

Report: 3SAT problém

Specifikace úlohy	2
Program	2
Popis algoritmů	3
SAT_V1	3
SAT_V2	4
SAT_V3	4
Hardwarová konfigurace	4
Naměřené výsledky	5
Porovnání algoritmů se základními parametry	5
Grafy vývoje hmotnosti a počtu splněných klauzulí	5
Porovnání nastavení parametrů	6
Koeficient ochlazování	6
Minimální teplota	7
Počet vnitřních cyklů	8
Počáteční teplota	9
Nastavení parametrů na základě měření	10
Závěr	11
Přiložené důležité soubory	11

Specifikace úlohy

Úkoly:

1. Řešení vážené splnitelnosti booleanovské formule (dále *problém*).
 - a. Použita pokročilá iterativní metoda Simulated Annealing (SA).
2. Zjištění, jak nastavit parametry pro SA pro daný problém
3. Experimentální vyhodnocení SA na poskytnutých datasetech s různým nastavením parametrů

Výpočetní složitost je měřena počtem navštívených konfigurací, tedy vyhodnocených booleanovských formulí. Je zachycen i reálný čas.

Plné zadání úlohy [zde](#).

Pro tento problém byli poskytnuty datasety s optimálními řešeními.

Program

Program byl napsán v jazyce Python. Algoritmy jsou též napsány v jazyce Python s optimalizací pomocí jazyka Cython.

Pro vyřešení *problému* lze využít tyto **algoritmy**:

Soubor	Popis
<code>package_algorithm/sat/sa_sat_v1.py</code>	Základní verze algoritmu SAT.
<code>package_algorithm/sat/sa_sat_v2.py</code>	Vylepšená verze algoritmu SAT_V2. Přidán reset teploty.
<code>package_algorithm/sat/sa_sat_v3.py</code>	Vylepšená verze algoritmu SAT_V3. Přidáno vylepšené vyhledávání sousedů.

Popis algoritmů

Všechny algoritmy přijímají tyto **parametry**:

Parametr	Typ	Hodnoty	Informace
Počáteční teplota	Celé číslo	<1, inf)	
Konečná teplota	Celé číslo	<1, inf)	Teplota, jejíž dosažení značí konec algoritmu.
Koeficient ochlazování	Desetinné číslo	(0, 1)	Určuje rychlost snižování teploty podle vztahu: $teplota_1 = teplota_0 * koeficient$
Počet interních cyklů	Celé číslo	(0, inf)	Kolikrát je hledáno a vyzkoušeno sousední řešení, než dojde ke snížení teploty.
Maximální počet resetů teploty	Celé číslo	(0, inf)	Kolikrát je možné resetovat teplotu, pokud za běhu algoritmu nebylo nalezeno správné řešení.

SAT_V1

Základní a nejjednodušší funkční verze algoritmu SAT s penalizační funkcí.

Nastavení počátečního řešení

Jako počáteční řešení je vždy zvolen nulový vektor, tedy všechny proměnné nastaveny na False. Způsob nastavení počátečního řešení není tak důležitý, protože na počátku má algoritmus vysokou šanci, že zvolí i horší řešení. Byl by důležitý, pokud by existoval jednoduchý a rychlý způsob, jak najít takové ohodnocení, aby formule byla splnitelná. Ten však pravděpodobně neexistuje.

Výběr sousedního řešení

Sousední řešení je získáno pomocí přepnutí 1 bitu řešení, tedy jedna z proměnných je negována.

Porovnání a výběr lepšího řešení

Oproti problému batohu je výběr lepšího řešení složitější, protože se řídí počtem splněných klauzulí a sumární hmotností.

Pro sousední (nové) řešení A a staré řešení B se algoritmus řídí touto tabulkou:

	B je splnitelné	B není splnitelné
A je splnitelné	Lepší je řešení s vyšší sumární hmotností	A je lepší
A není splnitelné	B je lepší	Lepší je řešení s vyšším počtem splněných klauzulí

Porovnání a výběr horšího řešení

Oproti problému batohu je též náhodný výběr horšího řešení složitější, protože se řídí počtem splněných klauzulí a sumární hmotností.

Výběr horšího řešení je dán vztahem $e^x > random(0..1)$; $x = \frac{newFitness - oldFitness}{currTemperature}$.

Pro sousední (nové) řešení A a staré řešení B se algoritmus řídí touto tabulkou:

	B je splnitelné	B není splnitelné
A je splnitelné	Za hodnoty newFitness a oldFitness je dosazena sumární hmotnost.	-
A není splnitelné	A není nikdy vybráno, nikdy nechci získat vyšší sumární hmotnost za cenu nesplnitelnosti formule.	Za hodnoty newFitness a oldFitness je dosazen počet splněných klauzulí.

Globálně nejlepší řešení

Kromě posledního nejlepšího řešení a jeho souseda si algoritmus udržuje globálně nejlepší řešení. Jde o řešení s nejvyšší fitness hodnotou, které bylo nalezeno během všech kroků algoritmu.

SAT_V2

Rozšíření algoritmu SAT_V1 o následující logiku.

Reset teploty

Pokud po dosažení minimální teploty nebylo nalezeno splnitelné řešení, algoritmus teplotu resetuje na počáteční teplotu a pokračuje ve výpočtu. Je možné nastavit maximální počet těchto resetů pomocí vstupního parametru.

SAT_V3

Rozšíření algoritmu SAT_V2 o následující logiku.

Vylepšený výběr sousedního řešení

Pokud poslední nejlepší řešení není splnitelné, sousední řešení není generováno pomocí přepnutí 1 náhodného bitu. Je vybrána jedna z proměnných, které figurují v největším počtu nesplněných klauzulí.

Součástí každého řešení je pole hodnot zvané *invalid_literals_per_var* délky rovné počtu proměnných. V tomto poli je pro každou proměnnou uložen počet nesplněných klauzulí, ve kterých proměnná figuruje.

Z tohoto pole je vybráno N proměnných s nejvyšším výskytem v nesplněných klauzulích, kde $N = \max(2, \frac{\text{numOfVars}}{5})$. Jedna z těchto proměnných je náhodně vybrána a přepnuta (negována). Vybírám z alespoň 2 proměnných, protože je tak menší šance, že se při výběru sousedního řešení zasekneme ve smyčce.

Hardwarová konfigurace

Operační systém	Lubuntu
Procesor	2,3 GHz Intel Core i5
RAM paměť	8 GB 2133 MHz LPDDR3
Programovací jazyk	Python 3.7

Naměřené výsledky

Všechny ukazované výsledky byli naměřeny na zadaných datasetech. Po několika rychlých testech jsem zvolil tyto základní parametry pro testování:

Parametr	Hodnoty
Počáteční teplota	1000
Konečná teplota	0,5
Koeficient ochlazování	0,95
Počet interních cyklů	50
Maximální počet resetů teploty	3

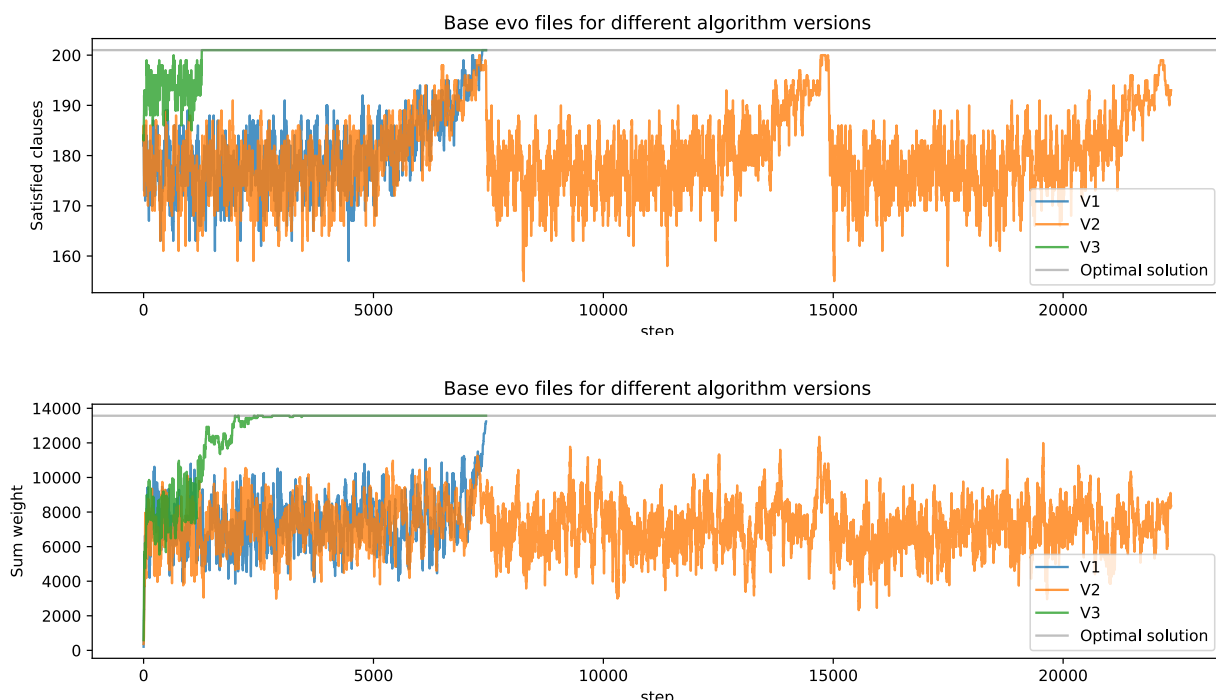
Porovnání algoritmů se základními parametry

Po vytvoření algoritmu SAT_V1 jsem provedl jeho analýzu na zadaných datasetech. Protože počet výsledných formulí, u kterých nebylo nalezeno splnitelné řešení, se mi zdál příliš vysoký (7%-15%), zaměřil jsem se na vylepšení algoritmu.

SAT_V2 přineslo zlepšení, počet těchto formulí pak nebyl vyšší než 5% a SAT_V3 téměř vždy dokázal najít validní řešení.

Grafy vývoje hmotnosti a počtu splněných klauzulí

Na následujících grafech vidíme, jak se algoritmy chovají v průběhu práce.



Vidíme, že algoritmus V2 během 3 resetů teploty nedosáhl splnitelného řešení. Algoritmy V1 a V2 splnitelného řešení dosáhly.

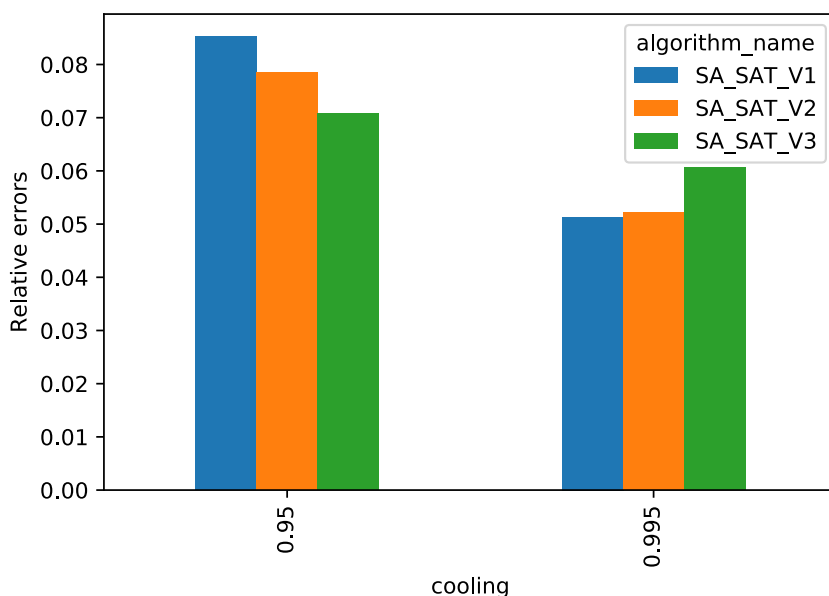
Algoritmy jsou randomizované, tento výsledek tedy není vždy stejný. Obecně platilo, že i při těchto vcelku nízko volených parametrech algoritmus V3 vždy dosáhl validního řešení nejvýše během 2 resetů. Ne vždy dosáhl optimálního řešení z hlediska hmotnosti.

Porovnání nastavení parametrů

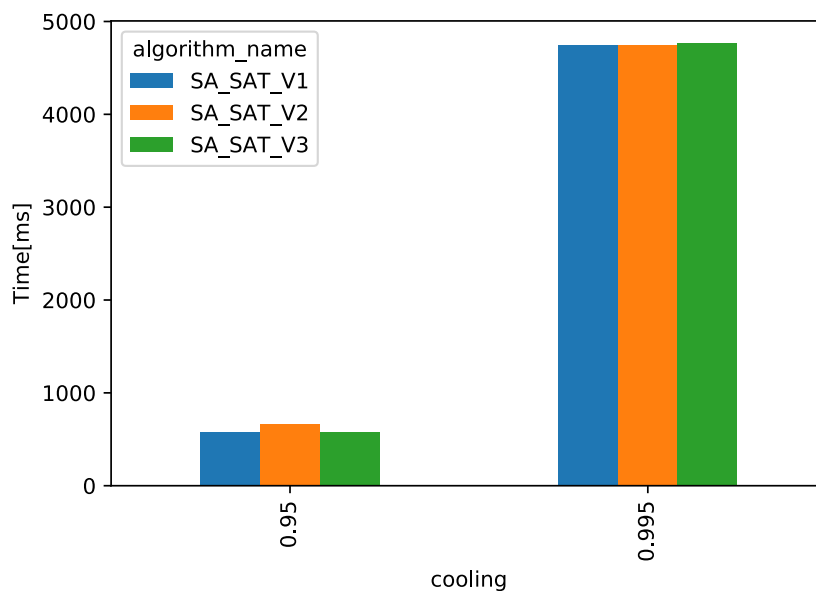
Koeficient ochlazování

Použití různých koeficientů ochlazování. Je vidět, že dojde ke zvýšení přesnosti, ale za cenu významného zvýšení výpočetního času. SAT_V3 také vždy našel validní řešení.

Avg error - cooling coefficients



Avg speed - cooling coefficients

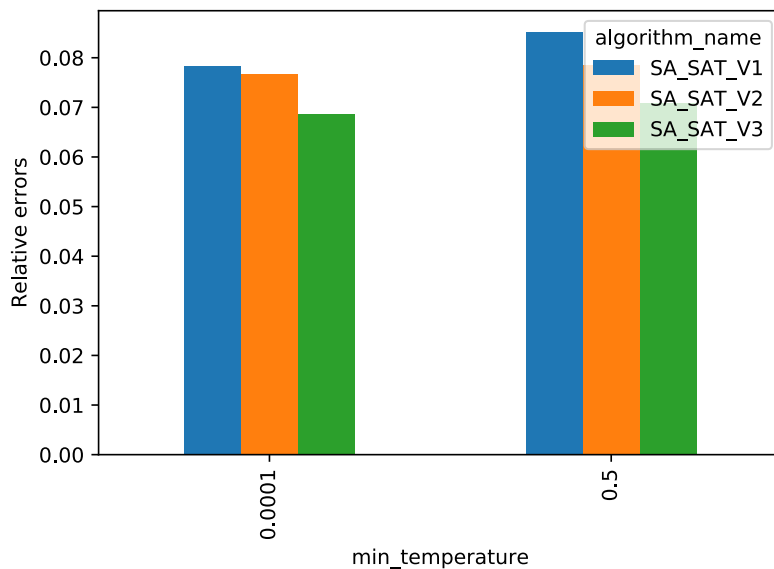


algorithm_name	% valid		
	SA_SAT_V1	SA_SAT_V2	SA_SAT_V3
cooling			
0,95	0,887	0,989	1
0,995	1	1	1

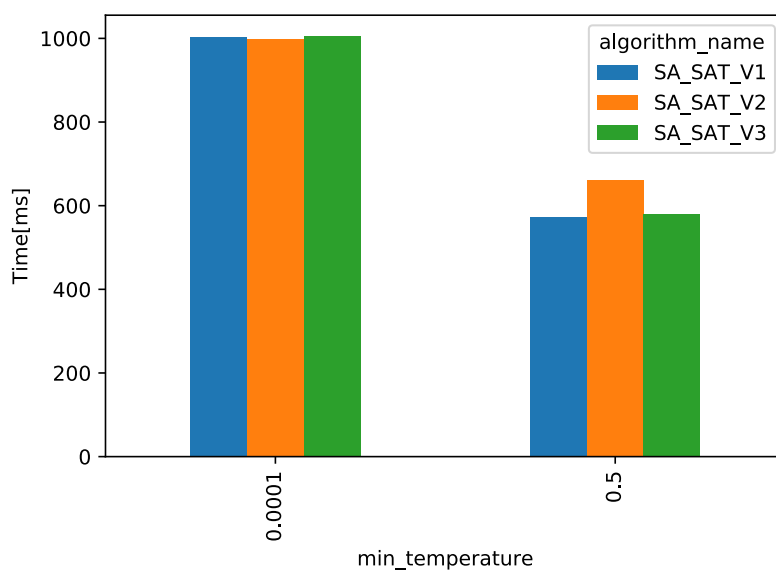
Minimální teplota

Použití různých minimálních teplot. Je vidět, že hodnota relativní chyby se příliš nemění, protože v danou chvíli se již algoritmus pravděpodobně nachází v lokálním extrému. SAT_V3 také vždy našel validní řešení.

Avg error - minimum temperature



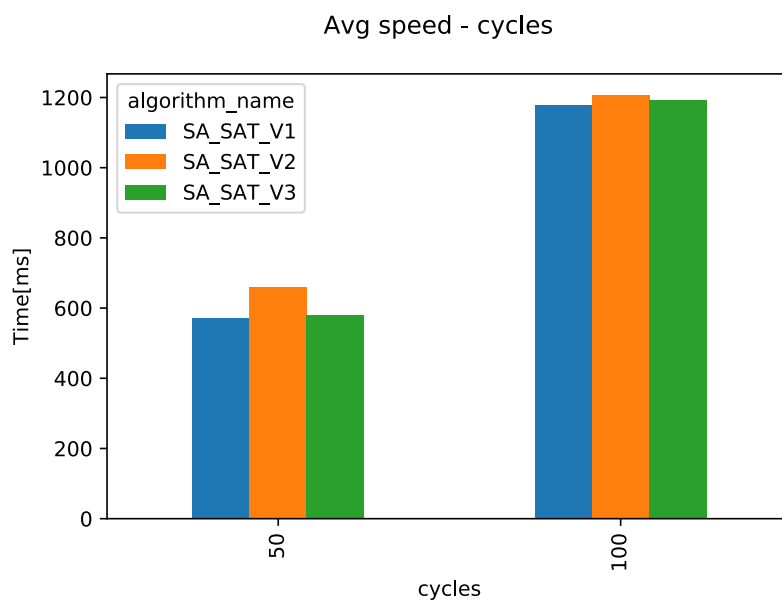
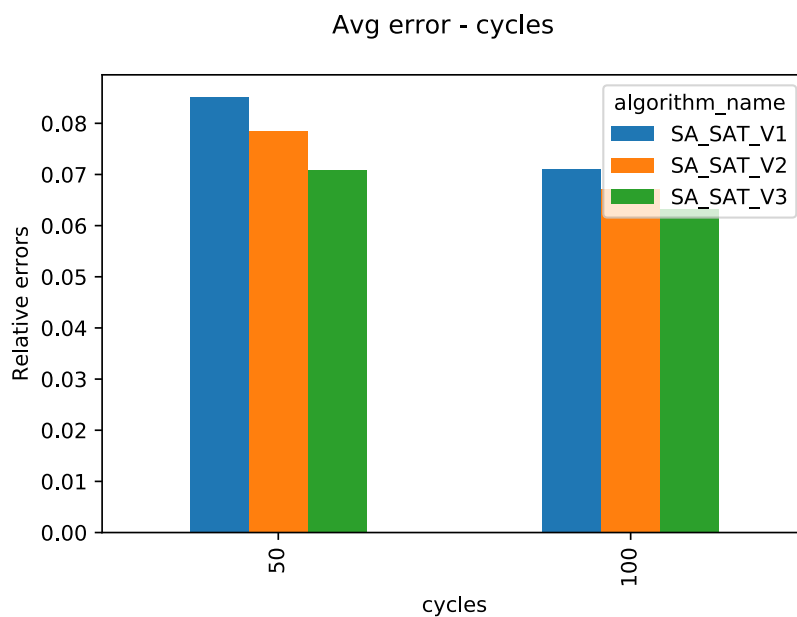
Avg speed - minimum temperature



algorithm_name	% valid		
	SA_SAT_V1	SA_SAT_V2	SA_SAT_V3
min_temperature			
0,0001	0,995	1	1
0,5	0,887	0,989	1

Počet vnitřních cyklů

Použití různého počtu vnitřních cyklů. Zvýšení počtu vnitřních cyklů má příliš velký dopad na rychlost algoritmu vzhledem k příliš malému zlepšení průměrné relativní chyby.

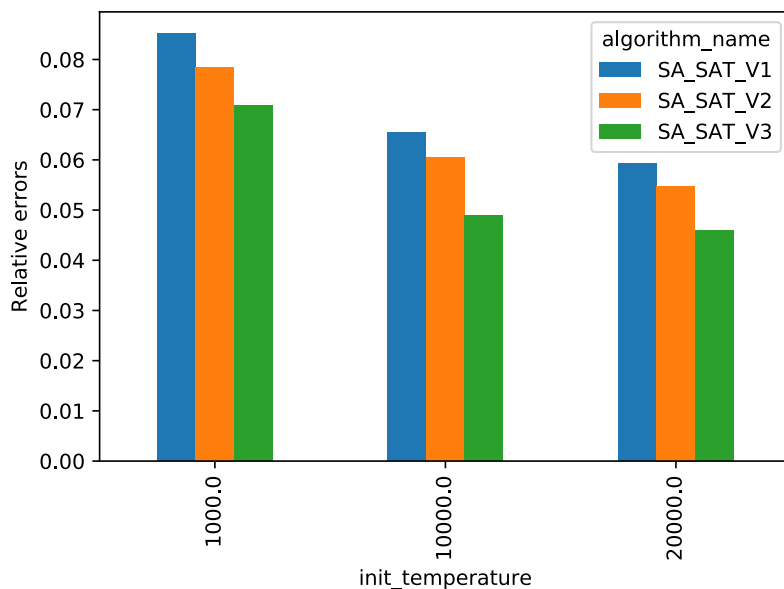


algorithm_name	% valid		
	SA_SAT_V1	SA_SAT_V2	SA_SAT_V3
cycles			
50	0,887	0,989	1
100	0,973	0,999	1

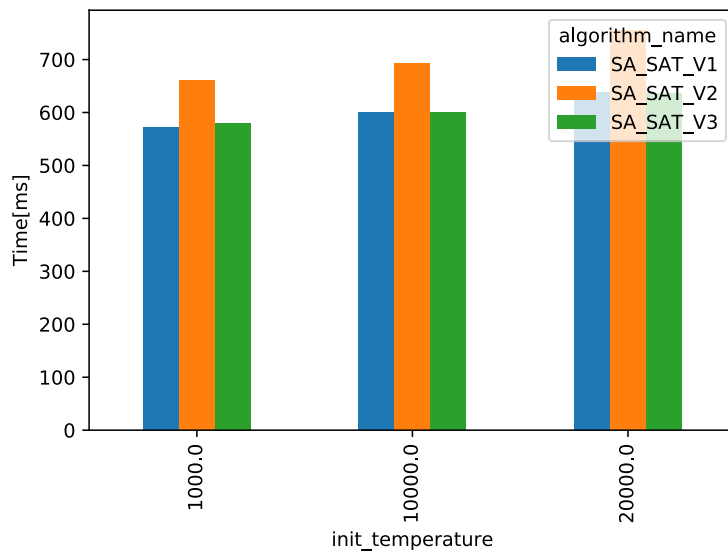
Počáteční teplota

Použití různých počátečních teplot. Zdá se, že použití teploty 10000 získám významné vylepšení přesnosti a zároveň průměrný čas se téměř nezvýší.

Avg error - init temperatures



Avg speed - init temperatures



	% valid		
algorithm_name	SA_SAT_V1	SA_SAT_V2	SA_SAT_V3
init_temperature			
1000	0,887	0,989	1
10000	0,885	0,991	1
20000	0,875	0,986	1

Nastavení parametrů na základě měření

Vzhledem k výsledkům měření se zdá toto nastavení parametrů jako nejlepší vzhledem k poměru velikosti průměrných relativních chyb a výkonu.

Parametr	Hodnoty
Počáteční teplota	10000
Konečná teplota	0,5
Koeficient ochlazování	0,95
Počet interních cyklů	50
Maximální počet resetů teploty	3

Závěr

V rámci této úlohy jsem implementoval algoritmus Simulovaného ochlazování (SA) pro problém vážené splnitelnosti booleanovské formule (SAT). Algoritmus nejdříve řeší problém MAX SAT (nalezení co nejvíce splněných klauzulí a celkového splnitelného řešení). Po nalezení splnitelného řešení pokračuje v řešení váženého SAT problému s cílem najít co nejvyšší cenu.

Můj algoritmus není omezen pouze na 3SAT, umožňuje přijmout libovolně velké klauzule.

Vyzkoušel jsem různé konfigurace algoritmu a algoritmus vylepšil tak, aby dosahoval přesnějších výsledků.

Algoritmus SAT_V3 i při nízko nastavených parametrech dokázal vždy najít validní řešení a velikost průměrné relativní chyby se pohybuje 5 %, což beru jako úspěch.

Příložené důležité soubory

Název	Popis
analysis	Složka s analýzou algoritmů. Obsahuje jupyter notebooky a výstupní hodnoty instancí, které byli algoritmy zpracovány.
data	Sady instancí řešeného <i>problému</i> .
algorithm_tester	Python modul testeru algoritmů.
tests	Unit testy programu.
analysis_sat.ipynb	Jupyter notebok použitý pro zpracování výstupních dat.
package_algorithms/sat	Balíček obsahující všechny implementované algoritmy pro řešení problému SAT.