|  |  |  |
| --- | --- | --- |
| Optimized_image_813b5ec2.png | دانشگاه تهران  پردیس دانشکده های فنی  دانشکده مهندسی کامپیوتر | Images__Logo_FE.gif |
|  | | |
|  | | |
| دسته‌بندی بسته در شبکه‌های نرم‌افزار محور  پايان‌نامه جهت دريافت درجه کارشناسی ارشد  در رشته مهندسی کامپیوتر گرايش معماری | | |
| نگارش:  **محمدرضا پیروزی** | | |
| استاد راهنما:  دکتر ناصر یزدانی | | |
| **بهمن ماه 1395** | | |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Optimized_image_813b5ec2.png | دانشگاه تهران  پردیس دانشکده های فنی  دانشکده مهندسی کامپیوتر | | Images__Logo_FE.gif | |
|  | | | | |
|  | | | | |
| دسته‌بندی بسته در شبکه‌های نرم‌افزار محور  پايان‌نامه جهت دريافت درجه کارشناسی ارشد  در رشته مهندسی کامپیوتر گرايش معماری | | | | |
| نگارش:  **محمدرضا پیروزی** | | | | |
| استاد راهنما:  دکتر ناصر یزدانی | | | | |
|  | | | | |
| **بهمن ماه 1395** | | | | |
| tu | |  | | ArmFanni_Orginal |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **دانشگاه تهران**  **پردیس دانشکده­های فنی**  **دانشکده مهندسی برق و کامپیوتر**  **گواهی دفاع از پایان­نامه کارشناسی ارشد**  هیأت داوران پایان­نامه کارشناسی ارشد آقاي / خانم ....................................................................به شماره دانشجويي...........................در رشته ...................................... گرايش .................................. را در تاريخ ............................. با عنوان ............................................................................................   |  |  |  |  | | --- | --- | --- | --- | |  | به عدد | به حروف | | | با نمره نهايي |  |  | | |  |  |  | | | و درجه |  | | ارزيابي کرد. | | | | | | | | | |
|  | |  | |  | |  | | |
| **رديف** | **مشخصات هيئت داوران** | | **نام و نام‌ خانوادگي** | | **مرتبه دانشگاهي** | | **دانشگاه يا مؤسسه** | **امضاء** |
| **1** | **استاد راهنما**  **استاد راهنماي دوم (حسب مورد)** | |  | |  | |  |  |
| **2** | **استاد مشاور** | |  | |  | |  |  |
| **3** | **استاد داور داخلي** | |  | |  | |  |  |
| **4** | **استاد مدعو** | |  | |  | |  |  |
| **5** | **نماينده كميته تحصيلات تكميلي دانشكده / گروه** | |  | |  | |  |  |
| نام و نام خانوادگي معاون تحصيلات تكميلي و  پژوهشي دانشكده / گروه:  تاريخ و امضاء: | | | | | نام و نام خانوادگي معاون آموزشي و تحصيلات تكميلي  پرديس دانشكده‌هاي فني:  تاريخ و امضاء: | | | |

|  |
| --- |
| **تعهدنامه اصالت اثر** |
| **باسمه تعالي** |
| **اينجانب محمدرضا پیروزی تائيد مي‌كنم كه مطالب مندرج در اين پایان‌نامه حاصل كار پژوهشي اينجانب است و به دستاوردهاي پژوهشي ديگران كه در اين نوشته از آنها استفاده شده است مطابق مقررات ارجاع گرديده است. اين پایان‌نامه قبلاٌ براي احراز هيچ مدرك هم سطح يا بالاتر ارائه نشده است.**  **كليه حقوق مادي و معنوي اين اثر متعلق به دانشكده فني دانشگاه تهران مي باشد.**  **نام و نام خانوادگي دانشجو :**  **امضاي دانشجو :** |

**تقديم به پدر و مادرم**

**چکيده**

در شبکه‌های نرم‌افزار محور، یک جریان بر اساس تعداد دلخواهی از فیلدها در هر لایه از سرآیند بسته، قابل تعریف است. به عنوان مثال در نسخه 1.3.1 پروتکل OpenFlow از 15 فیلد برای تعریف یک جریان استفاده می‌شود. این امر سبب شده تا اولاً عرض مدخل در جدول جریان افزایش یابد، ثانیاً کنترل ریزتر بر روی ترافیک شبکه باعث افزایش تعداد مدخل‌ها در جدول جریان شده است. در نتیجه طراحی یک راه‌حل دسته‌بندی بسته در محیط‌هایی که به کارآیی بالا نیاز دارند، به یک چالش بزرگ تبدیل گشته است. برای حل این مسئله در سوئیچ‌های نرم‌افزاری سعی می‌شود تا حد امکان اندازه فضای نمونه برای مطابقت دادن یک چندتایی با قوانین، کوچک و کوچک‌تر شود. در الگوریتم MC-SBC یک ساختار مبتنی بر درخت تصمیم دو سطحی برای جدول‌های جستجو مطرح شده، و در آن بیت‌های موثر جهت افراز مجموعه قوانین و ایجاد درخت، با استفاده از یک روش آماری مشخص می‌گردند. در این پایان‌نامه از الگوریتم ژنتیک برای پیدا کردن مجموعه بیت‌های موثر جهت افراز مجموعه قوانین استفاده شده است. نتایج گزارش‌شده نشان می‌دهد مجموعه قوانین، با استفاده از این الگوریتم به صورت بهتری افراز می‌شوند، به طوری که هم نسبت تکرار قوانین در آن کاهش می‌یابد، و هم بیشینه تعداد قوانین در برگ‌های درخت کمتر می‌شود. بهبود کیفیت افراز مجموعه قوانین باعث می‌شود هنگامی که اندازه مجموعه قوانین بزرگ است، سرعت عمل دسته‌بندی بسته‌ها در نمونه پیاده‌سازی شده بر روی پردازنده گرافیکی تا حدود 10 درصد بهبود یابد.

**واژه‌هاي كليدي**: شبکه‌های نرم‌افزار محور، دسته‌بندی بسته، سوئیچ OpenFlow، پردازنده گرافیکی، الگوریتم ژنتیک

فهرست مطالب

[فصل 1: مقدمه 2](#_Toc473543796)

[1-1- تعریف مسئله 3](#_Toc473543797)

[1-2- روش انجام پژوهش 4](#_Toc473543798)

[1-3- ساختار پایان‌نامه 7](#_Toc473543799)

[فصل 2: زمینه‌های تحقیق 8](#_Toc473543800)

[2-1- شبکه‌های نرم‌افزار محور 8](#_Toc473543801)

[2-1-1- محدودیت‌های شبکه‌های سنتی 9](#_Toc473543802)

[2-1-2- تعریف شبکه نرم‌افزار محور 10](#_Toc473543803)

[2-1-3- معماری شبکه‌های نرم‌افزار محور 11](#_Toc473543804)

[2-1-4- چالش‌های شبکه‌های نرم‌افزار محور 13](#_Toc473543805)

[2-2- واسط OpenFlow 14](#_Toc473543806)

[2-2-1- اجزاء سوئیچ OpenFlow 15](#_Toc473543807)

[2-3- مسئله دسته‌بندی بسته‌ها 17](#_Toc473543808)

[2-3-2- تعریف رسمی مسئله دسته‌بندی بسته‌ها 19](#_Toc473543809)

[2-3-3- انواع عمل تطابق 19](#_Toc473543810)

[2-3-4- SDN و چالش دسته‌بندی بسته‌ها 20](#_Toc473543811)

[2-4- استفاده از پردازنده‌های گرافیکی برای دسته‌بندی بسته 22](#_Toc473543812)

[2-4-1- انجام محاسبات همه‌منظوره روی پردازنده‌های گرافیکی 22](#_Toc473543813)

[2-4-2- معماری پردازنده گرافیکی 23](#_Toc473543814)

[2-4-3- برنامه‌های کاربردی CUDA 26](#_Toc473543815)

[2-5- الگوریتم ژنتیک 27](#_Toc473543816)

[2-6- تولید مجموعه داده 28](#_Toc473543817)

[2-7- جمع‌بندی 31](#_Toc473543818)

[فصل3: مروری بر کارهای قبلی 33](#_Toc473543819)

[3-2- معیارهای سنجش الگوریتم‌های دسته‌بندی 35](#_Toc473543820)

[3-3- راه‌حل‌های استفاده کننده از ویژگی‌های SDN 36](#_Toc473543821)

[3-3-1- برچسب گذاری دو لایه‌ای 36](#_Toc473543822)

[3-3-2- برچسب گذاری در سوئیچ لبه 38](#_Toc473543823)

[3-3-3- پیکربندی پویای الگوریتم‎های دسته‌بندی 39](#_Toc473543824)

[3-4- حل مسئله به صورت الگوریتمی 41](#_Toc473543825)

[3-4-1- الگوریتم‌های ارائه شده بدون تاکید بر تعداد فیلد زیاد 43](#_Toc473543826)

[3-4-2- مدار پیش‌بینی کننده جریان 47](#_Toc473543827)

[3-4-3- تجزیه فیلد‌ها و ادغام نتایج جزئی 48](#_Toc473543828)

[3-4-4- GSwitch 52](#_Toc473543829)

[3-4-5- MC-SBC 57](#_Toc473543830)

[3-5- جمع‌بندی 60](#_Toc473543831)

[فصل4: استفاده از الگوریتم ژنتیک برای پیدا کردن بیت‌های موثر 61](#_Toc473543832)

[4-1- بررسی MC-SBC‌ با جزئیات بیشتر 61](#_Toc473543833)

[4-1-2- برنامه‌ریزی قانون به صورت برون‌خط 63](#_Toc473543834)

[4-1-3- فرآیند دسته‌بندی بسته به صورت برخط 64](#_Toc473543835)

[4-1-4- معیارهای انتخاب بیت‌های موثر 66](#_Toc473543836)

[4-1-5- نتایج MC-SBC 68](#_Toc473543837)

[4-2- استفاده از الگوریتم ژنتیک برای یافتن بیت‌های موثر 69](#_Toc473543838)

[4-2-2- مقداردهی اولیه 69](#_Toc473543839)

[4-2-3- انتخاب 70](#_Toc473543840)

[4-2-4- تقاطع و جهش 72](#_Toc473543841)

[4-2-5- شرط خاتمه الگوریتم 74](#_Toc473543842)

[4-3- نتایج بخش برون‌خط 74](#_Toc473543843)

[4-3-1- تعیین تعداد تکرار الگوریتم ژنتیک 74](#_Toc473543844)

[4-3-2- مقایسه دو الگوریتم از لحاظ نسبت تکرار و Binth 75](#_Toc473543845)

[4-3-3- زمان اجرای الگوریتم‌ها 81](#_Toc473543846)

[4-4- نتایج بخش بر‌خط 84](#_Toc473543847)

[4-4-2- بررسی تاثیر پارامترهای GPU در کارآیی 86](#_Toc473543848)

[4-5- جمع‌بندی 88](#_Toc473543849)

[فصل5: نتیجه‌گیری و کارهای آینده 89](#_Toc473543850)

[5-1- روش پیشنهادی و ارزیابی 90](#_Toc473543851)

[5-2- پژوهش‌های آینده 90](#_Toc473543852)

[مراجع 92](#_Toc473543853)

[واژه‌نامه انگلیسی به فارسی 95](#_Toc473543854)

[واژه‌نامه فارسی به انگلیسی 100](#_Toc473543855)

فهرست اشکال

[شکل (2-1) معماری شبکه‌های نرم‌افزار محور 12](#_Toc473543856)

[شکل (2-2) اجزاء اصلی سوئیچ OpenFlow 15](#_Toc473543857)

[شکل (2-3) یک شبکه نمونه که در آن سرویس‌های مختلفی برای بقیه فراهم می‌کند. 18](#_Toc473543858)

[شکل (2-4) معماری پردازنده‌های گرافیکی امروزی 24](#_Toc473543859)

[شکل (2-5) معماری چندپردازنده جریانی 25](#_Toc473543860)

[شکل (2-6) مثالی از زمان‌بندی wrap 25](#_Toc473543861)

[شکل (2-7) نمودار گردش کار الگوریتم ژنتیک 28](#_Toc473543862)

[شکل (2-8) نمودار بلوکی مجموعه ابزار ClassBench 30](#_Toc473543863)

[شکل (3-1) طبقه بندی روش‌های دسته‌بندی بسته بر اساس سخت‌افزاری یا نرم‌افزاری بودن 34](#_Toc473543864)

[شکل (3-2) یک گراف نمونه و مسیرهای مختلف 37](#_Toc473543865)

[شکل (3-3) نگاشت جریان به مسیر 38](#_Toc473543866)

[شکل (3-4) معماری سیستم مطرح شده در [9] 39](#_Toc473543867)

[شکل (3-5) معماری سخت‌افزار مطرح شده در [10] 40](#_Toc473543868)

[شکل (3-6) خط لوله فرآیند جستجو 41](#_Toc473543869)

[شکل (3-7) مثالی از ساختار درختواره‌های سلسله مراتبی 44](#_Toc473543870)

[شکل (3-8) مثالی از ساختار درختواره‌های با شاخه‌های هرس‌شده 45](#_Toc473543871)

[شکل (3-9) مثالی از ساختار AQT 45](#_Toc473543872)

[شکل (3-10) ساختار درخت در روش HiCut 46](#_Toc473543873)

[شکل (3-11) مدار پیش‌بینی کننده جریان به ازای هر پورت 47](#_Toc473543874)

[شکل (3-12) ذخیره مجموعه‌های RID در میزبان و استفاده از اندیس در GPU 50](#_Toc473543875)

[شکل (3-13) انتقال داده ناهمگام بین CPU، و GPU 51](#_Toc473543876)

[شکل (3-14) عملیات درج قوانین در Tuple Space Search 53](#_Toc473543877)

[شکل (3-15) نحوه نگاشت بین کلاس‌ها و چندتایی‌ها به بلوک‌ها و ریسمان‌های GPU 55](#_Toc473543878)

[شکل (3-16) معماری دو مرحله‌ای ارائه شده برای دسته‌بندی بسته در MC-SBC 58](#_Toc473543879)

[شکل (3-17) مثالی از ساختار داده MC-SBC 59](#_Toc473543880)

[شکل (4-1) معماری دو مرحله‌ای ارائه شده برای دسته‌بندی بسته در MC-SBC 62](#_Toc473543881)

[شکل (4-2) مثالی از ساختار داده MC-SBC 63](#_Toc473543882)

[شکل (4-3) نتایج بخش برون‌خط MC-SBC 68](#_Toc473543883)

[شکل (4-4) یک نمونه کروموزوم شامل بیت‌های موثر 69](#_Toc473543884)

[شکل (4-5) انجام عمل تقاطع روی دو کروموزوم منتخب و 72](#_Toc473543885)

[شکل (4-6) شماره تکرار مشخص شدن بهترین کروموزوم برای تعداد بیت‌های موثر مختلف 75](#_Toc473543886)

[شکل (4-7) نمودار نسبت تکرار و Binth برای مجموعه قوانین IPC با نسبت Wildcard مختلف 76](#_Toc473543887)

[شکل (4-8) نمودار نسبت تکرار و Binth برای مجموعه قوانین ACL با نسبت Wildcard مختلف 76](#_Toc473543888)

[شکل (4-9) نمودار نسبت تکرار و Binth برای مجموعه قوانین FW با نسبت Wildcard مختلف 77](#_Toc473543889)

[شکل (4-10) نمودار نسبت تکرار و Binth برای مجموعه قوانین IPC با تغییر تعداد زیرمجموعه‌ها 77](#_Toc473543890)

[شکل (4-11)نمودار نسبت تکرار و Binth برای مجموعه قوانین ACL با تغییر تعداد زیرمجموعه‌ها 78](#_Toc473543891)

[شکل (4-12) نمودار نسبت تکرار و Binth برای مجموعه قوانین FW با تغییر تعداد زیرمجموعه‌ها 78](#_Toc473543892)

[شکل (4-13) نمودار کمینه، میانگین، و بیشینه تعداد قوانین در برگ‌ها بر اساس نسبت Wildcard برای ACL\_100k 80](#_Toc473543893)

[شکل (4-14) نمودار کمینه، میانگین، و بیشینه تعداد قوانین در برگ‌ها بر اساس نسبت Wildcard برای FW\_100k 80](#_Toc473543894)

[شکل (4-15) نمودار کمینه، میانگین، و بیشینه تعداد قوانین در برگ‌ها بر اساس نسبت Wildcard برای IPC\_100k 81](#_Toc473543895)

[شکل (4-16) زمان اجرای الگوریتم برون‌خط MC-SBC، و ژنتیک 84](#_Toc473543896)

[شکل (4-17) تاخیر دسته‌ای بخش بر‌خط 85](#_Toc473543897)

[شکل (4-18) نمودار تاخیر دسته‌ای بر اساس تعداد بلوک 86](#_Toc473543898)

[شکل (4-19) نمودار تاخیر بر اساس تعداد ریسمان‌ها در هر بلوک 87](#_Toc473543899)

[شکل (4-20) نمودار تاخیر بر اساس اندازه دسته 87](#_Toc473543900)

فهرست جداول

[جدول (2-1) تغییرات نسخه‌های اصلی OpenFlow 16](#_Toc473543901)

[جدول (2-2) سرویس‌های ارائه شده توسط برای مشتریان 18](#_Toc473543902)

[جدول (2-3) جریان‌های مربوط به بسته‌های ورودی از پورت X 19](#_Toc473543903)

[جدول (2-4) فیلدهای سرآیند بسته که در OpenFlow پشتیبانی می‌شوند. 21](#_Toc473543904)

[جدول (2-5) نسبت تعداد مقادیر یکتا به تعداد کل قوانین در مجموعه قوانین 31](#_Toc473543905)

[جدول (3-1) نتایج به دست آمده در [12] 52](#_Toc473543906)

[جدول (3-2) گذردهی بخش برخط در MC-SBC 59](#_Toc473543907)

[جدول (4-1) گذردهی بخش برخط در MC-SBC 69](#_Toc473543908)

[جدول (4-2) کمینه، میانگین، و بیشینه تعداد قوانین در برگ‌ها برای مجموعه قوانین ACL 79](#_Toc473543909)

[جدول (4-3) کمینه، میانگین، و بیشینه تعداد قوانین در برگ‌ها برای مجموعه قوانین FW 79](#_Toc473543910)

[جدول (4-4) کمینه، میانگین، و بیشینه تعداد قوانین در برگ‌ها برای مجموعه قوانین IPC 80](#_Toc473543911)

[جدول (4-5) مشخصات پردازنده گرافیکی استفاده شده 85](#_Toc473543912)

1. فصل 1: مقدمه

# فصل ۲: پس‌زمینة تحقیق

در این فصل مطالبی راجع به مباحث پایه‌ای مورد نیاز برای ورود به حوزه تحقیقات و روش‌های ارائه شده، مطرح شده است. ابتدا

مجموعه داده برای ارزیابی روش‌‌های ارائه شده را توضیح داده‌ایم.

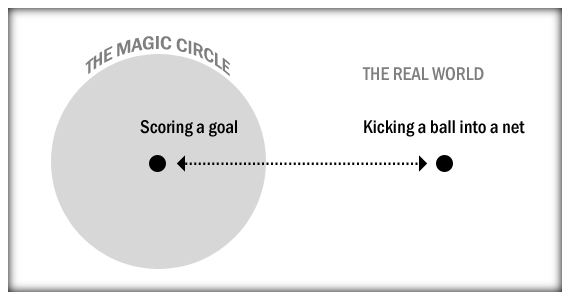
## بازی

آیا می‌توان به تعریف واحدی در مورد بازی (Game) رسید؟ وجه مشترک میان واترپولو، call of duty و بازی لِی‌لِی چیست؟ متفکران و فیلسوفان، به این سوال پاسخ‌های گوناگونی داده‌‌‌اند.

لودویک وینکنشتاین[[1]](#footnote-1) معتقد بود که نمی‌توان مرز مشخصی بین بازی و غیربازی دست یافت [1]. اما دیگران نظرات دیگری در این مورد داشتند. برای مثال برنارد سوییت[[2]](#footnote-2) تعریف زیر را برای بازی ارائه می‌کند:

«بازی کردن به معنای شرکت در فعالیتی است که در آن فقط باید از ابزارهایی که قانون مجاز می‌داند استفاده کرد، درحالی‌که استفاده از آن ابزارها در خارج از فضای آن قوانین محدودیت کمتری دارد؛ و تنها دلیلِ پذیرش این محدودیت‌ها این است که آنها فعالیت را ممکن می‌سازد.» [2]

برای مثال در خارج از دنیای قوانین فوتبال می‌توان به انواع مختلفی با توپ بازی کرد، اما وقتی بازیکنان قوانین فوتبال را می‌پذیرند، باید آن را در داخل یک محوطه مشخص نگه دارند و با دست به آن ضربه نزنند. و رعایت این قوانین است که بازیِ فوتبال را بامعنا و امکان‌پذیر می‌کند و با وضع مجموعه قوانینی دیگر، بازی‌های دیگری با توپ می‌توان انجام داد.



شکل ‏1‑1 مثالی از دایرة جادویی در بازی فوتبال

خطوط دور زمین فوتبال که محدودة بازی را مشخص می‌کنند را می‌توان به‌عنوان نمونه‌ای از مرزِ دایرة جادوییِ یوهان هایزینگا[[3]](#footnote-3) تلقی کرد. هایزینگا معتقد بود قوانین بازی، بازیکنان را در یک دایرة جادویی قرار می‌دهند که آنها را از قوانین دنیای واقعی جدا می‌کند و بین دنیای واقعی و دنیای بازی می‌توان یک مرزِ مفهومی در نظر گرفت و به‌محض اینکه از قوانین بازی تخطی شود، دنیای بازی فرومی‌پاشد و بازی به پایان می‌رسد. [3]

تذکر این نکته در اینجا خالی از لطف نیست که در فارسی واژة «بازی» به فعالیت‌های آزادانه و بدون قوانین مشخص کودکان نیز اطلاق می‌شود. چنین فعالیت‌هایی که صرفا با هدف سرگرم شدن، و نه برنده شدن، انجام می‌شوند در ادبیاتِ تحقیقاتِ بازی‌ در زبان انگلیسی «play» خوانده می‌شود و تعاریف جداگانه‌ای دارد.

## چرا بازی ها این­قدر جذاب هستند؟

پس از مشاهده یک مورد از تعاریفی که برای بازی ارائه شده، به‌دنبال این هستیم که دلیل جذابیت بازی‌ها را بررسی کنیم.

نیکول لازارو[[4]](#footnote-4) در [4]از تعدادی داوطلب هنگام انجام بازی‌های رایانه‌ای فیلم‌برداری کرد و با تحلیل حرکات آنها و سپس پاسخ­هایی به سوالات یک پرسش‌نامه داده بودند، نظریة «۴ کلید لذت[[5]](#footnote-5)» را معرفی کرد. او در این مقاله لذتی که به‌خاطر بازی‌کردن ایجاد می‌شود را در ۴ گروه دسته‌بندی کرده‌است که بازی‌های رایانه‌ای کمابیش سعی می‌کنند نمونه‌هایی از همة این لذایذ را پیاده‌سازی کنند.

* لذت دشوار[[6]](#footnote-6): لذتی که از غلبه بر سختی­ها و حل مشکلات و معماهای بازی حاصل می‌شود.
* لذت ساده[[7]](#footnote-7): احساس خوشایندی که از کنجکاوی و یافتن چیزهای جدید، گشت‌و‌گذار در محیط و توجه به جزییات و تخیل به دست می­آید. در واقع می­توان گفت لذتی که حاصل می‌شود در ساده بودن آن است. چون به حل معما و مشکلات متکی نیست.
* تغییر حالات درونی[[8]](#footnote-8): بهبودی که در احساسات و حالت درونی افراد بعد از بازی ایجاد می‌شود. مثلا احساس هیجانی در او ایجاد می‌شود یا مرحله‌ای که ذهنش را درگیر کرده بود تمام می‌شود و این دغدغه رفع می‌گردد.
* لذت اجتماعی[[9]](#footnote-9): بازیکنان از بازی به عنوان ابزاری برای تجربه­های اجتماعی و لذتِ کارِ گروهی و رقابت با دیگران استفاده می­کنند. این نوع fun را دو مورد اصلی ایجاد می­کنند. اول کار گروهی و همکاری در قالب یک تیم، و دوم رقابت با افراد دیگر و تلاش برای پیروزی.

1. فصل3: مروری بر کارهای قبلی
2. فصل4: استفاده از الگوریتم ژنتیک برای پیدا کردن بیت‌های موثر

1. Ludwig Wittgenstein فیلسوف اتریشی (۱۸۸۹-۱۹۵۱) [↑](#footnote-ref-1)
2. Bernard Suits فیلسوف کانادایی [↑](#footnote-ref-2)
3. Johan Huizinga فیلسوف و تاریخ‌شناس هلندی (۱۸۷۲-۱۹۴۵) [↑](#footnote-ref-3)
4. Nicole Lazzaro [↑](#footnote-ref-4)
5. 4 keys to fun [↑](#footnote-ref-5)
6. Hard fun [↑](#footnote-ref-6)
7. Easy fun [↑](#footnote-ref-7)
8. Altered state [↑](#footnote-ref-8)
9. Social fun [↑](#footnote-ref-9)