CPT4: An Optimal Temporal Planner Lost in a Planning Competition without Optimal Temporal Track

Vincent Vidal
ONERA – DCSD
Toulouse, France
Vincent.Vidal@onera.fr

Overview

Once again, in the 7th International Planning Competition (IPC) of 2011, the optimal temporal track has been cancelled due to a lack of competitors: CPT4, the fourth version of the CPT planner (Vidal and Geffner 2006), was the only submitted planner. This already happened in 2008 for the 6th IPC where CPT3 was alone, while in 2006 for the 5th IPC CPT2 had been compared with CPT1 only and awarded with a "Distinguished Performance" in temporal domains of the optimal track (but well, was it useful?...). In that competition, SAT-based planners were the most efficient for parallel domains, a special case of temporal planning with the conservative semantics first introduced in (Smith and Weld 1999) where actions have uniform durations and overlapping of mutex actions is forbidden.

The only meaningful comparison with other temporal planners, during a planning competition, has been made in 2004 for the 4th IPC, where the first version of CPT had been ranked second in the optimal track (SATPLAN'04 being ranked first), generally being the more efficient in temporal domains against TP4 and HSP^*_a (Haslum and Geffner 2001; Haslum 2004).

We think that optimal temporal planning is a meaningful problem, and that more planners should be submitted to the International Planning Competition in order to try to characterize some "state-of-the-art" among implemented systems. One difficulty is perhaps to follow the full PDDL2.1 semantics, which makes temporal planning EXPSPACE-complete (Rintanen 2007). Actually, the submitted version of CPT4 does not follow PDDL2.1 semantics, but rather the conservative one (Smith and Weld 1999). We think that temporal planning with the conservative semantics is difficult enough, and more closely related to typical scheduling problems. From some preliminary experiments with CPT4 on classical open-shop and job-shop problems, we can say that CPT4 is a relatively good scheduler (i.e. it solves some instances generally considered as difficult).

We decided to enter CPT4 into the 7th International Planning Competition in all tracks, in order to evaluate how far an optimal temporal planner can be from specialized planners in each track. But to make things clear, even if CPT4 participates to this IPC, it should not: the only track for which it has been designed for, the optimal temporal track, has been cancelled...

Implementation

The actual implementation of CPT is based on the version written in C described in (Vidal and Tabary 2006). It includes some additional pruning rules that first appeared in (Vidal and Geffner 2005), as well as the last conflict based reasoning (Lecoutre et al. 2009). The improvements brought to CPT4 are mostly minor: a few bug fixes and slight improvements in the way the constraint propagation rules are written.

One notable improvement is the use of a conflict counting heuristic, inspired by the $wdeg$ heuristic (Boussemart et al. 2004). Each time a contradiction occurs in the constraint propagation engine, a weight attached to the variables and constraints in relation with the violated constraint is incremented. Branching rules are then tweaked to always follow branches that constrain more variables with the highest weights.

Another improvement is a slightly better way of producing optimal sequential plans. As in CPT3, the cost associated to an action (1 for optimal length, any positive value for cost-based planning) is treated as a duration. But instead of forcing all actions to be pairwise mutex, an additional constraint is enforced: an action is excluded from any subsequent partial plan during search, when the sum of the costs of the actions that belong to the current partial plan plus the cost of the considered action exceeds the current bound on the makespan (interpreted as length or cost sum). This means that CPT4 still explores the space of partially ordered plans with concurrent actions, with a constraint that guarantees optimality.

Some Experiments

CPT has been designed with two objectives in mind: to match the performance of the best parallel planners, and to be a powerful planner for optimal temporal planning. While the former can be easily checked by a comparison with SAT-based planners, the latter is a bit more problematic due to the lack of recent efficient optimal temporal planners: we only compared CPT4 with its previous version, CPT3. All experiments are performed on an Intel Xeon X5670 running at 2.93GHz with 4GB of memory and a timeout of 30 minutes.

IPC	domain	#pbs	#solved					
			CPT3	CPT4	MAXPLAN	Mp	SASE	SATPLAN
1	grid	5	1 (2)	1 (2)	1 (2)	2 (1)	**3**	2 (1)
	gripper	20	4 (1)	4 (1)	3 (2)	3 (2)	**5**	3 (2)
	logistics	35	26 (1)	25 (2)	11 (16)	18 (9)	**27**	23 (4)
	mprime	35	19 (14)	24 (9)	28 (5)	**33**	**33**	25 (8)
	mystery	30	**29**	28 (1)	18 (11)	19 (10)	21 (8)	19 (10)
	total	125	79 (10)	82 (7)	61 (28)	75 (14)	**89**	72 (17)
	% solved		63.2%	65.6%	48.8%	60.0%	**71.2%**	57.6%
2	blocks	60	37 (1)	**38**	15 (23)	34 (4)	31 (7)	35 (3)
	elevator	150	43 (9)	44 (8)	20 (32)	33 (19)	46 (6)	**52**
	freecell	60	12 (19)	12 (19)	16 (15)	14 (17)	**31**	19 (12)
	logistics	198	38 (7)	42 (3)	29 (16)	27 (18)	44 (1)	**45**
	total	468	130 (22)	136 (16)	80 (72)	108 (44)	**152**	151 (1)
	% solved		27.8%	29.1%	17.1%	23.1%	**32.5%**	32.3%
3	depots	22	15 (1)	15 (1)	12 (4)	15 (1)	15 (1)	**16**
	driverlog	20	15 (2)	15 (2)	12 (5)	15 (2)	**17**	16 (1)
	freecell	20	3 (3)	3 (3)	5 (1)	4 (2)	**6**	4 (2)
	rovers	20	13 (5)	13 (5)	15 (3)	**18**	16 (2)	15 (3)
	zenotravel	20	15 (1)	**16**	12 (4)	14 (2)	**16**	15 (1)
	total	102	61 (9)	62 (8)	56 (14)	66 (4)	**70**	66 (4)
	% solved		59.8%	60.8%	54.9%	64.7%	**68.6%**	64.7%
4	pipesworld-notankage	50	16 (24)	16 (24)	26 (14)	19 (21)	**40**	37 (3)
	pipesworld-tankage	50	8 (18)	8 (18)	10 (16)	10 (16)	**26**	16 (10)
	psr-small	50	49 (1)	49 (1)	**50**	49 (1)	**50**	**50**
	total	150	73 (43)	73 (43)	86 (30)	78 (38)	**116**	103 (13)
	% solved		48.7%	48.7%	57.3%	52.0%	**77.3%**	68.7%
5	openstacks	30	0 (5)	0 (5)	0 (5)	0 (5)	**5**	**5**
	pathways	30	5 (4)	8 (1)	**9**	7 (2)	5 (4)	**9**
	tpp	30	17 (6)	**23**	20 (3)	14 (9)	16 (7)	20 (3)
	trucks	30	2 (8)	2 (8)	3 (7)	3 (7)	**10**	5 (5)
	total	120	24 (15)	33 (6)	32 (7)	24 (15)	36 (3)	**39**
	% solved		20.0%	27.5%	26.7%	20.0%	30.0%	**32.5%**
6	elevators	30	5 (9)	5 (9)	5 (9)	4 (10)	**14**	10 (4)
	openstacks	30	**3**	**3**	**3**	**3**	**3**	**3**
	parcprinter	30	27 (3)	28 (2)	28 (2)	25 (5)	**30**	29 (1)
	pegsol	30	10 (14)	10 (14)	11 (13)	18 (6)	**24**	19 (5)
	scanalyzer	30	14 (4)	15 (3)	12 (6)	17 (1)	**18**	14 (4)
	transport	30	9 (4)	9 (4)	6 (7)	6 (7)	**13**	11 (2)
	woodworking	30	**30**	**30**	**30**	**30**	5 (25)	**30**
	total	210	98 (18)	100 (16)	95 (21)	103 (13)	107 (9)	**116**
	% solved		46.7%	47.6%	45.2%	49.0%	51.0%	**55.2%**
total		1175	465 (105)	486 (84)	410 (160)	454 (116)	**570**	547 (23)
% solved			39.6%	41.4%	34.9%	38.6%	**48.5%**	46.6%

Table 1: Number and percentage of solved problems in selected parallel domains of the IPCs from 1998 to 2008. Numbers in bold indicate the best results and numbers in parenthesis indicate the number of unsolved problems with respect to the best result.

Parallel Planning

The first version of CPT had been ranked second in the 4$^{\text{th}}$, after SATPLAN'04; since then, SAT-based planners have been greatly improved. The first reason is that SAT solvers are more and more efficient; and the second, because better encodings have been found. We compare six state-of-the-art planners on parallel planning problems: MAXPLAN (Xing, Chen, and Zhang 2006), Mp (Rintanen 2010), SASE (Huang, Chen, and Zhang 2010), SATPLAN'06 (Kautz, Selman, and Hoffmann 2006), and the last two versions of CPT. We took many domains from the 1$^{\text{st}}$ to the 6$^{\text{th}}$ IPC, for a total of 1175 planning problems. The domains that do not appear in these results (e.g. airport and sokoban) make the available version of SASE (v0.1) crash: to be fair, we have not included them. All planners optimize the parallel plan length.

As can be seen from Table 1, the recent SASE planner, which is based on a SAS+ encoding, is the most efficient one: it solves 570 problems (48.5%). SATPLAN'06 is the next best planner, with 547 problems solved (46.6%). Then come CPT4 and CPT3, which solve 486 problems (41.4%) and 465 problems (39.6%) respectively. Finally, comes Mp with 454 solved problems (38.6%) and MAXPLAN with 410 problems (34.9%). Although CPT4 is not the most efficient parallel planner, it is able to outperform some SAT-based planners on this set of benchmarks. However, the difference between CPT and the best SAT-based planner is perhaps higher than what it was a few days ago: CPT has not evolved a lot over the last few years.

Figure 1 shows the cumulated number of solved problems in function of the total running time. For each CPU time t on the x axis, the corresponding value on the y axis gives the number of problems solved in under t seconds. We can

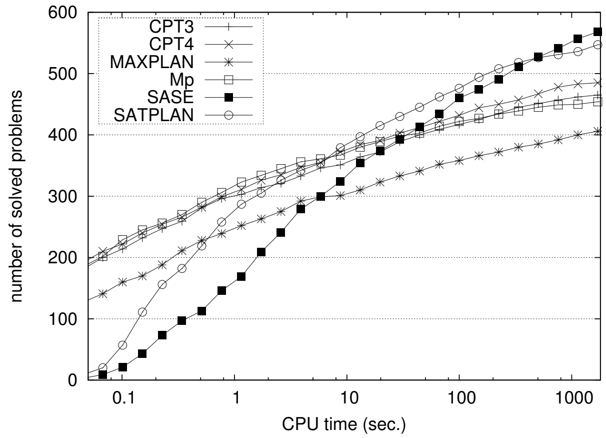

Figure 1: Cumulated number of solved problems in function of the search time for parallel planners.

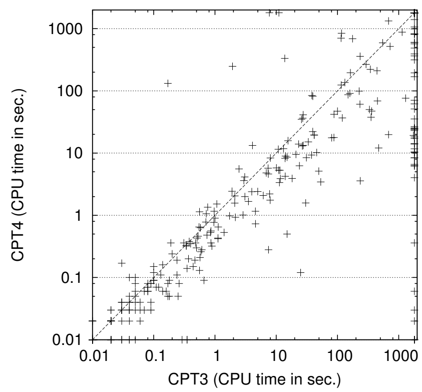

Figure 3: Comparison of the total running time between CPT3 and CPT4 on temporal planning problems.

see that CPT3, CPT4 and Mp solve more problems than the other planners with very small running times, but solve significantly fewer problems that require longer running time. Figure 2, which makes pairwise comparisons between CPT4 and the other planners, confirms this view.

Temporal Planning

CPT3 and CPT4 are then compared on all temporal problems without numerical fluents from all past IPCs, for a total of 664 planning problems. Table 2 shows the number of solved problems within the time limit. CPT4, which solves 316 problems (47.6%), clearly outperforms CPT3 which solves 271 problems (40.8%). Only one problem in the domain pipesworld-tankage is solved by CPT3 and not by CPT4. Figure 3, which compares the total running time of both planners, shows that CPT4 generally outperforms CPT3.

IPC	domain	#pbs	#solved CPT3	#solved CPT4
3	depots	22	8 (2)	**10**
	driverlog	20	**11**	**11**
	rovers	20	5 (1)	**6**
	satellite	20	12 (1)	**13**
	zenotravel	20	**14**	**14**
	total	102	50 (4)	**54**
	% solved		49.0%	**52.9%**
4	airport	50	41 (3)	**44**
	airport-timewindows	50	34 (10)	**44**
	pipesworld-notankage-deadlines	30	14 (4)	**18**
	pipesworld-notankage	50	15 (2)	**17**
	pipesworld-tankage	50	**9**	8 (1)
	satellite-time	36	**17**	**17**
	satellite-time-timewindows	36	7 (5)	**12**
	total	302	137 (23)	**160**
	% solved		45.4%	**53.0%**
5	openstacks	20	**0**	**0**
	storage	30	**15**	**15**
	trucks	30	2 (5)	**7**
	total	80	17 (5)	**22**
	% solved		21.2%	**27.5%**
6	crewplanning	30	5 (10)	**15**
	elevators	30	**2**	**2**
	openstacks	30	**4**	**4**
	parcprinter	30	22 (3)	**25**
	pegsol	30	**29**	**29**
	sokoban	30	**5**	**5**
	total	180	67 (13)	**80**
	% solved		37.2%	**44.4%**
total		664	271 (45)	**316**
% solved			40.8%	**47.6%**

Table 2: Number and percentage of solved problems in all temporal domains of the IPCs from 1998 to 2008. Numbers in bold indicate the best results and numbers in parenthesis indicate the number of unsolved problems with respect to the best result.

Acknowledgments

This work has been supported by the French National Research Agency (ANR) through COSINUS program (project DESCARWIN n°ANR-09-COSI-002).

References

Boussemart, F.; Hemery, F.; Lecoutre, C.; and Sais, L. 2004. Boosting systematic search by weighting constraints. In *Proc. ECAI*, 146–150.

Haslum, P., and Geffner, H. 2001. Heuristic planning with time and resources. In *Proc. ECP*, 121–132.

Haslum, P. 2004. TP4 '04 and HSP$^*_a$. In *Booklet of the* 4^{th} *IPC*.

Huang, R.; Chen, Y.; and Zhang, W. 2010. A novel transition based encoding scheme for planning as satisfiability. In *Proc. AAAI*.

Kautz, H.; Selman, B.; and Hoffmann, J. 2006. SATPLAN: Planning as satisfiability. In *Booklet of the* 5^{th} *IPC*.

Lecoutre, C.; Saïs, L.; Tabary, S.; and Vidal, V. 2009. Reasoning from last conflict(s) in Constraint Programming. *Artificial Intelligence* 173(18):1592–1614.

Rintanen, J. 2007. Complexity of concurrent temporal planning. In *Proc. ICAPS*, 280–287.

Rintanen, J. 2010. Heuristic planning with SAT: Beyond

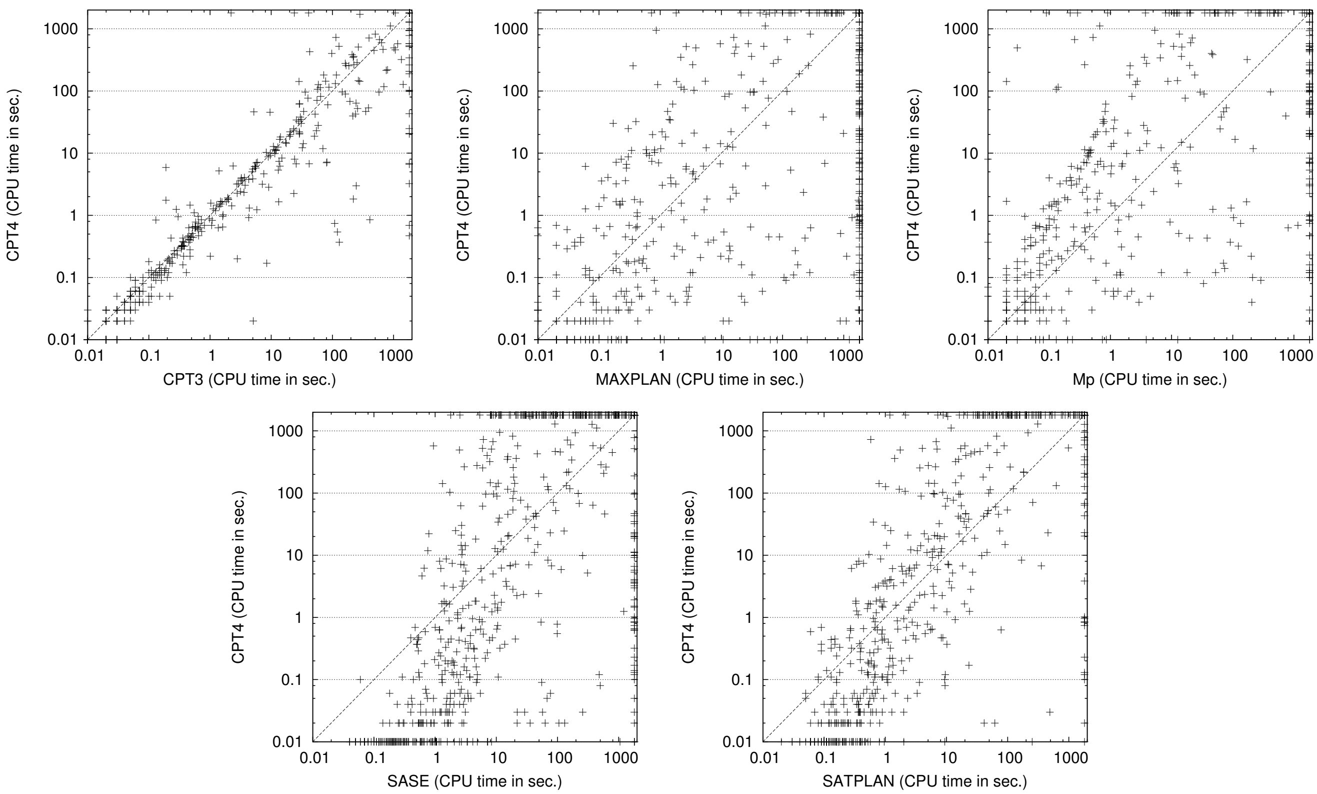

Figure 2: Comparison of the total running time between CPT4 and all other parallel planners.

uninformed depth-first search. In *Proc. Australasian Conf. on AI*, 415–424.

Smith, D. E., and Weld, D. S. 1999. Temporal planning with mutual exclusion reasoning. In *Proc. IJCAI*, 326–337.

Vidal, V., and Geffner, H. 2005. Solving simple planning problems with more inference and no search. In *Proc. CP*, 682–696.

Vidal, V., and Geffner, H. 2006. Branching and pruning: An optimal temporal POCL planner based on constraint programming. *Artificial Intelligence* 170(3):298–335.

Vidal, V., and Tabary, S. 2006. The new version of CPT, an optimal temporal POCL planner based on constraint programming. In *Booklet of the 5th IPC*.

Xing, Z.; Chen, Y.; and Zhang, W. 2006. Optimal strips planning by maximum satisfiability and accumulative learning. In *Proc. ICAPS*, 442–446.