arXiv:2305.01327v1 [cs.DM] 2 May 2023

采用网络约简识别异步布尔动力学中的吸引子

Elisa Tonello[1] 和 Lo¨ıc Paulev´e[2]

[1]德国柏林自由大学¨波尔多大学、CNRS、Bordeaux INP、LABRI、联合研究单位 5800，法国塔朗斯 F-33400[2]

2023年5月3日

摘要

吸引子的识别，即稳定状态和持续振荡的识别，是分析布尔模型和探索潜在变体的重要步骤。我们介绍一种搜索布尔网络异步循环吸引子的方法，以一种新颖的方式利用成熟的分量消去技术。计算简化网络的吸引子，可以识别数量有限的候选吸引子状态；随后，结合陷阱空间计算，采用可达性分析技术对这些状态进行筛选。一种整合了近期开发的布尔网络分析工具的实现方案，在生物模型和随机基准网络上进行了测试，结果显示该方案有望显著缩短运行时间。

1 引言

布尔网络被用作建模工具，用于整理知识并探索生物过程中可能出现的行为 [16, 23, 5]。根据描述物种间影响关系的逻辑，定义动力学以表达相关变量的演变。实现异步性的更新方案尤其受到关注，因为它们体现了内在随机性的一种形式 [18]。吸引子是基本结构，应能捕捉模型系统的演变归宿或稳定行为。近年来，识别异步动力学吸引子这一主题重新受到关注，并取得了多项进展。仅看过去两年， [15] 提出了一种结合基序检测、时间反演等不同技术的算法； [2] 开发了一种能够处理大型转移系统的符号方法，采用二元决策图表示布尔网络，并支持部分定义的更新函数 [1]； [21, 20] 介绍了基于反馈顶点集识别和模型检验可达性分析的方法。这些新技术使得处理复杂度不断提高的模型成为可能， [20]对此进行了总结。

本文旨在探究网络约简在研究异步状态转移图吸引子方面的效用。我们具体讨论一种广受欢迎的约简方法：如 [9, 10, 22] 所述，该方法通过迭代消去非自调控组分来实现。当选定一个变量进行消去时，会修改其靶标的调控函数，移除所选变量，并以该变量的调控函数替代。原始网络与通过这一约简过程得到的网络，其异步动力学和调控结构之间存在有益的关联；例如，两者具有相同的稳态，并且约简网络的吸引子数量不会少于原始网络。网络约简的性质已被用于吸引子识别，特别是在 [15]，中，约简方法在特定情况下与其他方法结合使用。本文将展示，将变量消去系统地作为网络分析的第一步，能够带来显著益处。

变量消去作为识别异步吸引子的有效工具，其依据在于以下性质。对于布尔网络 f的每个吸引子 $\mathcal{A}$ ，如果˜是通过迭代消去某些非自调控组分得到的，

f 则在˜的一个吸引子中，至少存在一个状态，可以通过逆向追踪约简过程，从该状态重构出 $\mathcal{A}$ 中的一个状态。˜这一性质使我们能够根据 f的吸引子，识别出一组有限的“候选”状态，并覆盖 f 的所有吸引子（第2.2节）。随后，需要对这些候选状态进行筛选，以滤除不属于 f的吸引子的状态。通过计算极小陷阱空间 [19, 6] ，我们可以首先检查某个候选状态是否属于一个包含在极小陷阱空间中的吸引子，以及该吸引子是否是该极小陷阱空间中唯一包含的吸引子。如果一个极小陷阱空间中存在多个候选吸引子，或者存在位于极小陷阱空间之外的候选状态，则需要进行其他检查，以确定它们是否确实属于某个吸引子。计算开销更高的技术（例如模型检验）可用于此步骤 [12, 21] ，以检查先前识别出的吸引子，或不包含该候选状态的陷阱空间，是否能够从该候选状态到达。

我们通过使用 colomoto 的 minibn [8] 实现变量消去，在生物网络和随机网络上测试了这些思路。随后使用 AEON [1] 和 mtsNFVS[20], 研究约简网络，以识别候选吸引子状态。我们使用 trappist 查找极小陷阱空间 [19] ，并据此对候选状态进行初步筛选。对于剩余候选状态的检查，我们使用了 mtsNFVS 的可达性分析软件 [20]。总体而言，我们发现计算时间有很大的缩减潜力（第 3.2 节）。此外，我们还观察到，整个流程中计算开销最大的任务——可达性检查——只需偶尔执行。事实上，尽管变量消去可能导致吸引子数量增加，但这种情况似乎相当少见，这意味着候选状态的数量通常非常接近吸引子的数量，甚至与之相等。因此，处理这类情况所需的工作量也有限。最后一节（第 4 节）中，我们将探讨一些可行的方向，以进一步改进文中讨论的技术。

2 背景

2.1 布尔网络

我们将 V 称为所关注的变量或物种集合，并设定 $n = |V|$。布尔网络由映射 $f = (f_1, \ldots, f_n)$定义：$2^n \to 2^n$。集合 2^n 称为该布尔网络的状态或构型集合。对于 $i = 1$、...、n 和 $x \in 2^n$，我们用 $\bar{x}^i$ 表示这样一个状态：除 i外，在所有分量上都与 x 一致。子空间是状态空间的一个子集，由一组分量取值相同的所有状态组成；这组分量称为该子空间的固定变量。

布尔函数 f_i、 $i = 1$、...、n有时称为变量 i的更新函数。 f的影响图 $G(f)$ 是一个带符号的多重有向图，其节点集合为 $\{1, \ldots, n\}$，边刻画了更新函数对各变量的依赖关系：当且仅当存在某个 $x \in 2^n$，使得 $f_j(\bar{x}^i) \neq f_j(x)$ 和 $s = (f_j(\bar{x}^i) - f_j(x))(\bar{x}^i_i - x_i)$ 时， $G(f)$ 中存在一条带有符号 s 的边 (i, j)。

与 f 相关的状态转移图或动力学，是一个由顶点集 2^n 和边集（也称为转移）构成的图，具体取决于所选语义。本文研究异步动力学，这是一种非确定性动力学，只包含状态之间相差一个分量的转移。具体而言，当且仅当 $f_i(x) \neq x_i$ 时，才存在一条从状态 x 指向状态 $\bar{x}^i$ 的边。我们用 $\Gamma(f)$ 表示 f 的异步状态转移图。

动力学的陷阱集是状态空间的一个子集，其中不存在任何向外转移。吸引子是在包含关系下极小的陷阱集。吸引子可以只包含一个状态，称为稳态或不动点；也可以包含多个状态，此时称为循环吸引子或复杂吸引子。

在异步或同步动力学下，判定给定状态是否属于某个吸引子是一个 PSPACE 完全问题 [13]。在实践中，计算通常依赖于（部分地、以符号方式）计算状态转移图；与同步情形相比，异步情形下的状态转移图可能复杂得多。

陷阱空间 是指不包含任何向外转移的子空间。它们可以被视为 g】【。

en- eralizations of s me

稳态，因为它们是动力学限制在 so 上的不动点。

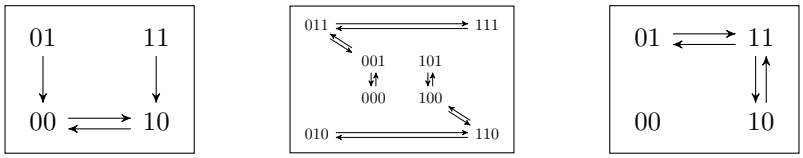


图1：例2.1中布尔网络 f、 g 和 h 的异步状态转移图。Γ（f）具有一个极小唯一吸引子， Γ（g）具有两个极小非唯一吸引子，而 Γ（h）具有一个极小唯一吸引子和一个非极小吸引子。

分量。与吸引子识别相比，陷阱空间及相关性质的计算要容易处理得多，因为其理论复杂度大幅降低 [6]。极小陷阱空间尤其值得关注，因为每个极小陷阱空间都至少包含一个吸引子。通常，作为生物模型的布尔网络的极小陷阱空间只包含一个吸引子 [3], ，这意味着，例如在可达性分析中，可以用它们替代所包含的吸引子，作为可达性靶标。尽管已有研究识别出一些能够保证获得“良好”吸引子景观的调控结构性质（吸引子仅存在于极小陷阱空间中，且极小陷阱空间中的吸引子具有唯一性） [14, 11], ，但这些性质的适用范围有限。一般而言，要确定布尔网络是否在某个极小陷阱空间中存在多个吸引子，或是否存在极小陷阱空间之外的吸引子（ [15]所称的“避开基序”吸引子），仍然是一项艰巨的任务。

基于这些考虑，我们可以将布尔异步动力学中的吸引子分为以下四类，这将在我们后续的讨论中发挥作用：

（A）稳态：这些状态“容易”找到。由于我们的算法会计算所有极小陷阱空间，因此我们将稳态确定为所有变量均为固定变量的极小陷阱空间。（B）极小唯一吸引子：我们采用 [3] 中引入的术语，指仅包含一个吸引子的极小陷阱空间。为方便起见，我们也将该术语用于包含在这些极小陷阱空间中的吸引子。（C）极小非唯一吸引子：这些吸引子包含在某个极小陷阱空间中，但并非该陷阱空间中包含的唯一吸引子。为了研究它们是否存在，我们可以将搜索空间限制在该极小陷阱空间内。（D）非极小或回避基序的 [15]：这是最难检测或排除其存在的一类。

例2.1. 这些映射的异步状态转移图

$$
\begin{aligned}
f(x_1, x_2) &= (x_1x_2 \vee \bar{x}_1\bar{x}_2, 0),\\
g(x_1, x_2, x_3) &= (x_2\bar{x}_1 \vee x_1\bar{x}_2, x_1(x_2x_3 \vee \bar{x}_2\bar{x}_3) \vee \bar{x}_1(x_2\bar{x}_3 \vee x_3\bar{x}_2), x_2x_3 \vee \bar{x}_2\bar{x}_3),\\
h(x_1, x_2) &= (x_1\bar{x}_2 \vee \bar{x}_1x_2, x_1\bar{x}_2 \vee \bar{x}_1x_2)
\end{aligned}
$$

如图1所示。

$\Gamma(f)$有一个周期吸引子和一个最小陷阱空间 $\{00, 10\}$。

$\Gamma(g)$有两个吸引子，它们无法唯一对应，因为它们位于同一个最小陷阱空间（整个空间）中。

$\Gamma(h)$有两个吸引子：一个稳态00和一个非极小吸引子 $\{01, 10, 11\}$。该稳态是唯一的最小陷阱空间。

下一节我们将回顾如何从布尔网络中移除分量，并讨论这种做法如何有助于识别吸引子。

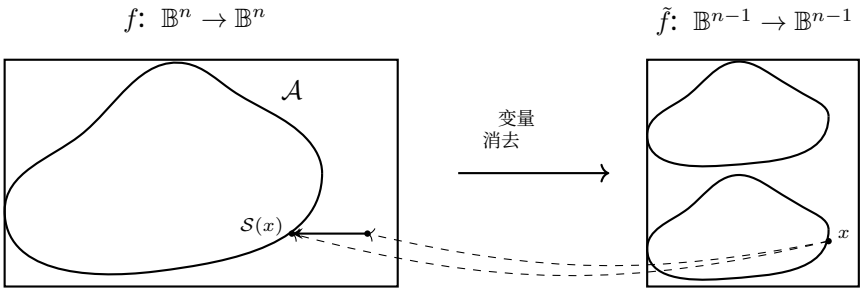


图 2：该方法的基本思路：约简网络吸引子中的状态 f 可用于寻找 f 吸引子中的候选状态。

2.2 约简

一种常用的布尔网络异步动力学约简方法是迭代消除未自调控的变量 [9, 10, 22]。近期，这种方法已扩展到负自调控组分 [17]。虽然本文的所有观察结果都可以相应地扩展到负自调控组分，但为求简洁，我们只讨论标准情形。

假设 $G(f)$ 中有一个没有环的节点。不失一般性，我们可以假设该节点为 n。我们记 $\pi\colon 2^n \to 2^{n-1}$ 为前 $n-1$ 个变量上的投影。

根据 $G(f)$ 的定义，对于所有 $x \in 2^{n-1}$，都有 $f_n(x,0) = f_n(x,1)$。特别地，f 的状态转移图恰好包含从 $(x,0)$ 到 $(x,1)$ 或从 $(x,1)$ 到 $(x,0)$ 的转移中的一个。因此，我们可以定义一个映射 $\mathcal{S}^n$，将一个“约简”状态关联到更大的状态空间中的一个状态，如下所示：

$$\mathcal{S}^n\colon 2^{n-1} \to 2^n$$
$$x \mapsto (x, f_n(x,0)) = (x, f_n(x,1)).$$

通过消去分量 n 得到的 f 的约简 $\tilde{f}$ 随后定义为

$$\tilde{f} = (f_1 \circ \mathcal{S}^n, \ldots, f_{n-1} \circ \mathcal{S}^n)\colon 2^{n-1} \to 2^{n-1}.$$

给定 $y \in 2^n$，状态 $\mathcal{S}^n(\pi(y))$ 可以视为状态对 $(\pi(y),0)$、$(\pi(y),1)$ 的“代表状态”，也可以视为“在约简中存活下来”的状态，因为离开状态 $\mathcal{S}^n(\pi(y))$ 的所有转移，在 $\Gamma(\tilde{f})$ 中都有对应的转移；而离开状态 $\mathcal{S}^n(\pi(y))^n$ 的转移则不一定能得到保留 [10]。

文献 [10] 证明了 f 和 $\tilde{f}$ 具有相同数量的稳态，并且 f 的吸引子数量大于或等于 $\tilde{f}$ 的吸引子数量。以下结果是我们方法的基础，它是由文献 [10] 中证明的性质直接推得的。

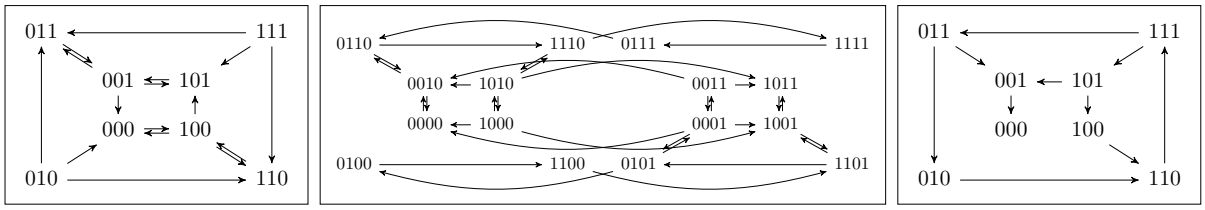
定理 2.2. 若 $\mathcal{A}$ 是 f 的一个吸引子，则 $\tilde{f}$ 在 $\pi(\mathcal{A})$ 中至少存在一个吸引子，并且对于 $\tilde{f}$ 的吸引子中包含的每个 $x \in \pi(\mathcal{A})$，$\mathcal{S}^n(x) \in \mathcal{A}$。

证明。第一部分源于以下观察：如果 B 是 $\Gamma(f)$ 的陷阱集，那么 $\pi(B)$ 是 $\Gamma(\tilde{f})$ 的一个陷阱集。

给定包含在 $\pi(\mathcal{A})$ 中的 $\tilde{f}$ 的某个吸引子内的状态 x，根据定义，$\mathcal{S}^n(x)$ 或 $\mathcal{S}^n(x)^n$ 位于 $\mathcal{A}$ 中；又因为 $\mathcal{S}^n(x)$ 可从 $\mathcal{S}^n(x)^n$ 到达，所以 $\mathcal{S}^n(x)$ 必定位于 $\mathcal{A}$ 中。 □

该定理给出了以下性质：如果 f 存在一个吸引子，那么我们可以通过求出约简版本 $\tilde{f}$ 的吸引子，从每个吸引子中抽取一个状态，并使用 $\mathcal{S}^n$ 将这些状态“提升”到原始状态空间中，从而识别出该吸引子中的一个状态（图 2）。

在更详细地讨论如何利用约简网络识别原始网络的吸引子之前，我们先来看几个例子，以了解变量消去可能会对吸引子产生什么影响。



ˆˆ 图3：例2.4至例2.6中的布尔网络 f、ˆg, h 的异步状态转移图。

例2.3。考虑例2.1中的映射 f。由于第二个变量不是自调控变量，因此可以使用本节介绍的消元法将其移除。常数值 0 替代了变量 x_2，出现在变量 x_1 的更新函数中，从而得到约简网络 $\tilde{f}(x_1) = \bar{x}_1$。我们有 $\mathcal{S}^2(0) = \mathcal{S}^2(00)$，并且 $\mathcal{S}^2(1) = \mathcal{S}^2(10)$。这个简单的例子足以说明：给定约简网络吸引子中的一个状态 x（此处以 $x = 0$ 为例），要找回原始网络吸引子中的一个状态，不能从原像 $\pi^{-1}(x)$ 中任意选取一个状态（状态 01 不适用），而是需要应用函数 $\mathcal{S}^2$。

例2.4。若 $\mathcal{A}$ 是一个布尔网络的吸引子，且 x 是 $\mathcal{A}$ 中的一个状态，那么 $\pi(x)$ 是否必然属于约简网络的某个吸引子？答案是否定的。例如，考虑

$$\hat{f}(x_1, x_2, x_3) = (\bar{x}_1\bar{x}_2 \vee \bar{x}_1\bar{x}_3 \vee x_2\bar{x}_3, x_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2x_3, x_1\bar{x}_2 \vee \bar{x}_1x_2).$$

移除第三个分量后，我们得到例2.1中的函数 f。状态 011 属于 f 的唯一吸引子，而 $\pi(011) = 01$ 不属于 f 的任何吸引子。

因此，一般而言，我们无法通过提升约简网络吸引子中的状态，找回布尔网络吸引子中的所有状态。我们只能找到其中一些状态；如有需要，可以从这些状态转移到该吸引子。

例2.5。布尔网络

$$\hat{g}(x_1, x_2, x_3, x_4) = (x_2\bar{x}_4 \vee \bar{x}_2x_4, x_4(x_2x_3 \vee \bar{x}_2\bar{x}_3) \vee \bar{x}_4(x_2\bar{x}_3 \vee x_3\bar{x}_2), x_2x_3 \vee \bar{x}_2\bar{x}_3, x_1)$$

有一个周期吸引子，它覆盖整个状态空间（见图3）。移除变量 x_4 后，我们得到例2.1中的网络 g，它有两个吸引子。

例2.6。布尔网络

$$\hat{h}(x_1, x_2, x_3) = (x_1\bar{x}_3 \vee x_2\bar{x}_3, x_1\bar{x}_3 \vee x_2\bar{x}_3, x_1x_2)$$

具有唯一吸引子，即不动点 000。消去变量 x_3 后，我们得到例2.1中的布尔网络 h，它有两个吸引子。

˜假设 f：$2^m \to 2^m$ 是通过从 f 迭代消去变量 n、$n-1$、.....、$m+1$ 得到的，并且 $\mathcal{A}_1$、.....、$\mathcal{A}_M$ 是 $\tilde{f}$ 的吸引子。从每个吸引子中取一个状态 x^1、.....、$x^M \in 2^m$。我们可以通过应用映射 $\mathcal{S} = \mathcal{S}^n \circ \mathcal{S}^{n-1} \circ \cdots \circ \mathcal{S}^{m+1}$，重建 2^n 中对应的状态。如何确定这些状态是否各自属于 f 的某个吸引子，以及 f 有多少个吸引子?

我们可以计算 f 的极小陷阱空间，并作如下观察。给定一组候选状态 $C = \{\mathcal{S}(x^1)$ .....，$\mathcal{S}(x^M)\}$:

˜(a) 如果 x 是 f 的稳态，那么 $\mathcal{S}(x)$ 就是 f 的稳态（并且可以用这种方式计算出 f 的所有稳态）。(b) 如果 $\mathcal{S}(x)$ 属于 f 的一个最小陷阱空间 t，且 C 中只有一个状态属于 t，那么 $\mathcal{S}(x)$ 就属于 f 的一个极小单值吸引子。我们将这些状态称为单一归属状态。

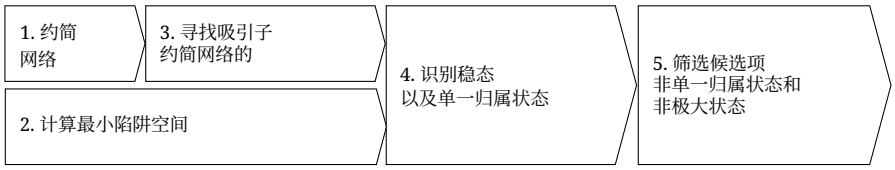


图 4：算法的主要步骤。

(c) 如果 $\mathcal{S}(x)$ 包含在 f 的一个最小陷阱空间 t 中，且 C 中不止一个状态包含在这个最小陷阱空间中，那么我们需要研究 t 中的动力学，以明确包含在 t 中的每个候选状态是否属于某个吸引子，以及 t 中的候选状态是否属于不同的吸引子。我们可以将这些状态称为候选非单一归属状态。 C 中包含在 t 中的状态数量，为包含在 t 中的吸引子数量提供了一个上限。

(d) 如果 $\mathcal{S}(x)$ 不包含于任何最小陷阱空间中，那么 $\mathcal{S}(x)$ 可能属于 f 的某个非极小吸引子。要确定是否确实如此，并找出非极小吸引子的数量，我们还需要进行额外的工作。我们将这些状态称为候选非极小状态。

请注意，如果考虑消去负自调控组分 [17]，则需要对这些观察结果进行略微调整。现在，我们可以给出基于分量消去的吸引子识别方法的描述。

3 方法

基于对布尔网络约简版本的吸引子与原始网络吸引子之间关系的分析，我们提出了以下流程（见图 4）。

首先通过消去变量 f 对网络进行约简（步骤 1），并且可以并行地寻找布尔网络 f 的极小陷阱空间（步骤 2）。然后，为约简后的 ˜网络 f 的每个不是稳态的吸引子识别一个状态（步骤 3），从而获得 f 的候选吸引子状态集合。

步骤 4 和步骤 5 处理这些候选吸引子状态的筛选。步骤 4 比较简单：我们检查每个候选状态是否包含于某个极小陷阱空间中。如果一个极小陷阱空间只包含一个候选状态，那么该状态就是唯一状态，我们也就识别出了 f 的唯一吸引子（上一节的(b)点）。

步骤5处理剩余的候选状态。这些状态可能是同一最小陷阱空间中包含的多个状态（上一节的（c）点），也可能是不属于任何最小陷阱空间的状态（上一节的（d）点）。在这两种情况下，我们都必须以某种方式研究状态转移图，以判断候选状态是否属于某个吸引子，以及存在多少个吸引子（某些候选状态可能属于同一个吸引子，而该吸引子在约简过程中被“拆分”，如例2.5所示；也可能属于由约简过程创建的吸引子，如例2.6所示）。此步骤可以采用模型检验方法（见下一节），也可以结合其他技术（见最后一节的讨论）。

3.1 实现

3.1.1 软件

在我们的实现中，我们使用 Python 库 colomoto 的 minibn [8] 来计算网络约简（图4中的步骤1）。在 minibn 中，布尔网络的局部函数以命题逻辑公式表示，并采用通常的布尔代数。给定一个要约简的分量 i，我们的实现只需将 x_i 替换为 f_i 的表达式，并将其代入该分量的所有靶标中。随后执行基本的布尔表达式化简，这可能会导致变量消去。

对于步骤 2，我们使用 trappist 计算极小陷阱空间 [19]。由于 trappist 依赖于布尔网络的 Petri网表示，因此这一步需要将每个局部布尔

函数 f_i 转换为析取范式中的两个表达式（一个用于 $f_i(x) = 1$，另一个用于 $f_i(x) = 0$），并由此推导出 Petri网转移。

对于第 3 步，即识别约简网络的吸引子，我们考察了近期开发的两种方法 AEON [1] 和 mtsNFVS [20]。在 [20], 的分析中，这两种方法被证明是目前最快的方法，同时采用了截然不同的处理方式。这两种方法的输出也颇为不同：例如，AEON 可以提供 mtsNFVS 无法直接提供的吸引子信息（基数、状态列表），还可以同时处理网络族、执行后处理控制任务等。这里我们只关注这些方法在识别吸引子数量及其性质（稳态或周期性）方面的表现，并主要探究约简是否有助于此。

各部分在 Python 中组合起来，并执行第 4 步。对于第 5 步，我们运行 Java 工具 mtsNFVS.jar，它是 mtsNFVS 的一部分。该工具首先使用 Pint [12] 通过静态分析进行检查，随后进行有界模型检验和精确模型检测 [21]。不过，我们注意到 mtsNFVS（未进行约简时）在若干实例中会产生错误结果（项目代码库中提供了一些示例）。因此，对 mtsNFVS 得出的结果应谨慎看待。对于我们考察的生物网络，我们确认所有方法找到的稳态和周期吸引子的数量相同。正如我们将在结果部分讨论的，可达性分析步骤对整个流程的影响有限，因为非极小候选很少出现，而且在任何生物网络或随机测试中都没有发现非唯一候选。

3.1.2 消去顺序及其他考量

待消去变量的顺序选择会影响约简网络以及运行时间。当消去一个具有 r 个调节因子和 t 个靶标的变量时，在最坏情况下，$r + t$ 个调控关系可能会被 $r \cdot t$ 个调控关系所替代。因此，在每一步中，我们选择使该乘积最小的变量进行消去。更系统地研究消去顺序如何影响运行时间，或许能够识别出更优的消去顺序。例如，可以尝试优先选择对陷阱空间数量影响最小的消去顺序。

使用变量消去时，另一个关键问题是决定何时停止迭代过程。遗憾的是，这个问题似乎没有简单的答案。在某些情况下，经过一段时间的消去过程后，约简可能会引入复杂的更新函数，从而减慢后续处理步骤。此外，尽管对于我们测试的方法而言，约简作为预处理步骤似乎是有益的，但我们将看到，不同程度的约简可能会以不同方式有利于不同的方法。在我们的实现中，我们考虑了两个用于停止约简过程的可能参数。一个参数规定约简网络的最小规模，即节点数量（stop at）。另一个参数考察每个节点的调节因子数量与靶标数量的最小乘积（即用于选择待消去变量的参数），并为该值设定上限（max product）。

我们还研究了布尔表达式简化对运行时间的影响。我们发现，尽管简化的开销较大，但它能加快消去过程。下一节将讨论消去步骤对生物网络和随机网络运行时间的影响。

3.2 结果

生物网络模型的实验在一台台式计算机上运行，该计算机配备 Intel(R) Core(TM) i7-8700 处理器、32GB 随机存取存储器，操作系统为 Debian GNU/Linux 11。随机网络的实验在高性能计算集群节点上运行，这些节点配备 AMD(R) Zen2 EPYC 7702 @ 2 GHz 处理器、1TB 随机存取存储器，操作系统为 CentOS Linux。代码可在 https://github.com/etonello/attractors-with-reduction 获取。

3.2.1 生物模型

我们使用 AEON [1] 和 mtsNFVS [20] 运行实现，以查找约简网络的吸引子，并将运行时间与 AEON 和 mtsNFVS 在未约简的网络上运行时进行比较

模型	文件名	文件来源	n. 节点	n. 边	稳态数 状态	名词：周期 吸引子
MAPK	Grieco 丝裂原活化蛋白激酶	PyBoolNet	53	108	12	6
IL-6	白细胞介素-6信号传导	mtsNFVS	55	95	28672	4096
EMT	Selvaggio EMT	PyBoolNet	56	159	1452	0
T-LGL	TLGLSurvival	mtsNFVS	58	193	172	146
CACC	结肠炎相关.....	mtsNFVS	66	144	2	8
AD	一个模型	mtsNFVS	74	198	0	2
AGS	id-148-AGS-CELL-FATE-.....	biodivine	83	193	1	0
CC	细胞周期 2019	mtsNFVS	87	467	8	0
SP	id-192-SEGMENT-POLA...	biodivine	102	432	65	0
SIPC	前列腺信号传导...	mtsNFVS	116	428	2460	300
DSP	id-210-药物协同作用-...	biodivine	144	367	0	1
C3.0	CASCADE3	mtsNFVS	176	449	0	1
EP	id-211-上皮衍生...	biodivine	183	602	0	1

表1：所考虑的 bnet模型的来源信息、规模以及稳态和周期吸引子的数量。

	(1) 最大乘积=20			(2) 最大乘积=50			(3) 最大乘积=100		
模型	节点数	边数	time	节点数	边数	time	节点数	边数	time
MAPK	10	41	0.1	10	41	0.0	10	41	0.1
IL-6	17	28	0.0	17	28	0.0	17	28	0.0
EMT	19	94	0.1	17	130	0.1	17	130	0.1
T-LGL	21	91	0.0	18	111	0.0	18	111	0.0
CACC	14	81	0.0	11	56	0.0	11	56	0.0
AD	11	93	0.0	10	97	0.0	10	97	0.0
AGS	2	4	0.0	2	4	0.0	2	4	0.0
CC	38	415	0.2	35	490	0.3	29	669	23.9
SP	35	234	0.0	33	273	0.1	32	357	0.1
SIPC	41	390	0.3	32	465	0.6	28	522	8.7
DSP	14	108	0.0	10	71	0.0	10	71	0.0
C3.0	23	191	0.0	14	175	0.1	13	193	0.1
EP	33	322	0.1	25	365	0.1	21	316	0.1

表2：表1所列生物模型的约简场景。最大乘积= k表示当所有节点的调节因子数量与靶标数量的乘积均高于 k 时，消去过程结束。还设置了当节点数量达到10时停止消去。

。我们考察了工具仓库中提供的 [20] 中收录的七个模型，以及从 PyBoolNet 仓库 [4] 和 biodivine-boolean-models 仓库（https://github.com/sybila/biodivine-boolean-models）中提取的其他模型。表1列出了各模型的来源信息以及网络规模。有关各模型的详细参考资料，请读者参阅相应来源；这些资料说明了模型是否以及如何根据其原始来源进行修改。

正如上一节所述，不同程度的约简会对消除过程以及吸引子识别过程其余部分的运行时间产生不同影响。这里报告三种场景下测得的运行时间：我们将上一节所述的参数 max product（所有节点的输入和输出调控关系乘积的最小值所能接受的最大值）分别设为三个水平：场景（1）为 20，场景（2）为 50，场景（3）为 100。三种情况下，我们都将约简网络的最小规模设为 10。在某些情况下，这些参数会使网络的节点数少于 10，因为可能会生成常量变量

	AEON 运行时间				mtsNFVS 运行时间			
模型	无红色。	(1)	(2)	(3)	无红色。	(1)	(2)	(3)
MAPK	5.7	0.3	0.3	0.3	28.9 ±5.7 （3 未完成）	0.7	0.7	0.7
IL-6	774.6	6.0	6.0	6.0	14.8 ±1.8	7.4	7.4	7.4
EMT	25.6	0.8	0.7	0.7	DNF	1.3	1.4	1.4
T-LGL	17.5	0.9	0.9	1.1	2.2	1.8	1.9	2.7
CACC	9.3	0.3	0.3	0.3	0.5	0.8	0.7	0.7
AD	361.9	0.3	0.4	0.4	0.7	1.0	0.8	0.8
AGS	1.7	0.3	0.3	0.3	0.6	0.7	0.7	0.7
CC	DNF	27.0	11.1	26.9	8.2 ±3.5	2.2	6.0	39.4
SP	DNF	0.9	0.8	0.8	DNF	1.0	1.1	1.4
SIPC	DNF	28.2	6.9	14.4	1664.7 ±506.3	47.8 ±5.6	53.7 ±7.4	138.8
DSP	DNF	0.4	0.4	0.4	2.3	0.8	0.7	0.7
C3.0	DNF	0.5	0.4	0.4	2.1	1.0	1.0	0.9
EP	DNF	1.5	0.6	0.5	62.7 ±64.2	2.5	2.4	1.9

表3：AEON [1] 和 mtsNFVS [20] 在未进行约简，以及表2所述三种约简场景下的运行时间。三种约简场景下的运行时间包含网络约简所需的时间。mtsNFVS 的运行时间取五次迭代的平均值；如果标准差大于 1 秒，则一并报告。“未完成”表示处理未能在一小时内完成。

通过消去过程，这些节点总是会被消去。表2展示了三种情景下得到的约简网络中的节点数量和带符号调控关系数量，以及约简过程所需的时间。除 CC 和 SIPC 这两个网络外，约简时间均不到一秒；在这两个网络中，约简时间分别达到 24 秒和 9 秒。

表3展示了两种方法在不同情景下（不进行约简、约简情景（1）、（2）和（3））的运行时间。mtsNFVS 算法依赖一些随机选择，因此其运行时间会有所波动，有时波动显著。我们报告了五次迭代中得到的最短和最长运行时间。我们将超时时间设为一小时，“DNF”表示处理未能在此时间内完成。

结果清楚地表明，先约简网络再处理吸引子识别问题，可以显著加快处理速度。我们还可以观察到，更激进的约简并不一定会缩短吸引子识别所需的时间。在 CC 和 SIPC 这两个网络中，两种吸引子识别方法都呈现出这一现象。观察这两个网络的数据还可以发现，使处理时间最短的约简程度对于 AEON 和 mtsNFVS 并不相同。具体而言，mtsNFVS 似乎在较保守的约简程度下效果更好；在一个案例中（网络 CC，情景（3）），约简过程所需的时间甚至超过了 mtsNFVS 不进行约简时的运行时间。

值得注意的是，仅在一个网络（TLGL）中发现了非极小候选，且没有网络出现非唯一候选，这意味着约简对整体吸引子构型的影响有限。考察随机生成的网络或许有助于厘清这些行为是否为生物模型所特有，还是更为普遍。

3.2.2 随机基准测试

与 [20], 类似，我们通过调用 R 包 BoolNet ([7]) 中的 generateRandomNKNetwork 来生成随机网络，并将每个变量的调节因子数固定为 $K = 2$ 。我们考虑包含$n = 100k$个节点的网络，其中 $k = 1$、.....、5，并为每种规模创建 10 个网络。

我们测试了约简过程的多种停止条件，但未能找到适用于所有情况、都能实现最优时间的通用规则。为了展示可能结果的范围，我们报告了三种约简场景下的运行时间，其中参数最大乘积分别设为 $\frac{n}{2}$ 、 n 和 $2n$,

	(1) 最大乘积=n/2			(2) 最大乘积=n			(3) 最大乘积=2n		
模型 size	平均值 节点数	平均值 边数	平均值 time	平均值 节点数	平均值 边数	平均值 time	平均值 节点数	平均值 边数	平均值 time
100	14.0	219.9	0.1	12.3	213.0	0.5	11.9	199.5	0.6
200	24.0	621.9	0.5	21.2	618.0	1.9	20.7	596.3	5.8
300	37.6	1418.5	2.1	34.4	1527.5	7.8	30.0	1240.0	19.7
400	49.4	2266.0	2.6	42.5	2471.0	23.1	41.1	2498.3	55.3
500	61.5	3489.8	18.4	55.3	3985.7	18.8	55.5	4683.5	286.5

表4：在 10 个随机网络（使用 BoolNet [7],设置 $K = 2$生成）上，三种约简场景的统计数据。最大乘积= k 表示当所有节点的调节因子数量与靶标数量的乘积均高于 k 时，消去过程结束。消去过程也设定为在节点数达到原始网络节点数的十分之一时停止。

而 stop at 设为 $\frac{n}{10}$。这三种场景的详细信息见表4。表5列出了在一小时超时限制内完成处理的网络的平均运行时间，以及未能及时完成处理的网络数量。

我们再次观察到，采用约简方法后，能够在给定超时范围内成功分析的网络数量显著增加。未进行约简时，AEON 能够处理 10 个规模为 100 的网络中的 4 个，但无法处理规模更大的网络。采用约简并使用 AEON 后，所有规模为 200 的网络、10 个规模为 300 的网络中的 8 个，以及一个规模为 400 的网络均可处理。规模为 100 的网络运行时间仅需几秒，而规模更大的网络运行时间则差异显著。

不进行约简时，mtsNFVS 工具能够处理规模为 100 的所有网络，以及规模为 200 和 300 的部分网络。进行约简后，则能够处理规模高达 500 的网络。然而，正如我们在第 3.1 节指出的，mtsNFVS 生成的结果可能需要进一步验证，因为在测试网络中发现了若干错误。

尽管如此，通过观察使用 AEON 得到的结果，我们可以注意到，需要通过可达性分析处理的候选状态数量仍然非常有限。对于采用约简并使用 AEON 处理的网络，仅识别出五个非最小候选状态，且没有非唯一候选状态。这表明，即使对于随机网络，非最小和非唯一吸引子也可能是罕见现象。

约简使网络规模得以大幅缩小，但表4中的结果说明，不同数量的约简步骤可能对不同网络产生不同影响。在情形（3）中，消去变量数量最多，只有此时部分网络才能在限定时间内完成处理；而对于其他网络，移除较少变量时的处理时间更短。

4 讨论

我们研究了约简如何加快吸引子识别过程。我们观察到，约简通常会使这一过程更加容易，但其影响取决于所采用的吸引子检测方法。消去变量虽然缩小了网络规模，却增加了影响图和更新函数的复杂度。反过来，这对我们测试的两种工具 AEON [1] 和 mtsNFVS [20], 产生了不同影响，因为它们采用了截然不同的吸引子识别方法。针对特定工具开展更深入的研究，或许能帮助解释为何某些约简程度比其他程度效果更好。与此同时，我们对消去顺序的影响仅进行了有限分析；我们不能排除这样的可能性：设计特定的消去顺序，使其更适合某种特定方法（例如，对于 mtsNFVS，可以研究那些不会增加最小反馈顶点集规模的顺序）。

尽管需要进行可达性分析的候选状态很少出现，但这一步或许可以通过进一步改进而受益。一项改进可以是筛查非唯一候选状态，因为 mtsNFVS 技术目前可能无法检测到多个候选状态的存在

AEON 运行时间				
模型 size	没有红色。	(1)	(2)	(3)
100	1104.6 ±676.0 (6)	2.3 ±1.3	3.0 ±1.5	3.2 ±1.5
200	(10)	83.7 ±205.7	48.9 ±113.7	146.1 ±402.4
300	(10)	1331.3 ±1394.7 (5)	696.0 ±684.1 (3)	888.9 ±1185.0 (2)
400	(10)	(10)	2994.6 (9)	3033.2 (9)
500	(10)	(10)	(10)	(10)

mtsNFVS 运行时间				
模型 size	无约简。	(1)	(2)	(3)
100	3.3	6.9 ±2.0	7.6 ±2.1	7.7 ±2.0
200	106.8 ±476.8 (11)	378.4 ±592.9 (5)	314.5 ±629.4 (7)	379.8 ±654.0 (5)
300	71.7 ±90.5(25)	1440.8 (49)	2298.6 ±653.3 (44)	3305.8 ±432.1 (46)
400	(50)	2059.3 ±644.0 (26)	2639.5 ±718.3 (43)	2239.0 ±482.2 (41)
500	(50)	1672.3 ±562.8 (44)	(50)	1276.6 ±418.3 (46)

表5：AEON [1] 和 mtsNFVS [20], 在未进行约简的随机网络以及表4所示的三种约简场景下的平均运行时间。括号中为未能在一小时内终止的进程数。由于 mtsNFVS 的运行时间波动较大，我们对每项测试运行五次。所示平均值和标准差基于给定规模下所有成功的测试。

包含于同一最小陷阱空间中的吸引子。此外，还可以结合其他技术，以排除非极大吸引子的存在。例如，目前仅使用了极小陷阱空间。更大的陷阱空间（不包含候选状态的极大陷阱空间）可以为可达性分析提供更大的靶标，同时只需对 trappist [19] 的方法进行少量修改即可轻松识别。

最后，吸引子检测工具还能够执行其他任务，例如，AEON 可以检测分岔并执行源-目标控制。约简也有可能被合理地用于加速这些活动。对于每项任务，都需要分别、谨慎地研究变量消去所带来的影响。

致谢

ET 获德国研究基金会（DFG）资助，资助项目为德国卓越战略——柏林数学研究中心 MATH+ (EXC-2046/1，项目编号 390685689)。LP 获法国国家研究署（ANR）资助，资助项目为“BNeDiction”（项目编号 ANR-20-CE45-0001）。本文所述实验使用 PlaFRIM 实验测试平台完成，该平台由 Inria、CNRS（LABRI 和 IMB）、波尔多大学、Bordeaux INP 和阿基坦大区委员会提供支持（参见 https://www.plafrim.fr）。

参考文献

[1] Nikola Beneˇs、Luboˇs Brim、Ondˇrej Huvar、Samuel Pastva、David ˇSafr´anek 和 Eva ˇSmij´akov´a。AEON.py：用于异步布尔网络吸引子分析的 Python 库。Bioinformatics，38(21)：4978–4980，2022年。

[2] Nikola Beneˇs、Luboˇs Brim、Samuel Pastva 和 David ˇSafr´anek。使用转移引导约简以符号方式计算底部强连通分量。载于《计算机辅助验证：第 33 届国际会议，CAV 2021》，线上会议，2021 年 7 月 20—23 日，论文集，第一部分 33，页码 505–528。Springer，2021年。

[3] Hannes Klarner 和 Heike Siebert。通过迭代 CTL 模型检验近似布尔网络的吸引子。《生物工程与生物技术前沿》，3:130，2015 年。[4] Hannes Klarner、Adam Streck 和 Heike Siebert。PyBoolNet：用于生成、分析和可视化布尔网络的 Python 软件包。《Bioinformatics》，33(5):770–772，2017 年。[5] Arnau Montagud、Jonas B´eal、Luis Tobalina、Pauline Traynard、Vigneshwari Subramanian、Bence Szalai、R´obert Alf¨oldi、L´aszl´o Pusk´as、Alfonso Valencia、Emmanuel Barillot、Julio Saez-Rodriguez 和 Laurence Calzone。患者特异性信号网络布尔模型为个性化治疗提供指导。《eLife》，11，2022年。[6] Kyungduk Moon、Kangbok Lee 和 Lo¨ıc Paulev´e。布尔网络中极小陷阱空间的计算复杂度。arXiv 预印本 arXiv:2212.12756，2022年。[7] Christoph M¨ussel、Martin Hopfensitz 和 Hans A Kestler。BoolNet——用于生成、重构和分析布尔网络的 R 软件包。《Bioinformatics》，26(10):1378–1380，2010 年。[8]Aur´elien Naldi、C´eline Hernandez、Nicolas Levy、Gautier Stoll、Pedro T Monteiro、Claudine Chaouiya、Tom´aˇs Helikar、Andrei Zinovyev、Laurence Calzone、Sarah Cohen-Boulakia 等。CoLoMoTo 交互式笔记本：让定性生物网络的计算分析易于使用且可复现。《生理学前沿》，9:680，2018 年。[9] Aur´elien Naldi、Elisabeth Remy、Denis Thieffry 和 Claudine Chaouiya。保留关键动力学性质的逻辑调控图约简。载于《系统生物学计算方法国际会议》，页码 266–280。Springer，2009 年。[10] Aur´elien Naldi、Elisabeth Remy、Denis Thieffry 和 Claudine Chaouiya。逻辑调控图的动力学一致约简。《理论计算机科学》，412(21):2207–2218，2011年。[11] Aur´elien Naldi、Adrien Richard 和 Elisa Tonello。布尔网络中的线性切割。arXiv 预印本 arXiv:2203.01620，2022年。[12] Lo¨ıc Paulev´e。Pint：用于定性网络瞬态动力学、带有 IPython 界面的静态分析器。载于《CMSB 2017——第 15 届系统生物学计算方法会议》，《计算机科学讲义》第 10545 卷，页码 370–316。Springer，2017 年。[13] Lo¨ıc Paulev´e 和 Sylvain Sen´e。布尔网络及其动力学：更新方式的影响。载于《系统生物学建模与分析：形式化生物信息学方法与工具》。Wiley，2022年。[14] Adrien Richard 和 Elisa Tonello。异步布尔网络中的吸引子分离与有符号环。《理论计算机科学》，页码 113706，2023 年。[15] Jordan C Rozum、Jorge G´omez Tejeda Za˜nudo、Xiao Gan、D´avid Deritei 和 R´eka Albert。奇偶性与时间反演阐明了经验模型中的决策过程以及临界布尔网络中的吸引子尺度。《Science Advances》，7(29)，2021年。[16] Julian D. Schwab、Nensi Ikonomi、Silke D. Werle、Felix M. Weidner、Hartmut Geiger 和 Hans A. Kestler。基于单细胞数据重构布尔网络集合，以揭示人类造血干细胞衰老过程中的动力学。《计算与结构生物技术杂志》，19:5321–5332，2021年。[17] Robert Schwieger 和 Elisa Tonello。异步布尔网络的约简：消去负自调控组分。arXiv 预印本 arXiv:2302.03108，2023 年。[18] Gautier Stoll、Barth´el´emy Caron、Eric Viara、Aur´elien Dugourd、Andrei Zinovyev、Aur´elien Naldi、Guido Kroemer、Emmanuel Barillot 和 Laurence Calzone。MaBoSS 2.0：随机布尔建模环境。《Bioinformatics》，33(14):2226–2228，2017 年。

[19] Van-Giang Trinh、Belaid Benhamou、Kunihiko Hiraishi 和 Sylvain Soliman。《逻辑模型的极小陷阱空间是其 Petri网编码的极大虹吸集》。载于《系统生物学计算方法：第20届国际会议，CMSB 2022》，布加勒斯特，罗马尼亚，2022年9月14日至16日，论文集，第158–176页。Springer，2022年。[20] Van-Giang Trinh、Kunihiko Hiraishi 和 Belaid Benhamou。《利用极小陷阱空间计算大规模异步布尔网络的吸引子》。载于第13届ACM国际生物信息学、计算生物学与健康信息学会议论文集，第1–10页，2022年。[21] Trinh Van Giang 和 Kunihiko Hiraishi。《一种寻找大规模异步布尔网络吸引子的改进方法》。载于《2021年IEEE生物信息学与计算生物学计算智能会议（CIBCB）》，第1–9页。IEEE，2021年。[22] Alan Veliz-Cuba。《布尔网络模型约简》。《理论生物学杂志》，289：167–172，2011年。[23] Jorge G´omez Tejeda Za˜nudo、Pingping Mao、Clara Alcon、Kailey Kowalski、Gabriela N. Johnson、Guotai Xu、Jose Baselga、Maurizio Scaltriti、Anthony Letai、Joan Montero、R´eka Albert 和 Nikhil Wagle。《乳腺癌细胞系特异性网络模型揭示潜在的PI3ka抑制剂耐药机制和药物组合》。《癌症研究》，81(17)：4603–4617，2021年。