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

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

Rice 1976
Overview

1. State of the art
2. Goal
3. The algorithm: lifelong learning
4. Results
5. Discussion & perspectives
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($X_1, X_2, \ldots$)
  - in extension: $C(X_1, X_2)$ holds for \{(1, 2), (2, 1), \ldots\}.

Example: Sudoku

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>6</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>9</td>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>M4</td>
<td>M5</td>
<td>M6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>M4</td>
<td>M5</td>
<td>M6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>4</td>
<td></td>
<td>1</td>
<td></td>
<td>2</td>
<td>M9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>M4</td>
<td>M5</td>
<td>M6</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Variables

Domains $D_i = \{1 \ldots 9\}$

Constraints All different ($X_{1,i} \ldots X_{9,i}$)
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
- domdeg
- weighted degree and wdomdeg
- score(variable)
- Novelty
- Novelty+: $wp$ Novelty + $(1 - wp)$ WalkSat
- Novelty+p: with anticipation
- Adaptive Novelty+: tune $wp$ depending on history
- Scaling and Probability Smoothing (weighting clauses)

First Fail Principle
Boussemart et al. 04
Selman Levêque Mitchell 92
McAllester Selman Kautz 97
Hoos 99
Li Wei Zhang 07
Hoos 02
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
After Rice 1976

Importance of pbs descriptors
Meta-Search Approaches

Characterize runtime
short vs long
predict runtime (Empirical Hardness Model)

predict time-to-solution

Horvitz et al. 01
Nudelman et al. 02, 04, 09
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
Devlin and O’Sullivan 08

other ML approaches
CP Hydra  

- 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

\[(\text{Time } I, \text{ algo } A) = 1 \text{ iff } A \text{ yields best solution at } I.\]
\[(\text{learning curve slope for each algo}).\]
Overview

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

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

General setting

Framework: Default

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
Framework

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

use $f$

revise $f$
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 $\text{time(trial)} \sim \text{time(default)}$. 
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
2. Balance Incomplete Block Design
3. Job shop scheduling
4. Geometric
5. Langford numbers

Experimental setting

- Gecode 2.1.1
- Cutoff 1,000 (increase × 1.5)
- Heuristics: mindom, domdeg, wdeg, dom-wdeg, impacts
- Value selection: min-dom

Perf: average on 10 random orderings of CS instances.
Langford numbers
More results

Geom

Job shop

nurse

bibd
## More results

### Time out 5 min

<table>
<thead>
<tr>
<th>Problem</th>
<th>dom-wdeg</th>
<th>wdeg</th>
<th>dyn-CS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#sol</td>
<td>time(h)</td>
<td>avg-time(m)</td>
</tr>
<tr>
<td>nsp</td>
<td>68</td>
<td>3.9</td>
<td>2.34</td>
</tr>
<tr>
<td>bibd</td>
<td>68</td>
<td>1.8</td>
<td>1.37</td>
</tr>
<tr>
<td>js</td>
<td>76</td>
<td>4.9</td>
<td>2.26</td>
</tr>
<tr>
<td>lfn</td>
<td>21</td>
<td>5.2</td>
<td>3.75</td>
</tr>
<tr>
<td>geom</td>
<td>64</td>
<td>3.9</td>
<td>2.34</td>
</tr>
<tr>
<td>Total</td>
<td>297</td>
<td>19.7</td>
<td>2.39</td>
</tr>
</tbody>
</table>

### Time out 3 min

<table>
<thead>
<tr>
<th>Problem</th>
<th>dom-wdeg</th>
<th>wdeg</th>
<th>dyn-CS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#sol</td>
<td>time(h)</td>
<td>avg-time(m)</td>
</tr>
<tr>
<td>nsp</td>
<td>61</td>
<td>2.8</td>
<td>1.68</td>
</tr>
<tr>
<td>bibd</td>
<td>62</td>
<td>1.3</td>
<td>0.94</td>
</tr>
<tr>
<td>js</td>
<td>74</td>
<td>3.1</td>
<td>1.43</td>
</tr>
<tr>
<td>lfn</td>
<td>20</td>
<td>3.2</td>
<td>2.31</td>
</tr>
<tr>
<td>geom</td>
<td>56</td>
<td>2.6</td>
<td>1.56</td>
</tr>
<tr>
<td>Total</td>
<td>273</td>
<td>13.0</td>
<td>1.57</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Timeout</th>
<th>bibd</th>
<th>nsp</th>
<th>geom</th>
<th>js</th>
<th>lfn</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 Min</td>
<td>64.5%</td>
<td>64.2%</td>
<td>79.2%</td>
<td>65.6%</td>
<td>68.2%</td>
<td>68.3%</td>
</tr>
<tr>
<td>5 Min</td>
<td>63.2%</td>
<td>58.8%</td>
<td>76.9%</td>
<td>63.6%</td>
<td>73.8%</td>
<td>67.3%</td>
</tr>
<tr>
<td>Average</td>
<td>63.9%</td>
<td>61.5%</td>
<td>78.0%</td>
<td>64.6%</td>
<td>71.0%</td>
<td>67.8%</td>
</tr>
</tbody>
</table>
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
- CPHydra
- self-AQME

Xu, Hutter, Hoos, Leyton-Brown 08
O’Mahony et al. 08
Pulina Tacchella 09

Issues

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