Meta Heuristic Assignment 2

Submitted by: Phalguni Rathod | Student Id: R00183770

Part 1: GWSAT

When GSAT algorithm is topped-up with RandomWalk Step, then the algorithm developed is the GWSAT.

1. In GWSAT, we start with generating random initial solution for given set of clauses.
2. Then, we find a list of all the unique variables in all the clauses.
3. Once, we get this list, then we start flipping each variable one by one and compute net gain for each of them, this net gain can be stored in the dictionary (as in my code).
4. Then we find the variable with the highest net gain and flip it to get a new assignment of solution and check if it satisfies the given clauses.
5. If no then continue till maximum #iterations/#flip given.

Main parameters for GWSAT are:

1. No. of Execution
2. No. of Restart
3. No. of Iterations
4. Wp (walk probability)

We are going to experiment with these parameters by varying them but only #Execution will be constant for all the experiments.

Comprehensive Description of Experiments:

We are going to experiment with 2 values for each parameter.

1. Restart: 10, 100
2. Iterations: 1000, 2000
3. Wp (Walk Probability): 0.2, 0.6

All the experiments are comprehensive i.e it includes all possible combinations of the parameters values taken by us for experimentation.

2 - values per parameter, 3 - parameters ⇒ 2^3 ⇒ 8 parameter sets to experiment with, for each instance for 30 execution.

Results to be judged on:

1. No. of Solutions obtained from 30 Executions for each parameter set
2. No. of Unique Solutions generated out of solutions obtained for each parameter set

Instance: uf20-01.cnf

20-01_GWSAT				
Restarts	**Iterations**	**Wp**	**#Solution Obtained**	**#Unique Solution**
10	1000	0.2	30	6
		0.6	30	7
	2000	0.2	30	6
		0.6	30	7
100	1000	0.2	30	6
		0.6	30	7
	2000	0.2	30	6
		0.6	30	7

Evaluation & Analysis:

- We are getting solution in each execution of the algorithm
- No. of uniques solutions varying with parameters
- Effect of: (Keeping all other parameters at constant)
    - Wp: With change in wp result is changing. Hence there is some effect of wp.
    - Iterations: No change in result.
    - Restarts: No change in result.
- Analysis: The result is changing with change in wp and nothing else, it implies that selection of algorithm is changing, when wp == 0.2, there is a high probability(0.9) that algorithm selected is GSAT and when wp == 0.6, there is an equal probability of selection of both the algorithm(Random Walk or GSAT)

Instance: uf20-02.cnf

20-02_GWSAT				
Restarts	**Iterations**	**Wp**	**#Solution Obtained**	**#Unique Solution**
10	1000	0.2	30	15
		0.6	30	15
	2000	0.2	30	15
		0.6	30	15
100	1000	0.2	30	15
		0.6	30	15
	2000	0.2	30	15
		0.6	30	15

Evaluation & Analysis:

- We are getting solution in each execution of the algorithm
- No. of uniques solutions doesn't vary with parameters
- Effect of: (Keeping all other parameters at constant)
    - Wp: No change in result
    - Iterations: No change in result.
    - Restarts: No change in result.
- Analysis: The result is not changing with change in any parameter. It is difficult to comment on the reasoning behind it yet one reason could be selection of variable is similar in all the cases which got us this result.

Instance: uf50-01.cnf

50-01_GWSAT				
Restarts	**Iterations**	**Wp**	**#Solution Obtained**	**#Unique Solution**
10	1000	0.2	30	16
		0.6	30	18
	2000	0.2	30	16
		0.6	30	18
100	1000	0.2	30	16
		0.6	30	18
	2000	0.2	30	16
		0.6	30	18

Evaluation & Analysis:

- We are getting solution in each execution of the algorithm
- No. of uniques solutions varying with parameters
- Effect of: (Keeping all other parameters at constant)
    - Wp: With change in wp result is changing. Hence there is some effect of wp.
    - Iterations: No change in result.
    - Restarts: No change in result.
- Analysis: The result is changing with change in wp and nothing else, it implies that selection of algorithm is changing, when wp == 0.2, there is a high probability(0.9) that algorithm selected is GSAT and when wp == 0.6, there is an equal probability of selection of both the algorithm(Random Walk or GSAT)

Part 2: WalkSAT with Tabu

WalkSAT

All the general steps of WalkSAT is similar to GWSAT, just the selection of variable to be flipped is the core difference.

1. In WalkSAT we calculate negative gain of variables and flip the one
    a. If Negative gain = 0, (break ties at random). ⇒ Flip it
    b. Else generate random prob and compare with wp,
        i. if wp<0.2 ⇒ Choose variable randomly from the clause & flip
        ii. Else, thake variable with minimum negative gain ⇒ Flip it

With Tabu

Tabu is a fixed length data structure where we store recently flipped variables and remove least recently flipped variables.

When we select a variable for negative gain calculation, we first check if it exist in tabu, if it does we skip it and move to the next variable. If all the variables in the clause are in tabu then we move to the next iteration.

- **My Implementation of tabu:** I have used a dictionary to store variables as keys and their iteration number when they can be flipped next(cuurent_itr + tl). If the current iteration number is less then that iteration number than that variable can't be flipped because it is tabu-ed.

Similar to GWSAT, we have our main parameters as:

1. No. of Execution
2. No. of Restart
3. No. of Iterations
4. Wp (walk probability)
5. TL (Tabu length)

We are going to experiment with these parameters by varying them but only #Execution will be constant for all the experiments.

Comprehensive Description of Experiments:

We are going to experiment with 2 values for each parameter.

1. Restart: 10, 100
2. Iterations: 1000, 2000
3. Wp (Walk Probability): 0.2, 0.6
4. Tl (Tabu length): 7, 11

All the experiments are comprehensive i.e it includes all possible combinations of the parameters values taken by us for experimentation.

2 - values per parameter, 4 - parameters ⇒ 2^4 ⇒ 16 parameter sets to experiment with, for each instance for 30 execution.

Results to be judged on:

1. No. of Solutions obtained from 30 Executions for each parameter set
2. No. of Unique Solutions generated out of solutions obtained for each parameter set

Instance: uf20-01.cnf

20-01_WS					
Restarts	Iterations	Wp	Tabu Length	#Solution Obtained	#Unique Solution
10	1000	0.2	7	30	7
			11	30	7
		0.6	7	30	7
			11	30	7
	2000	0.2	7	30	7
			11	30	7
		0.6	7	30	7
			11	30	7
100	1000	0.2	7	30	7
			11	30	7
		0.6	7	30	7
			11	30	7
	2000	0.2	7	30	7
			11	30	7
		0.6	7	30	7
			11	30	7

Evaluation & Analysis:

- We are getting solution in each execution of the algorithm
- No. of uniques solutions doesn't vary with parameters
- Effect of: (Keeping all other parameters at constant)
    - Wp: No change in result
    - Iterations: No change in result.
    - Restarts: No change in result.
    - Tl: No Change in result.
- Analysis: The result is not changing with change in any parameter. It is difficult to comment on the reasoning behind it yet one reason could be selection of variable is similar in all the cases which got us this result.

Instance: uf20-02.cnf

20-02_WS					
Restarts	Iterations	Wp	Tabu Length	#Solution Obtained	#Unique Solution
10	1000	0.2	7	30	16
			11	30	15
		0.6	7	30	14
			11	30	17
	2000	0.2	7	30	16
			11	30	15
		0.6	7	30	14
			11	30	17
100	1000	0.2	7	30	16
			11	30	15
		0.6	7	30	14
			11	30	17
	2000	0.2	7	30	16
			11	30	15
		0.6	7	30	14
			11	30	17

Evaluation & Analysis:

- We are getting solution in each execution of the algorithm
- No. of uniques solutions varying with parameters
- Effect of: (Keeping all other parameters at constant)
    - Tl: With change in tl, result is changing. Hence there is some effect of wp.
    - Wp: With change in wp, result is changing. Hence there is some effect of wp.
    - Iterations: No change in result.
    - Restarts: No change in result.
- Analysis: The result is changing with change in tl and wp,
    - We are getting maximum #UniqueSolution when tabu list is long i.e 11 & wp is unbiased (have equal prob to take random walk or minimum negative gain).
    - This could be because as the tabu list is long it doesn't allow more variables to flip (11 out of 20 will be tabu-ed), and algorithm is not going to the same solution again & again. Hence, it is diversifying the solution and we are getting more unique solution.

Instance: uf50-01.cnf

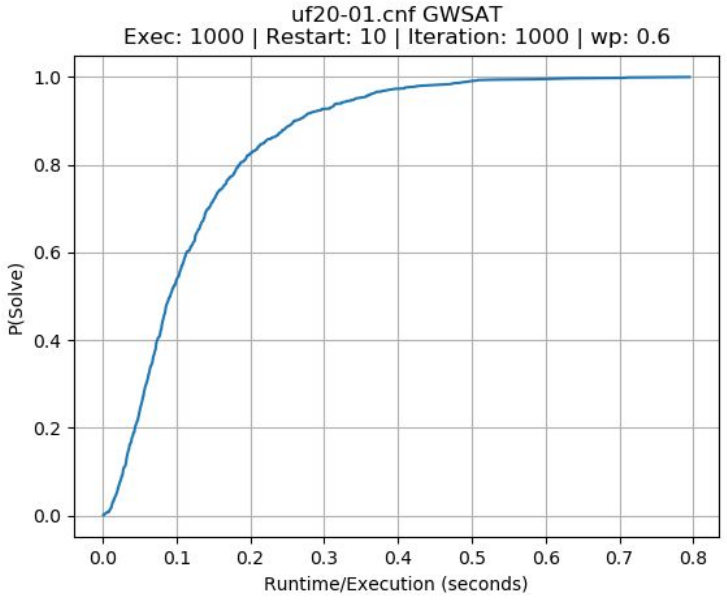
50-01_WS					
Restarts	Iterations	Wp	Tabu Length	#Solution Obtained	#Unique Solution
10	1000	0.2	7	1	1
			11	2	2
		0.6	7	3	3
			11	1	1
	2000	0.2	7	2	2
			11	1	1
		0.6	7	4	3
			11	1	1
100	1000	0.2	7	17	8
			11	12	8
		0.6	7	12	9
			11	11	10
	2000	0.2	7	20	11
			11	22	10
		0.6	7	17	10
			11	10	6

Evaluation & Analysis:

- We are getting limited number of solution in each execution of the algorithm
- No. of uniques solutions varying with parameters. Interesting pattern can be seen
    - In restart == 10, #Solution & #UniqueSolution are mostly similar
    - In restart == 100, we can see they are varying
- Effect of: (Keeping all other parameters at constant)
    - Tl: With change in tl, result is changing. Hence there is some effect of wp.
    - Wp: With change in wp, result is changing. Hence there is some effect of wp.
    - Iterations: With change in Iterations, result is changing. Hence there is some effect of Iterations.
    - Restarts: With change in Restarts, result is changing. Hence there is some effect of Restarts.
- Analysis: The result is changing with every set of parameter list.
    - Restarts: As mentioned above we can see a pattern, along with that it can be observed that increase in number of restarts have increased #SolutionObtained & hence, increase #UniqueSolution.
    - Iterations: There is very little change due to increase in iteration for restart=10. But major change can be seen when we keep restart = 10, iteration = 1000,2000, wp=0.2 & tl = 11 ⇒ #SolutionsObtained change from 12 to 22, which is a big increase but #UniqueSolution didn't see much growth
    - Wp & Tl: Yes, there are changes due wp & tl in results but these aren't the major shift in results.

Part 3: RTD

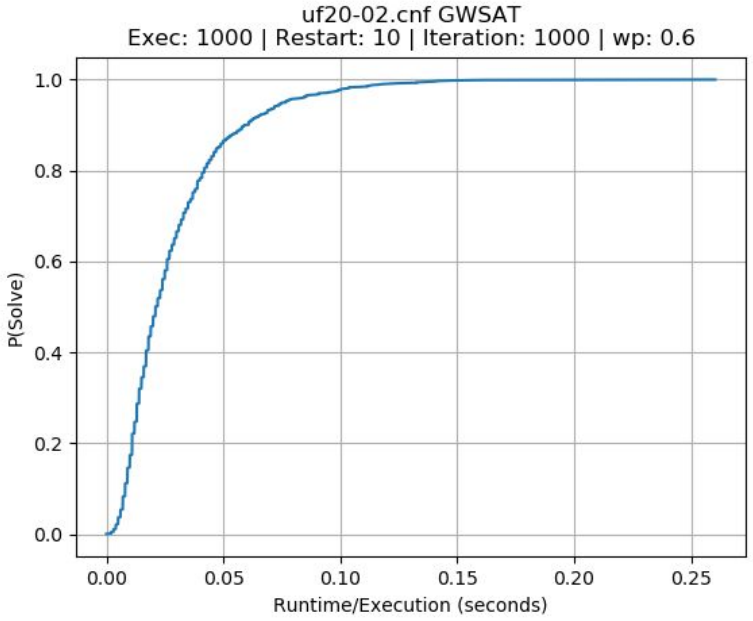
uf20-01-GWSAT



1. Instance: uf20-01
2. Algorithm: GWSAT
3. Config: Exec: 1000, Restart: 10, Iteration: 1000, wp: 0.6

Analysis: It can be seen that the maximum number of problems can be solved with 0.4 - 0.5 secs

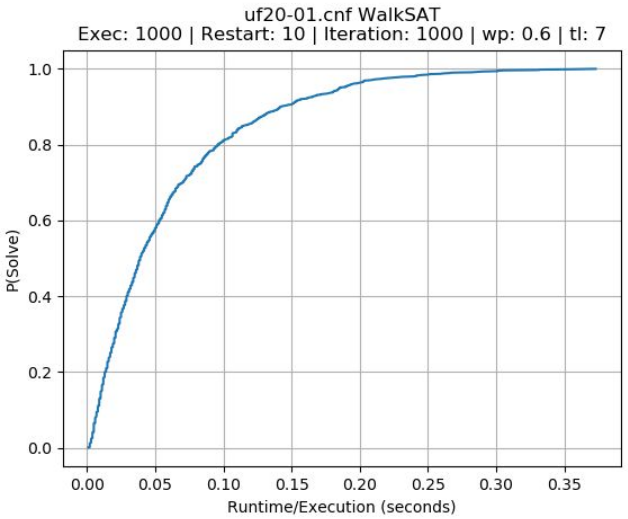
Uf20-02 - GWSAT



1. Instance : Uf20-02.cnf
2. Algorithm: GWSAT
3. Config: Exec: 1000, Restart: 10, Iteration: 1000, wp: 0.6

Analysis: It can be seen that the maximum number of problems can be solved with 0.12 secs

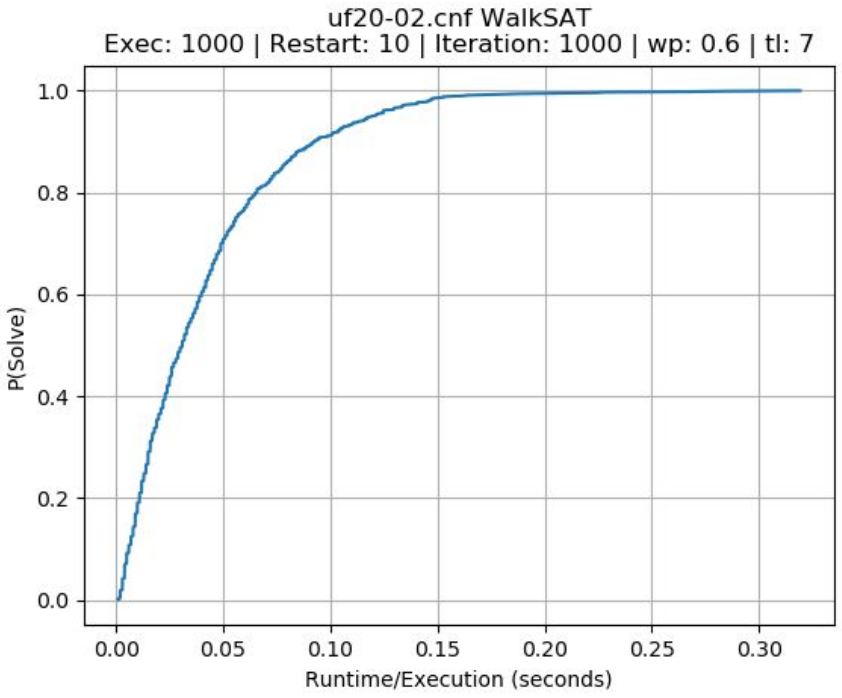
Uf20-01-WalkSAT



1. Instance: uf20-01
2. Algorithm: WalkSAT
3. Config: Exec: 1000, Restart: 10, Iteration: 1000, wp: 0.6, tl: 7

Analysis: It can be seen that the maximum number of problems can be solved with 0.25 secs

Uf20-02 WalkSAT



1. Instance: uf20-02
2. Algorithm: WalkSAT
3. Config: Exec: 1000, Restart: 10, Iteration: 1000, wp: 0.6, tl: 7

Analysis: It can be seen that the maximum number of problems can be solved within 0.15 secs

Major Observations:

1. All the algorithms solve maximum no of problem within 0.15 secs (approx)
2. It can be observed that first instance(uf20-01) takes more time in both algorithm then second instance (uf20-02) 15 secs for both algorithm.