Supplementary File S1

Multiobjective Strain Design: A Framework for Modular Cell Engineering

Sergio Garcia[1,2] and Cong T. Trinh[1,2,*]

[1] *Department of Chemical and Biomolecular Engineering, The University of Tennessee, Knoxville, TN*
[2] *Center for Bioenergy Innovation, Oak Ridge National Laboratory, Oak Ridge, TN*
[*] *Corresponding author: Tel: 865-974-2181. Email: ctrinh@utk.edu.*

Contents

1 Solution method: Multiobjective Evolutionary Algorithm

1.1 Definitions

Terms

Parent network: A parent network is a metabolic model of a host organism that is used to construct a modular cell.

Production module: A production module is a metabolic pathway that is added to a modular cell to synthesize a target product.

Production network: A production network is a combination of a parent network and a production module.

MOEA: Multiobjective evolutionary algorithm.

Sets

$\mathcal{I}_k$: Set of metabolite indices in production network k.

$\mathcal{J}_k$: Set of reaction indices in production network k.

$\mathcal{K}$: Set of production network indices, where $\forall k \in \mathcal{K}$.

$\mathcal{C}$: Set of candidate reaction deletion indices, where $\mathcal{C} \subseteq \mathcal{J}^{parent} \subseteq \mathcal{J}_k$, $\forall k \in \mathcal{K}$.

Continuous variables

v_{jk} : Flux of reaction j in production network k.

v_{Pk} : Flux of target product (P) reaction in production network k.

v_{Xk} : Flux of biomass (X) synthesis reaction in production network k.

f_k : Objective function for production network k.

f_k^{wGCP} : *wGCP* objective function for production network k.

f_k^{sGCP} : *sGCP* objective function for production network k.

f_k^{NGP} : *NPG* objective function for production network k.

p_k : Penalty objective function for production network k.

q^{enum} : Objective function for enumerating alternative solutions.

Binary variables

y_j : Reaction deletion indicator that takes a value of 0 if reaction j is deleted in a modular cell, and 1 otherwise.

z_{jk} : Endogeonous, module-specific reaction indicator that takes a value of 1 if reaction j is added back to the production module in network k, and 0 otherwise.

$d_{jk} = y_j \vee z_{jk}$: Modeling variable which takes a value of 1 if reaction j may carry flux in production network k, and 0 otherwise.

Parameters

S_{ijk} : Stoichiometric coefficient of metabolite i in reaction j of production network k.

l_{jk} : Lower bound flux for reaction j in production network k.

u_{jk} : Upper bound flux for reaction j in production network k.

α : Maximum number of deletion reactions in a modular cell.

β_k : Maximum number of endogenous module-specific reactions in the module of production network k.

ϵ : Small scalar used for tilting the biomass objective function to obtain the minimum product rate available at the maximum growth rate. In the simulation, we used $\epsilon = 0.0001$.

1.2 Solver

We used the *gamultiobj()* solver, an implementation of NSGA-II [1], from the MATLAB Optimization Toolbox to solve our combinatorial multiobjective optimization, formulated as an unconstrained multiobjective problem of the form:

$$\max(p_1, p_2, \ldots, p_{|\mathcal{K}|})^T \tag{1.1}$$

with a *bitstring* population type where each individual is a binary vector corresponding to the design variables y_j (reaction deletions) and z_{jk} (endogenous module-specific reactions). To enforce the constraints on the number of deletion and endogenous module reactions, a penalty function p_k, instead of f_k, (Section 1.3) and customized genetic operators (Section 1.6) were used, respectively.

1.3 Penalty Objective function

To restrict the maximum number of reaction deletions, we optimized the following penalty function p_k instead of f_k:

$$p_k = \begin{cases} \dfrac{f_k}{\sum\limits_{j \in C}(1 - y_j)} & \text{if } \sum\limits_{j \in C}(1 - y_j) > \alpha \\ f_k & \text{otherwise} \end{cases} \tag{1.2}$$

The penalty function is designed to decrease f_k of an individual proportionally to the number of deletion reactions exceeding the set limit α. Implementation of this penalty function helps the optimization problem converge rapidly because favorable deletion reaction candidates are likely kept to obtain desirable solutions. After simulation, only solutions satisfying the maximum reaction deletion constraint are preserved to obtain the Pareto set of our original problem.

1.4 Design objective computation

Depending on desirable applications, the following design objectives are considered:

$$f_k^{wGCP} = \frac{v_{Pk}^{\mu}}{v_{P_{max}k}^{\mu}} \in [0, 1], \qquad \forall k \in \mathcal{K} \tag{1.3}$$

$$f_k^{sGCP} = \frac{v_{Pk}^{\mu}}{v_{P_{max}k}^{\mu}} \frac{v_{Pk}^{\bar{\mu}}}{v_{P_{max}k}^{\bar{\mu}}} \in [0,1], \forall k \in \mathcal{K} \tag{1.4}$$

$$f_k^{NGP} = \frac{v_{Pk}^{\bar{\mu}}}{v_{P_{max}k}^{\bar{\mu}}} \in [0,1], \qquad \forall k \in \mathcal{K} \tag{1.5}$$

In (1.3)-(1.5), the terms v_{Pk}^{μ}, $v_{P_{max}k}^{\mu}$, $v_{Pk}^{\bar{\mu}}$, and $v_{P_{max}k}^{\bar{\mu}}$ are computed by solving the following linear programming problems:

$$\begin{aligned}
v_{Pk}^{\mu} &\in \arg\max\{v_{Xk} - \epsilon v_{Pk} : v_k \in \Pi_k^{\mu}(d_{jk})\} && (1.6)\\
v_{P_{max}k}^{\mu} &\in \arg\max\{v_{Pk} : v_k \in \Pi_k^{\mu}(d_{jk} = 1,\ \forall j \in \mathcal{J}_k)\} && (1.7)\\
v_{Pk}^{\bar{\mu}} &\in \arg\min\{v_{Pk} : v_k \in \Pi_k^{\bar{\mu}}(d_{jk})\} && (1.8)\\
v_{P_{max}k}^{\bar{\mu}} &\in \arg\max\{v_{Pk} : v_k \in \Pi_k^{\bar{\mu}}(d_{jk} = 1,\ \forall j \in \mathcal{J}_k)\} && (1.9)
\end{aligned}$$

The maximum product synthesis fluxes in (1.7) and (1.9), used to normalize the design objectives in (1.3)-(1.5), only need to be computed once for each network prior to solving the multiobjective problem (MOP).

In $(1.6) - (1.9), \Pi_k^{\mu}$ is the space of steady-state reaction fluxes of production network k where a minimum cell growth is required, defined as follows:

$$\begin{aligned}
\Pi_k^{\mu}(d_{jk}) := \{&v_{jk} \in \mathbb{R}\, \forall j \in \mathcal{J}_k :\\
&\sum_{j \in \mathcal{J}_k} S_{ijk} v_{jk} = 0, && \forall i \in \mathcal{I}_k && (1.10)\\
&l_{jk} \le v_{jk} \le u_{jk}, && \forall j \in \mathcal{J}_k && (1.11)\\
&l_{jk} d_{jk} \le v_{jk} \le u_{jk} d_{jk}, && \forall j \in \mathcal{C} && (1.12)\\
&v_{Xk} \ge \text{minimum growth rate}\} && && (1.13)
\end{aligned}$$

Constraints (1.10)-(1.11) correspond to mass balance and flux bounds, as described in the main text. Constraint (1.12) ensures that reaction j cannot carry any flux, if it is deleted in the modular cell and not present in module k. Constraint (1.13) specifies any minimum growth rate requirement.

When the design goals involve the stationary phase (1.8)-(1.9), the space of steady-state reaction fluxes for production network k, $\Pi_k^{\bar{\mu}}$, is defined as follows:

$$\begin{aligned}
\Pi_k^{\bar{\mu}}(d_{jk}) := \{&v_{jk} \in \mathbb{R}\, \forall j \in \mathcal{J}_k :\\
&\sum_{j \in \mathcal{J}_k} S_{ijk} v_{jk} = 0, && \forall i \in \mathcal{I}_k && (1.14)\\
&l_{jk} \le v_{jk} \le u_{jk}, && \forall j \in \mathcal{J}_k && (1.15)\\
&l_{jk} d_{jk} \le v_{jk} \le u_{jk} d_{jk}, && \forall j \in \mathcal{C} && (1.16)\\
&v_{Xk} = 0\} && && (1.17)
\end{aligned}$$

If any of the linear programs associated with f_k becomes infeasible, i.e., $\Pi_k^{\mu} = \emptyset$ or $\Pi_k^{\bar{\mu}} = \emptyset$, then f_k is set to 0.

1.5 Termination criteria

We implemented a non-domination termination criterion to determine when simulation must stop to retrieve a solution, as described in Algorithm 1.

Algorithm 1: Non-domination termination criterion for MOEA. PF: Pareto front, PS: Pareto set.

```
[PF, PS] = solveMOP(initialPoint = $\emptyset$, stall_generations, ...)
total_generations = 0
do
    PF_old = PF
    PS_old = PS
    [PF, PS] = solveMOP(initialPoint = PS_old, stall_generations, ...)
    total_generations = total_generations + stall_generations
while any(PF dominates PF_old) and total_generations $\leq$ max_total_generations
  and run_time $\leq$ max_run_time
```

Based on this criterion, the solution is retrieved if new non-dominated solutions cannot be found for a predefined number of stall generations. For our study, we used highly conservative, empirical values of 500 and 1000 stall generations with runtime limits of 1-2h and 10-15h for core and genome scale models, respectively.

1.6 Customized genetic operators to handle endogenous module-specific reactions

We modified the default scattered crossover and uniform mutation operators of *gamultiobj()* to enforce the constraint on the number of endogenous module reactions and improve convergence. First, we ensured both crossover and mutation operators to produce only individuals that meet the maximum module reaction constraint, i.e., $\sum_{j \in \mathcal{J}_k} z_{jk} \leq \beta_k$, $\forall k \in \mathcal{K}$. Next, we required that only reactions deleted in the modular cell can be used as endogenous module-specific reactions, i.e., $z_{jk} \leq 1 - y_j$, $\forall j \in \mathcal{J}$, $k \in \mathcal{K}$. Finally, we specified the crossover operator to perform crossover on the variables associated with reaction deletions and endogenous module reactions separately, for each production network.

1.7 Parameters

All MOEA parameters, except the population size, were left as default. In our study, we set the empirically conservative values for population sizes of 200 and 400 for core and genome-scale models to converge in 2 h and 15 h of simulation time, respectively.

1.8 Enumeration of alternative solutions

If a solution w produces the same objective vector as a Pareto optimal solution x^*, i.e., $f(w) = f(x^*)$ and $w \neq x^*$, we say that w constitutes an alternative solution of x^*. To enumerate alternative solutions for a specific Pareto optimal design, we iteratively solve a minimization problem of the form:

$$\min q^{enum} \tag{1.18}$$

using MATLAB's genetic algorithm *ga()*. Using the Jaccard similarity metric[1], we define q^{enum}, which takes a value of 0 if an alternative solution is found, as follows:

$$q^{enum} = \begin{cases} M & \text{if } \{y_j : j \in \mathcal{J}_k\} \in \text{ExcludedSol} \quad (1.19) \\ 1 - \mathbf{jacc}(f^*, f) + \sum_{j \in \mathcal{C}} (1 - y_j) & \text{if } \sum_{j \in \mathcal{C}} (1 - y_j) > \alpha \quad (1.20) \\ 1 - \mathbf{jacc}(f^*, f) & \text{otherwise} \quad (1.21) \end{cases}$$

Initially, ExcludedSol will contain at least a target solution for which we are interested in finding alternative solutions. A large scalar M is returned if the current set of deletions $\{y_j\}$ has been found previously, and hence cannot be an alternative solution (1.19). Likewise, a set of deletions $\{y_j\}$ may be a valid solution candidate but have more deletions than allowed (1.20). In that case, the negated Jaccard similarity is penalized according to the number of reaction deletions.

1.9 Optimizing algorithm performance

Variable declaration. To minimize the number of free variables in the optimization problem, we created binary variables, y_j, only for the reaction candidate set C instead of all reactions in the parent model. Similarly, endogenous module reaction variables, z_{jk}, were only created for $j \in \mathcal{C}$ if $\beta_k > 0$.

Selection of starting population. To accelerate convergence in our simulation, we used a predetermined starting population of individuals, if possible. A starting population can be derived from a previously obtained result; for instance, the solutions from $\alpha = 6$ can be used as a starting population to find solutions for $\alpha = 7$. In some cases, we also used design strategies determined experimentally or orginated from other strain design algorithms (e.g., Optknock).

Parallelization. To increase the simulation speed, we performed the objective function computations in parallel. This parallelization alleviated the bottlenecks of solving 1 linear programming problem (LP) in *wGCP* (or *NGP*) and 2 LPs in *sGCP*.

Archive of solutions. Since computing objective functions is one critical bottleneck, we used a table (archive mapping design variables to design objectives) of previously evaluated individuals to avoid repeating this calculation. The size of the table is determined by the amount of memory available. For instance, we stored at most 50,000 solutions, which can be handled for 20 design objectives by a personal computer. When the table becomes full, it is erased to allow for higher quality individuals to be archived.

[1] The Jaccard similarity between vectors r and s, $\mathbf{jacc}(r, s)$, corresponds to the fraction of common elements between r and s. If r and s are the same, the Jaccard similarity is 1; if both vectors do not share any elements, then it takes a value of 0.

2 Specifying the Set of Deletion Reaction Candidates for Manipulation

The set of deletion reaction candidates, C, is a subset of all reactions in the parent network, excluding reactions infeasible to eliminate in practice and irrelevant to desirable phenotypes, as described below. Some criteria used in [2] were adapted and implemented in our study.

Macromolecule-associated reactions. These reactions involve macromolecules whose biologically relevant roles are not well represented in the model (e.g., glycogen) or do not impact the optimal design of target product biosynthesis pathways. We identified macromolecule-associated reactions by screening metabolite IDs and formulas, for instance, those with total carbon number above 10 except currency metabolites (e.g., ATP, acetyl-CoA, etc.)

Non-metabolic reactions. These reactions belong to the functional categories such as ion transport, tRNA charging, etc. We identified these non-metabolic reactions by screening the reaction-subsystem annotation in the parent network.

Modeling reactions. These reactions are sink reactions and/or reactions which are not well characterized. We identified the modeling reactions by screening reaction IDs and sbo terms in the parent network.

Transport reactions. These reactions involve metabolites transported across cellular compartments. Most of these reactions are not included in C due to their unspecific annotation or non-enzymatic mechanisms except some well-annotated reactions such as ATP synthase or NAD(P) transhydrogenase. We identified these transport reactions by screening metabolites appearing in multiple compartments.

Exchange reactions. These reactions are pseudo-reactions used to simulate steady-state conditions. We identified these reactions based on the characteristics they only have either substrates or products.

Orphan reactions. These reactions do not have known encoding enzymes. We determined them by screening gene-protein-reaction associations in the parent model.

Essential reaction. The essential reactions are the reactions whose removal from the model makes the maximum growth rate fall below the minimum acceptable value (i.e, 10-20% of the predicted maximum growth rate). We identified these reactions by performing flux balance analysis combined with single reaction deletions.

Blocked reactions. These reactions carry a flux of 0 mmol/gCDW/hr across all production networks (Figure 1.A). We found these blocked reactions by performing flux variability analysis.

Reactions in fully correlated sets (co-sets). Sets of reactions that have linearly correlated fluxes are classified as co-sets. These reactions can belong to a linear pathway or more than one associated pathways. For each co-set, only one potential candidate reaction is needed to be considered in the reaction deletion candidate set. In our analysis, we considered all co-sets present in a master network, containing all production modules, to prevent potentially useful reaction deletions to be excluded from the candidate set (Figure 1.B). We found the co-sets by flux coupling analysis.

Special consideration for NPG designs. The NGP design objective does not involve the growth phase, unlike wGCP and sGCP. Thus, the set of reaction deletion candidates for NGP designs are determined as outlined above except: i) essential reactions that should be considered in the candidate set and ii) blocked reactions that are determined under non-growth conditions (i.e. biomass flux is constrained to be 0).

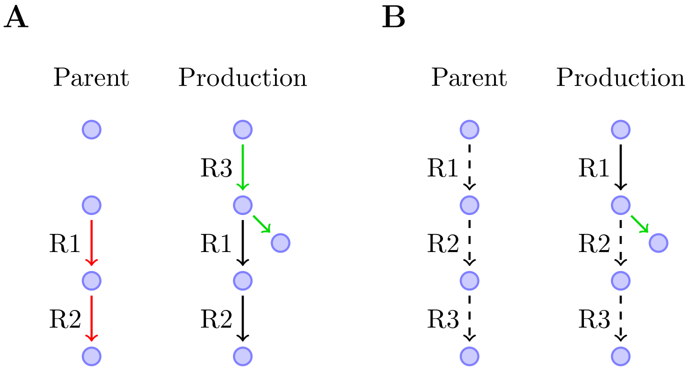

Figure 1: **A.** Blocked reactions. The red arrows represent reactions originally blocked in the parent model, because the substrate of R1 cannot be produced. When heterologus reactions (green arrows) are added to produce a target chemical, the originally blocked reactions may become active and drain an intermediate of the production pathway. **B.** Reaction co-sets. The dashed arrows are used to indicated a fully correlated set. The addition of heterologous reactions (green arrows), alters the co-set definition and has important effect in deletion candidates. If R1 is considered as a deletion candidate, instead of R2 or R3, that would prevent the elimination of a potentially undesired pathway.

References

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.

[2] A. M. Feist, D. C. Zielinski, J. D. Orth, J. Schellenberger, M. J. Herrgard, and B. Ø. Palsson, “Model-driven evaluation of the production potential for growth-coupled products of Escherichia coli,” *Metabolic engineering*, vol. 12, pp. 173–186, may 2010.