AI LAB 3 | [VND, Beam Search, Tabu search]

Shreyas Sathe | 180010033
Janhavi Dadhania | 180010012

Brief Description:

State space :

Every state is defined using class having literal values. Each variable is either 0 or 1.
In case of tabu search state representation is modified a bit and one more list to store tabu tenure values is added to state as below.

Variable Neighbourhood Descent(VND) Beam Search	State: ([a,b,c,d])
Tabu Search	State: ([a,b,c,d], [M1,M2,M3,M4])

Example: initial state with all literal equal to 0 and all literals are allowed to change is represented as : ([0,0,0,0]) for VND and beam search and ([0,0,0,0], [0,0,0,0]) for tabu search algorithm.

Start node:

We are starting with all literal values 0 and it is allowed to change any literal in the next move. The state is represented as starting state: ([0,0,0,0])
In case of tabu search, initial tabu tenure list values are 0 for all literals and state is represented as starting state: ([0,0,0,0],[0,0,0,0])

Goal Node:

Goal node is a node with a literal list [a,b,c,d] such that boolean values of a,b,c,d when put into a given 4-SAT expression returns 1 i.e it satisfies the 4-SAT CNF expression.

MoveGen function

Pseudo code for the moveGen function is as below.

```
movGen():
    neighbours = []

    for each variable:
        change it's bit to negation of it (i.e change 0 to 1 and 1 to 0)
        add new node to neighbours

    return neighbours
```

Variable Neighbourhood Descent movGen

MovGen function for Variable Neighbourhood descent has n components for n-SAT formulas, each perturbing k number of bits and k E {1,2,3,...,n}

```
movGen: #for VND
    neighbours = []
    index_list = #list containing all possible combinations of k indices

    for each combination in index_list:
        change all bits (i.e take negation) at each position in combination
        add resultant Node to neighbours

    return neighbours
```

Tabu Search movGen

```
movgen(): #Tabu search
        Define list neighbours to store possible moves
        for each literal in node(a,b,c,d):
                if M_literal is 0: ## i.e if hasn't been changed recently
                        Change the value of literal ##change 0 to 1 and 1 to 0
                        Change M_literal value to tabu_tenure
                        Decrease all other M_x values to max((M_x - 1), 0)
                add modified node to neighbours list
        return neighbours
```

GoalTest():

Pseudo code for goalTest is as below.

```
goalTest():
        If [a,b,c,d] satisfies CNF expression :
                return true
        else :
                return false
```

Heuristic Function considered

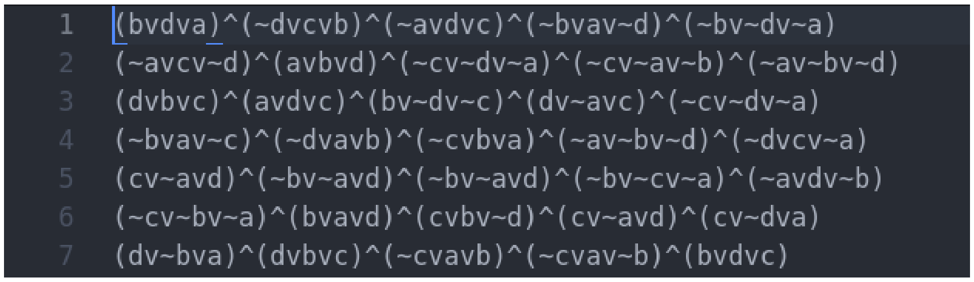
Heuristic Function returns an integer equal to the total number of clauses satisfied in the formula.

Input Generation

Generated clauses are of the form (_v_v_) ^ (_v_v_) ^ (_v_v_) ^ (_v_v_) ^ (_v_v_) with 4 variables a,b,c,d and 8 boolean values to choose from [a,~a,b,~b,c,~c,d,~d], each clause containing 3 literals and total 5 clauses.

Generating output :

- Run make file using command `$make testcases` you will be asked to type in total testcases to generate.
- All generated formulas will be stored in testcases.txt file in the same directory, one string per line as below.



```
(bvdva)^(~dvcvb)^(~avdvc)^(~bvav~d)^(~bv~dv~a)
(~avcv~d)^(avbvd)^(~cv~dv~a)^(~cv~av~b)^(~av~bv~d)
(dvbvc)^(avdvc)^(bv~dv~c)^(dv~avc)^(~cv~dv~a)
(~bvav~c)^(~dvavb)^(~cvbva)^(~av~bv~d)^(~dvcv~a)
(cv~avd)^(~bv~avd)^(~bv~avd)^(~bv~cv~a)^(~avdv~b)
(~cv~bv~a)^(bvavd)^(cvbv~d)^(cv~avd)^(cv~dva)
(dv~bva)^(dvbvc)^(~cvavb)^(~cvav~b)^(bvdvc)
```

NOTE: Zip folder includes multiple testcase generator to generate expressions with different number of variables. **Please read the README.md file for instructions on how to execute algorithm files and testcase generator scripts.**

To generate expressions with 4 literals per clauses, run testcase_generate-4cnf.py
To generate expressions with 7 variables and 4 literals per clauses, run 8_variable_testcase_generate-4cnf.py

Beam search analysis for different beam lengths

- When expressions have three literals in each clause and there are total 5 clauses, most of the expressions are getting satisfied in 0 or 1 iterations and very few takes 2 iterations.
- When analysed for different beam length, there is no change in total number of iterations required as irrespective of beam length expressions find solutions in the first move.
- Iterations required are tabulated for different beam lengths which are the same in this case.

Tabu search for different values of tabu tenure

- When expressions have three literals in each clause and there are total 5 clauses, most of the expressions are getting satisfied in 0 or 1 iterations and very few takes 2 iterations.
- When analysed for different tabu tenure, there is no change in total number of iterations required as irrespective of tabu tenure expressions find solutions in the first move.
- Iterations required are tabulated for different tabu tenures which are the same in this case.

Comparison of Variable neighborhood descent, Beam Search, Tabu Search: Nodes explored by each

- When expressions have three literals in each clause and there are total 5 clauses, most of the expressions are getting satisfied in 0 or 1 iterations and very few takes 2 iterations.
- All three algorithms explores 1 (i.e starting node) when initiated state is solution
- All three algorithms explores 5 (i.e starting node + 4 nodes with one bit perturbed) when solution is found in first iterations
- For few of the expressions solution is found in second iterations in that case
-

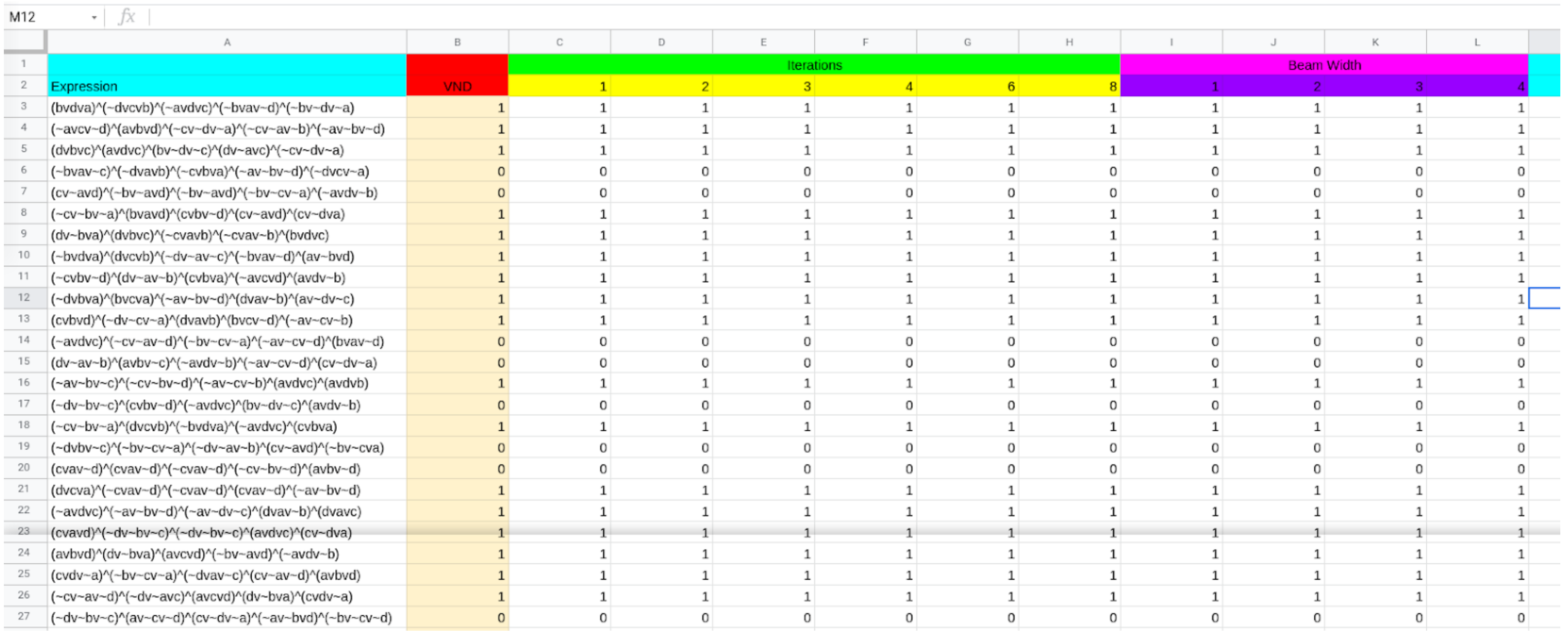
- **VND explores only 3 nodes** (i.e starting node, one node at level one and third solution node)
- **Beam search explores 4 nodes** (i.e starting node, two nodes at level two and third solution node) as the **beam width is increased** algorithm explores **more** nodes in vain.
- **Tabu search explores 3 nodes** (i.e starting node, one node at level one and third solution node)

Experiments with 7 variables

- When algorithms were run on expressions with 7 variables, comparatively more iterations were taken in all three cases. (results tabulated below)
- The iterations required were still almost similar for different tabu tenures and beam lengths because most of the expressions were getting **satisfied after just one or two perturbations in bits.**

Tabulating values:

https://docs.google.com/spreadsheets/d/1T7h70UADdVpgBV2loeKOQRR5rIjc-8gBSGCT6WVy7YY/edit?usp=sharing (full table could be found at the link above,link contains multiple sheets)

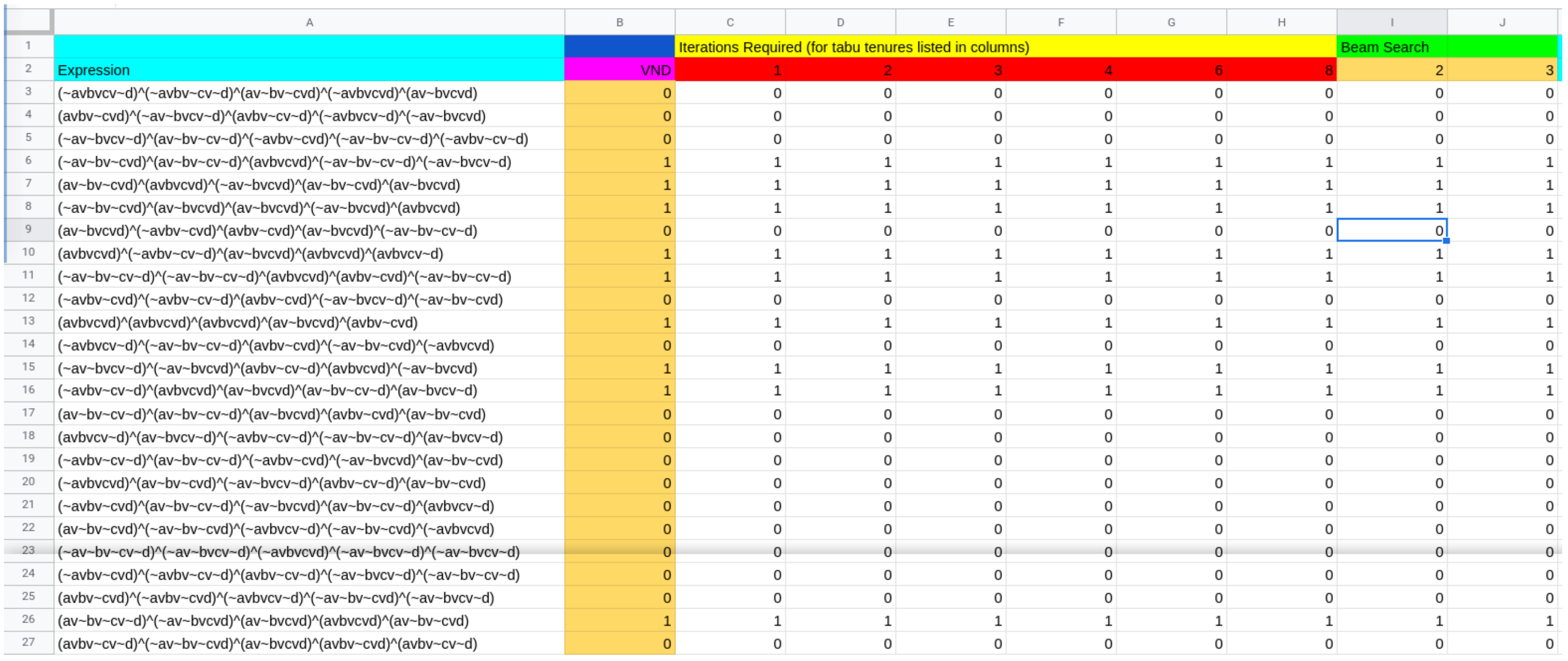


		Iterations						Beam Width			
Expression	VND	1	2	3	4	6	8	1	2	3	4
(bvdva)^(~dvcvb)^(~avdvc)^(~bvav~d)^(~bv~dv~a)	1	1	1	1	1	1	1	1	1	1	1
(~avcv~d)^(avbvd)^(~cv~dv~a)^(~cv~av~b)^(~av~bv~d)	1	1	1	1	1	1	1	1	1	1	1
(dvbvc)^(avdvc)^(bv~dv~c)^(dv~avc)^(~cv~dv~a)	1	1	1	1	1	1	1	1	1	1	1
(~bvav~c)^(~dvavb)^(~cvbva)^(~av~bv~d)^(~dvcv~a)	0	0	0	0	0	0	0	0	0	0	0
(cv~avd)^(~bv~avd)^(~bv~avd)^(~bv~cv~a)^(~avdv~b)	0	0	0	0	0	0	0	0	0	0	0
(~cv~bv~a)^(bvavd)^(cvbv~d)^(cv~avd)^(cv~dva)	1	1	1	1	1	1	1	1	1	1	1
(dv~bva)^(dvbvc)^(~cvavb)^(~cvav~b)^(bvdvc)	1	1	1	1	1	1	1	1	1	1	1
(~bvdva)^(dvcvb)^(~dv~av~c)^(~bvav~d)^(av~bvd)	1	1	1	1	1	1	1	1	1	1	1
(~cvbv~d)^(dv~av~b)^(cvbva)^(~avcvd)^(avdv~b)	1	1	1	1	1	1	1	1	1	1	1
(~dvbva)^(bvcva)^(~av~bv~d)^(dvav~b)^(av~dv~c)	1	1	1	1	1	1	1	1	1	1	1
(cvbvd)^(~dv~cv~a)^(dvavb)^(bvcv~d)^(~av~cv~b)	1	1	1	1	1	1	1	1	1	1	1
(~avdvc)^(~cv~av~d)^(~bv~cv~a)^(~av~cv~d)^(bvav~d)	0	0	0	0	0	0	0	0	0	0	0
(dv~av~b)^(avbv~c)^(~avdv~b)^(~av~cv~d)^(cv~dv~a)	0	0	0	0	0	0	0	0	0	0	0
(~av~bv~c)^(~cv~bv~d)^(~av~cv~b)^(avdvc)^(avdvb)	1	1	1	1	1	1	1	1	1	1	1
(~dv~bv~c)^(cvbv~d)^(~avdvc)^(bv~dv~c)^(avdv~b)	0	0	0	0	0	0	0	0	0	0	0
(~cv~bv~a)^(dvcvb)^(~bvdva)^(~avdvc)^(cvbva)	1	1	1	1	1	1	1	1	1	1	1
(~dvbv~c)^(~bv~cv~a)^(~dv~av~b)^(cv~avd)^(~bv~cva)	0	0	0	0	0	0	0	0	0	0	0
(cvav~d)^(cvav~d)^(~cvav~d)^(~cv~bv~d)^(avbv~d)	0	0	0	0	0	0	0	0	0	0	0
(dvcva)^(~cvav~d)^(~cvav~d)^(cvav~d)^(~av~bv~d)	1	1	1	1	1	1	1	1	1	1	1
(~avdvc)^(~av~bv~d)^(~av~dv~c)^(dvav~b)^(dvavc)	1	1	1	1	1	1	1	1	1	1	1
(cvavd)^(~dv~bv~c)^(~dv~bv~c)^(avdvc)^(cv~dva)	1	1	1	1	1	1	1	1	1	1	1
(avbvd)^(dv~bva)^(avcvd)^(~bv~avd)^(~avdv~b)	1	1	1	1	1	1	1	1	1	1	1
(cvdv~a)^(~bv~cv~a)^(~dvav~c)^(cv~av~d)^(avbvd)	1	1	1	1	1	1	1	1	1	1	1
(~cv~av~d)^(~dv~avc)^(avcvd)^(dv~bva)^(cvdv~a)	1	1	1	1	1	1	1	1	1	1	1
(~dv~bv~c)^(av~cv~d)^(cv~dv~a)^(~av~bvd)^(~bv~cv~d)	0	0	0	0	0	0	0	0	0	0	0

Almost similar result is obtained when each clause contains 4 literals (4SAT)from 8

Result is tabulated below.

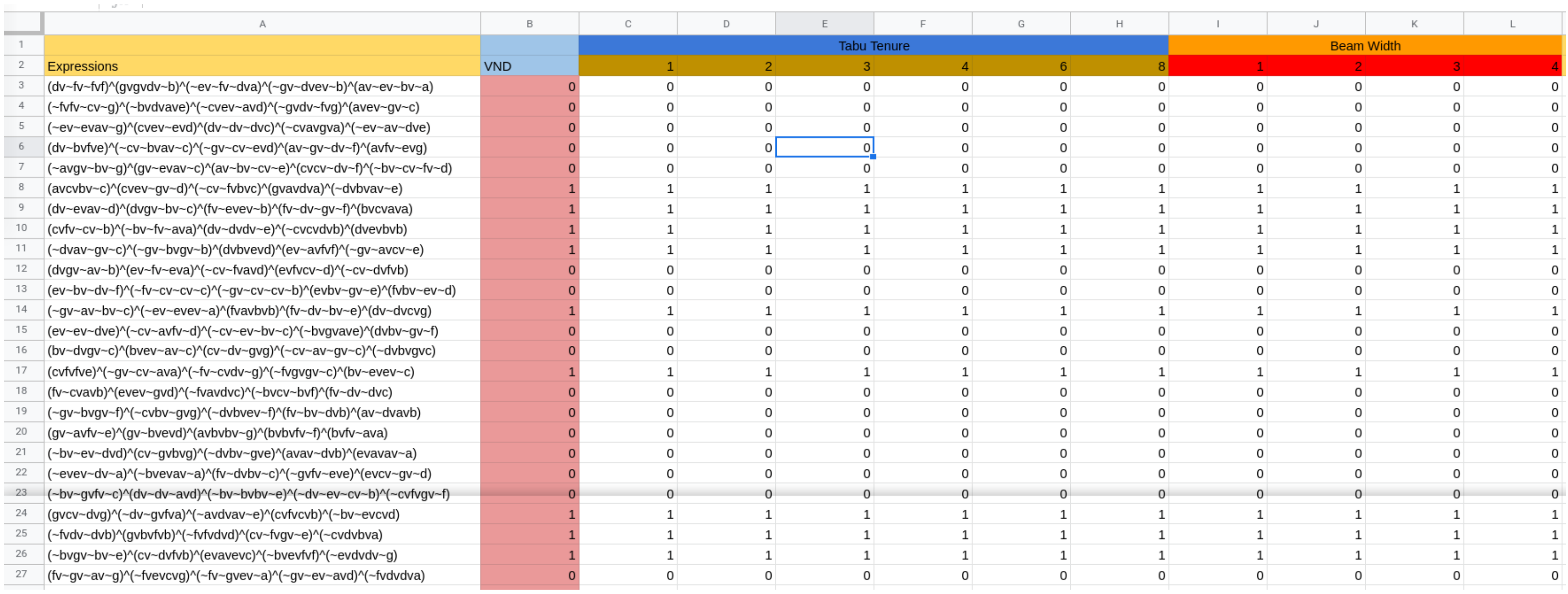
https://docs.google.com/spreadsheets/d/1T7h70UADdVpgBV2loeKOQRR5rIjc-8gBSGCT6WVy7YY/edit?usp=sharing



		Iterations Required (for tabu tenures listed in columns)						Beam Search	
Expression	VND	1	2	3	4	6	8	2	3
(~avbvcv~d)^(~avbv~cv~d)^(av~bv~cvd)^(~avbvcvd)^(av~bvcvd)	0	0	0	0	0	0	0	0	0
(avbv~cvd)^(~av~bvcv~d)^(avbv~cv~d)^(~avbvcv~d)^(~av~bvcvd)	0	0	0	0	0	0	0	0	0
(~av~bvcv~d)^(av~bv~cv~d)^(~avbv~cvd)^(~av~bv~cv~d)^(~avbv~cv~d)	0	0	0	0	0	0	0	0	0
(~av~bv~cvd)^(av~bv~cv~d)^(avbvcvd)^(~av~bv~cv~d)^(~av~bvcv~d)	1	1	1	1	1	1	1	1	1
(av~bv~cvd)^(avbvcvd)^(~av~bvcvd)^(av~bv~cvd)^(av~bvcvd)	1	1	1	1	1	1	1	1	1
(~av~bv~cvd)^(av~bvcvd)^(av~bvcvd)^(~av~bvcvd)^(avbvcvd)	1	1	1	1	1	1	1	1	1
(av~bvcvd)^(~avbv~cvd)^(avbv~cvd)^(av~bvcvd)^(~av~bv~cv~d)	0	0	0	0	0	0	0	0	0
(avbvcvd)^(~avbv~cv~d)^(av~bvcvd)^(avbvcvd)^(avbvcv~d)	1	1	1	1	1	1	1	1	1
(~av~bv~cv~d)^(~av~bv~cv~d)^(avbvcvd)^(avbv~cvd)^(~av~bv~cv~d)	1	1	1	1	1	1	1	1	1
(~avbv~cvd)^(~avbv~cv~d)^(avbv~cvd)^(~av~bvcv~d)^(~av~bv~cvd)	0	0	0	0	0	0	0	0	0
(avbvcvd)^(avbvcvd)^(avbvcvd)^(av~bvcvd)^(avbv~cvd)	1	1	1	1	1	1	1	1	1
(~avbvcv~d)^(~av~bv~cv~d)^(avbv~cvd)^(~av~bv~cvd)^(~avbvcvd)	0	0	0	0	0	0	0	0	0
(~av~bvcv~d)^(~av~bvcvd)^(avbv~cv~d)^(avbvcvd)^(~av~bvcvd)	1	1	1	1	1	1	1	1	1
(~avbv~cv~d)^(avbvcvd)^(av~bvcvd)^(av~bv~cv~d)^(av~bvcv~d)	1	1	1	1	1	1	1	1	1
(av~bv~cv~d)^(av~bv~cv~d)^(av~bvcvd)^(avbv~cvd)^(av~bv~cvd)	0	0	0	0	0	0	0	0	0
(avbvcv~d)^(av~bvcv~d)^(~avbv~cv~d)^(~av~bv~cv~d)^(av~bvcv~d)	0	0	0	0	0	0	0	0	0
(~avbv~cv~d)^(av~bv~cv~d)^(~avbv~cvd)^(~av~bvcvd)^(av~bv~cvd)	0	0	0	0	0	0	0	0	0
(~avbvcvd)^(av~bv~cvd)^(~av~bvcv~d)^(avbv~cv~d)^(av~bv~cvd)	0	0	0	0	0	0	0	0	0
(~avbv~cvd)^(av~bv~cv~d)^(~av~bvcvd)^(av~bv~cv~d)^(avbvcv~d)	0	0	0	0	0	0	0	0	0
(av~bv~cvd)^(~av~bv~cvd)^(~avbvcv~d)^(~av~bv~cvd)^(~avbvcvd)	0	0	0	0	0	0	0	0	0
(~av~bv~cv~d)^(~av~bvcv~d)^(~avbvcvd)^(~av~bvcv~d)^(~av~bvcv~d)	0	0	0	0	0	0	0	0	0
(~avbv~cvd)^(~avbv~cv~d)^(avbv~cv~d)^(~av~bvcv~d)^(~av~bv~cv~d)	0	0	0	0	0	0	0	0	0
(avbv~cvd)^(~avbv~cvd)^(~avbvcv~d)^(~av~bv~cvd)^(~av~bvcv~d)	0	0	0	0	0	0	0	0	0
(av~bv~cv~d)^(~av~bvcvd)^(av~bvcvd)^(avbvcvd)^(av~bv~cvd)	1	1	1	1	1	1	1	1	1
(avbv~cv~d)^(~av~bv~cvd)^(av~bvcvd)^(avbv~cvd)^(avbv~cv~d)	0	0	0	0	0	0	0	0	0

7 Variables - 5 clauses - four literals per clause

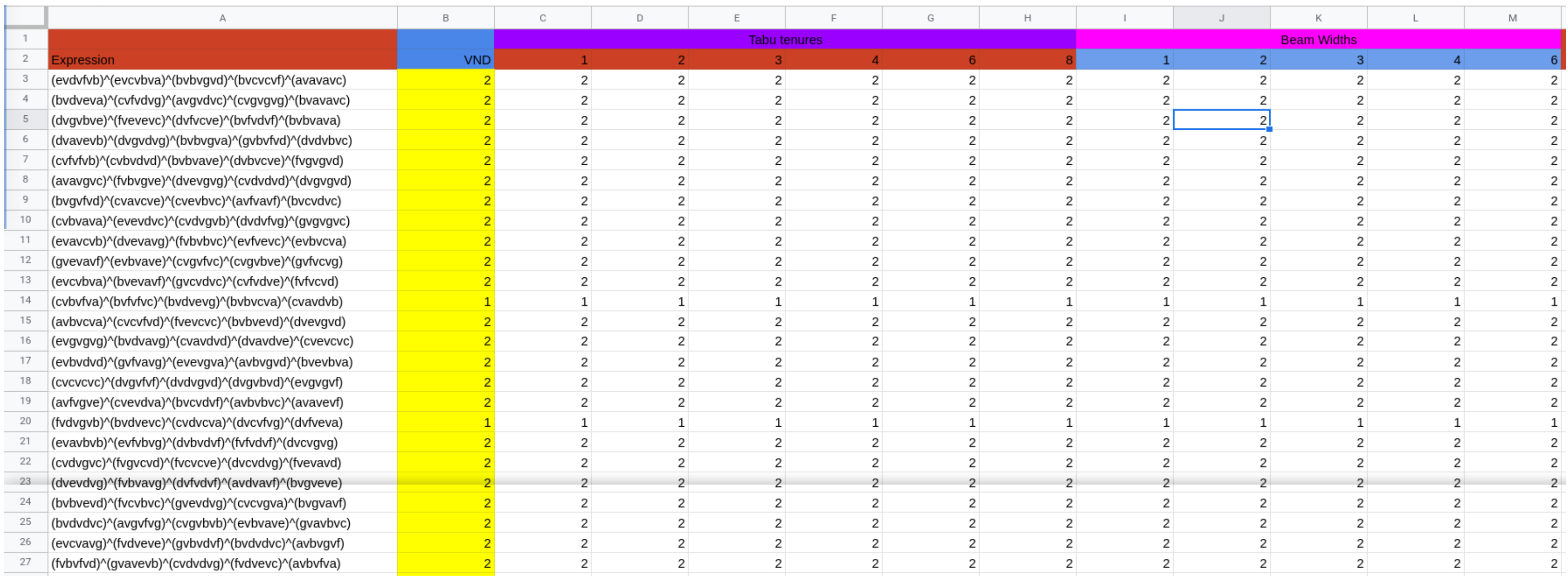
https://docs.google.com/spreadsheets/d/1T7h70UADdVpgBV2loeKOQRR5rIjc-8gBSGCT6WVy7YY/edit?usp=sharing



	A	B	C	D	E	F	G	H	I	J	K	L
1			Tabu Tenure						Beam Width			
2	Expressions	VND	1	2	3	4	6	8	1	2	3	4
3	(dv~fv~fvf)^(gvgvdv~b)^(~ev~fv~dva)^(~gv~dvev~b)^(av~ev~bv~a)	0	0	0	0	0	0	0	0	0	0	0
4	(~fvfv~cv~g)^(~bvdvave)^(~cvev~avd)^(~gvdv~fvg)^(avev~gv~c)	0	0	0	0	0	0	0	0	0	0	0
5	(~ev~evav~g)^(cvev~evd)^(dv~dv~dvc)^(~cvavgva)^(~ev~av~dve)	0	0	0	0	0	0	0	0	0	0	0
6	(dv~bvfve)^(~cv~bvav~c)^(~gv~cv~evd)^(av~gv~dv~f)^(avfv~evg)	0	0	0	0	0	0	0	0	0	0	0
7	(~avgv~bv~g)^(gv~evav~c)^(av~bv~cv~e)^(cvcv~dv~f)^(~bv~cv~fv~d)	0	0	0	0	0	0	0	0	0	0	0
8	(avcvbv~c)^(cvev~gv~d)^(~cv~fvbvc)^(gvavdva)^(~dvbvav~e)	1	1	1	1	1	1	1	1	1	1	1
9	(dv~evav~d)^(dvgv~bv~c)^(fv~evev~b)^(fv~dv~gv~f)^(bvcvava)	1	1	1	1	1	1	1	1	1	1	1
10	(cvfv~cv~b)^(~bv~fv~ava)^(dv~dvdv~e)^(~cvcvdvb)^(dvevbvb)	1	1	1	1	1	1	1	1	1	1	1
11	(~dvav~gv~c)^(~gv~bvgv~b)^(dvbvevd)^(ev~avfvf)^(~gv~avcv~e)	1	1	1	1	1	1	1	1	1	1	1
12	(dvgv~av~b)^(ev~fv~eva)^(~cv~fvavd)^(evfvcv~d)^(~cv~dvfvb)	0	0	0	0	0	0	0	0	0	0	0
13	(ev~bv~dv~f)^(~fv~cv~cv~c)^(~gv~cv~cv~b)^(evbv~gv~e)^(fvbv~ev~d)	0	0	0	0	0	0	0	0	0	0	0
14	(~gv~av~bv~c)^(~ev~evev~a)^(fvavbvb)^(fv~dv~bv~e)^(dv~dvcvg)	1	1	1	1	1	1	1	1	1	1	1
15	(ev~ev~dve)^(~cv~avfv~d)^(~cv~ev~bv~c)^(~bvgvave)^(dvbv~gv~f)	0	0	0	0	0	0	0	0	0	0	0
16	(bv~dvgv~c)^(bvev~av~c)^(cv~dv~gvg)^(~cv~av~gv~c)^(~dvbvgvc)	0	0	0	0	0	0	0	0	0	0	0
17	(cvfvfve)^(~gv~cv~ava)^(~fv~cvdv~g)^(~fvgvgv~c)^(bv~evev~c)	1	1	1	1	1	1	1	1	1	1	1
18	(fv~cvavb)^(evev~gvd)^(~fvavdvc)^(~bvcv~bvf)^(fv~dv~dvc)	0	0	0	0	0	0	0	0	0	0	0
19	(~gv~bvgv~f)^(~cvbv~gvg)^(~dvbvev~f)^(fv~bv~dvb)^(av~dvavb)	0	0	0	0	0	0	0	0	0	0	0
20	(gv~avfv~e)^(gv~bvevd)^(avbvbv~g)^(bvbvfv~f)^(bvfv~ava)	0	0	0	0	0	0	0	0	0	0	0
21	(~bv~ev~dvd)^(cv~gvbvg)^(~dvbv~gve)^(avav~dvb)^(evavav~a)	0	0	0	0	0	0	0	0	0	0	0
22	(~evev~dv~a)^(~bvevav~a)^(fv~dvbv~c)^(~gvfv~eve)^(evcv~gv~d)	0	0	0	0	0	0	0	0	0	0	0
23	(~bv~gvfv~c)^(dv~dv~avd)^(~bv~bvbv~e)^(~dv~ev~cv~b)^(~cvfvgv~f)	0	0	0	0	0	0	0	0	0	0	0
24	(gvcv~dvg)^(~dv~gvfva)^(~avdvav~e)^(cvfvcvb)^(~bv~evcvd)	1	1	1	1	1	1	1	1	1	1	1
25	(~fvdv~dvb)^(gvbvfvb)^(~fvfvdvd)^(cv~fvgv~e)^(~cvdvbva)	1	1	1	1	1	1	1	1	1	1	1
26	(~bvgv~bv~e)^(cv~dvfvb)^(evavevc)^(~bvevfvf)^(~evdvdv~g)	1	1	1	1	1	1	1	1	1	1	1
27	(fv~gv~av~g)^(~fvevcvg)^(~fv~gvev~a)^(~gv~ev~avd)^(~fvdvdva)	0	0	0	0	0	0	0	0	0	0	0

7 literals - 5 clauses - 4 literals per clause - negation not included for each variable

https://docs.google.com/spreadsheets/d/1T7h70UADdVpgBV2loeKOQRR5rIjc-8gBSGCT6WVy7YY/edit?usp=sharing



	A	B	C	D	E	F	G	H	I	J	K	L	M
1			Tabu tenures						Beam Widths				
2	Expression	VND	1	2	3	4	6	8	1	2	3	4	6
3	(evdvfvb)^(evcvbva)^(bvbvgvd)^(bvcvcvf)^(avavavc)	2	2	2	2	2	2	2	2	2	2	2	2
4	(bvdveva)^(cvfvdvg)^(avgvdvc)^(cvgvgvg)^(bvavavc)	2	2	2	2	2	2	2	2	2	2	2	2
5	(dvgvbve)^(fvevevc)^(dvfvcve)^(bvfvdvf)^(bvbvava)	2	2	2	2	2	2	2	2	2	2	2	2
6	(dvavevb)^(dvgvdvg)^(bvbvgva)^(gvbvfvd)^(dvdvbvc)	2	2	2	2	2	2	2	2	2	2	2	2
7	(cvfvfvb)^(cvbvdvd)^(bvbvave)^(dvbvcve)^(fvgvgvd)	2	2	2	2	2	2	2	2	2	2	2	2
8	(avavgvc)^(fvbvgve)^(dvevgvg)^(cvdvdvd)^(dvgvgvd)	2	2	2	2	2	2	2	2	2	2	2	2
9	(bvgvfvd)^(cvavcve)^(cvevbvc)^(avfvavf)^(bvcvdvc)	2	2	2	2	2	2	2	2	2	2	2	2
10	(cvbvava)^(evevdvc)^(cvdvgvb)^(dvdvfvg)^(gvgvgvc)	2	2	2	2	2	2	2	2	2	2	2	2
11	(evavcvb)^(dvevavg)^(fvbvbvc)^(evfvevc)^(evbvcva)	2	2	2	2	2	2	2	2	2	2	2	2
12	(gvevavf)^(evbvave)^(cvgvfvc)^(cvgvbve)^(gvfvcvg)	2	2	2	2	2	2	2	2	2	2	2	2
13	(evcvbva)^(bvevavf)^(gvcvdvc)^(cvfvdve)^(fvfvcvd)	2	2	2	2	2	2	2	2	2	2	2	2
14	(cvbvfva)^(bvfvfvc)^(bvdvevg)^(bvbvcva)^(cvavdvb)	1	1	1	1	1	1	1	1	1	1	1	1
15	(avbvcva)^(cvcvfvd)^(fvevcvc)^(bvbvevd)^(dvevgvd)	2	2	2	2	2	2	2	2	2	2	2	2
16	(evgvgvg)^(bvdvavg)^(cvavdvd)^(dvavdve)^(cvevcvc)	2	2	2	2	2	2	2	2	2	2	2	2
17	(evbvdvd)^(gvfvavg)^(evevgva)^(avbvgvd)^(bvevbva)	2	2	2	2	2	2	2	2	2	2	2	2
18	(cvcvcvc)^(dvgvfvf)^(dvdvgvd)^(dvgvbvd)^(evgvgvf)	2	2	2	2	2	2	2	2	2	2	2	2
19	(avfvgve)^(cvevdva)^(bvcvdvf)^(avbvbvc)^(avavevf)	2	2	2	2	2	2	2	2	2	2	2	2
20	(fvdvgvb)^(bvdvevc)^(cvdvcva)^(dvcvfvg)^(dvfveva)	1	1	1	1	1	1	1	1	1	1	1	1
21	(evavbvb)^(evfvbvg)^(dvbvdvf)^(fvfvdvf)^(dvcvgvg)	2	2	2	2	2	2	2	2	2	2	2	2
22	(cvdvgvc)^(fvgvcvd)^(fvcvcve)^(dvcvdvg)^(fvevavd)	2	2	2	2	2	2	2	2	2	2	2	2
23	(dvevdvg)^(fvbvavg)^(dvfvdvf)^(avdvavf)^(bvgveve)	2	2	2	2	2	2	2	2	2	2	2	2
24	(bvbvevd)^(fvcvbvc)^(gvevdvg)^(cvcvgva)^(bvgvavf)	2	2	2	2	2	2	2	2	2	2	2	2
25	(bvdvdvc)^(avgvfvg)^(cvgvbvb)^(evbvave)^(gvavbvc)	2	2	2	2	2	2	2	2	2	2	2	2
26	(evcvavg)^(fvdveve)^(gvbvdvf)^(bvdvdvc)^(avbvgvf)	2	2	2	2	2	2	2	2	2	2	2	2
27	(fvbvfvd)^(gvavevb)^(cvdvdvg)^(fvdvevc)^(avbvfva)	2	2	2	2	2	2	2	2	2	2	2	2

Conclusion:

Total Variables	Performance Comparison
4	All three algorithms perform the same irrespective of tabu tenure or beam width.
7	Even with 7 variables all algorithms performed almost similar and thus optimal tabu tenure is 1 and beam width is 1.
7 (no negation allowed)	To introduce more strict criteria, negations were not included in expressions i.e literals were randomly drawn from (a,b,c,d,e,f,g) only. Total number of iterations to reach the goal state increased compared to cases when negations were taken into consideration.

Thus optimal tabu tenure is 1 and optimal beam width is 1 for SAT problems with 4 variables.