The Minimal Seed Set Problem

Avitan Gefen and **Ronen I. Brafman**
Department of Computer Science
Ben-Gurion University of The Negev, Israel

Abstract

This paper defines and studies a new, interesting, and challenging benchmark problem that originates in systems biology. The *minimal seed-set problem* is defined as follows: given a description of the metabolic reactions of an organism, characterize the minimal set of nutrients with which it could synthesize all nutrients it is capable of synthesizing. Current methods used in systems biology yield only approximate solutions. And although it is natural to cast it as a planning problem, current optimal planners are unable to solve it, while non-optimal planners return plans that are very far from optimal. As a planning problem, it is inherently delete-free, has many zero-cost actions, all propositions are landmarks, and many legal permutations of the plan exist. We show how a simple uninformed search algorithm that exploits inherent independence between sub-goals can solve it optimally by reducing the branching factor drastically.

Introduction

Organism depend on their environment for the supply of certain nutrients, while they can synthesize others on their own. Researchers in the life sciences have long been studying these metabolic processes, and have accumulated a lot of data on this topic. Large databases describing the metabolic reactions characteristic to different organisms exist, and this information is used to study issues such as the evolution of species and their environments, the effect of environmental changes, etc. Researchers in systems biology have organized these metabolic reactions in a graphical structure (essentially a hyper-graph) called a metabolic network whose nodes correspond to nutrients and whose edges correspond to reactions – typically, but not necessarily, of a single organism – and have used this as a tool in the study of biological systems. A seed-set of such a network is a set of nutrients from which one can produce the entire set of nutrients in this network via reactions. In particular, systems biologists have thought to characterize minimal seed-sets for organisms, using which they can study questions such as: What is the effective biochemical environment of a specific species? How the structure of the organism biochemical network correspond to its life-style? And how biochemical networks of organisms evolve?

The goal of this paper is to suggest and study the problem of computing minimal seed-sets as a novel, interesting and challenging planning benchmark problem that is motivated by a real-world application. This problem has interesting special structure that seems to prevent existing optimal planners from solving: it is delete-free, has many zero-cost actions, every proposition is a landmark, and every legal plan for it has many legal permutations. Yet, this structure also allows for serious pruning that considerably reduces its effective branching factor and allows us to solve all instances of this problem in the KEGG database (Kanehisa et al. 2006).

The seed set problem

The study of minimal seed-sets and their applications in systems biology was initiated by Borenstein *et. al.* (2008) who defined the problem as follows: Let C be the set of nutrients associated with a specific organism. A biochemical (or metabolic) reaction is an ordered pair of sets $r = (X, Y)$. $X \subseteq C$ will be called the *substrate* set of nutrients and $Y \subseteq C$ the *product* set. This relationship is often written in the following way: $x_1 + \cdots + x_n \rightarrow y_1 + \cdots + y_m$. Many reactions are bi-directional, in this case we will represent the reaction as both (X, Y) and (Y, X). Note that reactions in this model do not remove the substrate nutrients, but only adds the product nutrients. The *metabolic network* of a given organism, R, can now be defined as the set of metabolic reactions associated with that organism. We use C to denote the nutrients that appear in the metabolic network R, that is all the nutrients that are either part of the substrate or product of a reaction in R.

A set of nutrients is *reachable* from a subset of nutrients if there exist a finite sequence of reactions, such that after applying this sequence, all nutrients are present. The *seed set* of a metabolic network is subset of nutrients from which C is reachable, meaning that any nutrient in C is either part of the seed set, or can be synthesized via some sequence of reactions from this seed set. A *minimal* seed set represents a minimal set of nutrients that the environment must provide the organism in order to exercise its full potential. In most real metabolic networks there are many minimal seed sets.

Seed Set Generation: Existing method

Finding a minimal seed set is NP-hard (e.g., by reduction from the set-cover problem), and it seems natural to cast it

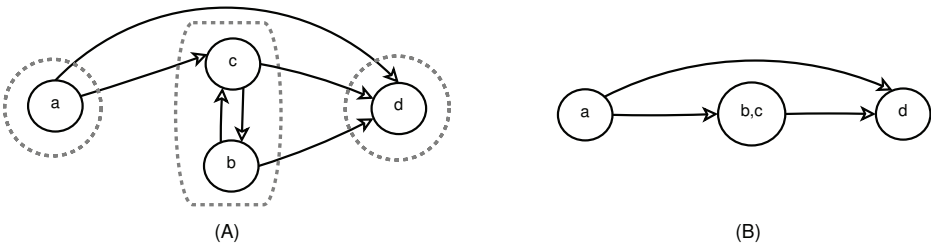

Figure 1: (A) graph representation G, for: $r_1 : a + b \rightarrow c + d$ and $r_2 : c \rightarrow b + d$ (B) G_{scc} of G

as a mixed-integer programming problem. Borenstein *et. al.* (2008) tried this approach, and they report that it does not scale up.[1] Consequently, they resorted to an approximation algorithm.

Their solution method first "flattens" the original hypergraph representing the metabolic network into a (regular) directed graph, known as a *directed substrate graph* (Klamt, Haus, and Theis 2009), which is commonly used in zing metabolic networks. The directed substrate graph is a digraph $G = (V, E)$, where V is the set of nutrients (C) and a directed arc $a = (x, y) \in E$ exists if and only if there is a reaction $r = (X, Y)$ where $x \in X$ and $y \in Y$. Naturally, this transformation already leads to some loss of information because the interaction between the different input nutrients is ignored. An example of a directed substrate graph built out of two reactions, is illustrated in Fig.1.A.

Next, we identify the strongly connected components (SCC) of G. The SCC's of G form a directed acyclic graph (DAG) the G_{scc}, as in Fig.1.B. A node in the G_{scc} is a SCC of G, for example, the node "b, c" in Fig.1.B is a SCC of G from Fig.1.A composed of nodes b and c. There is an edge from node n to n' if there is an edge from some node in the SCC represented by n to some node in the SCC represented by n'. Each node in the G_{scc} which has no incoming edges and at least one outgoing edge will be called a *source component node*, and it will represent a special type of SCC of G which we will call a *source component set*. In Fig.1.B the only source component node is a.

Since a source component node (of G_{scc}) has no incoming edges, none of the nutrients outside this component set (SCC in G) can be a precursor for any nutrient in this source component. Hence, at least one element of this source component must be part of any seed set. In Borenstein *et. al.* (2008) a representative element of this source component is selected randomly. This method does not find an accurate solution to the seed-set problem. For example, in Fig.1 the only source component has one node a, which by itself can not produce all nutrients (or even one other nutrient).

Seed Set Generation as Planning

When we ignore the quantitative aspects of chemical reactions – which is what researchers investigating the structure and topology of metabolic networks often do – it is not difficult to see that they are very much like planning operators, with domain variables describing the (in)existence of nutrients. They have input, or preconditions to their application, and they have output, or new effects.

Indeed, viewing organisms as dynamic systems, and reactions as operators that change the state of this system, the use of planning techniques in this context seems well motivated. In planning terms, a minimal seed-set is a minimal (w.r.t. true propositions) initial state from which the goal state of having all nutrients, is reachable via the available set of reactions. Although not a planning problem, in the classical sense, it can easily be cast as one, as we show next.

We model the minimal seed-set problem as a planning problem as follows: The set of facts corresponds to the set of nutrients, C. We have one proposition for each nutrient, which is *true* when this nutrient is present, and *false* otherwise. One *zero-cost* planning operator o_r is associated with each reaction $r = (X, Y) \in R$. $pre(o_r) = X$, $add(o_r) = Y$. Applying an operator o in state s, written $s[o]$, results in state $s \cup add(o)$, assuming o is applicable in s, i.e., $pre(o) \subseteq s$. Next, we introduce one new "insert" operator for every nutrient with a fixed, positive cost, e.g., one. The *insert(c)* operator has no precondition and a single add effect, c. Finally, we define the initial state as the state in which all propositions are *false* (i.e., no nutrient is available), and the goal state is the one in which all propositions are *true*. The set of nutrients inserted via an insert action corresponds to a seed-set, and a minimal-cost plan will be a plan that minimize insert actions, and consequently is a minimal seed-set of minimal cardinality, as well.

As an example, consider the metabolic network in Fig.1: (i) The propositions are the set of nutrients $C = \{a, b, c, d\}$. (ii) There are two reaction operators: r_1 with $pre(r_1) = \{a, b\}$ and r_2 with $pre(r_2) = \{c\}$, while their add effects are $add(r_1) = \{c, d\}$ and $add(r_2) = \{b, d\}$. Both operators have zero cost. (iii) Four insert operators will be constructed, one for each of the nutrients a, b, c, d. Their precondition is empty, and their add effect is a single nutrient. These operators will have cost higher than zero. (iv) All propositions are false in the initial state and true in the goal state.

Notice that this planning problem is not a typical one: many operators have zero cost; non-zero cost operators have no preconditions; there are no conditional effects; actions are delete-free, and all propositions must be achieved, and hence they are trivially landmarks. Furthermore, because in real-metabolic networks most reactions do not interact with each other, i.e., they influence different nutrients, most plans

[1] Another classic technique – reduction to SAT – failed to return a solution on all but the smallest problem instance.

have many legal permutations.

We extracted metabolic reactions information from the KEGG database (http://www.genome.jp/kegg/) for many different organisms which are considered to be well characterized.[2] The metabolic network was transformed to PDDL using the reduction described above and they are available at http://www.cs.bgu.ac.il/∼avitang/files/Kegg.zip. We applied the FD planner (Helmert 2006) with two different types of heuristics: the landmarks-based LM-Cut heuristic (Helmert and Domshlak 2009), and the newest variant of the abstraction based Merge-and-Shrink heuristic (Helmert, Haslum, and Hoffmann 2007). Unfortunately, neither planners was able to solve even the smallest instance, which has 305 nutrients and 298 reactions, particularly, Merge-and-Shrink exhausted its memory after a few minutes.

We believe the failure of these two heuristics is due to the special nature of this domain: extremely circular and wide (high outdegree in the metabolic network) with a large branching factor. Moreover, the fact that the goal state is the entire set of nutrients, i.e., all propositions are landmarks, leaves little hope for landmark-based heuristics. Indeed, we observe that the non-optimal LAMA planner (Richter, Helmert, and Westphal 2008) returned uninformative solutions, consisting of *insert* actions only. Finally, for a planning task the order of the actions matters, requiring the A* algorithm to review all legal permutations of temporarily minimal plans, many of which have identical outcomes.

New method

To overcome the above problems, we devised a variant of the A* algorithm that exploits two special properties of this domain. Our first observation is that the presence of many zero cost actions (i.e., the reactions) adds much to the complexity of the problem, and this complexity can be avoided by simply applying all of relevant actions once a new nutrient is inserted. We accomplish this by tweaking the A* algorithm as follows: Define $scope(s)$ to be the state derived from s by repeatedly applying all possible zero cost actions until no new proposition can be added. Alter A* so that the children of a state s will be all states of the form $scope(s[o])$ (rather than $s[o]$), where o is an insertion action (i.e., non-zero cost action) applicable in s. Unfortunately, this algorithm failed to find an optimal plan in a reasonable time.

Next, we, attempted to deal with the issue of action ordering. First, let us recall that in a planning problem with no delete effects, every action should be applied at most once, and that there could be many legitimate permutations of a specific plan. Second, with no delete effects, and provided the level of interaction between propositions is not high, one may consider only a subset of all applicable actions at each state, without loosing optimality. To accomplish this we will define $G(s)$ (graph G for state s) to be the graph obtained from the original substrate graph by removing all nutrients that were achieved (i.e., are true facts in the state s) and all edges they participate in. The $G_{scc}(s)$ is defined as before for $G(s)$. For each state after applying all zero cost actions possible, we can consider only insert actions that produce nutrients that reside in one source component of the current state substrate graph $G(s)$. The reason for this becomes intuitive when looking at the $G_{scc}(s)$. Since a source component has no incoming edges, there is no precursor that can reach it other than the nutrients in the source component itself. Thus, at least one of these actions must be in the plan. That is, the action in a source component constitute a disjunctive landmark. Moreover, we can consider only the insert actions of a specific source component each time, as their order does not matter – none supplies a precondition for the other, and there are no conditional effects.

These observations, combined, lead to a variant of the A* algorithm in which not all applicable insert operators are applied in each state, while all applicable reactions are applied after each insert, and using the blind heuristics. For the reasons discussed above, this algorithm does not compromise the optimality of the problem, and it is able to solve the seed-set problem for all organisms in the KEGG database.

Specifically, our algorithm ExpandState(s) (Algorithm 1) chooses the minimal source component (in number of nutrients) and ignores insert actions for nutrients outside this component. Thus, our A* variant ignores most insert operators at each step and "attacks" different parts of the problem one at a time. The changes to A* are all concentrated in the ExpandState(s) (Algorithm 1). The resulting planner successfully finds optimal plans for even the largest metabolic networks, such as the network for humans, in just a few minutes. The optimality of our algorithm is guaranteed by the fact that at least one of the alternative actions examined at each point in time must be part of an optimal plan, and that the order by which we introduce elements of different source components does not matter because they do not interact.

Algorithm 1 Expand State

1: **ExpandState(**s**)**
2: $E \leftarrow \emptyset$ {initialize expanded states set}
3: $G(s) \leftarrow$ discard all true facts from original graph
4: $G_{scc}(s) \leftarrow$ build from $G(s)$
5: $M_{sc}(s) \leftarrow$ find in $G_{scc}(s)$ a source component which is minimal in size
6: **for all** $nutrient \in M_{sc}(s)$ **do**
7: $o \leftarrow$ insert operator of $nutrient$
8: $newState \leftarrow scope(s[o])$
9: $E \leftarrow E \bigcup \{newState\}$
10: **end for**
11: **return** E

Empirical results

We chose 22 organisms from different taxonomy categories, from small bacteria to mammals. Many of these organisms are well known, well studied, model-type organisms.

LM-Cut and Merge and Shrink were both unsuccessful in solving the problem in reasonable time. In order to see if we could find a derivative of the problem these planners can handle, we took the following steps: (i) When examining the metabolic networks we noticed that they contain many

[2] Organisms with draft genomes or EST contigs were excluded.

(around 40 - 50%) nutrients that are never produced, meaning they are not an effect of any reaction and therefore must be part of the seed-set. We used a preprocess step to create a derivative of the problem where all these nutrients are in the initial state and there are no insert actions for them. (ii) In case the cost scheme is not suited for the planners we also used 3 different cost schemes: (1) cost of reactions = 1, cost of insert = 10 (2) cost of reactions = 0, cost of insert = 10 (3) cost of reactions = 1, cost of insert = # of reactions.

The steps mentioned did not help any one of the planners to solve the smallest instance of the problem, but LM-Cut managed to run 30 minutes without exhausting its memory.

Organism	# of nutrients	# of reactions	LM -Cut	Merge & Shrink	GSCC (h=0)
aae	2576	1699	-	-	86.84
avn	305	298	-	-	1.92
ayw	1733	400	-	-	26.18
bmu	3042	2942	-	-	150.84
bra	3139	3556	-	-	174.88
bxe	3106	3722	-	-	177.36
ecc	2901	3137	-	-	145.86
eco	2992	3237	-	-	154.67
ecp	2918	3166	-	-	145.99
ecv	2890	3161	-	-	144.13
ecx	2956	3197	-	-	152.71
hsa	3006	4010	-	-	176.59
mmu	3004	3959	-	-	174.35
rha	3219	3679	-	-	187.69
gga	2986	3514	-	-	158.60
xla	2956	2971	-	-	143.72
dre	2977	3734	-	-	165.49
dme	2973	3099	-	-	151.77
ath	3322	3290	-	-	184.67
cre	2958	563	-	-	104.72
cme	2940	2371	-	-	129.51
sce	2622	2635	-	-	110.59

Table 1: Methods are measured by runtime in seconds. Names of organisms are KEGG shortcuts, for example: hsa is Homo sapiens (human). For a full table of names see http://www.genome.jp/kegg/catalog/org_list.html

Conclusion

We described a new challenge domain for optimal planning motivated by a real-world problem of interest to system biologists. Existing optimal planners are unable to solve this problem, although a specialized search algorithm we designed can solve all existing instances of this problem.

An interesting question for future work is how existing planners might be altered to solve this domain. Our algorithm can be viewed as generating a disjunctive landmark, all of whose actions are currently applicable, and branching on the different elements, and then applying all possible zero-cost actions. The idea of "applying all zero-cost actions" is somewhat reminiscent of the use of axioms (Thiébaux, Hoffmann, and Nebel 2003), that is, we could view reactions as axioms. However, axioms are used to generate derived predicates, whereas in our domain there is no natural notion of derived predicates – those nutrients that can be derived by reactions can also be obtained as the effect of *insert* actions. While the description could be altered to fit the requirements of axioms, the lack of support for axioms by existing optimal planners make the utility of this questionable. We believe that a more promising direction is to simply integrate both ideas (prune by branching on landmarks and apply all "useful" zero-cost actions) into an existing planner, and we hope to pursue it. In particular, it will be interesting to see if it is possible to find disjunctive action landmarks of the form used here more generally.

Because biology deals with complex dynamical systems, we believe it is worth exploring the possibility of additional planning problems of interest to biologists. In particular, we are presently trying to solve a more complex problem of generating a seed-set that is not only minimal in terms of the number of nutrients, but also in terms of the cost (e.g., energy, number of reactions) of generating the entire set of nutrients. Extension that take into account nutrient concentration could pose an interesting challenge for metric-planning.

Acknowledgements: We thank Elhanan Borenstein for invaluable help in understanding the seed-set problem and his methods, Or Caspi for working on the SAT encoding, and the reviewers and SPC member for useful comments and suggestions. The authors were partly supported by ISF Grant 1101/07, the Paul Ivanier Center for Robotics Research and Production Management, and the Lynn and William Frankel Center for Computer Science.

References

Borenstein, E.; Kupiec, M.; Feldman, M. W.; and Ruppin, E. 2008. Large-scale reconstruction and phylogenetic analysis of metabolic environments. *Proceedings of the National Academy of Sciences* 105(38):14482–14487.

Helmert, M., and Domshlak, C. 2009. Landmarks, critical paths and abstractions: What's the difference anyway? In Gerevini, A.; Howe, A. E.; Cesta, A.; and Refanidis, I., eds., *ICAPS*. AAAI.

Helmert, M.; Haslum, P.; and Hoffmann, J. 2007. Flexible abstraction heuristics for optimal sequential planning. In Boddy, M. S.; Fox, M.; and Thiébaux, S., eds., *ICAPS*, 176–183. AAAI.

Helmert, M. 2006. The fast downward planning system. *J. Artif. Intell. Res. (JAIR)* 26:191–246.

Kanehisa, M.; Goto, S.; Hattori, M.; Aoki-Kinoshita, K. F.; Itoh, M.; Kawashima, S.; Katayama, T.; Araki, M.; and Hirakawa, M. 2006. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research* 34(suppl 1):D354–D357.

Klamt, S.; Haus, U.-U.; and Theis, F. 2009. Hypergraphs and cellular networks. *PLoS Comput Biol* 5(5):e1000385.

Richter, S.; Helmert, M.; and Westphal, M. 2008. Landmarks revisited. In *AAAI'08: Proceedings of the 23rd national conference on Artificial intelligence*, 975–982. AAAI Press.

Thiébaux, S.; Hoffmann, J.; and Nebel, B. 2003. In defense of pddl axioms. In Gottlob, G., and Walsh, T., eds., *IJCAI*, 961–968. Morgan Kaufmann.