8

Fuzzy Multi-Objective Optimization for Metabolic Reaction Networks by Mixed-Integer Hybrid Differential Evolution

Feng-Sheng Wang and Wu-Hsiung Wu
Department of Chemical Engineering, National Chung Cheng University, Taiwan

8.1 Introduction

The complexity of biological systems means that it is hard to study their behavior using experimental data alone. With the integration of experimental data and computational methods, the study of complex biological systems is now feasible. Computational biology, part of systems biology, focuses on the development of mathematical tools and aims to provide a powerful foundation for addressing critical biology questions. The optimization of metabolic reaction networks is an up-to-date approach in biotechnology [1–5]. Much research has discussed the applications of model-based optimization strategies in analyzing and designing a metabolic reaction network. Most published articles have focused on the mathematical foundations of optimization approaches [6–10] and their applications to processes [11–14]. Most metabolic models are nonlinear due to the complexity of reaction kinetics. The indirect optimization method (IOM) converts a nonlinear kinetic model, which is in GMA or Michaelis–Menten format, into an S-system formulation to facilitate progress. The optimization problem for the S-system model can be solved by a linear programming (LP) solver with logarithmic transformation [6,8,9,15]. Stochastic optimization methods are

Multi-Objective Optimization in Chemical Engineering: Developments and Applications, First Edition.
Edited by Gade Pandu Rangaiah and Adrián Bonilla-Petriciolet.

used to obtain a global optimum of nonlinear models [16]. Mixed-integer linear programming (MILP) methods were successfully applied to the identification of optimal regulatory structure of *E. coli* and the computation of production rate after metabolic engineering [17, 18]. After the regulatory structure was found by the MILP optimization approach, the next question was what the minimum set of manipulated enzymes in a metabolic system is given considerations regarding strain viability and maximum flux or yield of a desired product. To answer this question, a multi-objective optimization problem (MOOP) can be formulated and solved using efficient multi-objective optimization methods.

Many methods can be used to obtain the Pareto front of MOOPs [19–21]. Their advantages and disadvantages have been discussed in several articles [19–21]. These methods were classified into two categories: generating methods and preference-based methods. By using the scalarization approach, generating methods convert an MOOP into a single-objective optimization problem (SOOP) with different factors, and solve the SOOP to find one optimal Pareto solution. A series of SOOP with various factors can be solved to obtain the Pareto front of original MOOP. Although fitness evaluation in evolutionary algorithms is very time consuming, it is easy to implement for the identification of the Pareto front of MOOPs. After the Pareto front has been obtained, a solution can be selected from the Pareto front by decision makers (DM). In contrast, preference-based methods require the DM to give preferences in advance and then try to find a solution that satisfies the preference constraints. In general, it is difficult to specify preferences in advance without any information on the values of objective functions. An interactive algorithm is therefore necessary to find a compromise solution.

Experiments have shown that a strain may reveal resilience phenomena in response to environmental pressure and genetic perturbations [22, 23]. A mutant strain may respond to genetic perturbations with rapid and dramatic alterations in the distribution of metabolic fluxes. However, the mutant finally adapts to a new steady state that is only slightly different from that before the perturbation. The resilience phenomenon indicates that a mutant strain tries to recover from its original "wild-type" characteristics. To find the optimal modulation strategy that will result in a mutant that has survived, it is necessary to consider qualitative effects on metabolic reaction networks, for example the resilience phenomenon and cell viability constraints. In the practical optimization of metabolic reaction networks, designers have to manage the nature of uncertainty resulting from the qualitative character of metabolic reactions. Different types of uncertainty, such as imprecision, credibility, preference, possibility, and necessity, are described by different methods. For example, the imprecision of experimental data can be described by interval arithmetic, and the possibility of enzyme effects can be represented using probability methods. A deterministic approach does not give an adequate representation for metabolic reaction networks with uncertain characters. Fuzzy optimization formulations can be applied to cope with this problem, and the most widely used method for handling uncertainties is the fuzzy set method because of its generality and flexibility [24–26].

This chapter introduces a generalized fuzzy multi-objective optimization problem (GFMOOP) for finding the optimal engineering interventions on metabolic network systems considering the resilience phenomenon. This approach first formulates a constrained MOOP, that considers the resilience effects and minimum set of manipulated enzymes simultaneously by combining the concepts of minimization of metabolic adjustment (MOMA)

[22] and regulatory on/off minimization (ROOM) [23] into an optimization framework. In addition, nonlinear kinetic equations were directly applied to the optimization formulation, so it was formulated as a constrained mixed-integer nonlinear programming (MINLP) problem. In practice, this type of MINLP problems is highly nonlinear and nondifferentiable due to the discontinuous property of integer-valued variables. Most of the well known methods to solve MINLP optimization problems, including the cutting plane, branch-and-bound, and decomposition approaches, heavily depend on a good starting point and gradient information to yield an optimal solution. To overcome this drawback, many evolutionary algorithms (EAs) were developed and have been applied successfully to many practical problems. The mixed-integer hybrid differential evolution (MIHDE), a population-based evolution algorithm, was proposed to solve unconstrained MINLP optimization problems [27, 28]. It was extended to solve constrained MINLP problems through the implementation of constraint-handling techniques [29]. The MIHDE has been implemented as an optimization tool and will be used to solve GFMOOPs for the identification of optimal genetic manipulation strategies on metabolic reaction networks.

8.2 Problem Formulation

8.2.1 Primal Multi-Objective Optimization Problem

The dynamics of a metabolic reaction network can be represented generically using a set of nonlinear differential equations with the following structure:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{S}\mathbf{v}(\mathbf{x}, \mathbf{e}; \theta)$$

where $\mathbf{x} \in \mathrm{R}^n$ is a vector of concentrations of metabolites or pools, $\mathbf{e} \in \mathrm{R}^m$ is a vector of enzyme levels that correspond to the enzyme activities, $\theta \in \mathrm{R}^p$ is a vector of system parameters, $\mathbf{S} \in \mathrm{R}^{n \times m}$ is the stoichiometric matrix describing the interconnecting fluxes, and $\mathbf{v} \in \mathrm{R}^m$ is a vector of reaction rates. In the field of biological systems each reaction rate can be expressed by the power-law functions or Michaelis–Menten-based rate laws. To analyze the full dynamic behavior of a metabolic reaction network, its nonlinear differential equations have to be solved and evaluated numerically. It is time consuming when the nonlinear differential equations are complex. This problem can be addressed based on the assumption of a pseudo steady state. In general, most natural biological systems operate close to a steady state. Even in a disease state, a metabolic reaction network is still in a steady state with some of the steady-state concentrations different from normal [30]. In the following, we discuss the optimization problems for metabolic reaction networks at a steady state. All derivatives must be zero for a steady state, which implies

$$\mathbf{S}\mathbf{v}(\mathbf{x}, \mathbf{e}; \theta) = \mathbf{0} \tag{8.1}$$

Multi-objective optimization for a metabolic reaction network aims to determine the minimum set of allowable manipulation enzymes and the corresponding changes of enzyme activities and internal metabolite concentrations, so that the synthesis rates of the desired

products are maximized in steady state. The objective functions for this MOOP are generally expressed as follows:

$$\max_{\mathbf{e},\mathbf{x},\mathbf{y}} \frac{v_i}{v_i^{basal}},\ i \in \Sigma_O \tag{8.2}$$

$$\min_{\mathbf{e},\mathbf{x},\mathbf{y}} \sum_{j=1}^{m} y_j \tag{8.3}$$

where v_i^{basal} is the basal value of the ith flux $v_i \in \mathbf{v}$, $\Sigma_O \in \mathrm{N}^r$ is the set of indices of synthesis rates to be maximized, r is the number of target fluxes to be maximized, the binary variable $y_j \in \mathbf{y}$ is used to indicate whether the jth enzyme should be modulated, and is defined as follows:

$$y_j = \begin{cases} 1, & \text{if enzyme } j \text{ is modulated} \\ 0, & \text{otherwise} \end{cases}$$

Equation (8.2) is a general formulation for maximizing a set of metabolite synthesis rates simultaneously. Several researchers have introduced genetic manipulations to redistribute various metabolic fluxes in a metabolic network to enhance the desired synthesis rates [18, 31]. Equation (8.3) is used to obtain the minimum set of modulated enzymes in the metabolic reaction network.

Some additional constraints should be considered in metabolic reaction networks to obtain realizable solutions. The concentration of each enzyme should be bounded and is expressed as the following inequality constraints:

$$y_i e_i^{LB} + (1 - y_i)\, e_i^{basal} \le e_i \le y_i e_i^{UB} + (1 - y_i)\, e_i^{basal},\ i = 1, \ldots, m \tag{8.4}$$

$$\sum_{j=1}^{m} y_j \ge 1 \tag{8.5}$$

where e_i^{basal} is the basal value for the modulated enzyme i, $e_i^{LB} = \gamma_{e_i}^{LB} e_i^{basal}$ and $e_i^{UB} = \gamma_{e_i}^{UB} e_i^{basal}$ are the lower and upper bounds for each modulated enzyme, respectively. The lower bounded factor $\gamma_{e_j}^{LB}$ is less than 1 and the upper bounded factor $\gamma_{e_j}^{UB}$ is greater than 1. Both factors should be provided by the designer in advance. The concentration for each metabolite is practically restricted by its lower and upper bounds:

$$\gamma_{x_i}^{LB} x_i^{basal} \le x_i \le \gamma_{x_i}^{UB} x_i^{Basal},\ i = 1, \ldots, n \tag{8.6}$$

where $\gamma_{x_i}^{LB}$ and $\gamma_{x_i}^{UB}$ are the lower and upper bounded factors for each metabolite, respectively.

An abnormally high protein or intermediate concentration in a metabolic system renders a cell non-viable. This is because the burden on cellular metabolism is too high for the cell to survive or the cellular osmolarity constraint is violated. Several researchers have introduced constraints on the total enzyme concentration to overcome this issue and ensure that it never reaches an unacceptable value for the cell's viability [8,11,12,15]. Moreover, the cell viability and optimal synthesis rates effectively limit the total intermediate

metabolite concentration. The total metabolite and enzyme concentration constraints for cell viability are expressed as follows:

$$\sum_{i=1}^{n} x_i \leq \gamma_x \sum_{i=1}^{n} x_i^{basal} \tag{8.7}$$

$$\sum_{i=1}^{m} e_i \leq \gamma_e \sum_{i=1}^{m} e_i^{basal} \tag{8.8}$$

where γ_x and γ_e are the restriction factors for the constraints on total metabolite concentrations and total enzyme concentrations, respectively.

The primal multi-objective optimization problem formulated by Equations (8.1) to (8.8) is a multi-objective mixed-integer nonlinear programming problem. The objective function defined by Equation (8.3) can be straightforwardly converted into an ε-constraint, because it is easy to assign the ε-values. After the transformation, the primal MOOP is expressed as follows:

$$\max_{\mathbf{e},\mathbf{x},\mathbf{y}\in\Omega} \frac{v_i}{v_i^{basal}}, \ i \in \Sigma_O \tag{8.9}$$

subject to

$$\sum_{j=1}^{m} y_j \leq \varepsilon \tag{8.10}$$

where the allowable number ε of the manipulated enzymes is provided by the designer in advance, and the feasible set Ω consists of all feasible solutions that satisfy the constraints in Equations (8.1) and (8.4)–(8.8). The primal MOOP described by Equations (8.9) and (8.10) is still a multi-objective problem. The optimality of this transformation will be discussed in next section. If the lower and upper bounds for each objective value can be obtained in advance, the weighted infinite norm method, one of the reference-goal methods, can be used to determine a tradeoff solution. Based on this assumption, the MOOP defined by Equations (8.9) and (8.10) can be transformed into a weighted infinite-norm problem defined as follows:

$$\min_{\mathbf{e},\mathbf{x},\mathbf{y}\in\Omega^{\varepsilon}} \max_{i\in\Sigma_O} \left\{ \frac{v_i}{v_i^{UB} - v_i^{LB}} \right\} \tag{8.11}$$

where the lower bound v_i^{LB} is equal to its basal flux v_i^{basal}, the upper bound v_i^{UB} can be estimated by SOOP that maximizing v_i only, and the extended feasible set Ω^{ε} consists of the original feasible set Ω and the ε-constraint in Equation (8.10).

8.2.2 Resilience Problem

A strain may reflect resilience phenomena after genetic interventions. Segrè *et al.* [22] introduced a method, called MOMA, to obtain the flux distribution for a mutant strain by minimizing the distance between the flux vectors for the mutant strain and the original wild-type strain. Shlomi *et al.* [23] applied ROOM to calculate the flux distribution of a mutant with minimum number of component changes between the mutant flux vector and the original flux vector. However, the solutions obtained from both of these approaches are

based on the stoichiometric model and derive from the flux balance analysis (FBA). Lee *et al.* [32] present an MOMA kinetic model for the analysis of metabolic control mechanisms of transgenic plants on monolignol biosynthesis. Here, we introduce a generalized fuzzy multi-objective optimization formulation with a kinetic model to cope with the resilience effects and to minimize the enzyme manipulation set. The formulation combined both concepts of MOMA and ROOM with the primal MOOP. The primal MOOP is therefore extended to be a GFMOOP that can be expressed as follows:

$$\widetilde{\max_{\mathbf{e},\mathbf{x},\mathbf{y}}} \frac{v_i}{v_i^{basal}} = \widetilde{\max_{\mathbf{e},\mathbf{x},\mathbf{y}}} f_i \succsim \left[f_i^{LB}, f_i^{UB}\right],\ i \in \Sigma_O \tag{8.12}$$

$$\widetilde{\underset{\mathbf{e},\mathbf{x},\mathbf{y}}{\text{equal}}} \left(x_j \approx x_j^{basal}\right),\ j \in \Sigma_X \tag{8.13}$$

$$\widetilde{\underset{\mathbf{e},\mathbf{x},\mathbf{y}}{\text{equal}}} \left(e_k \approx e_k^{basal}\right),\ k \in \Sigma_E \tag{8.14}$$

$$\min_{\mathbf{e},\mathbf{x},\mathbf{y}} \sum_{j=1}^{m} y_j \tag{8.15}$$

where $\Sigma_X \in \mathrm{N}^n$ is the set of metabolite indices and $\Sigma_E \in \mathrm{N}^m$ is the set of enzyme indices. Here, the symbols, "$\succsim$" and "$\approx$" denote a relaxed or fuzzy version of the ordinary inequality "$\geq$" and equality "$=$", respectively. The fuzzy maximization, "$\widetilde{\max}$", in Equation (8.12) means that the enzyme manipulation is completely acceptable if the *i*th flux ratio exceeds its upper bound f_i^{UB}, which can be estimated from the primal MOOP. Conversely, the design is completely unacceptable if the *i*th flux ratio is less than the lower bound f_i^{LB}. The lower bound is generally equal to 1, meaning that the modified flux should exceed its basal value. Equations (8.13) and (8.14) are the "fuzzy equal ($\widetilde{\text{equal}}$)" objective functions that represent the fuzzy goals. For example, the metabolite concentration x_j and enzyme activity e_k should be restored to a state that is as close to the wild type as possible. Equation (8.15) is the crisp objective function, the same as Equation (8.3).

The cell viability constraints in Equations (8.7) and (8.8) are crisp limits, indicating that all cells die when any of the inequality constraints is violated. These constraints are not so strict in practical situations according to the growth patterns and kinetics of cells in culture [33]. In general, cells can survive when each total amount of metabolites and enzymes is within a wide interval over the critical value. The fuzzy inequality constraint can be applied to handle this practical situation. The restrictions for cell viability are softened as follows:

$$\sum_{i=1}^{n} x_i \precsim \left[\zeta_x^{LB}, \zeta_e^{UB}\right] \sum_{i=1}^{n} x_i^{basal} \tag{8.16}$$

$$\sum_{i=1}^{m} e_i \precsim \left[\zeta_e^{LB}, \zeta_e^{UB}\right] \sum_{i=1}^{m} e_i^{basal} \tag{8.17}$$

where the symbol "$\precsim$" denotes a fuzzy version of the ordinary inequality "$\leq$". Here, $\zeta_{x/e}^{LB}$ and $\zeta_{x/e}^{UB}$ are the lower and upper restriction factors, respectively, for the fuzzy constraints on total metabolite/enzyme concentrations. The interval bound $[\zeta_{x/e}^{LB}, \zeta_{x/e}^{UB}]$ indicates that the microbes have some degree of satisfaction if each metabolite/enzyme concentration is

within its boundary. The lower bounds of the fuzzy inequality constraints mean that the microbes completely survived if both total metabolite/enzyme concentration constraints in Equations (8.16) and (8.17) are less than their lower limits. Conversely, the microbes died if one of the total metabolite/enzyme concentration constraints exceeded its upper limit. This situation indicates that the solution is infeasible.

The objective functions of GFMOOP are defined in the fuzzy and crisp domains. This is the reason why we call it a general fuzzy MOOP. Almost no studies discuss how to obtain a Pareto-optimal solution of the GFMOOP. This problem is first converted into a fuzzy MOOP with ε-constraints, abbreviated as ε-FMOOP, by transforming the crisp objective function into an ε-constraint. The optimality of the ε-FMOOP problem will be discussed in next section. To solve the ε-FMOOP, each fuzzy objective function, fuzzy equal objective function, and fuzzy inequality constraint function was quantified by eliciting their corresponding membership function. Sakawa [20] proposed five types of membership functions: linear, exponential, hyperbolic, inverse, and piecewise linear functions to evaluate the membership grades. Having elicited the membership functions for each fuzzy objective function, fuzzy equal objective, and fuzzy inequality constraints, the ε-FMOOP can be expressed as the goal attainment problem:

$$\min_{\mathbf{e},\mathbf{x},\mathbf{y}} \eta_D = \min_{\mathbf{e},\mathbf{x},\mathbf{y}} \left\{ \max_{i\in\Sigma} [\bar{\eta}_i - \eta_i(f_i)] + \delta \sum_{i\in\Sigma} [\bar{\eta}_i - \eta_i(f_i)] \right\} \tag{8.18}$$

where $\bar{\eta}_i$ is the ideal preferred goal, $\Sigma = \Sigma_O \cup \Sigma_X \cup \Sigma_E$, and η_D denotes an aggregation function defined on the crisp domain Ω, which consists of the feasible solutions satisfied equation (8.1), the crisp bounds in equations (8.4)–(8.5) and the ε-constraint in equation (8.10). Sakawa introduced several aggregation functions in which the value of the aggregation function can be interpreted as an overall degree of satisfaction with user's fuzzy goals [20]. Here, the first term of the aggregation function in the brace of Equation (8.18) is applied to determine the optimal tradeoff solution that is nearest to the ideal preferred goal, $\bar{\eta}_i$, which indicates 100% satisfaction. The second term is introduced to avoid uniqueness testing for optimality of the solution and the constant δ is a small positive value between 10^{-3}–10^{-5}. The fuzzy goal attainment approach can find a satisfactory solution directly in the Pareto set without yielding the Pareto frontier of the problem.

8.3 Optimality

Here, we discuss the optimality of the primal MOOP and GFMOOP. A general MOOP problem can be rewritten in a compact form as follows:

MOOP:

$$\min_{\mathbf{u}\in\Omega} f_i(\mathbf{u}),\ i \in \Sigma \tag{8.19}$$

where $\Sigma \in N^r$ is a set of indices of objective functions, r is the number of objective functions, f_i is the ith objective function, the vector $\mathbf{u}$, an n-dimensional mixed-integer vector, consists of an n_I-dimensional vector of integer variables and an n_C-dimensional vector of continuous variables, and $n = n_I + n_C$. The feasible domain Ω consists of all

feasible solutions that satisfy the equality and inequality constraints and can be expressed as the following compact form:

$$\Omega = \left\{ \mathbf{u} \in \mathrm{Z}^{n_I} \times \mathrm{R}^{n_C} | h_i(\mathbf{u}) = 0, i = 1, \ldots, m; g_j(\mathbf{u}) \leq 0, \;\; j = 1, \ldots, p \right\} \quad (8.20)$$

where m and p are the number of equality constraints h and inequality constraints g, respectively. Without loss of generality, any maximization of objective functions can be transformed to a minimization form by changing the sign of the objective functions. Therefore, any MOOP can be expressed as Equation (8.19).

The MOOP is a natural extension of a traditional optimization of a single-objective function. If the multiple-objective functions are commensurate, minimizing one objective function will minimize all objectives and the problem can be solved using traditional optimization techniques to obtain a complete optimal solution. However, if the objective functions are incommensurate, or competing, then the minimization of one objective function requires a compromise in another objective function. The competition between multiple-objective functions gives rise to the distinction between the MOOP and a traditional SOOP. The problem is further complicated by the lack of a complete priority order for multiple objectives. Therefore, the concept of Pareto optimality or noninferiority is used to characterize an optimal solution to MOOPs. In order to explain the Pareto-optimal solution concisely, we introduce the following definition [19–21].

Definition 1: Pareto-optimal solution

A vector $\mathbf{u}^*$ is said to be a Pareto-optimal solution of the MOOP, if and only if there does not exist another $\mathbf{u} \in \Omega$ such that $f_i(\mathbf{u}) \leq f_i(\mathbf{u}^*)$ for all i and $f_j(\mathbf{u}) \neq f_j(\mathbf{u}^*)$ for at least one j.

The ε-constraint method finds the Pareto-optimal solutions of an MOOP by transferring it into a traditional SOOP. One of the objective functions in the MOOP is selected as the objective function of the SOOP and the others are transferred into inequality constraints. By this transformation, the ε-constraint formulation for the general MOOP is expressed as follows:

$$\begin{aligned} \min_{\mathbf{u}} \quad & f_i(\mathbf{u}) \\ \text{subject to} \quad & f_j(\mathbf{u}) \leq \varepsilon_j, \; j \in \Sigma; i \neq j \\ & \mathbf{u} \in \Omega \end{aligned} \quad (8.21)$$

Sakawa and Sawaragi *et al.* [20,21] have demonstrated that a Pareto-optimal solution of the general MOOP problem is also an optimal solution of its corresponding ε-constraint formulation for a specific objective function and some ε_j, and vice versa.

Like the ε-constraint method, the weighted min-max method determines the Pareto-optimal solutions of an MOOP by transferring it into a weighted min-max formulation that is expressed as follows:

$$\min_{\mathbf{u} \in \Omega} \max_{i \in \Sigma} \{ w_i f_i(\mathbf{u}) \} \quad (8.22)$$

where the weighting factor w_i is used to normalize each corresponding objective function f_i and can be defined as $1/(f_i^{UB} - f_i^{LB})$. The lower and upper bounds, f_i^{LB} and f_i^{UB}, are assigned by the user in advance. Sakawa and Sawaragi *et al.* [20,21] have also demonstrated that a Pareto-optimal solution of the general MOOP problem is also an optimal solution of its corresponding weighted min-max formulation for some weighting factors, and vice

versa. Many tools, including MIHDE, can be used to solve the ε-constraint and weighted min-max problems transferred from the general MOOP.

In traditional ε-constraint methods, only one of the objective functions is retained as the criterion and the others are converted into inequality constraints. In the following we discuss a hypothetical condition that some objective functions of the general MOOP can be easily transformed into ε-constraints, but others cannot. As a result, the transformed problem is still an MOOP, but its feasible domain is changed. Without loss of generality, this partial transformed MOOP, we call it ε-MOOP, from the general MOOP can be expressed as follows:

ε-MOOP:

$$\min_{\mathbf{u}\in\Omega^{\varepsilon}} f_i(\mathbf{u}),\ i \in \Sigma_I, \Omega^{\varepsilon} = \Omega \cap \{\mathbf{u} | f_j(\mathbf{u}) \le \varepsilon_j, j \in \Sigma_J\} \tag{8.23}$$

where $\Sigma = \Sigma_I \cup \Sigma_J$ and $\Sigma_I \cap \Sigma_J = \emptyset$ (empty set).

Here we prove that the general MOOP and its corresponding ε-MOOP problem with an extended feasible domain have the same optimality.

Lemma 1 If $\mathbf{u}^* \in \Omega^{\varepsilon}$ is a Pareto-optimal solution of the ε-MOOP problem with objective functions f_i, i $\in \Sigma_I$ and some ε_j, $j \in \Sigma_J$, then $\mathbf{u}^*$ is a Pareto-optimal solution of the corresponding MOOP.

Proof. Assume $\mathbf{u}^* \in \Omega^{\varepsilon}$ is a Pareto-optimal solution of the ε-MOOP problem with objective functions f_i, $i \in \Sigma_I$ and some ε_j, $j \in \Sigma_J$, but it is not a Pareto-optimal solution of the corresponding MOOP. According to the definition of a Pareto-optimal solution, there exists at least a $\mathbf{u} \in \Omega$ such that $f_k(\mathbf{u}) \le f_k(\mathbf{u}^*)$ for all k $\in \Sigma$ and $f_l(\mathbf{u}) \ne f_l(\mathbf{u}^*)$ for at least one $l \in \Sigma$. Without loss of generality, let $l \in \Sigma_I$ and $f_k(\mathbf{u}^*) = \varepsilon_k$ for all $k \in \Sigma_J$, then $\mathbf{u} \in \Omega$ is a feasible solution such that $f_j(\mathbf{u}) \le \varepsilon_j$ for all $j \in \Sigma_J$, $f_i(\mathbf{u}) \le f_i(\mathbf{u}^*)$ for all $i \in \Sigma_I$, and $f_l(\mathbf{u}) \ne f_l(\mathbf{u}^*)$ for at least one $l \in \Sigma_I$. This contradicts the fact that $\mathbf{u}^*$ is a Pareto-optimal solution of the ε-MOOP problem. So $\mathbf{u}^*$ is a Pareto-optimal solution of the corresponding MOOP problem.

Lemma 2 If $\mathbf{u}^* \in \Omega$ is a Pareto-optimal solution of the MOOP, then $\mathbf{u}^*$ is a Pareto-optimal solution to the corresponding ε-MOOP for some ε_j, $j \in \Sigma_J$.

Proof. Assume $\mathbf{u}^* \in \Omega$ is a Pareto-optimal solution of the MOOP, but it is not a Pareto-optimal solution of the ε-MOOP. According to the definition of a Pareto-optimal solution, there exists at least a $\mathbf{u} \in \Omega^{\varepsilon} \subset \Omega$ such that $f_i(\mathbf{u}) < f_i(\mathbf{u}^*)$ for some $i \in \Sigma_I$, $f_k(\mathbf{u}) \le f_k(\mathbf{u}^*)$ for all $k \in \Sigma_I$ and $i \ne k$, and $f_j(\mathbf{u}) \le \varepsilon_j = f_j(\mathbf{u}^*)$ for all $j \in \Sigma_J$. Here $f_j(\mathbf{u}^*)$ is an upper bound of the objective functions f_j for all $j \in \Sigma_J$ and can be used as the value of ε_j. Therefore, there exists at least a $\mathbf{u} \in \Omega$ such that $f_i(\mathbf{u}) < f_i(\mathbf{u}^*)$ for some $i \in \Sigma_I$, $f_k(\mathbf{u}) \le f_k(\mathbf{u}^*)$ for all k $\in \Sigma$ and $i \ne k$, and $f_j(\mathbf{u}) \le f_j(\mathbf{u}^*)$ for all $j \in \Sigma_J$. This contradicts the fact that $\mathbf{u}^*$ is a Pareto-optimal solution of the MOOP. We conclude that $\mathbf{u}^*$ is also a Pareto-optimal solution of the corresponding ε-MOOP for some specific ε_j, $j \in \Sigma_J$.

When the primal MOOP described by equations (8.1) to (8.8) is solved, it is transferred into an ε-MOOP problem by converting the objective function in Equation (8.3) into an inequality constraint. By lemmas 1 and 2, the primal MOOP and its corresponding ε-MOOP have the same optimality. The ε-MOOP with an extended feasible domain is a

specific case of the general MOOP. A Pareto-optimal solution of the ε-MOOP problem with an extended feasible domain can be obtained using the ε-constraint method or weighted min-max method according to the demonstration from Sakawa and Sawaragi *et al.* So we have the following remark.

Remark 1. A Pareto-optimal solution of the primal MOOP can be obtained by solving its corresponding ε-constraint problem or weighted min-max problem with an extended feasible domain.

If the designer considers the fuzzy metabolite adjustment issue and has the preference goals in advance, he or she can apply a fuzzy optimization method to achieve the goals. Such a fuzzy multi-objective optimization problem (FMOOP) can be formulated as follows:

FMOOP:

$$\begin{aligned} &\widetilde{\min_{\mathbf{u}\in\Omega}} f_i(\mathbf{u}),\ i \in \Sigma_O \\ &\widetilde{\underset{\mathbf{u}\in\Omega}{equal}} f_j(\mathbf{u}),\ j \in \Sigma_E \end{aligned} \tag{8.24}$$

where Σ_O is the set of indices of minimizing objectives and Σ_E is the set of indices of equal objectives. When the membership functions $\eta_i(f_i(\mathbf{u}))$ for each objective function has been determined, the FMOOP can be converted into the goal-attainment or min-max problem. The goal-attainment problem (GAP) is defined as follows:

$$\min_{\mathbf{u}\in\Omega} \max_{i\in\Sigma} \left\{\left[\bar{\eta}_i - \eta_i\left(f_i(\mathbf{u})\right)\right], i \in \Sigma\right\} \tag{8.25}$$

where $\bar{\eta}_i$ is the reference value of membership function f_i, $\Sigma = \Sigma_O \cup \Sigma_E$, and $\Sigma_O \cap \Sigma_E = \emptyset$.

Sakawa [20] defines the following definition for explanations of the optimality relationship between the FMOOP and the GAP problem.

Definition 2: M-Pareto-optimal solution

The vector $\mathbf{u}^* \in \Omega$ is said to be an M-Pareto-optimal solution of the FMOOP if and only if there does not exist another $\mathbf{u} \in \Omega$ such that $\eta_i(f_i(\mathbf{u})) \geq \eta_i(f_i(\mathbf{u}^*))$ for all $i \in \Sigma$ and $\eta_j(f_j(\mathbf{u})) \neq \eta_j(f_j(\mathbf{u}^*))$ for at least one j.

Sakawa [20] has demonstrated that a local M-Pareto-optimal solution of the FMOOP is also a unique optimal solution to the GAP for some reference membership levels.

Let us consider the GFMOOP as discussed in equations (8.12)–(8.15), i.e., the problem includes fuzzy and crisp objective functions, and is therefore rewritten as follows:

GFMOOP:

$$\begin{aligned} &\widetilde{\min_{\mathbf{u}\in\Omega}} f_i(\mathbf{u}),\ i \in \Sigma_O \\ &\widetilde{\underset{\mathbf{u}\in\Omega}{equal}} f_j(\mathbf{u}),\ j \in \Sigma_E \\ &\min_{\mathbf{u}\in\Omega} f_k(\mathbf{u}),\ k \in \Sigma_C \end{aligned} \tag{8.26}$$

where Σ_C is the set of indices of crisp objective functions f_k. Like the primal MOOP, the GFMOOP is converted to an FMOOP by transferring each crisp objective function into an ε-constraint. This transferred FMOOP with ε-constraint, abbreviated as ε-FMOOP, is an

FMOOP defined in domain $\Omega^{\varepsilon} = \Omega \cap \{\mathbf{u} | f_k(\mathbf{u}) \leq \varepsilon_k, \mathrm{k} \in \Sigma_C\}$. In the following, we define the generalized M-Pareto-optimal solution of GFMOOP and prove the following lemmas.

Definition 3: Generalized M-Pareto-optimal solution

The vector $\mathbf{u}^* \in \Omega$ is said to be a generalized M-Pareto-optimal solution of the GFMOOP, if and only if there does not exist another vector $\mathbf{u} \in \Omega$ such that $\eta_i(f_i(\mathbf{u})) \geq \eta_i(f_i(\mathbf{u}^*))$ for all $i \in \Sigma_O \cup \Sigma_E, \eta_j(f_j(\mathbf{u})) \neq \eta_j(f_j(\mathbf{u}^*))$ for at least one $j \in \Sigma_O \cup \Sigma_E$, and $f_k(\mathbf{u}) \leq f_k(\mathbf{u}^*)$ for all $k \in \Sigma_C$.

Lemma 3 If $\mathbf{u}^* \in \Omega^{\varepsilon}$ is an M-Pareto-optimal solution to the ε-FMOOP for some ε_k, $\mathrm{k} \in \Sigma_C$, then $\mathbf{u}^*$ is a generalized M-Pareto-optimal solution to the GFMOOP.

Proof. Assume $\mathbf{u}^* \in \Omega^{\varepsilon}$ is an M-Pareto-optimal solution of the ε-FMOOP problem with fuzzy objective functions $f_i, i \in \Sigma_O \cup \Sigma_E$ and some ε_j, $j \in \Sigma_C$, but it is not a generalized M-Pareto-optimal solution to the GFMOOP. According to the definition of a generalized M-Pareto-optimal solution, there exists at least a $\mathbf{u} \in \Omega$ such that $\eta_k(f_k(\mathbf{u})) \geq \eta_k(f_k(\mathbf{u}^*))$ for all $k \in \Sigma_O \cup \Sigma_E$, $\eta_l(f_l(\mathbf{u})) \neq \eta_l(f_l(\mathbf{u}^*))$ for at least one $l \in \Sigma_O \cup \Sigma_E$, and $f_m(\mathbf{u}) \leq f_m(\mathbf{u}^*)$ for all $m \in \Sigma_C$. Without loss of generality, let $f_k(\mathbf{u}^*) = \varepsilon_k$ for all $k \in \Sigma_C$, then $\mathbf{u} \in \Omega$ is a feasible solution such that $f_k(\mathbf{u}) \leq \varepsilon_k$ for all $k \in \Sigma_C$, $\eta_i(f_i(\mathbf{u})) \geq \eta_i(f_i(\mathbf{u}^*))$ for all $i \in \Sigma_O \cup \Sigma_E$, and $\eta_l(f_l(\mathbf{u})) \neq \eta_l(f_l(\mathbf{u}^*))$ for at least one $l \in \Sigma_O \cup \Sigma_E$. This contradicts the fact that $\mathbf{u}^*$ is an M-Pareto-optimal solution of the ε-FMOOP problem, so $\mathbf{u}^*$ is a generalized M-Pareto-optimal solution to the GFMOOP.

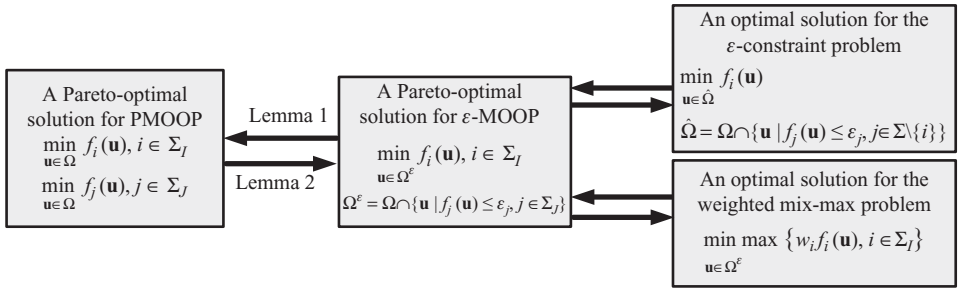
Lemma 4 If $\mathbf{u}^* \in \Omega$ is a generalized M-Pareto-optimal solution to the GFMOOP with $0 < \eta_i(f_i(\mathbf{u}^*)) < 1$ for all $i \in \Sigma_O \cup \Sigma_E$, and there exists $\varepsilon_k, k \in \Sigma_C$, such that $f_k(\mathbf{u}^*) \leq \varepsilon_k$, $k \in \Sigma_C$, then $\mathbf{u}^*$ is an M-Pareto-optimal solution to the ε-FMOOP.

Proof. Assume $\mathbf{u}^* \in \Omega$ is a generalized M-Pareto-optimal solution of the GFMOOP, but it is not an M-Pareto-optimal solution of the ε-FMOOP for some $\varepsilon_k, k \in \Sigma_C$. According to the definition of generalized M-Pareto-optimal solution of the ε-FMOOP, there exists at least a $\mathbf{u} \in \Omega^{\varepsilon} \subset \Omega$ such that $\eta_i(f_i(\mathbf{u})) \geq \eta_i(f_i(\mathbf{u}^*))$ for all $i \in \Sigma_O \cup \Sigma_E$, $\eta_j(f_j(\mathbf{u})) \neq \eta_j(f_j(\mathbf{u}^*))$ for at least one $j \in \Sigma_O \cup \Sigma_E$, and $f_k(\mathbf{u}) \leq \varepsilon_k$, $k \in \Sigma_C$. Since $\mathbf{u}^*$ is a feasible solution of GFMOOP, $f_k(\mathbf{u}^*)$ is an upper bound of objective function f_k for all $k \in \Sigma_C$ and can be used as the value of ε_k. Therefore, there exists at least a $\mathbf{u} \in \Omega$ such that $\eta_i(f_i(\mathbf{u})) \geq \eta_i(f_i(\mathbf{u}^*))$ for all $i \in \Sigma_O \cup \Sigma_E$, $\eta_j(f_j(\mathbf{u})) \neq \eta_j(f_j(\mathbf{u}^*))$ for at least one $j \in \Sigma_O \cup \Sigma_E$, and $f_k(\mathbf{u}) \leq f_k(\mathbf{u}^*)$, $k \in \Sigma_C$. This contradicts the assumption that $\mathbf{u}^*$ is a generalized M-Pareto-optimal solution of the GFMOOP, so we conclude that $\mathbf{u}^*$ is an M-Pareto-optimal solution of the corresponding ε-FMOOP for some $\varepsilon_k, k \in \Sigma_C$.

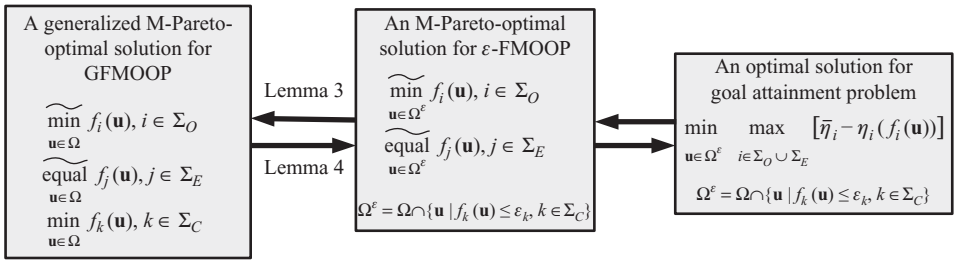
By Lemmas 3 and 4, the GFMOOP and its corresponding ε-FMOOP have the same optimality. An ε-FMOOP is a general FMOOP defined on a domain different from that of its original GFMOOP. According to the demonstration from Sakawa, an M-Pareto-optimal solution of the ε-FMOOP can be obtained by solving the corresponding GAP. So we have the following remark.

Remark 2. A generalized M-Pareto-optimal solution of a GFMOOP problem can be obtained by solving the corresponding goal-attainment problem.

Figure 8.1 depicts the steps to find a Pareto-optimal solution for the primal MOOP and GFMOOP by Remark 1 and Remark 2, respectively.



(A) Optimality relationship for the primal multiobjective optimization problem



(B) Optimality relationship for generalized fuzzy multiobjective optimization problems

Figure 8.1 *Optimality relationships for the primal MOOP and GFMOOP problems.*

8.4 Mixed-Integer Hybrid Differential Evolution

The primal MOOP and GFMOOP were converted into a goal-attainment or min-max problem that is a constrained SOOP and is easy to solve. Most of the objective functions discussed here are nonlinear, and some integer decision variables are included in the objective and constraint functions. The constrained SOOP transferred from the primal MOOP or GFMOOP is reformulated as a constrained MINLP problem. Because the continuity and differentiability of integer decision variables, EAs are the best choice for solving unconstrained MINLP problems. Here we introduce a mixed encoding EA, the MIHDE, which can be used to solve constrained MINLP problems through the implementation of constraint-handling techniques.

8.4.1 Algorithm

Let us consider an unconstrained MINLP problem as follows:

$$\min_{\mathbf{u}\in(\mathbf{x},\mathbf{y})\subset \mathrm{R}^{n_C}\times \mathrm{Z}^{n_I}} f(\mathbf{u}) \tag{8.27}$$

where $\boldsymbol{x}$ represents an n_C-dimensional vector of continuous variables and $\boldsymbol{y}$ is an n_I-dimensional vector of discrete or integer variables. The vector $\mathbf{u} = (\mathbf{x}, \mathbf{y})$ is composed of real and integer variables simultaneously.

Table 8.1 *The basic operations for the evolutionary algorithm and MIHDE.*

Evolutionary algorithm	Mixed-integer hybrid differential evolution
1. Representation and initialization	1. Mixed-coding representation and initialization
2. Mutation	2. Mutation with rounding operation
3. Crossover operation	3. Crossover operation
4. Selection and evaluation	4. Restriction operation
5. Repeat steps 2 to 4	5. Selection and evaluation
	6. Acceleration operation if necessary
	7. Migration operation performed naturally or enforced if necessary
	8. Repeat steps 2 to 6

The MIHDE is a population-based stochastic function optimization method. This method was extended from a real-valued version of hybrid differential evolution (HDE) [34] that was extended from the original algorithm of differential evolution (DE) introduced by Storn and Price [35, 36]. Differential evolution and HDE are unable to handle MINLP problems because they represent genes using a real number encoding strategy. In MIHDE, a rounding operator was embedded into the mutation operation for each integer gene. The basic operations for MIHDE and original DE are expressed in Table 8.1.

MIHDE is a parallel direct-search algorithm that utilizes all N_p individuals $(\mathbf{x}, \mathbf{y})$ in the population. Each encoded individual $(\mathbf{x}, \mathbf{y})$ in MIHDE is composed of real decision variables and integer/discrete decision variables. The initialization process randomly generates N_p individuals to cover the entire search space uniformly.

Unlike conventional evolutionary algorithms, the mutation operation of MIHDE uses the difference between two randomly chosen individuals as a search direction. The ith mutant individual $(\bar{\mathbf{x}}^G, \bar{\mathbf{y}}^G)_i$ in generation G is obtained through the difference of two or four random individuals as expressed in the following form:

$$\begin{aligned}(\bar{\mathbf{x}}^G, \bar{\mathbf{y}}^G)_i &= (\mathbf{x}^G, \mathbf{y}^G)_p + \left\lfloor \rho_m \left\{ (\mathbf{x}^G, \mathbf{y}^G)_k - (\mathbf{x}^G, \mathbf{y}^G)_l \right\} \right\rfloor \\ &= (\mathbf{x}^G, \mathbf{y}^G)_p + \left\{ \rho_m (\mathbf{x}_k^G - \mathbf{x}_l^G), \mathrm{INT}[\rho_m (\mathbf{y}_k^G - \mathbf{y}_l^G)] \right\} \end{aligned} \tag{8.28}$$

where random indices $k, l \in \{1, 2, \ldots, N_p\}$ are mutually different. The operator $\mathrm{INT}[\mathbf{b} = \rho_m(\mathbf{y}_k^G - \mathbf{y}_l^G)]$ in Equation (8.28) is employed to find the integer vector nearest to the real vector $\mathbf{b}$. The mutation factor ρ_m is a real random number and ranges between 0 and 1. This factor is used to control the step length along the searching direction. The DE provided five strategies to select a parent individual $(\mathbf{x}^G, \mathbf{y}^G)_p$ in Equation (8.28). The HDE/MIHDE implements an additional mutation strategy that applies a linear crossover for the ith individual and the best individual $(\mathbf{x}^G, \mathbf{y}^G)_b$ to generate the parent individual. The parent individual is therefore expressed as follows:

$$(\mathbf{x}^G, \mathbf{y}^G)_p = \rho_p (\mathbf{x}^G, \mathbf{y}^G)_b + (1 - \rho_p)(\mathbf{x}^G, \mathbf{y}^G)_i \tag{8.29}$$

where the factor ρ_p is a real random number that ranges between zero and one. The mutation operation may cause the mutant individual to escape from the search domain. If a mutant individual is outside the search domain, then it is replaced by its lower bound or upper bound so that each individual is restricted on the search domain.

The choice of mutation factor for the MIHDE is heuristic and random. While the population diversity is small, the candidate individuals will cluster together rapidly so that the individuals cannot be further improved, and this may result in a premature convergence. To increase the local diversity of the mutant individuals, a binomial crossover is applied to increase the population diversity locally.

The survivor selection operation for the HDE/MIHDE is a one-to-one competition between the parent and its offspring. The competition means that any parent will be replaced by its offspring if its fitness is worse than that of its offspring. On the other hand, any parent will be retained in the next generation if it has a better fitness than its offspring. Two phases are performed in the survivor selection. The first is one-to-one competition. The next phase is to determine the best individual in the population.

When an evolutionary algorithm is used to optimize a function, an acceptable tradeoff between convergence and diversity must be determined. Fast convergence is important, although it may lead to a local optimum. On the other hand, high diversity guarantees the probability of obtaining a global optimum. When population diversity is low, candidate individuals may be tightly clustered. In this case, the mutation and crossover operations of DE will no longer generate a better individual because a premature solution is achieved. HDE/MIHDE includes acceleration and migration operators that act as a tradeoff operation. The acceleration operation is used to speed up convergence. Generally, the best fitness does not descend continuously from generation to generation; it usually improves after several generations. In this situation, the acceleration operation can be used to speed up the convergence. When the mutation and crossover operators no longer improve the best fitness in the present generation, a descent method is applied to move the best individual toward a better solution.

The rate of convergence can be improved by the acceleration operator. However, faster convergence usually results in a premature solution. Performing this operation can also frequently cause candidate individuals to cluster gradually around the best individual, so that population diversity is decreased and cannot reproduce better individuals through the mutation and crossover operators. As a result, a migration operator was used to avoid local clustering. It is performed only if the measure of population diversity fails to satisfy the desired tolerance. Lin *et al.* [37] proposed the population diversity degree ζ to check whether the migration operation should be performed. In order to define the degree of population diversity, Lin *et al.* introduced the following gene diversity index for each real-valued gene x_{ji}^{G+1} and integer-valued gene y_{ji}^{G+1} at the (G+1) generation:

$$dx_{ji} = \begin{cases} 0, & \text{if } \left| \dfrac{x_{ji}^{G+1} - x_{jb}^{G+1}}{x_{jb}^{G+1}} \right| < \varepsilon_2,\ j = 1, \ldots, n_C; i = 1, \ldots, N_P; i \neq b \\ 1, & \text{otherwise} \end{cases} \tag{8.30}$$

$$dy_{ji} = \begin{cases} 0, & \text{if } y_{ji}^{G+1} = y_{jb}^{G+1},\ j = 1, \ldots, n_I; i = 1, \ldots, N_p; i \neq b \\ 1, & \text{otherwise} \end{cases} \tag{8.31}$$

where x_{jb}^{G+1} and y_{jb}^{G+1} are the jth gene of the best individual at the $(G+1)$th iteration, dx_{ji} and dy_{ji} are the gene diversity indices, and $\varepsilon_2 \in [0, 1]$ is a tolerance value for real-valued genes provided by the user. According to Equations (8.30) and (8.31), we set the jth gene diversity index for the ith individual to zero if this gene clusters around the best gene.

The population diversity degree ζ is defined as the ratio of the total diversified genes to the overall genes except the individual. From Equations (8.30) and (8.31), we have the population diversity degree as follows:

$$\zeta = \frac{\sum_{i=1, i\neq b}^{N_p} \left(\sum_{j=1}^{n_C} dx_{ji} + \sum_{j=1}^{n_I} dy_{ji} \right)}{(n_C + n_I)(N_p - 1)} \tag{8.32}$$

From Equations (8.30)–(8.32), we observe that the value of population diversity degree ranges between 0 and 1. A value of 0 implies that all of the genes are clustered around the best individual. On the other hand, a value of 1 indicates that the current candidate individuals are a completely diversified population. The desired tolerance for population diversity is assigned by user. A tolerance value of 0 implies that the migration operation in MIHDE is switched off, and a tolerance value of 1 implies that the migration operation is performed at every generation. Consequently, the user can set a tolerance value for population diversity degree, $\varepsilon_1 \in [0, 1]$, to assess whether migration should be performed. If ζ is smaller than ε_1, then MIHDE performs migration operations to regenerate a new population in order to escape a local point. On the other hand, if ζ is not less than ε_1 then MIHDE suspends the migration operation and maintains a constant search direction toward a target.

8.4.2 Constraint Handling

MIHDE can be used to solve a constrained MINLP problem through the implementation of constraint-handling techniques. A general constrained MINLP problem is composed of the unconstrained MINLP problem defined in Equation (8.27) and equality and inequality constraints as follows:

$$h_j(\mathbf{x}, \mathbf{y}) = 0,\ j = 1, \ldots, m_e \tag{8.33}$$

$$g_j(\mathbf{x}, \mathbf{y}) \leq 0,\ j = 1, \ldots, m_i \tag{8.34}$$

Penalty function methods are some of the most popular techniques for handling constraints [38]. Such techniques convert original constrained problem into an unconstrained problem by penalizing those solutions which are infeasible. A square penalty function is given as follows:

$$P(\mathbf{u}) = f(\mathbf{u}) + \sum_{k=1}^{m_e} \alpha_k h_k^2(\mathbf{u}) + \sum_{k=1}^{m_i} \beta_k \langle g_k(\mathbf{u}) \rangle_+^2 \tag{8.35}$$

where α_k and β_k are the positive penalty parameters and the bracket operation in Equation (8.35) is defined as $\langle g_k \rangle_+ = \max\{g_k, 0\}$. The penalty terms associated with equality and inequality constraints are added to the objective function. As a result, the penalty terms reflect violations of the constraints and assign a high cost of the penalty function to a candidate individual that is far from the feasible region. When we apply MIHDE or EA to solve constrained MINLP problems using the square penalty function as Equation (8.35), any candidate individual that violates the constraints should inherit poorer fitness. As a result, the candidate individual with a higher penalty value should be abandoned in the survivor selection operation.

The penalty function methods are easy to implement. However, the main limitation is the decision of the penalty degree for each constraint. Powell [39] has noted that the classical optimization methods employing penalty functions have certain weaknesses when the

penalty parameters are large. Large penalty parameters in the penalty function strongly distort the objective function of the corresponding unconstrained problem. As a result, the optimal solution of the original constrained problem is difficult to achieve by solving its corresponding unconstrained problem. On the other hand, small penalty parameters may result in an unfeasible solution. Therefore, how to choose appropriate penalty parameters is not trivial.

Gradient-based methods have used Lagrange transformation methods to solve real-valued constrained optimization problems. The transformation methods can significantly overcome the weaknesses for penalty methods. Lin *et al.* [29] have implemented a multiplier updating method into the MIHDE to solve constrained MINLP problems. The augmented Lagrange function for constrained MINLP problems is defined as

$$L_a(\mathbf{u}, \nu, \upsilon) = f(\mathbf{u}) + \sum_{k=1}^{m_e} \alpha_k \left\{ [h_k(\mathbf{u}) + \nu_k]^2 - \nu_k^2 \right\} + \sum_{k=1}^{m_i} \beta_k \left\{ \langle g_k(\mathbf{u}) + \upsilon_k \rangle_+^2 - \upsilon_k^2 \right\} \quad (8.36)$$

where α_k and β_k are positive penalty parameters, and the corresponding Lagrange multipliers are defined as $\lambda_k = 2\alpha_k \nu_k$ and $\mu_k = 2\beta_k \upsilon_k$. The penalty parameters in Equation (8.36) are generally fixed for the evolutionary iterative procedures. However, small penalty parameters for the augmented Lagrange function may still result in an infeasible solution. To overcome this drawback, Lin *et al.* have used a multiplier updating method to enforce global convergence for constrained MINLP problems. Box 8.1 shows the algorithm for MIHDE with a multiplier updating method, including adaptive penalty parameters. Steps 3, 4 and 6 in Box 8.1 are used to improve constraint violation and update the penalty parameters. In Step 4, if the constraint violation is not improved, i.e., $\tilde{\varepsilon}_K \geq \varepsilon_K$, then we increase the penalty parameters by the factor ω_2 (e.g., $\omega_2 = 10$) and reduce the corresponding multipliers by the same factor, thus keeping the multipliers unchanged. In Step 6, we use the factor ω_1 (e.g., $\omega_1 = 4$) to check whether the constraint violation has been improved by the factor ω_1. The penalty parameters and the corresponding multipliers are updated in this step only when the constraint violation is not improved by the factor ω_1.

Box 8.1 MIHDE with multipliers updating and adaptive penalty parameters.

Step 1: Set the initial iteration $l = 0$, the initial multipliers, $\nu_k^l = \nu_k^0 = 0, k = 1, \ldots, m_e$ and $\upsilon_k^l = \upsilon_k^0 = 0, k = 1, \ldots, m_i$, and the initial penalty parameters, $\alpha_k > 0, k = 1, \ldots, m_e$ and $\beta_k > 0, k = 1, \ldots, m_i$. Set the tolerance of the maximum constraint violation, ε_K (e.g., $\varepsilon_K = 10^{32}$), and the scalar factors, $\omega_1 > 1$ and $\omega_2 > 1$.
Step 2: Use the MIHDE algorithm to find a minimum solution of the augmented Lagrange function $L_a(\mathbf{u}, \nu^l, \upsilon^l)$. Let $\mathbf{u}_b^l = (\mathbf{x}_b^l, \mathbf{y}_b^l)$ be a minimum solution of the problem $L_a(\mathbf{u}, \nu^l, \upsilon^l)$.
Step 3: Evaluate the maximum constraint violation as

$$\tilde{\varepsilon}_K = \max \left\{ \max_k |h_k| , \max_k |\max (g_k, -\upsilon_k)| \right\}$$

and establish the following sets of equality and inequality constraints, violation of which will not be improved by the factor ω_1:

$$I_E = \{k : |h_k| > \varepsilon_K/\omega_1, k = 1, \ldots, m_e\},$$

$$I_I = \{k : |\max(g_k, -\upsilon_k)| > \varepsilon_K/\omega_1, k = 1, \ldots, m_i\}.$$

Step 4: If $\tilde{\varepsilon}_K \geq \varepsilon_K$, then let $\alpha_k = \omega_2\alpha_k$ and $\nu_k^{l+1} = \nu_k^l/\omega_2$ for all $k \in I_E$, and let $\beta_k = \omega_2\beta_k$ and $\upsilon_k^{l+1} = \upsilon_k^l/\omega_2$ for all $k \in I_I$, and go to Step 7. Otherwise, go to Step 5.

Step 5: Update the multipliers as follows:

$$\nu_k^{l+1} = h_k\left(\mathbf{u}_b^l\right) + \nu_k^l,$$

$$\upsilon_k^{l+1} = \left\langle g_k\left(\mathbf{u}_b^l\right) + \upsilon_k^l\right\rangle_+ = \upsilon_k^l + \max\left\{g_k\left(\mathbf{u}_b^l\right), -\upsilon_k^l\right\}.$$

Step 6: If $\tilde{\varepsilon}_K \leq \varepsilon_K/\omega_1$, then let $\varepsilon_K = \tilde{\varepsilon}_K$ and go to Step 7. Otherwise, let $\alpha_k = \omega_2\alpha_k$ and $\nu_k^{l+1} = \nu_k^{l+1}/\omega_2$ for all $k \in I_E$, and let $\beta_k = \omega_2\beta_k$ and $\upsilon_k^{l+1} = \upsilon_k^{l+1}/\omega_2$ for all $k \in I_I$. Let $\varepsilon_K = \tilde{\varepsilon}_K$ and go to Step 7.

Step 7: If the maximum iteration is achieved, stop. Otherwise, repeat Steps 2 to 6.

8.5 Examples

A test-constrained MINLP optimization problem was used to evaluate the performance of MIHDE with different penalty functions. The MIDHE with the best penalty function was applied to an FMOOP to maximize ethanol and glycerol production rates in the metabolic network of yeast with fuzzy cell viability and metabolic adjustment as considerations. All the computations were performed on a personal computer using Microsoft Windows 7. The MIHDE algorithm was implemented using Intel Visual Fortran, and required four setting factors provided by the user. These setting factors used for all runs in all computations are listed as follows: The crossover factor is set to be 0.5. Two tolerances used in the migration are set to be 0.05. The population size of five is used in the computations. The maximum iteration of 5000 is used for the inner loop of MIHDE solver. However, we use various maximum iterations in the outer loop for multiplier updating to inspect the solution progress.

Example 1: a test-constrained MINLP optimazation problem

This test-constrained MINLP optimization problem about chemical process design was expressed by Floudas [40] and includes two equality constraints and 21 inequality constraints. The problem involves minimizing the objective function as follows:

$$\begin{aligned}\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x},\mathbf{y}) = {} & 5y_1 + 8y_2 + 6y_3 + 10y_4 + 6y_5 + 7y_6 + 4y_7 + 5y_8 \\ & - 10x_1 - 15x_2 + 15x_3 + 80x_4 + 25x_5 + 35x_6 - 40x_7 \\ & + 15x_8 - 35x_9 + \exp(x_1) + \exp(x_2/12) - 65\ln(x_3 + x_4 + 1) \\ & - 90\ln(x_5 + 1) - 80\ln(x_6 + 1) + 120\end{aligned}$$

subject to

$$
\begin{aligned}
&h_1(\mathbf{y}) = y_1 + y_2 - 1 = 0\\
&h_2(\mathbf{y}) = -y_4 + y_6 + y_7 = 0\\
&g_1(\mathbf{x}) = -1.5\ln(x_5 + 1) - \ln(x_6 + 1) - x_8 \le 0\\
&g_2(\mathbf{x}) = -\ln(x_3 + x_4 + 1) \le 0\\
&g_3(\mathbf{x}) = -x_1 - x_2 + x_3 + 2x_4 + 0.8x_5 + 0.8x_6 - 0.5x_7 - x_8 - 2x_9 \le 0\\
&g_4(\mathbf{x}) = -x_1 - x_2 + 2x_4 + 0.8x_5 + 0.8x_6 - 2x_7 - x_8 - 2x_9 \le 0\\
&g_5(\mathbf{x}) = -2x_4 - 0.8x_5 - 0.8x_6 + 2x_7 + x_8 + 2x_9 \le 0\\
&g_6(\mathbf{x}) = -0.8x_5 - 0.8x_6 + x_8 \le 0\\
&g_7(\mathbf{x}) = -x_4 + x_7 + x_9 \le 0\\
&g_8(\mathbf{x}) = -0.4x_5 - 0.4x_6 + 1.5x_8 \le 0\\
&g_9(\mathbf{x}) = 0.16x_5 + 0.16x_6 - 1.2x_8 \le 0\\
&g_{10}(\mathbf{x}) = x_3 - 0.8x_4 \le 0\\
&g_{11}(\mathbf{x}) = -x_3 + 0.4x_4 \le 0\\
&g_{12}(\mathbf{x}, \mathbf{y}) = \exp(x_3) - 10y_1 - 1 \le 0\\
&g_{13}(\mathbf{x}, \mathbf{y}) = \exp(x_2/1.2) - 10y_2 - 1 \le 0\\
&g_{14}(\mathbf{x}, \mathbf{y}) = x_7 - 10y_3 \le 0\\
&g_{15}(\mathbf{x}, \mathbf{y}) = 0.8x_5 + 0.8x_6 - 10y_4 \le 0\\
&g_{16}(\mathbf{x}, \mathbf{y}) = 2x_4 - 2x_7 - 2x_9 - 10y_5 \le 0\\
&g_{17}(\mathbf{x}, \mathbf{y}) = x_5 - 10y_6 \le 0\\
&g_{18}(\mathbf{x}, \mathbf{y}) = x_6 - 10y_7 \le 0\\
&g_{19}(\mathbf{x}, \mathbf{y}) = x_3 + x_4 - 10y_8 \le 0\\
&g_{20}(\mathbf{y}) = y_4 + y_5 - 1 \le 0\\
&g_{21}(\mathbf{y}) = y_3 - y_8 \le 0\\
&\{0, \ldots, 0\} \le x_k \le \{2, 2, 1, 2, 2, 2, 2, 1, 3\},\ k = 1, \ldots, 9\\
&y_k \in \{0, 1\},\ k = 1, \ldots, 8
\end{aligned}
$$

Here x_k represents a real variable, and y_k a binary variable. A given minimum solution is 68.0097 for this problem. One hundred trials were carried out. To investigate the test result, the following terms are introduced: f_b and f_w are the best and the worst optimal objective function value, respectively, in the 100 trials; f_m is the mean value of the optimal objective function values in the 100 trials; and σ_f is the standard deviation of the 100 optimal solutions. The sum of the constraint violations SCV $(= \sum_{k=1}^{m_e} |h_k| + \sum_{k=1}^{m_i} \langle g \rangle_+)$ is defined to inspect the feasibility of a solution. σ_{SCV} is the standard deviation of SCVs in the 100 trials. The success rate R_s denotes the percentage of convergence to the exact global optimum in all trials. In the computations, the so-called exact global optimum needs to satisfy the following conditions: (i) $f_b \le VTR$ (optimal value to reach), (ii) $|h_k| \le 10^{-8}$, $k = 1, \ldots, m_e$, and (iii) $\langle g_k \rangle_+ \le 10^{-8}$, $k = 1, \ldots, m_i$. N_{fe} is the mean number of objective function evaluations. To improve the diversity of the population, we encoded all binary decision variables together as an integer variable as follows:

$$z = y_{n_I} 2^{n_I - 1} + y_{n_I - 1} 2^{n_I - 2} + \cdots + y_2 2^1 + y_1 2^0$$

The MIHDE with multipliers updating and adaptive penalty parameters (referred to as MIHDE-APP), MIHDE with multipliers updating and fixed penalty parameters (referred to as MIHDE-FPP), and MIHDE using the fixed-penalty function method (referred to as MIHDE-PFM) were applied to solve this artificial problem. Each trial contains 5000

iterations for the inner loop of MIHD-APP and MIHDE-FPP and 50 iterations for the outer loop as shown in Box 8.1. The maximum of 250 000 iterations were used in the MIHDE-PFM. Table 8.2 shows the computational results. The global minimum solution could be obtained by MIHDE-APP using small initial penalty parameters. Moreover, the SCV values were near to zero. The MIHDE-APP could get a premature solution and the success rate was reduced to 0.6, if we used a larger initial penalty parameter, such as 10^6. In this situation, the migration operation in MIHDE-APP cannot avoid a premature solution using a larger initial penalty parameter. A smaller fixed-penalty parameter results in MIHDE-FPP failing to reach the global minimum, as observed in Table 8.2. The success rate was 0.77 in the case of 10^3, but it declined to 0.46 in the case of 10^6. MIHDE-PFM has difficulty in achieving the global minimum as observed in Table 8.2. All objective function values were smaller than the "value to reach" (VTR) but the large SCV values show that the solutions were infeasible.

Example 2: maximization of the ethanol/glycerol production by yeast

Saccharomycescerevisiae is still the most important microorganism for ethanol/glycerol production to date. Many strategies have been developed to enhance the ethanol/glycerol productivity using yeast, because its metabolic network is well studied. Figure 8.2 shows a scheme of the simplified central metabolic network of *S. cerevisiae*. For more details on the simplifications, assumptions, and experimental evidences used to build this model, the reader can refer to the previous papers [7, 8, 41–43].

The dynamics of the simplified central metabolic network are governed by the following equations [42]:

$$
\begin{aligned}
\dot{x}_1 &= 0.9023x_2^{-0.2344}x_6 - 3.1847x_1^{0.7464}x_5^{0.0253}x_7 \\
\dot{x}_2 &= 3.1847x_1^{0.7464}x_5^{0.0253}x_7 - 0.5232x_2^{0.7318}x_5^{-0.3941}x_8 - 0.0009x_2^{0.7318}x_{11} \\
&\quad - 1.76898x_2^{0.0526}x_{15}^{0.9646} \\
\dot{x}_3 &= 0.5232x_2^{0.7318}x_5^{-0.3941}x_8 - 0.011x_3^{0.6159}x_5^{0.1308}x_9x_{14}^{-0.6088} \\
&\quad - 0.0516x_3^{0.05}x_4^{0.533}x_5^{-0.0822}x_{12} \\
\dot{x}_4 &= 0.022 \times \left(x_3^{0.6159}x_5^{0.1308}x_9x_{14}^{-0.6088}\right) - 0.0947x_3^{0.05}x_4^{0.533}x_5^{-0.0822}x_{10} \\
\dot{x}_5 &= 0.022 \times \left(x_3^{0.6159}x_5^{0.1308}x_9x_{14}^{-0.6088}\right) + 0.0947x_3^{0.05}x_4^{0.533}x_5^{-0.0822}x_{10} \\
&\quad - 3.1847x_1^{0.7464}x_5^{0.0243}x_7 - 0.0009x_2^{0.7318}x_{11} - 0.5232x_2^{0.7318}x_5^{-0.3941}x_8 \\
&\quad - 0.937905x_5x_{13}
\end{aligned}
\tag{8.37}
$$

The model consists of five nonlinear ordinary differential equations, nine nonlinear rate equations, and ten independent variables. The rate equations, fluxes, metabolites and their corresponding basal values are shown in Table 8.3.

MIHDE-APP and the commercial software GAMS 23.6 with seven solvers were applied to solve the metabolic network for finding the optimal enzyme manipulations in S. cerevisiae. The feasible region for each metabolite and enzyme can be estimated through biological understanding or determined by global optimization techniques [42, 44]. Here, the feasible region for each metabolite and enzyme is set to expand/shrink fivefold based on the basal value. The primal optimization problem for maximizing the ethanol and glycerol productivity in *S. cerevisiae* was respectively solved by MIHDE-APP to obtain the Pareto solution with various allowable manipulated enzymes from one to five.

Table 8.2 *Comparison of results for MIHDE-APP, MIHDE-FPP and MIHDE-PFM with various initial penalty parameters α_k and β_k. VTR = 68.0098.*

		α_k and β_k			
Method	Item	0.1	10	10^3	10^6
MIHDE-APP	f_b, SCV	68.00971, 5.424E-9,	68.00971, 2.011E-9	68.00971, 1.363E-9	68.00974, 2.097E-8
	f_m, SCV	68.00975, 8.784E-9	68.00976, 7.076E-9	68.17802, 6.267E-9	71.89475, 5.148E-9
	f_w, SCV	68.00978, 9.597E-10	68.00979, 1.172E-10	76.41937, 3.289E-13	98.69519, 2.719E-15
	σ_f, σ_{SCV}	1.049E-5, 6.152E-9	2.424E-5, 5.542E-9	1.177, 5.677E-9	6.726, 6.661E-9
	R_s	1.0	1.0	0.97	0.6
	N_{fe}	91241	80298	171750	660197
MIHDE-FPP	f_b, SCV	−54.0461, 19.518	49.85873, 0.946	68.00971, 2.146E-9	68.00971, 6.865E-9
	f_m, SCV	−44.3731, 17.291	55.03625, 0.822	68.00974, 6.426E-9	68.85071, 9.623E-9
	f_w, SCV	−38.2811, 13.357	62.29849, 0.573	68.00976, 1.262E-10	76.41937, 1.744E-11
	σ_f, σ_{SCV}	3.090, 1.744	1.981, 0.269	7.779E-6, 5.272E-9	2.523, 6.829E-9
	R_s	0.0	0.0	0.77	0.46
	N_{fe}	1268521	1270219	142854	390550
MIHDE-PFM	f_b, SCV	−201.18164, 31.491	−14.51017, 4.049	66.25476, 5.975E-2	68.00793, 6.058E-5
	f_m, SCV	−201.18164, 31.491	−14.51017, 4.049	66.25476, 5.975E-2	68.17611, 6.091E-5
	f_w, SCV	−201.18164, 31.491	−14.51017, 4.049	66.25476, 5.975E-2	76.41699, 7.724E-5
	σ_f, σ_{SCV}	0.0, 8.531E-8	0.0, 2.333E-8	0.0, 3.167E-9	1.177, 2.333E-6
	R_s	0.0	0.0	0.0	0.0
	N_{fe}	1267826	1256837	1259645	1259434

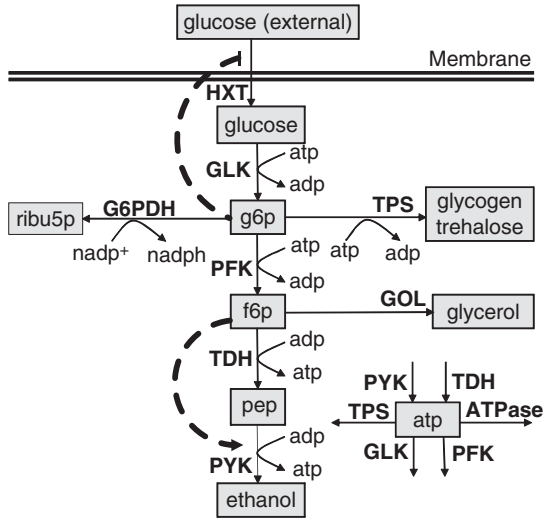


Figure 8.2 *Schematic representation of the central metabolism of yeast.*

The larger the allowable number of manipulated enzymes in the metabolic network, the higher the improved ethanol/glycerol flux ratio, v_{PYK}/v_{PYK}^{basal} or v_{GOL}/v_{GOL}^{basal}. Tables 8.4(a) and 8.4(b) show the optimal enzymatic modulations for maximizing ethanol and glycerol flux ratio, respectively, when various manipulated enzymes are allowable. Table 8.4(a) shows that the best modulated enzymes were glucose uptake (HXT) and glyceraldehyde-3-phosphate dehydrogenase (TDH) if two modulated enzymes are allowed. This result is different from that (HXT and PFK) obtained by Polisetty *et al.* [41] due to the consideration of cell viability constraints. Sorribas *et al.* [42] have shown that increasing HXT above

Table 8.3 *Basal metabolite concentrations and enzyme activities.*

Symbol	Name	Value
x_1	Internal glucose	0.0345 mM
x_2	Glucose-6-phosphate	1.011 mM
x_3	Fructose-1,6-diphosphate	9.144 mM
x_4	Phosphoenolpyruvate (PEP)	0.0095 mM
x_5	ATP	1.1278 mM
x_6	Glucose uptake (HXT)	19.7 mM/min
x_7	Hexokinase	68.5 mM/min
x_8	Phosphofructokinase (PFK)	31.7 mM/min
x_9	Glyceraldehyde-3-phosphate dehydrogenase (GAPD)	49.9 mM/min
x_{10}	Pyruvate kinase (PYK)	3440 mM/min
x_{11}	Polysaccharide production (glycogen + trehalose)	14.31 mM/min
x_{12}	Glycerol production (GOL)	203 mM/min
x_{13}	ATPase	25.1 mM/min
x_{14}	NAD^+/NADH ratio	0.042
x_{15}	Glucose 6-phosphate dehydrogenase (G6PDH)	1.0 mM/min

Table 8.4 *The optimal enzymatic modulations for maximizingethanol/glycerol flux ratio by S. cerevisiae obtained by solving the primal optimization problem using MIHDE-APP and various GAMS solvers (AlphaECP, BARON, BONMIN, COUENNE, DICOPT, LINDOGlobal, and SBB).*

(a) maximization of the ethanol flux ratio, $\max_{\mathbf{x},\mathbf{y},\mathbf{e}}(v_{PYK}/v_{PYK}^{basal})$

ε	$v_{PYK}^{*}/v_{PYK,basal}$	$v_{GOL}/v_{GOL,basal}$	Modulated enzymes
1	1.607	1.750	HXT
2	2.267	2.468	HXT, TDH
3	2.577	2.806	HXT, TDH, ATPase
4	3.832	2.018	HXT, PFK, PYK, NAD_ratio
5	4.444	2.341	HXT, GLK, PFK, PYK, NAD_ratio
	4.405†	2.324	HXT, GLK, PFK, TDH, PYK

(b) maximization of the glycerol productivity, $\max_{\mathbf{x},\mathbf{y},\mathbf{e}}(v_{GOL}/v_{GOL}^{basal})$

ε	$v_{PYK}/v_{PYK,basal}$	$v_{GOL}^{*}/v_{GOL,basal}$	Modulated enzymes
1	0.851	4.633	GOL
2	0.746	8.660	PYK, GOL
3	1.130	15.111	HXT, PYK, GOL
4	1.139	15.267	HXT, PYK, GOL, NAD_ratio
	1.136	15.119†	HXT, PFK, PYK, GOL
5	1.135	15.230	HXT, PFK, PYK, GOL, NAD_ratio
	1.136	15.119†	HXT, GLK, PFK, PYK, GOL

†denotes that the optimal solution obtained by GAMS is a premature result and ε is the number of allowable manipulated genes.

4.86-fold and PFK above 3.16-fold simultaneously led to unfeasible solutions. A smaller optimal ethanol flux ratio was predicted in our results because we expect to obtain a viable strain. The third column of Table 8.4(a) lists the corresponding glycerol flux ratios when maximizing the ethanol flux ratio. Similarly, the second column of Table 8.4(b) shows the corresponding relative changes for ethanol flux when maximizing the glycerol flux ratio. We also apply seven MINLP solvers in GAMS to solve the primal optimization problem with various allowable manipulated enzymes, and the optimal results obtained are also shown in Table 8.4. Most of the maximum ethanol/glycerol flux ratios and modulated enzymes are identical to those obtained by MIHDE-APP, but some premature solutions (shown in a different row) were obtained by GAMS, such as the maximum ethanol flux ratio of 4.405 and the maximum glycerol flux ratio of 15.119 for five allowable manipulated enzymes. Each premature solution can be improved if the convergent solution obtained by MIHDE is provided as the initial point for the GAMS solvers. However, more computation time is required for MIHDE to obtain a feasible solution.

Table 8.5 shows the results for the resilience optimization problem. The maximum ethanol flux ratio for different allowable numbers of manipulated enzymes is reduced by 10–40%. The best modulated enzymes appear to be HXT and TDH if two modulated enzymes are allowed. These two best modulated enzymes are exactly the same in the primal optimization

Table 8.5 *The optimal enzymatic modulations for maximizing ethanol/glycerol flux ratio by S. cerevisiae obtained by solving the resilience optimization problem using MIHDE-APP and various GAMS solvers (AlphaECP, BARON, BONMIN, COUENNE, DICOPT, LINDOGlobal, and SBB).*

(a) fuzzy maximization of the ethanol flux ratio, $\widetilde{\max}_{\mathbf{x},\mathbf{y},\mathbf{e}}(v_{PYK}/v_{PYK}^{basal})$ with $\sum_{i=1}^{n} x_i \precsim [1.6, 2] \sum_{i=1}^{n} x_i^{basal}$, $\sum_{i=1}^{m} e_i \precsim [1.6, 2] \sum_{i=1}^{m} e_i^{basal}$

ε	$v_{PYK}^{*}/v_{PYK,basal}$	$v_{GOL}/v_{GOL,basal}$	Selected enzymes
1	1.443	1.572	HXT
2	1.701	1.852	HXT, TDH
3	1.857	2.022	HXT, TDH, ATPase
4	2.481	1.459	HXT, PFK, TDH, PYK
5	2.773	1.830	HXT, GLK, PFK, TDH, PYK

(b) fuzzy maximization of the glycerol flux ratio, $\widetilde{\max}_{\mathbf{x},\mathbf{y},\mathbf{e}}(v_{GOL}/v_{GOL}^{basal})$, $\sum_{i=1}^{n} x_i \precsim [1.6, 2] \sum_{i=1}^{n} x_i^{basal}$, $\sum_{i=1}^{m} e_i \precsim [1.6, 2] \sum_{i=1}^{m} e_i^{basal}$

ε	$v_{PYK}/v_{PYK,basal}$	$v_{GOL}^{*}/v_{GOL,basal}$	Selected enzymes
1	0.920	2.906	GOL
2	1.458	4.794	HXT, GOL
3	1.073	7.056	HXT, PYK, GOL
4	1.357	7.228	HXT, PYK, GOL, ATPase
	2.036	7.219†	HXT, TDH, GOL, ATPase
5	1.391	7.243	HXT, PFK, PYK, GOL, ATPase
	1.403	7.245†	HXT, PFK, PYK, GOL, ATPase

†denotes that the optimal solution obtained by GAMS is a premature result and ε is the number of allowable manipulated genes.

problem, but the maximum ethanol flux ratio is reduced from 2.267 to 1.701. In contrast, the maximum glycerol flux ratio for different allowable numbers of manipulated enzymes is reduced by 37%–55%. In this case, the best modulated enzymes are HXT and GOL if two modulated enzymes are allowed. The modulated enzymes are different from those obtained from the primal optimization problem (PYK and GOL) and the maximum glycerol flux ratio is reduced from 8.66 to 4.794.

The results shown in Tables 8.5(a) and 8.5(b) are obtained for maximizing the ethanol flux ratio and glycerol flux ratio, respectively. As a result, we could obtain the maximum ethanol flux ratio, but the glycerol flux ratio was enhanced a little, and vice versa. Suppose that we want to maximize ethanol flux ratio and glycerol flux ratio simultaneously. Table 8.6 shows the optimal enzymatic modulations for maximizing ethanol and glycerol flux ratios simultaneously obtained by MIHDE-APP and GAMS solvers. Two different optimal Pareto solutions with identical improvement ratio were found when three manipulated enzymes were allowed.

Table 8.6 *The optimal enzymatic modulations for maximizing ethanol and glycerol flux ratios simultaneously by S. cerevisiae obtained by solving the resilience optimization problem using MIHDE-APP and various GAMS solvers (AlphaECP, BARON, BONMIN, COUENNE, DICOPT, LINDOGlobal, and SBB).*

$f_0 = \widetilde{\max}_{\mathbf{x},\mathbf{y},\mathbf{e}}(v_{PYK}/v_{PYK,basal})\ f_1 = \widetilde{\max}_{\mathbf{x},\mathbf{y},\mathbf{e}}(v_{GOL}/v_{GOL,basal}), \sum_{i=1}^{n} x_i \precsim [1.6, 2] \sum_{i=1}^{n} x_i^{basal}, \sum_{i=1}^{m} e_i \precsim [1.6, 2] \sum_{i=1}^{m} e_i^{basal}$

ε	$v^*_{PYK}/v_{PYK,basal}$	$v^*_{GOL}/v_{GOL,basal}$	Selected enzymes
1	1.565	1.704	HXT
	1.361†	1.482†	PFK (GAMS)
2	1.493	3.980	HXT, GOL
3	1.807	6.674	HXT, TDH, GOL
	1.807‡	6.674‡	HXT, GOL, NAD_ratio
4	2.106	7.884	HXT, TDH, GOL, ATPase
	1.979†	7.794†	HXT, PFK, GOL, NAD_ratio
5	2.402	6.820	HXT, PFK, TDH, PYK, GOL

†denotes that the optimal solution obtained by GAMS is a premature result.
‡denotes the Pareto-optimal solution where the improvement ratios are identical to the results modulated by {HXT, TDH, GOL}, and ε is the number of allowable manipulated genes.

8.6 Conclusions

To capture experimentally observed data of metabolic networks, it is essential to develop a kinetic model. The accuracy of optimal solutions depends heavily on the exactness of kinetic models used in metabolic engineering problems. In general, this type of metabolic engineering problem is formulated as a constrained MINLP problem. The challenge for solving constrained MINLP problems stems from the fact that they are highly nonlinear and nondifferentiable due to the combinatorial nature of the associated integer-valued decision variables. The results from performance evaluations show that MIHDE-APP is a good candidate approach to solve constrained MINLP problems.

The optimization of biological systems, which is a branch of metabolic engineering, has generated much industrial and academic interest for a long time. The ultimate goal of this optimization is to find the optimal modulation strategy for improving productivity. Model-based optimization strategies have been applied to analyze and design metabolic networks during the 2000s. To address the issues of optimizing the regulatory structure of metabolic networks, it is necessary to consider qualitative effects, for example, the resilience phenomena and cell viability constraints. The combination of qualitative and quantitative descriptions for metabolic networks makes it possible to design a viable strain and accurately predict the maximum possible flux rates of desired products. The analysis report of the GFMOOP shows that the maximum synthesis rates of target products by enzyme modulations are always overestimated in metabolic networks that do not consider the resilience effects.

Exercises

8.1. Global optimization is a technique most frequently encountered in the design of pressure vessels. Sandgren formulated a pressure-vessel design problem as a mixed-integer nonlinear programming problem [45]. The design variables are the dimensions required for the specifications of the vessel. The objective function is the combined costs of material, forming and welding of the pressure vessel. The constraints are set in accordance with the respective ASME codes. The problem was formulated as follows:

$$\begin{aligned}\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x},\mathbf{y}) = {} & 0.6224\,(0.0625y_1)\,x_1x_2 + 1.7781\,(0.0625y_2)\,x_1^2 \\ & + 3.1661\,(0.0625y_1)^2\,x_2 + 19.84\,(0.0625y_1)^2\,x_1\end{aligned}$$

subject to

$$\begin{aligned}& g_1(\mathbf{x},\mathbf{y}) = 0.0193x_1 - 0.0625y_1 \le 0,\\ & g_2(\mathbf{x},\mathbf{y}) = 0.00954x_1 - 0.0625y_2 \le 0,\\ & g_3(\mathbf{x},\mathbf{y}) = 750\times 1728 - \pi x_1^2x_2 - \frac{4\pi}{3}x_1^3 \le 0,\\ & g_4(\mathbf{x},\mathbf{y}) = x_2 - 240 \le 0,\\ & 0 \le x_1, x_2 \le 120.0,\ 1 \le y_1, y_2 \le 99,\ \mathbf{x}\in\mathbb{R}^2,\ \mathbf{y}\in\mathbb{Z}^2.\end{aligned}$$

8.2. Find the global optimal solution of the following system:

$$\min f(\mathbf{x},\mathbf{y}) = 7.5y_1 + 5.5y_2 + 5x_3 + 7x_4 + 6x_5$$

subject to

$$\begin{aligned}& y_1 + y_2 - 1 = 0\\ & x_6 - 0.9x_1[1-\exp(-0.5x_4)] = 0\\ & x_7 - 0.8x_2[1-\exp(-0.5x_5)] = 0\\ & x_6 + x_7 - 10 = 0\\ & x_1 + x_2 - x_3 = 0\\ & x_6y_1 + x_7y_2 - 10 = 0\\ & x_4 - 10y_1 \le 0\\ & x_5 - 10y_2 \le 0\\ & x_1 - 20y_1 \le 0\\ & x_2 - 20y_2 \le 0\\ & \mathbf{x}\ge\mathbf{0},\ \mathbf{y}\in\{0,1\}\end{aligned}$$

The global minimum is $(x_3, x_4, x_5, y_1, y_2; f) = (13.362272, 3.514237, 0, 1, 0; 99.245209)$.

8.3. The central carbon metabolism plays essential roles in many organisms, such as *E. coli* and *C. glutamicum*, providing energy metabolism and precursors for aromatic amino acids and serine syntheses. Chassagnole *et al.* [46] developed a nonlinear dynamic model for part of central carbon metabolism of *E. coli*. This model links the kinetics of sugar transporter PTS (phosphor-transferase system) with glycolysis and pentose-phosphate pathways and is used to support the exploration of the central carbon metabolism of *E. coli*. It consists of 18 nonlinear ordinal differential equations and 30 nonlinear rate equations. For brevity, details of the model and rate equations can

be found in additional file 2 of [26]. Seven co-metabolites (amp, adp, atp, nad, nadh, nadp, and nadph) are included in the model and their concentrations are assumed to be (0.955, 0.595, 4.27, 1.47, 0.1, 0.195, and 0.062). The maximum reaction rates can be found from the model database of JWS Online Cellular Systems Modeling at http://jjj.biochem.sun.ac.za (accessed 4 December 2012). Formulate the resilience problem for the central carbon metabolism of *E. coli* mentioned above and use a linear membership function for each fuzzy objective function, fuzzy equal objective, and fuzzy inequality constraint. Apply fuzzy inequality constraints to handle the cell viability constraints and use the lower and upper restriction factors of 1.6 and 2.0, respectively. Find the optimal enzyme manipulation strategies to maximize the flux ratios of DAHPS, PEPC, and SERS simultaneously in a variety of allowable manipulated enzyme numbers and check out the effects of the resilience phenomenon.

References

[1] J. Banga, Optimization in computational systems biology, *BMC Systems Biology*, 2(1), 47 (2008).

[2] C.S. Henry, M. DeJongh, A.A. Best, P.M. Frybarger, B. Linsay and R.L. Stevens, High-throughput generation, optimization and analysis of genome-scale metabolic models, *Nature Biotechnology*, 28(9), 977–982 (2010).

[3] S. Ranganathan, P.F. Suthers and C.D. Maranas, OptForce: an optimization procedure for identifying all genetic manipulations leading to targeted overproductions, *PLoS Computational Biology*, 6(4), e1000744 (2010).

[4] C. Pozo, G. Guillén-Gosálbez, A. Sorribas and L. Jiménez, A spatial branch-and-bound framework for the global optimization of kinetic models of metabolic networks, *Industrial and Engineering Chemistry Research*, 50(9), 5225–5238 (2010).

[5] F.J. Planes and J.E. Beasley, An optimization model for metabolic pathways, *Bioinformatics*, 25(20), 2723–2729 (2009).

[6] A. Marín-Sanguino and N.V. Torres, Optimization of biochemical systems by linear programming and general mass action model representations, *Mathematical Biosciences*, 184(2), 187–200 (2003).

[7] N.V. Torres, E.O. Voit, C. Glez-Alcón and F. Rodríguez, An indirect optimization method for biochemical systems: Description of method and application to the maximization of the rate of ethanol, glycerol, and carbohydrate production in Saccharomyces cerevisiae, *Biotechnology and Bioengineering*, 55(5), 758–772 (1997).

[8] J. Vera, P. De Atauri, M. Cascante and N.V. Torres, Multicriteria optimization of biochemical systems by linear programming: Application to production of ethanol by Saccharomyces cerevisiae, *Biotechnology and Bioengineering*, 83(3), 335–343 (2003).

[9] J. Vera, R. Curto, M. Cascante and N.V. Torres, Detection of potential enzyme targets by metabolic modelling and optimization: application to a simple enzymopathy, *Bioinformatics*, 23(17), 2281–2289 (2007).

[10] E.O. Voit and M. Del Signore, Assessment of effects of experimental imprecision on optimized biochemical systems, *Biotechnology and Bioengineering*, 74(5), 443–448 (2001).

[11] F. Alvarez-Vasquez, C. González-Alcón and N.V. Torres, Metabolism of citric acid production by *Aspergillus niger*: Model definition, steady-state analysis and constrained optimization of citric acid production rate, *Biotechnology and Bioengineering*, 70(1), 82–108 (2000).

[12] A. Marín-Sanguino and N.V. Torres, Optimization of tryptophan production in bacteria. Design of a strategy for genetic manipulation of the tryptophan operon for tryptophan flux maximization, *Biotechnology Progress*, 16(2), 133–145 (2000).

[13] A. Marín-Sanguino and N. Torres, Modelling, steady state analysis and optimization of the catalytic efficiency of the triosephosphate isomerase, *Bulletin of Mathematical Biology*, 64(2), 301–326 (2002).

[14] J. Vera, N.V. Torres, C.G. Moles and J. Banga, Integrated nonlinear optimization of bioprocesses via linear programming, *AIChE Journal*, 49(12), 3173–3187 (2003).

[15] J. Vera, C. González-Alcón, A. Marín-Sanguino and N. Torres, Optimization of biochemical systems through mathematical programming: methods and applications, *Computers and Operations Research*, 37(8), 1427–1438 (2010).

[16] F. Rodríguez-Acosta, C.M. Regalado and N.V. Torres, Non-linear optimization of biotechnological processes by stochastic algorithms: Application to the maximization of the production rate of ethanol, glycerol and carbohydrates by Saccharomyces cerevisiae, *Journal of Biotechnology*, 68(1), 15–28 (1999).

[17] J.E. Bailey, Toward a science of metabolic engineering, *Science*, 252(5013), 1668–1675 (1991).

[18] F.G. Vital-Lopez, A. Armaou, E.V. Nikolaev and C.D. Maranas, A computational procedure for optimal engineering interventions using kinetic models of metabolism, *Biotechnology Progress*, 22(6), 1507–1517 (2006).

[19] G.P. Rangaiah, (2009) *Multi-Objective Optimization: Techniques and Applications in Chemical Engineering*, World Scientific, Singapore.

[20] M. Sakawa, (1993) *Fuzzy Sets and Interactive Multiobjective Optimization*, Plenum Press, New York.

[21] Y. Sawaragi, H. Nakayama and T. Tanino, (1985) *Theory of Multiobjective Optimization*, Academic Press, London.

[22] D. Segrè, D. Vitkup and G.M. Church, Analysis of optimality in natural and perturbed metabolic networks, *Proceedings of the National Academy of Sciences*, 99(23), 15112–15117 (2002).

[23] T. Shlomi, O. Berkman and E. Ruppin, Regulatory on/off minimization of metabolic flux changes after genetic perturbations, *Proceedings of the National Academy of Sciences of the United States of America*, 102(21), 7695–7700 (2005).

[24] F.-S. Wang, C.-H. Jing and G.T. Tsao, Fuzzy-decision-making problems of fuel ethanol production using a genetically engineered yeast, *Industrial and Engineering Chemistry Research*, 37(8), 3434–3443 (1998).

[25] F.-S. Wang and C.-H. Jing, Application of hybrid differential evolution to fuzzy dynamic optimization of a batch fermentation, *Journal of the Chinese Institute of Chemical Engineers*, 31(5), 443–453 (2000).

[26] W.-H. Wu, F.-S. Wang and M.-S. Chang, Multi-objective optimization of enzyme manipulations in metabolic networks considering resilience effects, *BMC Systems Biology*, 5(1), 145 (2011).

[27] H.-J. Huang and F.-S. Wang, Fuzzy decision-making design of chemical plant using mixed-integer hybrid differential evolution, *Computers and Chemical Engineering*, 26(12), 1649–1660 (2002).
[28] C.T. Liao, W.J. Tzeng and F.S. Wang, Mixed-integer hybrid differential evolution for synthesis of chemical processes, *Journal of the Chinese Institute of Chemical Engineers*, 32(6), 491–502 (2001).
[29] Y.C. Lin, K.S. Hwang and F.S. Wang, An evolutionary Lagrange method for mixed-integer constrained optimization problems, *Engineering Optimization*, 35(3), 267–284 (2003).
[30] E.O. Voit, (2000) *Computational Analysis of Biochemical Systems*, Cambridge University Press, Cambridge.
[31] F.C. Lee, G.P. Rangaiah and D.-Y. Lee, Modeling and optimization of a multi-product biosynthesis factory for multiple objectives, *Metabolic Engineering*, 12(3), 251–267 (2010).
[32] Y. Lee, F. Chen, L. Gallego-Giraldo, R.A. Dixon and E.O. Voit, Integrative analysis of transgenic alfalfa (*Medicago sativa* L.) suggests new metabolic control mechanisms for monolignol biosynthesis, *PLoS Computational Biology*, 7(5), e1002047 (2011).
[33] M. Shuler and F. Kargi, (2002) *Bioprocess Engineering: Basic Concepts*, 2nd edn, Prentice Hall, New York.
[34] J.-P. Chiou and F.-S. Wang, Hybrid method of evolutionary algorithms for static and dynamic optimization problems with application to a fed-batch fermentation process, *Computers and Chemical Engineering*, 23(9), 1277–1291 (1999).
[35] R. Storn and K. Price, Minimizing the real functions of the ICEC'96 Contest by differential evolution, IEEE Conference on Evolutionary Computation, pp. 842–844 (1996).
[36] R. Storn and K. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous paces, *Journal of Global Optimization*, 11(4), 341–359 (1997).
[37] Y.-C. Lin, F.-S. Wang and K.-S. Hwang, A hybrid method of evolutionary algorithms for mixed-integer nonlinear optimization problems, Proceedings of the 1999 IEEE Congress on Evolutionary Computation (CEC 99), Washington D.C., pp. 2159–2166 (1999).
[38] Z. Michalewicz and M. Schoenauer, Evolutionary algorithms for constrained parameter optimization problems, *Evolutionary Computation*, 4, 1–32 (1996).
[39] M.J.D. Powell, Algorithms for nonlinear constraints that use Lagrangian functions, *Mathematical Programming*, 14(1), 224–248 (1978).
[40] C.A. Floudas (1995) *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*, Oxford University Press, New York.
[41] P.K. Polisetty, E.P. Gatzke and E.O. Voit, Yield optimization of regulated metabolic systems using deterministic branch-and-reduce methods, *Biotechnology and Bioengineering*, 99(5), 1154–1169 (2008).
[42] A. Sorribas, C. Pozo, E. Vilaprinyo, G. Guillén-Gosálbez, L. Jiménez and R. Alves, Optimization and evolution in metabolic pathways: Global optimization techniques in generalized mass action models, *Journal of Biotechnology*, 149(3), 141–153 (2010).
[43] E.O. Voit and T. Radivoyevitch, Biochemical systems analysis of genome-wide expression data, *Bioinformatics*, 16(11), 1023–1037 (2000).

[44] G. Guillén-Gosálbez and A. Sorribas, Identifying quantitative operation principles in metabolic pathways: a systematic method for searching feasible enzyme activity patterns leading to cellular adaptive responses, *BMC Bioinformatics*, 10(1), 386 (2009).

[45] E. Sandgren, Nonlinear integer and discrete programming in mechanical design optimization, *Journal of Mechanical Design*, 112(2), 223–229 (1990).

[46] C. Chassagnole, N. Noisommit-Rizzi, J.W. Schmid, K. Mauch and M. Reuss, Dynamic modeling of the central carbon metabolism of *Escherichia coli*, *Biotechnology and Bioengineering*, 79(1), 53–73 (2002).