Effect of Initial Assignment on Local Search Performance for Max Sat

Daniel Berend and Yochai Twitto

Ben-Gurion University of the Negev

June 2020

Outline

1. Introduction
   - The problem: Max r-Sat
   - The local search: CCLS
   - The initialization: MOCE
2. Correlation
   - Experimental settings
   - End-to-end correlation
   - Ongoing correlation
3. Improving CCLS
4. Conclusion

Max r-Sat

- In the Max r-Sat problem, we are given a **sequence with repetitions** of clauses over some boolean variables.
- Each clause is a disjunction of **exactly** r literals over **distinct** variables.

Example of instance

$$(v_1 \vee v_2 \vee v_3) \wedge (v_1 \vee \neg v_2 \vee \neg v_3) \wedge (v_1 \vee v_2 \vee v_3) \wedge (\neg v_1 \vee v_2 \vee v_3)$$

Max r-Sat

- In the Max r-Sat problem, we are given a **sequence with repetitions** of clauses over some boolean variables.
- Each clause is a disjunction of **exactly** r literals over **distinct** variables.

Example of instance

$$(v_1 \vee v_2 \vee v_3) \wedge (v_1 \vee \neg v_2 \vee \neg v_3) \wedge (v_1 \vee v_2 \vee v_3) \wedge (\neg v_1 \vee v_2 \vee v_3)$$

- We seek a truth (`true`/`false`) assignment for the variables, maximizing the number of satisfied (made `true`) clauses.

Max r-Sat

- n variables.
- m clauses.
- $\alpha = m/n$, and assume $\alpha > 0$ is a constant.

Local Search

- Local search heuristics **explore** the assignment space.
- They usually **start** from a randomly generated assignment.
- They **traverse** the search space by flipping variables.

Configuration Checking Local Search

- The leading solver **CCLS** follows the local search scheme.
- It flips variables until some predefined number of flips is executed or the allotted time has been used up.

Configuration Checking Local Search

- The leading solver **CCLS** follows the local search scheme.
- It flips variables until some predefined number of flips is executed or the allotted time has been used up.
- CCLS performs two types of flips.
  - **Random** flips, with some predefined probability p.
  - **Greedy** flips, with probability $1 - p$.

Random Flips

- Flip a randomly selected variable from a randomly selected **unsatisfied** clause.

Greedy Flips

- Flip the seemingly best possible variable.

Greedy Flips

- Flip the seemingly best possible variable.
- Among variables that have **two properties**:
  - Their configuration has been changed.
  - They satisfy at least one currently unsatisfied clause.

Greedy Flips

- Flip the seemingly best possible variable.
- Among variables that have **two properties**:
  - Their configuration has been changed.
  - They satisfy at least one currently unsatisfied clause.
- This variable is the one whose flipping will lead to the **maximum** number of satisfied clauses.
  - Ties are broken randomly.

The Method of Conditional Expectations

- **MOCE** iteratively constructs an assignment by going over the variables in some (random) order.

The Method of Conditional Expectations

- **MOCE** iteratively constructs an assignment by going over the variables in some (random) order.
- At each iteration, it sets the **seemingly better truth value** to the currently considered variable.

The Method of Conditional Expectations

- **MOCE** iteratively constructs an assignment by going over the variables in some (random) order.
- At each iteration, it sets the **seemingly better truth value** to the currently considered variable.
- This is done by comparing the **expected number of satisfied clauses** under each of the two possible truth values it may set to the current variable.

The Method of Conditional Expectations

- For a given truth value, the expected number of satisfied clauses is the **sum of three quantities**.

The Method of Conditional Expectations

- For a given truth value, the expected number of satisfied clauses is the **sum of three quantities**.
- The **first** is the number of clauses already satisfied by the values assigned to the previously considered variables.

The Method of Conditional Expectations

- For a given truth value, the expected number of satisfied clauses is the **sum of three quantities**.
- The **first** is the number of clauses already satisfied by the values assigned to the previously considered variables.
- The **second** is the additional number of clauses satisfied by the assignment of the given truth value to the current variable.

The Method of Conditional Expectations

- For a given truth value, the expected number of satisfied clauses is the **sum of three quantities**.
- The **first** is the number of clauses already satisfied by the values assigned to the previously considered variables.
- The **second** is the additional number of clauses satisfied by the assignment of the given truth value to the current variable.
- The **third** is the expected number of clauses that will be satisfied by a random assignment to all currently unassigned variables.

The Method of Conditional Expectations

- The truth value, for which the **sum is larger**, is the one selected for the current variable.
- Ties are broken randomly.

- The truth value, for which the **sum is larger**, is the one selected for the current variable.
- Ties are broken randomly.
- The whole **process is repeated** until all variables are assigned.

Outline

1. Introduction
   - The problem: Max r-Sat
   - The local search: CCLS
   - The initialization: MOCE
2. Correlation
   - Experimental settings
   - End-to-end correlation
   - Ongoing correlation
3. Improving CCLS
4. Conclusion

Diverse initial assignments

- To generate initial assignments of **diverse quality**, we manipulate MOCE

Diverse initial assignments

- To generate initial assignments of **diverse quality**, we manipulate MOCE
- We add to it a parameter that allows us to **invert its decision** regarding the truth value for the current variable.

Diverse initial assignments

- To generate initial assignments of **diverse quality**, we manipulate MOCE
- We add to it a parameter that allows us to **invert its decision** regarding the truth value for the current variable.
- This parameter, is the probability to assign to a variable the truth value **opposite** to the one chosen by MOCE.

Diverse initial assignments

- For a given **inversion probability** $0 \leq p \leq 1$, at each step, we assign to the current variable the truth value chosen by MOCE with probability $1 - p$, and the opposite truth value with probability p.

Diverse initial assignments

- For a given **inversion probability** $0 \leq p \leq 1$, at each step, we assign to the current variable the truth value chosen by MOCE with probability $1 - p$, and the opposite truth value with probability p.
- Thus, for $p = 0$ the algorithm is simply MOCE, while for $p = 1$ it is "anti-MOCE".

Diverse initial assignments

- For a given **inversion probability** $0 \leq p \leq 1$, at each step, we assign to the current variable the truth value chosen by MOCE with probability $1 - p$, and the opposite truth value with probability p.
- Thus, for $p = 0$ the algorithm is simply MOCE, while for $p = 1$ it is "anti-MOCE".
- We refer to this tailored algorithm as **PMOCE**.

Benchmark

- **5 families** of instances of Max 3-Sat.
- Each of the families consists of **150 instances** over 100,000 variables.
- The **densities** of the 5 families are 5, 7, 9, 12, 15.
- The instances in each family were generated uniformly at random.

End-to-end correlation

- For each instance in the family, we executed PMOCE with **51 inversion probabilities**, ranging from 0 to 1 in steps of 0.02.
- Thus, we obtained **51 initial assignments** with presumed diverse quality.

End-to-end correlation

- From each of these initial assignments, we started **a 30 minutes local search** using CCLS, and thus obtained 51 final assignments.
- By the end of the 51 executions, we had **51 pairs** of numbers.
- Each pair consisted of the number of clauses unsatisfied by the **initial** assignment generated by PMOCE, and the number of unsatisfied clauses at the **end** of the search done by CCLS.

Strong correlation

	correlation coefficient			regression slope	
density	**mean**	**std**	p**-value**	**mean**	**std**
5	0.52	0.11	$1.7 \cdot 10^{-3}$	$0.5 \cdot 10^{-3}$	$0.1 \cdot 10^{-3}$
7	0.74	0.06	$3.6 \cdot 10^{-7}$	$1.5 \cdot 10^{-3}$	$0.2 \cdot 10^{-3}$
9	0.79	0.12	$2.1 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$	$0.5 \cdot 10^{-3}$
12	0.73	0.17	$1.2 \cdot 10^{-3}$	$2.4 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$
15	0.83	0.08	$1.1 \cdot 10^{-5}$	$3.4 \cdot 10^{-3}$	$0.7 \cdot 10^{-3}$

Table: End-to-end correlation coefficients and regression slopes.

The results show a **strong positive correlation** between the quality of the initial and final assignment for all densities. The correlation is stronger for denser families.

Statistically significant

	correlation coefficient			regression slope	
density	**mean**	**std**	**p-value**	**mean**	**std**
5	0.52	0.11	$1.7 \cdot 10^{-3}$	$0.5 \cdot 10^{-3}$	$0.1 \cdot 10^{-3}$
7	0.74	0.06	$3.6 \cdot 10^{-7}$	$1.5 \cdot 10^{-3}$	$0.2 \cdot 10^{-3}$
9	0.79	0.12	$2.1 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$	$0.5 \cdot 10^{-3}$
12	0.73	0.17	$1.2 \cdot 10^{-3}$	$2.4 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$
15	0.83	0.08	$1.1 \cdot 10^{-5}$	$3.4 \cdot 10^{-3}$	$0.7 \cdot 10^{-3}$

Table: End-to-end correlation coefficients and regression slopes.

The ***p*-value** is lower by far than the conventional 0.05, which indicates that the correlation coefficients obtained in the experiments are **statistically very significant**.

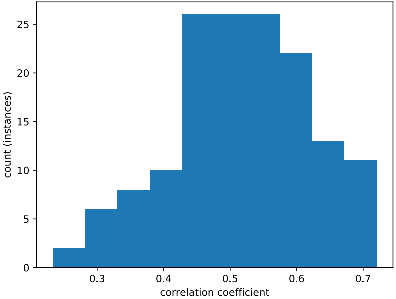
Small slope

	correlation coefficient			regression slope	
density	**mean**	**std**	p**-value**	**mean**	**std**
5	0.52	0.11	$1.7 \cdot 10^{-3}$	$0.5 \cdot 10^{-3}$	$0.1 \cdot 10^{-3}$
7	0.74	0.06	$3.6 \cdot 10^{-7}$	$1.5 \cdot 10^{-3}$	$0.2 \cdot 10^{-3}$
9	0.79	0.12	$2.1 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$	$0.5 \cdot 10^{-3}$
12	0.73	0.17	$1.2 \cdot 10^{-3}$	$2.4 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$
15	0.83	0.08	$1.1 \cdot 10^{-5}$	$3.4 \cdot 10^{-3}$	$0.7 \cdot 10^{-3}$

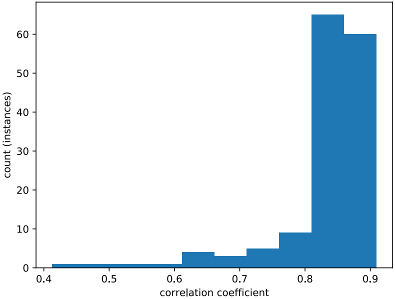
Table: End-to-end correlation coefficients and regression slopes.

The **regression slope** suggests that a large improvement in the initial assignment yields only a **small improvement** in the final assignment.

Histograms of end-to-end correlation coefficients



(a) Family of density 5.



(b) Family of density 15.

Figure: Histograms of end-to-end correlation coefficients.

The figure depicts histograms of the 150 end-to-end correlation coefficients of the family of density 5 (a) and for the family of density 15 (b).

Ongoing correlation

- Besides the end-to-end correlation, we explored the **ongoing** correlation during the experiment.

Ongoing correlation

- Besides the end-to-end correlation, we explored the **ongoing** correlation during the experiment.
- We recorded the minimum number of unsatisfied clauses found so far **after every 1000 flips** made by CCLS.

Ongoing correlation

- Besides the end-to-end correlation, we explored the **ongoing** correlation during the experiment.
- We recorded the minimum number of unsatisfied clauses found so far **after every 1000 flips** made by CCLS.
- Then we calculated the correlation coefficient between the number of clauses unsatisfied by the initial assignment and the number of unsatisfied clauses recorded at each 1000 flips snapshot.

Ongoing correlation

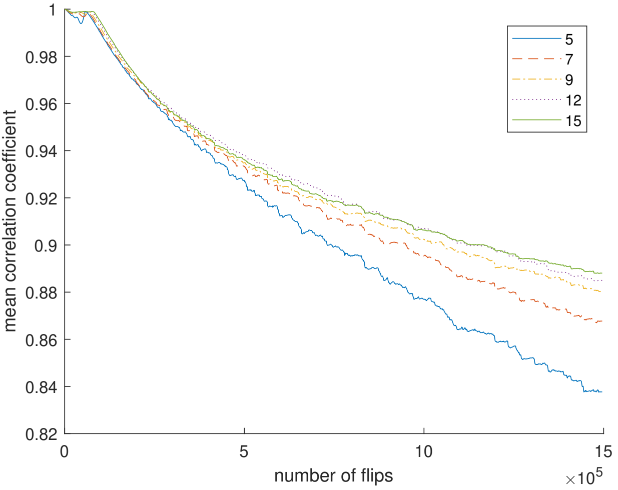


Figure: Ongoing correlation decay as a function of the number of flips.

Outline

1. Introduction
    - The problem: Max r-Sat
    - The local search: CCLS
    - The initialization: MOCE
2. Correlation
    - Experimental settings
    - End-to-end correlation
    - Ongoing correlation
3. Improving CCLS
4. Conclusion

Improving CCLS

- We study the improvement obtained by letting CCLS start its execution from **good initial assignments**.
- Specifically, the good initial assignments we use are assignments provided by MOCE.

Improving CCLS

- We study the improvement obtained by letting CCLS start its execution from **good initial assignments**.
- Specifically, the good initial assignments we use are assignments provided by MOCE.
- We refer to the algorithm that starts from the assignment provided by MOCE as **MOCE-CCLS**.
- To emphasize the fact that the original CCLS algorithm starts from a random assignment, we will call it **RAND-CCLS**.

Improving CCLS

α / n	3			5		
	RC	MC	**% improve**	RC	MC	**% improve**
10000	0	0	**NaN**	248	246	**0.81%**
50000	0	0	**NaN**	1417	1403	**0.99%**
100000	0	0	**NaN**	3038	3002	**1.18%**
500000	0	0	**NaN**	29976	22894	**23.63%**
1000000	72642	0	**100.00%**	320674	75260	**76.53%**
α / n	7			9		
	RC	MC	**% improve**	RC	MC	**% improve**
10000	1265	1264	**0.08%**	2546	2537	**0.35%**
50000	6647	6617	**0.45%**	13122	13052	**0.53%**
100000	13717	13588	**0.94%**	26770	26554	**0.81%**
500000	99976	83163	**16.82%**	178234	152512	**14.43%**
1000000	548044	210440	**61.60%**	769640	363037	**52.83%**

Table: MOCE-CCLS (MC) vs. RAND-CCLS (RC).

Improving CCLS

- We also compared MOCE-CCLS and RAND-CCLS on the random instances of **Max Sat Evaluation 2016**.

Improving CCLS

- We also compared MOCE-CCLS and RAND-CCLS on the random instances of **Max Sat Evaluation 2016**.
- While RAND-CCLS **wins** on the competition instances, it is enough to blow up the instances tenfold to have MOCE-CCLS achieve an overall **draw**.

Improving CCLS

- We also compared MOCE-CCLS and RAND-CCLS on the random instances of **Max Sat Evaluation 2016**.
- While RAND-CCLS **wins** on the competition instances, it is enough to blow up the instances tenfold to have MOCE-CCLS achieve an overall **draw**.
- When scaling the instances by a factor of 100, MOCE-CCLS wins **decisively**, and when scaling by a factor of 1000, it beats RAND-CCLS by a **knockout**.

Outline

1. Introduction
   - The problem: Max r-Sat
   - The local search: CCLS
   - The initialization: MOCE
2. Correlation
   - Experimental settings
   - End-to-end correlation
   - Ongoing correlation
3. Improving CCLS
4. Conclusion

Conclusion

We have explored the **correlation** between the quality of initial assignments provided to local search heuristics and that of the corresponding final assignments.

We have shown that this correlation is **significant and long-lasting**.

Thus, under practical time constraints, **the quality of the initial assignment is crucial** to the performance of local search heuristics.

Conclusion

We demonstrated our point by **improving** the state-of-the-art solver CCLS, by combining it with MOCE.

The combined MOCE-CCLS solver provided a **significant improvement** over CCLS.

Moreover, MOCE-CCLS proved to be **much more scalable** — it handles larger instances better.

Thank you!


THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
DANKSCHEEN
TASHAKKUR ATU
YAQHANYELAY
SUKSAMA
EKHMET
GRAZIE
MEHRBANI
KOMAPSUMNIDA
MAAKE
PALDIES
BOLZÏN
MERCI
BİYAN
SHUKRIA
TINGKI
GOZAIMASHITA
EFCHARISTO
JUSPAXAR
SPASSIBO
SNACHALHUYA
CHALTU
NUHUN
SPASIBO
DENKAUJA
NENACHALHYA
SIKOMO
MERASTAWHY
GAEJTHO
TAVTAPUCH
MEDAWAGSE
AGUYJE
FAKAAUE
MINMONCHAR