OXFORD

Systems biology

NIHBA: a network interdiction approach for metabolic engineering design

Shouyong Jiang [1,*], Yong Wang[2], Marcus Kaiser[3] and Natalio Krasnogor[3,*]

[1]School of Computer Science, University of Lincoln, Lincoln LN6 7TS, UK, [2]School of Automation, Central South University, Changsha 410083, China and [3]School of Computing, Newcastle University, Newcastle upon Tyne NE4 5TG, UK

*To whom correspondence should be addressed.

Associate Editor: Pier Luigi Martelli

Abstract

Motivation: Flux balance analysis (FBA) based bilevel optimization has been a great success in redesigning metabolic networks for biochemical overproduction. To date, many computational approaches have been developed to solve the resulting bilevel optimization problems. However, most of them are of limited use due to biased optimality principle, poor scalability with the size of metabolic networks, potential numeric issues or low quantity of design solutions in a single run.

Results: Here, we have employed a network interdiction model free of growth optimality assumptions, a special case of bilevel optimization, for computational strain design and have developed a hybrid Benders algorithm (HBA) that deals with complicating binary variables in the model, thereby achieving high efficiency without numeric issues in search of best design strategies. More importantly, HBA can list solutions that meet users' production requirements during the search, making it possible to obtain numerous design strategies at a small runtime overhead (typically ~1 h, e.g. studied in this article).

Availability and implementation: Source code implemented in the MATALAB Cobratoolbox is freely available at https://github.com/chang88ye/NIHBA.

Contact: math4neu@gmail.com or natalio.krasnogor@ncl.ac.uk

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

With the advance of genome-scale metabolic modelling (GSMM), the past decades have witnessed a significant number of computational tools for microbial metabolic engineering (Maia *et al.*, 2016). These tools facilitate improved strain performance for the production of a variety of high-value biochemicals and biosynthetic precursors, including vanillin (Brochado *et al.*, 2010), lycopene (Choi *et al.*, 2010), malonyl-CoA (Xu *et al.*, 2011) and alkane and alcohol (Fatma *et al.*, 2018).

A large number of strain design tools are based on bilevel optimization. OptKnock (Burgard *et al.*, 2003) is one of the earliest bilevel optimization-based tools. OptKnock maximizes target chemical production while assuming mutant strains at optimal growth in flux balance analysis (FBA). The resulting bilevel problem is solved through a reformulation that makes the inner level problem equivalent constraints under the condition of strong duality (Burgard *et al.*, 2003). The OptKnock model was latter extended to improve target production via gene up/down-regulation (Pharkya and Maranas, 2006), cofactor specificity (King and Feist, 2014) or heterologous pathways (Pharkya *et al.*, 2004), and to develop anti-cancer drugs by the identification of synthetic lethal genes (Pratapa *et al.*, 2015). These studies demonstrate the great effectiveness of the bilevel optimization-based framework in metabolic engineering.

However, the bilevel optimization-based framework in literature has numerous limitations. The first one is the intensive computational cost in search of optimal solutions. Bilevel optimization is often reformulated into a mixed-integer linear programming (MILP) so as to be solved by exact MILP solvers. It can take up to a week to solve a MILP resulting from a medium-sized GSMM (Feist *et al.*, 2010). Many practical strategies, such as model reduction and refinement of candidate knockout set (Feist *et al.*, 2010), have been used to reduce the computational time but may miss the best design strategies due to reduced search space. GDBB (Egen and Lun, 2012) introduced a truncated branch and bound to speed up the search process. GDLS used local search with multiple search paths to reduce the search space for each local MILP (Lun *et al.*, 2009). While finding optimal solutions are computationally costly for exact solvers, other studies resorts to inexact methods, such as genetic algorithms (Patil *et al.*, 2005; Rocha *et al.*, 2010) and swarm intelligence (Choon *et al.*, 2015). These methods, however, still scale poorly with the size of GSMM and are specially ineffective when a large number of genetic manipulations are allowed for target production, which is a

widely recognized issue for large-scale optimization (Piccand *et al.*, 2008).

In company with intensive computations, the resulting MILP often has weak LP relaxations due to disjunctive big-M constraints (Codato and Fischetti, 2006), another limitation of the current bilevel optimization-based methods. Big-M formulation can easily cause numeric issues, particularly in genome-scale metabolic models where stoichiometric coefficients often vary many orders of magnitude (Sun *et al.*, 2013). As a result, optimal strain design solutions returned from exact MILP solvers in Gurobi and/or CPLEX may turn out to be *in silico* infeasible. Model reformulation may alleviate numeric issues and potentially reduce computational costs. However, a proper model reformulation is often time-consuming and laborious as extra care has to be taken to prevent other numeric difficulties while fixing one.

The third limitation is that only a single solution is obtained in each execution of optimization by modern solvers. Multiple runs are required to generate more solutions, which inevitably increases computational burden. Despite that some commercial software platforms, such as Gurobi and CPLEX, provide options to preserve multiple solutions in a single run, many alternative solutions exist for the optimal production rate in underdetermined metabolic systems, and they are likely to have similar production envelopes (Lewis *et al.*, 2012). This similarity renders the bilevel optimization less attractive as little information can be gained about the trade-off between growth and production for decision-making. Also, these exact solvers consider two solutions different when their continuous but integer variables have different values. Such solutions lead to same design strategies, which is of no interest to decision-makers. Heuristic methods, such as local search in GDLS (Lun *et al.*, 2009) and population-based algorithms (Jiang *et al.*, 2018; Patil *et al.*, 2005), may help to find diverse solutions but often suffer from local optima.

Another limitation in most of bilevel optimization-based tools is potential biases induced by the optimization principle in the inner-level FBA (Lewis *et al.*, 2012). OptKnock and many of its derivatives assume mutant strains have a biologically meaningful objective which is often to maximize growth. However, this assumption is not always correct as some microorganisms seem to achieve a multiobjective trade-off of metabolism (Schuetz *et al.*, 2012), and mutants prefer small metabolic adjustments from the wild-type (Segrè *et al.*, 2002). It is there desirable that bilevel models eliminate the biased assumption on cell growth while optimizing the target production rate.

There are also a number of tools free of bilevel optimization. For example, approaches based on minimum cut set (MCS) identification have also been developed to remove all possible design strategies that do not meet specific requirements of a desired production strain(Apaolaza *et al.*, 2019; von Kamp and Klamt, 2014). MCS-based approaches, however, are also computationally intensive as they need to enumerate elementary modes of a given metabolic network. OptForce identifies genetic interventions by investigating the difference in flux distributions between the wild-type and the desired mutant (Ranganathan *et al.*, 2010). OptForce showed good predictions in *in vivo* studies (Xu *et al.*, 2011). However, the requirement on flux measurements of the wild-type, which is not always available, limits its wide applicability. It is noteworthy that strain design has been viewed as a multiobjective optimization problem in a number of studies (Sendin *et al.*, 2010; Sendín *et al.*, 2006; Torres *et al.*, 2018). These studies highlight the benefit of finding trade-off solutions that comprise multiple design objectives.

Evolutionary game theory for metabolic modelling has achieved great success, especially in situations not governed by simple optimization (Pusa *et al.*, 2019). It also seems useful for strain design since it avoids the assumption of growth optimality in FBA. For this reason, this article attempts to investigate the application of evolutionary game theory for strain design. Specifically, we consider metabolic engineering as a metabolic game (Pusa *et al.*, 2019), and employ network interdiction (NI), a well-known game theory model (Lim and Smith, 2007), for the identification of genetic manipulations. The NI involves one game player (host strain) that tries to avoid the overproduction of target chemicals for cellular homoeostasis, whereas the other opposing player (metabolic engineer) attempts to manipulate the metabolic network in order to maximally disrupt the first player's activity. Therefore, the NI is a max–min problem in which the objective involves only the target production, avoiding the use of the widely assumed growth optimality. The NI is a special case of general bilevel problems. The solution to this NI problem is a novel hybrid algorithm based on Benders decomposition (Codato and Fischetti, 2006), aiming to address the other limitations mentioned previously. NIHBA, the proposed approach, has shown the ability to efficiently find a large number of growth-coupled design strategies with diverse production envelopes in a single run and to scale well with the size of allowable knockouts.

2 Results

2.1 NIHBA: using NI and benders decomposition

More often than not, wild-type strains maintain homoeostasis and thus avoid overproducing a product of interest while maximizing biomass (Fig. 1A). Metabolic engineering of host strains requires metabolic network modifications leading to improved flux towards the biosynthetic pathway of the target product. Metabolic engineers who look for the best modifications can be considered adversaries or interdictors as they use limited engineering costs (time, financial cost, etc.) to intervene the host's activities and reverse the cell's low-target production (Fig. 1A). This strain design task is similar to NI, a game theoretic framework where a budget-constrained interdictor intervenes a network user's activity, e.g. commodity flow (Lim and Smith, 2007), by removing network arcs.

We proposed a NI model for identifying gene-associated reaction knockouts, but up-/down-regulation of genes can be considered in this model as well. The NI model is a special case of bilevel optimization. It was recast into a standard MILP problem (Fig. 1B) using a special reformulation approach (Section 4). The resulting MILP contains both complicating binary variables and easy continuous variables. It can be computationally intensive for a large size of binary variables and/or a high allowable number of knockouts, and likely to have numeric issues for exact solvers due to Big-M effects (Codato and Fischetti, 2006). We, therefore, resorted to Benders decomposition for this NI model. We proposed a hybrid Benders algorithm (HBA) with two novel techniques to solve the model efficiently and obtain as many design solutions as possible in a single run (Fig. 1C). The solutions from our approach, NIHBA, were then analysed in production envelopes (Fig. 1D), from which the most promising design solution can be selected for implementation.

2.2 Case studies

Our case studies investigate the production of both native biochemicals (i.e. succinate and ethanol) closely linked to energy metabolism and a non-native secondary metabolism product (i.e. lycopene).

2.2.1 Succinate and ethanol production

NIHBA was tested on iML1515 (Monk *et al.*, 2017), the largest Genome-scale metabolic (GEM) model for Escherichia coli, for the production of succinate and ethanol. For large models, Feist *et al.* (2010) suggested a preprocessing procedure to ease computational costs. In particular, reactions involving compounds that have more than a certain number (n_c) of carbons are not considered as knockout candidates since they are assumed unlikely to carry high flux. We tested different values of $n_c = \{10, 15, 22, 100\}$ ($n_c = 100$ indicates no reaction removed due to carbons), resulting in different sizes of candidate set (see n_s in Table 1). In all simulations, designed strains were required to have at least 10% cell growth of the wild-type to sustain growth (Feist *et al.*, 2010).

We show in Table 1 that the reduction of candidate set by carbon number has a significant effect on succinate production. Exclusion of reactions with a carbon number of over 10 (corresponding to 152 candidate knockouts in Table 1) results in a low succinate production flux of 6.8838–9.9430 mmol/gDW/h, which is
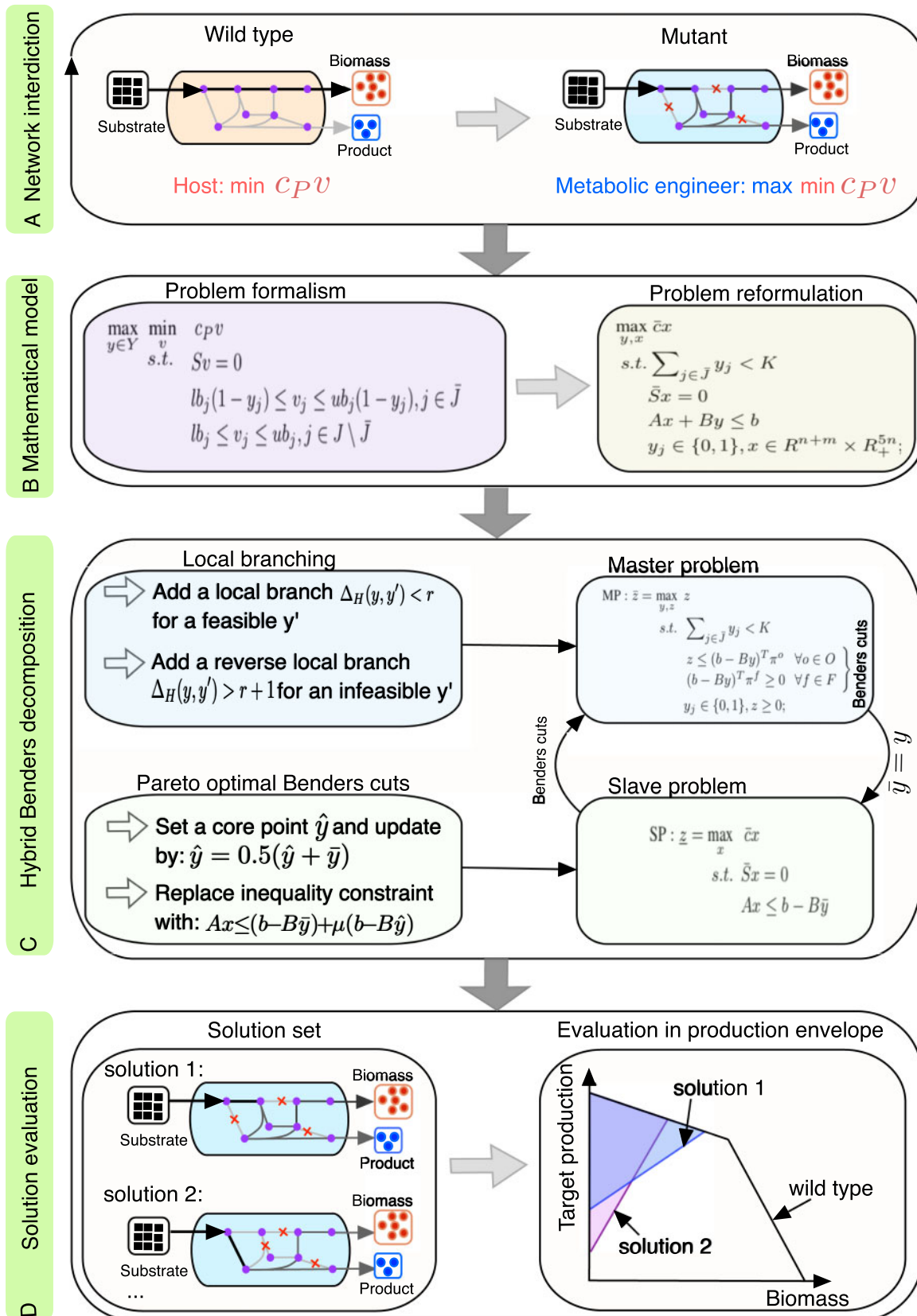
Fig. 1. A schematic workflow of the proposed NIHBA tool for strain design. (**A**) Illustration of network interdiction in strain design: host cells avoid overproducing a product (i.e. min c_{Pv}) whereas metabolic engineers interdict the host network to maximally impair the host's activity (i.e. max min c_{Pv}), where c_P is the coefficient vector for the product and v is a steady-state flux vector. (**B**) Mathematically modelling the network interdiction problem in strain design, followed by problem reformulation to obtain a standard MILP problem. (**C**) Hybrid Benders decomposition algorithm. The MILP is decomposed into a binary master problem and a linear slave problem, and Pareto-optimal cut generation and local branching are introduced to speed up the search of solutions. (**D**) Solutions that meet production requirements are stored and evaluated

Table 1. Succinate and ethanol production predicted by NIHBA with at most five knockouts for different sizes of candidate set

n_c	n_s	Succinate				Ethanol			
		#sol.	Growth	Min. prod.	Max. prod.	#sol.	Growth	Min. prod.	Max. prod.
10	152	1255	0.8729	6.8838	9.9430	21	0.1238	38.0668	38.0986
15	204	186	0.2023	21.1032	24.3682	37	0.1238	38.0668	38.0986
22	272	271	0.1908	22.2077	25.0055	52	0.1238	38.0668	38.0986
100	342	172	0.1549	25.0252	25.0660	84	0.1187	38.1716	38.2466

The growth rate (h^{-1}), minimum production (mmol/gDW/h) and maximum production (mmol/gDW/h) are from the solution with the highest minimum production rate.

less than a third of the theoretic maximum production (TMP; Feist *et al.*, 2010), which is computed by changing the FBA objective from growth maximization to target reaction flux maximization in iML1515). A slight relaxation of carbon number to 15 helps to identify a solution with around two-thirds of TMP, and succinate production reaches $\sim$25 mmol/gDW/h (73% TMP) when no carbon number is constrained in candidate reactions. This indicates that some reactions with large carbon numbers are very important in redirecting flux towards succinate, although they may not carry high flux values. For example, both pyruvate dehydrogenase (PDH) and pyruvate formate lyase (PFL) acting on a high carbon compound, i.e. Acetyl-CoA, provide a good reaction flux value in wild-type strains. The knockout of them together with transaldolase (TALA, pentose phosphate pathway) and L-lactate dehydrogenase (LDH_L) under anaerobic condition (where the oxygen uptake rate is zero) predicts high succinate production by our method (Fig. 2). This prediction agrees well with *in vivo* studies (Hong and Lee, 2002; Zhu *et al.*, 2007). It is also observed that many solutions are found by NIHBA, and all of them renders a growth-coupled production phenotype. Particularly, there exist a vast number of growth-coupled solutions with low succinate production rate, as indicated when the size of the candidate knockout set is $n_s = 152$ (see Table 1). Most of them do not appear in larger candidate sets, implying that NIHBA prefers high-production solutions. In the following, a set of 342 candidate reactions is used for NIHBA.

In contrast to succinate, ethanol can be easily produced at a high rate (e.g. 38 mmol/gDW/h or equivalently 95% TMP) by knocking

down only reactions of low carbon number. Therefore, excluding as many reactions of high carbon number is beneficial for computational efficiency while achieving a high-production design strategy. The number of growth-coupled solutions for ethanol is, however, much fewer than that for succinate.

Next, we analyse the performance of NIHBA on a varying number of knockouts. We observe that, for both succinate and ethanol, the production rate increases sharply when more knockouts are allowed but levels out from five knockouts (Fig. 3A and B). The number of growth-coupled solutions and the percentage of high-production solutions increase with the allowable number of knockouts. No solutions with >80% TMP were found within 15 knockouts for succinate, and only a tiny portion of solutions has a production rate of <20% TMP for ethanol. This indicates again that NIHBA favours high-production solutions during the search.

We show in Figure 4 the distribution of single knockouts of obtained solutions in different subsystems and their frequency in design solutions to understand which subsystems/knockouts are likely to be engineering targets. Specifically, all the solutions with >60% TMP identified from a limit of 10 knockouts were analysed in terms of knockout occurrence in different subsystems. We find that extracellular exchange reactions (mainly oxygen uptake), reactions from glycolysis, gluconeogenesis, pyruvate metabolism and pentose phosphate pathway are most likely knockout targets for the production of succinate and ethanol, which shows a good agreement with existing studies (Feist *et al.*, 2010; Tepper and Shlomi, 2010). All the solutions suggest oxygen depletion (Fig. 4C), which is consistent with the common sense that the two products are best produced in anaerobic environments. It is observed that glucose-6-phosphate isomerase and D-lactate dehydrogenase (LDH_D) appear frequently as knockout targets in design solutions for both products, and high succinate-producing strategies additionally disfavour ATP synthase (ATPS4rpp) (Fig. 4C). Similar results have been also reported in a recent study (Dinh *et al.*, 2018), where computationally intensive strain design tools were used to obtain a large collection of knockout designs.

The computational cost is low for NIHBA, as shown in Figure 3C and D. A short runtime (300–600 s) enables NIHBA to identify high-production solutions for both succinate and ethanol, indicating that our hybrid approach can quickly generate effective Benders cuts to reduce search space. Depending on the maximum allowable number of knockouts, the runtime required before reaching the convergence stage is different, but with a small variation. Generally, NIHBA starts to converge after $\sim$2000 s and 600 s for succinate and ethanol, respectively. The longer time required for succinate may be explained by relatively fewer high-production solutions in the design space. Despite that, the computational time required by NIHBA for a good solution is small (compared to days $\sim$weeks in existing methods (Feist *et al.*, 2010)) and does not increase exponentially with the number of knockouts, a widely recognized issue in exact solvers (Feist *et al.*, 2010; Lun *et al.*, 2009).

2.2.2 Lycopene biosynthesis
A heterologous lycopene biosynthesis pathway, as reported in Alper *et al.* (2005) and Choi *et al.* (2010), was added to the iML1515 model to predict lycopene production (Fig. 5). When eight
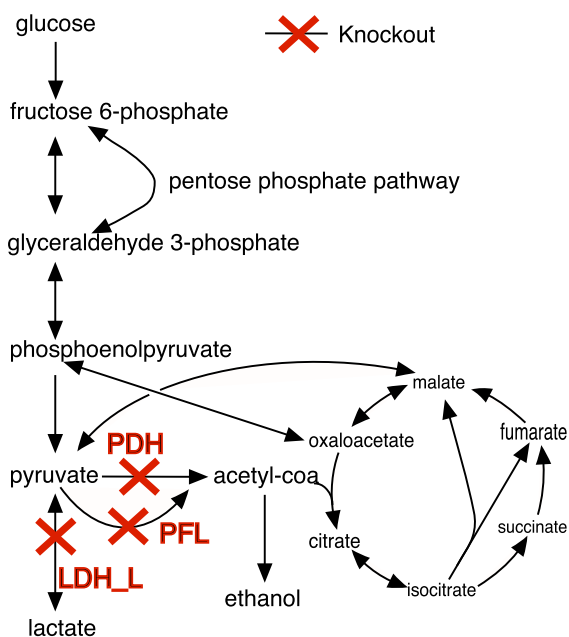

Fig. 2. Result of NIHBA simulation for succinate production. Abbreviations of reactions are as follows: PDH, pyruvate dehydrogenase; PFL, pyruvate formate lyase; LDH-L, L-lactate dehydrogenase
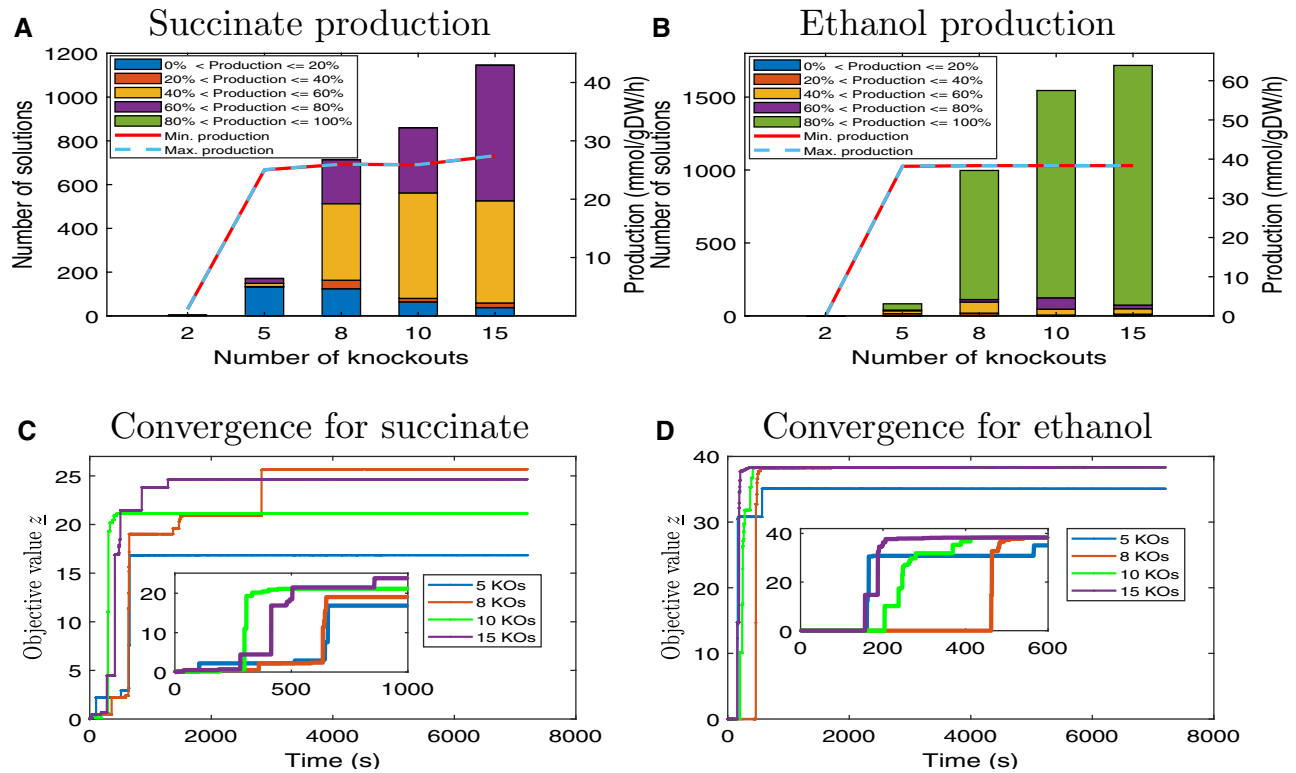
Fig. 3. Performance of NIHBA with different number of knockouts for succinate and ethanol production. (**A**) For succinate, the number of solutions achieving a certain percentage of the maximum theoretic production rate, and the minimum and maximum production rate of the best solution at optimal growth found for a different number of knockouts. (**B**) For ethanol, the number of solutions achieving a certain percentage of the maximum theoretic production rate, and the minimum and maximum production rate of the best solution at optimal growth found for a different number of knockouts (KOs). (**C**) The objective value of NIHBA against runtime for succinate production. (**D**) The objective value of NIHBA against runtime for ethanol production
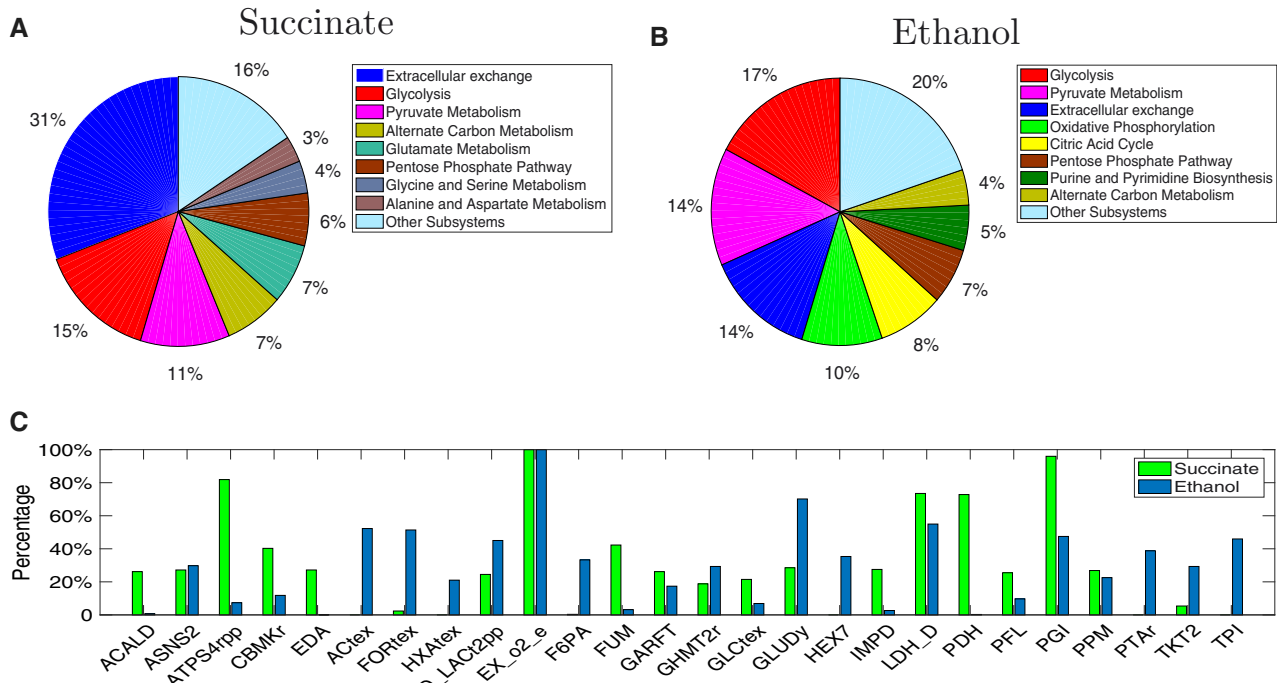

Fig. 4. Knockout distribution for succinate and ethanol production. (**A**) The percentage of solutions with at least one knockout from each subsystem for succinate production. (**B**) The percentage of solutions with at least one knockout from each subsystem for ethanol production. (**C**) The fraction of design solutions that have a specific knockout (only knockouts with at least a percentage 20% for either target products are displayed)
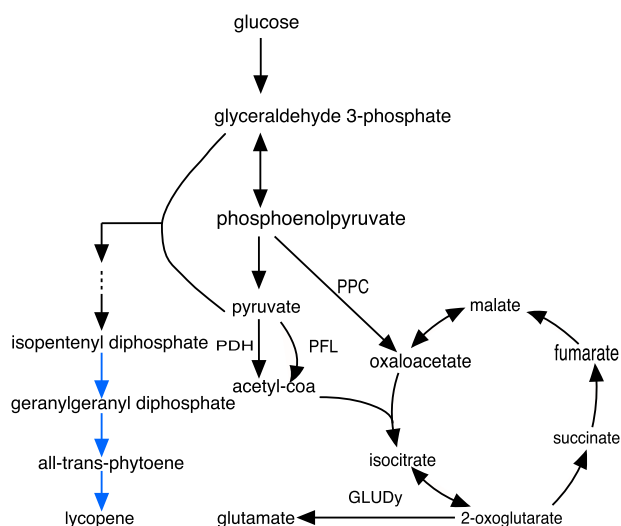
Fig. 5. Lycopene biosynthesis pathway (heterologous reactions are marked in blue). Abbreviations of reactions are as follows: PDH, pyruvate dehydrogenase; PFL, pyruvate formate lyase; PPC, phosphoenolpyruvate carboxylase; GLUDy, glutamate dehydrogenase. (Color version of this figure is available at *Bioinformatics* online.)

Fig. 6. Comparison of MOMA and NIHBA in obtaining the best design solutions for succinate production. The plots are production envelopes which illustrate the maximum and minimum guaranteed production rate while varying biomass. Black curves represent the wild-type and coloured curves represent production strains. (**A**) Production envelope of the best solution identified by MOMA. (**B**) Production envelope of the best solution identified by NIHBA

knockouts are allowed, running NIHBA on this expanded model generates a large number of solutions, with lycopene production rate ranging from 0.88 to 1.60 mmol/gDW/h (42–78% TMP) (see Supplementary Material A). Interestingly, most of the knockouts, e.g. PDH and PFL, are closely linked to pyruvate, which is a key upstream building blocks for lycopene biosynthesis. This suggests that increasing the availability of precursors could lead to high lycopene production. Apart from these, NIHBA also identified some non-intuitive knockouts, such as ribose-5-phosphate isomerase, glycine hydroxymethyltransferase (GHMT2r), phosphoenolpyruvate carboxylase (PPC) and glutamate dehydrogenase (GLUDy). It is worth noting that the knockout of GLUDy, PPC, GHMT2r and PDH identified by NIHBA has also been predicted by other methods (Choi *et al.*, 2010), but NIHBA shows more diverse combinations of these reactions as manipulation strategies. Additionally, although lycopene biosynthesis interferes less with cell growth, the simulation suggests NIHBA is still able to manipulate the metabolic network properly to lower growth capability (maximum growth rate) so that more substrate resources are available potentially for lycopene production. Reduced growth capability is undesired for a production system, however, it can be alleviated by adaptive laboratory evolution (Jensen *et al.*, 2019) since the designed production mutants are all growth coupled.

2.3 Comparison with other tools

2.3.1 Comparison with minimization of metabolic adjustment
The minimization of metabolic adjustment (MOMA) (Segrè *et al.*, 2002) has demonstrated great success in predicting genetic deletion targets for improving production strains. Here, NIHBA is compared with MOMA in identifying at most five reaction knockouts for succinate production. For MOMA, a sequential approach (Alper *et al.*, 2005) was used to identify multiple knockout solutions.

The best solutions (see Supplementary Material B) identified by MOMA and NIHBA are compared by their production envelopes, as shown in Figure 6. The production envelopes help us understand the production variability as growth increases. Figure 6 shows that the production strain designed by NIHBA has a significantly reduced maximum growth rate and the guaranteed lower bound of succinate production rate is > 5 mmol/gDW/h, regardless of growth rate. Thus, this is a strong growth-coupled design (Feist *et al.*, 2010). In contrast, the MOMA solution shows slightly reduced maximum growth rate. However, the succinate production rate for the MOMA solution varies widely, and the guaranteed lower bound is zero. This implies, although the solution identified by MOMA
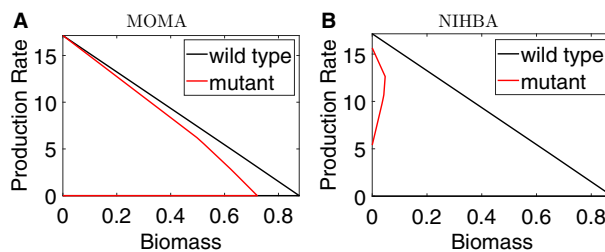
guarantees minimal metabolic adjustment, the production rate can be zero in the resulting mutant strain.

This simulation demonstrates that a simple optimization principle, such as minimal metabolic adjustments in MOMA, cannot ensure that the resulting production strain yields improved biochemical production. In contrast, a more rigorous model like NIHBA considering the equilibrium between multiple players in metabolic engineering games clearly works better.

2.3.2 Comparison with bilevel optimization-based tools
For comparison, the NI problem was also solved using the OptKnock (Burgard *et al.*, 2003) and GDLS (Lun *et al.*, 2009) approaches with the Gurobi MILP solver(Gurobi Optimization, 2018), called NI-OptKnock and NI-GDLS, respectively. NI-GDLS used $M = 5$ search paths and a search size of $k = 3$ in order to get multiple solutions. For efficiency, parameters in the Gurobi solver was set according to Egen and Lun (2012).

For succinate, when at most five knockouts are allowed, NIHBA found a large number of solutions whereas both NI-OptKnock and NI-GDLS failed to find a feasible solution. The failure is mainly due to numeric issues in Big-M formulation, which existed even although we switched to indicator constraints or CPLEX12.8 for MILP. This demonstrates that HBA overcomes such numeric issues. For readability, we only show a small number of selected solutions from NIHBA in the production envelope (Fig. 7A). As seen, NIHBA can obtain diverse solutions forming a good representative of the trade-off between cell growth and succinate production. Interestingly, NIHBA identified a strong growth-coupled design (non-zero production at no growth), despite its slightly suboptimal production rate at the maximum growth.

For ethanol, all the algorithms found feasible solutions with at most five knockouts, and all contain a solution with the maximum production rate, as illustrated in Figure 7B–D (a small portion of solutions from NIHBA are displayed for readability). This shows that NIHBA has comparable performance in terms of optimality. Despite three solutions found from NI-GDLS, one of them is not growth coupled and the other two have the same production envelope, from which little can be gained about the trade-off between cell growth and target production. In contrast, NIHBA found many solutions with diverse production envelopes, among which strong growth-coupled designs exist.

3 Discussion

The employment of bilevel optimization for identifying genetic manipulations has been found very helpful for metabolic engineering. Most existing bilevel-based tools assume that cells always grow optimally, a biased optimization principle that is found incorrect for mutants or certain microorganisms in some studies (Schuetz *et al.*, 2012; Segrè *et al.*, 2002). As a consequence, design strategies found by these tools may be biologically infeasible in spite of highest production rates at optimal growth. In addition, these tools involve
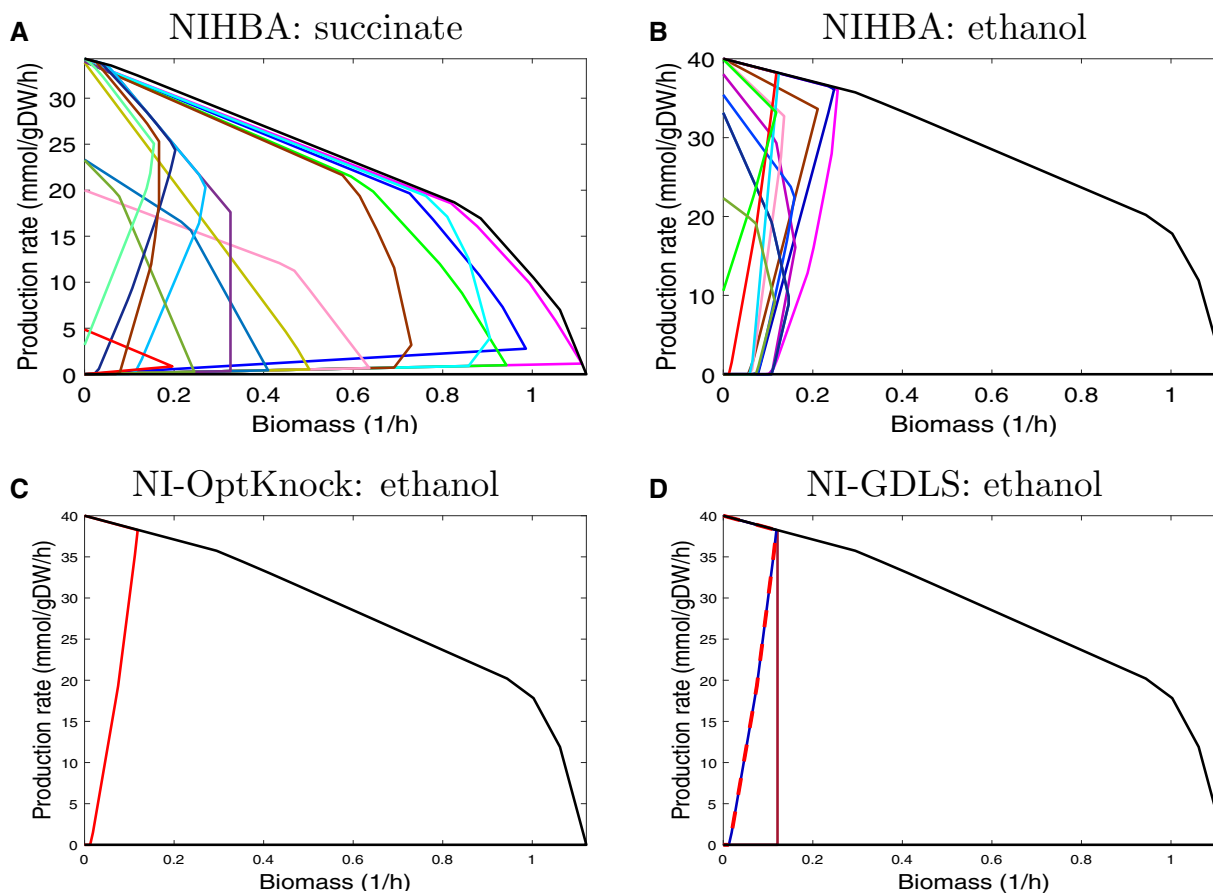
Fig. 7. Production envelopes of diverse design solutions obtained by different strain design approaches for succinate and ethanol production. Production envelope illustrates the maximum and minimum guaranteed production rate while varying biomass. Black curves represent the wild-type and coloured curves represent production strains. (**A**) Production envelopes of selected solutions by NIHBA for succinate production. (**B**) Production envelopes of selections solutions by NIHBA for ethanol production. (**C**) Production envelope of the solution by NI-OptKnock for ethanol production. (**D**) Production envelopes of the three solutions by NI-GDLS for ethanol production

solving a bilevel problem through big-M reformulation to a standard MILP that is suitable for exact solvers of commercial software like Gurobi and CPLEX. However, the resulting MILP is often large due to the genome scale of metabolic networks, and exact solution to MILP can be computationally prohibitive, particularly when a large design space (or numerous genetic manipulations) is allowed. Furthermore, big-M formulation produces a weak MILP, leading to numeric issues such that no feasible solutions can be found. This article have proposed to address biased assumptions from the point of view of game theory, leading to a network interdiction problem (NIP). The NIP is not handled using popular exact solvers, instead it is solved through an efficient hybrid Benders decomposition algorithm to lower computational costs and overcome numeric issues. The proposed approach, NIHBA, has shown its ability to obtain a large number of growth-coupled design strategies with diverse production phenotypes and achieve optimal production rates within an hour or so, regardless of the size of design space (the maximum allowable number of knockouts).

NIHBA uses a game theoretic framework to model the interaction (somehow competitive) between host cells and metabolic engineers. This framework assumes that host cells have a few objectives. These objectives are not necessarily to optimality individually but reach a trade-off between them. In this sense, NIHBA is different from traditional FBA approaches which often require a single biologically rigorous optimality objective, such as optimal growth or maximum energy generation (Schuetz *et al.*, 2012). Therefore, design strategies found by NIHBA do not necessarily yield the best production at optimal growth. Instead, they guarantee non-zero production when the cell achieves a minimal required growth rate to

sustain growth (Feist *et al.*, 2010). NIHBA employs an HBA, HBA, to solve the NIP. Our case studies have demonstrated numerous advantages of this algorithm. First, it is free of numeric issues, making it much more stable than exact MILP solvers in top-ranked optimization platforms, e.g. Gurobi and CPLEX. Second, it can be considered a parameter-free algorithm as opposed to other methods like GDLS that requires a setting of multiple parameters, although NIHBA uses a parameter μ for identifying Pareto optimal cuts. In practice, NIHBA is not sensitive until a value of $\mu < 10^{-6}$ is used. Third, it is computationally efficient such that 1 h on average is sufficient for NIHBA to identify high-production solutions, and the runtime for a high-production rate does not scale with the number of knockouts, which is not the case for existing methods. Last, it obtains numerous growth-coupled solutions in a single run. This is important as it not only helps understand the trade-off between target production and cell growth but also provides the possibility to examine and test multiple solutions, from which the most promising design can be chosen for experimental implementation.

In computational strain design, a model reduction procedure (Feist *et al.*, 2010) is often employed to reduce the search space for computational efficiency. One important step in this strategy is to exclude reactions acting on high-carbon metabolites. Many existing strain design tools rely on a predefined carbon number to reduce the number of candidates so that the resulting MILP has fewer binary variables (Feist *et al.*, 2010). As a result, optimal solutions may be eliminated. This work has observed this issue in the case study of succinate production. Accordingly, NIHBA suggests to discard the high-carbon reaction reduction step. In this sense, NIHBA is more likely to identify optimal solutions compared with other strain

design tools. It should be noted that the relaxation of search space can lead to increased runtime of MILP solvers. However, this has been alleviated by an efficient HBA in NIHBA.

The proposed HBA is not limited to NIPs. It can be applied to any bilevel or single-level optimization problems that have complicating mixed-integer variables. Although promising, HBA needs improvements on convergence at late stages for optimality proof. Like other exact solvers, an appropriate optimality gap or time limit may alleviate excessive exploration but cannot determine the optimality of solutions. Further improvements can be made along this direction to enhance the convergence of HBA. It is also noteworthy that multiple solutions by HBA are not searched in a systematic way. They may not form a perfect representative of the trade-off between target production and cell growth. Therefore, more investigations are required to extract limited but well-diversified solutions in the search process of HBA.

Despite numerous solutions found by NIHBA, the selection of promising solutions poses a new challenge to decision-makers. It is therefore important to have a good solution ranking approach. Solutions may be roughly ranked according to the frequency of individual knockouts in addition to their subsystem distribution or by a scoring system with manual settings (Schneider and Klamt, 2019). Thus, a more systematic solution ranking is desirable. Another limitation of this work is the use of constraint-based models. While constraint-based models make it possible to investigate large-scale metabolic networks, they do not capture the dynamic nature of biologic systems. Further investigations are needed to make NIHBA applicable to dynamic models or hybrid models (Kim *et al.*, 2018) for better metabolic engineering applications.

4 Materials and methods

4.1 Flux balance analysis

A metabolic network of m metabolites and n reactions has a stoichiometric matrix S that is formed by stoichiometric coefficients of the reactions. Let J be a set of n reactions and v_j the reaction rate of $j \in J$, Sv represents the concentration change rates of the m metabolites. FBA aims at optimizing a linear biological objective $c^T v$ when the system is at steady state (i.e. the concentration change rate is zero for all the metabolites) and v is subject to thermodynamic constraints:

$$\begin{aligned} \max_v \quad & c^T v \\ s.t. \quad & Sv = 0 \\ & lb_j \leq v_j \leq ub_j, j \in J \end{aligned}, \tag{1}$$

where lb_j and ub_j are the lower and upper flux bounds of reaction j, respectively. c is a weight vector specifying the degree of importance to the biological objective.

4.2 Network interdiction-based strain design and reformulation

NI for strain design considers metabolic engineers as interdictors or adversaries who attempt to maximally disrupt host cells' activity that biochemicals of interest are not overproduced due to homoeostasis. The strain design task can, therefore, be formulated as a max–min problem:

$$\max_{y \in Y} \quad \min_v \quad c_P v, \tag{2a}$$

$$s.t. Sv = 0, \tag{2b}$$

$$lb_j(1 - y_j) \leq v_j \leq ub_j(1 - y_j), j \in \bar{J}, \tag{2c}$$

$$lb_j \leq v_j \leq ub_j, j \in J \setminus \bar{J}, \tag{2d}$$

where c_P is a coefficient vector for the target biochemical. That is, c_P is a vector of zeros except for the P-th element (the index of the target biochemical reaction) which is set to one. $Y =$

$\{y \in \{0,1\}^{|\bar{J}|} | \sum_{j \in \bar{J}} y_j < K\}$ (K is the maximum allowable number of knockouts), and y_j indicates the reaction j is inactive ($v_j = 0$) if $y_j = 1$ and active otherwise. $\bar{J}$ is a subset of J, containing candidate knockout reactions.

Observing that in the follower problem $|v_j| y_j = 0$ always holds for all $j \in J$, we can eliminate all the flux constraints imposed by y_j, i.e. Equation (2c), by rephrasing the inner objective function in a Lagrangian manner:

$$\min_v \quad c_P v + \sum_{j \in \bar{J}} M_j |v_j| y_j, \tag{3a}$$

$$\begin{aligned} s.t. \quad & Sv = 0 \\ & lb_j \leq v_j \leq ub_j, j \in J \end{aligned} \Big\}, \tag{3b}$$

where M_j is a large positive Lagrange multiplier and $M = (M_1, \ldots, M_{|\bar{J}|})$. The reformulated follower problem is equivalent to the original problem in the sense that they have the same optimal value provided that M_j is sufficiently large for all $j \in \bar{J}$ such that $v_j = 0$ when $y_j = 1$. The value of M_j used in this work is around 100 (e.g. randomly drawn from [90,110]).

The reformulated follower function (3a) can be linearized by adding auxiliary variables $u_j = max(v_j, -v_j)$. As a result, we have the reformulated bi-level framework:

$$\max_{y \in Y} \quad c_P v, \tag{4a}$$

$$s.t. \quad \min_{v,u} \quad c_P v + \sum_{j \in \bar{J}} M_j u_j y_j, \tag{4b}$$

$$\begin{aligned} s.t. \quad & Sv = 0 \\ & u_j \geq v_j, u_j \geq -v_j, j \in \bar{J} \\ & lb_j \leq v_j \leq ub_j, j \in J \end{aligned} \Big\}. \tag{4c}$$

4.3 Hybrid benders algorithm

The bi-level problem (4) is reformulated to a standard MILP by applying LP duality to the follower problem (4b–4c). For simplicity, the resulting MILP is written in the following compact form

$$\max_{y,x} \quad \bar{c}x \tag{5a}$$

$$s.t. \sum_{j \in \bar{J}} y_j < K \tag{5b}$$

$$\bar{S}x = 0 \tag{5c}$$

$$Ax + By \leq b \tag{5d}$$

$$y_j \in \{0,1\}, x \in R^{n+m} \times R_+^{5n}; \tag{5e}$$

where $\bar{S} = [S \quad 0_{m,5n+m}]$, and

$$A = \begin{bmatrix} -I & 0 & 0 & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 & 0 \\ I & 0 & -I & 0 & 0 & 0 & 0 \\ -I & 0 & -I & 0 & 0 & 0 & 0 \\ 0 & S^T & 0 & I & -I & I & -I \\ c_P & 0 & 0 & -lb^T & ub^T & I & I \end{bmatrix}, B = \begin{bmatrix} 0 \\ diag(U) \\ 0 \\ 0 \\ -diag(M) \\ 0 \end{bmatrix}, b = \begin{bmatrix} 0 \\ U \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{6}$$

where U is a vector of maximum absolute flux for each reaction, i.e. $U = max(|lb_j|, |ub_j|), \quad \forall j \in J$.

The single-level reformulation (5) can be solved, like OptKnock, by modern MILP solvers. However, the big-M terms in (5) lead to a week LP relaxation (Codato and Fischetti, 2006), therefore, causing difficulties for MILP solvers. Besides, the model size of (6) increases rapidly for large metabolic networks, and as a result, a large-scale MILP has to be solved.

Benders decomposition avoids these drawbacks as it can deal with complicating binary variables and easy continuous variables separately. Like Benders decomposition (Codato and Fischetti, 2006), our HBA decomposes (5) into a binary integer programming master problem (MP) (7) and an LP slave problem (SP) (8) for fixed $y = \bar{y}$:

$$\text{MP}: \bar{z} = \max_{y,z} \quad z \tag{7a}$$

$$s.t. \quad \sum_{j \in \bar{J}} y_j < K \tag{7b}$$

$$\left. \begin{array}{ll} z \leq (b - By)^T \pi^o & \forall o \in O \\ (b - By)^T \pi^f \geq 0 & \forall f \in F \end{array} \right\} \text{ Benders cuts} \tag{7c}$$

$$y_j \in \{0, 1\}, z \geq 0; \tag{7d}$$

$$\text{SP}: {}_z = \max_x \quad \bar{c}x \tag{8a}$$

$$s.t. \quad \bar{S}x = 0 \quad [\lambda] \tag{8b}$$

$$Ax \leq b - B\bar{y} \quad [\pi] \tag{8c}$$

where O and F are sets that correspond to the extreme points π^o and extreme rays π^f of the dual of SP, respectively. In each iteration, the Benders decomposition algorithm derives the dual vector π from the SP (8) for y ($y = \bar{y}$) which is the solution to the MP in the previous iteration. In practice, a Benders cut is obtained by solving the dual of (8) rather than the primal. Two scenarios exist when solving the dual of (8): if the optimal value of the dual of SP is bounded, it means the SP is feasible, then an optimality cut $z \leq (b - By)^T \pi^o$ generated from the extreme point π^o is added to the MP; if it is unbounded, it means the SP is infeasible, then a feasibility cut $(b - By)^T \pi^f \geq 0$ generated from the extreme ray π^f is added to the MP to avoid unboundedness of the dual of SP in future iterations.

The classic Benders decomposition is not able to generate effective Benders cuts rapidly for our strain design problem, and therefore, requires a huge of iterations (consequently long computation time) before it converges. Here, we introduce an HBA with two strategies to speed up the convergence process.

Figure 8 shows a simplified flowchart of HBA. An implementation of the algorithm in MATLAB can be found in https://github.com/chang88ye/NIHBA.

4.3.1 Pareto optimal cuts
Let π^o be the dual vector of π corresponding to (8), a standard Benders optimality cut is:

$$Cut(\pi^o): z \leq (b - By)^T \pi^o \tag{9}$$

Since π^o may not be unique, it is important to select an effective cut $Cut(\pi^o)$. Magnanti and Wong (1981) proposed to use Pareto optimal cuts to improve convergence. $Cut(\pi^o)$ is said to be Pareto optimal if no other $Cut(\bar{\pi}^o)$ exists such that $(b - By)^T \bar{\pi}^o \leq (b - By)^T \pi^o$ for any $y \in Y$ and at least one $y \in Y$ enables a strict inequality. There are a few methods available for identifying a Pareto optimal cut, but most of them have to solve the SP (8) twice, which may increase computational time significantly. We turn to the approach of Sherali and Lunday (2013) where a Pareto optimal cut can be generated by solving only once in each iteration a slightly different SP:

$$\bar{SP}: \underline{z} = \max_x \quad \bar{c}x \tag{10a}$$

$$s.t. \quad \bar{S}x = 0 \quad [\lambda] \tag{10b}$$

$$Ax \leq (b - B\bar{y}) + \mu(b - B\hat{y}) \quad [\pi] \tag{10c}$$

where μ is a sufficiently positive value and $\hat{y}$ is a core point in the
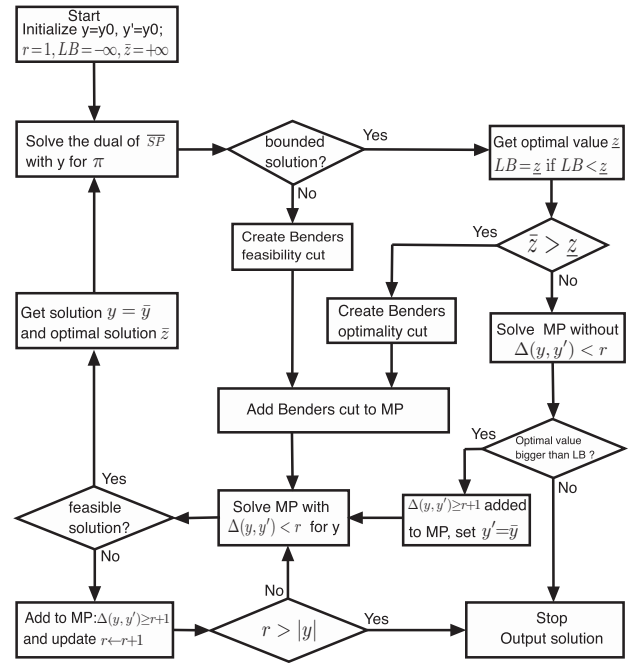

Fig. 8. Flowchart of HBA

relative interior of the convex hull of Y. In this paper, $\hat{y}$ is updated by $0.5(\hat{y} + \bar{y})$ whenever a new feasible $y = \bar{y}$ is produced in the iteration of Benders decomposition. μ is not calculated as in Sherali and Lunday (2013) but rather fixed to 1e−8 after multiple trials.

4.3.2 Local branching
Another technique we used for accelerating the convergence of Benders decomposition is local branching, which is particularly effective when problems have binary variables (Baena et al., 2018; Rei et al., 2009). Suppose y' is a feasible solution in Y, the idea behind local branching is to divide the feasible region of (7) into two subregions by the Hamming distance between y and y':

$$\Delta_H(y, y') = \sum_{y'_j = 0} y_j + \sum_{y'_j = 1} (1 - y_j). \tag{11}$$

In every iteration of Benders decomposition, the MP (7) is solved in the subregion $\Delta_H(y, y') < r$ (where r is a positive integer and the maximum is the cardinality $|y|$ of y). This leads to two scenarios: there is either a feasible or infeasible y in the subregion $\Delta_H(y, y') < r$. If a feasible solution $\bar{y}$ is obtained, Benders cuts are generated by solving (10) with $\bar{y}$. If not, it means the value of r may be too small, and $\Delta_H(y, y') > r + 1$ is added to the MP (7) to stop re-exploration in the neighbourhood of y' with the radius r. r is then increased by one at a time until $\Delta_H(y, y') < r$ renders the MP (7) feasible. Note that y' has to be updated by $\bar{y}$ if $\bar{y}$ gives (7) an objective value worse than that of the SP (8), implying that no better solution can be obtained from the neighbourhood of y'.

4.4 Additional improvement strategies
HBA involves solving the MP (7) and SP (8) in a repeated manner. For efficiency, the following two strategies are used:

- Terminating the MP program prior to optimality. Suboptimal solutions to the MP are sufficient to generate valid Benders cuts. Therefore, the MP is terminated when a MIP Gap of $1 + 300/(\text{iter}^{0.5} + 1)$ (where *iter* is the iteration counter) is reached.
- Reversing local branching whenever the $\bar{z}$ value of the MP is worse than $\underline{z}$ value of the SP. $\bar{z}$ estimates the upper bound of the problem (2). $\bar{z} < \underline{z}$ indicates the global optimum does not exist

in the corresponding local branching and a reverse local branching should, therefore, be used.

4.5 Model reduction and candidate selection

The truncation of model size and candidate knockout set has great computational benefits. GEM models can be significantly simplified by compressing linear reactions and removing dead end reactions (those carrying zero fluxes). Likewise, many reactions can be excluded from consideration with *a priori* knowledge that, for example, they are vital for cell growth or their knockout is not likely to improve target production. We followed the model reduction procedure by Lun *et al.* (2009) and candidate selection procedure by Feist *et al.* (2010), resulting in a candidate set of 150–350 reactions for different target products from the latest *E.coli* GEM iML1515 (Monk *et al.*, 2017) where the maximum uptake rates for glucose and oxygen are all 20 mmol/gDW/h.

4.6 Computational implementation

First of all, all the NI models were transformed into MILPs using duality theory (Burgard *et al.*, 2003). Then, the resulting MILPs were implemented in MATLAB 2018b to be compatible with the Cobra Toolbox 3.0 (Heirendt *et al.*, 2019) where we carried out simulations. All the MILPs were solved by Gurobi 7.5 (Gurobi Optimization, 2018) with both Heuristics and MIPFocus were set to 1 as suggested by Egen and Lun (2012). A time limit of 2 h was applied to each MILP while performing computations on Ubuntu 16.04 LTS with an Intel® CoreTM i5 Quad Core processor.

Data and software availability

The data and software used and the tool developed are all available online:

- GEM model: iML1515 from BIGG database (bigg.ucsd.edu).
- Simulation software: Cobra toolbox 3.0 (https://opencobra.github.io/).
- MILP solver: http://www.gurobi.com/.
- NIHBA: https://github.com/chang88ye/NIHBA.

Funding

References

Alper,H. *et al.* (2005) Identifying gene targets for the metabolic engineering of lycopene biosynthesis in *Escherichia coli*. *Metab. Eng.*, **7**, 155–164.

Apaolaza,I. *et al.* (2019) gMCS: fast computation of genetic minimal cut sets in large networks. *Bioinformatics*, **35**, 535–537.

Baena,D. *et al.* (2018) Stabilized benders methods for large-scale combinatorial optimization, with application to data privacy. *Ann. Oper. Res.*, **271**, 11–85.

Brochado,A.R. *et al.* (2010) Improved vanillin production in baker's yeast through *in silico* design. *Microb. Cell Fact.*, **9**, 84.

Burgard,A.P. *et al.* (2003) Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnol. Bioeng.*, **84**, 647–657.

Choi,H.S. *et al.* (2010) *In silico* identification of gene amplification targets for improvement of lycopene production. *Appl. Environ. Microbiol.*, **76**, 3097–3105.

Choon,Y.W. *et al.* (2015) Gene knockout identification using an extension of bees hill flux balance analysis. *BioMed Research International*, **2015**, 1–10.

Codato,G. and Fischetti,M. (2006) Combinatorial benders' cuts for mixed-integer linear programming. *Oper. Res.*, **54**, 756–766.

Dinh,H.V. *et al.* (2018) Identification of growth-coupled production strains considering protein costs and kinetic variability. *Metab. Eng. Commun.*, **7**, e00080.

Egen,D. and Lun,D.S. (2012) Truncated branch and bound achieves efficient constraint-based genetic design. *Bioinformatics*, **28**, 1619–1623.

Fatma,Z. *et al.* (2018) Model-assisted metabolic engineering of *Escherichia coli* for long chain alkane and alcohol production. *Metab. Eng.*, **46**, 1–12.

Feist,A.M. *et al.* (2010) Model-driven evaluation of the production potential for growth-coupled products of *Escherichia coli*. *Metab. Eng.*, **12**, 173–186.

Gurobi Optimization,LLC. (2018) Gurobi optimizer reference manual, http://www.gurobi.com.

Heirendt,L. *et al.* (2019) Creation and analysis of biochemical constraint-based models: the cobra toolbox v3. 0. *Nat. Protoc.*, **14**, 639–702.

Hong,S. and Lee,S. (2002) Importance of redox balance on the production of succinic acid by metabolically engineered *Escherichia coli*. *Appl. Microbiol. Biotechnol.*, **58**, 286–290.

Jensen,K. *et al.* (2019) OptCouple: joint simulation of gene knockouts, insertions and medium modifications for prediction of growth-coupled strain designs. *Metab. Eng. Commun.*, **8**, e00087.

Jiang,S. *et al.* (2018) Improving microbial strain design via multiobjective optimisation and decision making. In *Proceedings 27th Int. Joint Conf. Art. Intell. & 23rd Euro. Conf. Art. Intell. (IJCAI-ECAI)*, pp. 1–10.

Kim,O.D. *et al.* (2018) A review of dynamic modeling approaches and their application in computational strain optimization for metabolic engineering. *Front. Microbiol.*, **9**, 1690.

King,Z.A. and Feist,A.M. (2014) Optimal cofactor swapping can increase the theoretical yield for chemical production in *Escherichia Coli* and *Saccharomyces cerevisiae*. *Metab. Eng.*, **24**, 117–128.

Lewis,N.E. *et al.* (2012) Constraining the metabolic genotype–phenotype relationship using a phylogeny of in silico methods. *Nat. Rev. Microbiol.*, **10**, 291–305.

Lim,C. and Smith,J.C. (2007) Algorithms for discrete and continuous multi-commodity flow network interdiction problems. *IIE Trans.*, **39**, 15–26.

Lun,D.S. *et al.* (2009) Large-scale identification of genetic design strategies using local search. *Mol. Syst. Biol.*, **5**, 296.

Magnanti,T.L. and Wong,R.T. (1981) Accelerating benders decomposition: algorithmic enhancement and model selection criteria. *Oper. Res.*, **29**, 464–484.

Maia,P. *et al.* (2016) *In silico* constraint-based strain optimization methods: the quest for optimal cell factories. *Microbiol. Mol. Biol. Rev.*, **80**, 45–67.

Monk,J.M. *et al.* (2017) iML1515, a knowledgebase that computes *Escherichia coli* traits. *Nat. Biotechnol.*, **35**, 904–908.

Patil,K.R. *et al.* (2005) Evolutionary programming as a platform for *in silico* metabolic engineering. *BMC Bioinformatics*, **6**, 308.

Pharkya,P. and Maranas,C.D. (2006) An optimization framework for identifying reaction activation/inhibition or elimination candidates for overproduction in microbial systems. *Metab. Eng.*, **8**, 1–13.

Pharkya,P. *et al.* (2004) OptStrain: a computational framework for redesign of microbial production systems. *Genome Res.*, **14**, 2367–2376.

Piccand,S. *et al.* (2008) *On the Scalability of Particle Swarm Optimisation*. In Proceedings 2008 IEEE Congress on Evolutionary Computation, pp. 2505–2512.

Pratapa,A. *et al.* (2015) Fast-SL: an efficient algorithm to identify synthetic lethal sets in metabolic networks. *Bioinformatics*, **31**, 3299–3305.

Pusa,T. *et al.* (2019) Metabolic games. *Front. Appl. Math. Stat.*, **5**, 18.

Ranganathan,S. *et al.* (2010) OptForce: an optimization procedure for identifying all genetic manipulations leading to targeted overproductions. *PLoS Comput. Biol.*, **6**, e1000744.

Rei,W. *et al.* (2009) Accelerating benders decomposition by local branching. *INFORMS J. Comput.*, **21**, 333–345.

Rocha,I. *et al.* (2010) OptFlux: an open-source software platform for in silico metabolic engineering. *BMC Syst. Biol.*, **4**, 45.

Schneider,P. and Klamt,S. (2019) Characterizing and ranking computed metabolic engineering strategies. *Bioinformatics*, **35**, 3063–3072.

Schuetz,R. *et al.* (2012) Multidimensional optimality of microbial metabolism. *Science*, **336**, 601–604.

Segrè,D. *et al.* (2002) Analysis of optimality in natural and perturbed metabolic networks. *Proc. Natl. Acad. Sci. USA*, **99**, 15112–15117.

Sendin,J.O.H. *et al.* (2010) Multi-objective mixed integer strategy for the optimisation of biological networks. *IET Syst. Biol.*, **4**, 236–248.

Sendín,O.H. *et al.* (2006) Model based optimization of biochemical systems using multiple objectives: a comparison of several solution strategies. *Math. Comput. Model. Dyn. Syst.*, **12**, 469–487.

Sherali,H.D. and Lunday,B.J. (2013) On generating maximal nondominated benders cuts. *Ann. Oper. Res.*, **210**, 57–72.

Sun,Y. *et al.* (2013) Robust flux balance analysis of multiscale biochemical reaction networks. *BMC Bioinformatics*, **14**, 240.

Tepper,N. and Shlomi,T. (2010) Predicting metabolic engineering knockout strategies for chemical production: accounting for competing pathways. *Bioinformatics*, **26**, 536–543.

Torres,M. *et al.* (2018) Strain design as multiobjective network interdiction problem: a preliminary approach. In Herrera,F. *et al.* (eds) *Advances in Artificial Intelligence*. Springer International Publishing, Cham, pp. 273–282.

von Kamp,A. and Klamt,S. (2014) Enumeration of smallest intervention strategies in genome-scale metabolic networks. *PLoS Comput. Biol.*, **10**, e1003378.

Xu,P. *et al.* (2011) Genome-scale metabolic network modeling results in minimal interventions that cooperatively force carbon flux towards malonyl-coa. *Metab. Eng.*, **13**, 578–587.

Zhu,Y. *et al.* (2007) Homolactate fermentation by metabolically engineered *Escherichia coli* strains. *Appl. Environ. Microbiol.*, **73**, 456–464.