arXiv:2302.02812v1 [q-bio.PE] 6 Feb 2023

马尔可夫与非马尔可夫流行病的模拟算法

预印本

Guohao Dou计算机与通信科学学院 洛桑
联邦理工学院 1015 洛桑
guohao.dou@epfl.ch

2023年2月7日

摘要

研究人员采用随机模拟来验证其理论发现，并研究在解析上难以处理的传播动力学。在这两种情况下，模拟算法的正确性和效率都至关重要。本文证明，用于模拟非马尔可夫流行病的两种算法——下一反应法和非马尔可夫 Gillespie 算法——在统计上等价。我们还通过复杂度分析和数值实验，研究它们在各种情形下的性能和适用性。在数值模拟中，我们将下一反应法和 Gillespie 算法用于时变网络上的流行病模拟，以及包含协同感染的流行病模拟。这两项任务此前都只采用 Gillespie 算法完成，而我们表明，下一反应法是一个不错的替代方案。我们认为，本文还可为流行病学及其他领域的研究人员选择正确且高效的模拟算法提供指导。

关键词吉莱斯皮算法 · 流行病模拟

1 引言

模拟已成为高风险政策制定中极为重要的一环，在疾病控制与预防领域尤其如此。2020年，帝国理工学院的一个团队发布了一份报告Ferguson 等人 [2020] ，模拟了在不同程度的政府干预下新冠肺炎的传播。报告预测，若政府不采取任何行动，后果将十分严重，从而促使英国政府发布了一系列新的限制措施。

我们认为，对流行病模拟的理解应跟上其日益增长的政治相关性和社会影响。正如 Kiss 等人 [2017], 所指出的，从假设人群充分混合的经典模型（例如 Kermack 和 McKendrick [1927]）到复杂的基于主体的模型（例如 Ferguson 等人 [2020]），模型的现实性和复杂度都在不断提高；前者值得追求，后者则不然。与 Kiss 等人 [2017], 的作者一样，我们也认为，基于网络的流行病模拟处于复杂度谱系的中间位置，使我们能够在数学严谨性与便于纳入现实世界数据之间保持平衡。

在使网络上的流行病模拟更加贴近现实的过程中，会遇到一些技术挑战。首先，研究人员往往假设不同隔室之间的转移服从泊松过程，而流行病学家发现，恢复率可能会随时间增加，而非保持恒定；Lloyd [2001]指出，指数分布的康复时间意味着恢复率恒定。对于持续的新冠肺炎疫情，Lu 等人 [2020] 发现，根据全部 1158 名患者的数据，最适合描述潜伏期的模型是威布尔分布，伽马分布则紧随其后。其次，通常假设流行病过程发生于静态网络中；当网络演化的时间尺度远慢于流行病时，这种假设尚属合理，但对于通勤等更快速的网络演化，显然并不成立。此外，这种假设也未能充分利用数字追踪数据 Salathe 等人 [2012]Ferretti 等人 [2020]。

为了寻找一种严谨的方法，使流行病模拟更贴近现实，我们从计算化学中汲取灵感。在不同类型的分子共存的环境中，一组耦合化学反应以不同速率发生，其发生时间服从指数分布。每次迭代都会确定反应类型和反应时间，随后执行所选反应并更新分子数量。

在他于1976年发表的论文中，Gillespie [1976], Gillespie 提出了“首次反应法”：该方法在每次迭代中，为每种化学反应生成一个暂定时间，然后找出最先发生的反应。首次反应法的缺点是，每次迭代都需要生成与反应数量相同的随机数。为解决这一效率问题，Gibson 等人提出了下一反应法（NRM）Gibson和Bruck [2000], ，该方法维护一个暂定时间优先队列，并确保在可能的情况下对随机数进行重新缩放和复用，从而将每次迭代生成的随机数减少到一个。NRM 可以轻松扩展到非马尔可夫型情形，因为即使重新缩放不再适用，我们仍可复用优先队列中已生成的暂定时间。

在同一篇1976年的论文中，Gillespie 还提出了“直接法”，该方法无需对暂定时间进行采样，便可直接生成反应时间和反应类型。此方法大量利用了这样一个性质：指数分布随机变量的最小值仍服从指数分布，其速率等于所有单个速率之和。

人们为将直接法扩展到更一般的情形付出了大量努力。Boguñá 等人Boguñá 等人[2014] 提出了非马尔可夫 Gillespie 算法（nMGA），该算法适用于非泊松转移，并允许各个“反应”的暂定时间服从任意分布。2015 年，Vestergaard 等人Vestergaard 和 Génois [2015] 通过假设网络变化仅在给定的固定时间区间发生，将该方法扩展到时变网络上的流行病过程模拟。作者还尝试探讨一般的非马尔可夫情形，尽管最终不得不采用近似方法。在本文其余部分，为了与文献中的术语保持一致，我们将类似直接法的方法称为“Gillespie 算法”，并指出其定义性特征：无需生成暂定时间的实现即可计算其最小值。

假设所有事件间隔时间均服从指数分布，Gillespie [1976] 证明了首次反应法与直接法等价，而Gibson和 Bruck [2000]证明了直接法与下一反应法等价。在一般的非马尔可夫情形下，本文证明，正如 NRM 与直接法在马尔可夫型情形下等价一样，它们的非马尔可夫型扩展也等价。

2 事件发射器

本节将定义事件发射器，并给出无向简单图上 SIS 和 SIR 流行病的事件发射器示例。

定义 1.（事件发射器）设 $\mathcal{I}$ 为标识符空间，$\mathcal{P}(\mathcal{I})$ 为其幂集。$\mathcal{S}$ 是系统的状态空间。事件发射器 e 定义为元组 (ID Ψ，f，C，R)，其中

- ID $\in \mathcal{I}$ 是事件发射器 e 的标识符。
- Ψ 是用于指定事件间隔分布的累积分布函数（CDF），其中 $\Psi(t) = 0, \ \forall\, t \leq 0$。
- f： $\mathcal{S} \to \mathcal{S}$ 是状态转移函数。
- C： $\mathcal{S} \to \mathcal{P}(\mathcal{I})$ 将一个状态映射到一组标识符，这些标识符属于执行时将被创建的事件发射器。
- R： $\mathcal{S} \to \mathcal{P}(\mathcal{I})$ 将一个状态映射到一组执行时将被移除的标识符，此外还要求 $\forall s \in \mathcal{S}$，标识符 $\in R(s)$，也就是说，事件发射器会自行移除。

Ψ 指定事件何时触发，而 f，C 和 R 则指定当选中该事件发射器执行时会发生什么。C 和 R 是状态的函数，而不是子集，因为对于非平凡应用，我们需要利用系统状态来计算要创建或移除的内容。此外，我们要求所有事件发射器在执行时自行删除，以此确保 $\forall s \in \mathcal{S}$，标识符 $\in R(s)$。

在实现过程中，建议进行一些完整性检查。例如，如果 E 是系统中的事件发射器集合，我们可能希望确保 $\forall s \in \mathcal{S}, R(s) \subseteq E$ 和 $C(s) \cap E = \emptyset$，也就是说，不允许移除不存在的事件发射器，也不允许创建重复项。

表1： **网络上 SIS 流行病的事件发射器。**

	感染	康复
ID	感染($x \to y$)	恢复(x)
f	y.state ←I (已感染)	x.state ←S (易感)
C	{恢复(y)} ∪ {感染($y \to z$) | $z \in \mathcal{N}(y)$ z.状态=S}	{感染($z \to x$) | $z \in \mathcal{N}(x)$ z.state=I}
R	{感染($z \to y$) | $z \in \mathcal{N}(y)$ z.state=I}	{恢复(x)} ∪ {感染($x \to z$) | $z \in \mathcal{N}(x)$ z.state=S}

表注： {感染($y \to z$) \| $z \in \mathcal{N}(y)$ z.状态=S} 表示“对 y 的所有易感邻居执行 Inf($y \to z$)”；为简洁起见，我们还使用状态重新赋值，而不是状态转移函数。

表 2： **网络上 SIR 流行病的事件发射器。**

	感染	康复
ID	感染($x \to y$)	恢复(x)
f	y.state ←I	x.state ←R（已康复）
C	{恢复(y)} ∪ {感染($y \to z$) | $z \in \mathcal{N}(y)$ z.状态=易感}	∅
R	{感染($z \to y$) | $z \in \mathcal{N}(y)$ z.状态=已感染}	{康复(x)} ∪ {感染($x \to z$) | $z \in \mathcal{N}(x)$ z.状态=易感}

这种编排模拟任务的方式通过将具体细节封装在定义明确的事件发射器中，提升了代码模块化。此外，事件发射器被设计为“不可变”的，也就是说，其配置不应直接修改，任何修改都必须通过移除后再重新创建来实现。

示例 1. 我们为网络上的 SIS 流行病定义事件发射器。感染($x \to y$)表示从节点 x 沿边（x, y）到节点 y 的待处理感染，而康复(x)表示节点 x待处理的康复。表 1 中有意省略了事件间隔分布 Ψ，因为它通常由流行病的特定自然史决定。对于网络上的 SIR 流行病，我们也采取相同做法，如表 2 所示。请注意，在易感-感染-康复模型中，节点一旦康复，就不再受到其受感染邻居的威胁。因此，康复(x)新创建的事件发射器集合为 $\emptyset$。这种设计清楚地表明，我们假设所有逐边感染和所有康复彼此独立。

3 下一反应法

Gibson 等人提出了下一反应法（NRM）Gibson和Bruck [2000]，以改进 Gillespie [1976] 提出的首次反应法。改进的核心思想是复用先前生成的事件间隔时间。在首次反应法中，系统中每发生一次事件，都会为每个事件发射器生成一个暂定时间。Gibson 等人证明，当事件间隔时间服从指数分布且事件发射器集合固定时，通过谨慎地重新缩放，绝大多数暂定时间都可以复用，而不会牺牲统计精确性。

将 NRM 应用于网络上的流行病模拟并非易事。首先，耦合化学反应具有固定不变的反应物集合和速率可变的反应，而流行病学中的模拟任务则具有可变的待处理事件集合，以及预先定义且固定不变的事件间隔分布。待处理事件的可变性要求在算法设计中进行更多记录，而事件间隔分布的不变性则带来了优化机会。其次，尚不清楚高效的 NRM 是否适用于更一般的非马尔可夫情形。NRM 的原始论文Gibson和Bruck [2000] 指出：“一般而言，蒙特卡罗模拟假设随机数在统计上相互独立，因此通常不能复用随机数。在这个特定的特殊情形下，我们将证明复用随机数是合理的。”然而，Gibson和Bruck [2000] 中的证明是在假设分布为指数分布的前提下完成的。

需要对术语作一项重要澄清：我们所说的非马尔可夫流行病，在概念上更接近 Gibson和Bruck [2000] 的作者在 Gibson 第 4 节中所说的“时变马尔可夫过程”

和 Bruck [2000], 所指的情形，其重点是非指数分布的事件间隔分布。Gibson和Bruck [2000] 的作者证明，当假设所有事件间隔时间均服从指数分布时，NRM 等价于 Gillespie 直接法。本文证明了 NRM 与 nMGA（Boguñá 等人 [2014] ）之间的等价性，且不对事件间隔分布作任何假设。

本节中，我们首先使用事件发射器的记号重新表述 NRM。接着，我们证明 NRM 的一些有用性质，这些性质将有助于稍后证明 NRM 与 nMGA 的统计等价性。最后，我们讨论一些实现细节并分析其复杂度。

首先，我们定义一些描述该算法所需的量。

定义 2。（暂定执行时间）我们将发射器 i 在第 l 次迭代中的暂定执行时间记为 T_i^l。如果没有其他事件发射器先将其移除，那么它将在此时执行。

定义 3。（第 l 次执行的时间和位置） T^l 是第 l 次执行的时间， K^l 是触发第 l 次执行的事件发射器。只有拥有最早事件的事件发射器才能执行，也就是说

$$K^l \coloneqq \mathrm{argmin}_i T_i^l, \quad T^l \coloneqq \min_i T_i^l.$$

本文其余部分采用以下索引约定：上标表示迭代索引，下标表示发射器 ID。例如， T^l 读作“第 l 次执行的时间”， Ψ_i 读作“事件发射器 i 的事件间隔分布”。

我们在算法 1 中描述 NRM。

算法 1NRM

1.（初始化阶段）
   - $E^1 = \{e_1^1 ... e_{n^1}^1\}$ 是初始事件发射器 ID 集合
   - 系统处于初始状态 $s \in \mathcal{S}$
   - 设置时钟 $T^0 \leftarrow 0$
   - 设置迭代索引 $l \leftarrow 1$
   - 生成 $T_i^1 \sim \Psi_i, \quad i = 1, 2, \ldots\ldots n^1$

2.（主阶段）
   (a) The algorithm terminates if $|E^l| = 0$
   (b) Find argmin by $K^l \leftarrow \mathrm{argmin}_i T_i^l$
   (c) Update the clock by $T^l \leftarrow \min_i T_i^l$
   (d) Update emitters by $E^{l+1} \leftarrow (E^l \backslash R_{K^l}(s)) \cup C_{K^l}(s)$
   (e) For each $j \in C_{K^l}(s)$, generate interevent time by $\tilde{T}_j^{l+1} \sim \Psi_j$, and then convert it to time of execution by $T_j^{l+1} \leftarrow \tilde{T}_j^{l+1} + T^l$
   (f) Apply state transition by $s \leftarrow f_{K^l}(s)$
   (g) Increment iteration index by $l \leftarrow l + 1$
   (h) Go back to step (a)

我们从算法 1 中提取出以下更新规则。

$$T_i^{l+1} = \begin{cases} \tilde{T}_i^{l+1} + T^l & \text{if } i \in C_{K^l} \text{(“freshly generated”)} \\ T_i^l & \text{if } i \notin C_{K^l} \text{(“reused”)} \end{cases}$$

其中 $\tilde{T}_i^{l+1} \sim \Psi_i$。“新生成”和“复用”的含义不言自明，因为只有 C_{K^l}中的事件发射器需要进行随机数生成（RNG）。事实上，对于每次迭代，暂定时间要么是新生成的，要么是复用的，二者构成一个划分。在本文余下部分中，我们将迭代 l 中暂定时间被复用的发射器 ID 集合记为 $\mathcal{R}^l$。根据定义， $\mathcal{R}^l \coloneqq E^l \setminus C_{K^{l-1}}$。

鉴于算法的结果完全由每次迭代中生成的最小值和最小值对应参数所决定，我们将截至第 $l-1$ 次迭代的轨迹定义为一系列实现。

定义 第 4 节。（截至 $l-1$ 的轨迹）算法截至迭代 $l-1$ 的轨迹记为 a s

$$\{\overrightarrow{kt}^{l-1}\} \coloneqq \{K^1 = k^1, T^1 = t^1, ..., K^{l-1} = k^{l-1}, T^{l-1} = t^{l-1}\}.$$

序列 $\{t^0, t^1, ..., t^l, ...\}$ 是实数半轴上的一个点过程，只有在这些点上才能创建或移除事件发射器。

现在我们来探讨 T_i^l 的分布。首先需要指出， T_i^l 的分布始终是指以截至 $l-1$ 的轨迹为条件的分布。写下 T_i^l 的分布时，我们真正指的是

$$\Pr\{T_i^l \le t \mid \overrightarrow{kt}^{l-1}\}.$$

一旦生成了一个暂定时间，除非被其他事件发射器的执行移除，否则它永远不会改变。然而，由于轨迹中只记录最小值和 argmin，我们可以将 Ψ_i 视为先验分布，并将定位最小值的过程视为更新先验以获得后验分布的过程。我们所说的 T_i^l 的分布，是指以截至 $l-1$ 的轨迹为条件的后验分布。

知道 $\overrightarrow{kt}^{l-1}$ 后，我们可以恢复事件发射器集合 E^l 及其复用的划分 $\mathcal{R}^l$。

示例 2. 为使算法 1 不那么抽象，我们将通过图1所示的一个最小示例，逐步说明前三次迭代。事件发射器的行为定义见表1。为清晰起见，我们在下标中使用整数，而不使用完整的发射器 ID。

我们观察到，有些暂定时间是新生成的，例如 $T_1^1, T_2^1, T_3^1, T_3^2, T_4^2, T_5^3$；而另一些则是复用的，例如 $T_1^2, T_2^2, T_3^3, T_4^3$。因此，$\mathcal{R}^1 = \emptyset, \mathcal{R}^2 = \{1,2\}, \mathcal{R}^3 = \{3,4\}$。

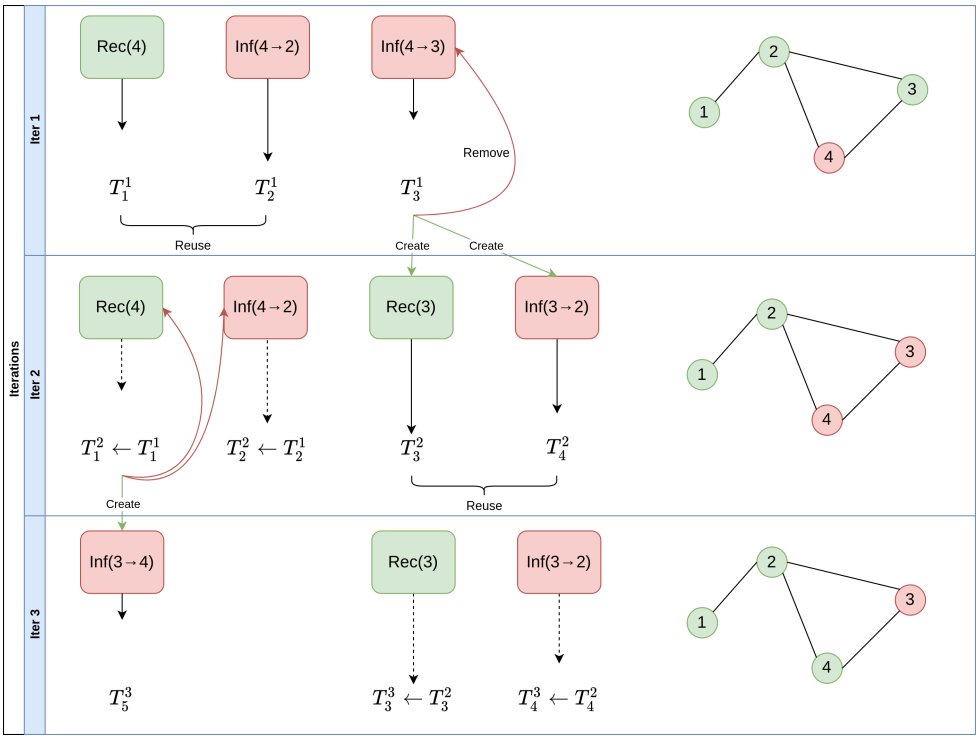

图 1：**NRM 示例：小型网络上的 SIS 流行病**。图的左侧描绘了事件发射器的演变，右侧则跟踪网络的状态。每个方框代表一个事件发射器，其下方的箭头长度表示暂定执行时间。正如我们在第一次迭代中看到的，Inf(4→3)抽中了最短的一根签，导致节点 3 感染、Inf(4→3) 被移除，同时生成了 Rec(3) 和 Inf(3→2)。至于第二次迭代，前两个虚线箭头反映出这两者无需额外进行随机数生成，因为复用了上一次迭代中的数值。与此同时，两个新生成的事件发射器 Rec(3) 和 Inf(3→2) 确实需要进行随机数生成，如实线箭头所示。

在介绍 NRM 之后，我们将展示它的一些性质，这些性质对于确立 NRM 与 nMGA 之间的统计等价性至关重要。我们首先对新生成的暂定时间的独立性作出一个假设。

假设 1. 设 $\{T_i^l\}_{i\in\mathcal{J}} \subseteq \{T_i^l\}_{i\in E^l}$ 是迭代 l 和 T_j^l 中暂定时间的任意子集，$j \in \mathcal{J}$ 是一个新生成的暂定时间。联合累积分布函数可分解如下，

$$\Pr\{\forall i \in \mathcal{J}, T_i^l \le t_i \mid \overrightarrow{kt}^{l-1}\} = \Pr\{T_j^l \le t_j \mid \overrightarrow{kt}^{l-1}\} \cdot \Pr\{\forall i \in \mathcal{J}\backslash\{j\}, T_i^l \le t_i \mid \overrightarrow{kt}^{l-1}\}.$$

注。 请注意，假设 1 弱于相互独立。例如，在图1中，尽管我们可以放心地进行

$$\Pr\{T_3^2 \le t_3, T_4^2 \le t_4 \mid \overrightarrow{kt}^1\} = \Pr\{T_3^2 \le t_3 \mid \overrightarrow{kt}^1\} \Pr\{T_4^2 \le t_4 \mid \overrightarrow{kt}^1\},$$

¯我们不能仅根据假设 1 将 $\Pr\{T_1^2 \le t_1, \overrightarrow{T_2^2} \le t_2 \mid kt^1\}$ 分解，因为 $T_1^2 \leftarrow T_1^1$和 $T_2^2 \leftarrow T_2^1$都被复用，而非**新** 生成。这一困难促使我们考察复用暂定时间的影响，并检查该算法是否会在原本相互独立的随机变量之间引入依赖关系。

在本文余下部分，我们假设暂定时间 T_i^l 是连续随机变量。我们还采用以下向量化简写

- 对于一个标识符集合 $\mathcal{J}$和一个标量 t

$$\Pr\{\vec{T}_{\mathcal{J}}^l \le t \mid \overrightarrow{kt}^{l-1}\} \coloneqq \Pr\{\forall j \in \mathcal{J}, T_j^l \le t \mid \overrightarrow{kt}^{l-1}\}.$$

- 对于一个 ID集合 $\mathcal{J}$和一个向量 $\vec{t}_{\mathcal{J}}$

$$\Pr\{\vec{T}_{\mathcal{J}}^l \le \vec{t}_{\mathcal{J}} \mid \overrightarrow{kt}^{l-1}\} \coloneqq \Pr\{\forall j \in \mathcal{J}, T_j^l \le t_j \mid \overrightarrow{kt}^{l-1}\}.$$

应当理解，在 $\Pr\{\vec{T}_{\mathcal{J}}^l \le t \mid \overrightarrow{kt}^{l-1}\}$或 $\Pr\{\vec{T}_{\mathcal{J}}^l \le \vec{t}_{\mathcal{J}} \mid \overrightarrow{kt}^{l-1}\}$中，我们只考虑$t > t^{l-1}$ 或 $\forall j \in \mathcal{J}$，$t_j > t^{l-1}$的情况。否则，两者都将为零。

现在我们可以证明定理 1 了。对于所有事件间隔时间都服从指数分布的情况，Gibson和Bruck [2000] 已经证明了暂定时间的相互独立性。这里，我们将证明非指数型暂定时间的相互独立性，并进一步考虑到事件的执行可能会创建或移除其他事件发射器。

定理 1. 对于任意 $l > 0$，$\{T_i^l\}_{i\in E^l}$ 由给定 $\overrightarrow{kt}^{l-1}$的相互独立的随机变量组成。

证明见附录S2。

推论 1. （定理 *1* 的推论）如果 $\{T_i^l\}_{i\in\mathcal{J}}$由复用的暂定时间组成，那么它们的联合累积分布函数可以表示为

$$\Pr\{\vec{T}_{\mathcal{J}}^l \le \vec{t}_{\mathcal{J}} \mid \overrightarrow{kt}^{l-1}\} = \prod_{i\in\mathcal{J}} \Pr\{T_i^{l-1} \le t_i \mid T_i^{l-1} > t^{l-1}, \overrightarrow{kt}^{l-2}\}.$$

我们已经成功地用上一轮迭代中 T_i^{l-1} 的分布来表示 T_i^l 的分布。为了进一步分析，我们现在用定义1中引入的用户指定事件间隔分布来表示 T_i^l 的分布。为此，我们首先定义事件发射器的创建时间。

定义5.（创建时间）t_i^l 是 l次迭代中事件发射器 i 的创建时间。

从算法的角度来看，跟踪创建时间很简单：只需在创建事件发射器时，将时钟值存储在其中即可。然而，从数学角度来说，只有掌握先前轨迹 $\{\overrightarrow{kt}^{l-1}\}$的完整信息，我们才能恢复创建时间。对于初始时已存在的事件发射器，它们的创建时间自然是 $t^0 = 0$。对于在第 l次迭代中创建的事件发射器，它们的创建时间是 t^l。在这两种情况下，t_i^l 只取自 $\{t^0, t^1, t^2, ..., t^{l-1}\}$。标识符相同的事件发射器也可能在被移除后重新创建，此时创建时间指向最近一次创建。

引理 1. 设 Ψ_i 为发射器 i的事件间隔分布的累积分布函数，t_i^l 为事件发射器 e_i^l 的创建时间。T_i^l 的累积分布函数为

$$\Pr\{T_i^l \le t \mid \overrightarrow{kt}^{l-1}\} = \frac{\Psi_i(t - t_i^l) - \Psi_i(t^{l-1} - t_i^l)}{1 - \Psi_i(t^{l-1} - t_i^l)}. \tag{1}$$

证明见S3附录。

为简化公式 (1)，我们采用 Boguñá 等人 [2014] 的记号作为简写。设 Ψ 为用户指定的事件间隔分布，例如指数分布、伽马分布或威布尔分布。将其平移 t 后的分布表示为公式 (2) 所示，

$$\Psi(\tau \mid t) \coloneqq \frac{\Psi(\tau + t) - \Psi(t)}{1 - \Psi(t)}, \quad \psi(\tau \mid t) \coloneqq \frac{\partial \Psi(\tau \mid t)}{\partial \tau} = \frac{\psi(\tau + t)}{1 - \Psi(t)}. \tag{2}$$

通过对式（1）和式（2）进行简单的模式匹配，可得到式（3）

$$\Pr\{T_i^l \le t \mid \overrightarrow{kt}^{l-1}\} = \Psi_i(t - t^{l-1} \mid t^{l-1} - t_i^l). \tag{3}$$

例 3. 我们再来看一下图1中的例子。我们记得，只有在给定先前轨迹的情况下，这些暂定时间才有明确定义。在这个特定情形中，T_1^2的存在表明，T_1^1必须大于第一次迭代中暂定时间的最小值 t^1，因此得到式（4）：

$$\Pr\{T_1^2 \le t \mid \overrightarrow{kt}^1\} = \Pr\{T_1^1 \le t \mid T_1^1 > t^1\} = \Psi_1(t - t^1 \mid t^1). \tag{4}$$

事实上，式（4）是式（3）在 $l = 2$和 $t_1^2 = t^0 = 0$时的特例。

现在我们来探讨 NRM 每次迭代的计算成本和重要的实现细节。我们作出以下假设：

- 活动事件发射器的数量显然始终不超过 N。特别是，对于网络上的流行病，N 按 $\mathcal{O}(V + E)$ 的比例增长。
- 每次迭代创建/移除的事件发射器数量为 $\mathcal{O}(1)$。这一假设适用于网络上的流行病模拟，因为这些数量通常不超过网络的平均度（例如，对于 x 的每个易感邻居 y，$\mathrm{Inf}(x \to y)$）。
- 生成每个事件间隔时间的复杂度为 $\mathcal{O}(1)$；生成的事件间隔时间严格遵循算法 1 中所示的分布，不涉及任何近似。

这些假设适用于本文所有算法的复杂度分析。

由于 NRM 会找出现有事件发射器提出的最小暂定时间，因此我们使用最小堆。最小堆数据结构应支持以下三种操作：

- **RemoveMin**：获取并移除堆顶元素，即最小暂定时间；时间复杂度为 $\mathcal{O}(\log\ N)$。
- **插入**：将一个事件发射器及其对应的暂定时间插入堆中；时间复杂度为 $\mathcal{O}(\log\ N)$。
- **按 ID 删除**：根据事件发射器的 ID 将其从堆中删除（不一定是堆顶元素），同时保持堆性质；时间复杂度为 $\mathcal{O}(\log\ N)$。

虽然 **RemoveMin** 和**插入** 是标准操作，但**按 ID 删除** 并不是，因为在经典的二叉堆实现中，只能访问堆顶元素。为此，我们需要一个哈希表，将事件发射器 ID 映射到其在堆中的位置，并从哈希表指示的位置而非堆顶开始执行下滤。时间复杂度仍为 $\mathcal{O}(\log\ N)$，因为其操作次数不会超过从堆顶开始下滤所需的操作次数。

每次迭代都会调用一次**RemoveMin**，其时间复杂度为 $\mathcal{O}(\log\ N)$。由于我们假设创建/删除的事件发射器数量始终为 $\mathcal{O}(1)$，因此**插入** 和**按 ID 删除** 的每次迭代时间复杂度也为 $\mathcal{O}(\log\ N)$。

因此，算法 1 的每次迭代时间复杂度为 $\mathcal{O}(\log\ N)$。

4 Gillespie 算法

Gillespie 算法在 Gillespie [1976], 中也称为“直接法”，旨在减少随机数生成。在算法 1 中，每次迭代需要生成的随机数数量与新创建的事件发射器数量相同。然而，在本节即将介绍的所有 Gillespie 算法变体中，每次迭代恰好只需生成两个随机数。我们将这些算法称为“Gillespie 算法”，并按照从功能多样但速度较慢到功能有限但速度较快的顺序逐一介绍。

4.1 一般非马尔可夫型流行病

Boguñá 等人在 Boguñá 等人 [2014] 中介绍了非马尔可夫 Gillespie 算法（nMGA），并将 Gillespie [1976] 中的马尔可夫型 Gillespie 算法扩展到更一般的非马尔可夫型情形。除了前面提到的减少随机数生成外，在计算事件率很简单、而生成事件间隔时间需要复杂数值方法的情况下，nMGA 也更适用；Boguñá 等人 [2014] 中给出的协同感染示例便说明了这一点。Boguñá 等人 [2014] 的作者还提出了 nMGA 的近似版本，该版本用阶梯函数近似实际的事件率函数。为避免混淆，本文将 nMGA 的精确版本称为“nMGA-Exact”，将近似版本称为“nMGA-Approx”。

我们首先回顾 nMGA 的两种版本：精确版本（算法 2）和近似版本（算法 3），并使用与相同的 e 记法 m 1.

算法 2nMGA-Exact

1.（初始化阶段）
   - E^1 给定为初始事件发射器集合
   - 系统处于初始状态 $s \in \mathcal{S}$
   - 设置时钟 $T^0 \leftarrow 0$
   - 设置迭代索引 $l \leftarrow 1$
   - 记录创建时间 $\forall i \in E^1,\ t_i^1 \leftarrow 0$
2.（主阶段）
   (a) Algorithm terminates if $|E^l| = 0$
   (b) Compute the global survival probability $\Phi^l(t) = \prod_{i \in E^l} \left[1 - \Psi_i(t \mid t^{l-1} - t_i^l)\right]$
   (c) Generate $\Delta^l \sim \Phi^l$ and progress in time by $T^l \leftarrow T^{l-1} + \Delta^l$
   (d) Compute PMF with weights $\Pi^l(i) \coloneqq \frac{\psi_i(0 \mid t^l - t_i^l)}{\sum_i \psi_i(0 \mid t^l - t_i^l)}$ and generate $K^l \sim \Pi^l$
   (e) Update emitters by $E^{l+1} \leftarrow (E^l \backslash R_{K^l}(s)) \cup C_{K^l}(s)$
   (f) Store times of creation $\forall i \in C_{K^l}(s), t_i^l \leftarrow T^l$
   (g) Apply state transition by $s \leftarrow f_{K^l}(s)$
   (h) Increment $l \leftarrow l + 1$
   (i) Go back to step (a)

算法 3nMGA-Approx

我们用以下两个步骤替换算法 2 中的步骤 (b) 和 (c)。首先计算

$$\lambda_i^l \coloneqq \psi_i(0 \mid t^{l-1} - t_i^l).$$

然后生成

$$\Delta^l \sim \text{Exp}\left(\sum_i \lambda_i^l\right),$$

and 按照惯例，通过 $T^l \leftarrow t^{l-1} + \Delta^l$ 推进时间。其余部分与算法 2 相同。

接下来，我们证明算法 1 和算法 2 在统计上等价。算法 1 和算法 2 在执行 l 次迭代后，各自都会生成一个 $2l$ 维随机向量 $\{K^1, T^1, K^2, T^2, ..., K^l, T^l\}$ 的实现。因此，如果对于任意 $l > 0$，两种算法截至 l 的轨迹都具有相同的分布，我们就称这两种算法等价。

由于随机向量 $\{K^1, T^1, K^2, T^2, ..., K^l, T^l\}$ 同时包含离散和连续随机变量，因此无法直接应用随机向量的概率密度函数等标准记法。不过，我们可以通过条件化来绕过这一难题。同样，令 $\overrightarrow{kt}^{l-1}$ 为前一条轨迹。请注意，T^l的密度函数

$$f_{T^l}(t^l) \coloneqq \frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t^l} \Pr\{T^l \leq t \mid \overrightarrow{kt}^{l-1}\}$$

由于连续性假设而有明确定义。同时，给定 $T^l = t^l$， K^l 是一个离散随机变量，其取值来自 E^l，其分布

$$\Pi^l(k^l \mid t^l) \coloneqq \Pr\{K^l = k^l \mid T^l = t^l, \overrightarrow{kt}^{l-1}\}$$

由.....描述 由概率质量函数描述。例如，在算法 2 中，该概率质量函数被明确定义为 b e

$$\tilde{\Pi}^l(k^l \mid t^l) \coloneqq \frac{\psi_{k^l}(0 \mid t^l - t^l_{k^l})}{\sum_{i \in E^l} \psi_i(0 \mid t^l - t^l_i)}. \tag{5}$$

将两者结合起来， T^l， K^l 的联合密度函数为

$$f_{T^l,K^l}(t^l, k^l \mid \overrightarrow{kt}^{l-1}) \coloneqq f_{T^l}(t^l) \cdot \Pi^l(k^l \mid t^l). \tag{6}$$

我们强调，该算法的行为完全由式（6）定义。设 $f^A_{T^l,\ K^l}$， $f^B_{T^l,\ K^l}$ 分别为算法 A 和算法 B 的联合密度函数。对于任意 $l > 0$，给定任意先前轨迹 $\overrightarrow{kt}^{l-1}$，若

$$f^A_{T^l,K^l}(\cdot \mid \overrightarrow{kt}^{l-1}) \equiv f^B_{T^l,K^l}(\cdot \mid \overrightarrow{kt}^{l-1}),$$

这显然意味着它们的轨迹联合分布也相同。

定理 2. 给定假设 *1*，算法 *1* 与算法 *2* 等价。

证明见附录 S4。

算法 2 的效率相当低，因为使用逆累积分布函数技术从 $\Phi^l(t) = \prod_i \left[1 - \Psi_i(t \mid t^{l-1} - t^l_i)\right]$ （算法 2 的步骤 (c)）生成随机数需要进行大量求根。对于这个特定问题，我们从初始区间 [0.1, 1]开始，然后不断将下界减半、上界加倍，直到两端异号。求根时，我们使用 SciPy（Virtanen 等人， [2020]）提供的 Brent 求根法 Brent [2013] 。

求根过程的每次计算复杂度均为 $\mathcal{O}(N)$；如果所需误差容限为 ϵ，Brent 法需要进行 $\mathcal{O}(\log \frac{1}{\epsilon})$ 次计算才能收敛。

因此，算法 2 的每次迭代时间复杂度为 $\mathcal{O}(N \log \frac{1}{\epsilon})$。

将算法 2 的步骤 (b) 和 (c) 替换为从指数分布中采样后，算法 3 无需再进行代价高昂的求根。Boguñá 等人 [2014] 还提出了更严谨的方法来证明使用指数分布的合理性；一种直观的理解方式是，将每个事件间隔时间视为服从相应瞬时速率所决定的指数分布，并在每次迭代中更新速率，如图 2 所示。很容易验证，当所有事件间隔分布恰好都是指数分布时，这种近似是精确的。事实上，这种近似假设风险函数 λ^l_i在时间区间 $(t^{l-1},\ t^l)$ 内保持恒定；如果 $t^l - t^{l-1}$ 恰好很小，那么近似会相当准确。

我们还对其误差界做了一些观察。回想一下，风险函数 $\lambda(t)$、概率密度函数 $\psi(t)$ 和互补累积分布函数（CCDF） $\Phi(t)$ 满足以下方程:

$$\lambda(t) \coloneqq \frac{\psi(t)}{\Phi(t)} = -\frac{\mathrm{d}}{\mathrm{d}t} \ln \Phi(t) \implies \int_0^t d\tau \lambda(\tau) = -\ln(\Phi(t)). \tag{7}$$

事件发射器 i 的风险函数为 $\lambda_i(t) \coloneqq {}^{\psi_{i(t)}\ \Phi_i(t)}$。由于在算法 3 中，风险函数仅在$t^0, t^1, t^2 \ldots$时重新计算，因此我们将事件发射器 i的阶梯风险函数 $\bar{\lambda}_i(t)$ 记为

$$\bar{\lambda}_i(t) \coloneqq \lambda_i(t^l) \text{ if } t^l \le t < t^{l+1}.$$

因此，我们有

$$-\ln(\Phi_i(t)) = \int_0^t d\tau \lambda_i(\tau) = \epsilon + \int_0^t d\tau \bar{\lambda}_i(\tau) = \epsilon + \sum_{l=0}^{t^l < t} \lambda_i(t^l) \cdot \Delta t^l$$

其中

$$\Delta t^l = \begin{cases} t^{l+1} - t^l & \text{if } t^{l+1} < t \\ t - t^l & \text{otherwise.} \end{cases}$$

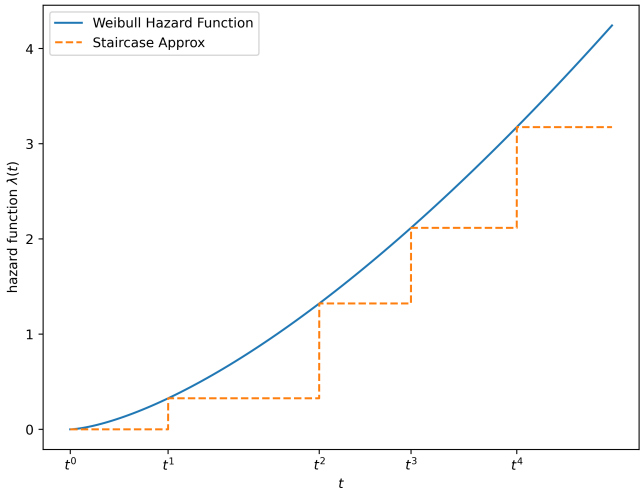

图2：**阶梯近似与实际风险函数的比较。** t^0、.....、t^4 是执行时间。具有 $\kappa = 1.5$、$\lambda = 1$ 的威布尔风险函数被阶梯函数近似，这是因为瞬时速率仅在事件发生时才会重新计算和更新。

注意，误差项 ϵ是积分 $\int_0^t d\tau\lambda_i(\tau)$与其左黎曼和之差。作为合理性检验，请注意，如果 $\lambda_i(t)$为常数（如指数分布中的情形），则 $\epsilon = 0$。由于 $\{t^0, t^1, t^2, ...t\}$ 构成区间 $[0, t]$的一个划分，当网格大小 $\max_l \Delta t^l$趋于零时，ϵ 也趋于零。为简单起见，若我们假设 $\Delta t^l \equiv \Delta t$ 和$M_1 := \sup_{\tau\in[0,t]} |\lambda_i'(\tau)|$，则 $\epsilon \leq M_1 t\Delta t$，这是由左黎曼和的选取所致。

至于复杂度，在每次迭代中，需要重新计算并求和所有瞬时速率 λ_i^l，其复杂度为 $\mathcal{O}(N)$；如果采用标准逆累积分布函数，从 PMF 生成 k 的复杂度也是 $\mathcal{O}(N)$。
因此，算法 3 的每次迭代时间复杂度为 $\mathcal{O}(N)$。

4.2 具有异质性感染率的马尔可夫流行病

在本节的设定中，我们假设所有已感染节点的恢复率均相同，速率为 γ。然而，逐边感染率不一定相同，也就是说，从节点 i 到节点 j 的逐边感染率应记为 $\beta_{i\to j}$。如果像前几节一样，假设各逐边感染相互独立，那么易感节点 j上的总感染率 β_j可表示为

$$\beta_j := \sum_{i\in\mathcal{N}(j)}^{i \text{ infected}} \beta_{i\to j}.$$

Kiss 等人 [2017] 考虑了这种设定，并提出了他们的吉莱斯皮算法变体。由于该算法需要计算所有易感节点上的总感染率，为了区分起见，我们称之为节点中心吉莱斯皮算法（以节点为中心的遗传算法），其步骤见算法4。

尽管其他所有步骤 都可以通过仔细跟踪优化至 $\mathcal{O}(1)$，但算法4中的第6步显然是瓶颈 ，
概率质量函数中采样需要 $\mathcal{O}(N)$。
因为从一个
因此，算法4每次迭代的复杂度为 $\mathcal{O}(N)$。

4.3 均匀感染速率下的马尔可夫型流行病

请注意，在算法4中，由于假设恢复率是同质的，因此在算法4的步骤5中，定位要恢复的感染节点并不费力；均匀随机采样即可 $\mathcal{O}(1)$。如果我们进一步假设所有逐边感染率均为 β，即 $\forall i,\ j,\ \beta_{i\to j} \equiv \beta$，就可以省去算法4中代价高昂的步骤6。将具有相同速率的事件发射器分组是很自然的想法，因此许多人一直这样做也就不足为奇了

算法4以节点为中心的遗传算法

1. 对每个易感节点 j，按以下方式计算 β_j

$$\beta_j = \sum_{i \in \mathcal{N}(j)}^{i \text{ infected}} \beta_{i \to j}.$$

2. 求出康复的总速率 λ_R 和感染的总速率 λ_I，方法如下：

$$\lambda_R = \gamma \cdot (\text{\# of infected nodes}), \quad \lambda_I = \sum_{\text{susceptible } j} \beta_j.$$

3. 下一事件发生的时间为 $t \sim$ 指数分布($\lambda_R + \lambda_I$)。4. 生成一个 $u \sim$ 均匀分布(0, 1)随机数。如果 $u < \lambda_R/(\lambda_R + \lambda_I)$，则下一事件为康复；否则为感染。5. 如果下一事件为康复，则从已感染节点中均匀随机地选择一个使其康复，并更新其邻居节点的感染率。6. 如果下一事件为感染，则根据概率质量函数 $p_j = \beta_j/\lambda_I$ 选择一个易感节点使其感染，并更新其邻居节点的感染率。7. 返回步骤1。

已经在不知不觉中进行了这种优化。事实上，在 Ferreira 等人 [2012], 的工作中，作者在离散化模拟中采用了这一思路。本节将介绍一种支持连续时间模拟的算法版本，并提供一些实现技巧。在本文余下部分，我们将这一算法称为优化遗传算法。

这项优化涉及维护一个已感染节点的哈希集合 S_R，以及另一个活跃边（连接感染节点与易感节点的边）的哈希集合 S_I。这个哈希集合数据结构应支持以下操作：

- $\mathcal{O}$(1) 插入。
- $\mathcal{O}$(1) 查找与删除。
- $\mathcal{O}$(1) 均匀随机地选择一个元素。

前两项是标准的哈希集合操作，第三项则需要一些工程学上的处理。一种可能的实现方式是维护一个哈希表，将元素映射到随机访问数组中的索引。该算法见算法 5。

算法 5优化遗传算法

1. 通过以下方式计算康复的总速率 λ_R 和感染的总速率 λ_I

$$\lambda_R = |S_R| \cdot \gamma, \quad \lambda_I = |S_I| \cdot \beta.$$

2. 到下一个事件的时间为 $t \sim$ Exp($\lambda_R + \lambda_I$)。3. 生成 $u \sim$ Uniform(0，1)。如果 $u < \lambda_R/(\lambda_R + \lambda_I)$，则下一个事件是康复；否则，是感染。4. 如果下一个事件是康复（感染），则从 S_R(S_I) 中均匀随机选择一个元素，以确定康复（感染）的具体位置。5. 根据表1更新两个哈希集合。6. 返回步骤1。

由于每个单独步骤的成本为 $\mathcal{O}$(1)，因此这种优化后的吉莱斯皮算法确实将单次迭代成本降至$\mathcal{O}$(1)。

尽管我们只展示了如何模拟易感-感染-易感流行病，但这种将具有相同泊松率的事件发射器分组的优化技术，适用于所有转移率均匀且恒定的基于舱室的网络模型。只要分组数不超过 $\mathcal{O}$(log N)，优化 GA 的表现就会优于 NRM。不过，对于更复杂的情形，例如感染率允许存在异质性的情况，这种方法并不适用；感染率可以

遵循某种分布 Qu 和 Wang [2017]Ferguson 等人 [2005]，（如伽马分布或对数正态分布），也可以根据经验接触强度数据 Buono 等人 [2013] 进行设定。它同样不适用于非马尔可夫情形，因为此时事件率随时间变化，且远非均匀。

5 实验

前文中，我们使用大 O 记号分析了不同算法的性能。本节旨在研究这些算法在实际中的表现。

5.1 非马尔可夫型模拟

本节中，我们使用针对非马尔可夫流行病的三种算法进行两次数值模拟：NRM (1)、nMGA-Exact (2) 和 nMGA-Approx (3)。
在这两项实验中，所用网络均为 Erdos-Renyi $G(n,\ p)$ 随机图Erd˝os 和 Rényi [1960]，其平均度为 $\langle k\rangle = 5$。网络生成后，将在所有重复运行中保持不变，以减少无关方差。两项实验均从 10 个初始感染节点开始。实现基本遵循表1，但我们需要格外仔细地记录已用时间（见 S1 附录），Boguñá 等人 [2014] 也提出了这一点。

所选流行病模型为易感-感染-易感流行病，其中每条边的感染时间服从威布尔分布($\kappa = 2\ \lambda =$ ，5.641895836)，平均感染时间为 $T_{\text{inf}} = 5$。康复时间服从指数分布(0.25)，平均康复时间为 $T_{\text{rec}} = 4$。
按照一般非马尔可夫型情形的定义，有效感染率为 $\mathcal{R} = \langle k\rangle \frac{\bar{T}_{\text{rec}}}{\bar{T}_{\text{inf}}} = 5 \cdot \frac{4}{5} = 4$，远高于 1，因此可确保系统处于地方性流行状态。
在这两项实验中，我们收集了每种算法生成的轨迹，并对其进行线性插值。随后，在等间隔的时间戳上评估这些插值后的轨迹，以便计算每个时间戳对应的样本均值和标准差，并据此绘制误差条。

第一种设置是在包含 1000 个节点的网络上模拟 SIS 流行病，并通过运行 10000 次迭代生成轨迹。每种算法均进行 150 次重复运行。轨迹如图 3 所示。性能基准测试结果汇总于表 3，结果表明 NRM 在统计上精确且性能优异。行 “$\Delta \sim \Phi$” 表示生成下一事件时间所耗费的时间，行 “$K \sim \Pi$” 表示选择事件发射器所耗费的时间。
第二种设置是在包含 10000 个节点的图上，使用相同的事件间隔分布模拟 SIR 流行病，并持续模拟，直至不再有已感染节点。轨迹如图 4 所示。性能基准测试结果汇总于表 4。

表 3：**性能基准测试：SIS 流行病， + 1000 个节点**。

	nMGA-Exact	nMGA-Approx	NRM
总计	∼26 分钟	∼65 秒	∼0.8 秒
$\Delta \sim \Phi$	∼25 分钟	∼30 秒	
$K \sim \Pi$	∼30 秒	∼30 秒	

表 4：**性能基准测试：SIR 流行病 + 10000 节点**。

	nMGA-Exact	nMGA-Approx	NRM
总计	∼80 分钟	∼150 秒	∼0.8 秒
$\Delta \sim \Phi$	∼80 分钟	∼75 秒	
$K \sim \Pi$	∼50 秒	∼75 秒	

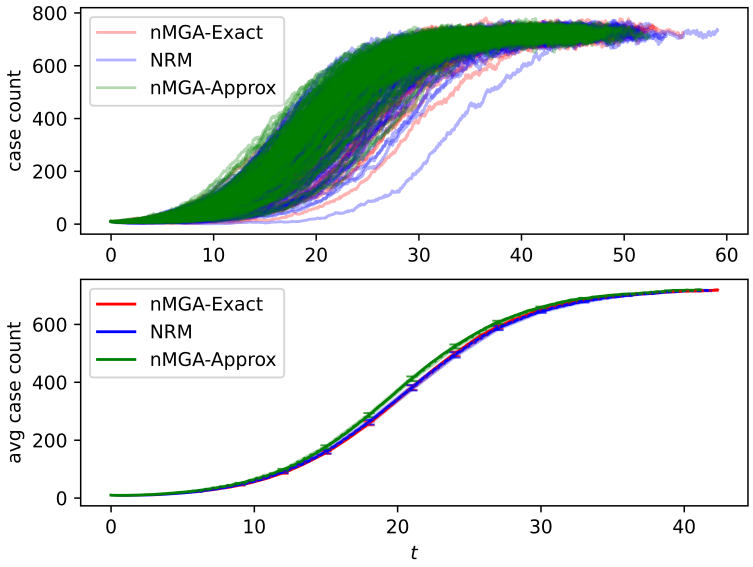

图3：**轨迹和平均轨迹：SIS 流行病 + 1000 节点**。x 轴表示时间，y 轴表示网络中的感染节点数量。在图 3 的下半部分，代表 nMGA-Approx 平均轨迹的绿色曲线与另外两条曲线出现偏离，这符合近似算法的预期。同时，NRM 和 nMGA-Exact 的平均轨迹无法区分，这也是意料之中的，因为二者在统计上是等价的。

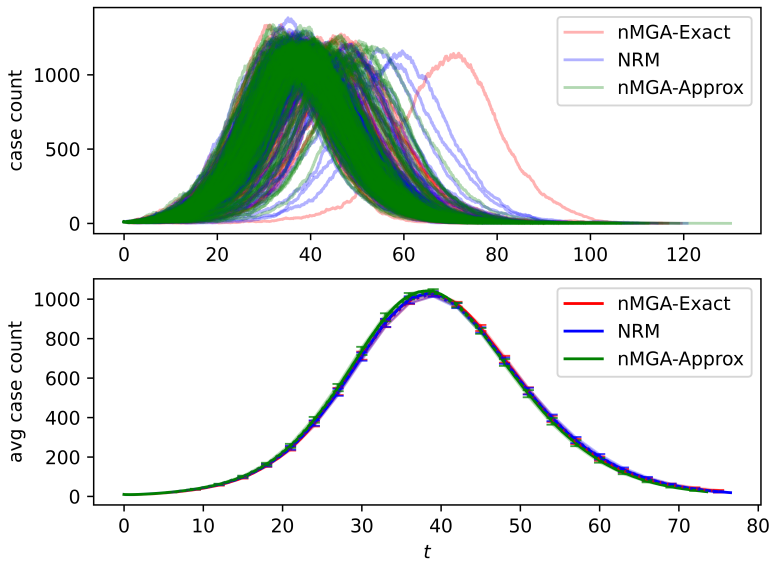

图4：**轨迹和平均轨迹：SIR 流行病 + 10000 节点**。我们观察到，NRM（蓝色）和 nMGA-Exact（红色）再次无法区分，而 nMGA-Approx（绿色）预测的峰值略早。

5.2 马尔可夫模拟

本节中，我们使用三种适用于均匀感染率马尔可夫流行病的算法，对不同图规模进行数值模拟：以节点为中心的遗传算法 (4)、NRM (1) 和优化遗传算法 (5)。为研究这三种算法的扩展性，我们采用不同规模的 ER 图，每个图中初始感染节点占 5%。对于每种图规模，每种算法运行 20 次，以测量平均耗时。每次运行中，模拟持续进行，直至模拟世界达到 30 天的时间范围。每条边的感染时间和恢复时间设置为

$$T_{\text{inf}} \sim \text{Exp}(0.125), \quad T_{\text{rec}} \sim \text{Exp}(0.25).$$

请注意，对于马尔可夫流行病，在给定时间范围时，达到该时间范围所需的迭代次数与图规模呈线性关系。假设给定的时间范围为 H，并且我们有 N 个速率为λ的自更新事件发射器。连续两个事件之间的时间间隔 T服从速率为 $N\lambda$的指数分布，其中 $\mathbf{E}[T] = (N\lambda)^{-1}$。因此，达到时间范围 H所需的预期迭代次数为 $^{H}_{\mathbf{E}[T]} = HN\lambda$。

三种算法的性能如图 5 所示。

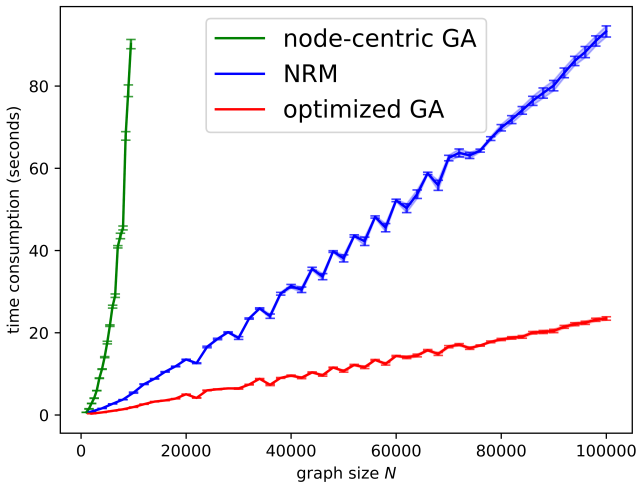

图 5：**具有齐次速率的马尔可夫流行病**。正如复杂度分析所预测的，对于优化遗传算法，单次迭代成本为 $\mathcal{O}(1)$，意味着总体成本为 $\mathcal{O}(N)$。与此同时，在所考虑的图规模下，NRM 的速度约慢 3 到 4 倍。然而，对于较大的图规模，Node-centricGA 已经难以处理。

5.3 时序/自适应网络

近年来，研究人员对流行病学领域中涌现的时序/自适应网络（Gross等人[2006]Wang等人 [2019] ）以及其他领域中的此类网络（Gross等人和Blasius [2008]）表现出兴趣。针对随机模拟，研究人员已开发出新的算法（Vestergaard和Génois [2015]）。此外，研究人员也在理论上尝试统一这一新兴领域（Wang等人 [2019]）。

时序网络是指其连接关系随时间变化的网络。一方面，这种连接关系的变化可能由外部驱动，即网络结构的变化独立于网络内部发生的动力学过程。例如，接触网络中的通勤模式可以说是由外部驱动的。如果此类数据是定期收集的，那么由接触者追踪/监测造成的网络变化也可能只是数据采集的产物。另一方面，网络变化也可能由内部驱动，即网络会随时间适应动力学过程。例如，与已感染节点相邻的易感节点可能会想要切断它们之间的边。

5.3.1 具有智能重连的自适应网络

在本节中，我们采用 Gross等人 [2006] 提出的简单重连机制，在一个内部驱动的自适应网络上模拟 SIS 流行病。Gross等人 Gross等人 [2006] 选择离散时间步进行模拟。对于

每个时间步，每个已感染节点以 r 的概率康复，每条活跃边以 p 的概率传播疾病，并以 w 的概率进行重连。本节的目标是使用连续时间模拟复现他们的部分结果。

我们先定义事件发射器。简单来说，重连是指一个易感节点断开与其已感染邻居的连接，然后重新连接到网络中随机选取的一个**易感**节点，同时确保网络仍然是简单图。我们在表5中定义重连事件发射器。

表5： **智能重连的事件发射器**。

	重连
ID	重连($x \to y$)
f	边 (x, y) 已移除; 随机选中节点 z.state=S; 边 (y, z) 已添加;
C	$\emptyset$
R	{感染($x \to y$),重连($x \to y$)}

我们还相应地修改了现有的感染和康复事件发射器。实质上，感染发射器 Inf($x \to y$) 和具有相同签名 Rewire($x \to y$) 的重连发射器应始终同时创建和移除。修改后的发射器如表 6 所示。

表 6：**采用智能重连的 SIS 流行病的感染和康复事件发射器**。

	感染	康复
ID	感染($x \to y$)	恢复(x)
f	y.state ←I (已感染)	x.state ←S (易感)
C	{恢复(y)} ∪ {感染($y \to z$)，重连($y \to z$) $\mid z \in \mathcal{N}(y)$ z.状态=S}	{感染($z \to x$)，重连($z \to x$) $\mid z \in \mathcal{N}(x)$ z.状态=I}
R	{感染($z \to y$)，重连($z \to y$) $\mid z \in \mathcal{N}(y)$ z.状态=I}	{恢复(x)} ∪ {感染($x \to z$)，重连($x \to z$) $\mid z \in \mathcal{N}(x)$ z.状态=S}

为了重现Gross等人 [2006], 的结果，我们假设所有转变均服从具有齐次速率的指数分布，因此适用 makingoptimizedGA(5)。我们将恢复率记为 γ，每条边的感染速率记为 β，重连速率记为 ω。

这种重连机制的一个结果是出现了滞后回线（Gross等人 [2006]）。为了重现这一现象，我们考虑两种不同的初始条件：

- 一个初始感染节点占比为 1% 的 Erdos-Renyi $G(N, p)$ 网络。
- 一个经过大量重连且已形成流行病传播的网络。

要生成第二个初始条件，我们只需从第一个初始条件开始模拟，并在系统达到地方病平衡态后保存网络状态。对于第一个初始条件，我们尝试不同的感染率，并记录平衡态下的流行病规模，以确定入侵阈值，即新发流行病扩散至人群所需的最低感染率。对于第二个初始条件，我们采用相同的方法来确定持续阈值，即已形成的流行病得以持续所需的最低感染率。结果汇总于图 6。

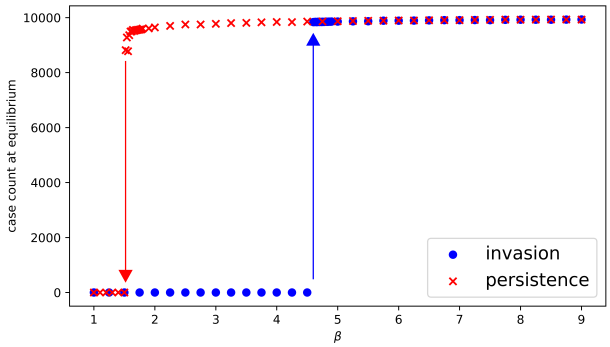

图6：**带有重连机制的滞后环**。 $\gamma = 1, \omega = 100, N = 10^4$，平均度$= 20$。蓝点是从第一个初始条件出发，使用不同感染率运行模拟得到的。红点是从第二个初始条件出发得到的。对于这两种情况，我们都记录平衡态下的平均病例数。如果超过90%的重复运行因流行病在达到给定迭代次数前消退而提前终止，则将平衡态下的平均病例数设为零；否则，取所有存活运行结果的平均值。入侵阈值约为4.6。持续阈值约为1.52。这些数值与Gross等人 [2006]缩放后的结果接近。

根据Gross等人 [2006]，，当重连率较高（例如为 $\omega = 300$）时，如果我们从第二个初始条件开始，随着 β 减小，原有平衡态将失去稳定性，系统随之进入稳定极限环。这种现象称为Hopf分岔Strogatz [2018]。我们成功复现了这一现象，并在图 7 中加以说明。

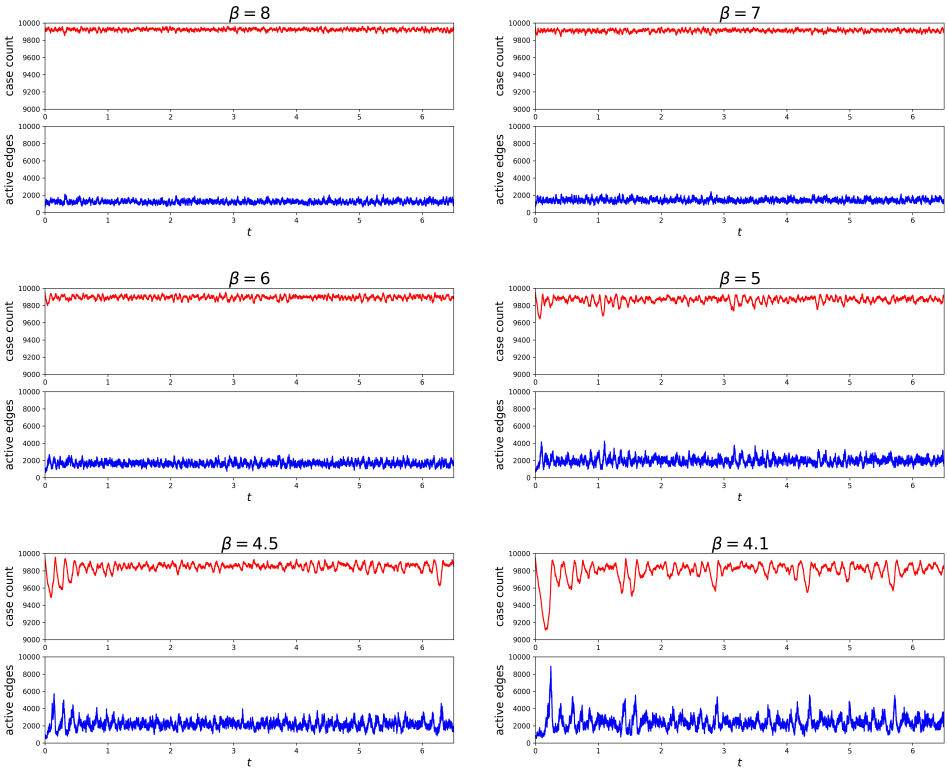

图7：**高重连率下的Hopf分岔**。 所有轨迹均从同一个经过大量重连、带有疾病的网络出发，该网络具有 $\gamma = 1, \beta = 16, \omega = 300$。持续阈值约为4.1。当重连率固定为300时，随着感染率 β 降低，振荡行为愈发明显。

5.3.2 带通勤的时序网络

本节中，我们在一个外部驱动时序网络上模拟 SIS 流行病，即家庭与学校之间的通勤模式。我们提出的模型受到新冠肺炎疫情期间许多学校和工作场所采用的混合模式启发：每天只有一部分人员可以到场，其余人员则须在家学习或工作。我们进行了两组实验。首先，我们测量三种算法（NRM (1)、以节点为中心的遗传算法 (4)、优化遗传算法 (5)）在不同规模图上的性能。随后，我们探究改变出勤概率（允许到场的人数比例）所产生的影响。在两组实验中，我们将逐边感染率设为 0.03，恢复率设为 0.25。

我们分别生成一张“学校”图和一张“家庭”图。两张图生成后均保持不变。当某个节点从一个网络转移到另一个网络时，它在第一个网络中会被标记为“关闭”，并在另一个网络中以其先前的状态（已感染/易感）标记为活动状态。在某个网络中被标记为“关闭”的节点不会参与该网络上的流行病过程，对其邻居节点而言实际上不可见。这种关闭机制使我们能够在不更改底层图数据结构的情况下将节点从一个网络移至另一个网络，从而提高实现效率。不失一般性地假设，模拟开始时所有节点都在家中。12 小时后，每个在家的节点都会以此前所述的出勤概率前往学校，使流行病过程能够在这两个图实际上各自对应的子图上继续进行。再过 12 小时，所有在校节点都会离校回家，完成一个周期。除通勤模式外，该模拟与标准的 SIS 流行病模拟完全相同。

在实现方面，NRM 只需进行少量调整，因为通勤模式可以通过两个相互更新的事件发射器“Home2School”和“School2Home”来实现，其事件间隔分布为狄拉克 δ 函数。对于以节点为中心的遗传算法和优化遗传算法，我们借鉴了 Vestergaard 和 Génois [2015] 的改进方法。每次迭代中，我们按照经典的 Gillespie 方法生成下一事件发生的时间。然后，将下一事件发生的时间与下一次通勤时间进行比较。如果下一事件发生的时间更短，就会发生该事件，无论是感染还是康复。否则，该事件将被拒绝，通勤则会发生。
在第一个实验中，我们从所有节点中均匀随机选取 5% 作为初始已感染节点。在模拟世界中，我们将每种算法运行至 60 天窗口结束，出勤概率设为 1/3。假设“家庭”网络为平均度为 10 的 ER 图，“学校”网络为平均度为 100 的 ER 图。平均耗时取 5 次重复运行的平均值。第一个实验的结果汇总于图 8。

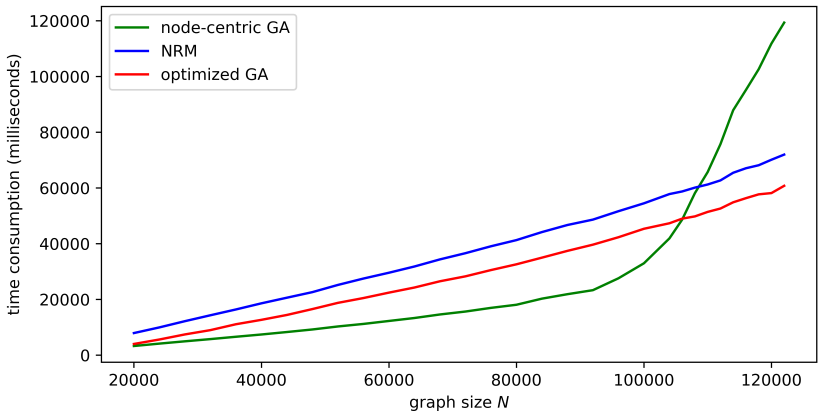

图 8：**在不同规模图上，采用齐次速率的马尔可夫流行病三种算法的性能**。可以看出，NRM 和优化遗传算法的性能相当，不过只有当图规模超过 100000 时，它们相对于以节点为中心的遗传算法的优势才明显。

为了更好地了解这三种算法的性能，我们还使用 callgrind 进行指令计数（统计中央处理器执行的指令数）和缓存分析（统计缓存命中/未命中的次数）。结果汇总于图9和表7中。

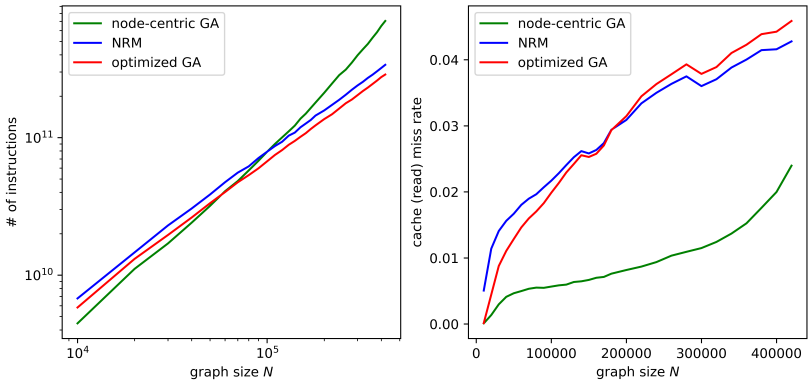

图9：**指令计数与缓存性能分析**。左图为双对数图，其斜率表征算法的复杂度，汇总于表7中。右图展示了缓存读取未命中率随图规模变化的情况，表明随着问题规模增大，缓存性能也会下降。由于实现中不可避免地存在间接寻址，NRM 和优化遗传算法的缓存性能较差；但尽管缓存友好性较低，它们最终仍优于以节点为中心的遗传算法。

表7：**双对数图中割线的斜率（图9）**。

	前半部分	后半部分	总体
以节点为中心的遗传算法 (4)	1.2988	1.6202	1.3544
NRM (1)	1.0481	1.0379	1.0464
优化遗传算法 (5)	1.0465	1.0286	1.0434

表注：NRM (1) 和优化遗传算法 (5) 看起来相当线性，这可能是因为问题规模仍然不够大，NRM 中的 $\log N$ 项尚未产生显著影响。以节点为中心的遗传算法的割线斜率从 1.2988 上升至 1.6202，这与我们此前的讨论一致：其渐近斜率为 2，因为二次部分最终会占据主导地位。

在第二个实验中，种群规模固定为 10000。假设“家庭”网络是一个块大小为 5、初始感染节点比例为 1% 的随机块模型，而“学校”网络是一个块大小为 50 的随机块模型。对于两个网络，如果 i，j 属于不同的块，则边概率 p_{ij} 为零；如果 i，j 属于同一个块，则边概率为一。这种构造表明，通勤起到了远距离连接的作用。每条平均轨迹均由 10 次重复运行取平均得到。第二个实验的结果汇总于图 10 中。

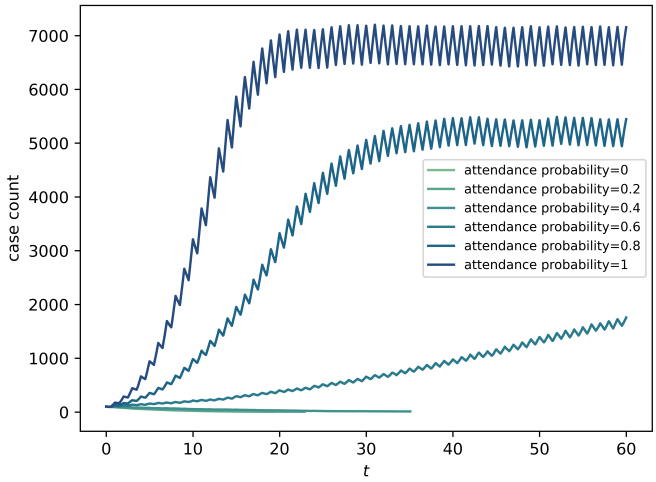

图10：**不同出勤概率下的平均流行病传播轨迹**。 当出勤概率为 0 或 0.2 时，感染率低于流行病阈值。我们看到，在这个特定的网络设置中，出勤概率越高，流行病阈值越低，流行病患病率越高。较高的出勤概率还会导致轨迹的方差更大，因为决定哪些节点应加入“学校”网络的过程是随机的。

5.4 协同感染

到目前为止，我们一直假设通过边发生的感染彼此独立。本节回顾 Boguñá 等人提到的一种协同感染情形，其中节点 x 上的总感染速率表示为

$$\lambda_{x,\text{total}}(t) = \left[\sum_{y \in \mathcal{N}(x), y.\text{state=I}} (\lambda_y(t))^{\frac{1}{\sigma}} \right]^{\sigma},$$

其中，$\mathcal{N}(x)$ 是顶点 x 的一跳邻域。

Boguñá 等人 [2014] 引入了协同感染。这种情形很有意思，因为可以直接应用 nMGA-Approx (3)，而如何应用 NRM 则并不明显。本节将展示，通过重新设计事件发射器并根据风险函数生成暂定时间，不仅可以将 NRM 应用于这种情形，还能使 NRM 的表现优于 nMGA-Approx。

当 $\sigma = 1$ 时，我们便回到经典情形：来自邻居节点的感染彼此独立。否则，仅给定风险函数（此处为 $\lambda_{x,\text{total}}(t)$）时，生成事件间隔时间并非易事。一种方法是利用式 7，这使我们能够通过逆累积分布函数技术：(i) 对 $u \sim \text{Uniform}(0, 1)$ 进行采样，然后 (ii) 求解 $\int_0^t d\tau \lambda(\tau) = -\ln(u)$，得到 t。要计算左侧的积分，可以使用牛顿–柯特斯求积。算法 6 展示了一种简单的实现。请注意，尽管算法 6 使用了主导误差项为 Δt^2 的中点法则，但可能的解只能取自 $\{\frac{\Delta t}{2} + k\Delta t \mid k = 0, 1, 2 \dots\}$，因此主导误差项为 Δt。

算法6 根据风险函数生成事件间隔时间 - 朴素方法

生成 $u \sim$ 均匀分布(0, 1)
$t \leftarrow \Delta t/2$
总和 $\leftarrow 0$
当 总和 $< -\ln u$ **时**
  总和 $\leftarrow$ 总和 $+ \Delta t \cdot \lambda(t)$
  $t \leftarrow t + \Delta t$
结束当
返回 t

一种更精细的方法是在上一个区间的左右边界之间进行插值。更准确地说，对于 $t_i := \frac{\Delta t}{2} + i\Delta t$，我们继续计算黎曼和，直到迭代 n。此时

$$\sum_{i=0}^{n} \lambda(t_i)\Delta t < -\ln u < \sum_{i=0}^{n+1} \lambda(t_i)\Delta t,$$

这意味着， t 的实际解位于（$n+1$）Δt 与（$n+2$）Δt之间的某处。

为求得 t，我们采用一种简单的梯形插值方案，如图 11 所示。该方法保证了求根 $\int_0^t d\tau\lambda$（τ） $= -\ln$（u）能够被精确求解；其中同时涉及中点法则和梯形插值。当 λ（t）为线性函数时，主导误差项 Δt^2。改进后的算法见算法 7。

算法 7 根据风险函数生成事件间隔时间——改进版

生成 $u \sim$ 均匀分布(0, 1)$t \leftarrow \Delta t/2$求和 $\leftarrow 0$**当** 为真**时**新_求和 $\leftarrow$ 求和 $+ \Delta t \cdot \lambda(t)$**如果** 新 求和 $< -\ln u$ **则**求和 $\leftarrow$ 新 求和_$t \leftarrow t + \Delta t$**否则** 中断 **结束 如果 结束 当**lo $\leftarrow t - \Delta t/2$; $h_1 \leftarrow \lambda$(lo) hi $\leftarrow t + \Delta t/2$; $h_2 \leftarrow \lambda$(hi)$D \leftarrow -\ln u -$ 求和; $k \leftarrow (h_2 - h_1)/\Delta t$返回 $t + \frac{-h_1 + \sqrt{h_1^2 + 2Dk}}{k}$

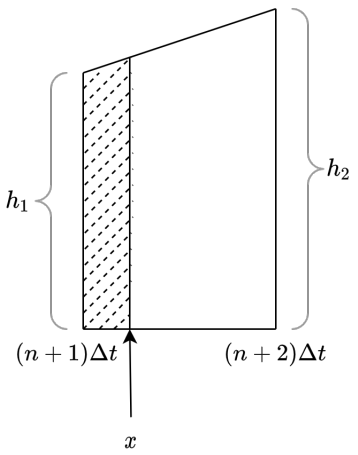

图 11：**梯形插值示意图**。令 $D := -\ln u - \sum_{i=0}^{n} \lambda(t_i)\Delta t > 0$, $h_1 := \lambda((n+1)\Delta t)$,$h_2 := \lambda((n+2)\Delta t)$。问题归结为求出 x，使其位于 $(n+1)\Delta t$ 和 $(n+2)\Delta t$ 之间，并使阴影区域的面积为 D。令 $k := (h_2 - h_1)/\Delta t$。因此， x 的正确值为 $\frac{-h_1 + \sqrt{h_1^2 + 2Dk}}{k}$。

我们还在表 8 中给出了协同感染的事件发射器规范。

表 8：**协同感染（易感-感染-易感）的事件发射器。**

	感染	康复
ID	感染(y)	恢复(x)
f	y.state $\leftarrow$ I	x.state $\leftarrow$S
C	{恢复(y)} $\cup$ {感染(z) $\mid z \in \mathcal{N}(y)$ z.状态=S}	{感染(x)} $\cup$ {感染(z) $\mid z \in \mathcal{N}(x)$ z.状态=易感}
R	{感染(y)} $\cup$ {感染(z) $\mid z \in \mathcal{N}(y)$ z.状态=易感}	{恢复(x)} $\cup$ {感染(z) $\mid z \in \mathcal{N}(x)$ z.状态=易感}

表注：每当在节点 x上执行一个事件（感染/康复）时，对于它的每个易感邻居 y，感染事件发射器感染(y)都会被替换，因为 y的邻域的风险函数已发生变化。

我们运行模拟，　　根据算法 7 和表 8，在不同规模且平均度为 6 的 ER 图上，　　f
初始感染节点为10个，占5%。结果如图12所示。

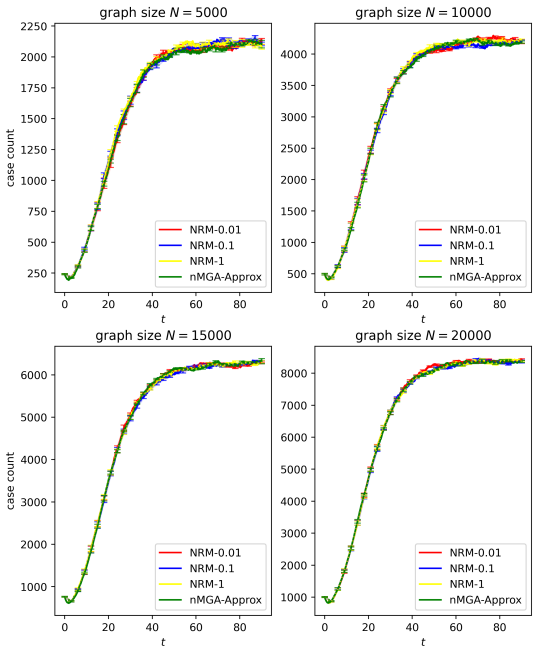

图12：**不同图规模下的轨迹**。每条轨迹均为5次重复运行的平均值。感染事件间隔分布设置为威布尔分布($\kappa = 2$, $\lambda = 10$)。恢复分布为指数分布(0.25)。合作参数 σ 为0.5。当某个算法被称为"NRM-Δt"时，表示该算法使用步长 Δt运行NRM。

图12中的算法呈现出无法区分的轨迹。然而，各算法的性能差异很大。在图规模为10000时，nMGA-Approx的运行时间约为5分钟；在图规模为20000时，运行时间约为25分钟。而NRM-0.01在图规模为10000时耗时4.3秒，在图规模为20000时耗时8.8秒。随着 Δt变大，时间消耗降低，因为在所有考虑的图规模下，NRM-0.1和NRM-1的运行时间都远低于1秒。

性能差异背后的原因在于，NRM 的主导误差项为 Δt^2，因此只需进行更少次数的速率函数求值，就能达到相同的精度。值得注意的是，中点法则只是许多可应用于此处的求积法则之一。只要插值步骤的主导误差项与数值积分的主导误差项相同，也可以采用其他主导误差项更小的求积法则。

6 讨论

我们回顾了非马尔可夫型和马尔可夫型流行病的模拟算法。我们了解到

- 当模拟任务涉及非马尔可夫型过程，或涉及具有异质速率的马尔可夫型过程时，NRM (1) 是最佳选择。
- 当模拟任务涉及具有齐次速率的马尔可夫型过程时，优化遗传算法 (5) 是最佳选择。
- 通过合理设计事件发射器，NRM 和优化遗传算法都可以适用于更复杂的场景，例如时变网络和协同感染。
- 当图规模较小时，以节点为中心的遗传算法 (4) 可能优于其他算法（如图 8 所示），尽管其渐近表现并非最优（图 8）。

使用事件发射器进行模拟也存在局限性。首先，必须明确识别模拟任务中的独立组成部分。例如，在协同感染场景中，当 $\sigma \neq 1$ 时，来自不同已感染邻居、作用于同一易感节点的感染并非独立事件，因此不再适用边级感染发射器 Inf(x, y)。与此同时，不同易感节点受到感染是相互独立的事件，因此我们改用 Inf(x)。尽管这会增加早期设计过程的复杂性，但我们认为，这能促使从业者明确说明其关于独立性的假设。其次，创建/移除的事件发射器数量并不总是能达到 $\mathcal{O}(1)$。例如，考虑在完全图上模拟 SIS 流行病。如果创建/移除的事件发射器数量为 $\mathcal{O}(N)$，那么单次迭代成本至少为 $\mathcal{O}(N)$。

事件发射器分组技术也体现了空间复杂度与时间复杂度之间的权衡。假设图有 V 个顶点和 E 条边。对于以节点为中心的遗传算法，空间复杂度为 $\mathcal{O}(V)$，因为需要计算并存储每个节点的速率。对于优化遗传算法，空间复杂度为 $\mathcal{O}(V + E)$，因为必须维护所有活跃边的哈希集合。

7 支持信息

S1 附录。 实际上，已用时间的记录方式是非马尔可夫型模拟中的一种设计选择。要了解原因，请考虑图 13 中的情形。

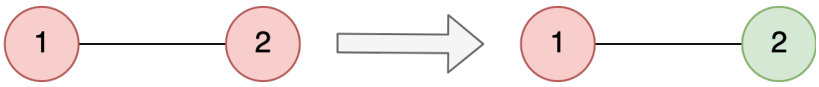

图 13： **两种不同的已用时间记录规则**。

随着 2 的康复，需要引入事件发射器 Inf($1 \to 2$)。然而，感染时间可以采用以下任一规则：

$$\begin{cases} T_{\text{inf}} \sim \psi(\cdot \mid 0) & \text{rule 1} \\ T_{\text{inf}} \sim \psi(\cdot \mid t_1) & \text{rule 2,} \end{cases}$$

其中 t_1 是节点1患病的持续时间。规则1更容易实现，而规则2更贴近现实，特别是当我们认为非马尔可夫型感染的根本原因是症状随时间的发展，而这会导致传染性发生变化时。幸运的是，规则2只需对表1稍作修改。

表 9： **网络上 SIS 流行病的感染数 Inf($x \to y$)，采用规则2。**

	感染
ID	Inf($x \to y$)
f	y.state ←I y.已感染_以来 ← t_{exe}
C	{恢复(y)} ∪ {感染($y \to z$) $\mid z \in \mathcal{N}(y)$ z.状态=S}
R	{感染($z \to y$) $\mid z \in \mathcal{N}(y)$ z.状态=I}
Ψ	Ψ($\cdot \mid t_{\text{intro}} - x$.已感染_以来)

t_{exe}：感染($x \to y$)执行的时间（即 y 被 x感染的时间）。t_{intro}：感染($x \to y$)由其他事件发射器创建的时间。

这项额外的记录工作确保每个已感染节点都“知道”自己何时被感染。在实践中，这只需对代码做极少的改动，而且可以轻松开启或关闭。因此，本文所有非马尔可夫型模拟均采用规则2。

附录S2。

定理 *1* 的证明。 我们通过对 l进行归纳法来证明该定理。

基础情形 $l = 1$
当 $l = 1$时，所有暂定时间都是新生成的，因此该定理是假设 1 的直接推论。

归纳情形 $l \geq 2$
为便于论证，假设该定理对所有 $l' < l$ 都成立，即 $\forall l' < l$， $\{T_i^{l'}\}_{i \in E^{l'}}$在给定 $\overrightarrow{kt}^{l'-1}$ 的条件下由相互独立的随机变量构成。特别地，由归纳假设可知， $\{T_i^{l-1}\}_{i \in E^{l-1}}$ 在给定 $\overrightarrow{kt}^{l-2}$ 的条件下相互独立。

给定 $\overrightarrow{kt}^{l-1}$，我们确定 E^l， $\mathcal{R}^l$。若 $\mathcal{R}^l = \emptyset$，则迭代 l 中的所有暂定时间均为新生成，因此该定理可直接由假设 1 得出。故我们只考虑非平凡情形，即 $\mathcal{R}^l \neq \emptyset$。
设 $\mathcal{J}$为 E^l的任意子集，由此可定义 $\mathcal{J}^l := \mathcal{R}^l \cap \mathcal{J}$， $\bar{\mathcal{J}}^l := \mathcal{J} \setminus \mathcal{J}^l$。根据假设 1，我们将联合累积分布函数分解为

$$\Pr\{\vec{T}^l_{\mathcal{J}} \leq \vec{t}_{\mathcal{J}}\} = \left(\prod_{i \in \bar{\mathcal{J}}^l} \Pr\{T_i^l \leq t_i \mid \overrightarrow{kt}^{l-1}\} \right) \cdot \Pr\{\vec{T}^l_{\mathcal{J}^l} \leq \vec{t}_{\mathcal{J}^l} \mid \overrightarrow{kt}^{l-1}\}. \tag{8}$$

如果 $\mathcal{J}^l = \emptyset$，也就是说，如果 $\mathcal{J}$ 中所有暂定时间都是在第 l 次迭代中新生成的，那么公式（8）中的第二个因子就会消失，因式分解也就已经完成。因此，我们只考虑 $\mathcal{J}^l \neq \emptyset$ 的非平凡情况，并继续

对第二个因子进行因式分解。

$$\Pr\{\vec{T}^{l}_{\mathcal{J}^l} \le \vec{t}_{\mathcal{J}^l} \mid \overrightarrow{kt}^{l-1}\} = \Pr\{\vec{T}^{l-1}_{\mathcal{J}^l} \le \vec{t}_{\mathcal{J}^l} \mid \overrightarrow{kt}^{l-1}\} \qquad (\mathcal{J}^l \text{ is reused}) \tag{9}$$

$$= \Pr\{\vec{T}^{l-1}_{\mathcal{J}^l} \le \vec{t}_{\mathcal{J}^l} \mid \overrightarrow{kt}^{l-2}, K^{l-1}=k^{l-1}, T^{l-1}=t^{l-1}\} \qquad (\text{Defn. 4})$$

$$= \lim_{\epsilon\to 0} \frac{\Pr\{\vec{T}^{l-1}_{\mathcal{J}^l} \le \vec{t}_{\mathcal{J}^l}, K^{l-1}=k^{l-1}, t^{l-1}-\epsilon \le T^{l-1} \le t^{l-1} \mid \overrightarrow{kt}^{l-2}\}}{\Pr\{K^{l-1}=k^{l-1}, t^{l-1}-\epsilon \le T^{l-1} \le t^{l-1} \mid \overrightarrow{kt}^{l-2}\}} \tag{10}$$

$$= \lim_{\epsilon\to 0} \frac{\Pr\{\vec{T}^{l-1}_{\mathcal{J}^l} \le \vec{t}_{\mathcal{J}^l};\ \vec{T}^{l-1}_{E^{l-1}\setminus\{k^{l-1}\}} > t^{l-1};\ t^{l-1}-\epsilon \le T^{l-1}_{k^{l-1}} \le t^{l-1} \mid \overrightarrow{kt}^{l-2}\}}{\Pr\{\vec{T}^{l-1}_{E^{l-1}\setminus\{k^{l-1}\}} > t^{l-1};\ t^{l-1}-\epsilon \le T^{l-1}_{k^{l-1}} \le t^{l-1} \mid \overrightarrow{kt}^{l-2}\}} \tag{11}$$

$$= \lim_{\epsilon\to 0} \frac{\Pr\{t^{l-1} < \vec{T}^{l-1}_{\mathcal{J}^l} \le \vec{t}_{\mathcal{J}^l};\ \vec{T}^{l-1}_{E^{l-1}\setminus(\mathcal{J}^l\cup\{k^{l-1}\})} > t^{l-1};\ t^{l-1}-\epsilon \le T^{l-1}_{k^{l-1}} \le t^{l-1} \mid \overrightarrow{kt}^{l-2}\}}{\Pr\{\vec{T}^{l-1}_{\mathcal{J}^l} > t^{l-1};\ \vec{T}^{l-1}_{E^{l-1}\setminus(\mathcal{J}^l\cup\{k^{l-1}\})} > t^{l-1};\ t^{l-1}-\epsilon \le T^{l-1}_{k^{l-1}} \le t^{l-1} \mid \overrightarrow{kt}^{l-2}\}} \tag{12}$$

$$= \lim_{\epsilon\to 0} \frac{\Pr\{t^{l-1} < \vec{T}^{l-1}_{\mathcal{J}^l} \le \vec{t}_{\mathcal{J}^l};\ \vec{T}^{l-1}_{E^{l-1}\setminus(\mathcal{J}^l\cup\{k^{l-1}\})} > t^{l-1};\ t^{l-1}-\epsilon \le T^{l-1}_{k^{l-1}} \le t^{l-1} \mid \overrightarrow{kt}^{l-2}\}}{\Pr\{\vec{T}^{l-1}_{\mathcal{J}^l} > t^{l-1};\ \vec{T}^{l-1}_{E^{l-1}\setminus(\mathcal{J}^l\cup\{k^{l-1}\})} > t^{l-1};\ t^{l-1}-\epsilon \le T^{l-1}_{k^{l-1}} \le t^{l-1} \mid \overrightarrow{kt}^{l-2}\}}$$

$$= \frac{\Pr\{t^{l-1} < \vec{T}^{l-1}_{\mathcal{J}^l} \le \vec{t}_{\mathcal{J}^l} \mid \overrightarrow{kt}^{l-2}\}}{\Pr\{\vec{T}^{l-1}_{\mathcal{J}^l} > t^{l-1} \mid \overrightarrow{kt}^{l-2}\}} \tag{13}$$

$$= \prod_{j\in\mathcal{J}^l} \frac{\Pr\{t^{l-1} < T^{l-1}_j \le t_j \mid \overrightarrow{kt}^{l-2}\}}{\Pr\{T^{l-1}_j > t^{l-1} \mid \overrightarrow{kt}^{l-2}\}} \tag{14}$$

$$= \prod_{j\in\mathcal{J}^l} \Pr\{T^{l-1}_j \le t_j \mid T^{l-1}_j > t^{l-1}, \overrightarrow{kt}^{l-2}\}, \tag{15}$$

其中，公式（10 $T^{l-1}\le t^{l-1}$ 11 $K^{l-1}=k^{l-1}\ t^{l-1}-\epsilon\le$）本质上是条件概率的定义，公式（12）改写了事件，，而公式（10 $\epsilon\to 0$）只是重新对指标进行分组。在公式（0 0）中引入 lim，以避免出现“/”。假设所有 T^l_i 均为连续随机变量，我们知道该极限存在。公式（13）和公式（14）使用了归纳假设。

由于我们已假设 $\mathcal{R}^l \neq \emptyset$，因此取 $j \in \mathcal{R}^l$ 并选择 $\mathcal{J} = \mathcal{J}^l = \{j\}$。由于 $\mathcal{J} \subset E^l$ 的选择是任意的，从公式（9）到公式（15）的推理仍然适用，从而得到

$$\Pr\{T^l_j \le t \mid \overrightarrow{kt}^{l-1}\} = \Pr\{T^{l-1}_j \le t \mid T^{l-1}_j > t^{l-1}, \overrightarrow{kt}^{l-2}\}. \tag{16}$$

将公式（16）代回公式（15），得到

$$\Pr\{\vec{T}^{l}_{\mathcal{J}^l} \le \vec{t}_{\mathcal{J}^l} \mid \overrightarrow{kt}^{l-1}\} = \prod_{j\in\mathcal{J}^l} \Pr\{T^l_j \le t_j \mid \overrightarrow{kt}^{l-1}\}. \tag{17}$$

将公式（17）代入公式（8），因式分解即告完成，这表明它们相互独立，从而完成证明。

□

S3附录。

引理 *1* 的证明。再次，我们通过对 l 进行归纳法来证明引理 1。

- **基础情形** $l = 1$ T^1_i 只能是新生成的，这意味着 $t^1_i = t^0 = 0$。公式（1）由此可得

$$\Pr\{T^1_i \le t \mid \overrightarrow{kt}^{l-1}\} = \Psi_i(t).$$

- **归纳情形** $l \ge 2$

  假设 $\forall l' < l$，$\Pr\{T^{l'}_i \le t \mid -\quad \overrightarrow{kt}^{l-1}\}$ 的形式如式 **(1)** 所示。

  – 如果 T^l_i 是新生成的，那么 $t^l_i = t^{l-1}$，这意味着 $\Psi_i(t^{l-1} - t^l_i) = 0$。因此，

$$\Pr\{T^l_i \le t \mid \overrightarrow{kt}^{l-1}\} = \Psi_i(t - t^{l-1}) = \frac{\Psi_i(t - t^l_i) - \Psi_i(t^{l-1} - t^l_i)}{1 - \Psi_i(t^{l-1} - t^l_i)}.$$

– 否则，如果 T_i^l 不是新生成的，也就是说，事件发射器 i 已存在于迭代 $l-1$中（因此 $t_i^{l-1}=t_i^l$），我们可以令 $s:=t_i^{l-1}=t_i^l$。根据归纳假设， T_i^{l-1} 满足

$$\Pr\{T_i^{l-1}\le t\mid \overrightarrow{kt}^{l-2}\}=\frac{\Psi_i(t-s)-\Psi_i(t^{l-2}-s)}{1-\Psi_i(t^{l-2}-s)}.$$

感谢推论 1，

$$\begin{aligned}\Pr\{T_i^l\le t\mid \overrightarrow{kt}^{l-1}\}=&\Pr\{T_i^{l-1}\le t\mid T_i^{l-1}>t^{l-1},\overrightarrow{kt}^{l-2}\}\\=&\frac{\Pr\{T_i^{l-1}\le t\mid \overrightarrow{kt}^{l-2}\}-\Pr\{T_i^{l-1}\le t^{l-1}\mid \overrightarrow{kt}^{l-2}\}}{1-\Pr\{T_i^{l-1}\le t^{l-1}\mid \overrightarrow{kt}^{l-2}\}}\\=&\frac{\Psi_i(t-s)-\Psi_i(t^{l-1}-s)}{1-\Psi_i(t^{l-1}-s)},\end{aligned}$$

这正是我们所需要的。

□

附录 S4。

定理 *2* 的证明。给定截至 $l-1$ 的先前轨迹，考虑 NRM 的暂定时间 $\{T_i^l\}_{i\in E^l}$。我们有：

- $\{T_i^l\}_{i\in E^l}$ are mutually independent given $\overrightarrow{kt}^{l-1}$ (Theorem 1).
- $\Pr\{T_i^l\le t\mid \overrightarrow{kt}^{l-1}\}=\Psi_i(t-t^{l-1}\mid t^{l-1}-t_i^l)$ (Eq (3)).

对于 NRM，第 l 次执行的时间为 $T^l:=\min_i T_i^l$。我们得到

$$\begin{aligned}\Pr\{T^l>t\mid \overrightarrow{kt}^{l-1}\}=&\Pr\{\min_i T_i^l>t\mid \overrightarrow{kt}^{l-1}\}=\Pr\{T_{E^l}^l>t\mid \overrightarrow{kt}^{l-1}\}\\=&\prod_i\Pr\{T_i^l>t\mid \overrightarrow{kt}^{l-1}\}=\prod_i\left[1-\Psi_i(t-t^{l-1}\mid t^{l-1}-t_i^l)\right].\end{aligned}$$

与此同时，对于 nMGA，第 l 次执行的时间 $\tilde{T}^l:=t^{l-1}+\Delta^l$ 满足

$$\begin{aligned}\Pr\{\tilde{T}^l>t\mid \overrightarrow{kt}^{l-1}\}=&\Pr\{\Delta^l+t^{l-1}>t\mid \overrightarrow{kt}^{l-1}\}=\Pr\{\Delta^l>t-t^{l-1}\mid \overrightarrow{kt}^{l-1}\}\\=&\Phi^l(t-t^{l-1})=\prod_i\left[1-\Psi_i(t-t^{l-1}\mid t^{l-1}-t_i^l)\right]\\=&\Pr\{T^l>t\mid \overrightarrow{kt}^{l-1}\},\end{aligned}$$

以及这一事实： $\forall t$，$\Pr\{T^l>t\mid \overrightarrow{kt}^{l-1}\}=\Pr\{\tilde{T}^l>t\mid \overrightarrow{kt}^{l-1}\}$ 意味着 $f_{T^l}\equiv\tilde{f}_{T^l}$。我们看到，执行时间 T^l 和 $\tilde{T}^l$ 确实具有相同的（边缘）分布。接下来，我们研究 NRM 中给定轨迹 $\overrightarrow{kt}^{l-1}$ 和 $T^l=t^l$ 时 K^l 的分布。

$$\begin{aligned}\Pi^l(k^l\mid t^l):=&\Pr\{K^l=k^l\mid T^l=t^l,\overrightarrow{kt}^{l-1}\}\\=&\lim_{\epsilon\to0}\frac{\Pr\{K^l=k^l,t^l\le T^l<t^l+\epsilon\mid \overrightarrow{kt}^{l-1}\}}{\Pr\{t^l\le T^l<t^l+\epsilon\mid \overrightarrow{kt}^{l-1}\}}\\=&\lim_{\epsilon\to0}\frac{\Pr\{K^l=k^l,t^l\le T^l<t^l+\epsilon\mid \overrightarrow{kt}^{l-1}\}}{\sum_k\Pr\{K^l=k,t^l\le T^l<t^l+\epsilon\mid \overrightarrow{kt}^{l-1}\}},\end{aligned}$$

其中，每个单独的 $\Pr\{K^l = k, t^l \le T^l < t^l + \epsilon \mid \overrightarrow{kt}^{l-1}\}$ 都可以展开为

$$\begin{aligned}
&\Pr\{K^l = k, t^l \le T^l < t^l + \epsilon \mid \overrightarrow{kt}^{l-1}\} = \Pr\{t^l \le T_k^l < t^l + \epsilon;\ \forall j \neq k, T_j^l > t^l \mid \overrightarrow{kt}^{l-1}\} \\
=&\psi_k(t^l - t^{l-1} \mid t^{l-1} - t_k^l) \cdot \epsilon \cdot \prod_{j \neq k} \left[1 - \Psi_j(t^l - t^{l-1} \mid t^{l-1} - t_j^l)\right] \quad \text{(Thm. 1, Lemma 1)} \\
=&\frac{\psi_k(t^l - t^{l-1} \mid t^{l-1} - t_k^l)}{1 - \Psi_k(t^l - t^{l-1} \mid t^{l-1} - t_k^l)} \cdot \epsilon \cdot \prod_{j} \left[1 - \Psi_j(t^l - t^{l-1} \mid t^{l-1} - t_j^l)\right] \\
=&\frac{\psi_k(t^l - t_k^l)}{1 - \Psi_k(t^l - t_k^l)} \cdot \epsilon \cdot \prod_{j} \left[1 - \Psi_j(t^l - t^{l-1} \mid t^{l-1} - t_j^l)\right] \\
=&\psi_k(0 \mid t^l - t_k^l) \cdot \epsilon \cdot \underbrace{\prod_{j} \left[1 - \Psi_j(t^l - t^{l-1} \mid t^{l-1} - t_j^l)\right]}_{\text{not dependent on } k}. \qquad (18)
\end{aligned}$$

由于 ϵ 和式（18）中不依赖于 k 的因子均相互抵消，对于 NRM，我们有

$$\Pi^l(k^l) = \frac{\psi_{k^l}(0 \mid t^l - t_{k^l}^l)}{\sum_k \psi_k(0 \mid t^l - t_k^l)},$$

这与算法 2 和式（5）中 nMGA 规定的概率质量函数权重相同。

总之，

$$\left.\begin{array}{l} f_{T^l} \equiv \tilde{f}_{T^l} \\ \Pi^l \equiv \tilde{\Pi}^l \end{array}\right\} \implies f^{\text{NRM}}_{T^l, K^l}(\cdot \mid \overrightarrow{kt}^{l-1}) \equiv f^{\text{nMGA}}_{T^l, K^l}(\cdot \mid \overrightarrow{kt}^{l-1}),$$

这表明二者等价，从而完成证明。 □

参考文献

Neil M Ferguson、Daniel Laydon、Gemma Nedjati-Gilani、Natsuko Imai、Kylie Ainslie、Marc Baguelin、Sangeeta Bhatia、Adhiratha Boonyasiri、Zulma Cucunubá、Gina Cuomo-Dannenburg 等。非药物干预措施（npis）对降低新冠肺炎死亡率和医疗需求的影响。帝国理工学院新冠肺炎应对团队。帝国理工学院新冠肺炎应对团队，第20页，2020年。

István Z Kiss、Joel C Miller、Péter L Simon 等。网络上的流行病数学。尚贝里：施普林格，598，2017年。

William Ogilvy Kermack 和 Anderson G McKendrick。对流行病数学理论的贡献。伦敦皇家学会会刊。*A*辑，数学与物理特征论文集，115(772):700–721，1927年。

Alun L. Lloyd. 流行病模型中传染期的现实分布：持续性与动力学模式的变化。《理论种群生物学》，60(1):59–71，2001年。ISSN 00405809。doi:10.1006/tpbi.2001.1525.

Qing-Bin Lu、Yong Zhang、Ming-Jin Liu、Hai-Yang Zhang、Neda Jalali、An-Ran Zhang、Jia-Chen Li、Han Zhao、Qian-Qian Song、Tian-Shuo Zhao 等。新冠肺炎的流行病学参数及其对中国患者传染性的影响，2020年1月1日至2月11日。《欧洲监测》，25(40):2000250，2020年。

Marcel Salathe、Linus Bengtsson、Todd J Bodnar、Devon D Brewer、John S Brownstein、Caroline Buckee、Ellsworth M Campbell、Ciro Cattuto、Shashank Khandelwal、Patricia L Mabry 等。数字流行病学。*PLoS*计算生物学，8 (7):e1002616，2012年。

Luca Ferretti、Chris Wymant、Michelle Kendall、Lele Zhao、Anel Nurtay、Lucie Abeler-Dörner、Michael Parker、David Bonsall 和 Christophe Fraser。量化 SARS-CoV-2传播表明，数字接触者追踪有助于控制流行病。《科学》，368(6491)，2020年。

Daniel T Gillespie。数值模拟耦合化学反应随机时间演化的一般方法。《计算物理学杂志》，22(4):403–434，1976年。

Michael A. Gibson 和 Jehoshua Bruck。对含有多种物种和众多通道的化学体系进行高效精确的随机模拟。《物理化学杂志*A*辑》，104(9):1876–1889，2000年。ISSN 10895639。doi:10.1021/jp993732q。

Marian Boguñá、Luis F. Lafuerza、Raúl Toral 和 M. Ángeles Serrano。《非马尔可夫随机过程的模拟》。*Physical Review E - Statistical, Nonlinear, andSoft Matter Physics*，90(4):1–9，2014年。ISSN 15502376。doi:10.1103/PhysRevE.90.042108。

Christian L. Vestergaard 和 Mathieu Génois。《时间吉莱斯皮算法：在时变网络上快速模拟传染过程》。*PLoS*计算生物学，11(10):1–28，2015年。ISSN 15537358。doi:10.1371/journal.pcbi.1004579。Richard P Brent。《无导数最小化算法》。Courier Corporation，2013年。Pauli Virtanen、Ralf Gommers、Travis E. Oliphant、Matt Haberland、Tyler Reddy、David Cournapeau、Evgeni Burovski、Pearu Peterson、Warren Weckesser、Jonathan Bright、Stéfan J. van der Walt、Matthew Brett、Joshua Wilson、K. Jarrod Millman、Nikolay Mayorov、Andrew R. J. Nelson、Eric Jones、Robert Kern、Eric Larson、C J Carey、 ̇Ilhan Polat、Yu Feng、Eric W. Moore、Jake VanderPlas、Denis Laxalde、Josef Perktold、Robert Cimrman、Ian Henriksen、E. A. Quintero、Charles R. Harris、Anne M. Archibald、Antônio H. Ribeiro、Fabian Pedregosa、Paul van Mulbregt 和 SciPy 1.0贡献者。《SciPy 1.0：Python科学计算的基础算法》。*Nature Methods*，17:261–272，2020年。doi:10.1038/s41592-019-0686-2。Silvio C Ferreira、Claudio Castellano 和 Romualdo Pastor-Satorras。网络上易感-感染-易感模型的流行病阈值：数值结果与理论结果的比较。*Physical Review E*，86(4):041125，2012年。Bo Qu 和 Huijuan Wang。异质性感染率下的易感者-感染者-易感者疫情传播。*IEEE*网络科学与工程汇刊，4(3):177–186，2017年。Neil M Ferguson、Derek AT Cummings、Simon Cauchemez、Christophe Fraser、Steven Riley、Aronrag Meeyai、Sopon Iamsirithaworn 和 Donald S Burke。东南亚遏制新发流感大流行的策略。《自然》，437(7056):209–214，2005年。Camila Buono、Federico Vazquez、Pablo Alejandro Macri 和 LA Braunstein。异质性感染率人群中的缓慢疫情消亡。*Physical ReviewE*，88(2):022813，2013年。Paul Erd˝os 和 Alfréd Rényi。随机图的演化。匈牙利科学院数学研究所刊，5(1):17–60，1960年。Thilo Gross、Carlos J Dommar D'Lima 和 Bernd Blasius。自适应网络上的流行病动力学。*Physical Review Letters*，96(20):208701，2006年。Wei Wang、Quan-Hui Liu、Junhao Liang、Yanqing Hu 和 Tao Zhou。复杂网络中的协同演化传播。*PhysicsReports*，820:1–51，2019年。Thilo Gross 和 Bernd Blasius。自适应协同演化网络：综述。*Journal of the Royal Society Interface*，5 (20):259–271，2008年。Steven H Strogatz。《非线性动力学与混沌：及其在物理学、生物学、化学和工程学中的应用》。CRC出版社，2018年。