Giovanni Minelli

SAT and CP-based approach to VLSI problem

Project report for Combinatorial Decision Making and Optimization Module 1

giovanni.minelli2@studio.unibo.it

1 SMT-based approach

This section show the resolution of the given problem by using a SMT solver. Satisfiability Modulo Theories (SMT) solvers takes systems in arbitrary format (first-order logic), while SAT solvers are limited to Boolean equations and variables, nevertheless they still mantain the speed and automation of today's Boolean engines.

1.1 Implementation

For the implementation of such part it has been used the Z3 Solver's Python API: **Z3Py**. Most of the script used in SAT was reused since the problem and the behaviour of i/o of the program didn't change.
This implementation make use, as before, of the approach reported in [1]. A different resolution approach could have made use of the knowledge acquired in the construction of the CP model but i preferred to face the problem with the suggestions of the paper used in the SAT part since the symmetries and other techniques to reduce the search space seemed very reliable despite the mediocre result obtained in the previous implementation.

1.1.1 Variables, Main constraints and objective function

The original paper, as described in the chapter before, covered only a SAT-based approach and therefore a bit of work of conversion was needed. Fortunately the expressive power of a SMT solver is higher then a SAT solver and so the implementation of the problem resulted in a much more cleaner piece of code. In particular making use of the optimization ability of the Z3 library, the previous work aimed to unify different decision problems executed with a different height is no more needed.

Partially similarly to the SAT implementation it has been used the boolean variables: *lr* and *ud*.
$lr_{i,j}$ is true if r_i are placed at the left to the r_j.
$ud_{i,j}$ is true if r_i are placed at the downward to the r_j.

To represent the position in the space of each circuit, differently from before, a normal encoding has been used resulting in the integer variables X and Y, with the domain properly constrained as follow:

$$0 \leq x_i \leq W - w_i$$
$$0 \leq y_i \leq H - h_i$$

Where W, H are the sizes of the plate and w_i, h_i the dimensions of circuit i.
Also the domain of H is constrained between *min_height* and *max_height* valorized respectively as the maximum height of the circuits in input and the sum of all the circuits heights.

To constrain the solution to a feasible displacement we have to avoid the overlapping, and so for each rectangle r_i, r_j (i > j), the following constraints are used:

$$lr_{i,j} \vee lr_{j,i} \vee ud_{i,j} \vee ud_{j,i}$$
$$\neg lr_{i,j} \vee x_i + w_i \leq x_j$$
$$\neg lr_{j,i} \vee x_j + w_j \leq x_i$$
$$\neg ud_{i,j} \vee y_i + h_i \leq y_j$$
$$\neg ud_{j,i} \vee y_j + h_j \leq y_i$$

Equivalently as before, just with the difference that the px and py variables have been substituted adapting the constraints to the new encoding with X and Y.
All other constraints become unnecessary and also the objective function can be easily expressed by telling the solver (optimizer) that the solution must minimize the value of H.

1.1.2 Symmetries

All techniques of search space reduction (except for the Large rectangles (**LR**) in vertical direction) as described before, has been implemented also for this approach. The difference from the previous part is just the definition of the constraints since the different spatial encoding.

1.1.3 Hypothetical model with rotation allowed

The encoding process similarly as before, can make use of the one hot encoding technique applied to a variable which indicate the rotation. Then the creation of additional clauses enable the other displacement, all tied to the truth value of the corresponding boolean variable which state the unique rotation for the circuit.

1.2 Results

To evaluate the results has been used the number of conflicts in the optimization process, the time of execution and the value of optimality in output. To be noted that in this case as value of time has been used the one of entire execution of the python script since it was just slightly greater than the real time measured for the execution of the optimizer, anyway the bound of 300s was maintained.
Confronting the results obtained with different combinations of symmetry breaking constraints it's possible to see that there is a decremental ordering by performance between models with no constraints at all, and models with all constraints enabled. This finding in the data collected, is valid both for the time and the number of conflict, but not for the optimality measure.
The best results are obtained with the model `SR+LRH` and `SR` which are able to solve 3 more instances in the time limit wrt the worst one. Anyway, even if both obtained similar results in most of the cases, for some instances in which both fail, the `SR` model report less conflicts.
Finally, as for the other approaches a comparison of execution with non ordered inputs have been made: the best model and the worst where tested. The results varied a lot for all the measure, in some cases the time and the conflicts decreased noticeably in one direction and sometimes increased. Also for what concern the optimality, the best model with unordered data solved three instances less than is counterpart, instead the worst model performed better with unordered data solving two more instances. Such result can be surely justified by a lucky occasional sequence and it's possible to conclude that both the SMT and SAT solver are randomly influenced by the presence of an ordering in the input data.

In conclusion, as it can be observed, the results obtained by the SMT solver are much better in comparison with the ones obtained by the SAT model. I can't exclude errors in my implementations however it's easy to notice how the encoding of the problem become easier when there isn't the boolean logic limitation.
To be noted also the gap of performance between the CP model and the other two.

	SR + LRH + LS (worst)			SR (best)			NO SYM OPT (ref)		
N°	TIME (s)	CONF	OPT	TIME (s)	CONF	OPT	TIME (s)	CONF	OPT
1	0,02	55	Y	0,02	33	Y	0,02	33	Y
2	0,03	29	Y	0,03	15	Y	0,03	15	Y
3	0,06	237	Y	0,05	250	Y	0,05	250	Y
4	0,07	364	Y	0,11	488	Y	0,09	488	Y
5	0,14	1263	Y	0,13	1130	Y	0,13	1130	Y
6	0,24	1779	Y	0,24	1879	Y	0,24	1879	Y
7	0,2	1415	Y	0,19	1309	Y	0,19	1309	Y
8	0,27	1101	Y	0,2	682	Y	0,2	682	Y
9	0,22	1080	Y	0,31	2456	Y	0,32	2456	Y
10	0,75	3736	Y	0,75	4016	Y	0,8	4016	Y
11	300,2	728922	N	300,21	582553	N	300,21	641937	N
12	2,35	10366	Y	1,31	5245	Y	2,64	12102	Y
13	1,99	18879	Y	3,03	29230	Y	1,76	18099	Y
14	7,8	41952	Y	4,17	30947	Y	12,14	67300	Y
15	6,33	18293	Y	4,51	13814	Y	2,34	6209	Y
16	300,35	161395	N	300,37	145302	N	300,31	725348	N
17	14,08	23609	Y	43,48	52332	Y	41,98	47220	Y
18	61,03	64376	Y	16,07	22683	Y	29,15	37307	Y
19	300,48	173884	N	300,47	123977	N	300,46	352363	N
20	300,43	515915	N	300,43	280142	N	300,44	217074	N
21	300,64	137136	N	300,44	373762	N	300,46	171396	N
22	300,57	230417	N	300,57	134636	N	300,62	170341	N
23	132,16	82176	Y	195,18	103645	Y	74,09	52302	Y
24	29,78	31116	Y	35,32	35121	Y	22,59	29740	Y
25	300,71	411406	N	300,75	90468	N	300,76	154342	N
26	300,69	141064	N	300,53	131410	N	300,53	139783	N
27	300,46	118982	N	122,17	71864	Y	166,58	84811	Y
28	300,5	119353	N	79,82	49680	Y	123,19	67280	Y
29	300,63	111471	N	294,88	106741	Y	300,53	112203	N
30	300,77	61330	N	300,85	61614	N	300,78	66306	N
31	28,68	36115	Y	25,82	28035	Y	37,85	40026	Y
32	300,86	47255	N	300,83	74715	N	300,84	60279	N
33	16,99	25312	Y	28,63	29703	Y	23,63	28850	Y
34	300,59	138649	N	300,57	147151	N	300,57	160877	N
35	300,71	136249	N	300,61	134957	N	300,7	121599	N
36	300,58	151275	N	300,6	172050	N	300,75	138121	N
37	300,69	110861	N	300,76	102092	N	300,87	105005	N
38	300,97	96099	N	300,79	95413	N	300,99	80875	N
39	300,97	124523	N	300,77	106013	N	300,82	129613	N
40	304,35	16514	N	304,19	18152	N	304,48	16381	N

Table 1: The complete tests results can be found in the SMT sources folder

References

[1] A SAT-based Method for Solving the Two-dimensional Strip Packing problem
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.163.7772&rep=rep1&type=pdf