Machine Learning for Constraint Solving

Alejandro Arbelaez, Youssef Hamadi, Michèle Sebag

TAO, Univ. Paris-Sud

Dagstuhl – May 17th, 2011

Position of the problem

Algorithms, the vision

- Software editor vision : A GO button
- Researcher vision : more is better

  add other functionalities (with control parameters)
- Community vision : different is interesting

  devise new algorithms (with control parameters)

Crossing the chasm : software life beyond research labs

- Automatically adjust algorithm parameters depending on current problem
- Select best (expected) algorithm depending on current problem

Meta-Learning

Meta-learning

for ML

- Select automatically best ML algorithm
  Bradzil, Bensoussan, Giraud-Carrier, Kalousis, Kietz, Maloberti...
- Main difficulty : devise problem descriptors

for Evolutionary Computation

- Adjust on-line operator rates
  Thierens et al 07, 08 ; Fialho et al. 08, 09, 10
- Main difficulty : devise operator "reward" ; adjust operator rate depending on its reward (Exploration vs Exploitation).

for Constraint Solving

Rice 1976

- Context : Microsoft / INRIA / CNRS
- Give the user the best performance she can get

Overview

Formal background

Notations

- Variables $X_1 \ldots X_n$
- X_i belongs to domain D_i
- Constraints $C_1 \ldots C_m$
  - in closed form : $X_i + X_j = X_k$; alldifferent$(X_1, X_2, ..)$
  - in extension : $C(X_1, X_2)$ holds for $\{(1,2),(2,1),\ldots\}$.

Example : Sudoku

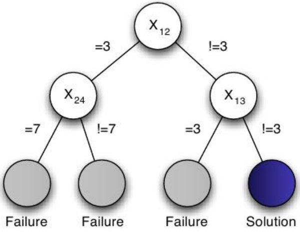
	1	2	3	4	5	6	7	8	9
1	8					1		4	
2	2		6	8				1	
3			9			6		8	
4	1	2	4						9
5									
6	9						8	2	4
7		5		4			1		
8		8			7		2		5
9		9		5					7

Variables	$X_{11} \ldots X_{99}$
Domains	$D_i = \{1 \ldots 9\}$
Constraints	All different $(X_{1,i} \ldots X_{9,i}$

Formal background, 2

Types of problem

- Constraint Solving — complete and incomplete search
- Constraint Satisfaction — boolean domains
- Constraint Optimization — min. # constraints insatisfied
- Quantified Boolean Formulae — Benedetti, Mangassarian 08



select variable — select value — restart

Some heuristics

Value selection — min, max, mid, random

Variable selection

- mindom — First Fail Principle
- domdeg
- weighted degree and wdomdeg — Boussemart et al. 04
- score(variable) — Selman Levêque Mitchell 92
- Novelty — McAllester Selman Kautz 97
- Novelty+ : wp Novelty + $(1 - wp)$ WalkSat — Hoos 99
- Novelty+p : with anticipation — Li Wei Zhang 07
- Adaptive Novelty+ : tune wp depending on history — Hoos 02
- Scaling and Probability Smoothing (weighting clauses) — Hutter Tompkins Hoos 02, Li Wei Zhang 07

Some other heuristics

Restart schedule

$(t_1, H_1, t_2, H_2, \ldots)$

• spend t_1 with H_1 (cutoff time)

• then increase cutoff time (often $t_{i+1} = c \times t_i$), and use another heuristics.

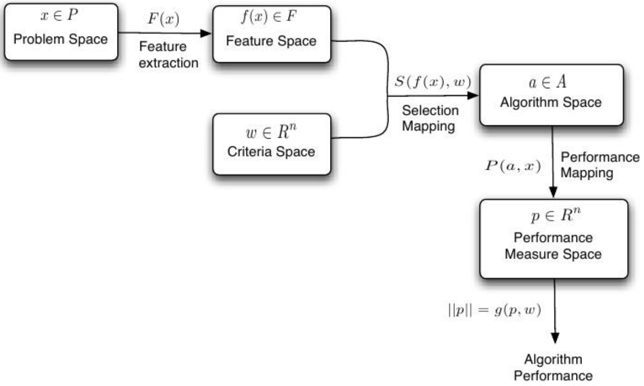
Note that using H_i might modify the choices for H_{i+1}.

Other

- Use a taboo list in incomplete search

Meta-Search

After Rice 1976



Importance of pbs descriptors

Meta-Search Approaches

Characterize runtime

short vs long — Horvitz et al. 01

predict runtime (Empirical Hardness Model)

Nudelman et al. 02, 04, 09

predict time-to-solution — Haim & Walsh 08, 09

SATzilla : given problem instances

- training : collect (instance description, solver performance)
- testing : identify candidate solvers ; run all of them for short time ; run the expected best one.
- Extensions : mixture of experts — Xu et al. 07
  other ML approaches — Devlin and O'Sullivan 08

Meta-Search Approaches, 2

CP Hydra — O'Mahony et al. 08

- portfolio + case-based reasoning
- build an archive of cases (problem instances)
- for a new instance, find k-nearest neighbor cases
- build a switching policy, running and stopping black-box solvers

Description

- Syntactic features (XSCP specifications)
- Semantic features : use a preliminary testing phase (2s) and collect general search statistics
- Pragmatic features — Beck Freuder 04
  (Time I, algo A) $= 1$ iff A yields best solution at I.
  (learning curve slope for each algo).

Overview

Goal

No Free Lunch

There is no such thing as a Universal Best CP strategy

The need

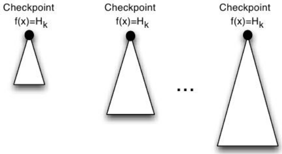
- Each user has a specific distribution of problem instances
- There exists a best strategy for this problem distribution
- This distribution is not known in advance ; it is prone to evolve

The opportunity

- The computer is idle most of its time
- Idle time can be used to self-play and learn the best strategy.

Continuous Search for CP

General setting



Framework : Default

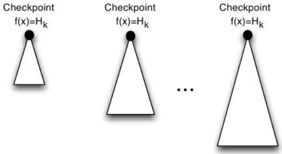
f = heuristic model

1. Checkpoint i : compute problem description x, find the best heuristics $H_i = f(x)$
2. Apply H_i until cutoff time t_i
3. Goto 1 (x has changed !)

(personal comment : more a reinforcement learning algorithm...)

Lagoudakis and Littman 2001

General setting



Framework : Default

f = heuristic model

1. Checkpoint i : compute problem description x, find the best heuristics $H_i = f(x)$
2. Apply H_i until cutoff time t_i
3. Goto 1 (x has changed !)

(personal comment : more a reinforcement learning algorithm...)

Lagoudakis and Littman 2001

Continuous Search for CP, 2

Framework

- Production/Exploitation mode : use Default — use f
- Learning/Exploration mode during idle time — revise f
  - try variants (use $H <> f(x)$)
  - collect examples $((x, 1_H), y$
  - relearn f

Representation : 95 features in toto

Static features

Problem Size Features:
1. **Number of clauses**: denoted c
2. **Number of variables:** denoted v
3. **Ratio:** c/v

Variable-Clause Graph Features:
4-8. **Variable nodes degree statistics:** mean, variation coefficient, min, max and entropy.
9-13. **Clause nodes degree statistics:** mean, variation coefficient, min, max and entropy.

Variable Graph Features:
14-17. **Nodes degree statistics:** mean, variation coefficient, min and max.

Balance Features:
18-20. **Ratio of positive and negative literals in each clause:** mean, variation coefficient and entropy.
21-25. **Ratio of positive and negative occurrences of each variable:** mean, variation coefficient, min, max and entropy.
26-27. **Fraction of binary and ternary clauses**

Proximity to Horn Formula:
28. **Fraction of Horn clauses**
29-33. **Number of occurrences in a Horn clause for each variable:** mean, variation coefficient, min, max and entropy.

DPLL Probing Features:
34-38. **Number of unit propagations**: computed at depths 1, 4, 16, 64 and 256.
39-40. **Search space size estimate:** mean depth to contradiction, estimate of the log of number of nodes.

Local Search Probing Features:
41-44. **Number of steps to the best local minimum in a run:** mean, median, 10th and 90th percentiles for SAPS.
45. **Average improvement to best in a run:** mean improvement per step to best solution for SAPS.
46-47. **Fraction of improvement due to first local minimum:** mean for SAPS and GSAT.
48. **Coefficient of variation of the number of unsatisfied clauses in each local minimum:** mean over all runs for SAPS.

Figure: Features from [Xu, Hutter, Hoos, Leyton-Brown, CP 2007].

Representation : 95 features in toto

Static features

Problem definition : density, tightness, ...

Variable size and degree (min, max, average, variance)

Constraint degree and cost category (exp, cubic, quadratic, lin. cheap, lin. expensive)

Dynamic features

Heuristic criteria(variable) : wdeg, domdeg, impact : min, max, average over all var 15

Constraint weight (wdeg) : min, max, average 12

Constraint filtering : min, max, average of number of times called by propagation 3

Checkpoint information : number of nodes, max depth, number of assigned var/sat constraints for the last non-failed node, wdeg and impact of non-assigned var 33

Everything normalized in $[-1, 1]$

Gathering examples and improving

During idle time consider the problem instance last solved, generate new trials

1. each trial : everything as in production mode except for i-th checkpoint
2. at i-th checkpoint, try the second best heuristics after the current h.model f. (x, h)
3. see if it improves on default y
4. Finally $\mathcal{E} = \bigcup\{((x, h), y)\}$
5. Learn f from $\mathcal{E}$ Gaussian SVM

Difficulty

- Hugely imbalanced problem ;
- $y = 1$ if $time(trial) \sim time(\text{default})$.

Overview

1. State of the art
2. Goal
3. The algorithm : lifelong learning
4. Results
5. Discussion & perspectives

Problems

496 CSP from MiniZinc and XCZP repositories

1	Nurse scheduling	100
2	Balance Incomplete Block Design	83
3	Job shop scheduling	130
4	Geometric	100
5	Langford numbers	83

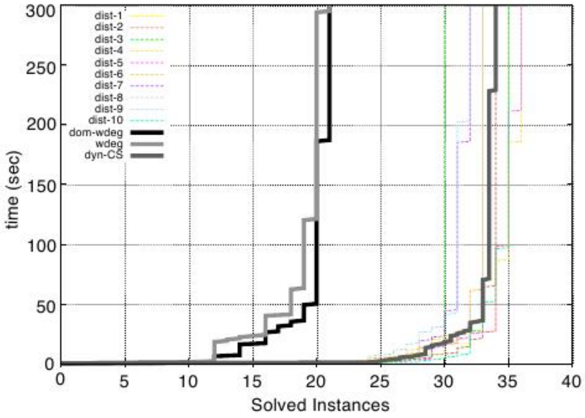
Experimental setting

- Gecode 2.1.1
- cutoff 1,000 (increase $\times$ 1.5)
- Heuristics : mindom, domdeg, wdeg, dom-wdeg, impacts
- Value selection : min-dom

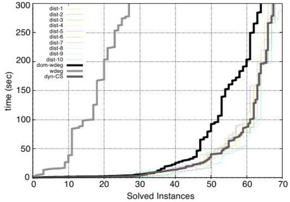
Perf : average on 10 random orderings of CS instances.

Results

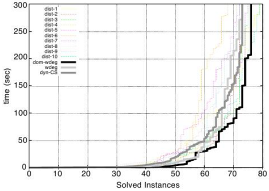
Langford numbers



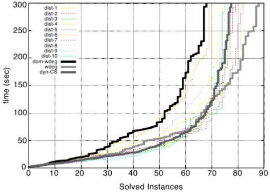
More results



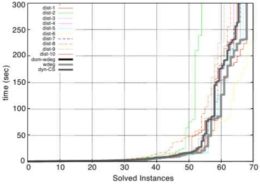
Geom



Job shop



nurse



bibd

More results

Time out 5 min

Problem	*dom-wdeg*			*wdeg*			*dyn-CS*		
	#sol	time(h)	avg-time(m)	#sol	time(h)	avg-time(m)	#sol	time(h)	avg-time(m)
nsp	68	3.9	2.34	**88**	**2.6**	**1.56**	77	2.9	1.74
bibd	**68**	**1.8**	**1.37**	68	1.8	1.37	65	2.0	1.44
js	**76**	**4.9**	**2.26**	73	5.1	2.35	73	5.2	2.4
lfn	21	5.2	3.75	21	5.3	3.83	**33**	**4.1**	**2.96**
geom	64	3.9	2.34	27	6.8	4.08	**67**	**3.3**	**1.98**
Total	297	19.7	2.39	274	21.6	2.61	**315**	**17.5**	**2.11**

Time out 3 min

Problem	*dom-wdeg*			*wdeg*			*dyn-CS*		
	#sol	time(h)	avg-time(m)	#sol	time(h)	avg-time(m)	#sol	time(h)	avg-time(m)
nsp	61	2.8	1.68	**81**	**2.1**	**1.26**	75	2.2	1.32
bibd	**62**	**1.3**	**0.94**	62	1.3	0.94	60	1.4	1.01
js	**74**	**3.1**	**1.43**	69	3.3	1.52	67	3.4	1.57
lfn	20	3.2	2.31	20	3.2	2.31	**32**	**2.5**	**1.81**
geom	56	2.6	1.56	20	4.3	2.58	**63**	**2.2**	**1.32**
Total	273	13.0	1.57	252	14.2	1.72	**297**	**11.7**	**1.42**

Comments

Benefits of online learning

- Can quickly converge to best heuristics — 3 out of 5
- Can switch to hybrid strategy — 2 out of 5

Meta-search accuracy vs model accuracy

Timeout	bibd	nsp	geom	js	lfn	Total
3 Min	64.5%	64.2%	79.2%	65.6%	68.2%	68.3%
5 Min	63.2%	58.8%	76.9%	63.6%	73.8%	67.3%
Average	63.9%	61.5%	78.0%	64.6%	71.0%	**67.8%**

Discussion

Most approaches are offline

- SATzilla — Xu, Hutter, Hoos, Leyton-Brown 08
- CPHydra — O'Mahony et al. 08
- self-AQME — Pulina Tacchella 09

Issues

- Data needed : quantity / representativity
- Passive vs Active learning
- Description : static, dynamic, pragmatic
- Reinforcement learning...

Discussion

Most approaches are offline

- SATzilla — Xu, Hutter, Hoos, Leyton-Brown 08
- CPHydra — O'Mahony et al. 08
- self-AQME — Pulina Tacchella 09

Issues

- Data needed : quantity / representativity
- Passive vs Active learning
- Description : static, dynamic, pragmatic
- Reinforcement learning...