Extension of Partial Atom-To-Atom Maps: Uniqueness and Algorithms

Marcos E. González Laffitte
Center for Scalable Data Analytics and Artificial Intelligence Dresden-Leipzig (ScaDS.AI), Leipzig, Germany
Bioinformatics Group, Department of Computer Science, and Interdisciplinary Center of Bioinformatics, Leipzig University, Germany

Tieu-Long Phan
Bioinformatics Group, Department of Computer Science, Leipzig University, Germany
Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark

Peter F. Stadler
Bioinformatics Group, Department of Computer Science, and Interdisciplinary Center of Bioinformatics, University of Leipzig, Germany
Max-Planck-Institute for Mathematics in the Sciences, Leipzig, Germany
Inst. f. Theoretical Chemistry, University of Vienna, Austria
Facultad de Ciencias, Universidad National de Colombia, Bogotá, Colombia
Santa Fe Institute, NM, USA

Abstract

Chemical reaction databases typically report the molecular structures of reactant and product compounds, as well as their stoichiometry, but lack information, in particular, on the correspondence of reactant and product atoms. These *atom-to-atom maps* (AAM), however, are crucial for applications including chemical synthesis planning in organic chemistry and the analysis of isotope labeling experiments in modern metabolomics. AAMs therefore need to be reconstructed computationally. This situation is aggravated, furthermore, by the fact that chemically correct AAMs are, fundamentally, determined by quantum-mechanical phenomena and thus cannot be reliably computed by solving graph-theoretical optimization problems defined by the reactant and product structures. A viable solution for this problem is to shift the focus into first identifying a *partial* AAM containing the reaction center, i.e., covering the atoms incident with all bonds that change during a reaction. This then leads to the problem of extending the partial map to the full reaction. The AAM of a reaction is faithfully represented by the *Imaginary Transition State* (ITS) graph, providing a convenient graph-theoretic framework to address the questions of when and how a partial AAM can be extended. We show that an unique extension exists whenever, and only if, these partial AAMs cover the reaction center. In this case their extension can be computed by solving a constrained graph-isomorphism search between specific subgraphs of ITS graphs. We close by benchmarking different tools for this task.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Discrete optimization

Keywords and phrases atom-to-atom maps, imaginary transition state (ITS) graphs, condensed graph of the reaction (CGR), chemical reaction mechanisms, molecular graphs, metabolic networks, chemical synthesis planning, constrained graph isomorphism

Digital Object Identifier 10.4230/LIPIcs.WABI.2025.12

Supplementary Material
Software (Source Code): `https://github.com/MarcosLaffitte/GranMapache` [24]
Software (Source Code): `https://github.com/MarcosLaffitte/GranMapache/tree/main/examples/Stable_Extensions`
Software (Source Code): `https://github.com/TieuLongPhan/PartialAAMs`

25th International Conference on Algorithms for Bioinformatics (WABI 2025).
Editors: Broňa Brejová and Rob Patro; Article No. 12; pp. 12:1–12:26
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Funding *Marcos E. González Laffitte*: Acknowledges financial support from the Federal Ministry of Education and Research of Germany (BMBF) and the Sächsische Staatsministerium für Wissenschaft Kultur und Tourismus in the program Center of Excellence for AI-research "Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig", project identification number: ScaDS.AI.
Tieu-Long Phan: European Unions Horizon Europe Doctoral Network programme under the Marie-Skłodowska-Curie grant agreement No 101072930 (TACsy – Training Alliance for Computational systems chemistry). Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.
Peter F. Stadler: Received support from the Federal Ministry of Education and Research of Germany (BMBF) through DAAD project 57616814 (SECAI, School of Embedded Composite AI), and from the *Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig*, funded jointly by the Federal Ministry of Education and Research of Germany (BMBF) and the Sächsische Staatsministerium für Wissenschaft, Kultur und Tourismus in the programme Centers of Excellence for AI-research, project identification number SCADS24B.

Acknowledgements We want to thank Annachiara Korchmaros for the interesting graph-theoretical questions and conversations on the topic.

1 Introduction and Motivation

A large part of chemical knowledge is encoded in chemical reactions and formalized as transformations of structural formulae, i.e., of labeled graphs that explicitly represent atoms as vertices and chemical bonds as edges. Large-scale data bases, such as `Reaxys`® [16] or the United States Patent and Trademark Office (`USPTO`) database [30], collect this knowledge in the form of some model for a reaction $G \longrightarrow H$, where G and H are the disjoint unions of the structural formulae of reactants and products, respectively. The same representation is used for metabolic reactions in the KEGG and EcoCyc databases.

For a wide variety of practical applications, ranging from chemical synthesis planning to the analysis of isotope labeling experiments in metabolic networks, the knowledge of G and H is insufficient, however. In addition, the exact correspondence of reactant and product atoms is required. This *atom-to-atom map* (AAM) is usually represented as a bijection $\alpha : V(G) \to V(H)$ between vertex sets of the reactants-graph G and the products-graph H modeling $G \longrightarrow H$. The AAM, moreover, preserves atom types and determines the bonds that are formed, broken and conserved in the course of the reaction [25]. Thus, AAMs can also be understood as a summary of the mechanism of a reaction, at least at the level of abstraction defined specified by structure formulas.

The databases introduced above, for example, typically do not provide AAMs together with the reactions, which therefore need to be constructed by computational means. This has turned out to be a non-trivial problem that still remains an area of active research. The main difficulty is that the *chemically correct* AAM is determined by the ground-truth mechanism of the reaction (or reactions in the case of multi-step transformations which are of particular relevance in enzyme-catalyzed biochemical reactions), which is inherently a quantum-mechanical phenomenon whose outcome is, at best, approximated by heuristic rules such as the Principle of Minimal Chemical Distance [23], geometric rules such as the Principle of Least Nuclear Motion [21], or maximum common subgraph approaches [10]. More recently, machine-learning methods have been devised to complement the shortcomings of the combinatorial methods, see [28] for are recent comparative benchmarking effort.

An alternative for seeking to infer an AAM in a single step, is to divide this task into three potentially easier subproblems: (a) determine the most likely *type* of the reaction, (b) identify the reaction center(s), i.e., the atoms incident with all bonds that change, and (c) construct the AAMs using the results of (a) and (b) as constraints.

The rationale for this approach is that, fundamentally, reactions are not arbitrary changes of bonds. On the contrary, chemical reactions usually follow specific localized patterns of bond changes and are restricted to particular classes of reactants. In the chemistry literature, such reaction types are often referred to as "name(d) reactions" such as Grignard reaction, Claisen condensation or Friedel–Crafts acylation [27].

Reaction types can be specified, moreover, by *reaction templates* describing the parts of the involved molecules that are actually affected and/or necessary for the reaction to take place. A reaction template thus is a pair of pattern graphs, one, L in the reactants-graph G and the other, R, in the products-graph H, together with a bijection $\xi : V(L) \to V(R)$ that establishes the AAM at the level of the *patterns*. A reaction template $L \longrightarrow R$ is said to *explain* the reaction $G \longrightarrow H$ if (i) L and R appear as (are isomorphic to) subgraphs of G and H, respectively, (ii) the bijection ξ can be extended to an AAM for G and H, and (iii) all bonds that change during $G \longrightarrow H$ are covered by $L \longrightarrow R$. Reaction templates therefore contain and typically extend beyond the reaction center. Formally, they can be interpreted as a special case of double pushout graph rewriting rules, see e.g. [8, 2, 3]. Preservation of mass and atom types, i.e., of the number of vertices and their labels, makes it possible to express the necessary theory in terms of graph isomorphisms and induced subgraphs. We therefore forego a description of the category-theoretic formalism of graph transformations for the purposes of this contribution.

▶ **Definition 1.** *Consider two reactions $G \longrightarrow H$ and $L \longrightarrow R$, where the latter is endowed with an AAM $\xi : V(L) \to V(R)$. Then (L, R, ξ) is a* pattern *for $G \longrightarrow H$, if there are subgraph isomorphisms $\mu : V(L) \to V(G)$ and $\nu : V(R) \to V(H)$ of the patterns into reactants and products, such that the induced* partial atom-to-atom map *$\pi : \mu(L) \to \nu(R)$ given by $\pi(x) = \nu(\xi(\mu^{-1}(x)))$ for all $x \in \mu(L) \subseteq V(G)$, can be extended to an AAM $\alpha : V(G) \to V(H)$ that coincides with π on $\mu(L)$, i.e., $\alpha(x) = \pi(x)$ for all $x \in \mu(L)$.*

Throughout this contribution we will assume that a reaction $G \longrightarrow H$ and a partial AAM $\pi : U \to W$ with $U \subseteq V(G)$ and $W \subseteq V(H)$ are given. In practice, such partial AAMs can be produced, e.g., by learning-based reaction mapping tools [6, 35]. Another source of partial AAM data are the RCLASS data provided by the KEGG database, although in this case extensive processing is required to obtain the partial AAMs explicitly [5]. We therefore do not need to concern ourselves with pattern graphs L and R and their embeddings μ and ν into the graphs G and H. The mathematical structure of full and partial AAMs has been studied in some details in [25, 26]. A key observation is that AAMs over balanced reactions can be represented equivalently by means of *Imaginary Transition State (ITS)* graph and its subgraphs. ITS graphs were introduced by Shinsaku Fujita [13] and Wilcox and Levinson [37] for storage and processing of reactions in chemical databases, and later utilized under the name Condensed Graph of the Reaction (CGR) in machine learning applications [22]. The one-to-one correspondence between AAMs and ITS graphs is, therefore, the basis for the graph-theoretical approach to AAMs that we pursue in this contribution.

Applications to Bioinformatics of our algorithms, therefore, are strongly dependent on their applications to Organic Chemistry. Figure 1, for example, exhibits the biochemical exchange mechanism, and the corresponding ITS graph, that allows *E. coli* to recycle and route nitrogen in an efficient way [33]. Studying domain-specific mechanisms such as this one is the focus of separate intended future contributions. In order to elucidate the mechanisms of such reactions, nonetheless, one requires in the first place to address the problem of comparing and extending the associated AAMs in a mathematically-correct fashion. Here we will focus on answering two **research questions**: **(1)** when does such extensions of a partial AAM exist? and **(2)** how can such an extension be computed efficiently?

In the following section we provide the necessary theoretical framework, establishing the equivalence of AAMs and ITS graphs and introducing the *remainder graphs*. It is shown, moreover, that certain isomorphisms of these remainder graphs are a sufficient condition for the construction of equivalent of AAMs. We then consider reaction centers and their associated partial AAMs. The main result of this section establishes that *good* partial AAMs are characterized by the existence of a unique *stable extension*, i.e., extensions preserving the reaction mechanism, that are in turn isomorphisms between remainder graphs, providing, therefore, the basis for our algorithms for the completion of good partial AAMs.

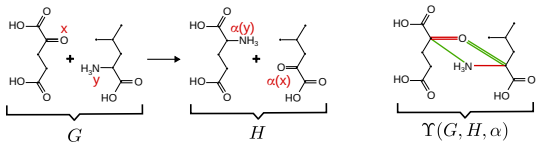

Figure 1 The reaction mechanism imposed by the AAM α over $G \longrightarrow H$ is characterized by the ITS graph $\Upsilon(G, H, \alpha)$, whose labels preserve all the information about the changing and conserved bonds.

2 Reactions, AAMs and ITS graphs

2.1 Basic Notation

Molecules are modeled as connected, labeled simple graphs with vertices representing atoms and edges corresponding to bonds. We write $V(G)$ and $E(G)$ for the vertex and edge set of a graph G. Atom types and bond types are specified by labeling functions $a_G : V(G) \to L_V$ and $b_G : E(G) \to L_E$, respectively, for non-empty and disjoint label sets L_V and L_E. We reserve a special symbol $\otimes \notin L_V \cup L_E$ for our construction (Definition 6 below). Charges can be associated as labels to loops. We will not, however, consider this issue explicitly here, hence examples will be simple graphs throughout. For standard terminology on Graph Theory, e.g. adjacency and connectedness, we refer to [19]. Two labeled graphs $G = (V(G), E(G), a_G, b_G)$ and $H = (V(H), E(H), a_H, b_H)$ are *isomorphic* if there is a bijection $\varphi : V(G) \to V(H)$ that preserves adjacency as well as vertex and edge labels. A map $\varphi : V(G) \to V(H)$ thus is an isomorphism if: (i) it is bijective, (ii) $\varphi(x)\varphi(y) \in E(H)$ if and only if $xy \in E(G)$, (iii) $a_H(\varphi(x)) = a_G(x)$ for all $x \in V(G)$ and (iv) $b_H(\varphi(x)\varphi(y)) = b_G(xy)$ for all $xy \in E(G)$. We write $\mathrm{Iso}(G, H)$ for the set of all isomorphisms from G to H, and $G \simeq H$ whenever G and H are isomorphic. Isomorphisms of G to itself are called automorphisms and form an algebraic group denoted $\mathrm{Aut}(G)$ when endowed the usual composition of functions. We express the composition of functions $\alpha : X \to Y$ and $\beta : Y \to Z$ as $\beta\alpha(x) = \beta \circ \alpha(x) := \beta(\alpha(x))$.

Reactions, denoted as $G \longrightarrow H$, are pairs of graphs whose connected components represent the reactant and product molecules, respectively. Note that both G and H may contain multiple copies of isomorphic connected components depending on the stoichiometry of the the reaction. A reaction is *balanced* if there is an AAM for it, that is, there exists a bijection $\alpha : V(G) \to V(H)$ preserving atom-labels, i.e., $a_H(\alpha(x)) = a_G(x)$ for all $x \in V(G)$. Equivalently, $G \longrightarrow H$ is balanced if for each atom type $c \in L_V$ we have $|a_G^{-1}(c)| = |a_H^{-1}(c)|$, i.e., reactants and products contain the same number of atoms of each type. Consider an AAM α for a balanced reaction $G \longrightarrow H$ and $xy \in E(G)$. We say that xy is a *reaction edge* of G induced by α, if either $\alpha(x)\alpha(y) \notin E(H)$, or if $b_H(\alpha(x)\alpha(y)) \neq b_G(xy)$ whenever $\alpha(x)\alpha(y) \in E(H)$. Equally we say that $uv \in E(H)$ is a reaction edge of H induced by α if $\alpha^{-1}(u)\alpha^{-1}(v) \notin E(G)$, or $b_G(\alpha^{-1}(u)\alpha^{-1}(v)) \neq b_H(uv)$ provided $\alpha^{-1}(u)\alpha^{-1}(v) \in E(G)$. A

vertex $x \in V(G)$ is a *reacting* vertex (for α) if x is incident with a reaction edge in G or $\alpha(x)$ is incident with a reaction edge in H. Analogously, $x \in V(H)$ is a reacting vertex (for α) if x is incident with a reaction edge in H or $\alpha^{-1}(x)$ is incident with a reaction edge in G. In particular, therefore, both ends of a reaction edge are reacting vertices.

▶ **Definition 2.** *Let α be an AAM for the balanced reaction $G \longrightarrow H$. The* remainder graph *of G under α, denoted $\hat{G}_\alpha$, is the subgraph of G obtained by removing all reaction edges of G, and preserving all vertex labels and all remaining edge labels. The remainder graph $\hat{H}_\alpha$ of H under α is defined analogously.*

All non-reaction edges are by construction preserved between G and H. That is, if $xy \in E(G)$ is not a reaction edge then $\alpha(x)\alpha(y) \in E(H)$ and, *vice versa*, if $uv \in E(H)$ is not a reaction edge in H, then $\alpha(x)\alpha(y) \in E(H)$. As an immediate consequence we obtain,

▶ **Lemma 3** ([26], Lemma 1)**.** *Every AAM α for a balanced reaction $G \longrightarrow H$ is an isomorphism from the remainder graph $\hat{G}_\alpha$ to the remainder graph $\hat{H}_\alpha$.*

2.2 Equivalent AAMs and Isomorphic ITS graphs

The problem of determining whether two AAMs α and β for the same reaction are actually "the same" arises, for example, when comparing the results of different reaction mapping tools, because each tools uses its own graph representation. As a consequence, for a formal treatment of this problem it becomes necessary to compare $\alpha : V(G) \to V(H)$ and $\beta : V(G') \to V(H')$ given that G and G', as well as H and H', respectively, are isomorphic,

▶ **Definition 4** ([25])**.** *Let α and β be AAMs for, respectively, the balanced reactions $G \longrightarrow H$ and $G' \longrightarrow H'$ with $G' \simeq G$ and $H' \simeq H$. We say that α and β are* equivalent*, denoted as $\alpha \equiv \beta$, if there exist isomorphisms $\varphi \in \mathrm{Iso}(G, G')$ and $\psi \in \mathrm{Iso}(H, H')$ such that $\psi\alpha = \beta\varphi$.*

There is a close connection between the AAM α and isomorphisms on the remainder graphs that plays a central role, in particular, for Theorem 20, which is the main result supporting the methods of our contribution,

▶ **Proposition 5.** *Let α be an AAM for the balanced reaction $G \longrightarrow H$ and let β be an isomorphism from the remainder graph $\hat{G}_\alpha$ to the remainder graph $\hat{H}_\alpha$. If $\alpha(x) = \beta(x)$ holds for all reacting vertices x of α, then α and β are equivalent AAMs for $G \longrightarrow H$.*

Proof. Included in Appendix A. ◀

The information contained in a balanced reaction $G \longrightarrow H$ and its AAM α can be compiled into the Imaginary Transition State (ITS) graph of the reaction by identifying atoms that correspond to each other via α. The ITS graphs thus contains the edges of both the product and the educt graph. Both vertices and edges are associated with label pairs that derive from the labels in G and H. Formally this is,

▶ **Definition 6.** *Let $G \longrightarrow H$ be a balanced reaction with AAM $\alpha : V(G) \to V(H)$. An ITS graph T of (G, H, α) is a graph with vertex set $V(T)$, edge set $E(T)$, vertex-labeling function $a_T : V(T) \to L_V \times L_V$ and edge-labeling function $b_T : E(T) \to (L_E \cup \{\otimes\}) \times (L_E \cup \{\otimes\})$, obtained from G by means of a bijection $\tau : V(T) \to V(G)$ such that*

- **(i)** *$x, y \in V(T)$ we have $xy \in E(T)$ if and only if $\tau(x)\tau(y) \in E(G)$ or $\alpha(\tau(x))\alpha(\tau(y)) \in E(H)$;*
- **(ii)** *each vertex $x \in V(T)$ is labeled by the ordered pair $a_T(x) = (a_G(\tau(x)), a_H(\alpha(\tau(x))))$,*
- **(iii)** *each edge $xy \in E(T)$ is labeled by the ordered pair $b_T(xy)$ determined as follows:*

$$b_T(xy) = \begin{cases} (b_G(\tau(x)\tau(y)), b_H(\alpha(\tau(x))\alpha(\tau(y)))) & \text{if } \tau(x)\tau(y) \in E(G) \text{ and } \alpha(\tau(x))\alpha(\tau(y)) \in E(H) \\ (b_G(\tau(x)\tau(y)), \otimes) & \text{if } \tau(x)\tau(y) \in E(G) \text{ and } \alpha(\tau(x))\alpha(\tau(y)) \notin E(H) \\ (\otimes, b_H(\alpha(\tau(x))\alpha(\tau(y)))) & \text{if } \tau(x)\tau(y) \notin E(G) \text{ and } \alpha(\tau(x))\alpha(\tau(y)) \in E(H) \end{cases}$$

Figure 2 below showcases the construction of an ITS graph. We will use, moreover, the notation $a_T(x) = (a_T^1(x), a_T^2(y))$ and $b_T(xy) = (b_T^1(xy), b_T^2(xy))$ to address the two components of the vertex and edge labels of an arbitrary ITS graphs.

For every balanced reaction with AAM α there exists an ITS graph. The construction is not unique, however, because of the arbitrary bijection τ between the vertices of G and the vertices of T. We note that the vertices of the ITS graph also bijectively map to H since for every $y \in V(H)$ there is a unique $v = \alpha^{-1}(y) \in V(G)$ and $x = \tau^{-1}(v) \in V(T)$, from where $y = \alpha(v) = \alpha(\tau(x)) = \tau'(x) \in V(H)$. Now we confirm, nonetheless, our intuition that ITS graphs are unique up to isomorphism,

▶ **Lemma 7.** *Let $\Upsilon^{\circledast}(G, H, \alpha)$ be the (non-empty) collection of all ITS graphs built for a balanced reaction $G \longrightarrow H$ with AAM α and consider a graph $T \in \Upsilon^{\circledast}(G, H, \alpha)$. Then $T' \in \Upsilon^{\circledast}(G, H, \alpha)$ if and only if $T' \simeq T$.*

Proof. Included in Appendix A. ◀

Thus it suffices to consider an arbitrary representative ITS from $\Upsilon^{\circledast}(G, H, \alpha)$, which we will denote by $\Upsilon(G, H, \alpha)$. In earlier work, moreover, the ITS graph is defined by setting $V(T) = V(G)$ and using the identity map on G for τ, see e.g. [25]. We will denote this (unique) particular representative, that we call *canonical ITS graph*, by $\Upsilon^{\perp} := \Upsilon^{\perp}(G, H, \alpha)$, see Figure 2. The uniqueness of $\Upsilon^{\perp}$ follows immediately from Definition 6. To see this suppose that there exist two such ITS graphs T and T'. Substituting τ with the identity map on G we get from (i) that $xy \in E(T)$ if and only if $xy \in E(T')$ for all $x, y \in V(T) = V(G)$, and from (ii) and (iii) it follows, respectively, $a_T = a_{T'}$ and $b_T = b_{T'}$.

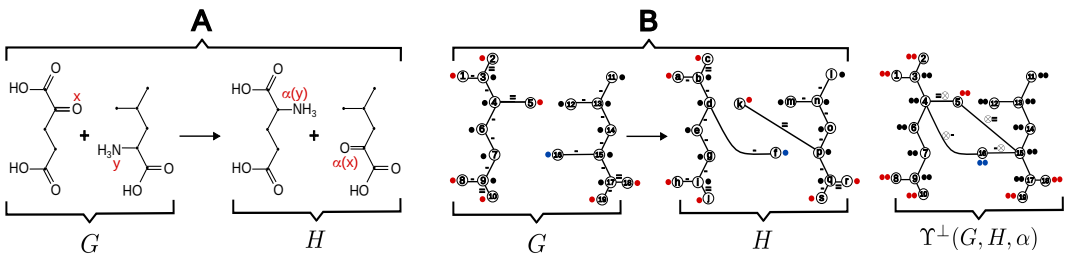

Figure 2 In **(A)** the reaction $G \longrightarrow H$ from Figure 1 is shown with usual chemical notation. In **(B)** the same reaction is given another representation resembling our formalism. Here, vertices are labeled by element types: "O" ●, "N" ● and "C" ● (hydrogen is omitted), while edges are labeled with "=" and "-" for double and single bonds, respectively. The canonical ITS graph $\Upsilon^{\perp}(G, H, \alpha)$ is shown on the right.

In [25], we proved the statement of the following Corollary for $\Upsilon^{\perp}$. With Lemma 7, on the other hand, we can now restate this result for arbitrary representatives,

▶ **Corollary 8** ([25], Cor. 1)**.** *Let $G \longrightarrow H$ and $G' \longrightarrow H'$ be balanced reactions with AAMs $\alpha : V(G) \to V(H)$ and $\beta : V(G') \to V(H')$ and assume $G' \simeq G$ and $H' \simeq H$. Then $\alpha \equiv \beta$ if and only if $\Upsilon(G, H, \alpha) \simeq \Upsilon(G', H', \beta)$ holds for any pair of ITS graphs for the two reactions.*

Corollary 8 shows that each equivalence class of AAMs for a reaction $G \longrightarrow H$ produces a unique equivalence class of isomorphic ITS graph representations, provided that the AAMs being compared are defined over reactions $G' \longrightarrow H'$ with isomorphic reactants $G' \simeq G$ and

products $H' \simeq H$. It is natural to ask whether there exist isomorphic ITS graphs $\Upsilon(G, H, \alpha)$ and $\Upsilon(G', H', \beta)$ for reactions with non-isomorphic graphs $G' \not\simeq G$ or $H' \not\simeq H$. The following result shows that this is indeed not possible,

▶ **Proposition 9.** *Let α and β be AAMs for, respectively, two balanced reactions $G \longrightarrow H$ and $G' \longrightarrow H'$, and let $\Upsilon(G, H, \alpha)$ and $\Upsilon(G', H', \beta)$ be their corresponding ITS representations. Then $\Upsilon(G, H, \alpha) \simeq \Upsilon(G', H', \beta)$ if and only if $G' \simeq G$, $H' \simeq H$, and $\alpha \equiv \beta$.*

Proof. Included in Appendix A. ◀

2.3 Reaction Centers

In Section 2.1 we defined reaction edges and reacting vertices for a reaction $G \longrightarrow H$ with a given AAM α. Since G, H, and α are uniquely defined up to graph isomorphisms and equivalence of AAMs by any representative ITS graph $\Upsilon(G, H, \alpha)$, these definitions carry over ITS graphs, i.e., they also have (a version of) reaction edges and reacting vertices. ITS graphs contain, moreover, an isomorphic copy of the remainder graphs $\hat{G}_\alpha$ and $\hat{H}_\alpha$,

▶ **Lemma 10.** *Let $\Upsilon := \Upsilon(G, H, \alpha)$ be an ITS representation of the balanced reaction $G \longrightarrow H$ with AAM α and let $\eta : V(G) \to V(\Upsilon)$ and $\eta' := \eta \circ \alpha^{-1} : V(H) \to V(\Upsilon)$ be the corresponding bijections that embed G and H into Υ, i.e., where $\eta := \tau^{-1}$ for $\tau : V(\Upsilon) \to V(G)$ as required by Definition 6. Then the following hold,*

- **(i)** *$xy \in E(G)$ is a reaction edge in G if and only if $b^1_\Upsilon(\eta(x)\eta(y)) \neq b^2_\Upsilon(\eta(x)\eta(y))$, and $x'y' \in E(H)$ is a reaction edge in H if and only if $b^1_\Upsilon(\eta'(x')\eta'(y')) \neq b^2_\Upsilon(\eta'(x')\eta'(y'))$.*
- **(ii)** *$xy \in E(\hat{G}_\alpha)$, and thus also $\alpha(x)\alpha(y) \in E(\hat{H}_\alpha)$, if and only if, $\eta(x)\eta(y) \in E(\Upsilon)$ and $b^1_\Upsilon(\eta(x)\eta(y)) = b^2_\Upsilon(\eta'(\alpha(x))\eta'(\alpha(y)))$.*

Proof. Included in Appendix A. ◀

We then refer to the edges $uv \in E(\Upsilon(G, H, \alpha))$ with unequal labels, i.e., with $b^1_\Upsilon(uv) \neq b^2_\Upsilon(uv)$, simply as reaction edges. Reacting vertices of (G, H, α) are thus represented in the ITS graph as the vertices incident with edges that have unequal labels. The reaction center of a reaction $G \longrightarrow H$ comprises all the bonds modified by the electron exchange occurring during the reaction. This notion appeared already in [14, 15, 32] and was used in [20] to classify reaction types. Its formal properties were studied in more detail in [26] and [11].

▶ **Definition 11.** *Let $\Upsilon(G, H, \alpha)$ be an ITS representation of the balanced reaction $G \longrightarrow H$ with AAM α. The* reaction center *is the subgraph $\Gamma(G, H, \alpha)$ of $\Upsilon(G, H, \alpha)$ comprising all the reaction edges and reacting vertices.*

It follows immediately from Prop. 9 that $\Gamma(G, H, \alpha) \simeq \Gamma(G', H', \beta)$ whenever $G \simeq G'$, $H \simeq H'$, and $\alpha \equiv \beta$. The converse statement, however, is not true in general. We show examples about this in [26], of graphs with isomorphic reaction centers, and (i) with $G \simeq G'$ but $H \not\simeq H'$ (Fig. 8 in [26]), and (ii) with $G \simeq G'$ and $H \simeq H'$ but $\alpha \not\equiv \beta$ (Fig. 10 in [26]). We will write, moreover, $\Gamma^\perp := \Gamma^\perp(G, H, \alpha)$ for the reaction center of the canonical representation $\Upsilon^\perp(G, H, \alpha)$ of the triple (G, H, α). It is worth mentioning that in [26] we considered the connectedness of these graphs. Though in general the connectedness of an ITS graph does not guarantee the connectedness of its reaction center or *vice versa* (see Figure 2 of [26]), we will, in practice, restrict ourselves to single-step reactions, which have connected reaction centers, i.e., a disconnected reaction center models two independent reactions. The following result, therefore, will also be of relevance for our algorithmic approach:

▶ **Lemma 12** ([26], Lemma 2). *Let α be an AAM for a balanced reaction $G \longrightarrow H$. If $\Upsilon(G, H, \alpha)$ is a connected graph, then every connected component of G and H contains at least one reacting vertex.*

2.4 Partial AAMs and their Partial ITS graphs

Theoretically speaking, both matches μ and ν of a reaction template, as in Definition 1, generally provide only a *partial AAM*. This is also the case, in practice, of computational mapping tools such as `LocalMapper` [6], which can focus on only determining a most plausible reaction center and necessary adjacent context.

▶ **Definition 13.** *Let $G \longrightarrow H$ be a reaction. A* partial AAM *is a bijection $\pi : U \to W$, for two subsets $U \subseteq V(G)$ and $W \subseteq V(H)$, which preserves vertex labels, i.e., such that $a_H(\pi(x)) = a_G(x)$ for all $x \in U$.*

Given a reaction $G \longrightarrow H$ and a partial AAM $\pi : U \to W$, it is immediate that the ITS graph $\Upsilon(G[U], H[W], \pi)$, together, in particular, with the canonical graphs $\Upsilon^{\perp}(G[U], H[W], \pi)$ and $\Gamma^{\perp}(G[U], H[W], \pi)$, are all well-defined, see Figure 3 for an example.

▶ **Definition 14.** *Let $G \longrightarrow H$ be a balanced reaction and let $\pi : U \to W$ with $U \subseteq V(G)$ and $W \subseteq V(H)$ be a partial AAM. Then an AAM $\alpha : V(G) \to V(H)$ is said to be an* extension *of π, or to* extend *π, if $\alpha(x) = \pi(x)$ for all $x \in U$.*

As noted in [26, Obs. 3], every partial AAM π for $G \longrightarrow H$ has an extension α. Moreover, it follows directly from the definition that the partial ITS graph $\Upsilon(G[U], H[W], \pi)$ representing π is isomorphic to the subgraph of the canonical ITS graph of (G, H, α) induced by U, i.e., $\Upsilon(G[U], H[W], \pi) \simeq \Upsilon^{\perp}(G, H, \alpha)[U]$. This isomorphism, furthermore, becomes the identity $\Upsilon^{\perp}(G[U], H[W], \pi) = \Upsilon^{\perp}(G, H, \alpha)[U]$ for the respective canonical ITS graphs, while for the canonical reaction centers we get $\Gamma^{\perp}(G[U], H[W], \pi) \subseteq \Gamma^{\perp}(G, H, \alpha)$. In general, therefore, $\Upsilon^{\perp}(G[U], H[W], \pi)$ will not contain all reaction edges. For many application scenarios, in particular the ones mentioned in the introduction, we do expect this to be the case. It is of interest, therefore, to determine whether a partial AAM, and thus its corresponding partial ITS graphs, already contains all reaction edges.

▶ **Definition 15.** *A partial AAM $\pi : U \to W$ with $U \subseteq V(G)$ and $W \subseteq V(H)$ for a balanced reaction $G \longrightarrow H$ is said to be a* good *partial AAM, if there is an extension $\alpha : V(G) \to V(H)$ of π such that $\Gamma^{\perp}(G, H, \alpha) = \Gamma^{\perp}(G[U], H[W], \pi)$.*

In other words, π is a good partial reaction map for a balanced reaction if and only if there is an extension α of π, such that the induced subgraph $\Upsilon^{\perp}(G, H, \alpha)[U]$ of the canonical ITS graph, contains all reaction edges of $\Upsilon^{\perp}(G, H, \alpha)$. In this case we call α a *stable extension* of π. Recall that $\hat{G}_\alpha$ and $\hat{H}_\alpha$ denote the remainder graphs obtained from G and H with respect to a full AAM α (see Definition 2). In order to better understand stable extensions we need to consider, additionally, the *remainder* graphs $\hat{G}_\pi$ and $\hat{H}_\pi$ induced by π, preserving vertex and edge labels of G and H, but obtained by removing from them, respectively, those reaction edges induced by $\pi : U \to W$ for $G[U]$ and $H[W]$, i.e., edges $xy \in E(G[U])$ such that $\pi(x)\pi(y) \notin E(H[W])$, or $\pi(x)\pi(y) \in E(H[W])$ but with $b_G(xy) \neq b_H(\pi(x)\pi(y))$, and edges $uv \in E(H[W])$ with $\pi^{-1}(u)\pi^{-1}(v) \notin E(G[U])$, or such that $\pi^{-1}(u)\pi^{-1}(v) \in E(G[U])$ but with $b_G(\pi^{-1}(u)\pi^{-1}(v)) \neq b_H(uv)$. Consider an arbitrary extension α of π. Clearly $V(\hat{G}_\pi) = V(G) = V(\hat{G}_\alpha)$ and $V(\hat{H}_\pi) = V(H) = V(\hat{H}_\alpha)$. Moreover, we have

▶ **Observation 16.** *Let α be an arbitrary extension of a partial AAM π for a balanced reaction $G \longrightarrow H$. Then, $E(\hat{G}_\alpha) \subseteq E(\hat{G}_\pi)$ and $E(\hat{H}_\alpha) \subseteq E(\hat{H}_\pi)$.*

Based on this simple observation, the following two results strengthen the connection between the remainder graphs induced by full and partial AAMs:

▶ **Lemma 17.** *Let $G \longrightarrow H$ be a balanced reaction and $\pi : U \to W$ with $U \subseteq V(G)$ and $W \subseteq V(H)$ be a partial AAM. Consider an extension α of π. Then, π is a good partial AAM with stable extension α, if and only if, $\hat{G}_\alpha = \hat{G}_\pi$ and $\hat{H}_\alpha = \hat{H}_\pi$.*

Proof. Included in Appendix A. ◀

▶ **Lemma 18.** *Let $G \longrightarrow H$ be a balanced reaction and $\pi : U \to W$ with $U \subseteq V(G)$ and $W \subseteq V(H)$ be a partial AAM. Consider an extension α of π. Then, α is an isomorphism from $\hat{G}_\pi$ to $\hat{H}_\pi$, if and only if, π is a good partial AAM with stable extension α.*

Proof. Included in Appendix A. ◀

The hypothesis of Lemma 18 requires the isomorphism between the remainder graphs $\hat{G}_\pi$ and $\hat{H}_\pi$ to be an extension of π. The existence of an arbitrary isomorphism between $\hat{G}_\pi$ and $\hat{H}_\pi$ indeed is not sufficient for the existence of a stable extension for π, see Figure 7 in Appendix B. With Lemmas 17 and 18, moreover, we recover the following results that we originally stated in [26, Prop. 1]:

▶ **Proposition 19.** *Let π be a partial AAM for a balanced reaction $G \longrightarrow H$ and α an extension of π. The following statements are equivalent,*

(i) *π is a good partial AAM and α is a stable extension of π,*
(ii) *$\hat{G}_\alpha = \hat{G}_\pi$ and $\hat{H}_\alpha = \hat{H}_\pi$,*
(iii) *$\alpha \in \mathrm{Iso}(\hat{G}_\pi, \hat{H}_\pi)$.*

2.5 Uniqueness of Stable Extensions

Good partial AAMs can have multiple non-identical stable extensions. Prop. 19, on the other hand, implies that all of them are isomorphisms between the remainder graphs $\hat{G}_\pi$ and $\hat{H}_\pi$.

▶ **Theorem 20.** *Let π be a good partial AAM for a balanced reaction $G \longrightarrow H$ and let α and β be two stable extensions of π. Then $\alpha \equiv \beta$.*

Proof. Included in Appendix A. ◀

Since all stable extensions of π are equivalent, their ITS representations are isomorphic,

▶ **Corollary 21.** *Let π be a good partial AAM for a balanced reaction $G \longrightarrow H$ and let α and β be two stable extensions of π. Then $\Upsilon(G, H, \alpha) \simeq \Upsilon(G, H, \beta)$.*

A good partial atom map for a balanced reaction, therefore, uniquely determines the ITS graph of the full AAM for the reaction, up to graph isomorphism.

3 Algorithms for Completing Good AAMs over Balanced Reactions

Conceptually, Definition 15 implies that the existence of a stable extension for a partial map π over a reaction $G \longrightarrow H$ ensures that π already provides all necessary information about the reaction mechanism of $G \longrightarrow H$. A bad partial AAM π, in contrast, fails to faithfully disclose the electron bookkeeping fundamental for understanding $G \longrightarrow H$. The characterization of stable extensions as isomorphisms of the reminder graphs $\hat{G}_\pi$ and $\hat{H}_\pi$ in Proposition 19, on the other hand, suggests to employ modified versions of algorithms for isomorphism search in order to test the existence of a stable extension, and thus the *goodness* of π, and to retrieve such extensions whenever they exist.

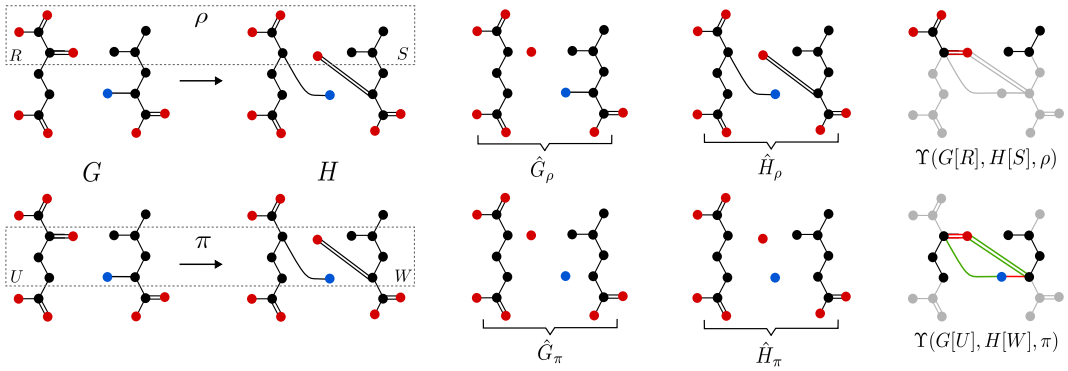

Figure 3 Two partial AAMs $\pi : U \to W$ and $\rho : R \to S$ over the reaction $G \longrightarrow H$ with full AAM α from Figure 1. The reaction centers induced by these maps are depicted by the edges drawn in red and green in the respective partial ITS graphs $\Upsilon(G[U], H[W], \pi)$ and $\Upsilon(G[R], H[S], \rho)$. Since π covers the reaction center induced by α, then π is a good partial AAM with stable extension α, and equivalently the remainder graphs $\hat{G}_\pi$ and $\hat{H}_\pi$ are isomorphic. On the other hand, ρ has no stable extensions, given that the corresponding remainder graphs $\hat{G}_\rho$ and $\hat{H}_\rho$ cannot be mapped to each other under an isomorphism.

From a theoretical point of view, this *stable extension problem* can be solved efficiently for chemical graphs. To see this, we note that the bounded valency of atoms implies that graphs that represent molecules must have bounded degree. As an immediate consequence ITS graphs also have bounded degree. A classical result establishes that isomorphisms of graphs with bounded degree can be computed in polynomial time [31], although algorithms following this approach are not competitive in practice. For recent progress we refer to [17]. No implementations of these polynomial-time algorithms seem to have become available, however. Hence we have to resort to general purpose algorithms for graph isomorphism. This situation conveys, furthermore, the relevance of the uniqueness of stable extensions of good partial AAMs in Theorem 20, given that these depend, therefore, on the existence of one and only one of such constrained isomorphisms and are unambiguously determined by it.

We address the stable extension problem through three algorithmic approaches: (1) an *anchored* isomorphism search, (2) a *relabeling-and-isomorphism* strategy, and (3) an ILP approach. In [26] we devised said ILP formulation based on Lemma 3, and here we recapitulate it in order to benchmark it against the other methods. As shown later, the graph-based isomorphism searches perform better than the ILP in practice. Nonetheless, Theorem 20 implies that our methodologies are all **mathematically equivalent**, i.e., they return the same stable extension (up to equivalence of AAMs), whenever it exists.

Anchored isomorphism tests. Conceptually these constitute a variant within the `VF2`-family of algorithms designed for the (sub-)graph isomorphism problem. A detailed formulation of the `VF2` algorithm can be found in [7]. This class of algorithms operates by progressively extending a candidate map for an isomorphism $\alpha : V(G) \to V(H)$ for arbitrary input graphs G and H. In each step, an ordered pair (x, y) called a *match*, with $x \in V(G)$ and $y \in V(H)$, is added to a collection $\mathcal{M}_\alpha$ portraying the vertex y, therefore, as the image of x under a prototype for α. Later, if (x, y) is added to $\mathcal{M}_\alpha$, further candidate pairs to extend the map are then selected, either from the sets of *unmatched* neighbors of x and y, called *terminal* sets, or arbitrarily selected from the remaining unmatched vertices. This last case is specifically designed to process disconnected graphs, since it allows the selection of vertices from distinct components once the terminal sets are exhausted.

Though such progressive procedure was originally designed as a recursive traversal, more recent variations construct the match set $\mathcal{M}_\alpha$ in an iterative manner [1].

The extension of $\mathcal{M}_\alpha$ depends on evaluating the *syntactic feasibility* i.e., a one-to-one correspondence of the edges connecting x and y respectively, to the vertices already included in $\mathcal{M}_\alpha$, as well as the *semantic feasibility.* The latter evaluates that x and y have the same vertex-labels, and that corresponding edges incident with them have compatible edge-labels. In this way, whenever the `VF2` finds that all available candidate pairs (x, y) are unfeasible for extending a current set of matches $\mathcal{M}_\alpha$, the algorithm backtracks by removing from $\mathcal{M}_\alpha$ the last added match, and then testing other alternative candidate pairs.

This progressive exploration behavior of available matches is ideal for our *anchored isomorphism* search, of which we include a high-level description in Algorithm 1. Given a partial map π, moreover, for a balanced reaction $G \longrightarrow H$, the collection $\mathcal{M}_\pi$ of pairs (x, y) with $\pi(x) = y$, actually constitutes an *initial state* for the set $\mathcal{M}_\alpha$ described before. In other words, since an isomorphism α between the reminder graphs $\hat{G}_\pi$ and $\hat{H}_\pi$, necessarily coincides with π on all reacting vertices if it is a (stable) extension of π, the set $\mathcal{M}_\pi$, prepared in lines 8 to 10 of Algorithm 1, therefore acts as an *anchor* for seed the `VF2` routine. This seed is then expanded by passing it to a further call to a regular `VF2` routine in line 12.

By definition the reaction center of $G \longrightarrow H$ is composed of unmatchable edges, that is, the reaction edges in G cannot be matched by the `VF2` with edges in H, and similarly, reaction edges in H will have no matching edge in G, i.e., these edges cannot satisfy either one or both feasibility tests of the `VF2`. Consequently, we remove the reaction edges in a preprocessing step in line 6 of our algorithm. We remove, moreover, all edges whose ends are both reacting vertices, simplifying the search even further whenever possible.

The remainder graphs $\hat{G}_\pi$ and $\hat{H}_\pi$ obtained after removing all reaction edges edges may be disconnected. As mentioned in Subsection 2.3, we are interested in applying Algorithm 1 only over balanced single-step reactions producing connected ITS graphs. Thus Lemma 12 implies that, even under such conditions, every connected component of $\hat{G}_\pi$ and $\hat{H}_\pi$ contains at least one reacting vertex. Hence all non-reacting vertices in $\hat{G}_\pi$ and $\hat{H}_\pi$ remain, in general, reachable during the progressive expansion with the `VF2` through the terminal sets. This implies a reduction in complexity of the search space, in particular, when processing molecular graphs, by avoiding the exhaustive evaluation of trivial or non-informative matches.

Algorithm 1 Anchored graph isomorphism search (`VF2`-variant).

Data: Reactants graph G, products graph H and partial AAM $\pi : U \to W$ as ordered list of matches $\mathcal{M}_\pi = [(g_k, h_k)]$ for $1 \leq k \leq |U|$, $g_k \in U \subseteq V(G)$, $h_k \in W \subseteq V(H)$, $\pi(g_k) = h_k$.
Result: An isomorphism α from $\hat{G}_\pi$ to $\hat{H}_\pi$ extending π (if any) as an ordered list of matches $\mathcal{M}_\alpha = [(g_{k'}, h_{k'})]$ for $1 \leq k' \leq |V(G)|$, $g_{k'} \in V(G)$, $h_{k'} \in V(H)$, $\alpha(g_{k'}) = h_{k'}$, and such that $\mathcal{M}_\alpha \supseteq \mathcal{M}_\pi$, or `False` if no such α exists.

```
1  /* Get anchor sets of input graphs                                          */
2  A_G ← {g_k ∈ V(G) | (g_k, h_k) ∈ M_π} and A_H ← {h_k ∈ V(H) | (g_k, h_k) ∈ M_π};
3  /* Get heuristic total order for V(G) necessary for VF2-search              */
4  T ← Total_Order_Routine(G, A_G);
5  /* Initialize search state + ignoring input anchors and their edges         */
6  S ← Search_State_Constructor(G, H, T, A_G, A_H);
7  /* Add M_π to search state + updating VF2's terminal and unmatched sets     */
8  foreach (g_k, h_k) ∈ M_π do S ← Extend_Matches_in_Search_State(g_k, h_k, S);
9  /* Call to regular VF2-search with primed state + extends matches internally */
10 S ← VF2_Isomorphism_Search(S);
11 /* Unpack isomorphism (if any) or list of matches of maximum cardinality    */
12 M_α ← S.list_of_matches;
13 /* Report result                                                            */
14 if |M_α| < |V(G)| then return(False); else return(M_α) ;
```

Publicly available implementations of the `VF2` algorithm, such as the one from the `Python` package `NetworkX` [18], do not provide default options to handle the initialization of a search state with an anchor map, as required by line 9 of Algorithm 1. We opted, therefore, to develop a custom implementation of this anchored search. For this we made use, in particular, of the `Cython` language [4], which allows the implementation of `C` and `C++` data structures through a `Python`-like syntax, and facilitates the back-and-ford conversion of these data containers and data types, respectively, with `Python` native objects and types.

Our implementation is available as the routine `search_stable_extension`, inside the `Python` package `GranMapache` (**GRA**phs-and-**N**etworks **MAP**ping **A**pplications with **C**ython and **HE**uristics), in the repository: `https://github.com/MarcosLaffitte/GranMapache`, where we provide diverse functionalities to address problems related to mappings between graphs.

Relabeling-and-isomorphism strategy. Throughout this contribution, except for a few cases, e.g. for the definition of ITS graphs, we have considered vertex labels to represent atom types. Formally speaking, nonetheless, these labels embody the broader notion of *comparability classes* of vertices, i.e., two vertices are *comparable* with each other if and only if they have the same label.

Here we device a relabeling strategy, condensed in Algorithm 2, that offers an equivalent, but simpler alternative, to the anchored isomorphism approach described earlier. To illustrate this, consider a balanced reaction $G \longrightarrow H$ with a partial AAM $\pi : U \to W$ with $U \subseteq V(G)$ and $W \subseteq V(H)$, and four vertices $u, x \in V(G)$ and $w, y \in V(H)$. Suppose that $\pi(u) = w$ and $a_G(x) = a_H(y)$, but $x \notin U$ and $y \notin W$. Assume, moreover, that there exists a stable extension α of π such that $\alpha(x) = y$. Thus, any algorithm capable of retrieving α as an isomorphism between the remainder graphs $\hat{G}_\pi$ and $\hat{H}_\pi$, has to (i) match again u and w without admitting for them any other matches and (ii) must be able to recognize x and y as comparable vertices. Clearly Algorithm 1 satisfies these conditions.

The same result, however, can be achieved by creating copies of $\hat{G}_\pi$ and $\hat{H}_\pi$ with new labeling functions, enforcing *comparability constrains* with these new labels. By slight abuse of notation we write $a'_G : V(G) \to \{0, 1, ..., |U|\} \times L_V$ for the labeling function on the copy of $\hat{G}_\pi$, and $a'_H : V(H) \to \{0, 1, ..., |W|\} \times L_V$ for the copy of $\hat{H}_\pi$. This labels are described formally in lines 6 and 7 of Algorithm 2. With them, for the example above, the vertices matched by π are now labeled by ordered pairs $a'_G(u) = (k, a_G(u)) = (k, a_H(w)) = a'_H(w)$ for a unique integer $k \in \{1, ..., |U|\}$, while remaining vertices are assigned a label $a'_G(x) = (0, a_G(x)) = (0, a_H(y)) = a'_H(y)$, thus satisfying as well conditions (i) and (ii) from before.

Once the new labeling functions are built, we only need to create the copies of $\hat{G}_\pi$ and $\hat{H}_\pi$ as stated in line 11 of Algorithm 2, and finally run for these graphs an arbitrary isomorphism routine handling labeled graphs, as in line 13, in order to recover the stable extension α. An example of the application of this algorithm is shown in Figure 8 in Appendix B.

We implemented this method in `Python`, making use of the `NetworkX` [18] package for the handling of graphs and their labeling functions. This implementation is also available in the repository `GranMapache`, in a directory dedicated to examples of usage of our functionalities: `https://github.com/MarcosLaffitte/GranMapache/tree/main/examples/Stable_Extensions`.

Integer Linerar Programming formulation. Finally, Algorithm 3 describes, as a pipeline, the formulation of the Integer Linear Programming (ILP) search for stable extensions, that we originally proposed in [26]. For this we made use of the `CBC solver 2.10.3` [29], made in `C++` and callable from `Python`.

Algorithm 2 Special relabeling and graph isomorphism search.

Data: Reactants graph G, products graph H and partial AAM $\pi : U \to W$ as ordered list of matches $\mathcal{M}_\pi = [(g_k, h_k)]$ for $1 \le k \le |U|$, $g_k \in U \subseteq V(G)$, $h_k \in W \subseteq V(H)$, $\pi(g_k) = h_k$.
Result: An isomorphism α from $\hat{G}_\pi$ to $\hat{H}_\pi$ extending π (if any) as an ordered list of matches $\mathcal{M}_\alpha = [(g_{k'}, h_{k'})]$ for $1 \le k' \le |V(G)|$, $g_{k'} \in V(G)$, $h_{k'} \in V(H)$, $\alpha(g_{k'}) = h_{k'}$, and such that $\mathcal{M}_\alpha \supseteq \mathcal{M}_\pi$, or `False` if no such α exists.

```
/* Get anchor sets of input graphs */
```
$\mathcal{A}_G \leftarrow \{g_k \in V(G) \mid (g_k, h_k) \in \mathcal{M}_\pi\}$ and $\mathcal{A}_H \leftarrow \{h_k \in V(H) \mid (g_k, h_k) \in \mathcal{M}_\pi\}$;
```
/* Get removable edges - with both ends in an anchor (includes reaction edges) */
```
$\mathcal{R}_G \leftarrow \{xy \in E(G) \mid x, y \in \mathcal{A}_G\}$ and $\mathcal{R}_H \leftarrow \{uv \in E(H) \mid u, v \in \mathcal{A}_H\}$;
```
/* Prepare new vertex-labeling functions */
```
the new vertex-labeling w.r.t G is the map $a'_G : V(G) \to \{0, 1, \ldots, |U|\} \times L_V$, given by,
$a'_G \leftarrow \{(g_k, (k, a_G(g_k))) \mid (g_k, h_k) \in \mathcal{M}_\pi\} \cup \{(x, (0, a_G(x))) \mid x \in V(G) \setminus \mathcal{A}_G\}$;
the new vertex-labeling w.r.t H is the map $a'_H : V(H) \to \{0, 1, \ldots, |W|\} \times L_V$, given by,
$a'_H \leftarrow \{(h_k, (k, a_H(h_k))) \mid (g_k, h_k) \in \mathcal{M}_\pi\} \cup \{(y, (0, a_H(y))) \mid y \in V(H) \setminus \mathcal{A}_H\}$;
```
/* Prepare new edge-labeling functions */
```
these are the restrictions b'_G of b_G to $E(G) \setminus \mathcal{R}_G$ and b'_H of b_H to $E(H) \setminus \mathcal{R}_H$;
```
/* Build (reduced) remainder graphs - no edges with both ends in an anchor */
```
$\hat{G}_\pi \leftarrow (V(G),\ E(G) \setminus \mathcal{R}_G,\ a'_G,\ b'_G)$ and $\hat{H}_\pi \leftarrow (V(H),\ E(H) \setminus \mathcal{R}_H,\ a'_H,\ b'_H)$;
```
/* Evaluate and get stable extension as isomorphism of remainder graphs */
```
$\mathcal{M}_\alpha \leftarrow$ `Search_Graph_Isomorphism`$(\hat{G}_\pi,\ \hat{H}_\pi)$;
```
/* Report result */
```
if $|\mathcal{M}_\alpha| = 0$ **then return(False); else return($\mathcal{M}_\alpha$)** ;

Algorithm 3 Isomorphism of remainder graphs with ILP.

Data: Reactants graph G, products graph H and partial AAM $\pi : U \to W$ as ordered list of matches $\mathcal{M}_\pi = [(g_k, h_k)]$ for $1 \le k \le |U|$, $g_k \in U \subseteq V(G)$, $h_k \in W \subseteq V(H)$, $\pi(g_k) = h_k$.
Result: An isomorphism α from $\hat{G}_\pi$ to $\hat{H}_\pi$ extending π (if any) as an ordered list of matches $\mathcal{M}_\alpha = [(g_{k'}, h_{k'})]$ for $1 \le k' \le |V(G)|$, $g_{k'} \in V(G)$, $h_{k'} \in V(H)$, $\alpha(g_{k'}) = h_{k'}$, and such that $\mathcal{M}_\alpha \supseteq \mathcal{M}_\pi$, or `False` if no such α exists.

```
/* Denote order of graphs by N := |V(G)| = |V(H)| and maximum bond order as K */
/* Get anchor sets of input graphs */
```
$\mathcal{A}_G \leftarrow \{g_k \in V(G) \mid (g_k, h_k) \in \mathcal{M}_\pi\}$ and $\mathcal{A}_H \leftarrow \{h_k \in V(H) \mid (g_k, h_k) \in \mathcal{M}_\pi\}$;
```
/* Obtain adjacency matrices with entries saving the bond orders, i.e., {1,2,3} */
```
$\mathbb{A}_G \leftarrow$ `Adjacency_Matrix`(G) and $\mathbb{A}_H \leftarrow$ `Adjacency_Matrix`(H);
```
/* Get total order for V(G) and V(H) extending the enumeration given by M_π */
```
$T_G \leftarrow \{(k, g_k) \mid (g_k, h_k) \in \mathcal{M}_\pi\} \cup$ `Enumeration`$(\{|\mathcal{A}_G| + 1, \cdots, N\} \times (V(G) \setminus \mathcal{A}_G))$;
$T_H \leftarrow \{(k, h_k) \mid (g_k, h_k) \in \mathcal{M}_\pi\} \cup$ `Enumeration`$(\{|\mathcal{A}_H| + 1, \cdots, N\} \times (V(H) \setminus \mathcal{A}_H))$;
```
/* Initialize ILP model with matrices 𝕏 for α and 𝔻 for degree differences */
```
$\mathcal{S} \leftarrow$ `ILP_Solver_Constructor`$(\mathbb{X} =$ `Zero_Matrix`$(N \times N),\ \mathbb{D} = \mathbb{A}_H\mathbb{X} - \mathbb{X}\mathbb{A}_G)$;
```
/* Add binary variables and set logical constraints and constraints from M_π */
```
for $i = 1;\ i \le N;\ i++$ **do**
  for $j = 1;\ j \le N;\ j++$ **do**
    $\mathcal{S}$.`add_binary_variables`$(s_{ij},\ g_{ij})$;
    $\mathcal{S}$.`add_constraints`$(\mathbb{D}_{ij} \le K g_{ij}$ and $\mathbb{D}_{ij} \ge -K + g_{ij}(K+1))$;
    $\mathcal{S}$.`add_constraints`$(-\mathbb{D}_{ij} \le K s_{ij}$ and $-\mathbb{D}_{ij} \ge -K + s_{ij}(K+1))$;
    if $i = j$ and $i \le |\mathcal{M}_\pi|$ **then** $\mathcal{S}$.`add_constraint`$(\mathbb{X}_{ij} = 1)$;
    if $a_G(T_G[i]) \ne a_H(T_H[j])$ **then** $\mathcal{S}$.`add_constraint`$(\mathbb{X}_{ij} = 0)$;
  end
end
```
/* Add linear constraints necessary for 𝕏 to be a permutation matrix */
```
for $i = 1;\ i \le N;\ i++$ **do** $\mathcal{S}$.`add_constraint`$(\Sigma_{j=1}^{N} \mathbb{X}_{ij} = 1)$;
for $j = 1;\ j \le N;\ j++$ **do** $\mathcal{S}$.`add_constraint`$(\Sigma_{i=1}^{N} \mathbb{X}_{ij} = 1)$;
```
/* We seek to maximize zero entries in 𝔻 and so we minimize the cost function f */
```
$\mathcal{S}$.`minimize`$(f = \Sigma_{ij}(g_{ij} + s_{ij}))$;
```
/* Get isomorphism (if any) from entries */
```
$\mathcal{M}_\alpha \leftarrow \{(T_G[i], T_H[j]) \mid \mathcal{S}.\mathbb{X}_{ij} = 1\}$;
```
/* Report result */
```
if $|\mathcal{M}_\alpha| < |V(G)|$ **then return(False); else return($\mathcal{M}_\alpha$)** ;

4 Benchmarking, Empirical Results and Discussion

We evaluate the methodologies discussed in the previous section over two sets of empirical data, one of real chemical reactions with partial AAMs covering exactly the ground-truth reacting vertices, and another set consisting of random graphs with AAMs inducing connected ITS graphs with connected reaction centers and an increasing number of nodes and edges.

Four different implementations are evaluated: (**GM**) our custom anchored isomorphism search from our package `GranMapache` [24] implementing Algorithm 1, two relabeling-and-isomorphism variants implementing Algorithm 2 from which one (**RB$_1$**) uses the `VF2` isomorphism function [9] from `NetworkX` and another (**RB$_2$**) uses a custom isomorphism function from `GranMapache`, and lastly (**ILP**) implementing the ILP formulation in Algorithm 3 from the package `AAMutils` introduced in previous work [26].

Stable extensions over real chemical reactions

The reactions were retrieved from `USPTO_50K` corpus of 50,016 reactions [36]. Each reaction was rebalanced with the tool `SynRBL` [34] and preprocessed with the `SynTemp` pipeline [35], which enforces ensemble consistency, resolves hydrogen ambiguity and validates that the partial maps are good. The preprocessing stage yield 39,732 rebalanced and consistent reactions with full AAMs, good partial AAMs derived from the full ones, and the corresponding ITS graphs. We process the mapped reactions in SMILES format, via another custom tool `SynKit` for the methods **GM**, **RB$_1$** and **RB$_2$**, and with the `AAMutils` API [26] for the **ILP**.

The 39,732 reactions were processed five times with each of the four methods. Here we report the average running time of these procedures over this data set. From Theorem 20 and Corollary 21, moreover, it follows that all possible full AAMs recovered by our algorithms are to be equivalent and thus produce isomorphic ITS graphs. As a back-up test for our implementations, therefore, we corroborate the successful recovery of the full AAMs by means of the isomorphism of the initial and recovered ITS graphs. The graph-based methods **GM**, **RB$_1$** and **RB$_2$** were able to recover 100% of the ground AAMs of the 39,732 reactions, while the **ILP** retrieved the AAMs successfully in 99.48% of the reactions. The few ILP mismatches are attributable to discrepancies in the canonicalization SMILES during the conversion of the output of reaction-mapping tools to the graph representations used here.

Figure 4 below shows the distributions of the average running time per reaction for each method, and with respect to each of the five repetitions. The numerical values of the average running times per trial are summarized in Table 1 in Appendix B. All benchmarks were run under Python 3.11 on a 12-core Intel Core i7-8700 @ 3.20 GHz, Fedora 37. The programs made for this analysis can be found in `https://github.com/TieuLongPhan/PartialAAMs`.

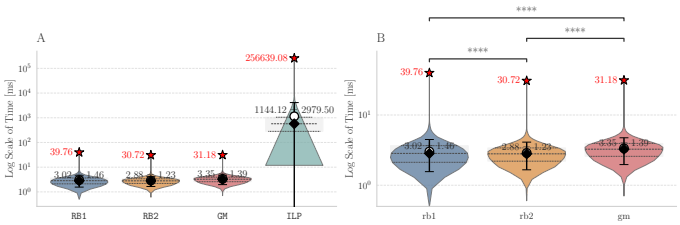

Figure 4 (**A**) Running times per reaction and per execution of the four methods. (**B**) Running time focused only on the graph-based methods **GM**, **RB$_1$** and **RB$_2$**. Numerical values [ms] of average and maximum outlier are reported in black and red, respectively.

While the ILP average running time exceeds 1 second, all graph-based methods take only a few milliseconds ($p < 0.05$). Among these, $\mathbf{RB_2}$ is the fastest on average for processing the molecular graphs (2.88 ± 1.22 ms), followed by $\mathbf{RB_1}$ (3.02 ± 1.46 ms) and **GM** (3.36 ± 1.39 ms). The small differences of 0.14 to 0.48 ms attained on this set of real molecular graphs, and disclosed by Figure 4, are attributable to the time required by the custom implementations $\mathbf{RB_2}$ and **GM** to convert native `Python` objects into `C++` containers and vice versa, carried by the `Cython` functions on run time when called by other external `Python` scripts.

Tests for scalability were carried by analyzing the obtained running times while varying the number of vertices in the input molecular graphs. The distribution of the reactions with each number of vertices is shown in Figure 9A in Appendix B. On the other hand, Figure 10 shows that for small graphs (less than 30 vertices), $\mathbf{RB_1}$ is faster than $\mathbf{RB_2}$ and **GM**, while beyond the 30-vertices threshold $\mathbf{RB_2}$ scales more favorably. Moreover, **GM** surpasses $\mathbf{RB_1}$ on larger graphs, suggesting that the custom implementations $\mathbf{RB_2}$ and **GM** offer an scalability advantage with respect to the amount of vertices in the input graphs. The small discrepancies in the running time of $\mathbf{RB_2}$ and **GM** is further explained by the implementation of different heuristics for building the total order required by `VF2`-like approaches. The trial and evaluation of such heuristics is out of the scope of this contribution.

Another scalability analysis was carried with respect to the proportion of reacting vertices vs total vertices in the ITS graph of each reaction. See the distribution of such proportions in Figure 9B. The scatter plot in Figure 5 below, together with Figures 11 and 12 in Appendix B, suggest that all graph-based methods perform best for reactions with bigger proportions of reacting vertices from the total amount of vertices. This is expected since smaller proportions of reacting vertices lead to a bigger search space for the `VF2`-based approaches.

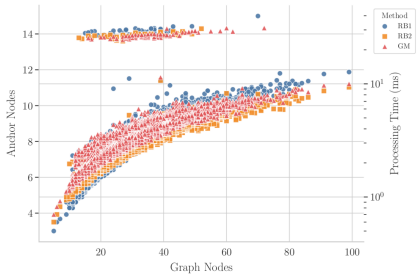

Figure 5 Scatter plot projection. Counts of graph vertex (x-axis) vs of reacting vertices (left y-axis) and running times (right y-axis [ms]) for the **GM**, $\mathbf{RB_1}$, and $\mathbf{RB_2}$ methods.

Stable extensions over random graphs

We implemented the generation and analysis of random ITS graphs in `Cython` making use, in particular, of `NetworkX` [18]. All scripts for this analysis can be found in the `GranMapache` repository [24], and were run under Python 3.11 on an 8-core 11th Gen Intel Core i7-1165G7 @ 2.80GHz, Lenovo ThinkPad E15 Gen 2 with 16GB and Ubuntu 22.04.

For this analysis we produced connected ITS graphs with connected reaction centers. These graphs were built with an increasing amounts of edges outside the reaction center so as to test the performance of our methods over (simple) labeled graphs with varying density, i.e., with an increasing proportion of existing edges in the graph with respect to the theoretical maximum. Based on this we built 5 data sets of such graphs having each a different number of non-reacting vertices, specifically for 100, 125, 150, 175 and 200 nodes.

Within each data set we built 10 ITS graphs per each percentage-point for the percentage of edges outside the reaction center, starting at 3% and up to 97%. This leads to a total of 4,750 randomly generated ITS graphs conforming ∼1.6 GB of labeled graphs serialized and stored with the package `Pickle` [12] from `Python`. The interval 3 - 97 % was chosen under theoretical reasons for the graphs to be connected and for them to have the exact specified density. All the graphs were produced with a (randomly generated) connected reaction center of 15 vertices and 20 edges, in addition to the specified number of non-reacting vertices.

The pairs of labels needed for vertices and edges of these ITS graphs were chosen uniformly at random from a set of 6 integer labels for the vertices, and 3 labels for the edges. The labels in the reaction center, moreover, were produced by selecting the *source* of the (reaction) edge uniformly at random from the reactants graph, the products graph, or both of them. Finally, we tested the extension of the partial map covering exactly the reaction center.

For this we made use of the graph-based methods $\mathbf{RB_1}$, $\mathbf{RB_2}$ and **GM**, but omitted the **ILP** approach due to its comparably slower performance. All methods successfully extended the reaction center in all cases. The average running times for the analysis over the graphs with 100 non-reacting vertices are summarized in Figure 6, while the results for graphs with 125-200 vertices are included in Figure 13 in Appendix B. These show a consistent hierarchical performance, where **GM** completes the analysis faster, followed by $\mathbf{RB_2}$, and then $\mathbf{RB_1}$. This is consistent with the observations over real reactions, where molecules have at most 98 vertices and 108 edges and thus density $\leq 2.28\%$. This suggests again that the custom methods **GM** and $\mathbf{RB_2}$ are appropriate for dealing with bigger graphs, while $\mathbf{RB_1}$ proves to be comparably efficient for processing smaller and more sparse graphs.

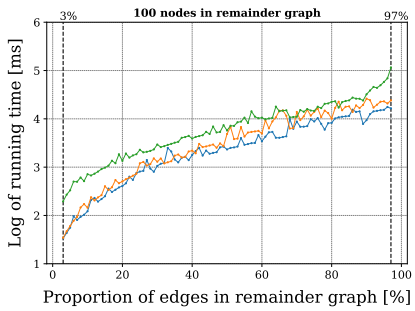

Figure 6 Average running times of different algorithms for stable extensions $\mathbf{RB_1}$: green, custom $\mathbf{RB_2}$: orange, and **GM**: blue) on ITS graphs with 100 nodes and varying densities.

5 Concluding Remarks

In this contribution we first gave a comprehensive mathematical description of the relationships between good partial AAMs and full descriptions of the underlying reaction. In particular, we showed that good partial AAMs, i.e., those that cover the reaction center, are exactly the partial AAMs that have unique extensions constituting isomorphisms of the remainder graphs of the reactant and product sides. These results extend our work in [26] by establishing equivalent characterizations of good AAMs. This shows that the practical problem of determining whether a partial AAM is good and, if so, retrieving its unique stable extension, is equivalent to a restricted graph isomorphism problem. Based on these theoretical insights we benchmark different implementations of graph isomorphism tests. Not all such methods lend themselves to incorporate additional constraints implied by the

partial AAM. In VF2-like methods, these constitute an immutable set of initial matches. Canonization-based methods can be used after modifying the vertex labels, such that the two vertices of the corresponding pairs $(x, \pi(x))$ are assigned unique matching labels. For comparison we also consider an ILP formulation. Benchmarking simulations show that the completion is feasible for graphs relevant to applications in chemistry. Moreover, we observe that dedicated graph isomorphism algorithms are much more efficient than the ILP.

In contrast to graph-based methods, however, the ILP can potentially be used for *bad* partial AAMs, for which the plausible extensions are, of course, no longer isomorphisms between $\hat{G}_\pi$ and $\hat{H}_\pi$, and are determined by minimizing the number of necessary reaction edges in an AAM α extending π [26]. This is of practical importance since most reaction mapping tools do not predict correspondences between hydrogen atoms, though these often take part in the reactions and hence are reactive vertices. A further extension to the – usually bad – partial AAMs on unbalanced reactions remains an open problem for future research.

References

1. Alpár Alpár Jüttner and Péter Madarasi. VF2++ – An improved subgraph isomorphism algorithm. *Discr. Appl. Math.*, 242:69–81, 2018. doi:10.1016/j.dam.2018.02.018.
2. Jakob L Andersen, Christoph Flamm, Daniel Merkle, and Peter F. Stadler. Inferring chemical reaction patterns using graph grammar rule composition. *J. Syst. Chem.*, 4:4, 2013. doi:10.1186/1759-2208-4-4.
3. Jakob Lykke Andersen, Christoph Flamm, Daniel Merkle, and Peter F. Stadler. A software package for chemically inspired graph transformation. In Rachid Echahed and Mark Minas, editors, *Graph Transformation, ICGT 2016*, volume 9761 of *Lecture Notes Comp. Sci.*, pages 73–88, Berlin, Heidelberg, D, 2016. Springer Verlag. doi:10.1007/978-3-319-40530-8_5.
4. Stefan Behnel, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn, and Kurt Smith. Cython: The best of both worlds. *Computing in Science Engineering*, 13(2):31–39, March 2011. doi:10.1109/MCSE.2010.118.
5. Nora Beier, Thomas Gatter, Jakob Lykke Andersen, and Peter F. Stadler. Computing double-pushout graph transformation rules and atom-to-atom maps from KEGG RCLASS data, 2025. submitted to *Alg. Mol. Biol.* doi:10.21203/rs.3.rs-6765982/v1.
6. Shuan Chen, Sunggi An, Ramil Babazade, and Yousung Jung. Precise atom-to-atom mapping for organic reactions via human-in-the-loop machine learning. *Nature Communications*, 15:2250, 2024. doi:10.1038/s41467-024-46364-y.
7. Luigi P. Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Trans. Pattern Analysis Machine Intelligence*, 26:1367–1372, 2004. doi:10.1109/TPAMI.2004.75.
8. Andrea Corradini, Ugo Montanari, Francesca Rossi, Hartmut Ehrig, Reiko Heckel, and Michael Löwe. Algebraic approaches to graph transformation–part i: Basic concepts and double pushout approach. In *Handbook Of Graph Grammars And Computing By Graph Transformation: Volume 1: Foundations*, pages 163–245. World Scientific, 1997.
9. NetworkX Documentation. NetworkX 3.4.2: VF2 Algorithm. https://networkx.org/documentation/stable/reference/algorithms/isomorphism.vf2.html.
10. Hans-Christian Ehrlich and Matthias Rarey. Maximum common subgraph isomorphism algorithms and their applications in molecular science: a review. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(1):68–79, 2011.
11. Christoph Flamm, Stefan Müller, and Peter F. Stadler. Every atom-atom map for neutral molecules can be explained by electron pair pushing diagrams. *Discr. Math. Chem.*, 2024. in press; arxiv doi:. doi:10.48550/arXiv.2311.13492.
12. Python Software Foundation. Pickle package for serialization of Python objects., 2025. Accessed: 2025-06-22. URL: https://docs.python.org/3/library/pickle.html.

13 Shinsaku Fujita. Description of organic reactions based on imaginary transition structures. 1. Introduction of new concepts. *J. Chem. Inf. Comput. Sci.*, 26:205–212, 1986. doi:10.1021/ci00052a009.

14 Shinsaku Fujita. Imaginary transition structures. a novel approach to computer oriented representation of organic reactions. *J. Synth. Org. Chem. Japan*, 47(5):396–412, 1989. doi:10.5059/yukigoseikyokaishi.47.396.

15 Kimito Funatsu, Tomoaki Endo, Norio Kotera, and Shin-Ichi Sasaki. Automatic recognition of reaction site in organic chemical reactions. *Tetrahedron Comput Methodology*, 1(1):53–69, 1988. doi:10.1016/0898-5529(88)90008-5.

16 Jonathan Goodman. Computer software review: Reaxys. *J. Chem. Inf. Model.*, 49(12):2897–2898, 2009. doi:10.1021/ci900437n.

17 Martin Grohe, Daniel Neuen, and Pascal Schweitzer. A faster isomorphism test for graphs of small degree. *SIAM J. Computing*, 52, 2023. doi:10.1137/19M1245293.

18 Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11–15, Pasadena, CA USA, 2008.

19 Frank Harary. *Graph theory*. CRC Press, FL, USA, 2018.

20 James B. Hendrickson. Comprehensive system for classification and nomenclature of organic reactions. *J. Chem. Inf. Comput. Sci.*, 37(5):852–860, 1997. doi:10.1021/ci970040v.

21 J. Hine. The principle of least nuclear motion. *Advances in Physical Organic Chemistry*, 15:1–61, 1977. doi:10.1016/S0065-3160(08)60117-3.

22 Frank Hoonakker, Nicolas Lachiche, Alexandre Varnek, and Alain Wagner. A representation to apply usual data mining techniques to chemical reactions – illustration on the rate constant of SN_2 reactions in water. *Int. J. Artif. Intelligence Tools*, 20:253–270, 2011. doi:10.1142/S0218213011000140.

23 Clemens Jochum, Johann Gasteiger, and Ivar Ugi. The principle of minimum chemical distance (PMCD). *Ang. Chem. Intl. Ed.*, 19(7):495–505, 1980. doi:10.1002/anie.198004953.

24 Marcos E. González Laffitte. GranMapache: GRAphs-and-Networks MAPping Applications with Cython and HEuristics). https://github.com/MarcosLaffitte/GranMapache.

25 Marcos E. González Laffitte, Nora Beier, Nico Domschke, and Peter F. Stadler. Comparison of Atom Maps. *MATCH Commun. Math. Comput. Chem.*, 90:75–102, 2023. doi:10.46793/match.90-1.075G.

26 Marcos E. González Laffitte, Klaus Weinbauer, Tieu-Long Phan, Nora Beier, Nico Domschke, Christoph Flamm, Thomas Gatter, Daniel Merkle, and Peter F. Stadler. Partial imaginary transition state (ITS) graphs: A formal framework for research and analysis of atom-to-atom maps of unbalanced chemical reactions and their completions. *MDPI Symmetry*, 16(9), 2024. doi:10.3390/sym16091217.

27 Jie Jack Li. *Name Reactions: A Collection of Detailed Reaction Mechanisms and Synthetic Applications*. Springer, 2025.

28 Arkadii Lin, Natalia Dyubankova, Timur I Madzhidov, Ramil I Nugmanov, Jonas Verhoeven, Timur R Gimadiev, Valentina A Afonina, Zarina Ibragimova, Assima Rakhimbekova, Pavel Sidorov, et al. Atom-to-atom mapping: a benchmarking study of popular mapping algorithms and consensus strategies. *Molecular Informatics*, 41(4):2100138, 2022.

29 Jeffrey T. Linderoth and Ted K. Ralphs. Noncommercial software for mixed-integer linear programming. In John Karlof, editor, *Integer programming: theory and practice*, pages 253–303. CRC Press, 2005.

30 Daniel Mark Lowe. Extraction of chemical structures and reactions from the literature. Technical report, Apollo – University of Cambridge Repository, 2012. doi:10.17863/CAM.16293.

31 Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comp. Syst. Sci.*, 25:42–65, 1982. doi:10.1016/0022-0000(82)90009-5.

32 Michael F. Lynch and Peter Willett. The automatic detection of chemical reaction sites. *J Chem Inf Comput Sci*, 18:154–159, 1978. doi:10.1021/ci60015a009.

33 Alfred J. Meijer, Wouter H. Lamers, and Robert A. F. M. Chamuleau. Nitrogen metabolism and ornithine cycle function. *Physiological Reviews*, 70(3):701–748, 1990. PMID: 2194222. doi:10.1152/physrev.1990.70.3.701.

34 Tieu-Long Phan, Klaus Weinbauer, Thomas Gärtner, Daniel Merkle, Jakob L Andersen, Rolf Fagerberg, and Peter F Stadler. Reaction rebalancing: a novel approach to curating reaction databases. *Journal of Cheminformatics*, 16(1):82, 2024. doi:10.1186/S13321-024-00875-4.

35 Tieu-Long Phan, Klaus Weinbauer, Marcos E. González Laffitte, Yingjie Pan, Daniel Merkle, Jakob L. Andersen, Rolf Fagerberg, Christoph Flamm, and Peter F. Stadler. SynTemp: Efficient Extraction of Graph-Based Reaction Rules from Large-Scale Reaction Databases. *Journal of Chemical Information and Modeling*, 65(6):1549–9596, 2025. doi:10.1021/acs.jcim.4c01795.

36 Nadine Schneider, Nikolaus Stiefl, and Gregory A Landrum. What's what: The (nearly) definitive guide to reaction role assignment. *Journal of chemical information and modeling*, 56(12):2336–2346, 2016. doi:10.1021/ACS.JCIM.6B00564.

37 Craig S. Wilcox and Robert A. Levinson. A self-organized knowledge base for recall, design, and discovery in organic chemistry. In T. H. Pierce and B. A. Hohne, editors, *Artificial Intelligence Applications in Chemistry*, volume 306 of *Am. Chem. Soc. symposium series*, chapter 18, pages 209–230. American Chemical Society, Washington, DC, 1986. doi:10.1021/bk-1986-0306.ch018.

A Appendix: Mathematical Results and Proofs

▶ **Proposition 5.** *Let α be an AAM for the balanced reaction $G \longrightarrow H$ and let β be an isomorphism from the remainder graph $\hat{G}_\alpha$ to the remainder graph $\hat{H}_\alpha$. If $\alpha(x) = \beta(x)$ holds for all reacting vertices x of α, then α and β are equivalent AAMs for $G \longrightarrow H$.*

Proof. To prove the equivalence of α and β it suffices to show that $\beta^{-1}\alpha \in Aut(G)$. To see this note first that by Lemma 3 we have $\alpha \in \mathrm{Iso}(\hat{G}_\alpha, \hat{H}_\alpha)$. Thus by inverses and composition of isomorphisms it follows $\beta^{-1}\alpha \in Aut(\hat{G}_\alpha)$ and therefore, by definition, we get $a_G(x) = a_{\hat{G}_\alpha}(x) = a_{\hat{G}_\alpha}(\beta^{-1}\alpha(x)) = a_G(\beta^{-1}\alpha(x))$ for all $x \in V(G)$. Consider now $xy \in E(G)$. Then either $xy \in E(\hat{G}_\alpha)$ or xy is a reaction edge of G. In the first case, from $\beta^{-1}\alpha \in Aut(\hat{G}_\alpha)$ follows that **(A1)** $xy \in E(\hat{G}_\alpha)$ if and only if $\beta^{-1}\alpha(x)\beta^{-1}\alpha(y) \in E(\hat{G}_\alpha)$, and again $b_G(xy) = b_{\hat{G}_\alpha}(xy) = b_{\hat{G}_\alpha}(\beta^{-1}\alpha(x)\beta^{-1}\alpha(y)) = b_G(\beta^{-1}\alpha(x)\beta^{-1}\alpha(y))$. Suppose, on the other hand, that $xy \in E(G)$ is a reaction edge, i.e., $xy \in (E(G) \setminus E(\hat{G}_\alpha))$. Then, both x and y are reacting vertices, and by hypothesis $\alpha(x) = \beta(x)$ and $\alpha(y) = \beta(y)$, or equivalently $\beta^{-1}\alpha(x) = x$ and $\beta^{-1}\alpha(y) = y$. Thus formally we also have **(A2)** $xy \in (E(G) \setminus E(\hat{G}_\alpha))$ if and only if $\beta^{-1}\alpha(x)\beta^{-1}\alpha(y) \in (E(G) \setminus E(\hat{G}_\alpha))$ and similarly $b_G(xy) = b_G(\beta^{-1}\alpha(x)\beta^{-1}\alpha(y))$. In this way, from **(A1)** and **(A2)** we get $xy \in E(G)$ if and only if $\beta^{-1}\alpha(x)\beta^{-1}\alpha(y) \in E(G)$, i.e., $\beta^{-1}\alpha$ preserves adjacency in G, and since it also preserves vertex and edge labels we have $\beta^{-1}\alpha \in Aut(G)$. Therefore, by setting $\varphi := \beta^{-1}\alpha \in Aut(G)$ and $\psi := i_H \in Aut(H)$ for the identity automorphism $i_H : V(H) \to V(H)$, we get $\psi\alpha = i_H\alpha = (\beta\beta^{-1})\alpha = \beta(\beta^{-1}\alpha) = \beta\varphi$ and thus $\alpha \equiv \beta$, proving the proposition. ◀

▶ **Lemma 7.** *Let $\Upsilon^{\circledast}(G, H, \alpha)$ be the (non-empty) collection of all ITS graphs built for a balanced reaction $G \longrightarrow H$ with AAM α and consider a graph $T \in \Upsilon^{\circledast}(G, H, \alpha)$. Then $T' \in \Upsilon^{\circledast}(G, H, \alpha)$ if and only if $T' \simeq T$.*

Proof. Suppose first that $T' \in \Upsilon^{\circledast}(G, H, \alpha)$. Let τ and τ' be the bijections required by Definition 6 for T and T', respectively.

Consider two arbitrary vertices u and v in $V(G)$ and their preimages $\tau^{-1}(u)$ and $\tau^{-1}(v)$ in $V(T)$. Condition (i) in the definition can be restated for u and v as $\tau^{-1}(u)\tau^{-1}(v) \in E(T)$, if and only if, $uv \in E(G)$ or $\alpha(u)\alpha(v) \in E(H)$. At the same time, for two vertices $x, y \in V(T')$ it holds $xy \in E(T')$, if and only if, $\tau'(x)\tau'(y) \in E(G)$ or $\alpha(\tau'(x))\alpha(\tau'(y)) \in E(H)$. From these statements it follows that $xy \in E(T')$ if and only if $\tau^{-1}(\tau'(x))\tau^{-1}(\tau'(y)) \in E(T)$, i.e., the bijection $\tau^{-1}\tau' : V(T') \to V(T)$ preserves adjacency between T and T'. We see, moreover, that $a_T(\tau^{-1}\tau'(x)) = (a_G(\tau(\tau^{-1}\tau'(x))), a_H(\alpha(\tau(\tau^{-1}\tau'(x))))) = (a_G(\tau'(x)), a_H(\alpha(\tau'(x)))) = a_{T'}(x)$, thus the map $\tau^{-1}\tau'$ preserves vertex labels. Similarly, and without loss of generality, we have

$$\begin{aligned} b_T(\tau^{-1}\tau'(x)\tau^{-1}\tau'(y)) &= (b_G(\tau(\tau^{-1}\tau'(x))\tau(\tau^{-1}\tau'(y))), b_H(\alpha(\tau(\tau^{-1}\tau'(x))))\alpha(\tau(\tau^{-1}\tau'(x)))) \\ &= \dots \\ \dots &= (b_G(\tau'(x)\tau'(y)), b_H(\alpha(\tau'(x))\alpha(\tau'(x)))) = b_{T'}(xy), \end{aligned}$$

that is, $\tau^{-1}\tau'$ also preserves edge labels and is, therefore, an isomorphism from T' to T, i.e., $T' \simeq T$ proving the forward direction.

Suppose, on the other hand, that there exist an isomorphism $\varphi : V(T') \to V(T)$ from T' to T. Given that φ preserves adjacency, vertex-labels and edge-labels, it follows that: (i) $xy \in E(T')$, if and only if, $\varphi(x)\varphi(y) \in E(T')$, and equivalently, $\tau(\varphi(x))\tau(\varphi(y)) \in E(G)$ or $\alpha(\tau(\varphi(x)))\alpha(\tau(\varphi(y))) \in E(H)$, also (ii) $a_{T'}(x) = a_T(\varphi(x)) = (a_G(\tau(\varphi(x))), a_H(\alpha(\tau(\varphi(y)))))$, and lastly (iii) $b_{T'}(xy) = a_T(\varphi(x)\varphi(y)) = (b_G(\tau(\varphi(x))\tau(\varphi(y))), b_H(\alpha(\tau(\varphi(x)))\alpha(\tau(\varphi(y)))))$. Thus $\tau\varphi : V(T') \to V(G)$ satisfies all the conditions required by Definition 6 for T' and therefore $T' \in \Upsilon^{\circledast}(G, H, \alpha)$, which proves the converse statement. ◀

▶ **Proposition 9.** *Let α and β be AAMs for, respectively, two balanced reactions $G \longrightarrow H$ and $G' \longrightarrow H'$, and let $\Upsilon(G, H, \alpha)$ and $\Upsilon(G', H', \beta)$ be their corresponding ITS representations. Then $\Upsilon(G, H, \alpha) \simeq \Upsilon(G', H', \beta)$ if and only if $G' \simeq G$, $H' \simeq H$, and $\alpha \equiv \beta$.*

Proof. Note that by Lemma 7 and by the transitivity of the isomorphism relation $\simeq$, from $\Upsilon(G, H, \alpha) \simeq \Upsilon(G', H', \beta)$, it follows that $\Upsilon^{\perp}(G, H, \alpha) \simeq \Upsilon^{\perp}(G', H', \beta)$ holds for the canonical ITS representations $\Upsilon^{\perp}_{\alpha} := \Upsilon^{\perp}(G, H, \alpha)$ of (G, H, α) and $\Upsilon^{\perp}_{\beta} := \Upsilon^{\perp}(G', H', \beta)$ of (G', H', β), having $V(\Upsilon^{\perp}_{\alpha}) = V(G)$ and $V(\Upsilon^{\perp}_{\beta}) = V(G')$, and for which the identity maps i_G over $V(G)$ and $i_{G'}$ over $V(G')$ satisfy Definition 6, respectively. Consider an isomorphism $\varphi \in \mathrm{Iso}(\Upsilon^{\perp}_{\alpha}, \Upsilon^{\perp}_{\beta})$, and note that this is also a bijection $\varphi : V(G) \to V(G')$. Thus when applying condition (i) in Definition 6 and given that φ preserves adjacency between $\Upsilon^{\perp}_{\alpha}$ and $\Upsilon^{\perp}_{\beta}$ it follows that, $xy \in E(G)$ or $\alpha(x)\alpha(y) \in E(H)$, if and only if, $\varphi(x)\varphi(y) \in E(G')$ or $\beta(\varphi(x))\beta(\varphi(y)) \in E(H')$. This suggests that the bijections φ and $\beta\varphi\alpha^{-1} : V(H) \to V(H')$ are the required isomorphisms $\varphi \in \mathrm{Iso}(G, G')$ and $\beta\varphi\alpha^{-1} \in \mathrm{Iso}(H, H')$. To actually prove this, note first that under φ all labels are preserved component-wise, i.e., for any vertices $x, y \in V(G)$ we have, **(P1)** $(a^1_{\Upsilon^{\perp}_{\alpha}}(x), a^2_{\Upsilon^{\perp}_{\alpha}}(x)) = a_{\Upsilon^{\perp}_{\alpha}}(x) = a_{\Upsilon^{\perp}_{\beta}}(\varphi(x)) = (a^1_{\Upsilon^{\perp}_{\beta}}(\varphi(x)), a^2_{\Upsilon^{\perp}_{\beta}}(\varphi(x)))$ and **(P2)** $(b^1_{\Upsilon^{\perp}_{\alpha}}(xy), b^2_{\Upsilon^{\perp}_{\alpha}}(xy)) = b_{\Upsilon^{\perp}_{\alpha}}(xy) = b_{\Upsilon^{\perp}_{\beta}}(\varphi(x)\varphi(y)) = (b^1_{\Upsilon^{\perp}_{\beta}}(\varphi(x)\varphi(y)), b^2_{\Upsilon^{\perp}_{\beta}}(\varphi(x)\varphi(y)))$. For G and G', **(P1)** implies $a_G(x) = a^1_{\Upsilon^{\perp}_{\alpha}}(x) = a^1_{\Upsilon^{\perp}_{\beta}}(\varphi(x)) = a_{G'}(\varphi(x))$, while from **(P2)** we get $b^1_{\Upsilon^{\perp}_{\alpha}}(xy) = b^1_{\Upsilon^{\perp}_{\beta}}(\varphi(x)\varphi(y)) \in L_E \cup \{\otimes\}$, which by condition (iii) in the definition can only happen if φ preserves adjacency, i.e., $xy \in E(G)$ if and only if $\varphi(x)\varphi(y) \in E(G')$, also yielding $b_G(xy) = b_{G'}(\varphi(x)\varphi(y))$ when the edges are present in these graphs. Thus $\varphi : V(G) \to V(G')$ preserves adjacency, vertex labels and edge labels, and therefore $\varphi \in \mathrm{Iso}(G, G')$. Similarly for H and H', (P1) implies $a_H(\alpha(x)) = a^2_{\Upsilon^{\perp}_{\alpha}}(\alpha(x)) = a^2_{\Upsilon^{\perp}_{\beta}}(\beta(\varphi(x))) = a_{H'}(\beta(\varphi(x)))$, which

we can rewrite as $a_H(v) = a_{H'}(\beta(\varphi(\alpha^{-1}(v))))$ for $v := \alpha(x) \in V(H)$, thus $\beta\varphi\alpha^{-1}$ preserves vertex labels. Since from (P2) it holds $b^2_{\Upsilon^\perp_\alpha}(\alpha(x)\alpha(y)) = b^2_{\Upsilon^\perp_\beta}(\beta(\varphi(x))\beta(\varphi(y))) \in L_E \cup \{\otimes\}$, we have $b_H(\alpha(x)\alpha(y)) = b_{H'}(\beta(\varphi(x))\beta(\varphi(y)))$, and equivalently $b_H(uv) = b_{H'}(\beta(\varphi(\alpha^{-1}(u)))\beta(\varphi(\alpha^{-1}(v))))$ for $v := \alpha(x)$ and $u := \alpha(y)$ in $V(H)$, whenever the respective edges are present, while in general, from this together with condition (*iii*) in Definition 6, we get again $\alpha(x)\alpha(y) \in E(H)$ if and only if $\beta(\varphi(x))\beta(\varphi(y)) \in E(H')$, or equivalently, $uv \in E(H)$ if and only if $\beta(\varphi(\alpha^{-1}(u)))\beta(\varphi(\alpha^{-1}(v))) \in E(H')$. This shows that $\beta\varphi\alpha^{-1}$ preserves adjacency, and vertex and edge labels, and thus $\beta\varphi\alpha^{-1} \in \mathrm{Iso}(H, H')$. Lastly, since $G' \simeq G$ and $H' \simeq H$ hold, the hypothesis $\Upsilon(G, H, \alpha) \simeq \Upsilon(G', H', \beta)$, together with Corollary 8, now implies also that $\alpha \equiv \beta$. The converse statement follows from Corollary 8. ◀

▶ **Lemma 10.** *Let $\Upsilon := \Upsilon(G, H, \alpha)$ be an ITS representation of the balanced reaction $G \longrightarrow H$ with AAM α and let $\eta : V(G) \to V(\Upsilon)$ and $\eta' := \eta \circ \alpha^{-1} : V(H) \to V(\Upsilon)$ be the corresponding bijections that embed G and H into Υ, i.e., where $\eta := \tau^{-1}$ for $\tau : V(\Upsilon) \to V(G)$ as required by Definition 6. Then the following hold,*

- **(i)** *$xy \in E(G)$ is a reaction edge in G if and only if $b^1_\Upsilon(\eta(x)\eta(y)) \neq b^2_\Upsilon(\eta(x)\eta(y))$, and $x'y' \in E(H)$ is a reaction edge in H if and only if $b^1_\Upsilon(\eta'(x')\eta'(y')) \neq b^2_\Upsilon(\eta'(x')\eta'(y'))$.*
- **(ii)** *$xy \in E(\hat{G}_\alpha)$, and thus also $\alpha(x)\alpha(y) \in E(\hat{H}_\alpha)$, if and only if, $\eta(x)\eta(y) \in E(\Upsilon)$ and $b^1_\Upsilon(\eta(x)\eta(y)) = b^2_\Upsilon(\eta'(\alpha(x))\eta'(\alpha(y)))$.*

Proof. Set $\eta := \tau^{-1}$ for τ as in Definition 6. This implies that $\eta(x)\eta(y) \in E(\Upsilon)$ if and only if $xy \in E(G)$ or $\alpha(x)\alpha(y) \in E(H)$. The definition of the edge labels and the bijection $\eta' : V(H) \to V(\Upsilon)$ now yield $b_\Upsilon(\eta(x)\eta(y)) = b_\Upsilon(\eta'(\alpha(x))\eta'(\alpha(y))) = (b_G(xy), b_H(\alpha(x)\alpha(y)))$. Recall that xy is a reaction edge in G if either, $\alpha(x)\alpha(y) \in E(H)$ in which case $b_G(xy) \neq b_H(\alpha(x)\alpha(y))$, or $\alpha(x)\alpha(y) \notin E(H)$ and thus $b_\Upsilon(\eta(x)\eta(y)) = (b_G(xy), \varnothing,)$. In either case Definition 6 yields $b^1_\Upsilon(\eta(x)\eta(y)) \neq b^2_\Upsilon(\eta(x)\eta(y))$. The same argument can be made for edges $\alpha(x)\alpha(y) \in E(H)$. The second statement now follows directly from Definition 2 since the remainder graph $\hat{G}_\alpha$ contains exactly the edges with $b_G(xy) = b_H(\alpha(x)\alpha(y))$ and hence $b^1_\Upsilon(\eta(x)\eta(y)) = b^2_\Upsilon(\eta'(\alpha(x))\eta'(\alpha(y)))$. ◀

▶ **Lemma 17.** *Let $G \longrightarrow H$ be a balanced reaction and $\pi : U \to W$ with $U \subseteq V(G)$ and $W \subseteq V(H)$ be a partial AAM. Consider an extension α of π. Then, π is a good partial AAM with stable extension α, if and only if, $\hat{G}_\alpha = \hat{G}_\pi$ and $\hat{H}_\alpha = \hat{H}_\pi$.*

Proof. Suppose first that π is a good partial AAM with stable extension α. Thus, by definition we have $\Gamma^\perp(G, H, \alpha) = \Gamma^\perp(G[U], H[W], \pi)$. Consider then a reaction edge xy of G induced by α. By taking, in Lemma 10, the bijection η (resp. τ) to be the identity mapping on G, it follows that xy is an edge in $\Upsilon^\perp := \Upsilon^\perp(G, H, \alpha)$ with $b^1_{\Upsilon^\perp}(xy) \neq b^2_{\Upsilon^\perp}(xy)$, and therefore $xy \in \Gamma^\perp(G, H, \alpha)$.

Then xy is also a reaction edge of $\Upsilon^\perp(G[U], H[W], \pi)$, and by another application of Lemma 10 we conclude that xy is also a reaction edge of $G[U]$. By contraposition, moreover, this implies that if $xy \in E(\hat{G}_\pi)$, then $xy \in E(\hat{G}_\alpha)$, proving the contention $E(\hat{G}_\pi) \subseteq E(\hat{G}_\alpha)$, which together with Observation 16, yields $E(\hat{G}_\alpha) = E(\hat{G}_\pi)$ and thus $\hat{G}_\alpha = \hat{G}_\pi$. By similar arguments, for a reaction edge uv of H induced by α it follows that $\alpha^{-1}(u)\alpha^{-1}(v)$ is an edge in $\Upsilon^\perp$ with $b^1_{\Upsilon^\perp}(\alpha^{-1}(u)\alpha^{-1}(v)) \neq b^2_{\Upsilon^\perp}(\alpha^{-1}(u)\alpha^{-1}(v))$. Then $\alpha^{-1}(u)\alpha^{-1}(v)$ is also an edge in $\Gamma^\perp(G[U], H[W], \pi)$, implying that $\alpha^{-1}(u), \alpha^{-1}(v) \in U$. But α and π coincide for all vertices U, and so $\alpha^{-1}(u) = \pi^{-1}(u)$ and $\alpha^{-1}(v) = \pi^{-1}(u)$, that is, $\pi^{-1}(u)\pi^{-1}(v)$ is an edge in $\Upsilon^\perp(G[U], H[W], \pi)$ with $b^1_{\Upsilon^\perp}(\pi^{-1}(u)\pi^{-1}(v)) \neq b^2_{\Upsilon^\perp}(\pi^{-1}(u)\pi^{-1}(v))$ and thus, applying Lemma 10, we see that uv is also a reaction edge of H induced by π. By contraposition this also proves the inclusion $E(\hat{H}_\pi) \subseteq E(\hat{H}_\alpha)$. Observation 16 implies $\hat{H}_\alpha = \hat{H}_\pi$, proving the forward direction.

For the proof of the converse statement note that the hypotheses $\hat{G}_\alpha = \hat{G}_\pi$ and $\hat{H}_\alpha = \hat{H}_\pi$, imply that α and π induce the same reaction edges for each G and H, that is, **(R1)** xy is a reaction edge of G w.r.t α if and only if xy is a reaction edge of G w.r.t π and similarly **(R2)** uv is a reaction edge of H w.r.t α if and only if uv is a reaction edge of H w.r.t π.

This implies, by definition of the remainder graphs, that the vertices $x, y, \alpha^{-1}(u), \alpha^{-1}(v)$ are all in U, and thus α and π coincide for each of the two ends of every reaction edge of G and/or H. Thus, denoting $\Upsilon^\perp_\alpha := \Upsilon^\perp(G, H, \alpha)$ and $\Upsilon^\perp_\pi := \Upsilon^\perp(G[U], H[W], \pi)$, through Lemma 10 condition **(R1)** implies **(R1')** given $xy \in E(G)$, xy is a reaction edge of $\Upsilon^\perp_\alpha$ if and only if xy is a reaction edge of $\Upsilon^\perp_\pi$, i.e., xy is labeled by an ordered pair with different entries and, moreover, $b_{\Upsilon^\perp_\alpha}(xy) = b_{\Upsilon^\perp_\pi}(xy)$ since $x, y \in U$. Similarly, **(R2)** implies **(R2')** given $uv \in E(H)$, $\alpha^{-1}(u)\alpha^{-1}(v)$ is a reaction edge of $\Upsilon^\perp_\alpha$ if and only if $\alpha^{-1}(u)\alpha^{-1}(v)$ is a reaction edge of $\Upsilon^\perp_\pi$, and again $b_{\Upsilon^\perp_\alpha}(\alpha^{-1}(u)\alpha^{-1}(v)) = b_{\Upsilon^\perp_\pi}(\alpha^{-1}(u)\alpha^{-1}(v))$ since $\alpha^{-1}(u), \alpha^{-1}(v) \in U$. But every reaction edge xy of $\Upsilon^\perp_\alpha$, and $\Upsilon^\perp_\pi$, is labeled by a pair $b_{\Upsilon^\perp_\bullet}(xy) \in \{(a, \otimes), (\otimes, b), (c, d)\}$ with $a, b, c, d \neq \otimes$, that is: xy is only a reaction edge of G, $\alpha(x)\alpha(y) = \pi(x)\pi(y)$ is only a reaction edge of H, or both edges are respective reaction edges of G and H. Thus **(R1')** and **(R2')** are exhaustive cases, and since the edge labels are the same in both $\Upsilon^\perp_\alpha$ and $\Upsilon^\perp_\pi$, we have $xy \in E(\Gamma^\perp(G, H, \alpha))$ if and only if $xy \in E(\Gamma^\perp(G[U], H[W], \pi))$ with $b_{\Gamma^\perp_\alpha}(xy) = b_{\Gamma^\perp_\pi}(xy)$. Thus $\Gamma^\perp(G, H, \alpha) = \Gamma^\perp(G[U], H[W], \pi)$. Therefore π is good and has α as stable extension. ◀

▶ **Lemma 18.** *Let $G \longrightarrow H$ be a balanced reaction and $\pi : U \to W$ with $U \subseteq V(G)$ and $W \subseteq V(H)$ be a partial AAM. Consider an extension α of π. Then, α is an isomorphism from $\hat{G}_\pi$ to $\hat{H}_\pi$, if and only if, π is a good partial AAM with stable extension α.*

Proof. Suppose first that $\alpha \in \mathrm{Iso}(\hat{G}_\pi, \hat{H}_\pi)$. Consider any edge $xy \in E(\hat{G}_\pi) \subseteq E(G)$. Since α by assumption preserves adjacency, vertex labels and edge labels, we obtain $\alpha(x)\alpha(y) \in E(\hat{H}_\pi) \subseteq E(H)$. Since $\hat{G}_\pi$ and $\hat{H}_\pi$ take their edge labels from G and H, respectively, we also have $b_G(xy) = b_H(\alpha(x)\alpha(y))$, i.e., xy is not a reaction edge of G w.r.t α and thus $xy \in E(\hat{G}_\alpha)$. Similarly, for $uv \in E(\hat{H}_\pi) \subseteq E(H)$, we get $\alpha^{-1}(u)\alpha^{-1}(v) \in E(\hat{G}_\pi) \subseteq E(G)$ and $b_G(\alpha^{-1}(u)\alpha^{-1}(v)) = b_H(uv)$, from where $uv \in E(\hat{H}_\alpha)$. This shows $E(\hat{G}_\pi) \subseteq E(\hat{G}_\alpha)$ and $E(\hat{H}_\pi) \subseteq E(\hat{H}_\alpha)$. Then, Observation 16 implies that $\hat{G}_\pi = \hat{G}_\alpha$ and $\hat{H}_\pi = \hat{H}_\alpha$, and from Lemma 17 it follows, therefore, that π is good and α is a stable extension for π.

To prove the converse, suppose α is a stable extension for π. Then, by definition, α is also a full AAM for $G \longrightarrow H$, and by Lemma 3 we see that α is also an isomorphism from $\hat{G}_\alpha$ to $\hat{H}_\alpha$. But from Lemma 17 we should have $\hat{G}_\pi = \hat{G}_\alpha$ and $\hat{H}_\pi = \hat{H}_\alpha$, and therefore $\alpha \in \mathrm{Iso}(\hat{G}_\pi, \hat{H}_\pi)$. ◀

▶ **Theorem 20.** *Let π be a good partial AAM for a balanced reaction $G \longrightarrow H$ and let α and β be two stable extensions of π. Then $\alpha \equiv \beta$.*

Proof. Suppose π is the map $\pi : U \to W$ for subsets $U \subseteq V(G)$ to $W \subseteq V(H)$. Since α and β are extensions of π, by definition we have $\alpha(x) = \pi(x)$ and $\beta(x) = \pi(x)$ for all $x \in U \subseteq V(G)$. Moreover U contains all reacting vertices induced by π. In symbols we have $V(\Gamma^\perp(G[U], H[W], \pi)) \subseteq V(\Upsilon^\perp(G[U], H[W], \pi)) = V(G[U]) = U$ and thus $\alpha(x) = \beta(x)$ holds in particular for all $x \in V(\Gamma^\perp(G[U], H[W], \pi))$. By definition we get $V(\Gamma^\perp(G, H, \alpha)) = V(\Gamma^\perp(G[U], H[W], \pi)) = V(\Gamma^\perp(G, H, \beta))$, since α and β are stable extensions of π. Therefore $\Gamma^\perp(G[U], H[W], \pi)$ contains all reacting vertices of $\Upsilon^\perp(G, H, \alpha)$ and $\Upsilon^\perp(G, H, \beta)$. Lemma 10 implies that α and β coincide for all reacting vertices induced, in particular, by α for $G \longrightarrow H$, i.e., understood as reacting vertices $x \in V(G)$ and $\alpha(x) \in V(H)$. In addition, statement *(iii)* of Proposition 19 yields $\beta \in \mathrm{Iso}(\hat{G}_\pi, \hat{H}_\pi)$, while statement *(ii)* ensures that $\hat{G}_\pi = \hat{G}_\alpha$ and $\hat{H}_\pi = \hat{H}_\alpha$. Thus β is an isomorphism from the remainder graph $\hat{G}_\alpha$ to the remainder graph $\hat{H}_\alpha$. Proposition 5 therefore applies to α and β, and implies $\alpha \equiv \beta$. ◀

B Appendix: Additional Figures and Tables

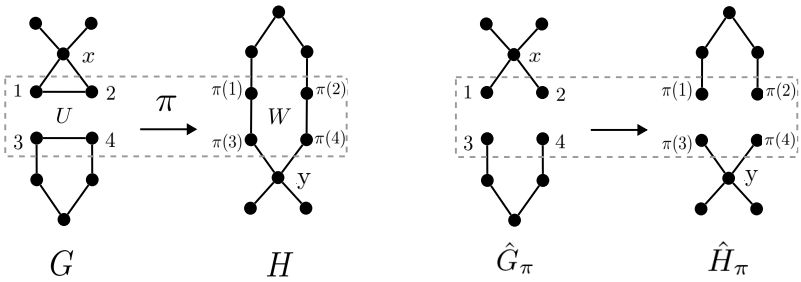

Figure 7 The remainder graphs $\hat{G}_\pi$ and $\hat{H}_\pi$ are isomorphic, and all isomorphisms between them must map the vertex $x \in V(\hat{G}_\pi)$ to $y \in V(\hat{H}_\pi)$, since these are the only vertices of degree 4 in them. Yet, the neighbors 1 and 2 of x in $\hat{G}_\pi$ are mapped by π to a different component than that of y in $\hat{H}_\pi$. Therefore, no isomorphism between these graphs extends the partial AAM π, i.e., π has no stable extensions.

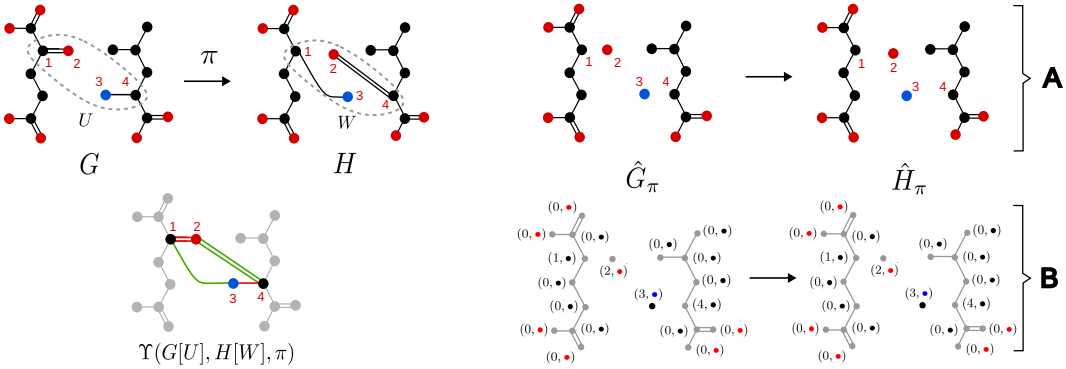

Figure 8 A good partial AAM π for the reaction in Figure 1 with full AAM α. **(A)** shows the remainder graphs w.r.t π, and **(B)** their respective computational representation used for Algorithm 2.

Table 1 Average time to complete one partial atom mapping (mean ± std in ms).

Trial	GM (ms)	RB_1 (ms)	RB_2 (ms)	ILP (ms)
1	3.39 ± 1.44	3.07 ± 1.54	2.92 ± 1.30	1153.77 ± 2983.54
2	3.33 ± 1.38	3.00 ± 1.45	2.87 ± 1.22	1142.43 ± 2978.49
3	3.36 ± 1.40	3.01 ± 1.46	2.87 ± 1.23	1139.71 ± 2979.89
4	3.35 ± 1.38	3.02 ± 1.49	2.87 ± 1.21	1135.71 ± 2967.40
5	3.34 ± 1.39	3.00 ± 1.45	2.88 ± 1.23	1148.98 ± 2991.35
Average	$\mathbf{3.36 \pm 1.39}$	$\mathbf{3.02 \pm 1.46}$	$\mathbf{2.88 \pm 1.22}$	$\mathbf{1144.12 \pm 2979.50}$

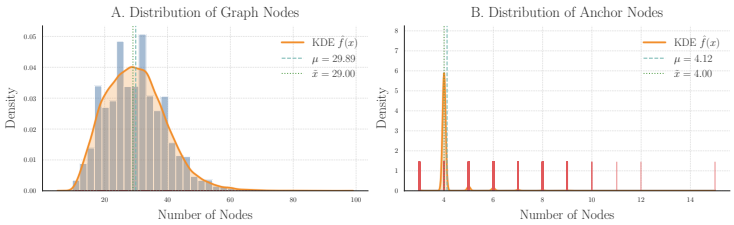

Figure 9 Histogram of number of vertices in molecular graphs (A) and in anchored graphs (B).

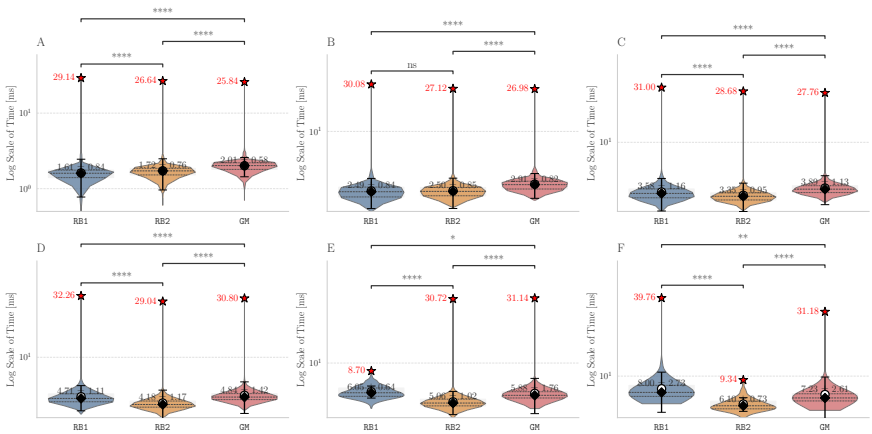

Figure 10 Running times of $\mathbf{RB_1}$, $\mathbf{RB_2}$, and **GM** as a function of ITS graph bins by number of vertices (top to bottom and left to right: less than 20, 20-30, 30-40, 40-50, 50-60, more than 60 vertices). $\mathbf{RB_1}$ is faster on average for small graphs (<30 vertices), while $\mathbf{RB_2}$ and **GM** exhibit better scalability on larger graphs, with a crossover at approximately 30 vertices.

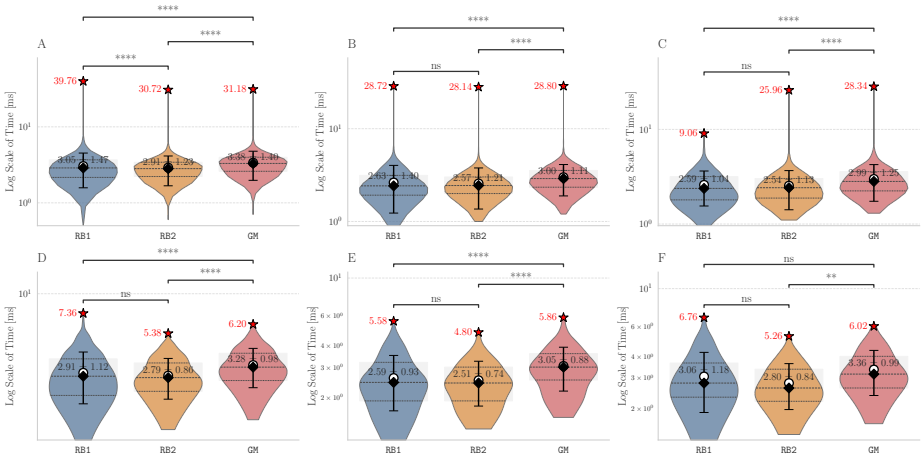

Figure 11 Running time comparison of $\mathbf{RB_1}$, $\mathbf{RB_2}$, and **GM** across six non-reacting vertex count-bins (top to bottom and left to right: less than 4, 4, 5, 6, 7, and at least 7 vertices).

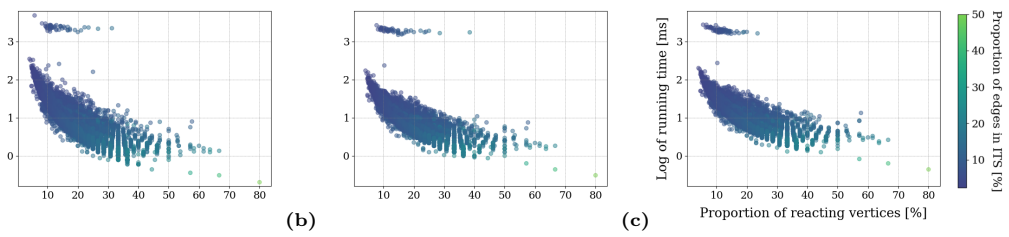

Figure 12 Scatter plots of running time against proportion of reacting vertices in real reactions for the graph based-methods **(a)** $\mathbf{RB_1}$, **(b)** $\mathbf{RB_2}$ and **(c)** **GM**. A smaller proportion of reacting vertices leads to a bigger search space for the `VF2`-based approaches and thus bigger running times.

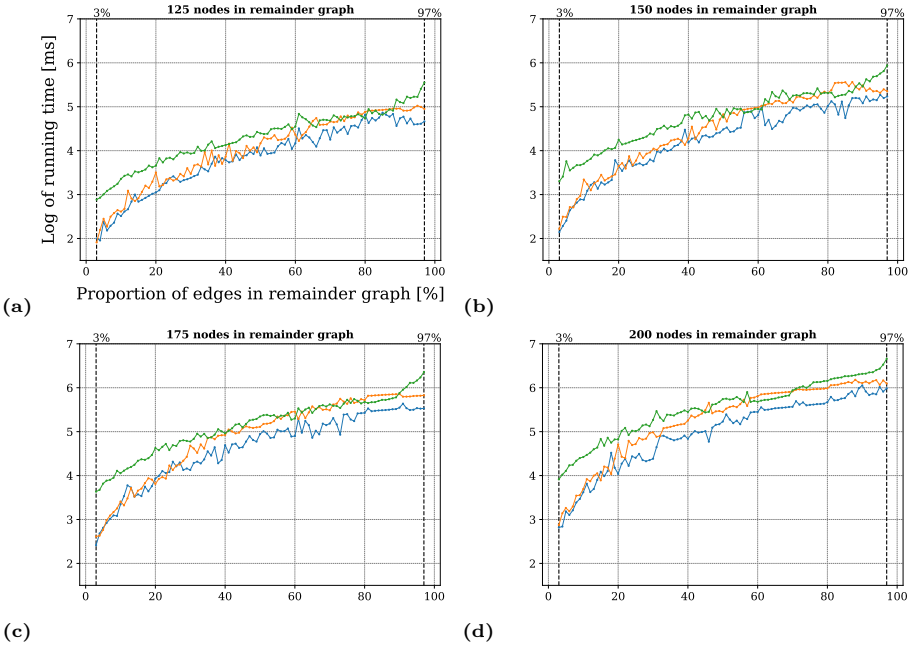

Figure 13 Average running times (10^k [ms] for each $k = 2, ..., 7$) for the methods $\mathbf{RB_1}$ (green), and the custom $\mathbf{RB_2}$ (orange) and **GM** (blue) over ITS graphs having different number of nodes: **(a)** 125, **(b)** 150, **(c)** 175 and **(d)** 200 nodes, and having a varying percentage of edges (3% to 97%).