شبیه‌سازی رایانه‌ای در فیزیک
تمرین چهارم: ول‌گشت

سینا معمر ۹۵۱۰۲۳۱۶

۱۸ شهریور ۱۴۰۰

۱ انحراف از معیار

$$
\begin{aligned}
\langle x^2(t)\rangle &= \langle (x(t-\tau)+al)^2\rangle \\
&= \langle x^2(t-\tau) + 2alx(t-\tau) + a^2l^2\rangle \\
&= \langle x^2(t-\tau)\rangle + 2l\,\langle a\rangle\,\langle x(t-\tau)\rangle + l^2\,\langle a^2\rangle \\
&= \langle x^2(t-\tau)\rangle + 2l(p-q)\,\langle x(t-\tau)\rangle + l^2(p+q) \\
&= \langle x^2(t-\tau)\rangle + 2l(p-q)\Big(\frac{l}{\tau}(p-q)(t-\tau)\Big) + l^2(p+q) \\
&= \langle x^2(t-\tau)\rangle + \frac{2l^2}{\tau}(p-q)^2(t-\tau) + l^2(p+q) \\
&= \langle x^2(t-\tau)\rangle + \frac{2l^2}{\tau}(p-q)^2 t - 2l^2(p-q)^2 + l^2(p+q) \\
&= l^2\frac{t}{\tau}\Big((p-q)^2(\frac{t}{\tau}+1) - 2(p-q)^2 + (p+q)\Big) \qquad (۱)
\end{aligned}
$$

$$
\begin{aligned}
\sigma^2(t) &= \langle x^2(t)\rangle - \langle x(t)\rangle^2 \\
&\overset{(۱)}{\Longrightarrow} = l^2\frac{t}{\tau}\Big((p-q)^2(\frac{t}{\tau}+1) - 2(p-q)^2 + (p+q) - (p-q)^2\frac{t}{\tau}\Big) \\
&= l^2\frac{t}{\tau}\Big((p-q)^2 - 2(p-q)^2 + (p+q)\Big) \\
&= l^2\frac{t}{\tau}\Big((p+q) - (p-q)^2\Big) \\
&= l^2\frac{t}{\tau}\Big(p - p^2 + q - q^2 + 2pq\Big) \\
&= l^2\frac{t}{\tau}\Big(p(1-p) + q(1-q) + 2pq\Big) \\
&= l^2\frac{t}{\tau}\Big(pq + qp + 2pq\Big) \\
&= \frac{4l^2}{\tau}pqt \qquad (۲)
\end{aligned}
$$

۲ ول‌گشت

کد این قسمت از تمرین در فایل q2.py قابل مشاهده است. در ابتدا باید یک object از کلاس RandomWalk بسازیم. سپس تابع render را با زمان و احتمال دل‌خواه فراخوانی می‌کنیم. روش کار این تابع به این صورت است که به اندازه‌ی زمان داده شده، به طور رندوم از بین دو عدد 1 و -1 با احتمال داده‌شده انتخاب می‌کند و آن‌ها را در متغیر self.steps ذخیره می‌کند. سپس برای به دست آوردن مکان نهایی در زمان‌های مورد نظر، تابع calc_positions را صدا می‌زنیم. این تابع قدم‌ها را به صورت تجمیعی جمع می‌کند و در خانه‌ی متناظر با آن زمان ذخیره می‌کند. برای این که به طور آماری میانگین آن را به دست بیاوریم، باید تابع calc_mean_positions را با زمان، طول گام‌ها، احتمال‌ها و تعداد نمونه‌های مورد نظر فراخوانی کنیم. سپس داده‌های به دست آمده را در متغیر sample_positions_ensemble ذخیره کرده و آن را برای استفاده‌های بعدی در فایلی با فرمت npy. می‌نویسیم. واریانس و میانگین داده‌های به دست آمده را برای احتمال‌های 0.2، 0.5 و 0.8 در شکل‌های ۱ تا ۶ می‌توان مشاهده نمود. شیب خط فیت شده به هر یک از این نمودارها در جدول ۱ ذکر شده است.

p	0.2	0.5	0.8
$\frac{l}{\tau}(p-q)$	-0.600	0.000	0.600
$\frac{4l^2}{\tau}pq$	0.640	1.001	0.642

جدول ۱: شیب خطوط فیت شده به نمودارهای میانگین و واریانس مکان ول‌گشت

همان‌طور که دیده می‌شود نتایج به دست آمده، مطابق با پیش‌بینی ما از تئوری هستند.

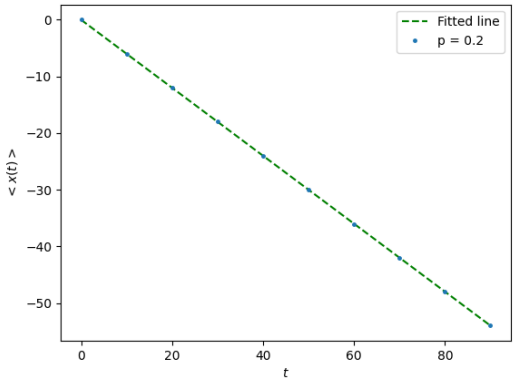
۳ ول‌گشت با تله

کد این بخش از تمرین را در فایل q3.py می‌توان مشاهده نمود. در ابتدا باید یک object از کلاس RandomWalk-With AbsorbingBarrier با طول دلخواه بسازیم. سپس تابع render را با مکان اولیه‌ی دل‌خواه صدا می‌زنیم. روش کار این تابع به این صورت است که از بین اعداد 1 و -1 به طور تصادفی و با احتمال برابر عدد انتخاب می‌کند و با مکان اولیه جمع می‌کند تا زمانی که ول‌گشت به مرزها برسد. سپس زمان به دست آمده را به عنوان طول عمر ول‌گشت ذخیره می‌کند. برای این که به طور آماری میانگین این مقدار را به دست بیاوریم، باید تابع calc_mean_life_time را با طول، گام و تعداد نمونه‌های مورد نظر فراخوانی کنیم. این تابع داده‌های به دست آمده را در یک متغیر ذخیره می‌کند و سپس آن را برای استفاده‌های بعدی، در یک فایل با فرمت npy. می‌نویسد. منحنی به دست آمده برای میانگین طول عمر ول‌گشت بر حسب مکان اولیه را در شکل ۷ می‌توان مشاهده نمود. سهمی فیت شده به این نمودار از رابطه‌ی (۳) پیروی می‌کند.
اجرای این کد برای 100 000 تکرار، 137 ثانیه زمان برد.

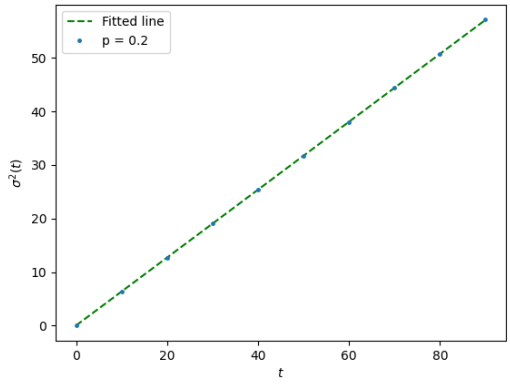
$$\langle \text{life time} \rangle = x_0(18.98 - x_0) + 0.26 \qquad (۳)$$

۴ ول‌گشت با تله (الگوریتم تعینی)

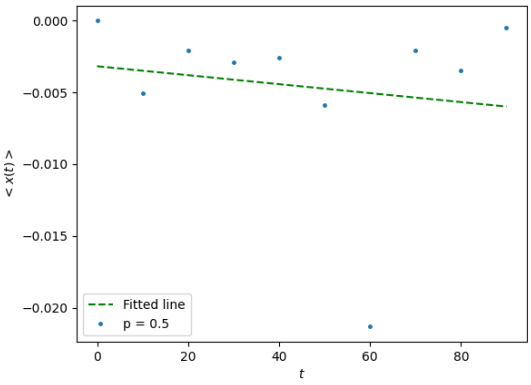
کد این بخش از تمرین را در فایل q4.py می توان مشاهده نمود. ابتدا باید یک object از کلاس RandomWalkWith AbsorbingBarrier بسازیم. سپس تابع render را با مکان اولیه‌ی دل‌خواه صدا می‌زنیم. روش کار این تابع به این صورت است که ابتدا یک لیست از احتمال حضور ول‌گشت در هر خانه می‌سازد. سپس در هر مرحله احتمال هر خانه را نصف کرده و به دو خانه‌ی کناری خود در یک لیست جدید اضافه می‌کند. مقدار احتمال‌هایی که در هر مرحله به مرزها داده می‌شود، احتمال مرگ ول‌گشت در آن زمان است. پس زمان را در این مقدار ضرب کرده و به میانگین عمر ول‌گشت اضافه می‌کنیم. این کار را تا جایی انجام می‌دهیم که مجموع احتمال‌های مرگ، تا حد ممکن به 1 نزدیک شود. برای به دست آوردن تغییرات متوسط طول عمر بر حسب مکان اولیه، تابع calc_mean_life_time را با طول و گام مورد نظر فراخوانی کنیم. این تابع داده‌های به دست آمده را در یک متغیر ذخیره می‌کند و سپس آن را برای



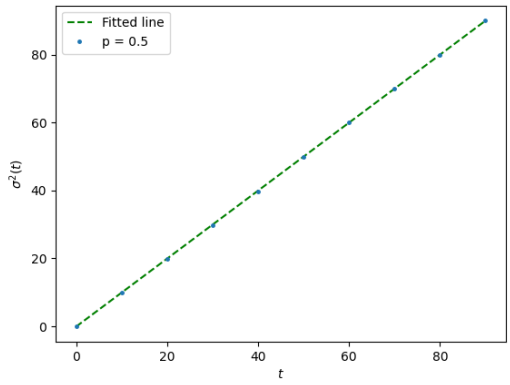
شکل ۱: تغییرات میانگین مکان بر حسب زمان برای $p = 0.2$، $l = 1$، $\tau = 1$ و با 100 000 بار تکرار



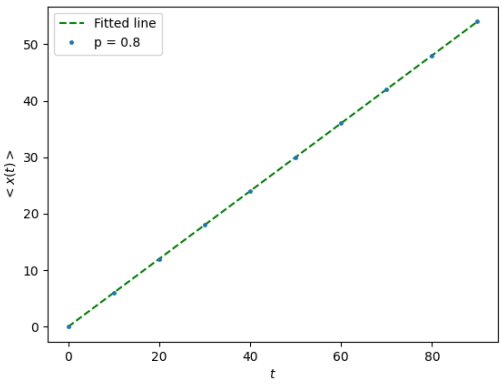
شکل ۲: تغییرات واریانس مکان بر حسب زمان برای $p = 0.2$، $l = 1$، $\tau = 1$ و با 100 000 بار تکرار



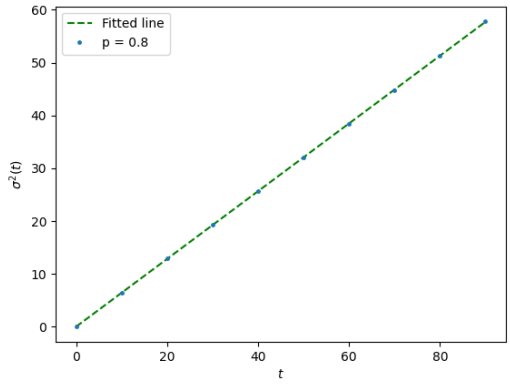
شکل ۳: تغییرات میانگین مکان بر حسب زمان برای $p = 0.5$، $l = 1$، $\tau = 1$ و با 100 000 بار تکرار



شکل ۴: تغییرات واریانس مکان بر حسب زمان برای $p = 0.5$، $l = 1$، $\tau = 1$ و با 100 000 بار تکرار

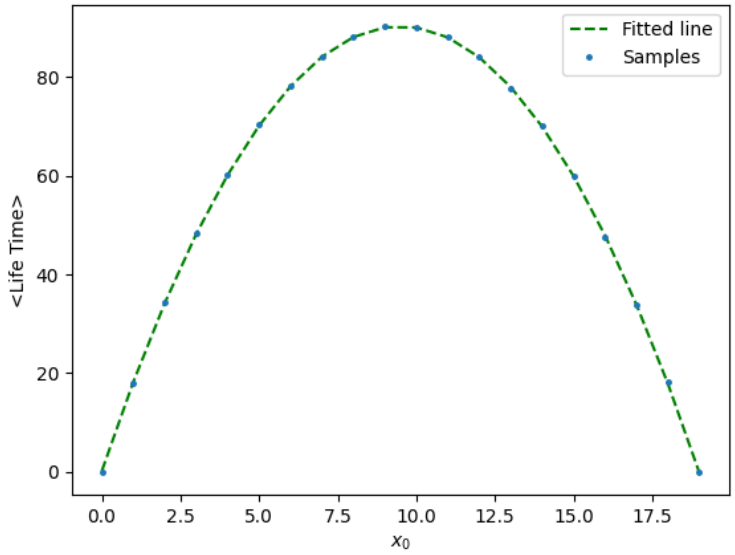


شکل ۵: تغییرات میانگین مکان بر حسب زمان برای $p = 0.8$، $l = 1$، $\tau = 1$ و با 100 000 بار تکرار



شکل ۶: تغییرات واریانس مکان بر حسب زمان برای $p = 0.8$، $l = 1$، $\tau = 1$ و با 100 000 بار تکرار

استفاده‌های بعدی، در یک فایل با فرمت npy. می‌نویسد. منحنی به دست آمده برای میانگین طول عمر ول‌گشت بر



شکل ۷: میانگین طول عمر ول‌گشت بر حسب مکان اولیه برای شبکه‌ای به طول 19 و با 100 000 بار تکرار

حسب مکان اولیه را در شکل ۸ می‌توان مشاهده نمود. سهمی فیت شده به این نمودار از رابطه‌ی (۴) پیروی می‌کند. اجرای این کد 0.55 ثانیه زمان برد.

$$\langle \text{life time} \rangle = x_0(19 - x_0) \tag{۴}$$

۵ ول‌گشت دو بعدی

کد این بخش از تمرین در فایل q5.py قابل مشاهده است. کد این بخش کاملا مشابه با کد تمرین ۲ است، تنها با این تفاوت که مکان‌ها به صورت یک عدد مختلط ذخیره می‌شوند و در هر مرحله ول‌گشت به چهار جهت می‌تواند حرکت کند. نمودارهای به دست آمده برای طول گام‌های 0.5، 1 و 2 را در شکل‌های ۹ تا ۱۱ می‌توان مشاهده نمود. شیب خطوط فیت شده به این نقاط نیز در جدول ۲ آورده شده است.

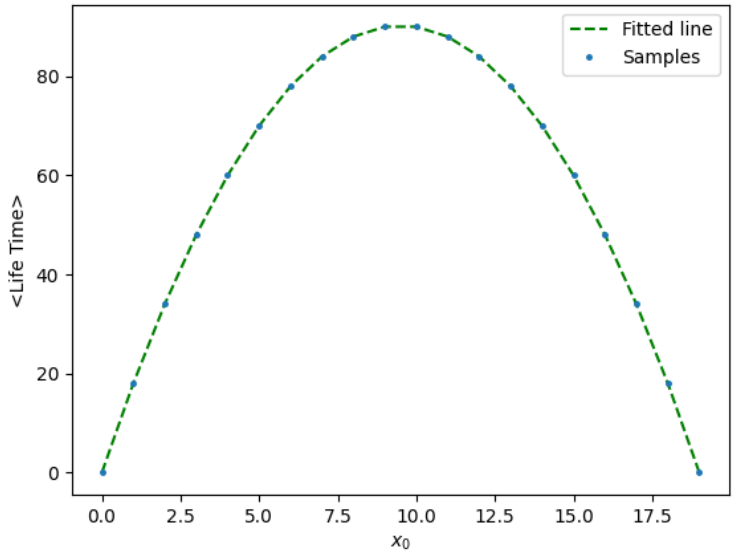
l	0.5	1	2
$\frac{l^2}{\tau}$	0.250	1.000	3.997

جدول ۲: شیب خطوط فیت شده به نمودار واریانس مکان ول‌گشت

همان‌طور که دیده می‌شود نتایج به دست آمده، مطابق با پیش‌بینی ما از تئوری هستند.

۶ Diffusion limited Aggregation

کد این بخش از تمرین در فایل q6.py قابل مشاهده است. ابتدا باید یک object از کلاس DiffusionLimited Aggregation با طول دل‌خواه بسازیم. سپس تابع render را با زمان دل‌خواه صدا می‌زنیم. شیوه کار این تابع به این صورت است که به طور تصادفی مکان اولیه‌ای برای ول‌گشت در حداقل ارتفاع ممکن نسبت به خوشه پیدا می‌کند. سپس مختصات به دست آمده را به تابع __final_position پاس می‌دهد. این تابع یک ول‌گشت را از آن نقطه شروع



شکل ۸: میانگین طول عمر ول‌گشت بر حسب مکان اولیه برای شبکه‌ای به طول 19 با الگوریتم تعینی

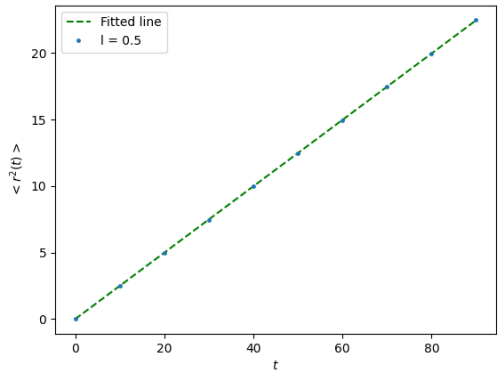
به حرکت می دهد تا نهایت یا از حد بالایی خارج شود و یا به خوشه بچسبد. اگر از حد بالا خارج شده باشد مقدار None و اگر چسبیده باشد مختصات نهایی را بر می‌گرداند. تابع render این حلقه را به قدری تکرار می‌کند تا به اندازه‌ی زمان داده شده، دانه به بذر اولیه چسبیده باشد. برای نمایش خوشه‌ی به دست آمده، تابع show را باید صدا بزنیم. نتیجه‌ی نهایی را در شکل ۱۲ می‌توان مشاهده نمود.

۷ ول‌گشت خودپرهیز

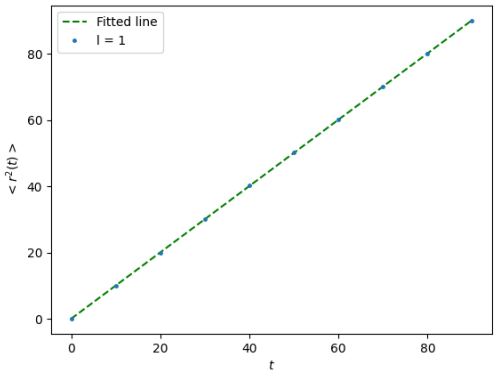
کد این بخش از تمرین در فایل q7.py قابل مشاهده است. برای رسم نمودارهای خواسته شده باید تابع show را با طول دلخواه صدا بزنیم. روش کار این تابع به این صورت است که ابتدا تابع count_self_avoiding_paths را با طول داده شده صدا می‌زند. این تابع در هر مرحله تعداد مسیرهای بسته را می‌شمارد. از آن‌جایی که هر مسیر بسته برای طول l، $4^{(l'-l)}$ مسیر بسته برای طول l' خواهد بود، پس این تعداد به دست آمده را طبق قاعده‌ی بالا به تمام طول‌های بزرگ‌تر مساوی l اضافه می‌کنیم. در نهایت داده‌های به دست آمده را در یک متغیر ذخیره کرده و برای استفاده‌های بعدی در فایلی با فرمت npy. می‌نویسیم. نمودارهای خواسته شده را در شکل‌های ۱۳ و ۱۴ می‌توان مشاهده نمود. همان‌طور که مشاهده می‌شود، نسبت تعداد گشت‌های خودپرهیز به تعداد کل گشت‌ها با افزایش N به صفر میل می‌کند که مطابق با انتظار ما است. داده‌های به دست آمده، در جدول ۳ آورده شده است.

N	گشت‌های خودپرهیز	N	گشت‌های خودپرهیز	N	گشت‌های خودپرهیز
1	4	7	2172	13	881500
2	12	8	5916	14	2374444
3	36	9	16268	15	6416596
4	100	10	44100	16	17245332
5	284	11	120292	17	46466676
6	780	12	324932	18	124658732

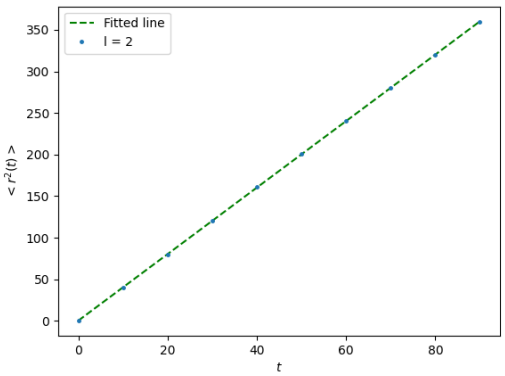
جدول ۳: تعداد گشت‌های خودپرهیز بر حسب N



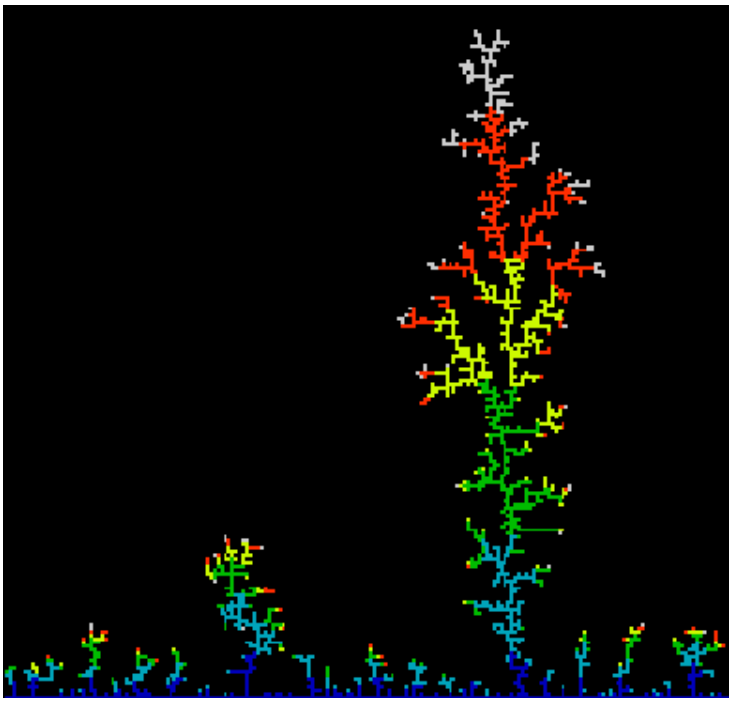
شکل ۹: تغییرات واریانس مکان بر حسب زمان برای $p = 0.25$، $d = 0.5$، $\tau = 1$ و با 100 000 بار تکرار



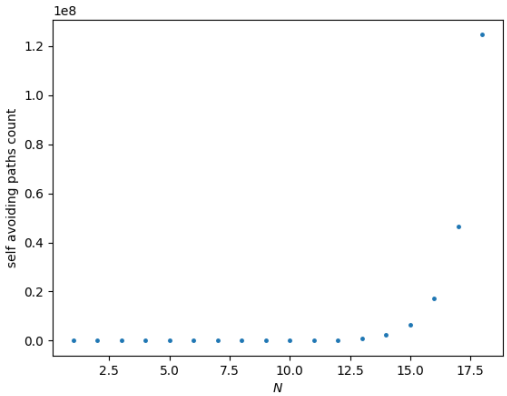
شکل ۱۰: تغییرات واریانس مکان بر حسب زمان برای $p = 0.25$، $d = 1$، $\tau = 1$ و با 100 000 بار تکرار



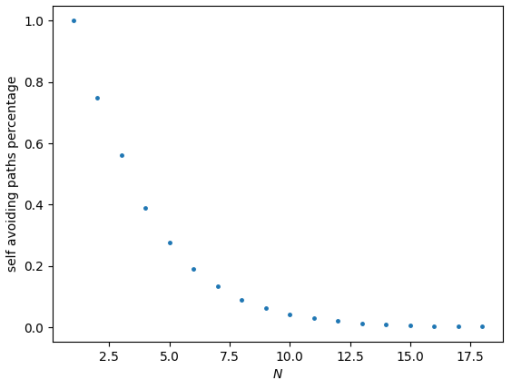
شکل ۱۱: تغییرات واریانس مکان بر حسب زمان برای $p = 0.25$، $d = 2$، $\tau = 1$ و با 100 000 بار تکرار



شکل ۱۲: خوشه‌ی تجمع پخش محدود برای بذر خطی با طول200 و 2 000 ذره



شکل ۱۳: تعداد گشت‌های خودپرهیز ممکن بر حسب N



شکل ۱۴: نسبت تعداد گشت‌های خودپرهیز بر تعداد گشت‌های آزاد بر حسب N