An Algorithm for Identifying Dominant-Edge Metabolic Pathways

Ehsan Ullah
Department of
Computer Science
Tufts University
ehsan.ullah@tufts.edu

Kyongbum Lee
Department of
Chemical and Biological Engineering
Tufts University
kyongbum.lee@tufts.edu

Soha Hassoun
Department of
Computer Science
Tufts University
soha@cs.tufts.edu

ABSTRACT

Metabolic pathway analysis seeks to identify critical reactions in living organisms and plays an important role in synthetic biology. We present in this paper an algorithm, DOMINANT-EDGE PATHWAY, for identifying a thermodynamically favored dominant-edge pathway forming a particular metabolite product from a particular reactant in a metabolic reaction network. The metabolic network is represented as a graph based on the stoichiometry of the reactions. The problem is formulated to first identify the path between the reactant and product with a limiting reaction based on Gibbs free energy changes, and then to augment this path with supplementary pathways with the goal of balancing the overall stoichiometry. Results of three representative test cases show that our algorithm efficiently finds potentially preferred reaction routes, offering a substantial run-time advantage over commonly used enumeration-based approaches.

1. INTRODUCTION

Synthetic biology has emerged as a powerful paradigm for producing a diverse array of complex natural and non-natural chemicals from simple building blocks using microbial hosts. Examples of recent successes include synthesis of antimalarial drug precursor arteminisin [11] and advanced biofuels such as branched-chain higher alcohols [2]. These successes largely reflected experimental efforts dependent on substantial domain-specific knowledge. However, the living cell is an exceedingly complex system, and empirical solutions are not always obvious. In this regard, computational analysis tools could complement the empirical approach by providing a systematic framework to learn from natural systems, to re-engineer them, and to create novel synthetic pathways.

Consider manipulating metabolic pathways in microbes to convert biomass into transportation fuels [8]. One of the grand technical challenges in biofuel synthesis is to achieve economic viability by improving the sugar-to-fuel conversion yield. Earlier efforts have focused on inserting [15] or deleting [12] one or two “key” enzymes to expand the range of sugar substrates utilized by the microbe or to limit the production of byproducts.

In these approaches, a crucial step was to identify the appropriate target pathway for manipulation. Given the very large number of metabolic reactions in a cell (~ hundreds) and the complexity of interactions between these reactions, the search for the optimal target can be daunting. Recently, Trinh et al. utilized a pathway analysis technique called elementary flux mode (EFM) analysis to engineer a strain of *E. coli* to maximally utilize both glucose and xylose as carbon sources to produce ethanol [17]. In principle, EFM analysis is capable of enumerating all of the feasible reaction routes available to a metabolic network at steady state [14]. As such, EFM analysis explores *all possible* routes capable of transforming a particular starting metabolite into a particular product. On the other hand, the activity or flux (rate of turnover of molecules) distribution through a cellular reaction network is highly uneven, and it is unlikely that every possible route leads to an equally valid target with the same capacity. A more plausible scenario is that the pathways’ degrees of engagement vary with the cell’s operating environment (e.g. temperature, pH and nutrient concentration) and regulatory state.

In this context, finding a favorable reaction route with the highest degree of engagement is an important next step for biochemical pathway analysis, especially for the purpose of engineering a synthetic pathway. An exhaustive approach is to investigate all possible EFMs that involve the input and output metabolites. However, recent studies have shown that this approach is computationally intractable [22]. A medium-scale model of *E. coli* intermediary metabolism with ca. 100 reactions can have 0.5 million EFMs [7]. Even a relatively small, simplified model with 60 reactions supports 30,000 or more EFMs (see section 6). Alternatively, we can apply a heuristic, weighted search algorithm that reflects expert knowledge regarding the cell’s biochemistry and operating condition, and thereby improve the efficiency of the search. One possible source of information is direct observations on the reaction fluxes. The experimental effort required to generate this data, however, can be substantial.

Here, we present a pathway search algorithm based on thermodynamic weights. We utilize the Gibbs free energy change (ΔG), a metric whose sign predicts if the reaction favors the formation of the reactants (positive sign) or products (negative sign). A ΔG close to zero indicates that a reaction is near equilibrium. Among parallel reactions, our algorithm selects the energetically favored or dominant reaction based on the sign and magnitude of the ΔG.

ICCAD’09, November 2–5, 2009, San Jose, California, USA.
Copyright 2009 ACM 978-1-60558-800-1/09/11...$10.00.

The major contributions of this paper are: formulating the dominant-edge pathway problem and solving it utilizing an efficient graph-based approach. These contributions are significant as they offer a computationally tractable framework for metabolic pathway analysis. To the best of our knowledge, the DOMINANT-EDGE PATHWAY algorithm is the first to address the metabolic pathway search problem through graph optimization techniques. Compared to enumeration approaches such as EFM analysis, our algorithm offers a substantial advantage in scalability.

The paper is organized as follows. The next section presents background material related to modeling metabolic networks, EFM analysis, and Gibbs free energy. Section 3 defines terms specific to our algorithm. Section 4 provides a problem statement. Section 5 describes the algorithmic solution to this problem. Section 6 evaluates our algorithm's capabilities and performance using three representative tests cases. Section 7 summarizes and discusses the major findings.

2. BACKGROUND

2.1 Metabolic Networks

A metabolic network consists of metabolites and reactions. A metabolite is classified either as internal or external. An internal metabolite is assumed to be at steady-state. With this assumption, the following mass conservation relationship is applicable: the total rate of production of an internal metabolite equals its total rate of consumption. The steady-state assumption is not applied to an external metabolite, i.e. its concentration may vary over time. The general equation for reaction R_i is written as follows: $\alpha_iX_i + \alpha_jX_j + \ldots \rightarrow \beta_iY_i + \beta_jY_j + \ldots$. This equation states that α_i molecules of metabolite X_i, α_j molecules of X_j, etc. are transformed into β_i molecules of metabolite Y_i, β_j molecules of metabolite Y_j, etc. Metabolites X and Y are referred to as *reactants* and *products*, respectively.

The metabolic network is structurally represented as a directed graph. A vertex represents a metabolite, and can either be an internal or external vertex. An edge represents a reaction, or part of a reaction if it involves more than a single reactant and product. When an edge represents part of a reaction we refer to the edge as a sibling edge. The reactants or products of a single reaction are referred to as sibling vertices. Reactions may be reversible or irreversible. A reversible reaction is represented by two edges with opposite directions.

Example 1: Figure 1 illustrates a graph representation of a sample metabolic network. The network consists of metabolites A … I, and reactions R_1, R_2, … R_8. In the figure, reaction R_1: A → B involves only one reactant (A) and one product (B). Reaction R_4: B → E + F involves one reactant (B) and two products (E and F). Vertices E and F are sibling vertices. Reaction R_4 is represented by two sibling edges to reflect the proper stoichiometric weight.

2.2 Elementary Mode and Flux Analysis

An elementary flux mode (EFM) refers to a minimal (non-decomposable) set of reactions that could operate at steady state, with the reactions weighted by their relative fluxes. In principle, any steady state flux pattern can be expressed as a non-negative linear combination of these modes. The EFM analysis produces an exhaustive enumeration of all feasible reaction routes supported by a metabolic network at steady-state. Since its introduction, continued improvements have been made to the implementation of the EFM algorithm [20]. However, complete enumeration of the EFMs for large (e.g. genome) scale metabolic networks remains computationally intractable, as the number of distinct reaction routes may exceed several million. In cases where the computation is tractable, EFM analysis has yielded a number of useful design insights for metabolic engineering. Examples include improving the production of a desired metabolite [3] and enhancing recombinant protein production in bacteria [18].

$R_1 : A \rightarrow B$
$R_2 : B \rightarrow C$
$R_3 : C \rightarrow D$
$R_4 : B \rightarrow E + F$
$R_5 : B \rightarrow 2G$
$R_6 : G \rightarrow H$
$R_7 : F \rightarrow H$
$R_8 : H \rightarrow I$

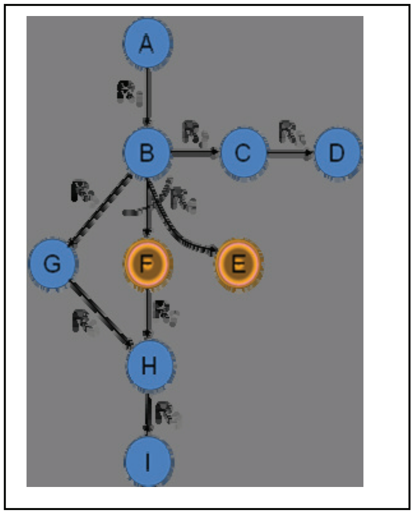

Figure 1. Example metabolic reactions and representative network.

In addition to pathway enumeration, EFMs may also be used to compute the steady-state reaction flux distribution of a metabolic network. Computing the flux distribution requires the estimation of weights that define the contribution of each EFM's flux (activity) to the overall network flux. In practice, obtaining these weights is a difficult task, with an experimental effort requirement comparable to that for metabolic flux analysis (MFA).

Example 2: Figure 2 illustrates the decomposition of the example network in Figure 1 into three elementary modes.

2.3 Free Energy

Gibbs free energy is most useful for chemical processes at constant temperature and pressure (isothermal and isobaric) and often used in biology [13]. In this paper, we use the standard Gibbs free energy change to estimate the expected likelihood of the corresponding reaction. The Gibbs free energy change (ΔG) of a reaction is a thermodynamic quantity whose sign in principle indicates whether a reaction is likely to occur (negative) or not occur (positive) spontaneously. Very recently, a related thermodynamic quantity, entropy (ΔS), of an EFM has been shown to significantly correlate with its flux [21]. Here, we use group contribution theory [5] to estimate the standard ΔG (ΔG°) values of metabolic reactions, which in turn serves as a first-order approximation of the "true" ΔG values under a well-defined and idealized condition (1 M concentrations of all reactants and

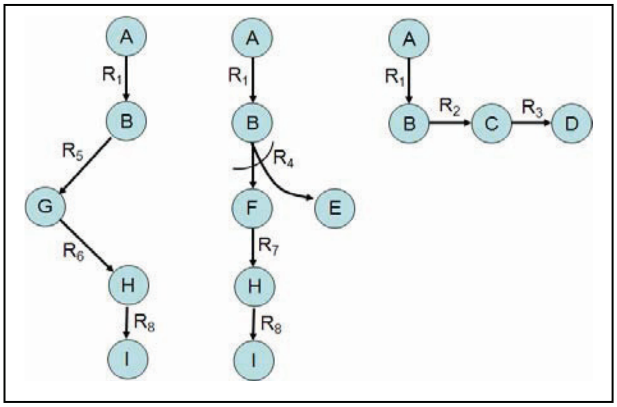

FIGURE 2. EFM decomposition of the network in Figure 1. Note that the EFMs represent alternative and partially overlapping reaction routes for the network input, in this case A.

products, 25 °C and neutral pH). Our algorithm uses ΔG to weigh the edges to conceptually simplify the formulation.

3. PRELIMINARIES

3.1 Representation of Metabolic Networks

We represent a metabolic network as a graph G_m= (V, E), where V and E are sets of metabolites and reactions, respectively. We associated a value g_e with each edge, representing the ΔG of the corresponding reaction. A path $s \rightarrow t$ is defined as a sequence of vertices and edges starting from vertex *s* and ending at vertex *t*.

A transpose of the network graph is obtained by reversing the direction of every edge, maintaining sibling relationships. We utilize the union operator ∪ between a subgraph N and a set of edges or nodes to create an equal or larger subgraph in terms of number of nodes and/or edges.

3.2 Dominant Paths

The algorithmic objective is to identify a network path that is energetically favored or “dominates” in the production of a particular metabolite. The limiting step (bottleneck) in production along a path is the reaction (edge) that has the smallest g_e (i.e. least negative ΔG). Among several parallel paths, a dominant-edge path will have the largest limiting step. The bottleneck shortest path is a well-known problem [10]. We therefore utilize terms similar to those in the Bottleneck Shortest Path Problem, whose goal is to determine the limiting capacity of any path between two specified vertices in a given network [19].

The *bottleneck energy* b_p of a path *p* from *s* to *t* is defined as:

$$b_p = max_{e \in p}\, g_e$$

The edge along *p* responsible for setting the bottleneck energy for the path is referred to as the *bottleneck edge* for path *p*. If *s* = *v*, then $b_p = \infty$.

The bottleneck of a vertex *t* is defined as:

$$b_v = min_{p:p \text{ is a } s \rightarrow t \text{ path}}\, b_p$$

3.3 Stoichiometrically Balanced Pathways

Isolating a *path* in a graph sense is desirable and meaningful in many conventional applications (e.g., traveling salesman problem, network flow algorithms). However, in the context of biological applications, it is more meaningful to identify a *pathway*. A *pathway* from *s* to *v* is a subgraph in the network that contains a path $s \rightarrow v$, and an augmenting set of connected edges and nodes. These augmenting components are needed to ensure overall stoichiometric balance. That is, a stoichiometrically balanced pathway will not have any dangling internal nodes. The augmenting components can also be thought of as paths that complete the conversion of any remaining intermediates (unused by the main path) to the target metabolite.

4. PROBLEM

We seek to solve the following problem: Given a metabolic network graph $G_m = (V, E)$, and starting and ending vertices *s* and *t,* find the dominant-edge pathway from *s* to *t*. In this paper, we define a dominant-edge pathway based on Δ*G*. However, we could also use other measures such as flux data, if available, to determine the dominant-edge pathway.

We identify two sub-problems. The first involves finding the dominant-edge path $s \rightarrow t$, and the second consists of augmenting the dominant-edge path to create a stoichiometrically balanced pathway. The first problem resembles the Bottleneck Shortest Path Problem. However, as we explain shortly, when we solve the first problem, we not only find a path from $s \rightarrow t$, but we also find additional sibling edges and sibling nodes that are integral parts of the dominant-edge pathway. The second problem therefore involves graph traversals to identify the relevant augmenting components to produce a stoichiometrically balanced pathway. We therefore provide an algorithm, DOMINANT-EDGE PATHWAY, which first finds a partially dominant pathway, PDP, and then augments it to produce a stoichiometrically balanced pathway, SBP.

5. ALGORITHM – DOMINANT-EDGE PATHWAY

The details of our algorithm are given in Figure 3. Algorithm DOMIANNT-EDGE PATHWAY begins by finding a set of edges *R* that contains all edges responsible for setting the bottleneck energy for all vertices in G_m. Next, based on *R,* the function EXTRACT-DOMINANT-PATH determines the dominant-edge path from $s \rightarrow t$, along with all sibling edges and vertices associated with this path. That path is referred to as PDP. Then, to ensure stoichiometric balance, our augmentation technique must be implemented iteratively in both the forward and backward directions, because sibling edges and vertices can occur in either the forward or reverse direction. Therefore, we first call AUGMENT-PATHWAY based on PDP, and then we call AUGMENT-PATHWAY based on the transpose of PDP. The process repeats until SBP does not grow.

To find the dominant-edge path, we utilize a modified Dijkstra’s algorithm [4], which identifies the single-source shortest path. In Dijkstra’s algorithm, all distances are initialized to infinity with the exception of the source vertex distance, which is initialized to zero. Each vertex’s predecessor is set to NIL. Dijkstra’s algorithm

```
DOMIANNT-EDGE-PATHWAY ($G_m$, s, t)
1-  R := FIND-BOTTLENECK-ENERGIES (s)
2-  PDP := EXTRACT-DOMINANT-PATH (s, t, R)
3-  SBP := PDP
4-  while SBP is growing
5-      SBP := AUGMENT-PATHWAY (s, t, SBP)
6-      SBP := SPB∪transpose (AUGMENT-PATHWAY (t, s, transpose (SBP)))
7-  return SBP

FIND-BOTTLENECK-ENERGIES (s)
1-  INITIALIZE-DOMINANT-PATH (s)
2-  Q := the set of all nodes in $G_m$ except s
3-  S := {s}; R := {}
4-  while Q is not empty
5-      x := extract lowest energy vertex in Q
6-      r := {} if previous[x] is undefined OR r := edge(previous[x], x)
7-      U := set of products of r in Q ∪ {x}
8-      S := S ∪ U ; R := R ∪ {r}
9-      for each vertex v in U
10-         RELAX (v)
11-         remove v from Q
12- return R

INITIALIZE-DOMINANT-PATH (s)
1-  for each vertex v in $G_m$
2-      energy[v] := ∞
3-      previous[v] := undefined
4-      reaction[v] := undefined
5-  for each neighbor v of s
6-      energy[v] := $g_e$(edge(s,v) )

RELAX (u)
1-  for each neighbor v of u
2-      alt := max(energy[u], $g_e$(edge(u,v) ) )
3-      if alt < energy[v]
4-          energy[v] := alt
5-          reaction[v] := edge (u, v)
6-          previous[v] := u

EXTRACT-DOMINANT-PATH(s, t, R)
1-  PDP = {t}
2-  u := t
3-  while u is not equal to s,
4-      PDP = {previous[u], reaction[u]} ∪ PDP
5-      if edge e is a sibling edge, then
                PDP := PDP ∪ {sibling edges(u)}∪ {sibling vertices(u)}
6-      u := source (e)

AUGMENT-PATHWAY (s, t, PDP)
1-  Augment-More := TRUE
2-  while Augment-More
3-      Augment-More:= FALSE
4-      for each vertex v in PDP
5-          if outdegree(v) = 0 and v is not t
6-              R' := FIND-BOTTLENECK-ENERGIES (v)
7-              PDP = PDP ∪ EXTRACT-DOMINANT-PATH(v, t, R')
8-              Augment-More := TRUE
9-  return PDP
```

Figure 3. Pseudo code for DOMINANT PATHWAY Algorithm

utilizes relaxation. Relaxing an edge *(u,v)* checks if the shortest distance to *v* found so far can be improved by going through *u*, and if so, the shortest distance to *v* is updated. The predecessor to *v* responsible for this new shortest path value is also updated. Dijkstra's algorithm maintains a set *S* of vertices whose shortest-path weights from the source have already been determined. The algorithm repeatedly selects the vertex *u*, not in *S*, with the minimum shortest-path estimate, adds *u* to *S*, and relaxes all edges leaving *u*. A min-priority queue keyed by the distance values of the vertices is used to efficiently extract *u*.

Our algorithm, FIND-BOTTLENECK-ENERGIES, differs from Dijkstra's shortest path algorithm as follows. We associate with each vertex three variables: energy, reaction and previous. energy[*v*] refers to the bottleneck energy of path from source to the vertex *v* that will be assigned to *v*. reaction[*v*] refers to the edge from a vertex *u* to *v* responsible for setting energy[*v*]. previous[*v*] refers to a vertex *u* connected to *v* through an edge (*u*, *v*) where *u* is responsible for setting reaction[*v*]. The initialization step sets the energies to ∞, except for the source vertex from which the search begins. Note that an edge in our graph may have more than one source, and thus we need both reaction and previous variables for implementing the algorithm. Another difference is in the relaxation step. The energy assigned to a vertex v is the maximum energy of g_e, the ΔG associated with edge *e* leading from *u* to *v*, and the energy of vertex *u*. We utilize a min-priority queue, *Q*, keyed by energy of the vertices stored in *Q*. The set *S* stores all the vertices whose bottleneck energies have already been determined.

The algorithm FIND-BOTTLENECK-ENERGIES works as follows. While visiting vertices, this algorithm models the effect of selecting favored reactions by including minimum energy vertex (metabolite) into its frontier, *S*. All variables are initialized as shown in INITIALIZE-DOMINANT-PATH. *Q* is initialized with all nodes in G_m. The algorithm repeatedly extracts a vertex *x* with the minimum energy, and process it as follows. First, the set of all sibling vertices associated with vertex *x* are found and stored into a set *U* (steps 6 & 7). In step 8, *U* is added to *S,* as the bottleneck energy of all vertices in *U* are now determined. This step ensures that once a reaction was used to set the bottleneck energy of a vertex, the energy of all sibling vertices are set and cannot be changed by further processing of the vertices. Similarly, *R* is augmented to include an edge *r* responsible for placing a vertex *x* in *S*. Each outgoing edge of the sibling vertices is then relaxed. The extraction continues until all vertices in Q have been processed.

Once the bottleneck energy from source vertex *s* to every node in the graph and each reaction[*v*] values are found, all edges in R are removed that do not belong to the dominant-edge path. Function EXTRACT-DOMINANT-PATH executes a traversal from the target to the source, adding vertices and edges to PDP, including sibling vertices and edges. The traversal includes sibling edges and sibling vertices and thus results in a PDP (as opposed to a path).

The function AUGMENT-PATHWAY finds a dangling node *d* (no outgoing edges) in PDP (line 5), and finds a partial dominant pathway, PDP, from *d* to *t.* This operation occurs by computing bottleneck energies starting with *d* using FIND-BOTTLENECK-ENERGIES, and then adding vertices and edges found using EXTRACT-DOMINANT-PATH between *d* and *t.* This process

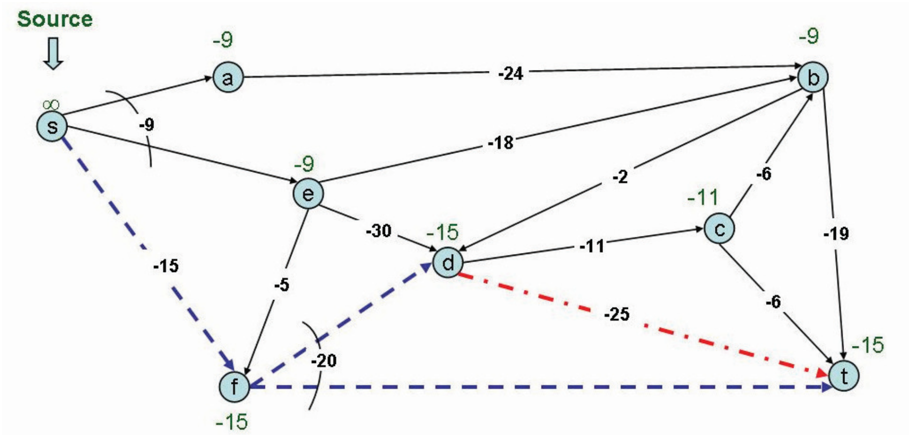

FIGURE 4. Example metabolic network. Numbers along edges indicate the Gibbs Free Energy Change. Numbers above each vertex denote bottleneck energies associated with each vertex. The dashed and dotted lines are the edges associated with the dominant pathway.

applies to all dangling nodes originally in PDP as well as nodes found during finding partial dominant pathways from *d*.

The runtime of FIND-BOTTLENECK-ENERGIES is similar to Dijkstra's algorithm. It depends on the implementation of the priority queue. For a binary max-heap implementation when all vertices are reachable from the source, the run time is *O(|E| lg |V|)*. The run time of EXTRACT-DOMINANT-PATH is *O(|V|)*. The run time for AUGMENT-PATHWAY is dominated by FIND-BOTTLENECK-ENERGIES, which is executed multiple times, but less than |*V*|. Based on our empirical results, the number of times FIND-BOTTLENECK-ENERGIES is called is typically small, and can be treated as a constant.

Example 3. The goal is to find the dominant-edge pathway from vertex *s* to vertex *t* in figure 4. The energy of each reaction edge is marked along the edges. Consider four parallel paths from *s* to *t*: {*s, e, f, d, t*}, {*s, e, d, c, t*}, {*s, f, d, t*}, and {*s, f, t*}. The bottleneck energy along each of these paths is -5, -9, -15, and -15. In this example, the first two paths are thus not dominant paths. However, the last two paths contain sibling vertices *d* and *t*. Our dominant pathway cannot have one of the paths and not the other to ensure stoichiometric balance. If, for example, we chose vertices {*s, f, t*}, then the reaction with Gibbs energy -20 will produce metabolite *d*. Therefore will must include *both* of these paths to produce a dominant-edge pathway including vertices {*s, f, t, d*}, and edges {(*s, f*), (*f, t*), (*f, d*), (*d, t*)}. The bottleneck energies associated with applying FIND-BOTTLENECK-ENERGIES are denoted next to each vertex. The dashed edges are found using EXTRACT-DOMINANT-PATH. The dotted edge (*d, t*) is found using AUGMENT-PATHWAY.

6. RESULTS

The DOMINANT-EDGE PATHWAY algorithm was tested on three examples with varying numbers of metabolites and reactions. Our results were compared to those found using the EFM analysis tool *efmtool* [16]. The examples are culled from the literature as there currently are no benchmark suites available to evaluate our algorithm. We computed the Δ*G* for each reaction using an available on-line tool [6]. The first test case consists of 21 metabolites and 20 reactions and includes pathways comprising the central carbon network of *Zymomonas mobilis* expressing heterologous enzymes for *xylose* utilization [1]. The number of EFMs for this model was 2. The second test case is based on a recently published model of an ethanol producing strain of *Escherichia coli* [17]. This network consists of 47 metabolites and 60 reactions, with three metabolites inputs: fructose, glucose, and xylose. The number of EFMs for this model was 33,000. We modified the second test case by removing a reaction responsible for biomass production (cell growth). We refer to this modified model as case 2A. The number of edges of the graph for 2A was significantly reduced because the reaction removed was associated with several sibling edges. The third test case is a model of the rat liver cell [9]. This model consists of 38 metabolites and 60 reactions. While this model is of the same scale as the E. coli test models, it supports a larger number of reversible input-output pairings. A more detailed liver model with 110 metabolites and 119 reactions is also considered as test case. This detailed model is referred to as 3A.

We highlight the relationship between the EFMs and the pathways found using DOMINANT-EDGE PATH, before presenting the results of each case study. One way to use EFM analysis to find the dominant-edge pathway is to analyze all elementary modes connecting the source and target metabolites. This subset of elementary modes can then be rank-ordered based on the least negative reaction ΔG in each mode to identify the pathway containing the lowest thermodynamic barrier. The pathway(s) found using this method may coincide, be part of, or partially overlap with the dominant-edge pathway. EFM analysis does not necessarily find the same pathway identified using DOMINANT-EDGE PATHWAY as EFM finds all possible pathways that can cover the source and destination metabolites.

The overlap possibilities between EFM and DOMINANT–EDGE PATHWAY results are summarized in Table 1. The columns in the table indicate the following: test case number, source metabolite, target metabolite, number of metabolites and reactants along the dominant-edge pathway found using our algorithms, and the number of relevant modes found by EFM.

In the first test case, the EFM and DOMINANT-EDGE PATH analyses produce identical pathways. In the second example (test case 2), the three dominant-edge pathways, each corresponding to a different input metabolite, are proper subsets of 169, 156, and 725 elementary modes. In test case 2A, two of the dominant-edge

TABLE 1. Results of DOMINANT PATH Algorithm compared to paths (modes) found using EFM analysis

Test Case	Inputs	Output	No. of Metabolites	No. of Reactions	EFM Modes
1	Glucose	Ethanol	13	12	1
1	Xylose	Ethanol	20	19	1
2	Fructose	Ethanol	12	11	169
2	Glucose	Ethanol	13	12	156
2	Xylose	Ethanol	21	19	725
2A	Fructose	Ethanol	12	11	1
2A	Glucose	Ethanol	13	12	1
2A	Xylose	Ethanol	21	19	1
3	Alanine	Glucose	10	10	14
3	Alanine	Urea	8	8	3
3	Cysteine	Glucose	10	10	14
3	Cysteine	Urea	8	8	3
3	Glycine	Alanine	2	3	1
3	Glycine	Glucose	11	11	20
3	Glycine	Cysteine	2	3	1
3	Glycine	Urea	9	9	9
3	Tyrosine	Glucose	12	1	18
3	Tyrosine	Urea	8	7	5
3	Acetyl-CoA	Glucose	15	14	30
3	Acetyl-CoA	Urea	11	10	9
3	Serine	Glucose	10	10	15
3	Serine	Urea	8	8	5
3A	Alanine	Glucose	33	33	-
3A	Alanine	Urea	13	15	-
3A	Cysteine	Glucose	54	62	-
3A	Cysteine	Urea	17	17	-
3A	Glycine	Alanine	47	52	-
3A	Glycine	Glucose	56	63	-
3A	Glycine	Cysteine	6	4	-
3A	Glycine	Urea	20	19	-
3A	Tyrosine	Glucose	42	46	-
3A	Tyrosine	Urea	38	42	-
3A	Acetyl-CoA	Glucose	33	33	-
3A	Acetyl-CoA	Urea	21	26	-
3A	Serine	Glucose	54	61	-
3A	Serine	Urea	21	20	-

pathways are identical to those found using EFM, and the third pathway was having partial overlap with EFM. In the third example, there was only partial overlap between the dominant-edge pathways and the modes found using EFM analysis. The number reported in the last column in the table indicates the number of elementary modes that contained at least 50% of the reactions found in the corresponding dominant-edge pathway. For test case 3A, the EFM analysis cannot be completed. However, over 1.5 million modes were reported by the tool before crashing after 3 days of execution. We therefore report only the number of metabolites and the number of reactions present in dominant-edge pathways.

The significance of our algorithm thus lies in its ability to efficiently identify thermodynamically favored reaction routs without costly enumeration-based path analysis. This is evident in the runtime and memory requirements needed to perform the analysis. The run time for all test cases was < 1 second using a single 3 GHz quad-core Pentium computer with 4 GB of RAM. The *efmtool* run time was 1 second for the first and third examples, and 10 seconds for the second example. When the Dominant-Edge Pathway algorithm was applied to the detailed model, the resulting pathways had similarity with the pathways found using reduced model while maintaining a run time of less than 1 second.

7. CONCLUSION

This paper presents a novel algorithm for biochemical pathway analysis. Our DOMINANT-EDGE PATHWAY algorithm departs from prior efforts on exhaustive enumeration in the following ways. Given a desired pathway feature, e.g. thermodynamic favorability, our algorithm merges the weight assignment and the path identification steps. The algorithm provides an efficient search process compared to enumeration-based approaches such as EFM and extreme pathway analysis. Stoichiometric balancing is applied at the end of the search process, after the main trunk of the path has been generated, again saving run-time. The chief limitation of our algorithm deals with the uncertainty of the Gibbs free energy estimates used to characterize the thermodynamic favorability of the reactions. On the other hand, the algorithm is general with respect to the type of the reaction (edge) weight, and could be expanded to use measurement derived steady-state flux weights. In conclusion, the results of our analysis indicate that the algorithm presented in this paper provides an efficient alternative to the enumeration based approaches, especially for applications where the input and output metabolites are *a priori* defined.

8. ACKNOWLEDGMENTS

Our sincere thanks to Gautham Sridharan, a graduate student in Chemical and Biological Engineering at Tufts, for his help in preparing the data for the test cases. This work was funded by NFS grant no. 0829899.

9. REFERENCES

[1] Altintas, M.M., Eddy, C.K., Zhang, M., McMillan, J.D. and Kompala, D.S. 2006. Kinetic modeling to optimize pentose fermentation in Zymomonas mobilis. *Biotechnol Bioeng 94*, 273-295.

[2] Atsumi, S., Hanai, T. and Liao, J.C. 2008. Non-fermentative pathways for synthesis of branched-chain higher alcohols as biofuels. *Nature 451*, 86-89.

[3] Carlson, R., Fell, D. and Srienc, F. 2002. Metabolic pathway analysis of a recombinant yeast for rational strain development. *Biotechnol Bioeng 79*, 121-134.

[4] Dijkstra, E.W. 1959. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik 1*, 269-271.

[5] Forsythe, R.G., JR., Karp, P.D. and Mavrovouniotis, M.L. 1997. Estimation of equilibrium constants using automated group contribution methods. *Comput Appl Biosci 13*, 537-543.

[6] Jankowski, M.D., Henery C.S., Broadbelt, L.J. and Hatzimanikatis, V. 2008. Group contribution method for thermodynamic analysis of complex metabolic networks. *Biophys J 95*, 1487-1499.

[7] Klamt, S. and Stelling, J. 2002. Combinatorial complexity of pathway analysis in metabolic networks. *Mol Biol Rep 29*, 233-236.

[8] Lee, S.K., Chou, H., Ham, T.S., Lee, T.S. and Keasling, J.D. 2008. Metabolic engineering of microorganisms for biofuels production: from bugs to synthetic biology to fuels. *Curr Opin Biotechnol 19*, 556-563.

[9] Nolan, R.P., Fenley, A.P. and Lee, K. 2006. Identification of distributed metabolic objectives in the hypermetabolic liver by flux and energy balance analysis. *Metab Eng 8*, 30-45.

[10] Pollack, M. 1960. The Maximum Capacity through a Network. *INFORMS 8*, 733-736.

[11] RO, D.K., Paradise, E.M., Ouellet, M., Fisher, K.J., Newman, K.L., Ndungu, J.M., HO, K.A., Eachus, R.A., Ham, T.S., Kirby, J., Chang, M.C., Withers, S.T., Shiba, Y., Sarpong, R. and Keasling, J.D. 2006. Production of the antimalarial drug precursor artemisinic acid in engineered yeast. *Nature 440*, 940-943.

[12] Roca, C., Haack, M.B. and Olsson, L. 2004. Engineering of carbon catabolite repression in recombinant xylose fermenting Saccharomyces cerevisiae. *Appl Microbiol Biotechnol 63*, 578-583.

[13] Rodriguez, J., Lema, J.M. and Kleerebezem, R. 2008. Energy-based models for environmental biotechnology. *Trends Biotechnol 26*, 366-374.

[14] Schuster, S., Fell, D.A. and Dandekar, T. 2000. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nat Biotechnol 18*, 326-332.

[15] Sonderegger, M., Schumperle, M. and Sauer, U. 2004. Metabolic engineering of a phosphoketolase pathway for pentose catabolism in Saccharomyces cerevisiae. *Appl Environ Microbiol 70*, 2892-2897.

[16] Terzer, M. and Stelling, J. 2008. Large-scale computation of elementary flux modes with bit pattern trees. *Bioinformatics 24*, 2229-2235.

[17] Trinh, C.T., Unrean, P. and Srienc, F. 2008. Minimal Escherichia coli cell for the most efficient production of ethanol from hexoses and pentoses. *Appl Environ Microbiol 74*, 3634-3643.

[18] Vijayasankaran, N., Carlson, R. and Srienc, F. 2005. Metabolic pathway structures for recombinant protein synthesis in Escherichia coli. *Appl Microbiol Biotechnol 68*, 737-746.

[19] Volker Kaibel, M.A.F.P. 2006. On the Bottleneck Shortest Path Problem. *ZIB-Report 06-22*.

[20] Von Kamp, A. and Schuster, S. 2006. Metatool 5.0: fast and flexible elementary modes analysis. *Bioinformatics 22*, 1930-1931.

[21] Wlaschin, A.P., Trinh, C.T., CARLSON, R. and Srienc, F. 2006. The fractional contributions of elementary modes to the metabolism of Escherichia coli and their estimation from reaction entropies. *Metab Eng 8*, 338-352.

[22] Yeung, M., Thiele, I. and Palsson, B.O. 2007. Estimation of the number of extreme pathways for metabolic networks. *BMC Bioinformatics 8*, 363.