



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)  
دانشکده ریاضی و علوم کامپیوتر

پروژه  
علوم کامپیوتر

پیاده سازی بازی کاکورو

نگارش  
مهدی عباسعلی پور

استاد راهنما  
جناب آقای دکتر قطعی

استاد مشاور  
جناب آقای یوسفی مهر

آذرماه ۱۴۰۲

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## چکیده

در این پروژه هدف بر این است که بازی کاکورو با استفاده از زبان برنامه نویسی پایتون پیاده سازی شود. بازی به این صورت می باشد که اعداد ۱ تا ۹ باید با توجه به قوانینی که در راهنمای پروژه آمده است در خانه ای جدول نوشته شوند . برای حل این بازی دو عامل پیاده سازی می شوند که عامل اول بدون هیچ گونه ابتکاری به دنبال جواب می گردد اما عامل دوم با استفاده از حدس هایی ابتدا خانه هایی خاص را پر می نماید .

## واژه های کلیدی:

بازی کاکورو ، مسئله ارضای محدودیت، روش های فرا ابتکاری

## فهرست مطالب

صفحه

عنوان

۲	۱ بازی کاکورو
۳	۱-۱ مقدمه
۳	۲-۱ طرح مسئله
۴	۳-۱ لینک گیت هاب کد
۵	۲ پیاده سازی بازی و عامل ها
۶	۱-۲ محیط بازی
۶	۲-۲ مروری بر توابع اصلی پیاده سازی بازی
۶	۳-۲ عامل های بازی
۶	۱-۳-۲ عامل اولیه
۷	۲-۳-۲ عامل هوشمند
۸	۳ جمع بندی و نتیجه گیری
۱۰	مراجع

صفحه	شکل	فهرست تصاویر
۳	۱-۱ بازی kakuro	.....



# فصل اول

## بازی کاکورو

## ۱-۱ مقدمه

کاکورو یک بازی معمایی می باشد. می توان اینگونه گفت که کاکورو شباهت زیادی با سودوکو دارد. همانطور که از اسم این معما بر می آید، کاکورو خاستگاهی ژاپنی دارد [۱]. جدول کاکورو از چند سطر و ستون که هر سطر یا ستون ممکن است چند بخش باشند تشکیل شده و می تواند ابعاد مختلفی داشته باشد. در سمت چپ یا بالای هر بخش عددی قید شده که نشانگر مجموع اعداد آن قسمت از سطر یا ستون است. سطرها و ستون ها توسط خانه های خاکستری رنگ از یکدیگر جدا شده اند. برای حل کاکورو باید از بین اعداد ۱ تا ۹، اعداد مناسب را برای هر سطر یا ستون طوری انتخاب نمایید که اولاً مجموع اعداد آن سطر یا ستون با عدد نشان داده شده در ابتدای آن یکسان باشد و ثانیاً در هیچ سطر یا ستونی عددی تکراری نباشد.

	23	30			27	12	16
16				24			
17			29	17			
			15				
35						12	
	7			8			7
		16		7			
	11	10					
21					5		
6					3		

شکل ۱-۱: بازی kakuro  
[۲]

## ۲-۱ طرح مسئله

در اسن مسئله بایستی با استفاده از الگوریتم پس گرد عاملی طراحی کنیم تا جدول را حل نماید و سپس این عامل را بهبود دهیم. در نهایت عامل ها را با یکدیگر مقایسه نماییم.



## ۳-۱ لینک گیت هاب کد

با مراجعه به [https://github.com/mahdialipoo/AI\\_project4](https://github.com/mahdialipoo/AI_project4) می توانید کد مربوط به پیاده سازی این بازی را مشاهده نمایید .

## فصل دوم

### پیاده سازی بازی و عامل ها

## ۱-۲ محیط بازی

صفحه بازی به صورت آرایه ای جدولی از اعداد می باشد که خانه هایی که با صفر مشخص می شوند مجاز به دریافت مقدار می باشند و باقی خانه ها با ۱- پر می شوند . برای مشخص کردن هر بخش و حاصل جمع آن بخش از لیستی از قیود استفاده می شود . که مختصات خانه ی شروع و طول و حاصل جمع و عمودی بودن یا افقی بودن بخش در لیست نگهداری می شود .

## ۲-۲ مروری بر توابع اصلی پیاده سازی بازی

اگر دو تابع `tab-solved` و `constrain-hope` که جدول و قیود را به عنوان ورودی می پذیرند `True` شود جدول حل شده است .

توابع `search-constrain-v` و `search-constrain-h` مختصات یک خانه را می گیرند و به ترتیب قید های متناظر سطری و ستونی آن را باز می گردانند . اگر یکی از این قیود را نداشته باشد نیز مقدار `True` را باز می گردانند .

تابع `search` جدول و قید ها به همراه یک تابع `selection` را می گیرد و جدول را با جست و جو در فضای حالات به روش `DFS` حل می نماید . تابع `selection` در هر مرحله مشخص می نماید که کدام خانه مقدار دهی شود . با تغییر این تابع می توان عاملی را که کل فضای حالات را می گردد تبدیل به عاملی کرد که با روشی ابتکاری به جست و جو می پردازد و جست و جو سمت و سو می دهد .

## ۳-۲ عامل های بازی

در بخش پیشین این نکته ذکر شد که با تغییر تابع `selection` که به عنوان ورودی تابع `search` است می توان عامل را هوشمند کرد .

### ۱-۳-۲ عامل اولیه

در ابتدا انتخاب بر مبنای این بود که خانه ها به ترتیب از مختصات مبدا مقدار دهی شوند . به ترتیب مقادیر ۱ تا ۹ را بپذیرند و امید این که به جواب می رسد در هر مرحله بررسی شود که زمان حل بسیار طولانی می شد .

پس از این آزمون اولیه انتخاب خانه ها به صورت تصادفی انجام می شد که زمان حل بهبود یافت .

## ۲-۳-۲ عامل هوشمند

با استفاده از استراتژی انتخاب محدود ترین متغیر به مقدار دهی متغیر ها پرداخته شد. این نوع انتخاب در تابع select-cell-intelligent پیاده سازی شده است .

## فصل سوم

### جمع بندی و نتیجه گیری

با استفاده از الگوریتم back tracking سعی شد تا بازی کاکورو حل شود و با توجه به طولانی بودن زمان حل به این روش در ادامه سعی شد به روش های فرا ابتکاری مسلح شود. از مهم ترین چالش ها پس از پیاده سازی روش انتخاب بهترین خانه بود که سعی شد با معیار های متفاوتی انجام پذیرد و در نهایت با استفاده از LCV<sup>۱</sup> سعی شد این بهبود حاصل شود. اما به علت زیاد بودن محاسبات حل جدول توسط هیچ کدام از عامل ها مقدور نشد. اما در مقادیر دامنه ی اعداد کمتر از ۵ (به جای اعداد یک تا نه) حل جدول در حدود دودقیقه توسط عامل اول و در حدود یک دقیقه توسط عامل دوم انجام پذیرفت.

---

<sup>۱</sup>Least Constraining Value

## مراجع

[1] jadvalyab. how to play kakuro. , 2023.

[2] wikioedia. Kakuro. , 2023.