

## Absztrakt

Az "Abstraction and Reasoning Corpus" (ARC) a gépi tanuló algoritmusok generalizációjának mérésére szolgáló módszer. A meglévő programindukációs megoldások hátránya, hogy az alkalmazott terület specifikus nyelv (DSL) által definiált programfa, amelyben a keresés történik, a program hosszával exponenciálisan növekszik, így a keresési terület túl nagy a megfelelő program megtalálásához. Programindukció alapú megoldásunk arra törekedett, hogy kiküszöbölje a DSL programfájában való keresést. Ez megköveteli, hogy az indukált program ne legyen procedurális, azaz ne álljon egymásra épülő utasítások sorozatából. Az ilyen programok indukálása sokkal egyszerűbb, mivel az utasítások kölcsönös függésének lebontása miatt a keresési idő csak lineárisan nő a program hosszával. Ez a módszer sikeresen csökkentette a probléma keresési területét. A korábbi módszerekhez képest az átlagos feladatmegoldási idő lényegesen alacsonyabb volt, mint bármely más publikációban. Az oktatókészletben 41, a nyilvános teszt sorozatban 10 feladatot tudott megoldani, ami a korábbi publikációk között a negyedik legjobb eredmény.

## Kulcsszavak

absztrakció, következtetés, program indukció

## Motiváció

Napjainkban a mély tanulás számos területen egyre elterjedtebbé válik, azonban megvannak a maga limitációi: a módszer csak akkor képes hatékonyan működni, ha a tanítás folyamán a problémater sűrű mintavételezésen esik át, valamint a teszt minták kellő képpen hasonlóak a tanítási mintákhoz. Tehát bármely variáció hatására a hálózat teljesítménye drasztikusan lecsökken, ha az algoritmus a tanulás során erről a variációról nem kapott elegendő tanító mintát. Ennek a problémára keresi a megoldást az "Abstraction and Reasoning Corpus (ARC)", amely az extrém generalizáció mérésére szolgál, amit megoldva közelebb juthatunk egy általánosabb tanuló algoritmus megalkotása felé.

## Hasonló fejlesztések

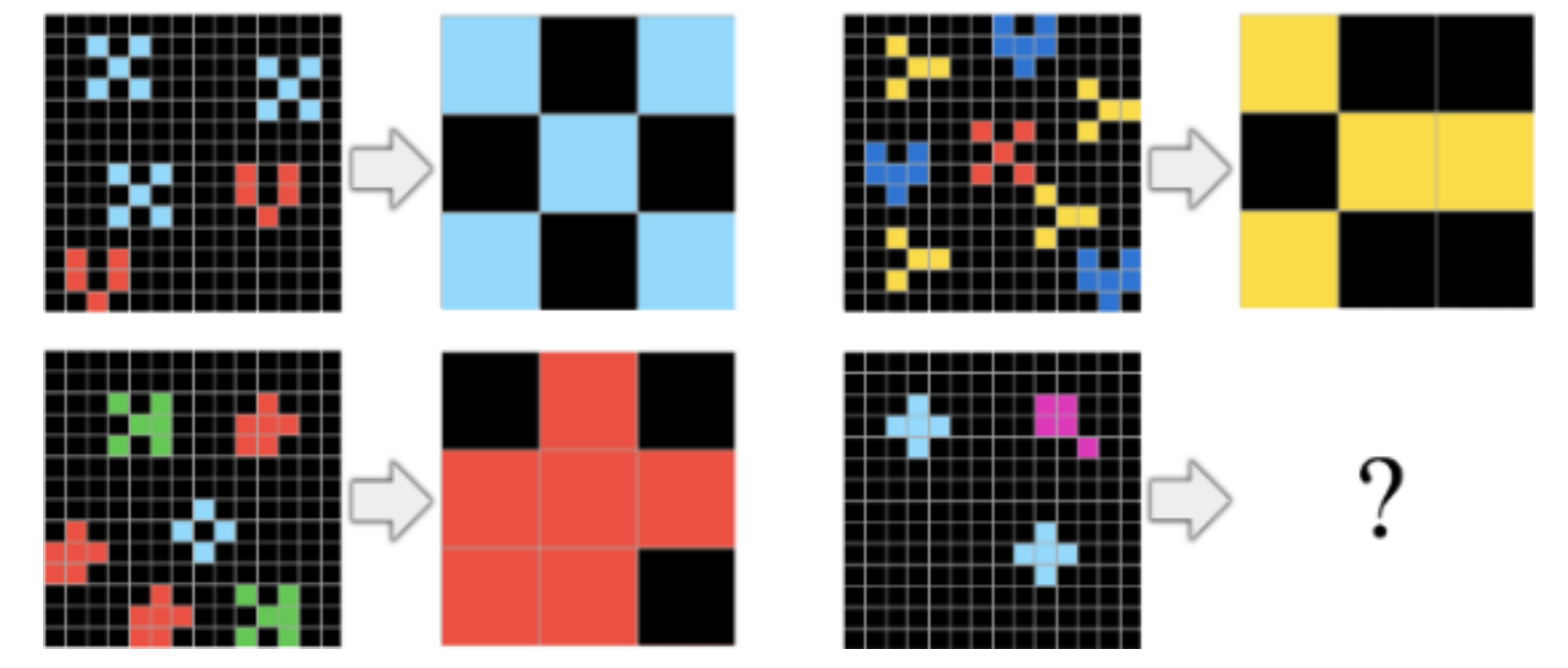
- ▶ A hasonló megoldások egyik csoportja a rendelkezésre álló kevés számú mintát különböző absztrakt relációk felfedezésével próbálja megoldani, amihez mély tanulást, program indukciót, illetve látens vektortereket használnak.
- ▶ A másik csoportba program indukción alapuló eljárások tartoznak, melyeket evolúciós algoritmussal, brute force-al vagy két irányú kereséssel próbálja megoldani a problémát.
- ▶ Az egyes megoldások tekintetében kijelenthető, hogy még nem született olyan megoldás, amely meg tudná közelíteni az emberi teljesítményt, ami a mérések alapján átlagosan 80%.

## Sikerek

- ▶ Kari TDK 2. helyezett
- ▶ 1 konferenciatick

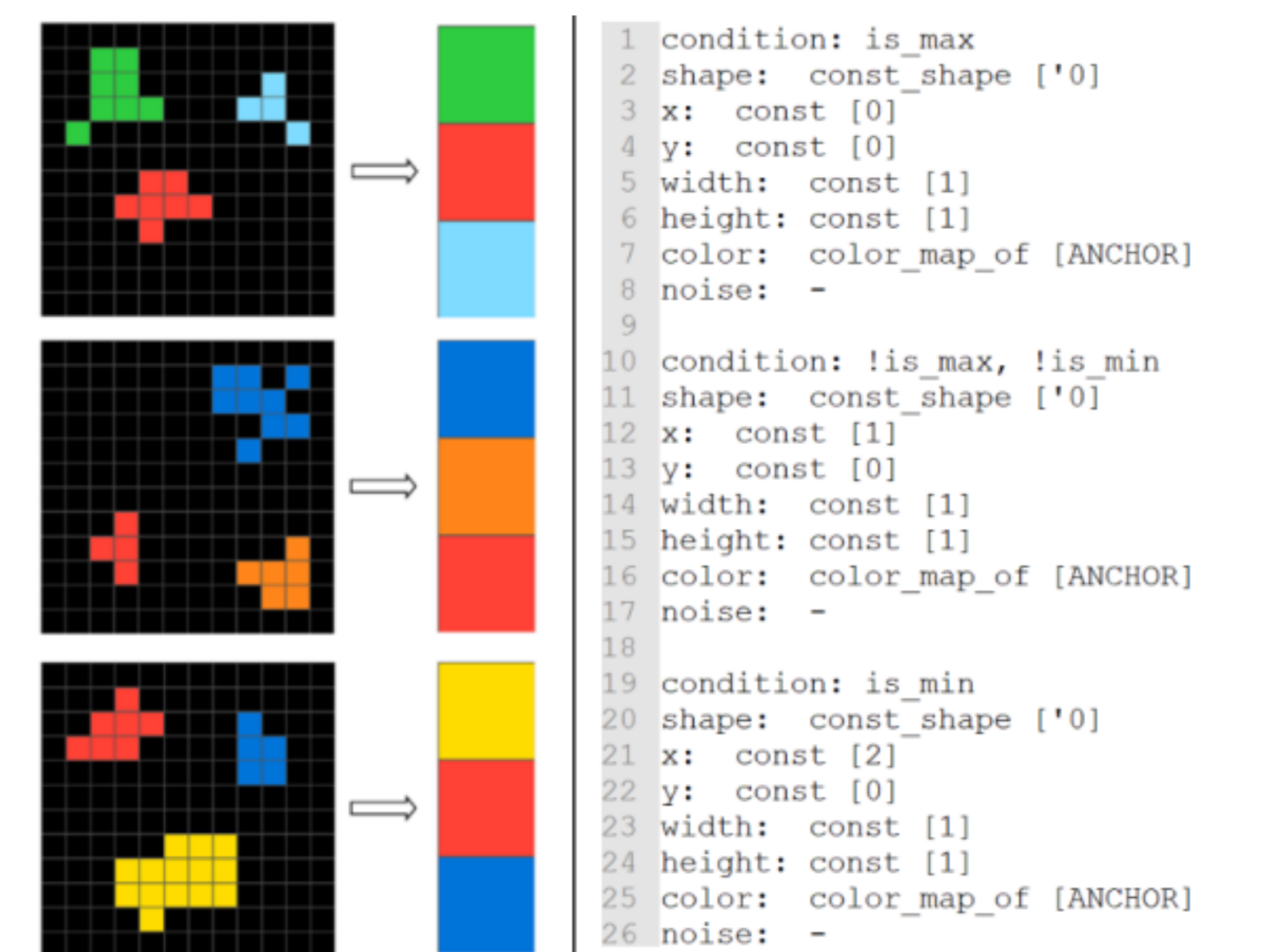
## Saját módszer bemutatása

Az ARC 1000 (400 tanítási, 600 tesztelési) képalapú érvelési feladatot tartalmaz. Egy feladatban meghatározott számú (általában három) demonstrációs példát adnak meg. Egy példa egy bemenetből és a hozzá tartozó kimenetből áll, ahol a kimenetet a bemenet komplex transzformációjából kapjuk. Mind a bemenet, mind a kimenet egy 1×1-től 30×30-ig terjedő négyzetrácsból áll, ahol minden pozíció egy-egy színként megjeleníthető szimbólumot tartalmaz, mint például a jobb oldali ábrán. Összesen tíz szín lehetséges. A feladatot megoldó algoritmusnak fel kell ismernie a demonstrációs példákon végrehajtott transzformációt, majd alkalmaznia kell a tesztpéldákra. Az algoritmus csak a válasz sikerességéről kap visszajelzést, az elkövetett konkrét hibákról nem.



A feladat megoldására tett kísérlet programindukcióval történt, melynek első lépéseként objektumcentrikus módon specifikálva lett a feladatokat értelmezni képes nyelv, ami leírja a képen látható objektumok tulajdonságait (alakzat; x, y koordináta; szélesség, magasság, szín, zaj). Ezután a nyelv műveletei lettek meghatározva művelet\_név [arg0] [arg1] formátumban. Például az x\_of [obj] visszatér a paraméterül kapott objektum X koordinátájával. Ezek alapján elkészíthető minden kép programfája.

A nyelv leírását követően az indukciós algoritmus készült el, mely lehetővé teszi a programfába való keresés elhagyását, melynek feltétele, hogy az indukált program ne procedurális legyen, azaz a program nem utasítások egymásra épülő sorozatából álljon. Az ilyen programok indukciója jóval egyszerűbb, mivel az egymásra épülés felbontásával a keresési idő a program hosszával csupán lineárisan nő. Ez a módszer sikeresen lecsökkentette a probléma keresési terének méretét.

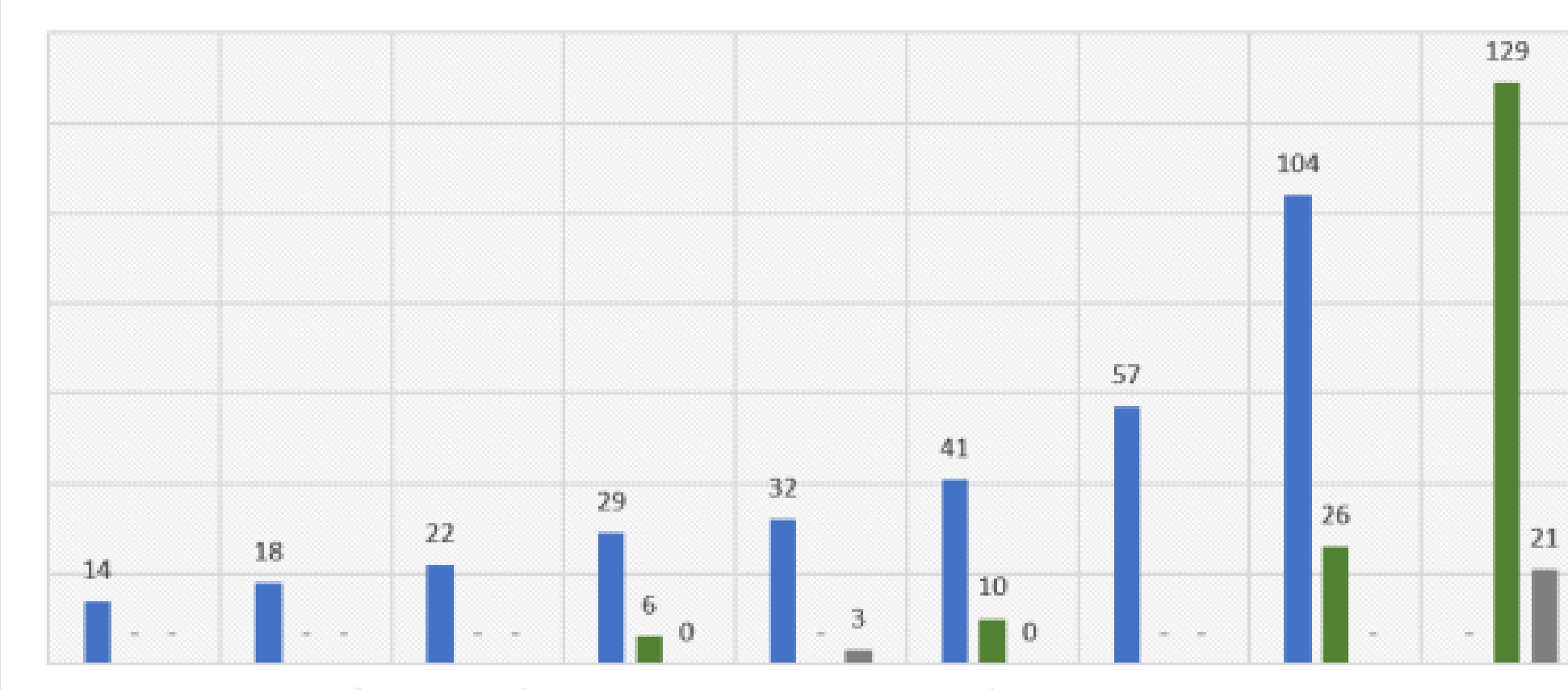


Jobb oldali képen látható az ARC "f8ff0b80" azonosítójú feladat, illetve mellette az erre indukált program.

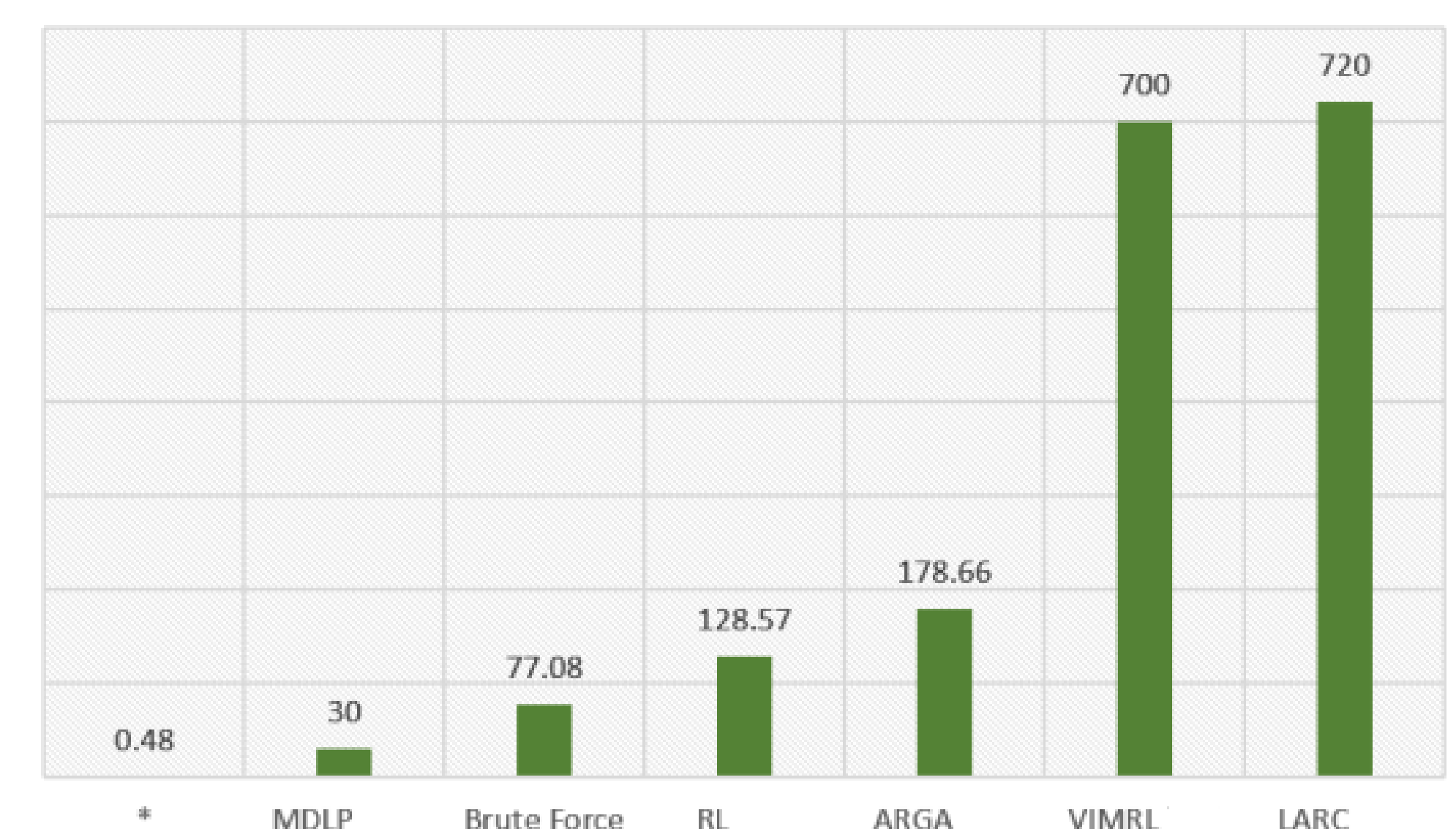
## Eredmények értékelése

Az elkészült rendszer eredményeit az alábbi táblázat tartalmazza. A 400 feladatból álló tanítási halmazból 41, a publikus teszt halmazból 10 feladatot oldott meg sikeresen. Más publikációkkal összehasonlítva ez a negyedik legjobb eredmény (alsó baloldali ábra). A végrehajtási időt tekintve az átlagos feladatmegoldási idő 0.48, illetve 1.67 másodperc a tanítási és teszt adatokon. Ezt a mérőszámot több más megoldásban is említették, de a nem egységes végrehajtási környezet miatt ezek összehasonlítása nem ad pontos képet az egyes algoritmusok tényleges sebességéről. Ennek ellenére megfigyelhető, hogy a bemutatott módszer átlagos végrehajtási ideje nagyságrendekkel kisebb, mint a többi (alsó jobb oldali ábra).

	Tanító halmaz	Publikus Teszt halmaz	Privát Teszt halmaz
<b>Megoldott feladatok száma</b>	41	10	0
<b>Átlagos futásidő (másodpercben)</b>	0.48	1.67	-



Az egyes megoldások teljesítménye a tanító (kék), publikus teszt (zöld), és privát teszt (szürke) adathalmazokon. Az itt tárgyalt megoldás eredményei a "\*" oszlop jelöli.



Az egyes megoldások átlagos feladatmegoldási ideje másodpercben. Az itt tárgyalt megoldás eredményei a "\*" oszlop jelöli.

Sok más algoritmushoz hasonlóan a bemutatott rendszer egyetlen feladatot sem tudott megoldani a privát teszt halmazon. Ennek pontos okát csak a feladatok ismeretében lehetne meghatározni, de a létező megoldások eredményeiből arra lehet következtetni, hogy a privát feladatok nem, vagy csak valamilyen bonyolult módon oldhatóak meg objektumcentrikus megközelítéssel. A fenti bal oldali ábrán látható, hogy az összes ilyen szemléletet alkalmazó algoritmus teljesítménye nullára esett. A két algoritmus, ami viszont ennél jobb eredményt ért el, az pont az a két módszer, amely teljes képes műveleteket használ, igaz ezek teljesítménye is körülbelül a tizedére csökkent a publikus adathalmazhoz képest.