbf{u}(t)$ is the $m \times 1$ control vector; Ω is the feasible region in the control action space $\mathbf{u}(t)$ that satisfies some constraints, $\Omega = \{\mathbf{u}(t)|\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) = \dot{\mathbf{x}}(t) - \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)),\ \mathbf{x}(0) = \mathbf{x}_0, \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \leqslant \mathbf{0}, \underline{\mathbf{u}} \leqslant \mathbf{u}(t) \leqslant \bar{\mathbf{u}}\}$; $\mathbf{h}(\bullet) = [h_1(\bullet), \ldots, h_L(\bullet)]^{\mathrm{T}}$ and $\mathbf{g}(\bullet) = [g_1(\bullet), \ldots, g_K(\bullet)]^{\mathrm{T}}$ are equality and inequality constraints, respectively; $\underline{\mathbf{u}}$ and $\bar{\mathbf{u}}$ are the lower/upper bounds for control vector; and $\mathbf{J}(\mathbf{x}(t_f))$ denotes an $I \times 1$ dimensional objective function vector.

The multi-objective dynamic optimization problem is to determine the optimal control policy $\mathbf{u}^*(t)$ over $t \in [0, t_f]$ which brings the state vector $\mathbf{x}(t)$ from the initial condition $\mathbf{x}_0$ to the final state $\mathbf{x}(t_f)$ so that these objectives in $\mathbf{J}(\mathbf{x}(t_f))$, regardless of the commensurability, are minimized under given constraints. These objective functions, however, are usually conflicted with one another in practice. It is thus impossible to attain their own optimum, $J_i^*(\mathbf{x}(t_f))'s$, simultaneously. The optimization of one objective implies the sacrifice of other targets. Therefore, the decision maker must make some compromise among these goals. In contrast to the optimality used in single objective optimization problems, Pareto optimality characterizes the solutions in a multi-objective optimization problem [9–11].

Definition 1

$\mathbf{u}^*(t) \in \Omega$ is said to be Pareto optimal for Equation (2), if and only if there exists no $\mathbf{u}(t)$ such that $J_i(\mathbf{u}(t), \mathbf{x}(t_f)) \leqslant J_i(\mathbf{u}^*(t), \mathbf{x}^*(t_f))$ for all $i \in \{1, \ldots, I\}$ and $J_j(\mathbf{u}(t), \mathbf{x}(t_f)) < J_j(\mathbf{u}^*(t), \mathbf{x}^*(t_f))$ for some $j \in \{1, \ldots, I\}$.

From the above definition, the number of solutions satisfying Pareto optimality in a multi-objective optimization problem can be infinite. It is difficult for the DM to attribute a set of incompatible objectives, such as economic efficiency, safety, reliability, or environmental impact, without knowledge of the possible level of attainment for those objectives. It is thus a fuzzy problem for finding a Pareto optimal solution that best satisfies the decision maker.

3. FUZZY DECISION MAKING IN MODOP

In this work, we extended the fuzzy set theory of Zadeh [12] to deal with the multi-objective dynamic optimization problem. By considering the uncertain property of human thinking, it is quite natural to assume that the DM has multiple fuzzy goals, $\mathscr{J}_i$, $i = 1, \ldots, I$, where an interval $[J_i^\ell, J_i^u]$ exists for each fuzzy objective $\mathscr{J}_i$. For the ith minimum objective, it is thoroughly satisfied as the objective value $J_i(\mathbf{x}(t_f))$ is less than J_i^ℓ, and it is unacceptable as $J_i(\mathbf{x}(t_f)) > J_i^u$. For a $J_i(\mathbf{x}(t_f))$ value in between J_i^ℓ and J_i^u, the extent of satisfaction by the DM decreases with an increase in its value. A strictly monotonic decreasing membership function, $\mu_{\mathscr{J}_i}(J_i(\mathbf{x}(t_f))) \in [0, 1]$, can be used to characterize such a transition from the objective value, $J_i(\mathbf{x}(t_f))$, to the degree-of-satisfaction, $\mu_{\mathscr{J}_i}$ [7]

$$\mu_{\mathscr{J}_i}(J_i(\mathbf{x}(t_f))) = \begin{cases} 1 & \text{for } J_i(\mathbf{x}(t_f)) < J_i^\ell \\ F_{\mathscr{J}_i}(J_i(\mathbf{x}(t_f)); J_i^\ell, J_i^u) & \text{for } J_i^\ell \leqslant J_i(\mathbf{x}(t_f)) \leqslant J_i^u \\ 0 & \text{for } J_i(\mathbf{x}(t_f)) > J_i^u \end{cases} \tag{3}$$

Here, a membership value of 1 denotes absolute satisfaction and 0 means unacceptable. The original MODOP is now equivalent to look for a suitable control policy that can provide the maximal degree-of-satisfaction for the multiple fuzzy objectives.

$$\begin{aligned} &\max_{\mathbf{u}(t)\in\Omega} \mu_{\mathscr{J}_1}(J_1(\mathbf{x}(t_f))) \\ \text{and} \quad &\max_{\mathbf{u}(t)\in\Omega} \mu_{\mathscr{J}_2}(J_2(\mathbf{x}(t_f))) \\ &\vdots \\ \text{and} \quad &\max_{\mathbf{u}(t)\in\Omega} \mu_{\mathscr{J}_I}(J_I(\mathbf{x}(t_f))) \end{aligned} \tag{4}$$

or in a more compact form,

$$\max_{\mathbf{u}(t)\in\Omega} \mu(\mathbf{x}(t_f)) = [\mu_{\mathscr{J}_1}(J_1(\mathbf{x}(t_f))), \ldots, \mu_{\mathscr{J}_I}(J_I(\mathbf{x}(t_f)))]^{\mathrm{T}} \tag{5}$$

Under incompatible objective circumstances, a DM must make a compromise decision that provides a maximum degree-of-satisfaction for all of these conflict objectives. The new optimization problem, Equation (5), can be interpreted as the synthetic notation of a conjunction statement (maximize jointly all objectives). The result of this aggregation, $\mathscr{D}$, can be viewed as a fuzzy intersection of all fuzzy objectives, $\mathscr{J}_i, i = 1, \ldots, I$, and is still a fuzzy set.

$$\mathscr{D} = \mathscr{J}_1 \cap \mathscr{J}_2 \cap \cdots \cap \mathscr{J}_I \tag{6}$$

The final degree-of-satisfaction resulting from certain kinds of control actions, $\mu_{\mathscr{D}}(\mathbf{x}(t_f))$, over $t \in [0, t_f]$ can be determined by aggregating the degree-of-satisfaction for all objectives, $\mu_{\mathscr{J}_i}(J_i\mathbf{x}(t_f))$ via specific t-norm, $\mathbb{T}$.

$$\mu_{\mathscr{D}}(\mathbf{x}(t_f)) = \mathbb{T}\{\mu_{\mathscr{J}_1}(J_1(\mathbf{x}(t_f))), \ldots, \mu_{\mathscr{J}_I}(J_I(\mathbf{x}(t_f)))\} \tag{7}$$

The fundamental properties for a fuzzy set and the related operators can be found in Reference [20]. As the firing level for each control policy is determined by the above procedure, the best control policy $\mathbf{u}^*(t)$ with the maximal firing level, $\mu^*_{\mathscr{D}}(\mathbf{u}(t))$, over $t \in [0, t_f]$ can be selected.

$$\begin{aligned} \mu^*_{\mathscr{D}}(\mathbf{x}(t_f), \mathbf{u}^*(t)) &= \max_{\mathbf{u}(t)\in\Omega} \mu_{\mathscr{D}}(\mathbf{x}(t_f)) \\ &= \max_{\mathbf{u}(t)\in\Omega} \mathbb{T}\{\mu_{\mathscr{J}_1}(J_1(\mathbf{x}(t_f))), \ldots, \mu_{\mathscr{J}_I}(J_I(\mathbf{x}(t_f)))\} \end{aligned} \tag{8}$$

Two famous t-norms are discussed here.

1. $\mathbb{T}$ = algebraic product

$$\mu^*_{\mathscr{D}}(\mathbf{x}(t_f), \mathbf{u}^*(t)) = \max_{\mathbf{u}(t)\in\Omega} \prod_{i=1}^{I} \mu_{\mathscr{J}_i}(J_i(\mathbf{x}(t_f))) \tag{9}$$

2. $\mathbb{T}$ = Zadeh-minimum

$$\mu^*_{\mathscr{D}}(\mathbf{x}(t_f), \mathbf{u}^*(t)) = \max_{\mathbf{u}(t)\in\Omega} \min\{\mu_{\mathscr{J}_1}(J_1(\mathbf{x}(t_f))), \ldots, \mu_{\mathscr{J}_I}(J_I(\mathbf{x}(t_f)))\} \tag{10}$$

Notably, an equivalent expression when the Zadeh-minimum is used as the t-norm is,

$$\max_{\mathbf{u}(t)\in\Omega^+} \alpha \tag{11}$$

where $\mathbf{\Omega}^+ = \mathbf{\Omega} \cap \{\mu_{\mathscr{J}_i}(J_i(\mathbf{x}(t_f))) \geqslant \alpha,\ i = 1, \ldots, I\}$.

4. RELATED MATHEMATICAL THEOREMS

Since the original problem, Equation (2), has been modified as an another multi-objective problem, Equation (5), using the t-norm, this new problem is converted into a new single objective problem, Equation (8). It is necessary to ensure that the solution from this procedure is Pareto for Equation (2). Since the original problem may be non-convex, only local Pareto is guaranteed if a local optimizer is applied [9]. Should the global optimum is emphasized for non-convex non-linear optimal control problems, one can refer to recent work done by Esposito and Floudas [21]. However, the calculations involved can be hard to unravel. Notably, that Equation (5) is also a multi-objective problem, it makes sense to give the following definitions [10]:

Definition 2
$\mathbf{u}^*$ in Ω is said to be (local) $\mathscr{M}$-Pareto optimal for Equation (2) if and only if it is (local) Pareto for Equation (5), that is, there exists no $\mathbf{u}$ such that $\mu_{\mathscr{J}_i}(\mathbf{u}) \geqslant \mu_{\mathscr{J}_i}(\mathbf{u}^*)\ \forall i$, with strict inequality for some i, where $\mathscr{M}$ refers to membership.

Note that the set of (local) Pareto optimal solutions is a subset of the set of (local) $\mathscr{M}$-Pareto optimal solutions. Using the above definitions, it is clear that only (local) $\mathscr{M}$-Pareto solutions can be looked for as Equation (2) is solved through Equation (5) and a local optimizer is used. Thus, it is quite natural to question

1. How can we guarantee a (local) $\mathscr{M}$-Pareto solution for Equation (2) as the t-norm is used?
2. Under what conditions is a (local) $\mathscr{M}$-Pareto solution a (local) Pareto?

To answer these queries, the following propositions [22] can be applied:

Proposition 1
Let $\mathbb{T}$ be an arbitrary t-norm. If $\mathbf{u}^*$ is the unique (local) optimal solution for Equation (8), then $\mathbf{u}^*$ is (local) $\mathscr{M}$-Pareto.

Proposition 2
If $\mathbf{u}^*$ is a (local) optimal solution for Equation (8) such that $\mu_{\mathscr{J}_i} \neq 0\ \forall i$, and $\mathbb{T}$ is strictly monotonous, then $\mathbf{u}^*$ is (local) $\mathscr{M}$-Pareto for Equation (2).

Based on Propositions 1 and 2, the following corollaries arise:

Corollary 1
An (local) optimal solution for Equation (9), say $\mathbf{u}^*$ is (local) $\mathscr{M}$-Pareto for Equation (2), if either it is the only (local) solution or $\mu_{\mathscr{J}_i}[\mathbf{u}^*] \neq 0\ \forall i$, since the Product is a strictly monotonous t-norm.

Corollary 2
An (local) optimal solution from Equation (10) or Equation (11) will be (local) $\mathscr{M}$-Pareto when it is unique, because the Zadeh-min is not strictly monotonous.

Based on these corollaries and the property of the product operator, as the control policy causes some objectives to deviate from their own optimum to an unacceptable extent, membership values for these objectives will become zero. At that time, this control policy will be rejected. Thus, as the product is used for solving Equation (8), any meaningful solution will be (local) $\mathscr{M}$-Pareto. On the other hand, when Zadeh-min is used as the aggregate, a test for the uniqueness of the (local) optimal solution for Equation (10) or Equation (11) is unavoidable to ensure the (local) $\mathscr{M}$-Pareto solution. Furthermore, as strictly decreasing functions are used to calculate membership value for each objective, the solution with (local) Pareto optimality can be found from the following propositions:

Proposition 3
Let $\mathbf{u}^*$ be (local) $\mathscr{M}$-Pareto for Equation (2) with $\mu_{\mathscr{J}_i} \in (0, 1)\ \forall i$. If all of the $\mu_{\mathscr{J}_i}$ functions are strictly decreasing, then $\mathbf{u}^*$ is (local) Pareto for Equation (2).

Proposition 4
Let $\mathbf{u}^*$ be an (local) optimal solution for Equation (8) with all of the $\mu_{\mathscr{J}_i}$ being strictly decreasing. $\mathbf{u}^*$ is the (local) Pareto for Equation (2) if either

1. $\mathbf{u}^*$ is the unique (local) solution or
2. $\mathbb{T}$ is strictly monotonous and $\mu_{\mathscr{J}_i} \in (0,1)\ \forall i$.

From the above discussions, if Zadeh-min is selected as the t-norm, the key point that guarantees (local) Pareto solution is to determine the unique optimal solution. To achieve this, two strategies can be utilized. The first is to employ a global optimization technique to determine the solution [21]. However, it is difficult to justify whether the solution is the only global optimum or not. The second strategy is to employ the augmented minimax algorithm [10]. The merit of this method is that it circumvents the necessity for testing the uniqueness of the solution by modifying the objective function. Based on this algorithm, Equation (10) can be reformulated as follows:

$$\min_{\mathbf{u}(t)\in\Omega} \left\{ \max_{\forall i} [\bar{\mu}_i - \mu_{\mathscr{J}_i}(J_i)] + \rho \sum_{i=1}^{I} [\bar{\mu}_i - \mu_{\mathscr{J}_i}(J_i)] \right\} \tag{12}$$

or equivalently

$$\begin{aligned} &\min_{\mathbf{u}(t)\in\Omega} \quad \omega \\ &\text{s.t.} \quad \bar{\mu}_i - \mu_{\mathscr{J}_i}(J_i) \leqslant \omega - \rho \sum_{i=1}^{I} [\bar{\mu}_i - \mu_{\mathscr{J}_i}(J_i)] \quad i = 1,\ldots,I \end{aligned} \tag{13}$$

where $\bar{\mu}_i$ is the reference membership level determined by the DM, and ρ is a sufficiently small positive constant. For such a modification, we can apply the following theorem, which is similar to the works done by Sakawa *et al.*, to ensure the properties of the solution:

Theorem 1
Let $\mathbf{u}^*$ be a (local) optimal solution for Equation (12) or Equation (13) for some $\bar{\mu}_i,\ i = 1,\ldots,I$. Then $\mathbf{u}^*$ is a (local) Pareto optimal solution of Equation (2).

5. THE ITERATIVE DYNAMIC PROGRAMMING (IDP) INCORPORATING FUZZY DECISION

Methods for solving the dynamic optimization problems can be classified into three major categories: (1) the variation-based approach, (2) the non-linear programming (NLP) approach and (3) the dynamic programming. The variation-based methods, a direct applications of Pontryagin's minimum principle, transforms the original problem into a two-point-boundary-value problem (TPBVP). It is usually a complex and extreme difficult task to solve the resulting TPBVP, however. The NLP approaches, including complete parameterization [1, 3] and control parameterization [2, 5, 6, 19], transform the dynamic optimization problem into a general non-linear optimization problem. Any standard NLP technique can then be used to determine the solution. All of these parameterization methods exhibit convergence difficulties, and it is difficult to obtain a global optimum because of the highly non-linear, multi-modal, and/or discontinuous natures of these systems. The direct use of dynamic programming in solving dynamic optimization problems is usually difficult. Two major barriers discourage the use of dynamic programming. The greatest difficulty is the problem of setting up the grid values for the state and control. To produce a meaningful result, the state grids must be sufficiently fine. At each time stage, therefore, a large numbers of integration must be performed for each state grid and each allowable control value. A greater problem arises when the trajectory calculated for a particular grid point does not meet a grid point at the next time step. Under such condition, interpolation can be used, but the resulting approximation is usually unreliable.

Recently, Luss [4] proposed the so-called iterative dynamic programming (IDP) to alleviate the computational burdens of the original dynamic programming. By using the accessible grid points and region-reduction strategy, the IDP can successfully overcome the curse of dimensionality, and its computational effectiveness has been elucidated in many reports, such as Reference [23]. Although IDP is not a deterministic global optimizer which can guarantee convergence to the global solution, there are many advantages. First, it is easy for implementation and no gradient information is required. Second, the probability of obtaining the global solution is high and can be even higher by increasing the parameter resolution and by repeating the optimization process with several different initial trials. These properties are very practical in real industrial environments. One of the best inherent properties of dynamic programming is that it is easy to extend the algorithm into parallel or distributed computation. As the computation efforts is heavily related to the number of state grids, the computation effort can be reduced significantly if multiple searching loops work simultaneously. Therefore, the barrier of numerous integrations is overcome by the parallel version of IDP, proposed by Hartig *et al.* [24]. In this article, only the regular version is demonstrated because the major purpose of this paper is to illustrate the fuzzy decision making.

In order to apply the IDP to solve our problem, Equation (8), we first divide the entire time horizon into P time stages, $P = t_f/T$. Further assume that the control action within each duration T is kept constant. Therefore, the original infinite-dimensional problem can then be put into the following finite-dimensional form:

$$\max_{\mathbf{u}_i \in \tilde{\Omega}_d} \mathbb{T}\{\mu_{\mathscr{J}_1}(J_1(\mathbf{x}(t_f))), \ldots, \mu_{\mathscr{J}_I}(J_I(\mathbf{x}(t_f)))\} \tag{14}$$

where $\tilde{\Omega}_d$ is the feasible searching space defined as

$$\begin{aligned}\tilde{\Omega}_d = \{\mathbf{u}_0, \ldots, \mathbf{u}_{P-1} \,|\, \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) = \dot{\mathbf{x}}(t) - \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \mathbf{x}(0) = \mathbf{x}_0; \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \leqslant \mathbf{0};\\ \underline{\mathbf{u}} \leqslant \mathbf{u}(t) = \mathbf{u}_i \leqslant \bar{\mathbf{u}} \;\; \forall t \in [iT, iT+T], \;\; i = 0, \ldots, P-1\}\end{aligned}$$

The multi-objective dynamic optimization problem can now be regarded as finding a series of piecewise constant control actions $\mathbf{u}_i, i = 0, \ldots, P-1$ with the highest firing level to attain the least compromise between the objectives, meanwhile the state equations and related constraints are all satisfied.

The IDP-based algorithm for the MODOP can be stated as follows:

- *Preliminary phase (I): generation of state grid points*
  1. Divide the time horizon t_f into P time stages each with length $T = t_f/P$.
  2. Choose the number of state grid points, N; the number of allowable control values, M; and the testing region r for control action.

Table I. The payoff table for a standard multi-objective optimization problem.

	$J_1(\mathbf{u}_i^*)$	$\cdots$	$J_s(\mathbf{u}_i^*)$	$\cdots$	$J_I(\mathbf{u}_i^*)$
$J_i(\mathbf{u}_1^*)$	$J_1^*(\mathbf{u}_1^*)$	$\cdots$	$J_s(\mathbf{u}_1^*)$	$\cdots$	$J_I(\mathbf{u}_1^*)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
$J_i(\mathbf{u}_s^*)$	$J_1(\mathbf{u}_s^*)$	$\cdots$	$J_s^*(\mathbf{u}_s^*)$	$\cdots$	$J_I(\mathbf{u}_s^*)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
$J_i(\mathbf{u}_I^*)$	$J_1(\mathbf{u}_I^*)$	$\cdots$	$J_s(\mathbf{u}_I^*)$	$\cdots$	$J_I^*(\mathbf{u}_I^*)$

3. By Choosing M sets of control actions at the time stages, integrate the dynamic equation M times to generate the state grids at the time stages.

- *Preliminary phase (II): generation of payoff table*

1. Determine the optimal value for each single objective, J_i^*, and record the corresponding control policy, $\mathbf{u}_i^*$. Meanwhile record the other objective values when applying $\mathbf{u}_i^*$, i.e. $J_{s,s\neq i}\ s = 1, \ldots, I$.
2. Construct the payoff table (Table I),
3. Let $J_i^\ell = J_i^*(\mathbf{u}_i^*)$ and $J_i^u = \max_{\forall s}\{J_i(\mathbf{u}_s^*)\}$. Determine the membership function $F_{\mathscr{J}_i}$ for each objective.
4. Choose a suitable t-norm to perform aggregation. If the Zadeh-min is used, it is needed to modify the objective function and assign the $\bar{\mu}_i$ and ρ values. The default values for $\bar{\mu}_i$ and ρ can be set as 1.0 and 10^{-3}, respectively.

- *Search by the IDP*:

1. Begin at the $(P-1)T$, corresponding to time $t_f - T$. Integrate the dynamic equation from time $t_f - T$ to t_f, starting from each grid point and using each of the M allowable control values.
2. Calculate membership values for all objects at t_f attained by the M control actions at each grid point.
3. Take aggregation on each objective via the selected t-norm operator to calculate firing levels for the control actions.
4. Select the control action with highest firing level as the best choice for the state grid and record it.
5. Step back one stage, corresponding to time $t_f - 2T$. Integrate the dynamic equation from time $t_f - 2T$ to t_f for each state grid with M allowable control actions. To continue integration from $t_f - T$ to t_f, use the best control action for the closet state grid point $t_f - T$.
6. Re-calculate membership values for all objects at t_f, since state trajectory at t_f does not exactly meet target value obtained from previous computation. Take aggregation on objects to calculate firing level for each control value used at this stage.
7. Select the control action with highest firing level as the best choice.
8. Continue the above procedure until $t = 0$. Store control actions with maximal fuzzy decisions and store the corresponding state trajectory.

9. Reduce the region for state grid and region for allowable control values using a contracting factor ε; i.e.

$$r^{(j+1)} = (1-\varepsilon)r^{(j)} \tag{15}$$

where j is iteration index. Use the optimal state trajectory from previous search as the midpoint for the state grid at each time stage. Use the optimal control policy from previous steps as the midpoint for allowable values for control.

10. Increment the iteration index j by 1 and go back to step 1. Continue the iteration for a specific number of iterations, such as 20, and examine the results.

Notably, the number of required integrations (N_T) for each iteration can be analytically figured out as follows:

$$N_T = M \times P \times \left[1 + \frac{N(P-1)}{2}\right] \tag{16}$$

One can estimate the overall computation time if M, N, P and the number of iterations are given. Though the proposed method requires huge resources of computer processing time, it has a significant potential to solve real industrial problems due to its easy implementation. The feasibility and superiority of the IDP for solving the MODOP is demonstrated in next section.

6. NUMERICAL EXAMPLES

Two chemical processes are used to demonstrate the proposed algorithm. The first example concerns the multi-objective dynamic optimization for the non-vaporized Nylon-6 polymerization in the batch process. The second is production of penicillin G via fed-batch fermentation. We use the Pentium-100 personal computer, and DVERK and LSODI integration packages for computation.

6.1. Example 1: Nylon 6 polymerization problem

The optimization for Nylon 6 polymerization has drawn considerable attention in the past decade. Ray and Gupta [25] explored the optimum temperature profiles using the minimum principle for respective single objective. Recently, Wajge and Gupta [26] applied the so-called surrogate worth trade-off (SWT) method to study the operation under two objective functions. Based on the mass and moment equations proposed by Ramagopal *et al.* [27], we attempt to find the temperature profiles $T(t)$ which simultaneously optimize the following three objectives:

$$\begin{aligned}
\min J_1 &= \text{concentration of unreacted monomer in product}\\
\min J_2 &= \text{concentration of undesirable cyclic compounds}\\
&\quad\ \text{(primarily, cyclic dimer) in product, } [C_2]_{t_f} \text{ (mol/kg)}\\
\min J_3 &= \text{reaction time, } t_f(h)
\end{aligned}$$

Meanwhile, the following constraints are considered:

1. To ensure a single phase polymerization, the control variable (temperature) should be limited within the range between 220 and 270°C,

$$220^{\circ}\mathrm{C} \leqslant T(t) \leqslant 270^{\circ}\mathrm{C} \tag{17}$$

2. To guarantee processing properties (number average molecular weight $M_N(t_f)$ within 1.4×10^4 and 2×10^4), the stopping criterion is setting the number average chain-length of polymer, $CL_N(t_f)$, to be 140

$$CL_N(t_f) = 140 \tag{18}$$

Under this condition, the number average molecular weight $M_N(t_f)$ is about 1.6×10^4.

3. To facilitate and smooth control profile for implementation, the change in temperature between successive time intervals should not exceed 20°C

$$|T(t_k) - T(t_{k-1})| \leqslant 20^{\circ}\mathrm{C} \tag{19}$$

Furthermore, each time interval must be larger than 15 min

$$t_k - t_{k-1} > 15 \text{ min} \tag{20}$$

In this example, we set $N = 49$, $M = 5$, $P = 10$ and 30 iterations for each run of optimization. With the same physical data, initial conditions and tolerance as that of Wajge and Gupta [26], we first determine optimal values for considered objectives to establish the payoff table, as shown in Table II. The results in Table II show that:

- When the polymerization temperature is maintained at its highest value, 270°C, the concentration of unreacted monomer can be kept at the lowest level and the reaction time the shortest. The cyclic compound concentration is increased, however.
- The reaction temperature should be kept at the lowest value, 220°C, to suppress the cyclic compound concentration.

From the above results, the control profiles can be qualitatively divided into three stages:

Stage 1: Initializing the reaction with the highest temperature to promote the monomer conversion.
Stage 2: Keeping the temperature at its lowest level to limit the concentration of undesired product as low as possible.
Stage 3: Re-increasing the temperature to raise the product concentration.

Table II. Payoff table for Nylon 6 batch polymerization.

$T(t)$	$J_1(\mathbf{u}_i^*) = [C_1]_{t_f}$	$J_2(\mathbf{u}_i^*) = [C_2]_{t_f}$	$J_3(\mathbf{u}_i^*) = t_f$
$J_i(\mathbf{u}_1^* = 270^{\circ}\mathrm{C})$	1.9005	0.01202	6.071
$J_i(\mathbf{u}_2^* = 220^{\circ}\mathrm{C})$	2.2430	0.00735	35.33
$J_i(\mathbf{u}_3^* = 270^{\circ}\mathrm{C})$	1.9005	0.01202	6.071

Based on Table II, we set up the following linear type membership functions, designated as Condition-1, to express our preference on each objective.

$$\mu_{\mathcal{J}_1} = \begin{cases} 1 & \text{if } J_1 < 1.9005 \\ \dfrac{2.2430 - J_1}{2.2430 - 1.9005} & \text{if } 1.9005 \leqslant J_1 \leqslant 2.2430 \\ 0 & \text{if } J_1 > 2.2430 \end{cases} \tag{21}$$

$$\mu_{\mathcal{J}_2} = \begin{cases} 1 & \text{if } J_2 < 0.00735 \\ \dfrac{0.01202 - J_2}{0.01202 - 0.00735} & \text{if } 0.00735 \leqslant J_2 \leqslant 0.01202 \\ 0 & \text{if } J_2 > 0.01202 \end{cases} \tag{22}$$

$$\mu_{\mathcal{J}_3} = \begin{cases} 1 & \text{if } J_3 < 6.071 \\ \dfrac{35.330 - J_3}{35.330 - 6.071} & \text{if } 6.071 \leqslant J_3 \leqslant 35.330 \\ 0 & \text{if } J_3 < 35.330 \end{cases} \tag{23}$$

Here, both min and product are utilized as fuzzy aggregation for exploring the effect on final results, such as shown in Table III. Table III illustrates the fact that min operator has the tendency to distribute the preference on average. It can be seen in the third column of the table that membership values are all equal to 0.61. However, the product is more inclined to concentrate the preference on some objectives. This can be shown in the fifth column that membership values for $\mathcal{J}_1$ and $\mathcal{J}_3$ are larger than that of $\mathcal{J}_2$. Control profiles for Condition-1 are shown in the following. Notably, applying the min operator in aggregation does not guarantee that the final results is (local) Pareto optimal. We modify the max–min type objective function into the augmented min–max type objective function. The simulation results for this modification are listed in Table IV, Comparing the results from Tables III to IV, we find that although we have replaced the max–min type objective function with the augmented min–max type function, the final results do not change significantly. The possible reason is that IDP determines the unique solution for the former objective function. But, to ensure (local) Pareto optimality, we still recommend using the augmented min–max type objective function (Figure 1).

Table III. Comparisons between min and product operator under Condition-1 (Example 1).

	min		product	
		$\mu_{\mathcal{J}}$		$\mu_{\mathcal{J}}$
$[C_1]_{t_f}$	2.0339	0.61	1.9634	0.8164
$[C_2]_{t_f}$	0.00917	0.61	0.00989	0.4563
t_f	17.472	0.61	14.354	0.7169

Table IV. Results for augmented min–max type function under Condition-1 (Example 1).

	Aug. min–max ($\rho = 10^{-3}$)		Aug. min–max ($\rho = 10^{-5}$)	
		$\mu_{\mathscr{J}}$		$\mu_{\mathscr{J}}$
$[C_1]_{t_f}$	2.0342	0.61	2.0341	0.61
$[C_2]_{t_f}$	0.00917	0.61	0.00917	0.61
t_f	17.441	0.61	17.478	0.61

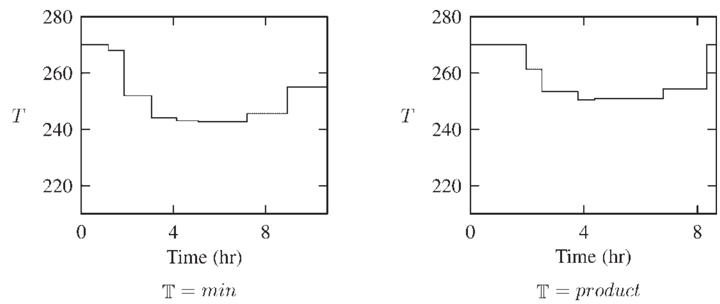

Figure 1. Temperature profiles for Condition-1 (Example 1).

Now considering a practical operation (8 h per day), the maximal allowable operating time is changed from 35.330 to 15 h (Condition-2). The membership function for operating time becomes,

$$\mu_{\mathscr{J}_3} = \begin{cases} 1 & \text{if } J_3 < 6.071 \\ \dfrac{15 - J_3}{15 - 6.071} & \text{if } 6.071 \leqslant J_3 \leqslant 15 \\ 0 & \text{if } J_3 > 15 \end{cases} \tag{24}$$

For this modification, we also use the min and the product operators to perform the aggregation at the beginning. The results are listed in Table V, which are similar to results from Condition-1. The control profiles are shown in Figure 2. We replaced the min–max type objection function by the augmented max–min objective function. The simulation results are shown in Table VI.

Table V. Comparisons when using min and product under Condition-2 (Example 1).

	min		product	
		$\mu_{\mathscr{J}}$		$\mu_{\mathscr{J}}$
$[C_1]_{t_f}$	2.0751	0.49	1.9917	0.7338
$[C_2]_{t_f}$	0.00973	0.49	0.01056	0.3126
t_f	10.624	0.49	8.747	0.7003

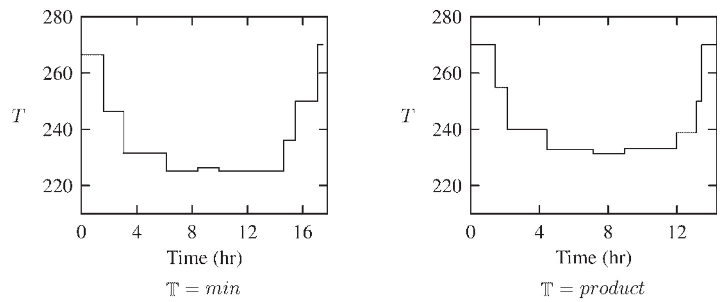

Figure 2. Temperature profiles for Condition-2 (Example 1).

Table VI. Results for augmented min–max type function under Condition-2 (Example 1).

	Aug. min–max ($\rho = 10^{-3}$)		Aug. min–max ($\rho = 10^{-5}$)	
		$\mu_{\mathscr{J}}$		$\mu_{\mathscr{J}}$
$[C_1]_{t_f}$	2.0759	0.49	2.0752	0.49
$[C_2]_{t_f}$	0.00974	0.49	0.00973	0.49
t_f	10.636	0.49	10.632	0.49

6.2. Example 2: Penicillin G fed-batch fermentation

The simplified Heijnen's model [28] was considered:

$$\begin{aligned}\frac{dS}{dt} &= -\sigma X + u(t)\\ \frac{dX}{dt} &= vX\\ \frac{dP}{dt} &= \pi X - k_h P\\ \frac{dG}{dt} &= \frac{1}{C_{s,in}} u(t) - 0.0008G\\ \frac{dC_s}{dt} &= \frac{1}{G}\frac{dS}{dt} - \frac{C_s}{G}\frac{dG}{dt}\end{aligned} \tag{25}$$

In these equations, S, X and P denote the amount of substrate (glucose), cell mass and product (penicillin) in broth, respectively. G is the total broth weight. u is substrate feed rate (mole/hr). $C_{s,in}$ and C_s designate the glucose concentration in feed and broth, respectively. k_h is the penicillin hydrolysis constant. The specific rates σ, v, and π are modelled as follows.

1. The specific substrate rate σ is modelled by a Monod-type relationship.

$$\sigma = Q_{s,max} \frac{C_s}{K_s + C_s} \tag{26}$$

where $Q_{s,max}$ denotes the maximum specific sugar uptake rate and K_s is the Monod constant for the sugar uptake.

2. The (overall) specific growth rate v is given by

$$v = Y_{X/S}(\sigma - m - \pi/Y_{P/S}) \tag{27}$$

where m is the overall specific maintenance demand. $Y_{X/S}$ is the biomass-on-substrate yield coefficient. $Y_{P/S}$ denotes the product-on-substrate yield coefficient.

3. The specific production rate π is assumed to be directly coupled with the specific growth rate μ, following a Blackman-type relation

$$\pi(v) = Q_{\mathrm{p,max}} \begin{cases} \dfrac{v}{v_{\mathrm{crit}}} & \text{for } v \leqslant v_{\mathrm{crit}} \\ 1 & \text{for } v \geqslant v_{\mathrm{crit}} \end{cases} \tag{28}$$

where $Q_{\mathrm{p,max}}$ denotes the maximal specific production rate, and v_{crit} is critical specific growth rate.

The initial conditions and the related physical data are shown in Table VII. Now, if we limit the glucose feeding rate as not exceeding 2000 mole/h, and assign the total amount of glucose during the operation as equal to 2×10^5 mol:

$$0 \leqslant u(t) \leqslant 2000 \text{ mol/h} \tag{29}$$

$$\int_0^{t_f} u(t)\,\mathrm{d}t = 2 \times 10^5 \text{ mol} \tag{30}$$

Such a fermentation process can be divided into two phases [28]. The first phase is rapid growth with almost no product formation and the second phase has limited growth during product formation. Thus the optimization problem can now be formulated as finding the glucose feeding policy $u(t)$ that simultaneously maximizes the final product amount $J_1 = P(t_f)$ and finishes the operation in the shortest time $J_2 = t_f$. That is,

$$J_1 = \max_{u(t)} P(t_f) = \min_{u(t)} -P(t_f) \tag{31}$$

$$J_2 = \min_{u(t)} t_f \tag{32}$$

Table VII. Initial conditions and constants used in Penicillin G fermentation model.

S	5500	mole	k_{h}	0.002	$\frac{1}{\mathrm{h}}$
X	4000	mol dry weight	$Q_{\mathrm{s,max}}$	0.0245	$\frac{\text{mole}}{(\text{mol dry weight})(h)}$
P	0	mol	$Q_{\mathrm{p,max}}$	3.3×10^{-4}	$\frac{\text{mole}}{(\text{mol dry weight})(h)}$
G	1×10^5	kg	m	0.0034	$\frac{\text{mole}}{(\text{mol dry weight})(h)}$
$C_{\mathrm{s,in}}$	1/0.36	$\frac{\text{mole}}{\text{kg}}$	$Y_{X/S}$	3.67	$\frac{\text{mol}}{\text{mol dry weight}}$
C_{s}	0.055	$\frac{\text{mole}}{\text{kg}}$	$Y_{P/S}$	0.46	$\frac{\text{mole}}{\text{mole}}$
K_{s}	0.0056	$\frac{\text{mole}}{\text{kg}}$	v_{crit}	0.01	$\frac{1}{\mathrm{h}}$

As stated previously, we must search for the optimal value for each single objective to build the payoff table. By using the IDP with $N = 49$, $M = 5$, $P = 10$ and the number of iteration is 30, we can produce 7878 mol of penicillin G during 232.2 h of operation if only the maximum production is considered. However, one can obtain 3463 mole of product when the shortest operating time, $t_f = 100$ h, is involved. The payoff table can be found in Table VIII. The linear type membership functions can then be constructed to depict the change in preference for individual objective.

$$\mu_{J_1} = \begin{cases} 1 & \text{if } J_1 > 7878 \\ \dfrac{J_1 - 3463}{7878 - 3463} & \text{if } 7878 \geqslant J_1 \geqslant 3463 \\ 0 & \text{if } J_1 < 3463 \end{cases} \tag{33}$$

$$\mu_{J_2} = \begin{cases} 1 & \text{if } J_2 < 100 \\ \dfrac{232.3 - J_2}{232.3 - 100} & \text{if } 100 \leqslant J_2 \leqslant 232.3 \\ 0 & \text{if } J_2 > 232.3 \end{cases} \tag{34}$$

As we use the min operator to perform the fuzzy aggregation, profiles for control policy and specific growth rate are shown in Figure 3. As product is used as the aggregating operator, the resulting control policy and specific growth rate are shown in Figure 4. The numerical results using different operators that are slightly different, as depicted in Table IX.

7. CONCLUSIONS

An algorithm applying iterative dynamic programming and fuzzy inference to solve the multi-objective optimal control problems is proposed. The optimal control policy for each objective is

Table VIII. Payoff table for Penicillin G fed-batch fermentation.

	$J_1(\mathbf{u}_i^*)$	$J_2(\mathbf{u}_i^*)$
$J_i(\mathbf{u}_1^*)$	7878	232.3
$J_i(\mathbf{u}_2^*)$	3463	100

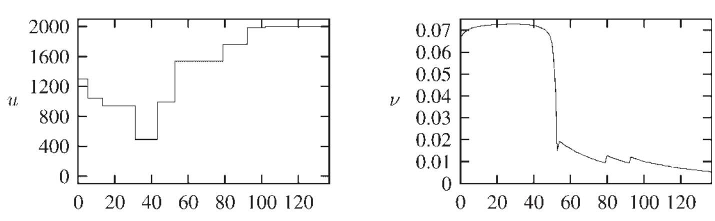

Figure 3. Control policy and specific growth rate μ ($\mathbb{T}$ = min).

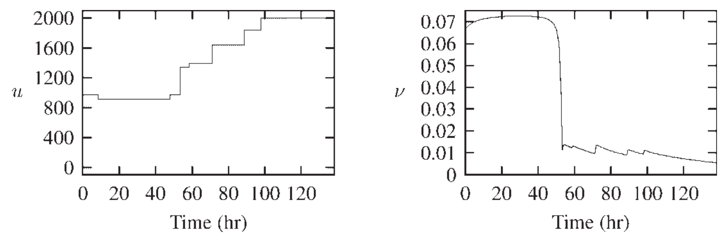

Figure 4. Control policy and specific growth rate ($\mathbb{T}$ = product).

Table IX. Comparisons for different t-norms for Penicillin G fed-batch fermentation.

Obj.	$\mathbb{T}$ = min	$\mathbb{T}$ = product
$P(t_f)$ (mol)	6649.6	6716.7
t_f (hrs)	136.8	138.2

determined sequentially at first. The payoff table is established by applying these optimal control policies to individual objective. A simple monotonic increasing or decreasing membership function is then used to define the degree of satisfaction for each objective function. The final optimal control policy is searched by maximizing the aggregated membership values. Herein, two popular t-norms, the min and the product, have been applied as the fuzzy aggregation. By applying the fuzzy aggregation and payoff table approach, one can obtain single Pareto optimal solution that best satisfies the decision maker with least subjective knowledge. This hybrid algorithm is not only easy to implement, but powerful and efficient in computation. Two chemical processes, the Nylon 6 batch polymerization and the Penicillin G fed-batch fermentation, are used to demonstrate the feasibility and superiority of the proposed algorithm.

REFERENCES

1. Biegler LT. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers and Chemical Engineering* 1984; **8**:243–248.
2. Goh CK, Teo KL. Control parameterization: a unified approach to optimal problems with general constraints. *Automatica* 1988; **24**:3–18.
3. Cuthrell JJ, Biegler LT. Simultaneous optimization and solution methods for batch reactor profiles. *Computers and Chemical Engineering* 1989; **13**:49–62.
4. Luus R. Optimal control by dynamic programming using grids points and region reduction. *Hungarian Journal of Industrial Chemistry* 1989; **17**:523–543.
5. Vassiladis RWVVSS, Pantelides CC. Solution of a class of multistage dynamic optimization problems—1. problems without path constraints. *Industrial Engineering and Chemical Research* 1994; **33**:2111–2122.
6. Vassiladis RWVVSS, Pantelides CC. Solution of a class of multistage dynamic optimization problems—2. problems with path constraints. *Industrial Engineering and Chemical Research* 1994; **33**:2123–2133.
7. Chen CL, Sun DY. Solution of fuzzy dynamic optimization problems by adaptive stochastic algorithm. *International Journal of Artificial Intelligence Tools* 2000; **9**(4):527–535.
8. Choi KY, Butala DN. An experimental study of multi-objective dynamic optimization of a semi-batch copolymerization process. *Polymer Engineering and Science* 1991; **31**(5):353.
9. Miettinen K. *Nonlinear Multi-objective Optimization*. Kluwer Academic: New York, 1999.
10. Sakawa M. *Fuzzy Sets and Interactive Multi-objective Optimization*. Plenum Press: New York, 1993.
11. Cohon JL. *Multi-objective Programming and Planning*. Academic Press: New York, 1985.

12. Zadeh LA. Fuzzy sets. *Information and Control* 1965; **8**:338–353.
13. Bellman RE, Zadeh LA. Decision making in a fuzzy environment. *Management Science* 1970; **17**:141–164.
14. Tanaka TO, Okuda H, Asai K. On the fuzzy mathematical programming. *Journal of Cybernetics* 1974; **3**:37–46.
15. Zimmermann HJ. Description and optimization of fuzzy systems. *International Journal of General Systems* 1976; **2**:209–215.
16. Sakawa M, Inuiguchi M, Kato K, Ikeda T. A fuzzy satisfying method for multi-objective linear optimal control problems. *Fuzzy Sets and Systems* 1996; **78**:223–229.
17. Wang FS, Jing CH, Tsao GT. Fuzzy-decision-making problems of fuel ethanol production using a genetically engineered yeast. *Industrial Engineering and Chemical Research* 1998; **37**:3434–3443.
18. Wang FS, Shieh TL. Extension of iterative dynamic programming to multi-objective optimal control problems. *Industrial Engineering and Chemical Research* 1997; **36**:2279–2286.
19. Carrasco EF, Banga JR. Dynamic optimization of batch reactors using adaptive stochastic algorithms. *Industrial Engineering and Chemical Research* 1997; **36**:2252–2261.
20. Klir GJ, Yuan B. *Fuzzy Sets and Fuzzy Logics-Theory and Application*. Prentice Hall: New York, 1995.
21. Esposito WR, Floudas CA. Deterministic global optimization in nonlinear optimal control problems. *Journal of Global Optimization* 2000; **17**:97–126.
22. Delgado M, Verdegay JL, Vila MA. A possibility approach for multi-objective programming problems: efficiency of solutions. In *Stochastic Versus Fuzzy Approaches to Multi-objective Mathematical Programming under Uncertainty*, Slowiniski R, Teghem J (eds). Kluwer Academic: New York, 1991.
23. Luus R. Optimal control of batch reactor by iterative dynamic programming. *Journal of Process Control* 1994; **4**:216–218.
24. Hartig F, Mandel K, Keil FJ. Parallelization of iterative dynamic programming. *Periodica Polytechnica Series in Chemical Engineering* 1999; **43**(1):3–16.
25. Ray AK, Gupta SK. Optimization of non-vaporizing nylon 6 reactors with stopping conditions and end-point constraints. *Polymer Engineering and Science* 1986; **26**:1033–1044.
26. Wajge RM, Gupta SK. Multi-objective dynamic optimization of a nonvaporizing nylon 6 batch reactor. *Polymer Engineering and Science* 1994; **34**:1161–1172.
27. Ramagopal AK, Kumar A, Gupta SK. Optimal temperature profiles for Nylon 6 polymerization in plug-flow reactor. *Journal of Applied Polymer Science* 1983; **28**:2261–2279.
28. van Impe JF. Optimal control of the penicillin G fed-batch fermentation: an analysis of the model of heijnen *et al.* *Optimal Control Applications and Methods* 1994; **15**:13–34.

