Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

An Effective Numerical Procedure for Evaluating Flexibility Indices of Dynamic Systems with Piecewise Constant Manipulated Variables

Shoeb Moon Ali, Shang-Wei Hwang, Chuei-Tin Chang*, Jo-Shu Chang

Department of Chemical Engineering, National Cheng Kung University, Tainan, Taiwan 70101

ARTICLE INFO

Article history:
Received 10 March 2021
Revised 22 June 2021
Accepted 24 July 2021
Available online 28 July 2021

Keywords:
dynamic flexibility index
unsteady state process
vertex method
genetic algorithm

ABSTRACT

Although chemical processes are traditionally evaluated according to economic criteria under the designated normal conditions, it is still necessary to ensure the operational feasibility of a given process design in conditions deviated from the nominal levels. The purpose of the current study is to reduce the computation time and improve the practical applicability of the quantification method for calculating dynamic flexibility index. In this study, the two-level optimization problem for computing flexibility index has been handled with two different solution strategies. The lower-level maximization is performed with a deterministic solver whereas the upper-level minimization a metaheuristic algorithm. To expedite implementation of the proposed methodology, the manipulated variables are treated as piecewise-constant functions of time. Two numerical examples of varying complexity are presented in this paper to demonstrate the feasibility and effectiveness of the aforementioned computation procedure.

1. Introduction

The chemical processes, designed on the basis of nominal operating conditions and parameter values, have traditionally been evaluated with economic criteria. This approach may end up with a plant which becomes inoperable in realistic environment if some of the operating conditions and/or model parameters significantly deviate from their nominal levels. These uncertainties are not always stochastic, as they may arise either from the unexpected exogenous disturbances (such as those in feed qualities, product demands, and environmental conditions) or from the inexplicable errors in estimation of model parameters (such as heat transfer coefficients, reaction rate constants, and other physical properties). Therefore, there is a need to incorporate considerations for uncertainties at the design stage. The ability of a chemical process to maintain feasible operation despite unexpected deviations from the nominal state is often referred to as its *operational flexibility*. Various computation approaches to facilitate quantitative flexibility analysis have already been proposed (Zhou et al, 2009; Chang and Adi, 2018).

The *steady-state flexibility index*, denoted as FI_s in this paper, has been used basically as a gauge of the feasible region in the parameter space for the continuous processes (Swaney and Grossmann, 1985I; Swaney and Grossmann, 1985II; Lima et al, 2010). This index is associated with the maximum allowable deviations of the uncertain parameters from their nominal values, throughout which a feasible operation can be ensured with proper adjustment of the manipulated variables. It was also shown that, under convexity assumptions, the critical points that limit flexibility must lie on the vertices of the hypercube inscribed in the parameter space. This particular insight is the foundation of the so-called “vertex method” for computing the flexibility index. In a later study, Grossmann and Floudas (1987) tried to simplify the two-level optimization problem of the vertex method by exploiting the fact that the active constraints represent bottleneck of a design and developed successive mixed integer linear (MILP) and mixed integer nonlinear programming (MINLP) models for computing the flexibility index. This calculation procedure has been referred to as the “active set method.”

Dimitriadis and Pistikopoulos (1995) later suggested characterizing an unsteady system with the *dynamic flexibility index* (FI_d). Two computation algorithms, which are the extended versions of the aforementioned vertex method and active set method, have also been proposed by Kuo and Chang (2016) and Wu and Chang (2017) respectively. Although successful applications on specific examples were reported, these methods are still not mature enough for the flexibility analysis in practice. It is often tedious to compute FI_d even for moderately complex dynamic systems. This is due to an overwhelmingly large number of vertices created by the need to discretize the differential equations in implementing the extended vertex method. In particular, if n_θ is the number of uncertain parameters and M is the number of discretized intervals

* Corresponding author.
E-mail address: ctchang@mail.ncku.edu.tw (C.-T. Chang).

https://doi.org/10.1016/j.compchemeng.2021.107464
0098-1354/© 2021 Elsevier Ltd. All rights reserved.

over the entire time horizon, then $(2^{n_\theta})^{M+1}$ should be the total number of vertices. On the other hand, in the case of the active set method, the necessary conditions for the constrained minimization problem must be incorporated to formulate the mathematical model. Since a large number of extra integer variables and the corresponding constraints are introduced, convergence to the global optimum is not always guaranteed.

Therefore, the primary objective of the current study is to develop a generalized numerical procedure so as to improve the efficiency of prevalent methods for computing the dynamic flexibility index. This procedure should in turn indirectly enhance the applicability of flexibility analysis for dynamic systems. The rest of the paper is organized as follows. Firstly a brief review of the conventional computation methods for evaluating the dynamic flexibility index is presented in section 2. The suggested improvements in the form of additionally imposed constraints are presented in section 3. Subsequently, the proposed computation strategies are presented in section 4. Section 5 depicts the numerical results obtained from application of these strategies to two dynamic processes and the critical analysis of these results. Section 6 highlights the advantages of the proposed methodology along with the comparison with its conventional counterpart. The final concluding remarks are given in section 7.

2. Review of Available Solution Methods

To facilitate clear illustration of the proposed computation strategies, a few related issues are first reviewed in the sequel.

2.1. Definition of dynamic flexibility index

Two index sets, I and J, have been introduced to enumerate and classify all constraints in the given model:

$$\mathrm{I} = \{i \mid i \text{ is the index of an equality constraint in the design model}\} \tag{1}$$

$$\mathrm{J} = \{j \mid j \text{ is the index of an inequality constraint in the design model}\} \tag{2}$$

The i^{th} equality constraint in the model can be expressed in general as

$$h_\mathrm{i}\Big(\mathbf{d}, \boldsymbol{z}(\mathrm{t}), \boldsymbol{x}(\mathrm{t}), \dot{\mathbf{x}}(\mathrm{t}), \boldsymbol{\theta}(\mathrm{t})\Big) = \dot{x}_\mathrm{i}(\mathrm{t}) - \varphi_i\Big(\mathbf{d}, \boldsymbol{z}(\mathrm{t}), \boldsymbol{x}(\mathrm{t}), \boldsymbol{\theta}(\mathrm{t})\Big) = 0 \tag{3}$$

where, $x_i(0) = x_i^0$; $i \in \mathrm{I}$; $t \in [0, H]$ and H is the length of time horizon; $\mathbf{d}$ represents a constant vector in which all design specifications are stored; $\boldsymbol{z}(\mathrm{t}) \in R^{n_z}$ denotes the manipulated variables at time t, $\boldsymbol{x}(\mathrm{t}) \in R^{n_x}$ denotes the state variables at time t and $\boldsymbol{\theta}(\mathrm{t}) \in R^{n_\theta}$ denotes the uncertain parameters at time t. Notice that φ_i is a given function of three types of functions of time, i.e., $\boldsymbol{z}(\mathrm{t})$, $\boldsymbol{x}(\mathrm{t})$, $\boldsymbol{\theta}(\mathrm{t})$, and it is usually established to model the dynamic behaviour of an unsteady process over the given time horizon. In this paper, the total number of equality constraints is denoted by n_e. Finally, notice that Eq. (3) implies that the given dynamic system consists of ordinary differential equations (ODEs) only.

Similarly, the j^{th} inequality constraint in this model can be written as

$$g_j\Big(\mathbf{d}, \boldsymbol{z}(\mathrm{t}), \boldsymbol{x}(\mathrm{t}), \boldsymbol{\theta}(\mathrm{t})\Big) \leq 0 \tag{4}$$

where, $j \in \mathrm{J}$ and g_j is also a given function. Note that Eq. (4) is often adopted to reflect the physical and/or chemical boundaries in a given process (e.g. the capacity limit). The total number of inequality constraints is denoted as n_i.

The anticipated upper and lower bounds on the uncorrelated uncertain parameters can be incorporated in the present model as follows.

$$\boldsymbol{\theta}^N(\mathrm{t}) - \Delta\boldsymbol{\theta}^-(\mathrm{t}) \leq \boldsymbol{\theta}(\mathrm{t}) \leq \boldsymbol{\theta}^N(\mathrm{t}) + \Delta\boldsymbol{\theta}^+(\mathrm{t}) \tag{5}$$

These bounds may be extracted from historical records for specific applications.

Let us next introduce a feasibility functional Ψ, whose *scalar* value is dependent upon the given design specifications in $\mathbf{d}$ and also the chosen feasible time profiles of parameters in $\boldsymbol{\theta}(t)$. More specifically, this functional must be determined by solving a two-level optimization problem described below:

$$\Psi\Big(\boldsymbol{d}, \boldsymbol{\theta}(\mathrm{t})\Big) = \min_{\boldsymbol{x}(\mathrm{t}),\boldsymbol{z}(\mathrm{t})} \max_{j,t} g_j\Big(\boldsymbol{d}, \boldsymbol{z}(\mathrm{t}), \boldsymbol{x}(\mathrm{t}), \boldsymbol{\theta}(\mathrm{t})\Big) \tag{6}$$

subject to the constraints in Eq. (3) and (4) for both the lower and upper-level optimization problems. Also, it is assumed in this study that the uncertain deviations in the system are realized before decisions can be taken to adjust the manipulated variables so as to counter the system upsets. Note that the given system can be guaranteed to be always operable only if the feasibility functional value is non-positive, i.e., $\Psi \leq 0$.

In order to facilitate the evaluation of dynamic flexibility index, FI_d, a scalar variable δ is introduced to adjust the ranges mentioned in Eq. (5), i.e.

$$\boldsymbol{\theta}^\mathrm{N}(\mathrm{t}) - \delta\Delta\boldsymbol{\theta}^-(\mathrm{t}) \leq \boldsymbol{\theta}(\mathrm{t}) \leq \boldsymbol{\theta}^\mathrm{N}(\mathrm{t}) + \delta\Delta\boldsymbol{\theta}^+(\mathrm{t}) \tag{7}$$

The corresponding dynamic flexibility index FI_d can be computed by solving another multi-level optimization problem (Dimitriadis and Pistikopoulos, 1995) i.e.

$$FI_d = \max \delta \tag{8}$$

subject to Eq. (7) and the inequality constraint presented below

$$\max_{\boldsymbol{\theta}(\mathrm{t})} \Psi\Big(\mathbf{d}, \boldsymbol{\theta}(\mathrm{t})\Big) \leq 0 \tag{9}$$

2.2. Extended vertex method

Under the assumption that the manipulated variables can be adjusted arbitrarily at will, an extended version of the traditional vertex method was developed by Kuo and Chang (2016) for computing FI_d. It was assumed that the critical points must be located at the vertices in a functional space formed by $\boldsymbol{\theta}(t)$. Based on this assumption, a two-level optimization problem was developed for computing the dynamic flexibility index, i.e.

$$FI_d = \min_k \max_{\delta,\boldsymbol{z}(t),\boldsymbol{x}(t)} \delta \tag{10}$$

subject to Eqs. (3) and (4) for the lower-level optimization problem and also the following constraints in a function space formed by all possible time profiles of $\boldsymbol{\theta}(t)$:

$$\boldsymbol{\theta}(t) = \boldsymbol{\theta}^k(t) = \boldsymbol{\theta}^N(t) + \delta\Delta\boldsymbol{\theta}^k(t) \tag{11}$$

where, $\Delta\boldsymbol{\theta}^k(t)$ denotes a vector pointing from the nominal point $\boldsymbol{\theta}^N(t)$ towards the k^{th} vertex ($k = 1, 2, \cdots, 2^{n_\theta}$) at time t. Note that each element in $\Delta\boldsymbol{\theta}^k(t)$ should be obtained from the corresponding entry in either $-\Delta\boldsymbol{\theta}^-(t)$ or $\Delta\boldsymbol{\theta}^+(t)$. Finally, since the corresponding computation procedure has already been presented elsewhere (Kuo and Chang, 2016), the detailed algorithm of the extended vertex method is omitted for the sake of brevity.

2.3. Extended active set method

Under the assumption of unconstrained manipulated variables, the mathematical program for computing the feasibility functional defined in Eqs. (3), (5) and (6) was reformulated by Wu and Chang (2017) with an extra scalar variable $u(t)$ as follows

$$\Psi\left(\mathbf{d},\boldsymbol{\theta}(t)\right)=\min_{\boldsymbol{x}(t),\boldsymbol{z}(t),u(t)} u(t)|_{t=H} \tag{12}$$

subject to the equality constraints in Eq. (3), and also

$$\dot{u}(t)=0 \tag{13}$$

$$g_j\left(\mathbf{d},\boldsymbol{z}(t),\boldsymbol{x}(t),\boldsymbol{\theta}(t)\right)\le u(t) \tag{14}$$

To facilitate derivation of the necessary conditions for this constrained functional minimization problem, all equality constraints in Eq. (3) were rewritten in a vector form as

$$\boldsymbol{\varphi}\left(\mathbf{d},\boldsymbol{z}(t),\boldsymbol{x}(t),\boldsymbol{\theta}(t)\right)-\dot{\boldsymbol{x}}(t)=0;\ \ \boldsymbol{x}(0)=\boldsymbol{x}^0 \tag{15}$$

An aggregated objective functional can be constructed by introducing Lagrange multipliers (denoted as μ_1, $\boldsymbol{\mu}_2$ and $\boldsymbol{\lambda}$) to incorporate of all constraints in this optimization problem, i.e.

$$L=u(H)+\int_0^H\left\{\mu_1(t)[0-\dot{u}]+\boldsymbol{\mu}_2(t)^T[\boldsymbol{\varphi}-\dot{x}]+\boldsymbol{\lambda}(t)^T[\mathbf{g}-u\mathbf{1}]\right\}dt \tag{16}$$

where, $\mathbf{1}=[1,1,\cdots,1]^{\mathrm{T}}$. The multipliers of all equality constraints, i.e., μ_1 and $\boldsymbol{\mu}_2$, are real, while those for the inequalities, i.e., $\boldsymbol{\lambda}$, should be real and nonnegative. By taking the first variation of L and then setting it to zero, one can produce the following four groups of necessary conditions from Eqs. (12) – (16):

$$\mu_1(0)=0,\mu_1(H)=1,\mathbf{x}(0)=\mathbf{x}_0,\boldsymbol{\mu}_{\mathbf{2}}(H)=0 \tag{17}$$

$$\dot{\boldsymbol{\mu}}_2=-\boldsymbol{\mu}_2^{\mathrm{T}}\left(\frac{\partial\varphi}{\partial\boldsymbol{x}}\right)-\boldsymbol{\lambda}^T\left(\frac{\partial g}{\partial\boldsymbol{x}}\right),\dot{\mu}_1=\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{1} \tag{18}$$

$$\boldsymbol{\mu}_2^{\mathrm{T}}\left(\frac{\partial\varphi}{\partial z}\right)+\boldsymbol{\lambda}^{\mathrm{T}}\left(\frac{\partial g}{\partial z}\right)=0^{\mathrm{T}} \tag{19}$$

$$\dot{\mathbf{x}}=\boldsymbol{\varphi},\dot{u}=0,\boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{g}-\mathrm{u}\mathbf{1})=\boldsymbol{\lambda}\ge 0 \tag{20}$$

Since, at least one of the inequality constraints must be active at a certain time instance when the extremum is reached, it is necessary to set $u(t)=0$ for the entire horizon. Thus, the conditions in Eq. (20) should be modified as follows:

$$\dot{\mathbf{x}}=\boldsymbol{\varphi},u=0,\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{g}=0,\boldsymbol{\lambda}\ge 0,\mathbf{g}\le 0 \tag{21}$$

Therefore, the dynamic flexibility index can be determined by minimizing δ, subject to the necessary conditions specified in Eqs. (17) – (19), (21), and also the aforementioned constraints imposed upon the uncertain parameters, i.e., Eq. (7), for computing the dynamic flexibility index.

Next, by introducing a time-dependent slack variable $s_j(t)$ and a corresponding binary variable $y_j(t)$ to characterize each inequality in Eq. (4), the last three constraints in Eq. (21), i.e., $\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{g}=0$, $\boldsymbol{\lambda}\ge 0$ and $\mathbf{g}\le 0$, can be rewritten as

$$\mathrm{g_j}\left(\mathbf{d},\boldsymbol{z}(t),\boldsymbol{x}(t),\boldsymbol{\theta}(t)\right)+s_j(t)=0 \tag{22}$$

$$s_j(t)-U\left[1-y_j(t)\right]\le 0 \tag{23}$$

$$\lambda_j(t)-y_j(t)\le 0 \tag{24}$$

where, $j\ \in\mathrm{J}$; $y_j(t)\in\{0,1\}$; $s_j(t)\ge 0$; $\lambda_j(t)\ge 0$; $t\in[0,H]$; U is a sufficiently large positive constant. Based on the constraints in (22) and (23), it can be deduced that $y_j=1$ implies $s_j=0$ and, thus, the corresponding inequality constraint becomes active at time t. As a result, the dynamic flexibility index can be determined by solving the programming model given below:

$$\mathrm{FI_d}=\min_{\delta,\mu_1(\mathrm{t}),\boldsymbol{\mu}_2(\mathrm{t}),\boldsymbol{\lambda}(\mathrm{t}),\boldsymbol{s}(\mathrm{t}),\boldsymbol{y}(\mathrm{t}),\boldsymbol{x}(\mathrm{t}),\boldsymbol{z}(\mathrm{t})}\delta \tag{25}$$

subject to the constraints in Eqs. (7), (15), (17) – (19) and (22) – (24). Notice that, for the purpose of computing FI_d, the original maximization problem is replaced here with a minimization one. This is because of that fact that Eqs. (3), (6) - (9) imply that at least one inequality constraint must be active at a particular time instance and this requirement can be guaranteed with the aforementioned model.

Notice that, although $\boldsymbol{z}(\mathrm{t})\in R^{n_z}$ are considered to be unspecified arbitrary functions of time over $[0,H]$ in (3) and (4), it is computationally more convenient and practically more feasible to view them as piecewise-constant profiles. Consequently, the dimension of space formed by the manipulated variables can be transformed from infinite to finite at n_zN_z+1 (where N_z is the number of horizontal line segments in the time profiles of manipulated variables) and, furthermore, the upper limit of the number of aforementioned *active* constraints in an optimum solution may be set to be this particular finite value. Finally, notice that $\boldsymbol{z}(\mathrm{t})$ should reduce to the original arbitrary functions of time in (3) and (4) if N_z approaches infinity. As a result, the flexibility index value should increase as the number of manipulated-variable pieces, i.e., N_z, increases. Furthermore, in most cases, this number does not have to be raised to a very high level for the corresponding FI_d to saturate and such a stabilized value should be taken as the actual dynamic flexibility index of the given system.

3. Auxiliary Constraints

As mentioned before, the presented study aims to overcome the inherent limitations in the prevalent methods in calculating the dynamic flexibility index. A few auxiliary constraints have been introduced to improve computation efficiency and they are outlined below.

Firstly, the constraints in (4) can usually be represented more explicitly in practical applications by dividing them into two groups of simpler inequalities. Specifically, these two groups may be written as

$$x_{j,lo}\le x_j(t)\le x_{j,up}\ \ \forall j\in\mathrm{J}^{(x)} \tag{26}$$

$$z_{j,lo}\le z_j(t)\le z_{j,up}\ \ \forall j\in\mathrm{J}^{(z)} \tag{27}$$

where, $\mathrm{J}^{(x)}\cup\mathrm{J}^{(z)}=\mathrm{J}$; $x_{j,lo}$ and $x_{j,up}$ are constants which denote the lower and upper bounds of the j^{th} state variable, respectively, and $z_{j,lo}$ and $z_{j,up}$ are the constants which represent the lower and upper bounds of the corresponding manipulated variables, respectively. To facilitate clear discussion later in this paper, the binary variables in (23) and (24) are rewritten as $y_{j,up}^{(x)}(t)$, $y_{j,lo}^{(x)}(t)$, $y_{j,up}^{(z)}(t)$ and $y_{j,lo}^{(z)}(t)$ for representing the binary variables corresponding to the inequality constraints in (26) and (27).

Secondly, since both the active set method and the vertex method are established on the basis of the necessary conditions of a stationary point, there may be multiple optimum solutions. In addition, it seems impractical to allow instantaneous and continuous adjustments of the manipulated variables in practice. Therefore, it is possible to assume that, without yielding suboptimal results, the manipulated variables are piecewise constant. As mentioned previously in subsection 2.3, the maximum number of active constraints in an optimum solution may then be set according to the number of the horizontal line segments. This assumption may be expressed mathematically as follows

$$\boldsymbol{z}(t) = \sum_{k=1}^{N_z} \boldsymbol{z}_k(t) \tag{28}$$

$$\dot{\boldsymbol{z}}_k\ (t) = 0, \forall t \in \left[\hat{t}_{k-1}, \hat{t}_k\right] \tag{29}$$

$$\boldsymbol{z}_k(t) = 0,\ \text{ if } t \notin \left[\hat{t}_{k-1}, \hat{t}_k\right] \tag{30}$$

where, $k = 1, 2, \cdots, N_z$ and $0 = \hat{t}_0 < \hat{t}_1 < \cdots < \hat{t}_{N_z} = H$. It should be noted that $N_z \neq n_z$, $N_z \leq$ M and, for convenience, $\hat{t}_k$ is chosen to be one of the time points (t_p) used for discretization. Notice also that $\boldsymbol{z}_k(t)$ should also be bounded according to (27) and, thus, $y^{(z)}_{j,up}(t)$ and $y^{(z)}_{j,lo}(t)$ can be replaced respectively by $y^{(z)}_{j,k,up}$, and $y^{(z)}_{j,k,lo}$ for $t \in [\mathrm{t}_{\mathrm{k}-1}, \mathrm{t}_{\mathrm{k}}]$.

Thirdly, Inequality (31) below represents the requirement that the sum of the total number of active constraints in the form of both (26) and (27) should not be more than ($n_z Nz + 1$).

$$\sum_{e=1}^{M} \sum_{j \in \mathrm{J}^{(x)}} \left[y^{(x)}_{j,up}(t_e) + y^{(x)}_{j,lo}(t_e)\right] + \sum_{k=1}^{N_z} \sum_{j \in \mathrm{J}^{(z)}} \left[y^{(z)}_{j,k,up} + y^{(z)}_{j,k,lo}\right] \leq n_z N_z + 1 \tag{31}$$

4. Proposed Computation Strategies

The previously described extended vertex method has been integrated with an effective metaheuristic search method, namely, the genetic algorithm (Holland, 1992), to devise an efficient computation procedure. Also, for efficient computation, the given set of dynamic equations is discretized and transformed into a set of algebraic equations with the trapezoidal rule. For the sake of completeness, more specific discretization steps are outlined in the Appendix. The detailed description of the proposed search strategies and the complete computation procedure are provided henceforth.

4.1. Vertex based search strategy using genetic algorithm

After discretization and incorporation of the auxiliary constraints, (3), (4), (10) and (11) can be replaced with the following formulation:

$$FI_d = \min_{k} \max_{\delta_k, \boldsymbol{Z}, \boldsymbol{X}} \delta_k \tag{32}$$

subject to (26), (27) and

$$\boldsymbol{\theta}(t_p\ ) = \boldsymbol{\theta}^N(t_p\ ) +\ \delta_k \Delta\boldsymbol{\theta}^k(t_p\ ) \tag{33}$$

where, $p = 0, 1, 2, \cdots, M$; $\mathrm{k} = 1, 2, \cdots, (2^{n_\theta})^{M+1}$; $\boldsymbol{X} = [\boldsymbol{x}(t_1),\ \boldsymbol{x}(t_2),\ \ldots\ldots \boldsymbol{x}(t_M)]$; $\boldsymbol{Z} = [\boldsymbol{z}(\hat{t}_1),\ \boldsymbol{z}(\hat{t}_2),\ \ldots\ldots \boldsymbol{z}(\hat{t}_{N_z})]$.

To implement the genetic algorithm, every chromosome is encoded with the time points at which the critical corner of the feasible region of uncertain parameters shifts. Specifically, the chromosome can be constructed with a sequence of $n_z N_z + 1$ genes, i.e.,$(\tilde{t}_1, \tilde{t}_2, \cdots, \tilde{t}_{n_z N_z+1})$, where $\tilde{t}_l$ denotes the l^{th} $(l = 1, 2, \cdots, n_z N_z + 1)$ time point at which the aforementioned corner shift takes place. Note that the total number of such time points (i.e. $\tilde{t}_1, \tilde{t}_2, \cdots, \tilde{t}_{n_z N_z+1}$) is bounded according to (31) and n_z and N_z are assumed to be given. This is because of the fact that an inequality constraint most likely goes active in response to drastic change in one or more uncertain parameter. The proposed search strategy may be summarized as follows:

i The individuals (chromosomes) in the first generation are created with the random number generator embedded in MATLAB. Each gene in a chromosome should correspond to $\tilde{t}_l$. Specifically, the chromosome structure may be expressed as $[\tilde{t}_1, \tilde{t}_2, \cdots, \tilde{t}_{n_z N_z+1}]$ and $\tilde{t}_1 < \tilde{t}_2 < \cdots < \tilde{t}_{n_z N_z+1}$.

ii The lower-level maximization problem in Eq. (32) along with the corresponding constraints in Eqs. (A1), (A2) in the Appendix, (28)-(31) and (33) for every individual in each generation is solved with GAMS.

iii The upper-level minimization in Eq. (32) for each generation is performed in MATLAB by selecting the smallest value among all δ_k obtained in the said generation.

iv In implementing the genetic algorithm, the individuals to be propagated to the next generation are selected through Roulette-wheel selection method. Subsequently, the single-point crossover and mutation operations are performed according to their respective probabilities provided by the user.

v Steps (ii) to (iv) are repeated till the given generation number is reached. The smallest δ_k value amongst all generations is chosen as the dynamic flexibility index, FI_d.

For the lower-level maximization calculations suggested in Eq. (32), the CONOPT solver in GAMS (version 27.3.0) was deployed. The use of a local solver here is appropriate since the same optimization run is carried out repeatedly for every individual in each generation. This practice increases the probability of ultimately finding the global optimum in a computationally efficient way.

The genetic algorithm has been realized with a MATLAB code (MATLAB version 2020a). The data flows between GAMS and MATLAB were facilitated with interface software GDXMRW (GAMS Data Exchange MATLAB Read Write). The input to GAMS is an individual chromosome. The CONOPT solver in GAMS is used to evaluate the maximum value of δ_k value for each chromosome in every generation that corresponds to lower-level maximization problem described in Eq. (32), which is subject to the constraints in (A1), (A2) in the Appendix, (28)-(31) and (33), and then return to the MATLAB, where the minimum δ_k values of all generations are compared and the smallest of them is reported as the dynamic flexibility index, FI_d. The GA parameters in MATLAB code are provided by the user, which include: the upper and lower bounds of the genetic value (time domain), the maximum number of generations, the total number of individuals in each generation and the respective probabilities for crossover and mutation. One can find further details from the flowchart in Fig. 1.

4.2. A generalized computation procedure

It has been explained previously in subsection 2.3 that it is advantageous to treat the manipulated variables as piecewise-constant time profiles. This approach has been incorporated into the aforementioned GA-assisted vertex enumeration strategy to develop the following generalized compuation procedure.

i In the first step, for a given n_z and N_z, FI_d is evaluated with the GA-assisted vertex search strategy described in the previous subsection. The time points at which the corner shifts might occur can also be obtained from the corresponding optimization result.

ii The evaluation of FI_d in the above step is performed repeatedly after gradually increasing the number of manipulated-variable instances, N_z. The computation is terminated when the FI_d value converges. The resulting FI_d is considered to be the dynamic flexibility index for the given system.

The flowchart of the above calculation procedure is presented in Fig. 2.

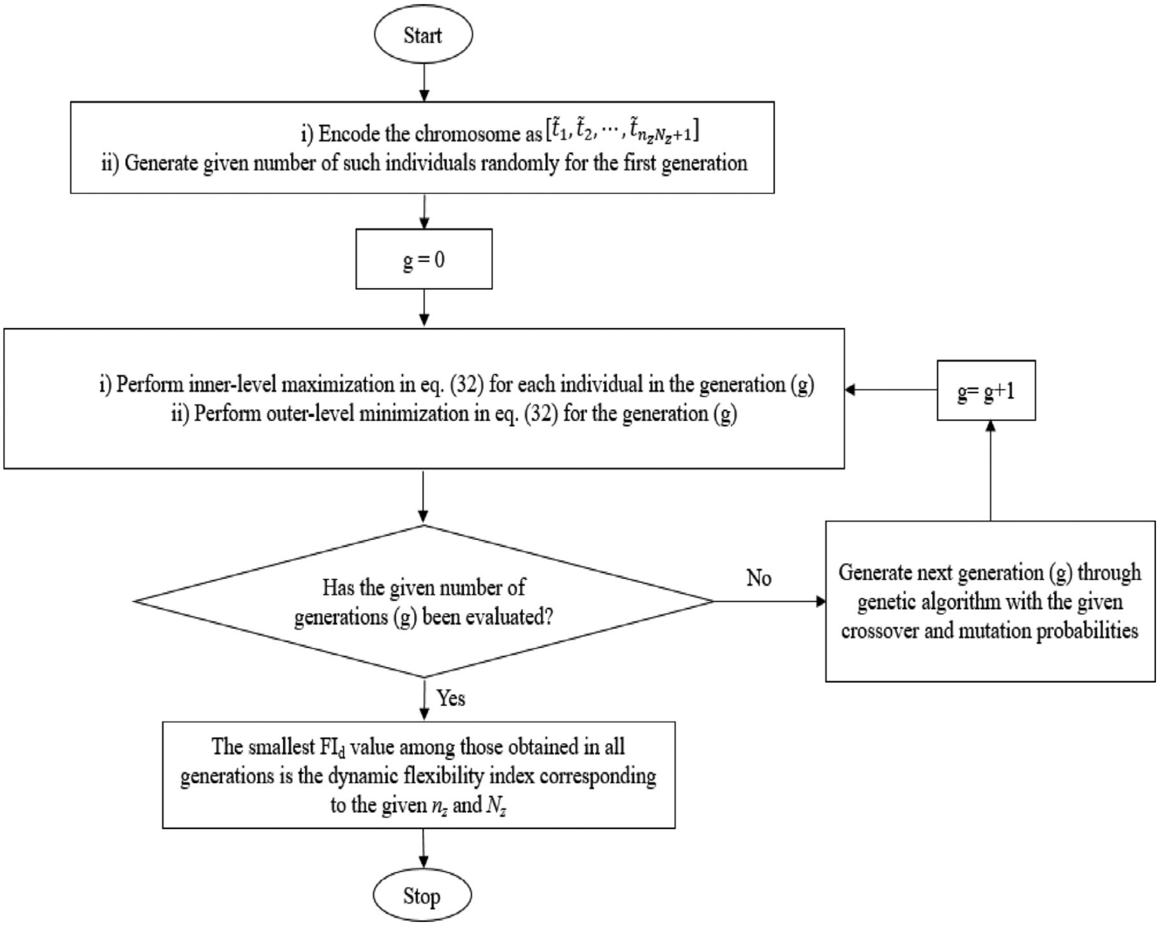
Start
i) Encode the chromosome as $[\tilde{t}_1, \tilde{t}_2, \cdots, \tilde{t}_{n_z N_z+1}]$
ii) Generate given number of such individuals randomly for the first generation
g = 0
i) Perform inner-level maximization in eq. (32) for each individual in the generation (g)
ii) Perform outer-level minimization in eq. (32) for the generation (g)
g= g+1
Has the given number of generations (g) been evaluated?
No
Generate next generation (g) through genetic algorithm with the given crossover and mutation probabilities
Yes
The smallest FI_d value among those obtained in all generations is the dynamic flexibility index corresponding to the given n_z and N_z
Stop

Fig. 1. Genetic algorithm assisted vertex enumeration

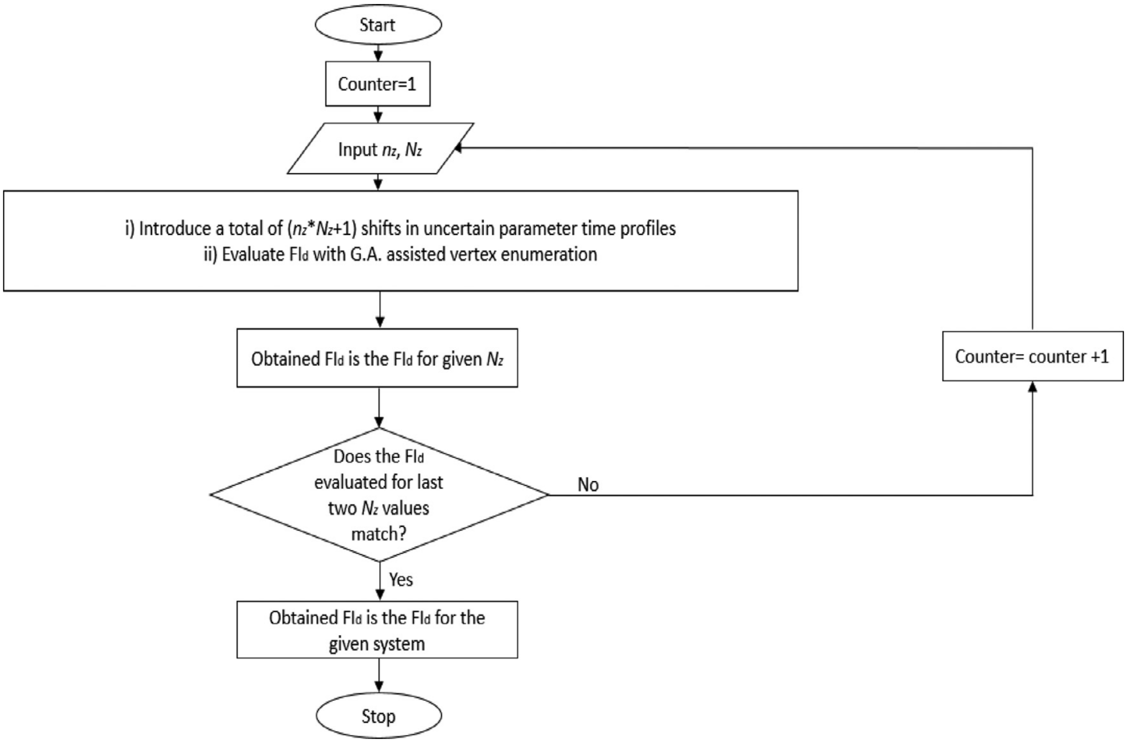
Start
Counter=1
Input n_z, N_z
i) Introduce a total of (n_z*N_z+1) shifts in uncertain parameter time profiles
ii) Evaluate FI_d with G.A. assisted vertex enumeration
Obtained FI_d is the FI_d for given N_z
Counter= counter +1
Does the FI_d evaluated for last two N_z values match?
No
Yes
Obtained FI_d is the FI_d for the given system
Stop

Fig. 2. Generalized computation procedure to evaluate FI_d

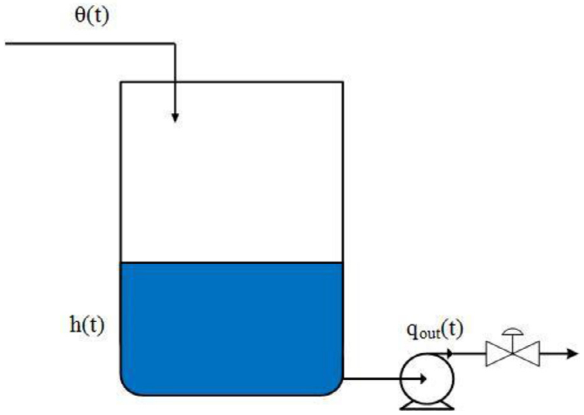

Fig. 3. A buffer tank

5. Numerical Examples

5.1. Buffer Tank

Let's consider the buffer tank shown in Fig. 3. The corresponding dynamic model can be written as;

$$A\frac{dh(t)}{dt} = \theta(t) - q_{out}(t) \tag{34}$$

where, h denotes the height of liquid level (m) and it is the only state variable in the present example; A $(= 5\ \mathrm{m}^2)$ is the cross-sectional area of the tank and it is a design specification; θ denotes the feed flowrate $(\mathrm{m}^3/\mathrm{min})$ and it is treated as an uncertain parameter; q_{out} denotes the flow rate in the outlet pipeline $(\mathrm{m}^3/\mathrm{min})$ and is the manipulated variable.

The allowed ranges of state and manipulated variables are chosen in this example as follows;

The upper bound on the height of water level in tank is 10 m, while the minimum allowable height is set at 1 m due to the head requirement of pump suction. In other words, the state variable h should be kept in the following range

$$1 \le h \le 10. \tag{35}$$

Since the maximum outlet flowrate is dependent on the pump discharge pressure and control valve specifications, it is assumed that this flow rate can be varied in the range of

$$0 \le q_{out} \le 0.7 \tag{36}$$

To facilitate dynamic flexibility quantification, let us assume that the time horizon covers a span of 800 minutes, i.e., $0 \le t \le 800$, with an initial height of liquid level at 5 m. Although the proposed procedure can be adopted for computing the flexibility index of any dynamic system, let us consider the steady operation of the above buffer tank for illustration convenience. Specifically, it is assumed that the nominal level of feed rate is kept constant throughout the horizon at $\theta^N(t) = 0.5\ \mathrm{m}^3/\mathrm{min}$ and, due to uncertain upstream disturbances, the corresponding largest possible positive and negative deviations are: $\Delta\theta^+(t) = \Delta\theta^-(t) = 0.5\ \mathrm{m}^3/\mathrm{min}$.

As mentioned before, the maximum number of active constraints in the given system can be determined according to the total number of shifts in the uncertain parameter profile. Specifically, starting from $N_z = 2$ in the present example, the total number of shifts in the uncertain parameter profile has been calculated based on Eq. (31), i.e., $n_z N_z + 1$. The proposed computation procedure has been applied to the buffer system described above. Note that, the number of generations adopted in GA was taken to be 100, whereas the probabilities for crossover and mutation were chosen to be 0.8 and 0.2, respectively. It has been observed that FI_d of the present system tends to increase as the number of manipulated-variable instances, N_z, increases. However, the value of the former becomes saturated at a finite value of the latter. In other words, beyond a certain value of N_z, FI_d tends to be stabilized eventually. This final value of FI_d is taken to be the dynamic flexibility of the given system.

A summary of the computation results obtained in the present example can be found in Table 1. In this table, the number of manipulated-variable instances, the dynamic flexibility value, the time points of active constraints and the convergence time are listed in column 1 – column 4, respectively. The time profiles of the state variable (h), the manipulated variable (q_{out}) and the uncertain parameter (θ) for $N_z = 2$ and $N_z = 8$ are shown in Figs. 4(a) - 4(c) and Figs. 5(a) - 5(c) respectively. Notice that the dashed lines in Fig. 4(c) and Fig. 5(c) both denote the nominal level of the uncertain parameter. It can also be observed from Table 1, Fig. 4(a) and Fig. 5(a) that the state variable (h) is activated 3 times for the case when $N_z = 2$ whereas h gets activated only once for the case when $N_z = 8$. The time profile of manipulated variable (q_{out}) in Fig. 4(b) can be found to be at two different values within the specified horizon when $N_z = 2$, while it can be seen from Table 1 and Fig. 5(b) that q_{out} always remains activated at the upper bound $(0.7\ \mathrm{m}^3/\mathrm{min})$ when $N_z = 8$.

It can also be witnessed from Figs. 4 and 5 that activation of constraint(s) in state and/or manipulated variable(s) is expectedly preceded by a corner shift in the uncertain parameters. Let us consider the numerical results in Table 1, Figs. 4(a) and 4(c) when $N_z = 2$. It can be observed that the liquid height in tank (h) gets activated at upper bound around 526 minutes, which occurs in response to the corner shift from negative to positive direction in uncertain parameter (θ) around 200 minutes.

5.2. Alcoholic Fermentation Process

Beer making is one of the earliest biochemical reactions known to humans. The brewing method involves soaking of starch-containing grains in water to ferment with other organic substances. Fig. 6 shows an alcoholic-fermentation system (Fabro and Arruda, 2003; Serra et al, 2005), which can be considered as a continuously stirred tank reactor (CSTR). In particular, the composition of the outlet stream is assumed to be the same as that within the tank. $S_a(t)$ in Fig. 6 denotes the nutrient, i.e., glucose, concentration (g/L) in the feed stream and is assumed to be an uncertain parameter in this example. On the other hand, $T(t)$ is another uncertain parameter which represents the temperature (°C) of the culture. $F_{in}(t)$ and $F_{out}(t)$ represent the flowrates (L/h) of the incoming and outgoing streams respectively and they are treated as the manipulated variables in this example. The state variables of the system include the glucose concentration, i.e., $S(t)$, the yeast-cell concentration, i.e., $C(t)$, the alcohol concentration, i.e., $P(t)$, and the total volume of the solution in the fermentation tank, i.e., $V(t)$, respectively.

The dynamic model of the alcoholic fermentation system may be expressed as follows:

$$\frac{dS}{dt} = -\frac{1}{Y_{C/S}}\mu C + \frac{F_{in}}{V}S_a - \frac{F_{out}}{V}S \tag{37}$$

$$\frac{dC}{dt} = \mu C - \frac{F_{out}}{V}C \tag{38}$$

$$\frac{dP}{dt} = \frac{Y_{P/S}}{Y_{C/S}}\mu C - \frac{F_{out}}{V}P \tag{39}$$

$$\frac{dV}{dt} = F_{in} - F_{out} \tag{40}$$

Table 1
Dynamic flexibility index of buffer tank

Number of manipulated- variable pieces (N_z)	FI_d	Time points of activated constraints (Minutes)	Convergence time (Seconds)
2	0.264	h: 176, 526, 790	957
4	0.468	h: 85,650; q_{out}: 201-400	1190
6	0.480	h: 800; q_{out}: 1-800	1371
8	0.480	h: 633; q_{out}: 1-800	1495

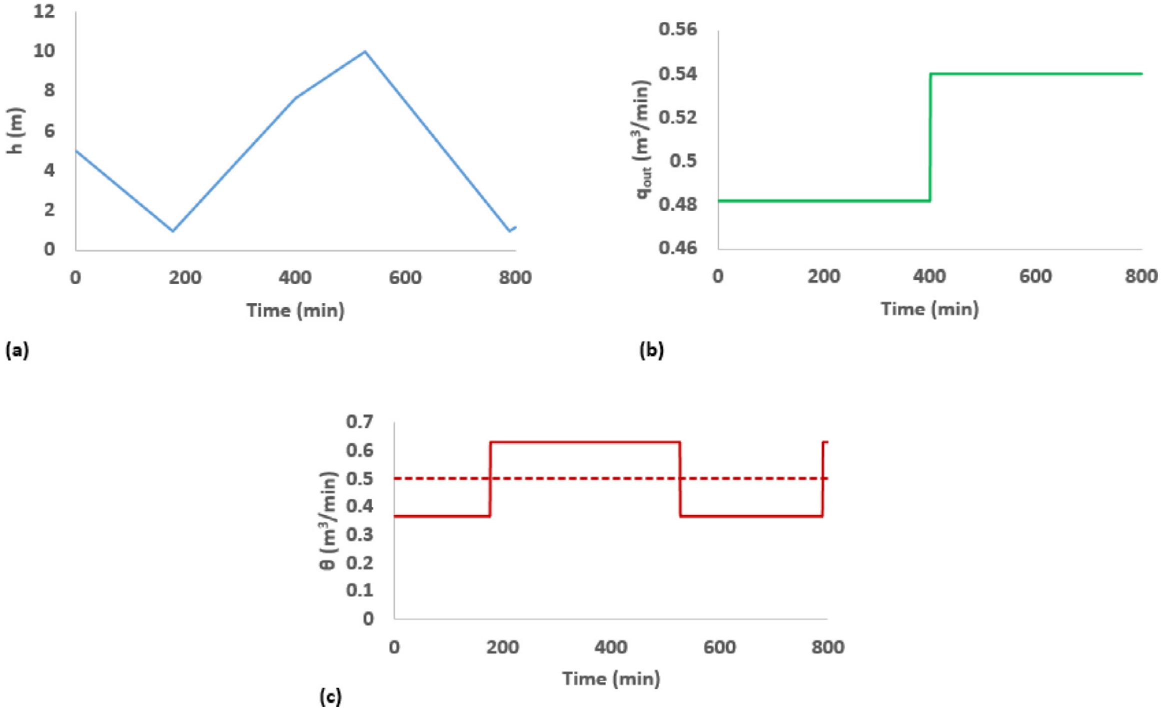

Fig. 4. Buffer tank variable time profiles for case N_z=2. (a) Height of liquid in tank, h; (b) Outlet flow rate, q_{out}; (c) Inlet flow rate, θ

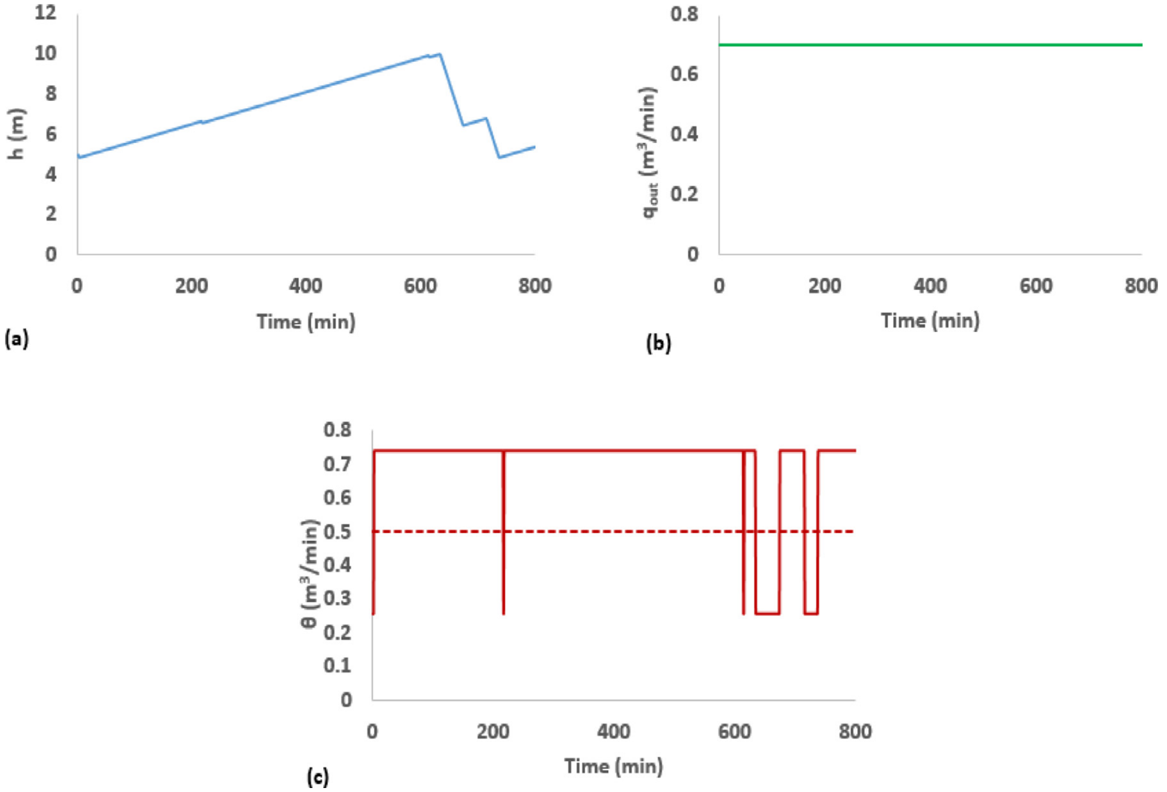

Fig. 5. Buffer tank variable time profiles for case N_z=8. (a) Height of liquid in tank, h; (b) Outlet flow rate, q_{out}; (c) Inlet flow rate, θ

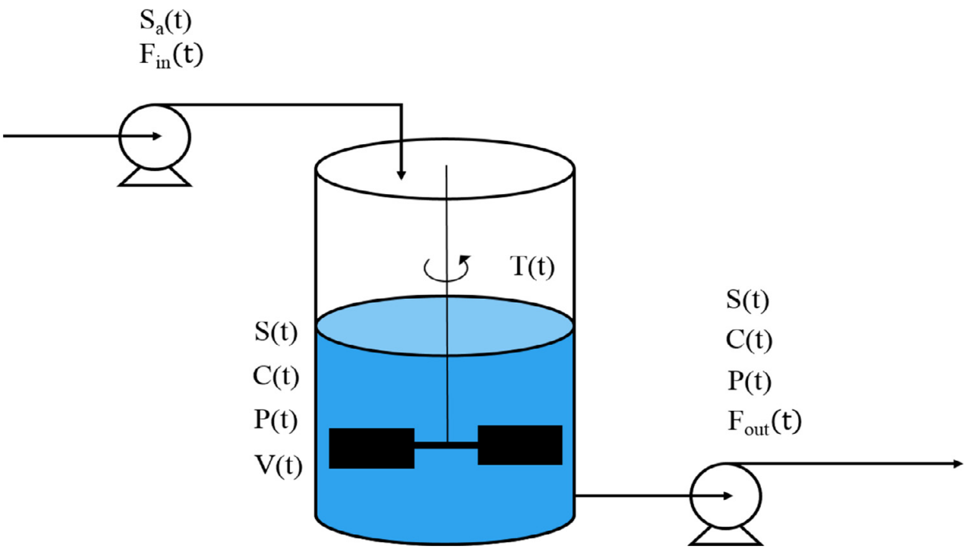

Fig. 6. Alcoholic fermentation in a CSTR

where, $Y_{C/S}$ $(= 0.07)$ is the nutrient-and-cell transformation constant; $Y_{P/S}$ $(= 0.44)$ is the nutrient-and-product conversion constant; μ (h^{-1}) is the specific growth rate of yeast cells, which can be expressed as

$$\mu = \mu_0 \frac{S}{K_s + S_a}\left(1 - \frac{P}{P_m}\right) \tag{41}$$

where, K_S $(= 10\ g/L)$is the Michaelis-Menten constant; P_m $(= 10\ g/L)$ is the inhibitory coefficient of alcohol on the cell growth; μ_0 (h^{-1}) is the maximum specific growth rate of yeast cells. The effect of culture temperature on the maximum specific growth rate of yeast can be described with the following empirical relationship:

$$\begin{aligned}\mu_0 = & -0.000049205 \ \times T^4 + 0.00569477 \ \times T^3 \\ & -0.24584 \ \times T^2 + 4.7132 \ \times T - 33.435\end{aligned} \tag{42}$$

The following ranges of state variables are adopted in this example:

$$1.5 \le V \le 5.0 \tag{43}$$

$$0 < C \le 15 \tag{44}$$

$$0.5 < S \le 80 \tag{45}$$

In addition, since the alcohol concentration of commercial beer is around 5 vol %, i.e., 40 g/L, the following constraint is imposed upon the product concentration:

$$P \ge 40 \tag{46}$$

On the other hand, the limits of the manipulated variables are imposed as follows:

$$0.01 \le F_{in}(t) \le 0.5 \tag{47}$$

$$0.05 \le F_{out}(t) \le 0.5 \tag{48}$$

Finally, the entire duration of operation is set to be 240 hours in this example, and the initial states are assumed to be: $S(0) = 4.5\ g/L$, $C(0) = 5\ g/L$, $P(0) = 50\ g/L$ and $V(0) = 1.5$ L. The nominal value of uncertain parameter $S_a(t)$ is fixed at 100 g/L with a possible deviation of 50 g/L in either positive or negative direction, while the nominal level of $T(t)$ is chosen to be 25 °C with a possible deviation of 10 °C in either direction.

In the present example, all possible combinations of the initial directions of uncertain parameters were tested. For the sake of brevity, Table 2 only shows the results for the case when both S_a and T deviate toward the positive direction initially. Also, the GA parameters were kept the same as those adopted in the previous example. From the results presented in Table 2, the dynamic flexibility index of the given system can be seen to increase with the increase in manipulated-variable instances. This trend stabilizes at a value of 0.490 when $N_z = 10$. Also presented in Figs. 7(a) - 7(d) and 8(a) – 8(d) are the time profiles of state variables, uncertain parameters and manipulated variables for the case $N_z = 10$. The dashed lines in the Figs. 8(a) and 8(b) signify the nominal levels of respective uncertain parameters.

For this particular case when $N_z = 10$, it can be observed that the inequality constraint imposed on alcohol concentration (P) touches lower bound (40 g/L) at six time instances (see Fig. 7(b) and Table 2); It can also be observed that the liquid volume (V) goes active on the upper bound (5 L) at the final time point (see Fig. 7(d) and Table 2); The inlet flowrate (F_{in}) can be found to be activated at the minimum value of 0.01 L/h in the time interval from 49 h to 96 h (Fig. 8(c) and Table 2), while the outlet flowrate (F_{out}) remains activated at the minimum level of 0.05 L/h throughout the entire time horizon (see Fig. 7(d) and Table 2).

Let us take a closer look at Table 2, Figs. 7(b), 8(a), 8(b), and 8(c) when $N_z = 10$. Notice that the alcohol concentration (P) touches the lower limit at 99 hours and this time instance is preceded by the corner shift from positive to negative direction in the uncertain parameter T around 70 hours. Notice also that the inlet flowrate (F_{in}) reaches the allowed minimum at 49 hours in response to the corner shift from positive to negative direction in uncertain parameter S_a around 40 hours.

6. Additional Discussions on Computation Results

First of all, it should be noted that the aforementioned constraint in (31) can be confirmed with the numerical results presented in the two examples. Secondly, as mentioned before in subsection 2.3, it was expected that the dynamic flexibility index (FI_d) would increase with the increase in the number of manipulated-variable pieces (N_z). This trend should continue until FI_d stabilizes and attains a fixed value which can be considered as the actual flexibility index of the given system. As can be seen in the second column of either Table 1 or Table 2, the numerical re-

Table 2
Dynamic flexibility index of an alcoholic fermentation system

Number of manipulated-variable pieces (N_z)	FI_d	Time points of activated constraints (hours)	Convergence Time (Seconds)
2	0.455	P: 207, 240; V: 240 F_{out}: 1-240	4507
4	0.460	P: 121, 122, 227, 228; V: 240 F_{out}: 1-240	5047
6	0.470	P: 108, 109, 167, 168; V: 240 F_{in}: 41-80; F_{out}: 1-240	6040
8	0.490	P: 128, 129, 174, 240; V: 240 F_{in}: 61-90; F_{out}: 1-240	6148
10	0.490	P: 99, 132, 147, 183, 184, 233; V: 240 F_{in}: 49-96; F_{out}: 1-240	6723

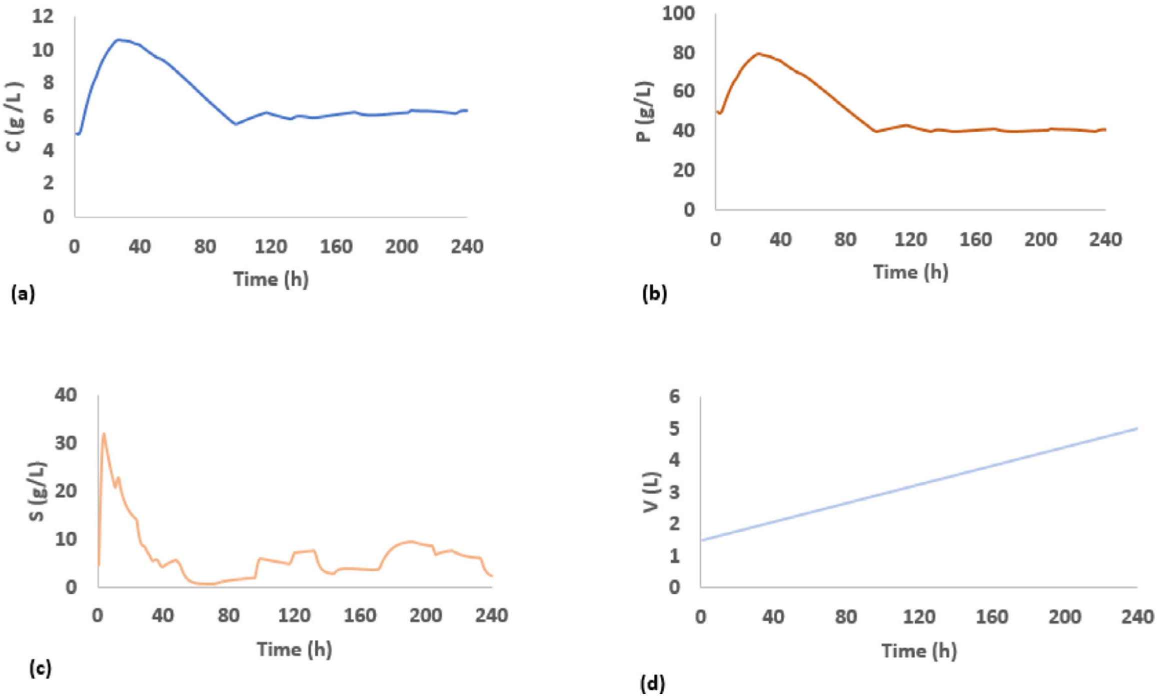

Fig. 7. Alcoholic fermentation tank state variable time profiles. (a) Yeast cell concentration, C; (b) Product alcohol concentration, P; (c) Glucose concentration, S; (d) Volume of liquid in tank, V

Table 3
A comparison of conventional and proposed vertex enumeration methods

	FI_d	Time points of activated constraints (Minutes)	Convergence time (Seconds)
Conventional vertex enumeration	0.157	h; 321, 800	17186
GA-assisted vertex enumeration	0.157	h; 322, 800	895

sults confirm the aforementioned assertion. The proposed method functions well for both examples and the finally saturated value of FI_d can be obtained in a few (3-5) iterations.

Finally, notice that the computation time needed in implementing the proposed method should be significantly less than that needed by its conventional counterpart. Let us consider the special case of $N_z = 1$ and two corner shifts in the buffer-tank example for illustration clarity. Table 3 below show a comparison between the convergence times required by the conventional vertex enumeration method and the proposed GA-assisted method. It can be observed from this table that, although the values of dynamic flexibility index and the time points of activated constraints in both cases are almost identical, the corresponding computation loads are drastically different. In particular, the conventional approach takes 171,186 seconds to converge, while the proposed approach only needs 895 seconds.

This stark difference in the convergence times may be explained by the difference in the number of iterations to reach optima. In the buffer-tank example presented in subsection 5.1, since 20 individuals were adopted in each population for every 100-generation GA run, 2000 iterations were performed before securing an optimum solution. As mentioned before in the Introduction section, if the conventional exhaustive vertex enumeration approach is applied according to (10) and (11) without additional limitation on the number of corner shifts, the number of vertexes should be $(2^{n_\theta})^{M+1}$. Since $n_\theta = 1$ and $\mathrm{M} = 800$ in this example, $2^{801} (= 1.33 \times 10^{241})$ iterations are required and the corresponding computation load is clearly overwhelming. Due to the fact that it is unrealistic to carry out the above calculation, an additional constraint of two corner shifts was introduced to generate the results in Table 3. Notice that the total number of vertexes with this added constraint should be $2 \times \binom{800}{2}$. Since $2^{801} = 2 \times \sum_{i=0}^{800} \binom{800}{i}$, it is

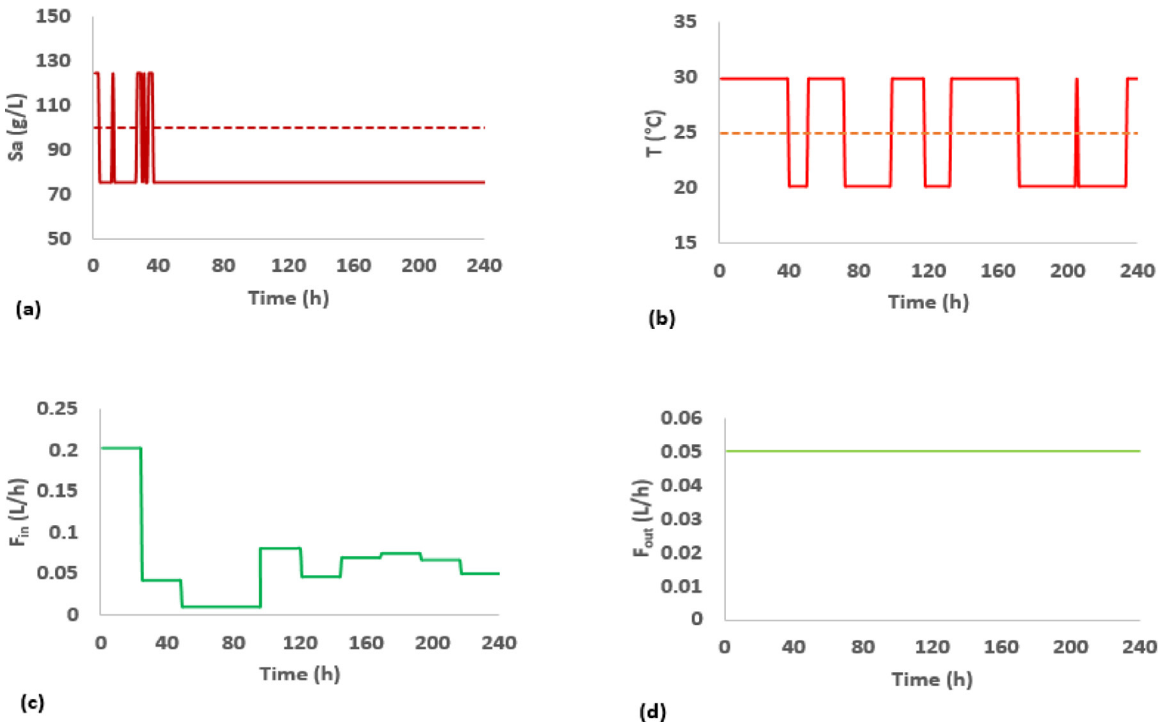

Fig. 8. Alcoholic fermentation tank uncertain parameter and manipulated variable time profiles. (a) Inlet glucose concentration, S_a; (b) Temperature of culture, T; (c) Inlet flow rate, F_{in}; (d) Outlet flow rate, F_{out}

obvious that $2^{800} \gg \binom{800}{2}$. Therefore, if the actual exhaustive vertex enumeration is to be implemented according to (10) and (11), the needed computation time must be much larger than that listed in Table 3.

In summary, the conventional vertex enumeration technique evidently would have convergence issues. On the other hand, with the proposed GA assisted vertex enumeration we could efficiently evaluate the dynamic flexibility index, in relatively much lesser time. Therefore, the proposed methodology could be regarded as significant in improving the practical applicability of flexibility index quantification of the dynamic process systems.

7. Conclusions

In this study, significant improvements have been introduced to modify the existing methods for *efficient* evaluation of flexibility index of the dynamic process systems. The proposed changes mainly include (1) classification of inequality constraints in terms of state and control variables and (2) treatment of the manipulated variables as piecewise-constant functions of time rather than continuous functions. Moreover, a metaheuristic search algorithm, namely genetic algorithm, has been incorporated to develop a generalized computation procedure for evaluating dynamic flexibility index. Two numerical examples have been included to demonstrate the effectiveness of the proposed methodology in terms of its assured convergence with reasonable accuracy and computation efficiency.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Discretization of Dynamic Model

A practically indispensable step for numerically computing FI_d is to discretize the constraints in Eqs. (3) and (4) using a credible numerical technique (Chang and Adi, 2018). Although numerous schemes are available, the simplest one, i.e., the trapezoidal rule, is presented here to facilitate a clear understanding. Let us first partition the horizon $[0, H]$ into M equal intervals with their end points denoted sequentially as $p = 1, 2, \cdots, M$. Thus, the length of each time interval should be H/M (denoted as Δt). By applying trapezoidal rule, one can approximate Eq. (3) as follows

$$x_i(t_p) - x_i(t_{p-1}) = \frac{\Delta t}{2}\Big[\varphi_i\big(\mathbf{d}, \boldsymbol{x}(t_{p-1}), \boldsymbol{z}(t_{p-1}), \boldsymbol{\theta}(t_{p-1})\big) + \varphi_i\big(\mathbf{d}, \boldsymbol{x}(t_p), \boldsymbol{z}(t_p), \boldsymbol{\theta}(t_p)\big)\Big] \tag{A1}$$

where, $x_i(t_0) = x_i(0) = x_i^0$, $i \in \mathrm{I}$ and $p = 1, 2, \cdots, M$. Similarly, Eq. (4) can also be discretized according to the aforementioned $M + 1$ boundary points as follows

$$g_j\big(\mathbf{d}, \boldsymbol{x}(t_p), \boldsymbol{z}(t_p), \boldsymbol{\theta}(t_p)\big) \le 0 \ \forall \mathrm{j} \in \mathrm{J} \tag{A2}$$

where, $p = 0, 1, 2, \cdots, M$.

Notice that, in order to apply the extended vertex method, it is necessary to treat every state variable, every manipulated variable and every uncertain parameter at each boundary point individually as distinct quantity. In other words, there should be $n_x M$ state variables, $n_z(M+1)$ manipulated variables and $n_\theta(M+1)$ uncertain parameters in the discretized model. Therefore, the total number of vertexes should be $(2^{n_\theta})^{M+1}$ and it is clear that the corresponding computation load can be overwhelming even for a moderately complex system.

On the other hand, if the extended active set method is to be implemented, the time-varying, slack variables, $s_j(t)$ and $j = 1, 2, \cdots, n_i$, the binary variables $y_j(t)$ and $j = 1, 2, \cdots, n_i$, along with the Lagrange multipliers, $\boldsymbol{\lambda}(t)$, $\mu_1(t)$ and $\boldsymbol{\mu}_2(t)$, also have to be discretized. It should be noted that $\boldsymbol{\lambda}(t) = [\ \lambda_1(t) \quad \lambda_2(t) \quad \cdots \quad \lambda_{n_i}(t)\ ]^T$ and

$\boldsymbol{\mu}_2(t) = [\ \mu_{2,1}(t) \quad \mu_{2,2}(t) \quad \cdots \quad \mu_{2,n_e}(t)\ ]^T$. In general, the total number of these variables is proportional to the number of time points (t_p) chosen within the operation horizon $[0, H]$. The total numbers of discretized slack and binary variables should both be $n_i(M+1)$, while those of the Lagrange multipliers, $\boldsymbol{\lambda}(t)$, $\mu_1(t)$ and $\boldsymbol{\mu}_2(t)$, should be $n_i(M+1)$, $M-1$ and n_eM, respectively. This significant increase in the total number of variables along with the fact that the constraints mentioned previously in Eqs. (13)-(24) now need to be satisfied at each of the time points, should drastically increase the computation load of the optimization problem described in subsection 2.3, and sometimes even leads to difficulties in convergence.

CRediT authorship contribution statement

Shoeb Moon Ali: Conceptualization, Methodology, Software, Writing – original draft. **Shang-Wei Hwang:** Methodology, Software. **Chuei-Tin Chang:** Supervision, Conceptualization, Visualization, Writing – review & editing. **Jo-Shu Chang:** Supervision, Writing – review & editing.

References

Chang, C.T., Adi, V.S.K., 2018. Deterministic Flexibility Analysis; Theory, Design and Applications. CRC Press, Taylor & Francis Group, Boca Raton, FL, USA ISBN: 978-1-4987-4816-2.

Dimitriadis, V.D., Pistikopoulos, E.N., 1995. Flexibility analysis of dynamic systems. Ind. Eng. Chem. Res. 34, 4451–4462.

Fabro, J.A., Arruda, L.V., 2003. Fuzzy-neuro predictive control tuned by genetic algorithms applied to a fermentation process. IEEE International Symposium on Intelligent Control 194–199.

Grossmann, I.E., Floudas, C.A., 1987. Active constraint strategy for flexibility analysis in chemical process. Comput. Chem. Eng. 11, 675–693.

Holland, J.H., 1992. Genetic Algorithms. Scientific American 267, 66–73.

Kuo, Y.C., Chang, C.T., 2016. On heuristic computation and application of flexibility indices for unsteady process design. Industrial & Engineering Chemistry Research 55 (3), 670–682.

Lima, F.V., Jia, Z., Ierapetritou, M., Georgakis, C., 2010. Similarities and differences between the concepts of operability and flexibility: the steady-state case. AIChE J 56, 702–716.

Serra, A., Strehaiano, P., Taillandier, P., 2005. Influence of temperature and pH on Saccharomyces bayanus var. uvarum growth; impact of a wine yeast interspecific hybridization on these parameters. International journal of food microbiology 104 (3), 257–265.

Swaney, R.E., Grossmann, I.E., 1985. An index for operational flexibility in chemical process design part I: formulation and theory. AIChE J 31, 621–630.

Swaney, R.E., Grossmann, I.E., 1985. An index for operational flexibility in chemical process design part II: formulation and theory. AIChE J 31, 631–641.

Wu, R.S., Chang, C.T., 2017. Development of mathematical programs for evaluating dynamic and temporal flexibility indices based on KKT conditions. Journal of the Taiwan Institute of Chemical Engineers 73, 86–92.

Zhou, H., Li, X.X., Qian, Y., Chen, Y., Kraslawski, A., 2009. Optimizing the initial conditions to improve the dynamic flexibility of batch processes. Ind. Eng. Chem. Res. 48, 6321–6326.