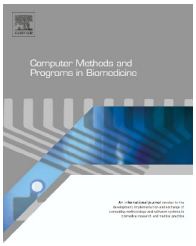
ELSEVIER

journal homepage: www.intl.elsevierhealth.com/journals/cmpb



Development and application of efficient pathway enumeration algorithms for metabolic engineering applications

F. Liu [a], P. Vilaça [a,b], I. Rocha [a], M. Rocha [a,*]

[a] Centre for Biological Engineering, University of Minho, Campus Gualtar, 4710-057 Braga, Portugal
[b] SilicoLife Lda., Rua do Canastreiro 15, 4715-387 Braga, Portugal

ARTICLE INFO

Article history:
Received 28 July 2014
Received in revised form
31 October 2014
Accepted 26 November 2014

Keywords:
Synthetic biology
Optimal pathway design
Pathway enumeration
Hypergraphs
Constraint-based modeling
Metabolic engineering

ABSTRACT

Metabolic Engineering (ME) aims to design microbial cell factories towards the production of valuable compounds. In this endeavor, one important task relates to the search for the most suitable heterologous pathway(s) to add to the selected host. Different algorithms have been developed in the past towards this goal, following distinct approaches spanning constraint-based modeling, graph-based methods and knowledge-based systems based on chemical rules. While some of these methods search for pathways optimizing specific objective functions, here the focus will be on methods that address the enumeration of pathways that are able to convert a set of source compounds into desired targets and their posterior evaluation according to different criteria. Two pathway enumeration algorithms based on (hyper)graph-based representations are selected as the most promising ones and are analyzed in more detail: the Solution Structure Generation and the Find Path algorithms. Their capabilities and limitations are evaluated when designing novel heterologous pathways, by applying these methods on three case studies of synthetic ME related to the production of non-native compounds in *E. coli* and *S. cerevisiae*: 1-butanol, curcumin and vanillin. Some targeted improvements are implemented, extending both methods to address limitations identified that impair their scalability, improving their ability to extract potential pathways over large-scale databases. In all case-studies, the algorithms were able to find already described pathways for the production of the target compounds, but also alternative pathways that can represent novel ME solutions after further evaluation.

1. Introduction

In the last decades, the quest for sustainable industrial processes has driven an increased interest in Industrial Biotechnology. Typically, to reach acceptable levels of productivity in these processes, there is the need to re-engineer the microbes' metabolism [1]. The main goal of Metabolic Engineering (ME) is to identify the most suitable genetic alterations to impose to host microbes, to make them fit for the production of valuable compounds. The development of microbial cellfactories usually requires an iterative

* *Corresponding author*. Tel.: +351 253604456.
E-mail address: mrocha@di.uminho.pt (M. Rocha).

http://dx.doi.org/10.1016/j.cmpb.2014.11.010
0169-2607/

process involving several steps, including the search for suitable hosts. In many cases, the selected hosts do not possess the ability to conduct the full set of necessary chemical transformations or these do not fulfill the desired properties (e.g., in terms of productivity or yield). In these cases, the insertion of heterologous pathways allows to augment the hosts' capabilities to produce non-native compounds.

Advances in algorithms and computational tools have provided automated methods to predict viable pathways for either biodegradation or biosynthesis of valuable compounds [2]. However, the complexity of this task is quite challenging, given the dimension of the search spaces that are imposed by the growing size of the databases containing metabolic data, such as for instance the Kyoto Encyclopedia of Genes and Genomes (KEGG) [3,4] and MetaCyc [5].

Metabolic pathway optimization is not a novel topic and a few methods have been proposed over the past decades. Until now, most available literature is either based on graph-based methods or rule-based (or knowledge-based) systems. An alternative comes from the use of constraint-based modeling (CBM) approaches that have gained considerable importance within ME. These alternatives will be discussed below, highlighting their main features and limitations.

In graph-based representations, compounds and/or reactions are represented as graph nodes, being compounds connected to reactions through their role as substrates and products, defining the direction of the graph edges [6]. Path searching algorithms are used to extract minimal length sequences of transformations between compounds with the purpose of identifying viable pathways.

There are several limitations that arise from graph-based representations of metabolic networks. In most scenarios, the shortest path between two compounds in a graph does not represent a biological meaningful path, since chemical reactions usually contain cofactors and pool metabolites (e.g., *ATP*, *NAD*, H_2O, H^+). The high connectivity of these compounds reroutes the shortest path (that is directly translated from a metabolic network) to favor pool metabolites, which in most cases leads to biologically meaningless solutions [7].

Distinct alternatives have been proposed to address this issue. One solution to overcome this problem is to strip cofactors and pool metabolites (also known as currency metabolites) from the network, leaving most reactions with a single substrate and a single product. This, however, involves user expertise and manual curation of the network. Also, by removing the entire set of currency metabolites, it is impossible to obtain solutions that are able to synthesize these compounds (e.g., ATP).

An alternative is to apply weights to each compound node based on their degree [7]. Compounds with high degree are penalized, allowing shortest path methods to find the proper route avoiding currency metabolites. Nonetheless, false positives remain a problem, but compared to the previous solution, the usage of compound weights does not require chemical knowledge about the content of the network.

Perhaps the most accurate method is to use chemical knowledge about the compounds to induce the correct transition between the main substrates and products of a reaction, distinguishing from co-factors and other secondary metabolites. The atom tracking approach [8–10] uses the chemical structure of the compounds and identifies conservation of atoms in chemical reactions. This allows to track, for instance, the conservation of carbon atoms between substrates and products, and therefore the conservation of carbon atoms in an entire pathway. This approach is able to generate core substrate-product pairs that together assemble the full reaction. An example is provided by the KEGG RPAIR database which contains metabolite pairs of the KEGG Reaction database, which allow to prune the network [11]. In a biological sense, the use of this knowledge within shortest path algorithms leads to more meaningful solutions.

Alternatively to topological methods, rule-based approaches share a common trait with the atom tracking approach, as they both use chemical structures to infer pathways. These approaches apply rules over the chemical compounds to generate reactions. This allows not only to identify pathways but also to infer novel reactions [12,13]. The advantage to infer novel pathways comes with the price of increased computational complexity due to large number of hypothetical reactions. Also, these systems require a higher degree of validation [13].

Besides the issues related to network pruning discussed above, the graph-based systems analyzed are usually limited to linear paths over the graph. This is an important limitation since many relevant biochemical reactions have two or more substrates and/ or compounds. One alternative to overcome this limitation is the implementation of further techniques to infer branched pathways over regular graphs. One example is provided by the ReTrace method [14].

The use of a more complex graph structure allows to overcome many problems related to directed graph search. Hypergraphs or process graphs (which are similar to directed bipartite graphs) are so called set systems representations, which are capable to model chemical reactions with higher detail. This allows to address the problem of multiple products and reactants, since edges connect to vertex sets instead of a single vertex.

Process graphs were used by Friedler et al. [15,16] in an exhaustive approach for decision mapping in synthesis processes through the Solution Structure Generation (SSG) algorithm [17], being later adapted for pathway identification [18]. More recently, the work of Carbonell et al. [19] introduced Find Path, an enumeration strategy to extract pathways using hypergraphs. Both algorithms are enumeration approaches that attempt to list all possible pathways towards the desired target. Given their core role in this work, they are further explored in detail in the next sections of the paper.

Orthogonally to the aforementioned approaches, constraint based modeling (CBM) has been often adopted for *in silico* analysis of genome-scale metabolic models (GSMM), since it does not require kinetic information. Using this approach, the system is subjected to several constraints, such as reaction stoichiometry and reversibility. Typically, mainly for phenotype simulation purposes, systems are assumed to be in pseudo-steady state [20], allowing the computation of a feasible flux space. Flux Balance Analysis (FBA) is a popular method to determine the flux distribution that maximizes a target objective (e.g., related to cellular growth) using linear programming [21].

Among many other applications within pathway optimization, FBA has been used to determine producible non-native compounds [22] by merging GSMMs with large databases such as KEGG, allowing to infer putative heterologous reactions for defined purposes. Also within the CBM framework, the OptStrain algorithm [23] searches within a domain of reactions and metabolites (e.g., coming from a database such as KEGG) for the pathway with the smallest number of heterologous reactions that allows to produce the target compound. Since the constrains used in this case are different, that determines the need to resort to mixed integer linear programming (MILP).

As their main advantage, CBM based approaches avoid the combinatorial explosion of possible pathways in graph-based methods, through optimization based on a selected objective function. Furthermore, the constraints imposed in the system are able to guarantee that the obtained solutions are stoichiometrically valid. However, a limitation is the capability to determine only a single solution and, therefore, in this regard have similar limitations to the shortest path approaches based in regular graphs. Indeed, these methods do not enumerate exhaustively other alternative solutions, which may offer valuable information on alternative routes.

Still within the CBM framework, Elementary Flux Modes (EFM) of a metabolic model are defined as non-reducible subsets of reactions that can maintain steady state. The enumeration of the EFMs in a network that include the desired target provides an enumeration of all possible pathways producing this compound. However, their computation is still restricted to small or medium models, being impossible to extend to GSMMs [24]. de Figueiredo et al. [25] propose an enumeration strategy to compute the k-shortest EFMs expanding the size of computable problems, but still the enumeration is computationally expensive and restricted to small values of k. Indeed, database size networks (e.g., KEGG or MetaCyc) still offer an impossible challenge for exhaustive EFM computation. For large-scale networks (e.g., GSMMs), the only option is to apply heuristics to reduce the search space or to use stochastic approaches [26].

In this scenario, given the advantages and limitations of the proposed methods, researchers have to choose the best option for their particular task. The field of ME has been resorting to CBM approaches in their quest for improved microbial cell factories. However, in many cases, it is difficult to define a suitable objective function for pathway optimization as multiple criteria need to be taken into account. Also, given the complexity of the problems and underlying biological phenomena, it is highly desirable to be able to identify alternative solutions leading to the desired products. As a result, we opted to focus our attention in the most promising methods using set systems representations, in which a network is represented as explained above by a set of sets. Indeed, these are able to overcome limitations of shortest path approaches over regular graphs, while providing the means to address the enumeration of multiple solutions for pathway optimization problems.

In this work, two previously identified algorithms for multiple pathway enumeration are analyzed: the Solution Structure Generation (SSG) and the Find Path (FP) algorithms. Both operate over set system representations (process graphs and hypergraphs, respectively). These are implemented, evaluated and improved through three case studies, regarding the production of butanol, vanillin and curcumin, using as hosts the bacterium *Escherichia coli* and the yeast *Saccharomyces cerevisiae*, two model organisms for which there are available GSMMs. The results obtained by both are provided and discussed, being clear the need to introduce some improvements to allow the scalability of the methods.

The next section introduces a more formal definition of the problem and related concepts; the following section details the SSG and FP algorithms and the improvements developed in this work; the next section details the case studies, some implementation issues and the experimental setup; next, the results are presented and discussed; and the paper closes with some conclusions and further work.

2. Problem definition

In a topological approach, a pathway extraction problem can be defined as a dependency problem. Thus, a reaction needs to be satisfied and satisfies metabolites (that are dependencies of other reactions), which correspond to reactants and products, respectively. Here, the notation used in the following is defined. Mostly, it is based on the axioms and algorithms presented in [15–17].

2.1. Metabolic network and its components

Networks will be composed only by metabolites and reactions. In this system, metabolites are the vertex entities, while reactions are represented by an ordered pair $\langle M_1, M_2\rangle$, that connects two disjoint sets of metabolites.

Definition 1. (Reaction) A reaction is an ordered pair $\langle M_1, M_2\rangle$ of two disjoint sets of metabolites (i.e., $M_1 \cap M_2 = \emptyset$). The first set represents the reactants, while the second represents the products.

Definition 2. (Metabolic Network) A metabolic network Σ is a pair composed by a set of metabolites Π and a set of reactions Υ.

A reversible reaction r is represented by including another entity r', such that the metabolite sets are swapped. Additionally, a network $\Sigma' = \langle \Pi', \Upsilon'\rangle$ is defined as a subnetwork of $\Sigma\langle \Pi, \Upsilon\rangle$ if every element of Σ' is contained in Σ (i.e., $\Pi' \subseteq \Pi$ and $\Upsilon' \subseteq \Upsilon$), then $\Sigma' \subseteq \Sigma$.

2.2. Synthetic metabolic problem

A retrosynthetic metabolic problem can be defined as follows:

Definition 3. (Retrosynthetic Metabolic Problem) A retrosynthetic metabolic problem Γ is defined by a triplet $\langle \Sigma, S, T\rangle$, where Σ is a metabolic network that represents the search space, while S and T are two disjoint sets of metabolites (i.e., $S \cap T = \emptyset$) which are the constraints of the heterologous pathways. The set S keeps the initial substrates (e.g., supplies or raw materials), while the set T defines the target compounds of interest.

A heterologous pathway is a set of reactions, in most cases a subnetwork of a larger network (defined as the search space), that satisfies the following conditions.

Definition 4. (Heterologous Pathway) A heterologous pathway σ of a synthetic problem Γ is any network (or subnetwork) $\Sigma = \langle M, R \rangle$, such that: (a) the product set T is included in M, i.e., $T \subset M$ and (b) for every metabolite m in the subnetwork that is not included in the substrate sets of Γ (i.e., $M - S$) there is a reaction r in R such that m is a product of r.

The heterologous pathway definition is not sufficient to guarantee that the solution is feasible in steady state, because it omits the stoichiometry of the reactions. Both algorithms addressed in this work do not take into account this property for the computation of heterologous solutions. This eventually will lead to the computation of unfeasible solutions that later can be verified by applying FBA.

3. Algorithms

In this section, a detailed description of the algorithms addressed in this work, SSG and FP, is provided. In both cases, the original algorithm will be described first, together with the limitations found. Afterwards, the proposed improvements towards better computational efficiency and scalability will be described.

3.1. Solution Structure Generation

3.1.1. Original algorithm description

The Solution Structure Generation (SSG) algorithm (shown as Algorithm 1) enumerates heterologous pathways of Γ by recursively branching all possible combinations. This technique, denoted as decision mapping, can be described as follows: let Σ' be a subnetwork such that condition (a) in Definition 4 verifies. Then, in order to fulfill condition (b), the sub-problem Γ' is solved producing the unsatisfied metabolites in Σ'. Given for example $\Sigma = \langle T, \emptyset \rangle$, a network containing T and no reactions, then a) trivially verifies. Then, $\wp(\text{producersof } t), t \in T$ where $\wp(X)$ denotes the power set of X, are candidates for partial solutions of Γ, since if solutions of Γ exist, at least one element of $\wp$ eventually must be present in one or more solutions of Γ. Recursively, we solve the sub-problem Γ', with the new target set $T' = R - S - M$, where R is the set of reactants of the newly introduced reactions (minus the initial set S and producible metabolites in the partial solution), until eventually either there are no possible reactions to add (this implies that we have reached a dead end that happens when we pick a producer of T that does not belong to any solution) or $T' = \emptyset$ which implies that we achieved a solution.

There are several limitations of the SSG method. The first is the high amount of memory that is required to compute power sets which grow exponentially with the number of elements (2^n). Additionally, this generates an extensive amount of possible combinations. If the network is not pruned, meaning that the network contains reactions that do not belong to any solution, then the algorithm may contain branches that return no solutions and depending on the depth of these branches, this increases severely the computation time to obtain solutions. Friedler et al. [16] proposed a polynomial algorithm to prune process graphs to remove all reactions that might exhibit such behavior. Because of these limitations, in the next section, we propose some modifications to the original algorithm in order to be able to compute larger networks.

3.1.2. Improving SSG by computing minimal solution heuristics

The major bottleneck of the SSG algorithm is the computation of the power set (line 6 in Algorithm 1). Furthermore, because of the union closure property of the solutions, it implies that every combination of two distinct solutions σ_α and σ_β is also a solution (i.e., $\sigma_\alpha \cup \sigma_\beta$ is a valid solution). This severely increases the amount of candidate solutions and the computation complexity of the problem.

Algorithm 1. Solution Structure Generation

```
1:  procedure SSG(T, M, δ[M])
2:    if T = ∅ then
3:      return δ[M] ▷ δ[M] is a solution structure
4:    end if
5:    let x ∈ P
6:    C ← ℘(Δ(x)) \ {∅} ▷ Generate all combinations of Δ(x)
7:    for c ∈ C do ▷ For each combination test if is valid
8:      if ∀y ∈ m, c ∩ δ̄(y) = ∅ ∧ (Δ(x)\c) ∩ δ(y) = ∅ then
9:        δ[m ∪ {x}] ← δ[m] ∪ {(x, c)}
10:       SSG((p ∪ φ⁻(c)) \ (R ∪ m ∪ {x}), m ∪ {x}, δ[m ∪ {x}])
11:     end if
12:   end for
13:   return
14: end procedure
```

We propose modifications to this algorithm in such way that: (a) we compute only minimal solutions; and (b) we generate partitions of the power set instead of generating the entire set. A minimal solution is a solution that satisfies the steady state condition and no reaction can be removed from it.

In this scenario, the solutions obtained from the SSG algorithm are closely related to EFMs of a metabolic network, with a few exceptions: an EFM must obey the stoichiometry and the steady state assumptions. Since SSG performs only topological analysis, it is likely that a few solutions are unfeasible at steady-state (these can be later evaluated and discarded). It is interesting to note that only a few EFMs are of interest for the synthetic pathway extraction problem, namely those capable to produce the product of interest, which are the ones computed by the SSG algorithm. From a graph extraction viewpoint, a minimal solution implies that it cannot be disassembled into sub solutions. The condition b) allows to reach a) as it will be explained below.

Let us consider $\wp_n(X)$, which filters the power set in such way that it contains only the subsets with n elements. Then, instead of performing $C \leftarrow \wp(\Delta(x)) \setminus \{\emptyset\}$, we loop through $n = 1$ to $|\Delta(x)|$, by assigning $C \leftarrow \wp_n(\Delta(x))$. This is equivalent to the line 6 of the SSG algorithm, with the advantage that we do not hold in memory the entire power set during the search.

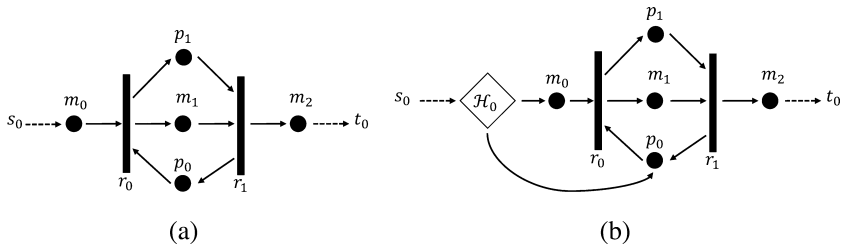


Fig. 1 – An example of a cyclic network. Vertex s_0 is the input substrate and t_0 the target metabolite. Circles represent metabolites and vertical bars represent reactions. (a) A network that does not contain pathways to produce neither p_0 nor p_1, leading to an infeasible problem to the Find Path algorithm, since no ordering is possible for reactions r_0, r_1. (b) The same network but now containing a pathway $\mathcal{H}_0$ producing p_0.

We conjecture that, assuming a solution exists for a combination $c \in \wp_i(X)$, then every combination of higher degree $\wp_{i+1}(X)$, that contains c, can be excluded, as these do not generate the minimal solution.

Example 1. If $X = \{a, b, c\}$ is a set with 3 elements, where $\wp(X) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$, then $\wp_0(X) = \{\emptyset\}$ is a subset of $\wp(X)$ with sets of 0 elements. Subsequently, $\wp_1(X) = \{\{a\}, \{b\}, \{c\}\}$ is the subset with all sets of 1 element and so on. Note that, for $\wp(X)$, every $\wp_n(X)$, where $n > 3$, is the empty set (i.e., $\wp_4(X) = \emptyset$).

Given Example 1, assuming a, b, c are reactions, if we are able to find a solution for the singleton set $\{a\}$, then we exclude combinatorial sets with a (e.g., $\{a, b\}$, $\{a, c\}$, $\{a, b, c\}$). This allows to remove many, if not all, non minimal solutions thus severely increasing the capability of the SSG algorithm to perform well over larger domains.

3.2. *Find Path*

3.2.1. *Original algorithm definition*

The Find Path (FP) algorithm proposed by Carbonell et al. [19] enumerates pathways by using hypergraphs. In a metabolic context, both hypergraphs and process graphs are similar (Definition 5). A solution of the FP algorithm is defined as a *hyperpath* (Definition 6). P, which is an hypergraph (i.e., a subgraph) where the hyperarcs (reactions) can be ordered as r_1, r_2, …, r_m, such that r_i is dependent only on the substrates in S and the products of the previous reactions.

Definition 5. (Hypergraph) A hypergraph $\mathcal{H} = \langle V, E \rangle$ with vertices V and hyperarcs E, can be defined in this context to be isomorphic to a metabolic network Σ (Definition 2), where V represents the set of metabolites Π and E the set of reactions Υ. Additionally, a hyperarc has a structure to a reaction (Definition 1), both encompassing two disjoint sets of vertices $\langle V_1, V_2 \rangle$ (each vertex corresponds to a metabolite).

Definition 6. (Hyperpath [19]) A hyperpath P going from a source subset $S_\mathcal{H}$ of V to a target subset T_P of P in a hypergraph $\mathcal{H} = \langle V, E \rangle$ is a hypergraph $\mathcal{H}_P = \langle V_P, E_P \rangle$ with $V_P \subseteq V$, $E_P \subseteq E$, such that there is an ordering F of the hyperarcs E_P with the following properties:

- $\forall_k \in \{0, \ldots, |F|\}$, $substrates(F_k) \subseteq S_H \cup (\cup_{j<k} products(F_j))$
- $T_P \subseteq S_H \cup (\cup_{e_q \in E_p} products(e))$

While addressing many of the problems of using shortest paths over regular graphs to represent metabolic pathways, this representation still has limitations. Indeed, not all pathways can be expressed by the definition of an hyperpath (Definition 6). Let us consider for instance co-factor metabolites p_0 and p_1.

Usually, these metabolites are both present in a single reaction $r_0 = \langle M_1, M_2 \rangle$ where $p_0 \in M_1$ and $p_1 \in M_2$ or vice versa (Fig. 1). These reactions can be satisfied by each other in a way where there is an $r_1 = \langle M_1', M_2' \rangle$ such that $p_1 \in M_1'$ and $p_0 \in M_2'$. Therefore, it is impossible to sort a hyperpath if neither p_0 or p_1 are included in S. Given the example in Fig. 1a, assuming $s_0 - m_0$ and $m_2 - t_0$ is feasible, then, $s_0 - t_0$ should be also feasible. But a hyperpath (Definition 6) dictates that reactions (or hyperarcs) in the hyperpath must be sortable in a particular order, where given any reaction F_k it must be satisfiable by the previous instances of F_j, $j < k$ or the initial set of substrates S_H. Now considering the two reactions r_1, r_2, this condition could never be achieved since they are dependent of each other. Examples of these metabolites are the pairs ATP-ADP and NADH-NAD. Fortunately, if assuming S to be an organism chassis (host), these metabolites are usually included in S since they are part of the metabolism. However, this does not guarantee that other more complex cycles do not exist.

This issue enables the generation of redundant solutions. Let $\Gamma = \langle \Sigma, \{s_0\}, \{t_0\} \rangle$ be a retrosynthetic problem. Assume that: (a) a heterologous pathway $\Sigma' \subset \Sigma$ exists from s_0 to t_0, such that b) $r_0, r_1 \in \Sigma'$, where $r_0 = \langle \{m_0, p_0\}, \{m_1, p_1\} \rangle$ and $r_1 = \langle \{m_1, p_1\}, \{m_2, p_0\} \rangle$. The FP algorithm can only identify such pathway if $\Gamma' = \langle \Sigma, \{s_0\}, \{p_0, m_0\} \rangle$ is feasible. Instead of reaching from $s_0 - m_0$ as it should, the algorithm will eventually find a workaround route from $s_0 - \{m_0, p_0\}$ (Fig. 1b). Since r_0, r_1 satisfy the metabolites p_0, p_1 of each other (i.e., $r + r' = \langle \{m_0\}, \{m_2\} \rangle$) this implies that any effort to produce p_0 in Γ' is unnecessary and every solution that (b) verifies may contain multiple redundant solutions (the reactions included in the solutions are unique but in steady state they are redundant). A solution to circumvent this problem is to add p_0 to the set of substrates, such metabolites are commonly referred as bootstrap compounds since they promote the propagation of the network, however they must be identified prior to the computation of the solutions.

The Find Path algorithm (Algorithm 4) makes use of the Find All (Algorithm 2) and Minimize (Algorithm 3) subroutines. Find All (FA) implements a pruning algorithm that reduces an hypergraph $\mathcal{H}$ to $\mathcal{H}'$, with a special property: the reactions $\Upsilon \in \mathcal{H}'$ are sorted by the definition of a hyperpath. This ordering is only essential to the Find All algorithm to branch correctly, while it can be discarded (i.e., any order is acceptable) in the Minimize routine.

Algorithm 2. Find All

```
1:  procedure FindAll(H, S) ▷ H hypergraph, S source metabolites
2:    for each r ∈ H do
3:      m[r] ← Ψ⁻(r)
4:    end for
5:    V ← S
6:    D ← S
7:    F ← ∅
8:    while V ≠ ∅ do
9:      let x be an element of V
10:     V ← V \ x
11:     D ← S ∪ x
12:     for each r ∈ H ∧ x ∈ m[r] do
13:       m[r] ← m[r] \ x
14:       if m[r] = ∅ then
15:         F ← {F, r}
16:         for each j ∈ Ψ⁺(r) ∧ x ∉ D do
17:           V ← V ∪ j
18:         end for
19:       end if
20:     end for
21:   end while
22:   return F
23: end procedure
```

The Minimize routine reduces a network to the minimal set of reactions by testing each reaction in the network $\mathcal{H}$ (Algorithm 3, line 7), so that if the reaction is removed from the network, the set of products is still reachable. This testing mechanism can be achieved by invoking FA with the new network (i.e., without the reaction to be removed). If FA returns a solution without the product, then the reaction is assumed to be critical. This implies that, for each reaction in $\mathcal{H}$, an invocation of FA is performed. Therefore, the Minimize routine shows quadratic complexity to the number of reactions in the network.

Algorithm 3. Minimize

```
1:  procedure Minimize(H, Rf, S, T) ▷ H hypergraph, Rf reactions to not test, S source set, T target set
2:    F ← FindAll(H, S) ▷ 2-4 Test if exists solution
3:    H' ← H
4:    if T ∩ Ψ⁺(F) = ∅ then
5:      H' ← ∅ ▷ Return empty set
6:    else ▷ Proceed to minimization
7:      for each r ∈ H do ▷ For each reaction not in Rf test if solution exists if H\r
8:        if r ∉ Rf then
9:          F ← FindAll(H\r, S)
10:         if T ∩ Ψ⁺(F) ≠ ∅ then
11:           H' ← H'\r ▷ Remove reaction from hypergraph
12:         end if
13:       end if
14:     end for
15:   end if
16:   return H' ▷ Return either ∅ or a minimal solution structure of H
17: end procedure
```

Algorithm 4. Find Path

```
1:  procedure FindPath(H, Rf, S, T) ▷ H hypergraph, S source metabolites, T target metabolites, Rf for branching solutions (initially as ∅)
2:    F ← FindAll(H, S)
3:    H' ← ∅
4:    H' ← H' ∪ F ∪ Rf
5:    Hσ ← Minimize(H', Rf, S, T) ▷ The first minimal solution (if exists)
6:    En ← ∅
7:    if Hσ ≠ ∅ then
8:      En ← Hσ
9:      F ← FindAll(Hσ, S)
10:     for k ∈ {|F| ... 1} do ▷ For each element in F branch alternative solutions
11:       r = Fk
12:       if r ∉ Rf then
13:         En ← {En, FindPath(H\r, Rf, S, T)}
14:         Rf ← Rf ∪ r
15:       end if
16:     end for
17:   end if
18:   return En
19: end procedure
```

3.2.2. *Improved Minimize heuristic*

In this work, we propose an alternative to the Minimize heuristic that aims to overcome the problem of its quadratic computational complexity. We address this issue by proposing a different heuristic to test the reactions in the Minimize routine.

Assume that $\Gamma = \langle \Sigma, S, T \rangle$ contains valid solutions that are searchable using the Find Path algorithm. Assume that we increase the size of the search space to $\Sigma' = \langle \Pi', \Upsilon' \rangle$, where $|\Upsilon'|$ is much larger than $|\Upsilon|$. This also implies that the previous searchable solutions of Γ are preserved, since it is impossible to invalidate a solution by adding more reactions to the search space. The computational cost of the previous solutions in Γ will eventually increase because of: (a) there are more reactions in the new network to test, therefore the computational cost of Find All increases; and (b) the Minimize now contains more reactions to remove in order to achieve the

previous minimal solutions of Γ. Furthermore, it is natural that new solutions may be possible because of the newly added reactions in Υ'.

Our goal is to reduce the penalty to compute solutions when adding more reactions to the set. Instead of testing each reaction r (Algorithm 3, line 7), we test the removal of an entire set R of reactions. This speeds up the computation cost, specially in the search of the smallest solutions in huge networks generated from large databases, such as KEGG and MetaCyc. The size of R is an important factor, since it impacts the speed up obtained by the bulk removal of reactions.

We follow the strategy of the bisection optimization method to find the reactions that cannot be removed, thus generating a minimal set of reactions. Let X be the entire set of reactions in a network, we split X into two halves X^L and X^R, we attempt to remove from left to right each half. If X^L cannot be removed, i.e., if by removing X^L the Find All routine returns a sequence without the set T, this implies that X^L contains a reaction that must be present in the minimal solution; otherwise, there is no solution possible. Then, we split X^L into further halves X'^L, X'^R and perform again the Find All test. This routine is recursively performed until either the entire subset can be removed or we have a singleton set that cannot be removed, which implies that the reaction belongs to the minimal solution. This will generate a tree pattern where the leafs are either a singleton set with only one element (i.e., the reaction that belongs to the minimal solution) or sets of reactions that were discarded.

No modifications were made to the main Find Path algorithm.

4. Experimental setup

4.1. Case studies

The algorithms were tested through their application to three case studies of synthetic metabolic engineering. The first example is the production of 1-butanol using *E. coli* [27], the second concerns vanillin synthesis using *S. cerevisiae* [28] and last the biosynthesis of curcumin in *E. coli*. Both modified SSG and FP algorithms are applied using the set of compounds in the KEGG Ligand and MetaCyc databases as the chemical search space. Additionally, to integrate and test the obtained solutions *in silico*, a GSMM is required: the *i*JO1366 [29] GSMM for *E. coli* and iMM904 [30] GSMM for *S. cerevisiae* were used. In both cases aerobic conditions were used with an uptake flux of glucose of 10 mmol/gDW/h. Therefore, a total of 12 result sets were generated for the two algorithms, three case studies and two search spaces (databases).

4.2. Data preprocessing

Before running the algorithms, several pre-processing tasks were required. The first was to select and define the constraints of the problem, selecting the search space Σ, the initial set S and the target compounds T. For all case studies, the target set is a singleton containing only the compound of interest (i.e., 1-butanol, vanillin and curcumin). For the substrate set, all metabolites included in the GSMMs were selected. This later will allow to integrate the obtained solutions with these models and evaluate their performance. The BiGG database [31] aided in the transformation of the species identifiers of the model to those in the databases. The species that did not match any cross-referencing were discarded.

Part of the reference pathway of the 1-butanol synthesis was mostly present in the *i*JO1366 GSMM as part of the Membrane Lipid Metabolism pathways. So, to obtain alternative pathways, we removed the following species: M_btcoa_c (Butanoyl-CoA), M_btal_c (Butanal), M_b2coa_c (Crotonyl-CoA), M_3hbcoa_c (3-hydroxybutyryl-CoA), M_aacoa_c (Acetoacetyl-Coa). Additionally, every reaction connected to these compounds was also removed. The impact in the biomass value calculated using the FBA was minimal (less than 1%). Removing these species will allow to find alternative paths from other internal metabolites of *i*JO1366 to 1-butanol. This is done because we wanted to reach alternative solutions to the identified in [27], which may not be optimal, depending on the desired criteria. Furthermore, the algorithms do not generate solutions with reactions producing substrates in the initial set, since these are defined as supplied compounds. The curcumin case study required a new substrate in the medium, which involved the addition of a new metabolite to the *i*JO1366 GSMM, the ferulic acid.

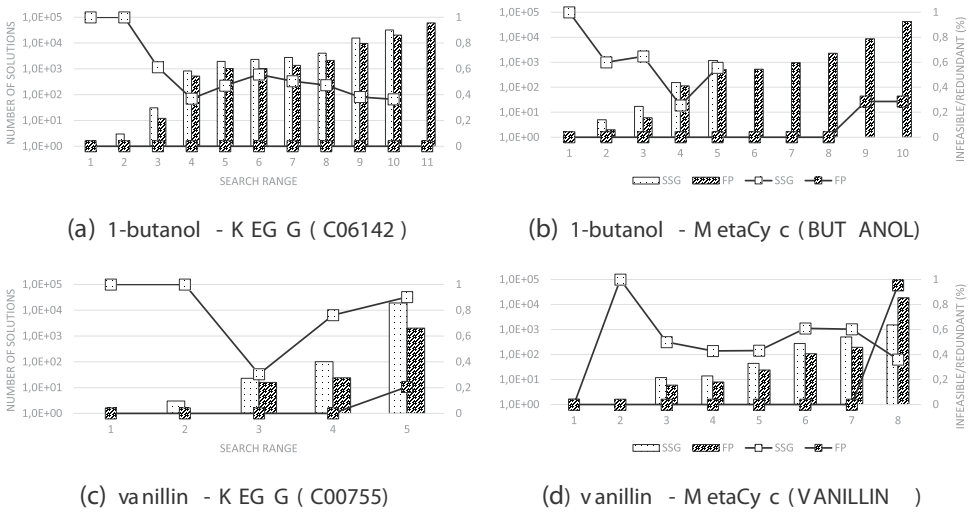
A minor modification was made to the MetaCyc database, since it contains reactions with the metabolite pairs `NAD-P-OR-NOP/NADH-P-OR-NOP` which are an instance of either `NAD/NADH` or `NADP/NADHP`. These reactions were unfolded to their correct instances. This is essential for instance to infer the 1-butanol reference pathway, as several reactions of this pathway were expressed in this format. The KEGG Ligand database did not require any pre-processing.

4.3. Implementation details

Both algorithms and the described modifications were implemented in Java according to the algorithms previously defined. All experiments were run on a machine running CentOS 6.4 (Linux 2.6.32) with two Intel® Xeon X5650 (2.66 GHz) and 64GBytes of memory. The java programs were compiled and run with JDK™ 7 (version 1.7.0_45). The implementation of FBA and other CBM related methods over GSMMs was taken from the core packages of the OptFlux ME platform [32] (version 3.1). The CPLEX solver (version 2.14) was used to perform the linear optimization tasks related to FBA. The KEGG information was obtained from the release 68.0 (October 1, 2013) and the MetaCyc database was taken at the same time period (release 17.5, October 11, 2013).

4.4. Algorithm setup

Because of the combinatorial explosion of possible pathways, it is impossible to obtain every solution existing in a database size network using any of the algorithms. To compare the algorithms' performance, the search space was split into subsets by *radius*. The *radius* is an integer that defines the minimum number of links (i.e., reactions) required to reach that reaction from an initial set of metabolites. This approach was used previously by Handorf et al. [33] to analyze large metabolic networks. The strategy is to pick one or more seed metabolites



(a) 1-butanol - KEGG (C06142) (b) 1-butanol - MetaCyc (BUTANOL)

(c) vanillin - KEGG (C00755) (d) vanillin - MetaCyc (VANILLIN)

Fig. 2 – Pathways computed for each of the problems by radius. Bar plot (left axis) shows the total number of solutions (logarithmic scale); line plot (right axis) shows the percentage of unfeasible/redundant solutions; SSG – dotted; FP – dashed.

and expand the network from these seeds by capturing their neighbor reactions. This implies that a reaction belonging to radius i also belongs to $i+1$, and therefore a subnetwork Σ_i of *radius* i always complies to $\Sigma_i \subseteq \Sigma_{i+1}$. For our case studies these seeds are the target product which is a single compound.

With these reduced search spaces, solutions were computed using each of the algorithms. An attempt was made to obtain the entire set of candidate solutions for each *radius*, until either the process crashed due to lack of memory or exceeded computational time allotted (>24 h). To validate the solutions, FBA was used to maximize the product flux of the target compound and validate its feasibility integrating the solution into the respective GSMM.

5. Results and discussion

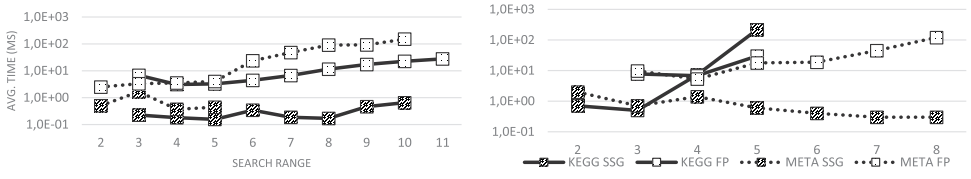
Fig. 2 shows the number of solutions computed and their feasibility. SSG is more limited than FP by the size of the search space. A major problem of the SSG algorithm is the high memory demand because of the power set computation. With the reduction of the power set size (only partial sets are computed), it still presents high memory demand to branch all the possible combinations. Moreover, the SSG computes every solution that satisfies Definition 4 which eventually leads to the computation of infeasible pathways.

In general, the SSG shows better performance in the computation of solutions (Fig. 3) mainly because of the branching technique which gives a major advantage to the computation time per solution because of the backtracking. As the algorithm moves to a candidate solution, the next solution reuses the previous partial solution. This results in a neglectable impact on the computation time per solution as the search space increases (i.e., increasing size of the radius). However, since the number of solutions exponentially grows with the increasing size of the search space, the total computation time increases.

Table 1 – Number of solutions found by the SSG implementations (original and improved). The * means the process did not terminate given the amount of memory taken.

Radius	Original	Modified
1	1	1
2	4	3
3	715	31
4	*	831

The exponential growth of the number of solutions renders the original implementation of the algorithm limited to the very small values for the radius, since achieving every solution is impracticable (Table 1). The extensive amount of solutions found by the original implementation are the result of the non-minimal solutions from the combination of the minimal solutions found by the improved version.



(a) 1-butanol (b) vanillin

Fig. 3 – Time cost (ms) per each solution (logarithmic scale). On the *x*-axis the search radius is shown (a higher radius implies a larger search space).

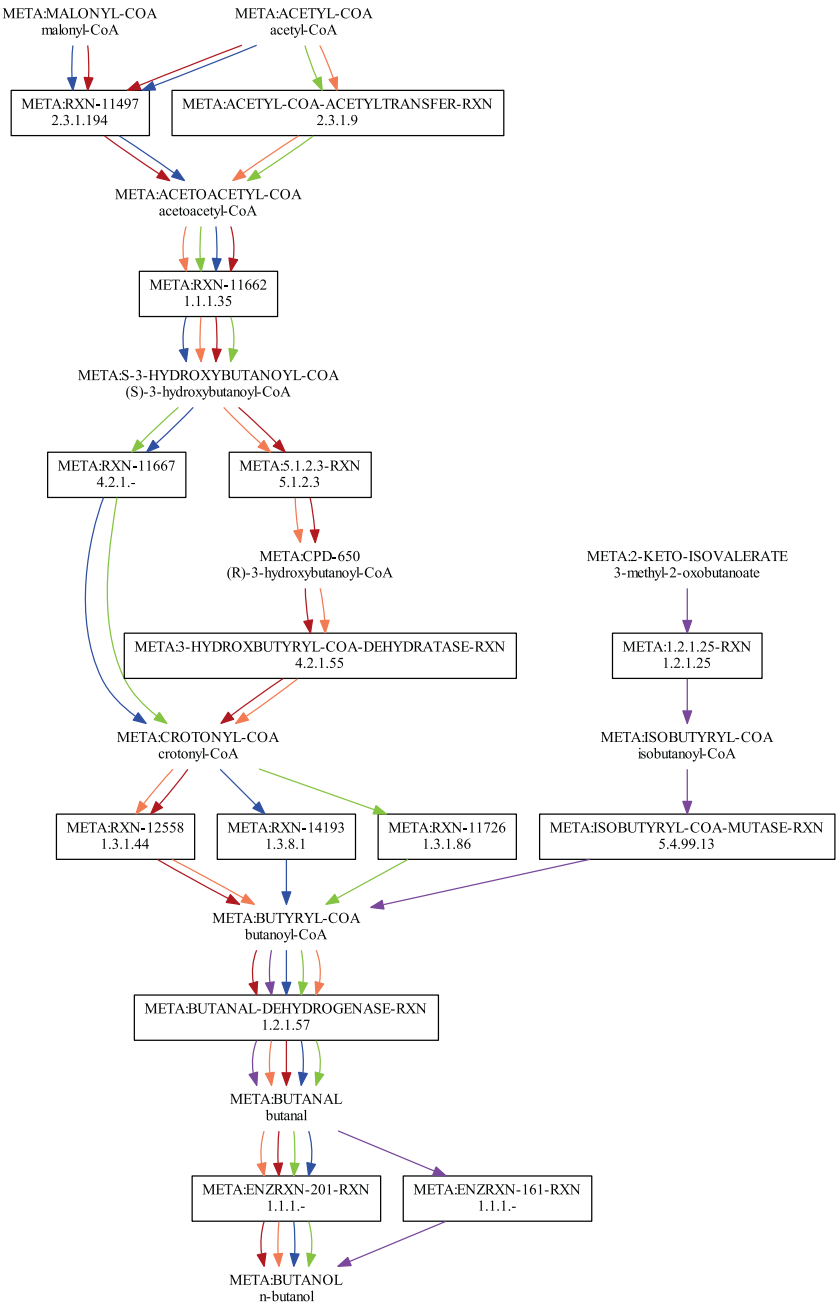


Fig. 4 – Main alternative pathways discovered for butanol production. Besides the pathway from *Clostridia*, alternative paths were discovered that include the cyanobacteria alternative starting with malonyl-CoA and pathways starting at 2-keto-isovalerate which are both associated with intellectual property. Moreover, some variations to the original *Clostridia* pathway were found as previously validated for example [34].

However, even when only non-minimal solutions are found in our improved version their number grows combinatorially with the increase in the radius. In Fig. 4, selected solutions for the 1-butanol case study are shown to illustrate the behavior of the algorithms and the combinatorial explosion of the number of solutions found, even if only minimal solutions are present. Both algorithms combine reactions in the network to generate distinct pathways, and in many cases for a single step between two compounds there are multiple viable reactions (e.g., in MetaCyc the step between crotonyl-CoA and butanoyl-CoA shows three viable reactions META:RXN-12558, META:RXN-14193, META:RXN-14193 in Fig. 4). Such reactions greatly increase the number solutions and these cases are common, mainly considering variants of reactions varying only in the used co-factors (e.g., NADH, NADPH or FADH).

While the SSG modifications were focused to filter the solutions to only minimal solutions (which improved the capability to search larger networks), this property was already natural to the FP algorithm. Our strategy to achieve larger domains for the FP algorithm implied the modification of the

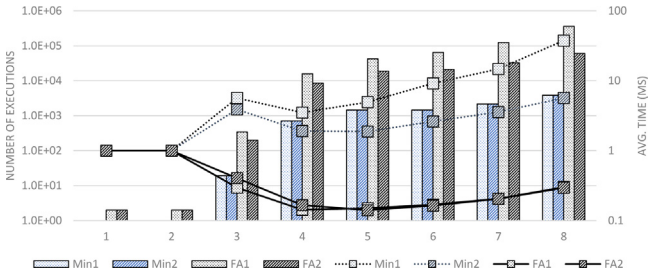


Fig. 5 – The number of executions of the Minimize (Min1/Min2) and Find All (FA1/FA2) subroutines of the Find Path algorithm in each sub-domain (horizontal axis). Min1 and FA1 represent the original implementation, while Min2 and FA2 the improved one. The left axis (Bar Plot) represents the number of executions; the right axis (Line Plot) represents the mean execution time (ms).

search heuristic of the Minimize kernel. Fig. 5 reflects the changes of the number of executions of the Find All subroutine compared to the original implementation. There is a significant decrease of the number of calls to this subroutine due to the bisection optimization strategy applied, therefore reflecting in the total computation time of the Minimize routine.

The strict topology that the Find All kernel implies that there are fewer solutions obtained from the FP algorithm. While the SSG attempts to combine every subset, this more aggressive strategy is capable to find every solution of the FP scope. The FP is capable to compute larger search spaces, being the major bottleneck the computation time per solution, since the internal Minimize routine has quadratic complexity to the number of reactions [19]. The comparison between the solutions found between the two algorithms shows that the SSG is capable to obtain more solutions (Fig. 6). This trait was also excepted because of the rules imposed by the Find All routine. Additionally, there were scenarios found where FP computes multiple distinct redundant solutions, due to the problems explained above in detail.

The curcumin case study revealed a much lesser solution diversity (Table 2) since the amount of solutions is highly dependent on the diversity of reactions in the search space. Curcumin is a compound found originally in a few plants and thus the diversity of pathways for its production is still low. Both SSG and FP were able to fully compute the entire dataset of reactions in MetaCyc obtaining just a few solutions. The

Table 2 – Number of solutions obtained for the curcumin case study (on the right the number of solutions feasible with the iJO1366 GSMM). For the KEGG dataset, solutions are up to radius 5 and 3 for FP and SSG, respectively. The MetaCyc dataset was fully computed.

	Find Path		SSG	
	Total	Feasible	Total	Feasible
KEGG	285	217	5	5
MetaCyc	10	7	10	7

FP method was able to compute a much higher amount of solutions using the KEGG reaction set; however the SSG was unable to pass the 4th radius having only five solutions in the 3rd radius of the KEGG search space. The KEGG dataset showed increased complexity compared to the MetaCyc reactions which led the SSG algorithm to block due to memory limitations. Again the FP algorithm prove to be more capable of obtaining complex pathways mostly due to the assumption that pathways are acyclic.

For every solution that satisfies the feasibility test, the fitness was evaluated by integrating it into the corresponding GSMM. The farthest *radius* that either algorithm was able to compute was selected for this process. For the 1-butanol case, from the 42,482 and 60,356 solutions obtained from the FP algorithm, a total of 32,692 and 22,968 were compatible with the iJO1366 GSMM for search spaces of MetaCyc and KEGG, respectively. In the vanillin case, 944 out of 974 computed solutions are valid (MetaCyc), being the numbers for KEGG of 1600 out of 1852. Finally, for the curcumin pathways 217 out of 285 KEGG pathways and 7 out of 10 MetaCyc pathways were feasible with the iJO1366 GSMM. The 1-butanol case shown a massive amount of solutions mostly because of the NAD/NADH alternatives for many reactions.

The KEGG dataset provided the solution with highest yield for vanillin and curcumin. Moreover, 152 pathways were found in KEGG with the maximum yield for 1-butanol (0.99, given by 9.99 mmol/gDW/h for the butanol production flux divided by 10 mmol/gDW/h for glucose uptake) compared to 114 pathways from MetaCyc, while for the curcumin case study the amount of solutions obtained from MetaCyc is quite limited. There is a noticeable difference in the configuration of the yield distribution between KEGG and MetaCyc (Fig. 7), which demonstrates that there are key reactions that are unique to each database, therefore leading to different pathway configurations.

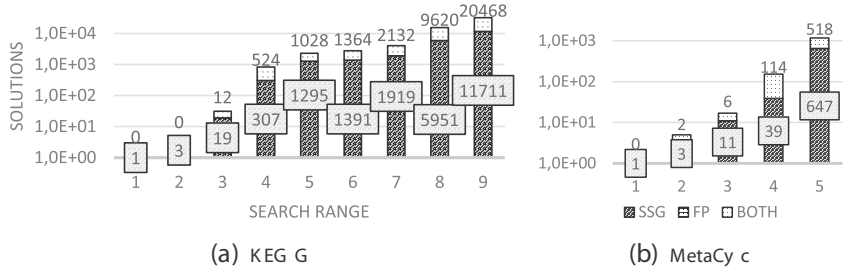


Fig. 6 – Difference between the number of solutions obtained in the 1-butanol case study for FP and SSG. Panel (a) shows the case for the KEGG domain, while (b) considers the MetaCyc domain. The Find Path algorithm did not return any solutions not found by SSG.

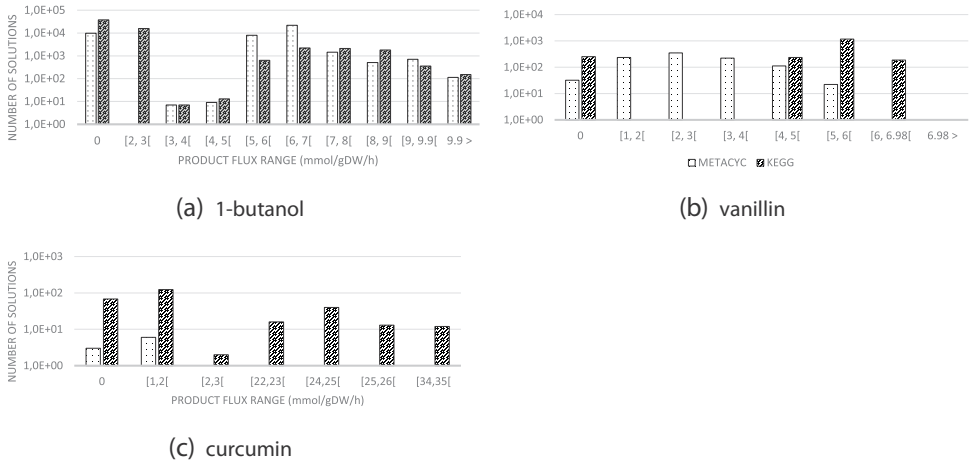


Fig. 7 – Histogram of theoretical flux values of each case study (1-butanol/curcumin – iJO1366; vanillin – iMM904). Last value is the optimal solution (highest product flux).

Details on the best solutions found can be checked in the supplementary material and the main families of solutions found for butanol are represented in Fig. 4. In summary, it can be concluded that overall the algorithms were able to find widely known efficient pathways but also less utilized ones. For example, in the case of butanol, the best performing pathways in terms of yield include the commonly used pathway from *Clostridium acetobutylicum*, which has also been validated [27] as a heterologous pathway in *E. coli*, together with a diversity of variations in a few steps. Also, less common pathways have been found, that have been recently patented and that use 2-ketoisovalerate as an intermediate [35].

Moreover, pathways from cyanobacteria deriving from malonyl-CoA, which have already been reported as good alternatives to pathways starting at acetoacetyl-CoA [34] and which have associated patent applications were also discovered [36] by the algorithm and are represented in Fig. 4.

In the case of curcumin, most of the solutions take tyrosine as a precursor, as has been described elsewhere [37]. Nevertheless, in both cases there are many alternatives that are stoichiometrically feasible but for which no reports have been found in the literature. Those cases need to be further inspected for biological and biochemical consistency before implementation. Nevertheless, they constitute promising alternatives to produce valuable products.

6. Conclusions and future perspectives

The algorithms analyzed (SSG and FP) both present shortcomings in the computation of heterologous pathways. Although topologically they are correct, they may be stoichiometrically inconsistent within a microorganism's context, as they have the common goal of inferring heterologous pathways (subnetworks) that satisfy the rules of initial substrates and target product. However, by using post-processing methods such as FBA, stoichiometrically valid solutions can be identified, which allows to correctly enumerate multiple steady-state pathways. The case study of 1-butanol shows that there are many viable and optimally efficient (regarding yields) routes for the production of this compound using as basis the iJO1366 model. Moreover, even if a problem contains only a single optimal solution (e.g., vanillin in iMM904), examples of sub-optimal pathways also show a broad range of yield value near the optimal. Due to their nature, deterministic methods hardly can achieve such a range of feasible steady state heterologous pathways.

Overall, the FP has proven to be more flexible regarding the complexity and the size of the graph, and although being more penalized with the number of reactions in the search space, it is more capable to compute larger sets.

Thus, it is shown that although neither of the algorithms is readily suitable to compute steady state heterologous pathways for large databases, they are still able extract potential pathways, after targeted improvements in scalability. Additionally, they offer a generic method to infer pathways for multiple purposes, since they do not follow any strict objective function (e.g., yield or size).

As future work, both these algorithms can still be improved towards their scalability. One line of work will certainly be the efficient parallelization of these algorithms resorting to adequate software development tools [38]. A complementary research topic will address the comparison of these approaches with recent proposals within EFM research.

Conflicts of interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work is partially funded by ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT (Portuguese Foundation for Science and Technology) within projects ref. COMPETE FCOMP-01-0124-FEDER-015079 and Strategic Project PEst-OE/EQB/LA0023/2013,

and also by Project 23060, PEM – Technological Support Platform for Metabolic Engineering, co-funded by FEDER through Portuguese QREN under the scope of the Technological Research and Development Incentive system, North Operational.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.cmpb.2014.11.010.

REFERENCES

[1] K.A. Curran, H.S. Alper, Expanding the chemical palate of cells by combining systems biology and metabolic engineering, Metab. Eng. 14 (4) (2012) 289–297, http://dx.doi.org/10.1016/j.ymben.2012.04.006 http://www.ncbi.nlm.nih.gov/pubmed/22595280

[2] G. Rodrigo, J. Carrera, K.L.J. Prather, A. Jaramillo, DESHARKY: automatic design of metabolic pathways for optimal cell growth, Bioinformatics 24 (21) (2008) 2554–2556, http://dx.doi.org/10.1093/bioinformatics/btn471 http://www.ncbi.nlm.nih.gov/pubmed/18776195

[3] M. Kanehisa, S. Goto, KEGG: kyoto encyclopedia of genes and genomes, Nucleic Acids Res. 28 (1) (2000) 27–30 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=102409&tool=pmcentrez&rendertype=abstract

[4] M. Kanehisa, S. Goto, Y. Sato, M. Kawashima, M. Furumichi, M. Tanabe, Data, information, knowledge and principle: back to metabolism in KEGG, Nucleic Acids Res. 42 (Database issue) (2014) D199–D205, http://dx.doi.org/10.1093/nar/gkt1076 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3965122&tool=pmcentrez&rendertype=abstract

[5] R. Caspi, T. Altman, R. Billington, K. Dreher, H. Foerster, C.A. Fulcher, T.A. Holland, I.M. Keseler, A. Kothari, A. Kubo, M. Krummenacker, M. Latendresse, L.A. Mueller, Q. Ong, S. Paley, P. Subhraveti, D.S. Weaver, D. Weerasinghe, P. Zhang, P.D. Karp, The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases, Nucleic Acids Res. 42 (Database issue) (2014) D459–D471, http://dx.doi.org/10.1093/nar/gkt1103 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3964957&tool=pmcentrez&rendertype=abstract

[6] D. Croes, F. Couche, S.J. Wodak, J. van Helden, Metabolic PathFinding: inferring relevant pathways in biochemical networks, Nucleic Acids Res. 33 (Web Server issue) (2005) W326–W330, http://dx.doi.org/10.1093/nar/gki437 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1160198&tool=pmcentrez&rendertype=abstract

[7] K. Faust, P. Dupont, J. Callut, J. van Helden, Pathway discovery in metabolic networks by subgraph extraction, Bioinformatics 26 (9) (2010) 1211–1218, http://dx.doi.org/10.1093/bioinformatics/btq105 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2859126&tool=pmcentrez&rendertype=abstract

[8] M. Arita, In silico atomic tracing by substrate-product relationships in Escherichia coli intermediary metabolism, Genome Res. 13 (11) (2003) 2455–2466, http://dx.doi.org/10.1101/gr.1212003 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=403765&tool=pmcentrez&rendertype=abstract

[9] F. Boyer, A. Viari, Ab initio reconstruction of metabolic pathways, Bioinformatics 19 (Suppl. 2) (2003) ii26–ii34, http://dx.doi.org/10.1093/bioinformatics/btg1055 http://bioinformatics.oxfordjournals.org/cgi/doi/10.1093/bioinformatics/btg1055

[10] A.P. Heath, G.N. Bennett, L.E. Kavraki, Finding metabolic pathways using atom tracking, Bioinformatics 26 (12) (2010) 1548–1555, http://dx.doi.org/10.1093/bioinformatics/btq223 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2881407&tool=pmcentrez&rendertype=abstract

[11] K. Faust, D. Croes, J. van Helden, Metabolic pathfinding using RPAIR annotation, J. Mol. Biol. 388 (2) (2009) 390–414, http://dx.doi.org/10.1016/j.jmb.2009.03.006, URL http://www.ncbi.nlm.nih.gov/pubmed/19281817

[12] A. Cho, H. Yun, J.H. Park, S.Y. Lee, S. Park, Prediction of novel synthetic pathways for the production of desired chemicals, BMC Syst. Biol. 4 (2010) 35, http://dx.doi.org/10.1186/1752-0509-4-35 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2873314&tool=pmcentrez&rendertype=abstract

[13] J. Wu, Z. Guan, Q. Zhang, A.K. Singh, X. Yan, Static and dynamic structural correlations in graphs, IEEE Trans. Knowl. Data Eng. 25 (9) (2013) 2147–2160, http://dx.doi.org/10.1109/TKDE.2012.133 http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6226407

[14] E. Pitkänen, P. Jouhten, J. Rousu, Inferring branching pathways in genome-scale metabolic networks, BMC Syst. Biol. 3 (2009) 103, http://dx.doi.org/10.1186/1752-0509-3-103 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2791103&tool=pmcentrez&rendertype=abstract

[15] F. Friedler, K. Tarján, Y. Huang, L. Fan, Graph-theoretic approach to process synthesis: axioms and theorems, Chem. Eng. Sci. 47 (8) (1992) 1973–1988, http://dx.doi.org/10.1016/0009-2509(92)80315-4, URL http://www.sciencedirect.com/science/article/pii/0009250992803154, http://linkinghub.elsevier.com/retrieve/pii/0009250992803154

[16] F. Friedler, K. Tarjan, Y.W. Huang, L. Fan, Graph-theoretic approach to process synthesis: polynomial algorithm for maximal structure generation, Comput. Chem. Eng. 17 (9) (1993) 929–942, http://dx.doi.org/10.1016/0098-1354(93)80074-W http://linkinghub.elsevier.com/retrieve/pii/009813549380074W

[17] F. Friedler, J. Varga, L. Fan, Decision-mapping: a tool for consistent and complete decisions in process synthesis, Chem. Eng. Sci. 50 (11) (1995) 1755–1768, http://dx.doi.org/10.1016/0009-2509(95)00034-3 http://linkinghub.elsevier.com/retrieve/pii/0009250995000343

[18] D.-Y. Lee, L. Fan, S. Park, S.Y. Lee, S. Shafie, B. Bertók, F. Friedler, Complementary identification of multiple flux distributions and multiple metabolic pathways, Metab. Eng. 7 (3) (2005) 182–200, http://dx.doi.org/10.1016/j.ymben.2005.02.002 http://www.ncbi.nlm.nih.gov/pubmed/15885617

[19] P. Carbonell, D. Fichera, S.B. Pandit, J.-L. Faulon, Enumerating metabolic pathways for the production of heterologous target chemicals in chassis organisms, BMC Syst Biol. 6 (1) (2012) 10, http://dx.doi.org/10.1186/1752-0509-6-10 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3311073&tool=pmcentrez&rendertype=abstract

[20] J. Heino, D. Calvetti, E. Somersalo, Metabolica: a statistical research tool for analyzing metabolic networks, Comput. Methods Progr. Biomed. 97 (2) (2010) 151–167, http://dx.doi.org/10.1016/j.cmpb.2009.07.007 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2814918&tool=pmcentrez&rendertype=abstract

[21] J.D. Orth, I. Thiele, B.O. Palsson, What is flux balance analysis? Nat. Biotechnol. 28 (3) (2010) 245–248,

http://dx.doi.org/10.1038/nbt.1614
http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3108565&tool=pmcentrez&rendertype=abstract

[22] S. Chatsurachai, C. Furusawa, H. Shimizu, An in silico platform for the design of heterologous pathways in nonnative metabolite production, BMC Bioinformatics 13 (1) (2012) 93, http://dx.doi.org/10.1186/1471-2105-13-93
http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3506926&tool=pmcentrez&rendertype=abstract

[23] P. Pharkya, A.P. Burgard, C.D. Maranas, OptStrain: a computational framework for redesign of microbial production systems, Genome Res. (814) (2004) 2367–2376, doi:10.1101/gr.2872004.14.

[24] D. Machado, Z. Soons, K.R. Patil, E.C. Ferreira, I. Rocha, Random sampling of elementary flux modes in large-scale metabolic networks, Bioinformatics 28 (18) (2012) i515–i521, http://dx.doi.org/10.1093/bioinformatics/bts401
http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3436828&tool=pmcentrez&rendertype=abstract

[25] L.F. de Figueiredo, A. Podhorski, A. Rubio, C. Kaleta, J.E. Beasley, S. Schuster, F.J. Planes, Computing the shortest elementary flux modes in genome-scale metabolic networks, Bioinformatics 25 (23) (2009) 3158–3165, http://dx.doi.org/10.1093/bioinformatics/btp564
http://www.ncbi.nlm.nih.gov/pubmed/19793869

[26] J. Gebauer, S. Schuster, L.F. de Figueiredo, C. Kaleta, Detecting and investigating substrate cycles in a genome-scale human metabolic network, FEBS J. 279 (17) (2012) 3192–3202, http://dx.doi.org/10.1111/j.1742-4658.2012.08700.x
http://www.ncbi.nlm.nih.gov/pubmed/22776428

[27] S. Atsumi, A.F. Cann, M.R. Connor, C.R. Shen, K.M. Smith, M.P. Brynildsen, K.J.Y. Chou, T. Hanai, J.C. Liao, Metabolic engineering of Escherichia coli for 1-butanol production, Metab. Eng. 10 (6) (2008) 305–311, http://dx.doi.org/10.1016/j.ymben.2007.08.003
http://www.ncbi.nlm.nih.gov/pubmed/17942358

[28] E.H. Hansen, B.L. Møller, G.R. Kock, C.M. Bünner, C. Kristensen, O.R. Jensen, F.T. Okkels, C.E. Olsen, M.S. Motawia, J.R. Hansen, De novo biosynthesis of vanillin in fission yeast (*Schizosaccharomyces pombe*) and baker's yeast (*Saccharomyces cerevisiae*), Appl. Environ. Microbiol. 75 (9) (2009) 2765–2774, http://dx.doi.org/10.1128/AEM.02681-08
http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2681717&tool=pmcentrez&rendertype=abstract

[29] J.D. Orth, T.M. Conrad, J. Na, J.A. Lerman, H. Nam, A.M. Feist, B.O. Palsson, A comprehensive genome-scale reconstruction of *Escherichia coli* metabolism-2011, Mol. Syst. Biol. 7 (535) (2011) 535, http://dx.doi.org/10.1038/msb.2011.65
http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3261703&tool=pmcentrez&rendertype=abstract

[30] M.L. Mo, B.O. Palsson, M.J. Herrgård, Connecting extracellular metabolomic measurements to intracellular flux states in yeast, BMC Syst. Biol. 3 (2009) 37, http://dx.doi.org/10.1186/1752-0509-3-37
http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2679711&tool=pmcentrez&rendertype=abstract

[31] J. Schellenberger, J.O. Park, T.M. Conrad, B.O. Palsson, BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions, BMC Bioinform. 11 (2010) 213, http://dx.doi.org/10.1186/1471-2105-11-213
http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2874806&tool=pmcentrez&rendertype=abstract

[32] I. Rocha, P. Maia, P. Evangelista, P. Vilaça, S.A. Soares, J.P. Pinto, J. Nielsen, K.R. Patil, E.C. Ferreira, M. Rocha, OptFlux: an open-source software platform for in silico metabolic engineering, BMC Syst. Biol. 4 (2010) 45, http://dx.doi.org/10.1186/1752-0509-4-45
http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2864236&tool=pmcentrez&rendertype=abstract

[33] T. Handorf, O. Ebenhöh, R. Heinrich, Expanding metabolic networks: scopes of compounds, robustness, and evolution, J. Mol. Evol. 61 (4) (2005) 498–512, http://dx.doi.org/10.1007/s00239-005-0027-1
http://www.ncbi.nlm.nih.gov/pubmed/16155745

[34] E.I. Lan, J.C. Liao, ATP drives direct photosynthetic production of 1-butanol in cyanobacteria, Proc. Natl. Acad. Sci. U. S. A. 109 (16) (2012) 6018–6023, http://dx.doi.org/10.1073/pnas.1200074109
http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3341080&tool=pmcentrez&rendertype=abstract

[35] L. Wu, J. Perkins, G. Schyns, Alternative Butanol Production Process in a Microbial Cell, 2010
http://www.google.com.ar/patents/WO2010031772A3?cl=en

[36] J.C. Liao, E.I. Lan, Atp Driven Direct Photosynthetic Production of Fuels and Chemicals, 2013
http://www.google.com/patents/WO2013126855A1?cl=en

[37] Y. Katsuyama, M. Matsuzawa, N. Funa, S. Horinouchi, Production of curcuminoids by *Escherichia coli* carrying an artificial biosynthesis pathway, Microbiology 154 (Pt 9) (2008) 2620–2628, http://dx.doi.org/10.1099/mic.0.2008/018721-0
http://www.ncbi.nlm.nih.gov/pubmed/18757796

[38] J. Pinho, J.A.L. Sobral, M. Rocha, Parallel evolutionary computation in bioinformatics applications, Comput. Methods Progr. Biomed. 110 (2) (2013) 183–191, http://dx.doi.org/10.1016/j.cmpb.2012.10.001
http://www.ncbi.nlm.nih.gov/pubmed/23127284