Extending the Metabolic Network of *Ectocarpus Siliculosus* using Answer Set Programming

Guillaume Collet[1,5], Damien Eveillard[2], Martin Gebser[3], Sylvain Prigent[4,5], Torsten Schaub[3], Anne Siegel[1,5], and Sven Thiele[5,6,1]

[1] CNRS, UMR 6074 IRISA, Campus de Beaulieu, 35042 Rennes, France.
[2] Université de Nantes, UMR 6241 LINA, 2 rue de la Houssinière, 44300 Nantes, France.
[3] Universität Potsdam, Institut für Informatik, August-Bebel-Str. 89, D-14482, Deutschland.
[4] University of Rennes 1, UMR 6074 IRISA, Campus de Beaulieu, 35042 Rennes, France.
[5] INRIA, Centre Rennes-Bretagne-Atlantique, Projet Dyliss, Campus de Beaulieu, 35042 Rennes cedex, France.
[6] INRIA-CIRIC, Rosario Norte 555, Of. 703, Las Condes, Santiago de Chile, Chile.

Abstract. Metabolic network reconstruction is of great biological relevance because it offers a way to investigate the metabolic behavior of organisms. However, reconstruction remains a difficult task at both the biological and computational level. Building on previous work establishing an ASP-based approach to this problem, we present a report from the field resulting in the discovery of new biological knowledge. In fact, for the first time ever, we automatically reconstructed a metabolic network for a macroalgae. We accomplished this by taking advantage of ASP's combined optimization and enumeration capacities. Both computational tasks build on an improved ASP problem representation, incorporating the concept of reversible reactions. Interestingly, optimization greatly benefits from the usage of unsatisfiable cores available in the ASP solver *unclasp*. Applied to *Ectocarpus siliculosus*, only the combination of *unclasp* and *clasp* allowed us to obtain a metabolic network able to produce all recoverable metabolites among the experimentally measured ones. Moreover, 70% of the identified reactions are supported by an homologous enzyme in *Ectocarpus siliculosus*, confirming the quality of the reconstructed network from a biological viewpoint.

1 Introduction

Systems biology is a field at the crossover of biology, computer science, and mathematics, which aims to elucidate the response of a living organism. Among all biological processes occurring in a cell, metabolic networks are in charge of transforming input nutrients into both energy and output nutrients necessary for the functioning of other cells. From an industrial viewpoint, it is crucial to estimate and control the capability of an organism to produce products of interest. Many computational and mathematical methods have been developed to model the response of such systems to external perturbations, and applied to well-studied organisms [1–3].

In the last few years, sequencing technologies have drastically evolved, such that it is now possible to sequence the genome of many less-studied organisms. As a natural follow-up, one needs to estimate the metabolic capability of an "exotic" organism

on the basis of its genome, and then apply well-established control methods to the network. The usual strategy consists in checking whether the genome contains known enzymatic "bricks", that is, genomic sequences that appropriately match with genomic sequences of enzymes characterized in other model organisms, such as *Escherichia coli* [4] or *Arabidopsis thaliana* [5], whose genomes and networks have been manually curated over several years [6]. The combination of metabolic reactions associated with the identified enzymes provides a draft of the metabolic network for the studied organism. The integration of the different heterogeneous bio chemical resources leads to inconsistencies and ambiguities in the draft network. Semantic web approaches solve these inconsistencies and rank the retrieved information by exploiting existing ontologies [7]. Nonetheless, genomes are of low quality and the expert community on "exotic" organisms is too small to provide a wide manual curation of this network. Concretely, automatic genome-scale reconstructed networks suffer from substantial incompleteness, and many networks are only partially defined. To overcome this limitation, the next step consists in *filling the gaps* of the draft network. To that end, we rely on reference databases of metabolic reactions and check whether adding such reactions to the network improves its ability to produce metabolite compounds of interest from the growth media of the organism. Several approaches to automatically reconstruct the missing parts of metabolic networks have been proposed. To restore a desired metabolic behavior they propose reactions (picked from reaction databases) that can be added to the network. The reactions are chosen to optimize either graph-based criteria [8] or a linear score modeling the quantitative production of the system [9]. The main limitation of all approaches is the increasing size of the search space, since reaction databases like KEGG[7] or MetaCyc[8] have substantially grown with the availability of high-throughput methods in molecular biology. Other studies propose to overcome this limitation by using sampling heuristics [10], but unfortunately they give little information on the size of solution sets and the quality of the sampling methods.

In previous work [11], we reformulated the gap filling problem as a qualitative combinatorial (optimization) problem, and modeled it using Answer Set Programming (ASP) [12]. The basic idea is that reactions apply only if all their reactants are available, either as nutrients or provided by other metabolic reactions. Starting from given nutrients, referred to as *seeds*, this allows for extending a metabolic network by successively adding operable reactions and their products. The set of metabolites in the resulting network is called the *scope* of the seeds and represents all metabolites that can principally be synthesized from the seeds. In metabolic network completion, we query a database of metabolic reactions looking for minimal sets of reactions that can restore the observed bio-synthetic behavior.

As a follow-up to [11], we attempted to apply the same approach to reconstruct the "exotic" metabolic network of *Ectocarpus siliculosus*, using the MetaCyc database. This organism is a brown algae that belongs to the heterokonts, whose closest relative (diatoms) exhibits a large phylogenetic distance to most other plant model species. Such distinctions make a reconstruction of the metabolic network of *Ectocarpus siliculosus* particularly challenging. In fact, we could not solve the reconstruction problem with the

[7] `http://www.genome.jp/kegg`
[8] `http://metacyc.org`

original approach that hits its limits with large databases like MetaCyc, which doubled in size over the last four years.

In this work, we push former limits by taking advantage of ASP's combined optimization and enumeration capacities. For one, we introduce an improved ASP problem representation incorporating the concept of reversible reactions. For another, optimization greatly benefits from the usage of unsatisfiable cores available in the ASP solver *unclasp* [13]. Applied to *Ectocarpus siliculosus*, only the combination of *unclasp* and *clasp* [14] allowed us to obtain a metabolic network able to produce all recoverable metabolites among the experimentally measured ones. Moreover, 70% of the identified reactions are supported by an homologous enzyme in *Ectocarpus siliculosus*, confirming the quality of the reconstructed network from a biological viewpoint.

In what follows, we assume some familiarity with ASP, its semantics as well as its basic language constructs. In particular, our encodings are written in the input language of *gringo* 3 [15]. Comprehensive treatments of ASP can be found in [12, 16].

2 Metabolic Network Completion

Metabolism is the sum of all chemical reactions occurring within an organism. As the products of a reaction may be reused as reactants, reactions can be chained to complex chemical pathways. Such complex pathways are described by a metabolic network.

A *metabolic network* is commonly represented as a directed bipartite graph $G = (R \cup M, E)$, where R and M are sets of nodes standing for *reactions* and *metabolites*, respectively. When $(m, r) \in E$ (or $(r, m) \in E$) for $m \in M$ and $r \in R$, the metabolite m is called a *reactant* (or *product*) of reaction r. More formally, for any $r \in R$, define $reac(r) = \{m \in M \mid (m, r) \in E\}$ and $prod(r) = \{m \in M \mid (r, m) \in E\}$.

The biological concept of the synthetic capabilities of a metabolism can be expressed in terms of reachability. Given a metabolic network $(R \cup M, E)$ and a set $S \subseteq M$ of *seed* metabolites, a reaction $r \in R$ is *reachable* from S if all reactants in $reac(r)$ are reachable from S. Moreover, a metabolite $m \in M$ is *reachable* from S if $m \in S$ or if $m \in prod(r)$ for some reaction $r \in R$ that is reachable from S. The *scope* of S, written $\Sigma_{(R \cup M, E)}(S)$, is the closure of metabolites reachable from S.

Given a metabolic network $(R \cup M, E)$, two sets $S, T \subseteq M$ of seed and target metabolites, and a reference network $(R' \cup M', E')$, the *metabolic network completion problem* is to find a set $R'' \subseteq R' \setminus R$ of reactions such that $T \subseteq \Sigma_G(S)$, where

$$\begin{aligned} G &= ((R \cup R'') \cup (M \cup M''), E \cup E'') \,, \\ M'' &= \{m \in M' \mid r \in R'', m \in reac(r) \cup prod(r)\} \,, \text{ and} \\ E'' &= E' \cap ((M'' \times R'') \cup (R'' \times M'')) \,. \end{aligned}$$

We call R'' a *completion* of $(R \cup M, E)$ from $(R' \cup M', E')$ wrt (S, T).

For reconstructing *Ectocarpus siliculosus*, we are interested in *cardinality-minimal completions* as well as *necessary reactions* belonging to every completion. Therefore, we need to solve the following sub-tasks:

– **Problem 1**: Compute the minimum size (number of reactions) of a completion.

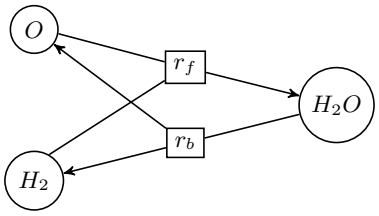

Fig. 1. Example of the first method on $H_2 + O \rightleftharpoons H_2O$.

- **Problem 2**: Enumerate all cardinality-minimal completions.
- **Problem 3**: Compute the intersection of all cardinality-minimal completions.

As shown in [17, 18], the reconstruction of metabolic networks and related problems are NP-hard.[9] Problem variants (of higher computational complexity) rely on subset- rather than cardinality-minimal completions. Further refinements may also optimize on the distance between seeds and targets or minimize forbidden side products.

3 Reversible Reactions

Chemical reactions are in essence reversible. However, taking the metabolic context into account (i.e. reactants and products) leads to considering some of them as irreversible in view of energetic cost [19]. In the following, we describe two alternative methods to capture reversible and irreversible reactions.

The first method represents a reversible reaction by two inverse reactions that are separate nodes within the network. For example, given the metabolites H_2, O, and H_2O and the reversible reaction $r = H_2 + O \rightleftharpoons H_2O$, we can construct the metabolic network $(\{H_2, O, H_2O, r_f, r_b\}, \{(H_2, r_f), (O, r_f), (r_f, H_2O), (H_2O, r_b), (r_b, H_2), (r_b, O)\})$, as illustrated in Figure 1. This method allows us to apply the framework presented in [11]. Unfortunately, it also roughly doubles the number of reactions that must be considered when looking for completions.

For an alternative method, let us represent a metabolic network as a graph $G = (R_{rev} \cup R_{irrev} \cup M, E)$, where R_{rev}, R_{irrev}, and M are sets of nodes standing for *reversible reactions*, *irreversible reactions*, and metabolites, respectively. The difference to our previous approach is that we distinguish between nodes for reversible and irreversible reactions. For any reaction $r \in R_{rev} \cup R_{irrev}$, the edges in E describe exactly one direction, that is, $(m, r) \in E$ (or $(r, m) \in E$) expresses that the metabolite $m \in M$ is a reactant (or product) of r. Taking r to be reversible, the network $(\{H_2, O, H_2O, r\}, \{(H_2, r), (O, r), (r, H_2O)\})$ thus captures both reactions displayed in Figure 1.

Given a metabolic network $(R_{rev} \cup R_{irrev} \cup M, E)$ and a set $S \subseteq M$ of seed metabolites, a reaction $r \in R_{rev} \cup R_{irrev}$ is reachable from S if all reactants in $reac(r)$ are reachable from S; when $r \in R_{rev}$ is reversible, r is also *reachable* from S if all products in $prod(r)$ are reachable from S. This reflects that, depending on the direction in which a reversible reaction is applied, the roles of reactants and products may be

[9] That is, the underlying decision problems are NP-hard.

interchanged. Moreover, a metabolite $m \in M$ is *reachable* from S if $m \in S$ or if $m \in reac(r) \cup prod(r)$ for some reaction $r \in R_{rev} \cup R_{irrev}$ that is reachable from S. As in the previous section, the scope of S, written $\Sigma_{(R_{rev} \cup R_{irrev} \cup M, E)}(S)$, is the closure of metabolites reachable from S.

Using this alternative representation, the metabolic network completion problem for a network $(R_{rev} \cup R_{irrev} \cup M, E)$, two sets $S, T \subseteq M$ of seed and target metabolites, and a reference network $(R'_{rev} \cup R'_{irrev} \cup M', E')$ is to find a set $R'' \subseteq (R'_{rev} \cup R'_{irrev}) \setminus (R_{rev} \cup R_{irrev})$ of reactions such that $T \subseteq \Sigma_G(S)$, where

$$\begin{aligned} G &= ((R_{rev} \cup R_{irrev} \cup R'') \cup (M \cup M''), E \cup E'') \ , \\ M'' &= \{m \in M' \mid r \in R'', m \in reac(r) \cup prod(r)\} \ , \text{ and} \\ E'' &= E' \cap ((M'' \times R'') \cup (R'' \times M'')) \ . \end{aligned}$$

We call R'' a *completion* of $(R_{rev} \cup R_{irrev} \cup M, E)$ from $(R'_{rev} \cup R'_{irrev} \cup M', E')$ wrt (S, T).

Our ASP implementation addresses the alternative representation of reversible reactions by additional facts and rules in comparison to the seminal encoding [11]. In particular, an instance of the network completion problem now contains additional facts $reversible(r)$ for reactions $r \in R_{rev} \cup R'_{rev}$, and our new encoding utilizes reversibility information. For instance, the following rules define the scope of a network:

$$\begin{aligned} &scope(M) \leftarrow seed(M) \\ &scope(M) \leftarrow product(M, R), reaction(R), scope(M') : reactant(M', R) \\ &scope(M) \leftarrow reactant(M, R), reversible(R), scope(M') : product(M', R) \end{aligned}$$

These rules illustrate the changes in our logic program.[10] The first rule states that all metabolites given as seeds are available in an organism, and the second rule derives the products of a reaction whose reactants are available. Moreover, the third rule takes care of interchanged roles of reactants and products in a reversible reaction, where reactants can be derived from available products.

For instance, for implementing the example shown in Figure 1, one may consider the metabolites H_2 and O as seeds as well as H_2O as target. The ground programs obtained with the two alternative methods to represent reversible reactions are given in Listing 1 and 2. Both include similar rules to derive H_2 and O as available in the scope. However, the first program relies on two reactions, r_f and r_b, while the second program addresses the inverse reaction r_b via a rule for reversibility.

The outcomes of the program in Listing 1 are given by the sets $\{r_f\}$ and $\{r_f, r_b\}$ of reactions, the first of which is cardinality-minimal. This tells us that r_f is necessary to produce H_2O from H_2 and O. The unique outcome $\{r\}$ of the program in Listing 2 likewise yields the necessity of applying r, where the actual direction of r needed to produce H_2O from H_2 and O can be inferred easily.

4 Experiments

In order to successfully solve the three problems introduced above, we propose to divide the metabolic network completion into two phases. In the first phase, we compute

[10] The full encoding is available at `http://pypi.python.org/pypi/meneco`.

Listing 1. Ground logic program instance without reversibility.

```
seed(H2). seed(O). target(H2O).

{ reaction(rf) }.                    { reaction(rb) }.
reactant(H2,rf). reactant(O,rf).     reactant(H2O,rb).
product(H2O,rf).                     product(H2,rb). product(O,rb).

scope(H2) :- seed(H2).
scope(O) :- seed(O).

scope(H2O) :- product(H2O,rf), reaction(rf), scope(H2), scope(O).

scope(H2) :- product(H2,rb), reaction(rb), scope(H2O).

scope(O) :- product(O,rb), reaction(rb), scope(H2O).

:- target(H2O), not scope(H2O).

#minimize{ reaction(rf), reaction(rb) }.
```

Listing 2. Ground logic program instance with reversibility.

```
seed(H2). seed(O). target(H2O).

{ reaction(r) }. reversible(r).
reactant(H2,r). reactant(O,r).
product(H2O,r).

scope(H2) :- seed(H2).
scope(O) :- seed(O).

scope(H2O) :- product(H2O,r), reaction(r), scope(H2), scope(O).

scope(H2) :- reactant(H2,r), reversible(r), scope(H2O).

scope(O) :- reactant(O,r), reversible(r), scope(H2O).

:- target(H2O), not scope(H2O).

#minimize{ reaction(r) }.
```

the minimum size of a network completion (**Problem 1**). To this end, ASP provides powerful optimization techniques based on branch-and-bound algorithms. Albeit such techniques can be highly effective, our application pinpoints their current limitations. Hence, we take advantage of *unclasp* (version 0.1), whose usage of unsatisfiable cores

Table 1. Ranges of minimum size and number of cardinality-minimal completions for MetaCyc subsets. The time-outs of *clasp* are also reported with and without the reversibility encoding.

Number of reactions	5000	6000	7000	8000	9000	10000	Full
Minimum completion size	[6,14]	[7,22]	[7,29]	[9,29]	[16,47]	[33,50]	52
clasp time-outs							
with reversibility	0	0	1	3	9	10	10
without reversibility	0	0	0	2	8	10	10
Minimal completions	[4,32]	[6,324]	[6,1728]	[16,3456]	[80,1150]	[180,22800]	2600

is inspired by respective approaches to Maximum Satisfiability (MaxSAT) [20]. In the second phase, we rely on *clasp* (version 2.2.1) to enumerate all minimal completions (**Problem 2**) or to compute the intersection of all minimal completions (**Problem 3**). The experiments were run on a cluster of three machines equipped with 128 to 144 GB RAM and totaling 48 cores, clocked from 2.39 to 2.66 GHz.

4.1 Reconstruction of the Metabolic Network of *Ectocarpus siliculosus*

As a first experiment, we complete a draft metabolic network of the brown algae *Ectocarpus siliculosus* [21] with reactions from MetaCyc. The draft network, produced by merging expert annotations [22] with orthologs in *Arabidopsis thaliana* [23], contained 1210 reactions and 1454 metabolites. Moreover, we consider 44 metabolites as seeds, provided by biological experts, and 51 metabolites, which have been experimentally shown to be natural products of *Ectocarpus siliculosus*, as targets. We checked that the draft network can only produce 23 of the 51 experimentally established targets, which exhibits the insufficiency of the draft network to recover some of the main metabolic capabilities of the brown algae. This also shows that metabolic reconstruction via manual methods is not sufficiently detailed for an "exotic" species like *Ectocarpus siliculosus*.

Applying *unclasp* and *clasp* as described above, we could solve **Problem 1**, **2**, and **3** for the draft network. It turns out that at least 52 reactions from the MetaCyc database are required to produce 48 metabolites among the 51 experimentally established targets (**Problem 1**). We checked that the three remaining targets are not producible via reactions from MetaCyc. Moreover, enumeration led to 2600 cardinality-minimal completions (**Problem 2**), whose intersection consists of 45 reactions (**Problem 3**).

The union of all cardinality-minimal completions, 70 reactions, was then added to the draft network to reconstruct the first metabolic network of *Ectocarpus siliculosus*. A comparison of the resulting network, containing 1280 reactions and 1507 metabolites, to sequence information showed that 70% of the reactions are relevant in the brown algae. This suggests that reconstruction by means of ASP is biologically meaningful.

4.2 Study of Scalability

Given that the size of the reaction database constitutes a primary factor regarding the performance of metabolic network reconstruction, we further investigated the scalability of our approach and the benefit of introducing the new model for reversible reactions.

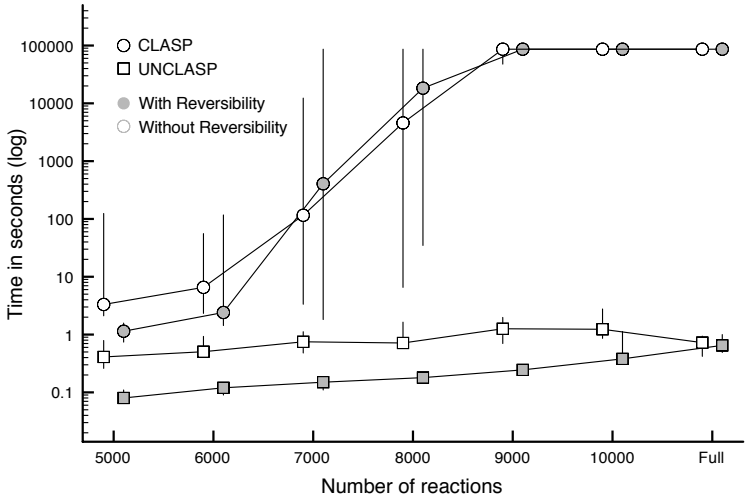

Fig. 2. Runtimes of *clasp* and *unclasp* for computing the minimum size of a completion (Problem 1). The circles and squares provide the median runtimes of *clasp* and *unclasp*, respectively. In addition, minimum and maximum runtimes are reported as vertical lines.

We thus applied our method to the completion of *Ectocarpus siliculosus* relying on databases of different sizes. We created 10 different subsets of MetaCyc, each containing 10000 randomly selected reactions. Starting from them, smaller subsets of size 9000, 8000, 7000, 6000, and 5000 were created by randomly and successively removing reactions, yielding 10 distinguished benchmarks for each size. Each subset includes the same proportion of reversible reactions as the full MetaCyc database ($\approx 42\%$).

Table 1 summarizes the minimum network completion sizes, the time-outs of *clasp* upon computing (or proving, respectively) minimum sizes, and the numbers of cardinality-minimal completions for MetaCyc subsets of different sizes. Notably, the minimum sizes of completions recovering producible targets remain relatively small (≤ 50). The small sizes and apparent locality of network completions promote *unclasp*, which turns out to be highly effective upon optimization in the first phase. As the current functionalities of *unclasp* do not include enumeration or intersection computation, the respective experiments are limited to *clasp* in the second phase.

Solving Problem 1. In order to determine the minimum sizes of network completions, we ran *unclasp* in its default configuration as well as *clasp* with the options `--time-limit=86400 --restart-on-model --reset-restarts --local-restarts --opt-heu --save-progress`. The latter configure *clasp*'s sign heuristic to falsify literals subject to minimization and also foster restarts to avoid getting stuck in local minima. However, the runtimes plotted in Figure 2 stay around one second with *unclasp* but grow exponentially with *clasp*. Moreover, the explicit representation of reversible reactions speeds up *unclasp* by factors from 2 to 11, while it leads to more time-outs with *clasp* (cf. Table 1).

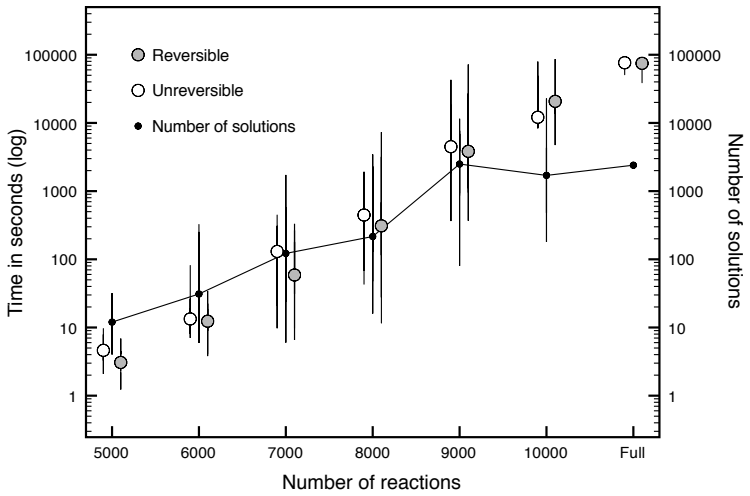

Fig. 3. Runtimes of *clasp* for enumerating all cardinality-minimal completions (Problem 2). The gray and white circles provide the median runtimes of *clasp*; dots indicate the median number of cardinality-minimal completions. Minimum and maximum values are reported as vertical lines.

Solving Problem 2. For enumerating cardinality-minimal completions, we ran *clasp* with the options `--time-limit=86400 --configuration=handy --opt-all=`*optimum*, where the "`handy`" configuration is geared towards large problems. The results plotted in Figure 3 show that the runtimes of *clasp* and the numbers of solutions tend to grow exponentially with the size of the reaction database. The number of solutions, however, reaches a plateau from 9000 reactions on, thus exhibiting a correlation with the minimum completion sizes given in Table 1.

Solving Problem 3. Adding the option `--enum-mode=cautious` switches *clasp* from enumeration to computing the intersection of cardinality-minimal completions. The runtimes plotted in Figure 4 still parallel those for enumeration. Unlike this, the intersection size grows much more moderately than the number of cardinality-minimal completions, so that future advancements of ASP solving technology may shrink the efforts of computing consequences below those of enumeration.

5 Conclusions

As a first conclusion, we note that *unclasp* enables the calculation of minimum completion size from an unabridged reaction database, which is necessary to accomplish the metabolic reconstruction of an "exotic" organism like *Ectocarpus siliculosus*. While *clasp* cannot solve this problem for the full MetaCyc database (in allotted time), *unclasp* completes the same task in a few seconds. Moreover, Figure 2 shows that *unclasp* remains almost unaffected by database growth. In fact, the usage of unsatisfiable cores allows for exploiting local problem structure to quickly converge to an optimal solu-

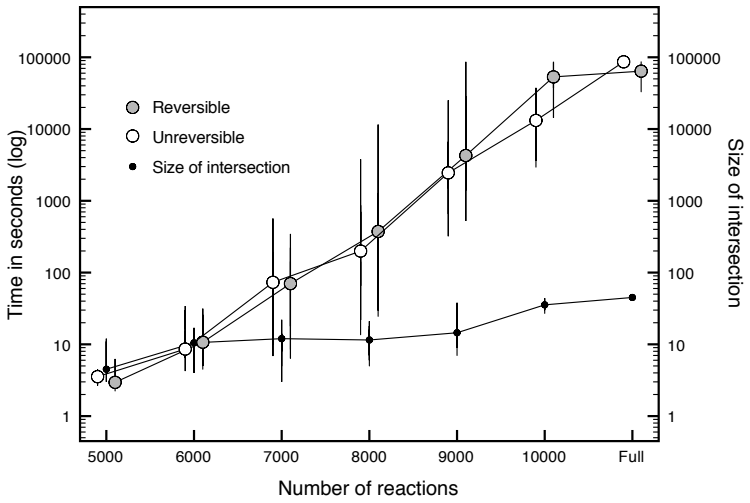

Fig. 4. Runtimes of *clasp* for computing the intersection of all cardinality-minimal completions (Problem 3). The gray and white circles provide the median runtimes of *clasp*; dots indicate the median intersection size. Minimum and maximum values are reported as vertical lines.

tion. Therefore, it appears that *unclasp* is especially well-suited to solve problems with plenty abducibles ($>$10000 reactions) but rather small optima (about 50 reactions).

As a second conclusion, integrating the reversibility concept into our ASP encoding reduces the runtime of *unclasp* by up to one order of magnitude. Somewhat surprisingly, *clasp* cannot benefit from the improved ASP encoding, even though reversibility reduces the number of candidate reactions from MetaCyc by about one third.

As a third conclusion, only the combination of *unclasp* and *clasp* allowed us to reconstruct the metabolic network of *Ectocarpus siliculosus* because the current functionalities of *unclasp* do not include enumeration or intersection computation. Fortunately, these two tasks can be accomplished by *clasp* when the minimum completion size is known. While enumeration enables an exhaustive exploration of (cardinality-minimal) completions, their intersection yields necessary reactions needed to produce target metabolites. Such information is crucial for the biological post-validation of a metabolic network without manual curation.

In summary, the combination of ASP modeling and solving capacities enabled the successful automatic reconstruction of the first metabolic network of a macroalgae. However, Figure 3 and 4 also indicate that the capabilities of *clasp* to compute all cardinality-minimal completions or their intersection almost hit the limits in view of the current size of the MetaCyc database. Anticipating its future extension, reconstruction tasks will be difficult to address without further advances in ASP solving. To this end, the incorporation of domain knowledge and heuristics to guide the solving process appear to be promising. As a direction for future work, we aim at the development of dedicated heuristics and their employment in the recent ASP solver *hclasp* [24].

Acknowledgments. This work was supported by ANR Biotempo (ANR-10-BLANC-0218), IDEALG (ANR-10-BTBR-04), and DFG (SCHA 550/10-1).

References

1. Barabási, A., Oltvai, Z.: Network biology: Understanding the cell's functional organization. Nature Reviews Genetics **5**(2) (2004) 101–113
2. Joyce, A., Palsson, B.: The model organism as a system: Integrating 'omics' data sets. Nature Reviews Molecular Cell Biology **7**(3) (2006) 198–210
3. Yamada, T., Bork, P.: Evolution of biomolecular networks: Lessons from metabolic and protein interactions. Nature Reviews Molecular Cell Biology **10**(11) (2009) 791–803
4. Orth, J., Conrad, T., Na, J., Lerman, J., Nam, H., Feist, A., Palsson, B.: A comprehensive genome-scale reconstruction of Escherichia coli metabolism. Molecular Systems Biology **7** (2011) Article 535
5. de Oliveira Dal'Molin, C., Quek, L., Palfreyman, R., Brumbley, S., Nielsen, L.: AraGEM, a genome-scale reconstruction of the primary metabolic network in Arabidopsis. Plant Physiology **152**(2) (2009) 579–589
6. Zengler, K., Palsson, B.: A road map for the development of community systems (CoSy) biology. Nature Reviews Microbiology **10**(5) (2012) 366–372
7. Swainston, N., Smallbone, K., Mendes, P., Kell, D., Paton, N.: The subliminal toolbox: automating steps in the reconstruction of metabolic networks. Journal of Integrative Bioinformatics **8** (2011) Article 186
8. Handorf, T., Ebenhöh, O., Heinrich, R.: Expanding metabolic networks: Scopes of compounds, robustness, and evolution. Journal of Molecular Evolution **61**(4) (2005) 498–512
9. Satish Kumar, V., Dasika, M., Maranas, C.: Optimization based automated curation of metabolic reconstructions. BMC Bioinformatics **8** (2007) Article 212
10. Christian, N., May, P., Kempa, S., Handorf, T., Ebenhöh, O.: An integrative approach towards completing genome-scale metabolic networks. Molecular BioSystems **5**(12) (2009) 1889–1903
11. Schaub, T., Thiele, S.: Metabolic network expansion with ASP. In Hill, P., Warren, D., eds.: Proceedings of the Twenty-fifth International Conference on Logic Programming (ICLP'09). Volume 5649 of Lecture Notes in Computer Science, Springer-Verlag (2009) 312–326
12. Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press (2003)
13. Andres, B., Kaufmann, B., Matheis, O., Schaub, T.: Unsatisfiability-based optimization in clasp. In Dovier, A., Santos Costa, V., eds.: Technical Communications of the Twenty-eighth International Conference on Logic Programming (ICLP'12). Volume 17 of Leibniz International Proceedings in Informatics, Dagstuhl Publishing (2012) 212–221
14. Gebser, M., Kaufmann, B., Schaub, T.: Conflict-driven answer set solving: From theory to practice. Artificial Intelligence **187-188** (2012) 52–89
15. Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., Thiele, S.: A user's guide to gringo, clasp, clingo, and iclingo. `http://potassco.sourceforge.net`
16. Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Answer Set Solving in Practice. Morgan and Claypool Publishers (2012)
17. Nikoloski, Z., Grimbs, S., May, P., Selbig, J.: Metabolic networks are NP-hard to reconstruct. Journal of Theoretical Biology **254**(4) (2008) 807–816
18. Nikoloski, Z., Grimbs, S., Selbig, J., Ebenhöh, O.: Hardness and approximability of the inverse scope problem. In Crandall, K., Lagergren, J., eds.: Proceedings of the Eighth International Workshop on Algorithms in Bioinformatics (WABI'08). Volume 5251 of Lecture Notes in Computer Science, Springer-Verlag (2008) 99–112

19. Beard, D., Liang, S., Qian, H.: Energy balance for analysis of complex metabolic networks. Biophysical Journal **83**(1) (2002) 79–86
20. Li, C., Manyà, F.: MaxSAT. In Biere, A., Heule, M., van Maaren, H., Walsh, T., eds.: Handbook of Satisfiability. Volume 185 of Frontiers in Artificial Intelligence and Applications, IOS Press (2009) 613–631
21. Tonon, T., Eveillard, D., Prigent, S., Bourdon, J., Potin, P., Boyen, C., Siegel, A.: Toward systems biology in brown algae to explore acclimation and adaptation to the shore environment. Omics: A Journal of Integrative Biology **15**(12) (2011) 883–892
22. Karp, P., Paley, S., Romero, P.: The Pathway Tools software. Bioinformatics **18**(Suppl 1) (2002) S225–S232
23. Loira, N., Dulermo, T., Nicaud, J., Sherman, D.: A genome-scale metabolic model of the lipid-accumulating yeast Yarrowia lipolytica. BMC Systems Biology **6** (2012) Article 35
24. Gebser, M., Kaufmann, B., Otero, R., Romero, J., Schaub, T., Wanko, P.: Domain-specific heuristics in answer set programming. In desJardins, M., Littman, M., eds.: Proceedings of the Twenty-Seventh National Conference on Artificial Intelligence (AAAI'13). AAAI Press (2013) To appear