Exponential Steepest Ascent from Valued Constraint Graphs of Pathwidth Four

arxiv.org/abs/2405.12906

Artem Kaznatcheev, **Melle van Marle**

Utrecht University

September 6, 2024

Forecast

Question: What properties must a Boolean Valued Constraint Satisfaction Problem (VCSP) satisfy in order to contain an exponentially long steepest ascent?

Forecast

Question: What properties must a Boolean Valued Constraint Satisfaction Problem (VCSP) satisfy in order to contain an exponentially long steepest ascent?

Forecast

Question: What properties must a Boolean Valued Constraint Satisfaction Problem (VCSP) satisfy in order to contain an exponentially long steepest ascent?

Forecast

Question: What properties must a Boolean Valued Constraint Satisfaction Problem (VCSP) satisfy in order to contain an exponentially long steepest ascent?

Result: This paper lowers the bound on the necessary pathwidth of the constraint graph from 7 to 4.

Forecast

Question: What properties must a Boolean Valued Constraint Satisfaction Problem (VCSP) satisfy in order to contain an exponentially long steepest ascent?

Result: This paper lowers the bound on the necessary pathwidth of the constraint graph from 7 to 4.

Forecast

Question: What properties must a Boolean Valued Constraint Satisfaction Problem (VCSP) satisfy in order to contain an exponentially long steepest ascent?

Result: This paper lowers the bound on the necessary pathwidth of the constraint graph from 7 to 4.

Idea: We define a VCSP with a long *ordered ascent*, which can be “simulated" by a steepest ascent on a Boolean VCSP of pathwidth 4.

Motivation

For us: local algorithms are *ascents*.

[1] Bertelè, U., & Brioschi, F. (1973).On non-serial dynamic programming. *Journal of Combinatorial Theory, Series A*, *14*(2), 137–148.

Motivation

For us: local algorithms are *ascents*.

Finding a global maximum on VCSPs of bounded treewidth can be done in polynomial time[1].

[1] Bertelè, U., & Brioschi, F. (1973).On non-serial dynamic programming. *Journal of Combinatorial Theory, Series A*, *14*(2), 137–148.

Motivation

For us: local algorithms are *ascents.*

Finding a global maximum on VCSPs of bounded treewidth can be done in polynomial time[1].

But there are situations where local search is *forced upon us.*

[1] Bertelè, U., & Brioschi, F. (1973).On non-serial dynamic programming. *Journal of Combinatorial Theory, Series A*, *14*(2), 137–148.

Motivation

For us: local algorithms are *ascents*.

Finding a global maximum on VCSPs of bounded treewidth can be done in polynomial time[1].

But there are situations where local search is *forced upon us*.

Example: biological evolution.

[1] Bertelè, U., & Brioschi, F. (1973).On non-serial dynamic programming. *Journal of Combinatorial Theory, Series A*, *14*(2), 137–148.

Motivation

For us: local algorithms are *ascents.*

Finding a global maximum on VCSPs of bounded treewidth can be done in polynomial time[1].

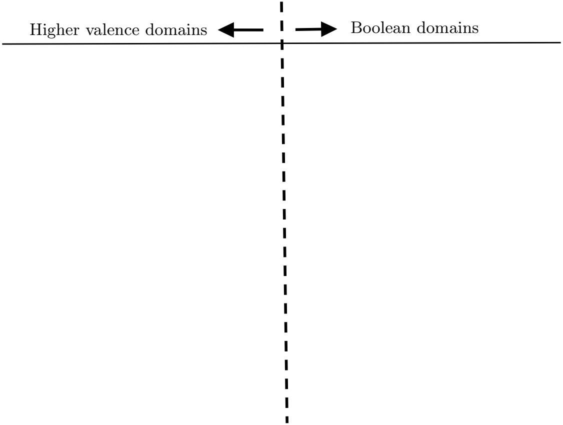
But there are situations where local search is *forced upon us.*

Example: biological evolution.

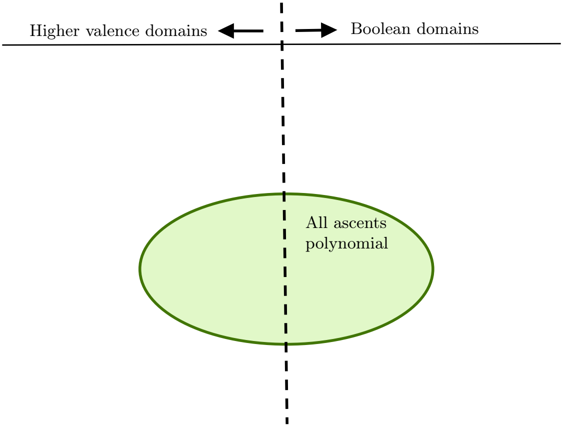
Important question in biology: is evolution open-ended?

[1] Bertelè, U., & Brioschi, F. (1973).On non-serial dynamic programming. *Journal of Combinatorial Theory, Series A, 14*(2), 137–148.

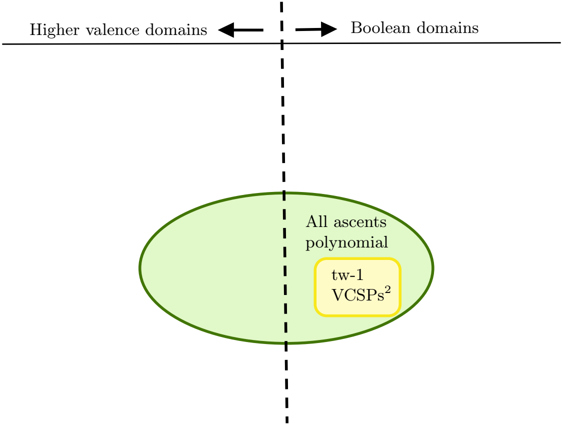
Previous results



Previous results

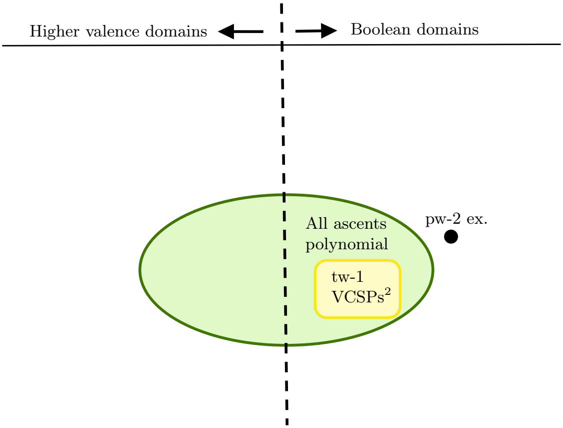


Previous results



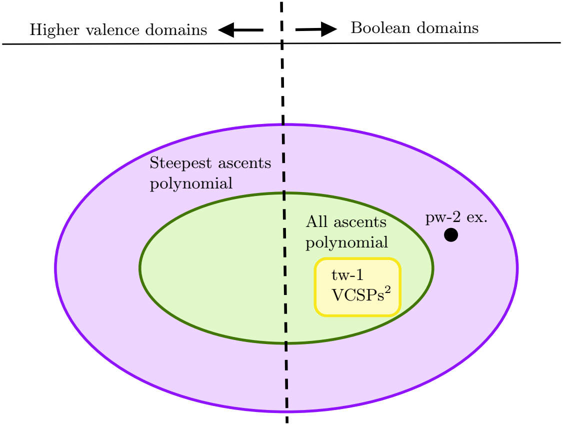
[2] Kaznatcheev, A., Cohen, D. A., & Jeavons, P. (2020).Representing fitness landscapes by valued constraints to understand the complexity of local search. *Journal of Artificial Intelligence Research*, *69*, 1077–1102.

Previous results



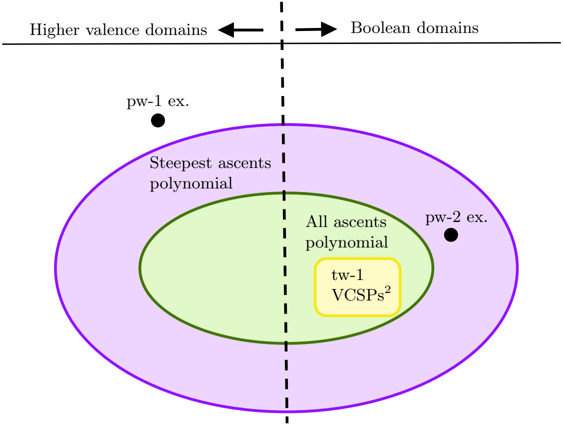
[2] Kaznatcheev, A., Cohen, D. A., & Jeavons, P. (2020).Representing fitness landscapes by valued constraints to understand the complexity of local search. *Journal of Artificial Intelligence Research*, *69*, 1077–1102.

Previous results



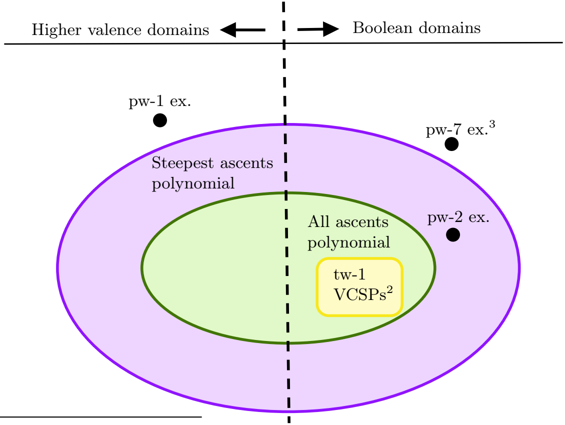
[2] Kaznatcheev, A., Cohen, D. A., & Jeavons, P. (2020).Representing fitness landscapes by valued constraints to understand the complexity of local search. *Journal of Artificial Intelligence Research*, *69*, 1077–1102.

Previous results



[2] Kaznatcheev, A., Cohen, D. A., & Jeavons, P. (2020).Representing fitness landscapes by valued constraints to understand the complexity of local search. *Journal of Artificial Intelligence Research*, *69*, 1077–1102.

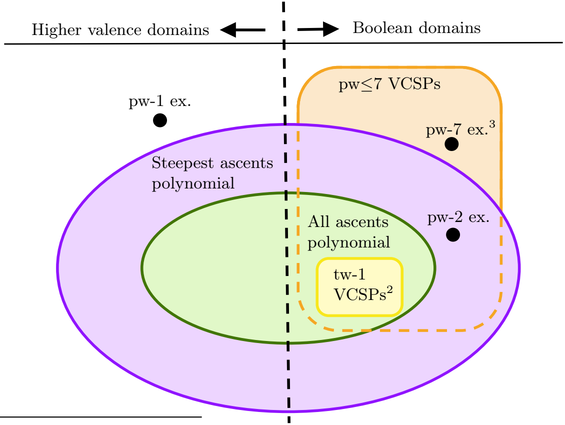
Previous results



[2] Kaznatcheev, A., Cohen, D. A., & Jeavons, P. (2020).Representing fitness landscapes by valued constraints to understand the complexity of local search. *Journal of Artificial Intelligence Research*, *69*, 1077–1102.

[3] Cohen, D. A., Cooper, M. C., Kaznatcheev, A., & Wallace, M. (2020).Steepest ascent can be exponential in bounded treewidth problems. *Operations Research Letters*, *48*, 217–224.

Previous results



[2] Kaznatcheev, A., Cohen, D. A., & Jeavons, P. (2020).Representing fitness landscapes by valued constraints to understand the complexity of local search. *Journal of Artificial Intelligence Research*, *69*, 1077–1102.

[3] Cohen, D. A., Cooper, M. C., Kaznatcheev, A., & Wallace, M. (2020).Steepest ascent can be exponential in bounded treewidth problems. *Operations Research Letters*, *48*, 217–224.

Fitness landscape

Assignments in $\prod_{i=1}^{n} D_i$.

Fitness landscape

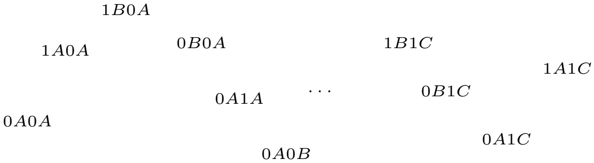


Figure 1: Some assignments in $\{0,1\} \times \{A,B,C\} \times \{0,1\} \times \{A,B,C\}$.

Assignments in $\prod_{i=1}^{n} D_i$.

Fitness landscape

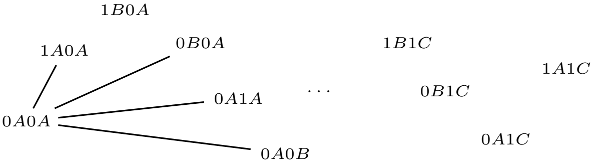


Figure 1: Some assignments in $\{0, 1\} \times \{A, B, C\} \times \{0, 1\} \times \{A, B, C\}$.

Assignments in $\prod_{i=1}^{n} D_i$.

Neighbourhoods for assignments.

Fitness landscape

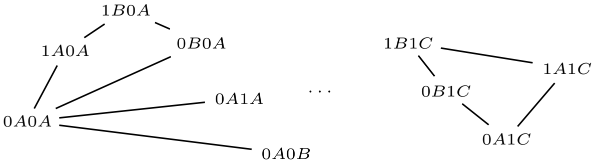


Figure 1: Some assignments in $\{0,1\} \times \{A,B,C\} \times \{0,1\} \times \{A,B,C\}$.

Assignments in $\prod_{i=1}^{n} D_i$.

Neighbourhoods for assignments.

Fitness landscape

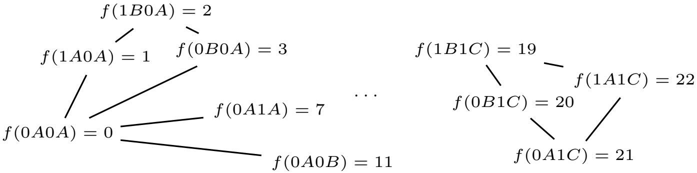


Figure 1: Some assignments in $\{0,1\} \times \{A,B,C\} \times \{0,1\} \times \{A,B,C\}$.

Assignments in $\prod_{i=1}^{n} D_i$.

Neighbourhoods for assignments.

Fitness function $f : \prod_{i=1}^{n} D_i \to \mathbb{Z}$

Definition: VCSP instance

Valued constraint $C_S : \prod_{i \in S} D_i \to \mathbb{Z}$ with **scope** $S \subseteq [n]$ and **arity** $|S|$.

Definition: VCSP instance

Valued constraint $C_S : \prod_{i \in S} D_i \to \mathbb{Z}$ with **scope** $S \subseteq [n]$ and **arity** $|S|$.

Definition

A **VCSP instance** on $\prod_{i=1}^{n} D_i$ is a collection of valued constraints $\{C_{S_1}, C_{S_2}, \ldots, C_{S_m}\}$.

Definition: VCSP instance

Valued constraint $C_S : \prod_{i \in S} D_i \to \mathbb{Z}$ with **scope** $S \subseteq [n]$ and **arity** $|S|$.

Definition

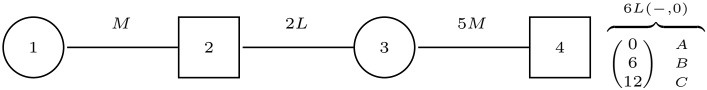
A **VCSP instance** on $\prod_{i=1}^{n} D_i$ is a collection of valued constraints $\{C_{S_1}, C_{S_2}, \ldots, C_{S_m}\}$. The **fitness function** $f : \prod_{i=1^n} D_i \to \mathbb{Z}$ is given by

$$f(x) = \sum_{i=1}^{m} C_{S_i}((x_j)_{j \in S_i}).$$

Example: VCSP instance on 4 variables

Assignments in $\{0,1\} \times \{A,B,C\} \times \{0,1\} \times \{A,B,C\}$.

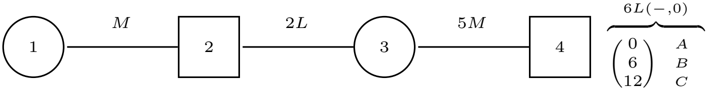
$$\text{Constraints } L = \begin{array}{cc} 0 & 1 \\ \end{array} \begin{pmatrix} 0 & 2 \\ 1 & 1 \\ 2 & 0 \end{pmatrix} \begin{array}{c} A \\ B \\ C \end{array}, \quad M = \begin{array}{ccc} A & B & C \end{array} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{array}{c} 0 \\ 1 \end{array}$$



Example: VCSP instance on 4 variables

Assignments in $\{0,1\} \times \{A,B,C\} \times \{0,1\} \times \{A,B,C\}$.

Constraints $L = \begin{array}{cc} 0 & 1 \\ \end{array}$ $\begin{pmatrix} 0 & 2 \\ 1 & 1 \\ 2 & 0 \end{pmatrix} \begin{array}{c} A \\ B \\ C \end{array}$, $M = \begin{array}{ccc} A & B & C \end{array}$ $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{array}{c} 0 \\ 1 \end{array}$

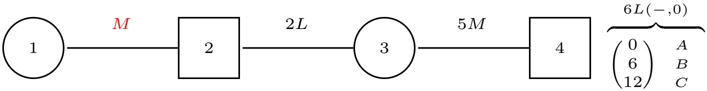


$$f(0B1C) = M(0,B) + 2L(B,1) + 5M(1,C) + 6L(C,0)$$

Example: VCSP instance on 4 variables

Assignments in $\{0,1\} \times \{A,B,C\} \times \{0,1\} \times \{A,B,C\}$.

Constraints $L = \begin{array}{cc} 0 & 1 & \\ \left(\begin{matrix} 0 & 2 \\ 1 & 1 \\ 2 & 0 \end{matrix}\right) & \begin{matrix} A \\ B \\ C \end{matrix} \end{array}$, $M = \begin{array}{cc} A \quad B \quad C & \\ \left(\begin{matrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{matrix}\right) & \begin{matrix} 0 \\ 1 \end{matrix} \end{array}$

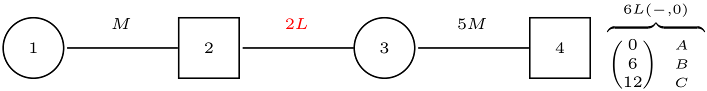


$$f(0B1C) = M(0,B) + 2L(B,1) + 5M(1,C) + 6L(C,0)$$

Example: VCSP instance on 4 variables

Assignments in $\{0,1\} \times \{A,B,C\} \times \{0,1\} \times \{A,B,C\}$.

$$\text{Constraints } L = \begin{array}{cc} 0 & 1 \\ \left(\begin{matrix} 0 & 2 \\ 1 & 1 \\ 2 & 0 \end{matrix}\right) & \begin{matrix} A \\ B \\ C \end{matrix} \end{array}, \; M = \begin{array}{cc} \begin{matrix} A & B & C \end{matrix} & \\ \left(\begin{matrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{matrix}\right) & \begin{matrix} 0 \\ 1 \end{matrix} \end{array}$$

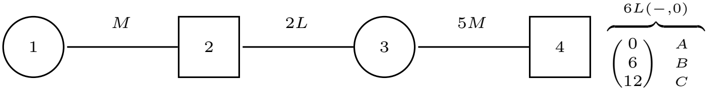

1
M
2
$2L$
3
$5M$
4
$\overbrace{\begin{pmatrix} 0 \\ 6 \\ 12 \end{pmatrix} \begin{matrix} A \\ B \\ C \end{matrix}}^{6L(-,0)}$

$$f(0B1C) = M(0,B) + 2L(B,1) + 5M(1,C) + 6L(C,0)$$

Example: VCSP instance on 4 variables

Assignments in $\{0,1\} \times \{A,B,C\} \times \{0,1\} \times \{A,B,C\}$.

Constraints $L = \begin{array}{cc} 0 & 1 & \\ \left(\begin{array}{cc} 0 & 2 \\ 1 & 1 \\ 2 & 0 \end{array}\right) & & \begin{array}{c} A \\ B \\ C \end{array} \end{array}$, $M = \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} & \\ \left(\begin{array}{ccc} 0 & 1 & 0 \\ 1 & 0 & 1 \end{array}\right) & \begin{array}{c} 0 \\ 1 \end{array} \end{array}$

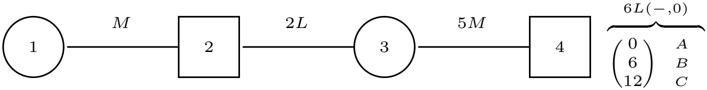


$$f(0B1C) = M(0,B) + 2L(B,1) + 5M(1,C) + 6L(C,0)$$

Example: VCSP instance on 4 variables

Assignments in $\{0,1\} \times \{A,B,C\} \times \{0,1\} \times \{A,B,C\}$.

$$\text{Constraints } L = \begin{array}{cc} 0 & 1 \\ \end{array} \begin{pmatrix} 0 & 2 \\ 1 & 1 \\ 2 & 0 \end{pmatrix} \begin{array}{c} A \\ B \\ C \end{array}, \quad M = \begin{array}{ccc} A & B & C \end{array} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{array}{c} 0 \\ 1 \end{array}$$



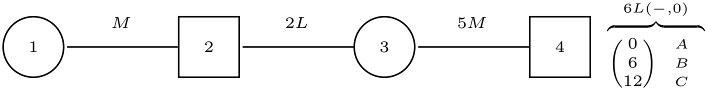
$$f(0B1C) = M(0,B) + 2L(B,1) + 5M(1,C) + 6L(C,0)$$

Example: VCSP instance on 4 variables

Assignments in $\{0,1\} \times \{A,B,C\} \times \{0,1\} \times \{A,B,C\}$.

Constraints $L = \begin{array}{cc} 0 & 1 \\ \end{array}$

$$L = \begin{array}{c|cc} & 0 & 1 \\ \hline A & 0 & 2 \\ B & 1 & 1 \\ C & 2 & 0 \end{array}, \quad M = \begin{array}{c|ccc} & A & B & C \\ \hline 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{array}$$

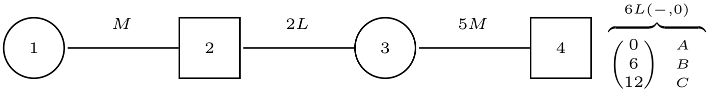


$$f(0B1C) = 1 + 2L(B,1) + 5M(1,C) + 6L(C,0)$$

Example: VCSP instance on 4 variables

Assignments in $\{0,1\} \times \{A,B,C\} \times \{0,1\} \times \{A,B,C\}$.

Constraints $L = \begin{array}{cc} \begin{array}{cc} 0 & 1 \end{array} & \\ \begin{pmatrix} 0 & 2 \\ 1 & 1 \\ 2 & 0 \end{pmatrix} & \begin{array}{c} A \\ B \\ C \end{array} \end{array}$, $M = \begin{array}{cc} \begin{array}{ccc} A & B & C \end{array} & \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} & \begin{array}{c} 0 \\ 1 \end{array} \end{array}$

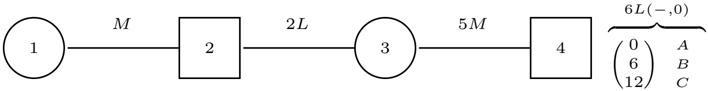


$$f(0B1C) = 1 + 2L(B,1) + 5M(1,C) + 6L(C,0)$$

Example: VCSP instance on 4 variables

Assignments in $\{0,1\} \times \{A,B,C\} \times \{0,1\} \times \{A,B,C\}$.

$$\text{Constraints } L = \begin{array}{cc} \begin{matrix} 0 & 1 \end{matrix} & \\ \begin{pmatrix} 0 & 2 \\ 1 & 1 \\ 2 & 0 \end{pmatrix} & \begin{matrix} A \\ B \\ C \end{matrix} \end{array}, \quad M = \begin{array}{cc} \begin{matrix} A & B & C \end{matrix} & \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} & \begin{matrix} 0 \\ 1 \end{matrix} \end{array}$$

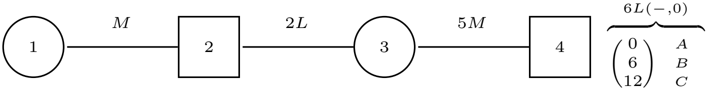


$$f(0B1C) = 1 + 2 + 5M(1,C) + 6L(C,0)$$

Example: VCSP instance on 4 variables

Assignments in $\{0,1\} \times \{A,B,C\} \times \{0,1\} \times \{A,B,C\}$.

Constraints $L = \begin{array}{cc} 0 & 1 & \\ \left(\begin{matrix} 0 & 2 \\ 1 & 1 \\ 2 & 0 \end{matrix}\right) & \begin{matrix} A \\ B \\ C \end{matrix} \end{array}$, $M = \begin{array}{cc} \begin{matrix} A & B & C \end{matrix} & \\ \left(\begin{matrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{matrix}\right) & \begin{matrix} 0 \\ 1 \end{matrix} \end{array}$

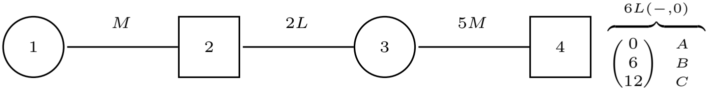

1
M
2
2L
3
5M
4
$\overbrace{\begin{pmatrix} 0 \\ 6 \\ 12 \end{pmatrix} \begin{matrix} A \\ B \\ C \end{matrix}}^{6L(-,0)}$

$$f(0B1C) = 1 + 2 + 5 + 12$$

Example: VCSP instance on 4 variables

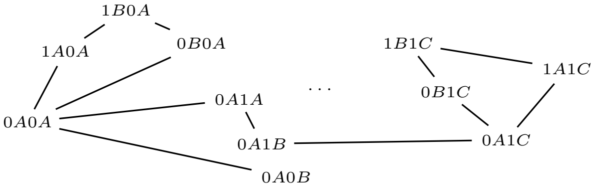
Assignments in $\{0,1\} \times \{A,B,C\} \times \{0,1\} \times \{A,B,C\}$.

$$\text{Constraints } L = \begin{array}{ccc} 0 & 1 & \\ 0 & 2 & A \\ 1 & 1 & B \\ 2 & 0 & C \end{array}, \quad M = \begin{array}{cccc} A & B & C & \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{array}$$

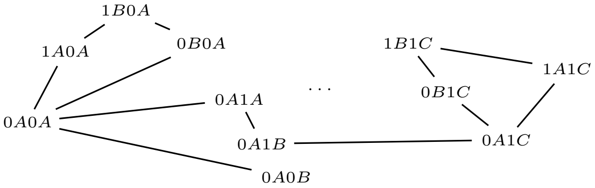

1
M
2
$2L$
3
$5M$
4
$\overbrace{\begin{pmatrix}0\\6\\12\end{pmatrix}\begin{matrix}A\\B\\C\end{matrix}}^{6L(-,0)}$

$$f(0B1C) = 20$$

Definition: ascents

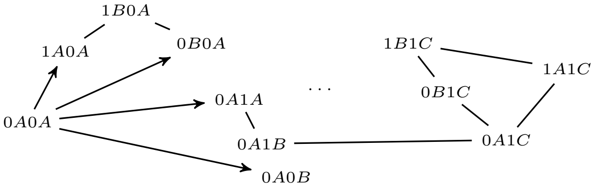


Definition: ascents



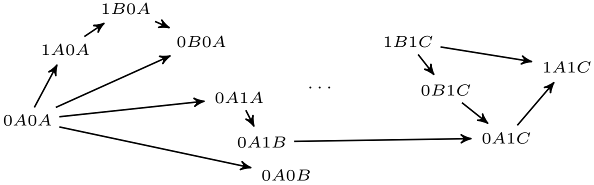
Ascent is sequence of assignments with increasing fitness, where each pair of consecutive assignments are neighbors.

Definition: ascents



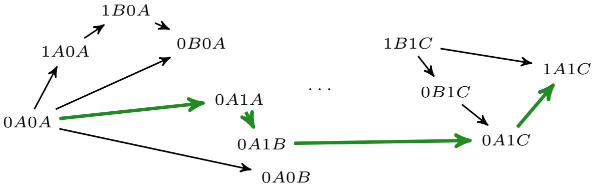
Ascent is sequence of assignments with increasing fitness, where each pair of consecutive assignments are neighbors.

Definition: ascents



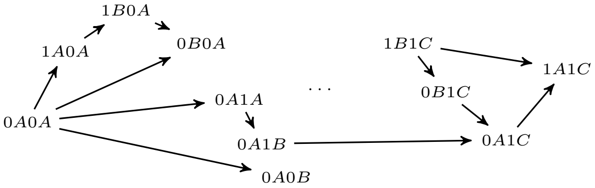
Ascent is sequence of assignments with increasing fitness, where each pair of consecutive assignments are neighbors.

Definition: ascents



Ascent is sequence of assignments with increasing fitness, where each pair of consecutive assignments are neighbors.

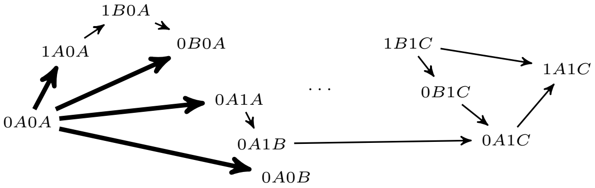
Definition: ascents



Ascent is sequence of assignments with increasing fitness, where each pair of consecutive assignments are neighbors.

Steepest ascent is ascent in which every step is the one with highest fitness increase.

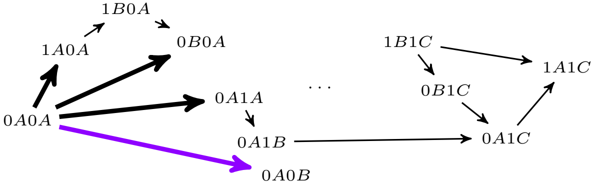
Definition: ascents



Ascent is sequence of assignments with increasing fitness, where each pair of consecutive assignments are neighbors.

Steepest ascent is ascent in which every step is the one with highest fitness increase.

Definition: ascents



Ascent is sequence of assignments with increasing fitness, where each pair of consecutive assignments are neighbors.

Steepest ascent is ascent in which every step is the one with highest fitness increase.

Definition: ordered ascents

Definition

Let $\mathcal{C} = \{C_{S_1}, \ldots, C_{S_m}\}$ be a VCSP instance on $\prod_{i=1}^{n} D_i$. Let $\prec$ be an ordering on $\{1, 2, \ldots, n\}$. We call an ascent $p = (x^0, \ldots, x^T)$ a $\prec$-**ordered ascent** on $\mathcal{C}$ if at any time, p changes an entry in the domain with $\prec$-minimal index where a change can yield a fitness increase.

Definition: ordered ascents

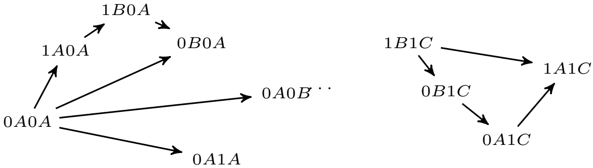


Figure 2: $<$-ordered step from assignment $0A0A$.

Definition

Let $\mathcal{C} = \{C_{S_1}, \ldots, C_{S_m}\}$ be a VCSP instance on $\prod_{i=1}^{n} D_i$. Let $\prec$ be an ordering on $\{1, 2, \ldots, n\}$. We call an ascent $p = (x^0, \ldots, x^T)$ a $\prec$-ordered ascent on $\mathcal{C}$ if at any time, p changes an entry in the domain with $\prec$-minimal index where a change can yield a fitness increase.

Definition: ordered ascents

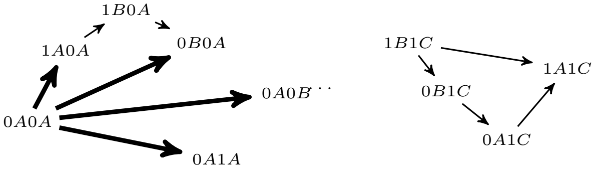


Figure 2: $\prec$-ordered step from assignment $0A0A$.

Definition

Let $\mathcal{C} = \{C_{S_1}, \ldots, C_{S_m}\}$ be a VCSP instance on $\prod_{i=1}^{n} D_i$. Let $\prec$ be an ordering on $\{1, 2, \ldots, n\}$. We call an ascent $p = (x^0, \ldots, x^T)$ a $\prec$-**ordered ascent** on $\mathcal{C}$ if at any time, p changes an entry in the domain with $\prec$-minimal index where a change can yield a fitness increase.

Definition: ordered ascents

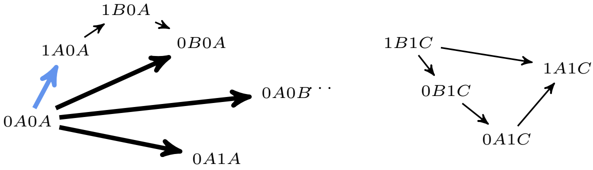


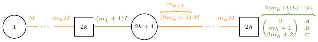
Figure 2: $\prec$-ordered step from assignment $0A0A$.

Definition

Let $\mathcal{C} = \{C_{S_1}, \ldots, C_{S_m}\}$ be a VCSP instance on $\prod_{i=1}^{n} D_i$. Let $\prec$ be an ordering on $\{1, 2, \ldots, n\}$. We call an ascent $p = (x^0, \ldots, x^T)$ a $\prec$-**ordered ascent** on $\mathcal{C}$ if at any time, p changes an entry in the domain with $\prec$-minimal index where a change can yield a fitness increase.

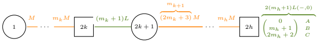
The construction

STEP 1: 2-by-3 VCSP with long ordered ascent.

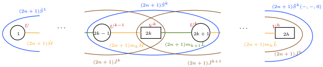


The construction

STEP 1: 2-by-3 VCSP with long ordered ascent.

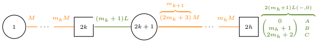

1
M
$\cdots$
$m_k M$
$2k$
$(m_k+1)L$
$2k+1$
$\overbrace{(2m_k+3)}^{m_{k+1}} M$
$\cdots$
$m_h M$
$2h$
$\overbrace{\begin{pmatrix} 0 \\ m_h+1 \\ 2m_h+2 \end{pmatrix} \begin{matrix} A \\ B \\ C \end{matrix}}^{2(m_h+1)L(-,0)}$

STEP 2: "Padded" VCSP with large domains and long steepest ascent.

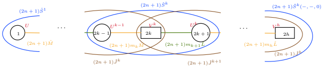


The construction

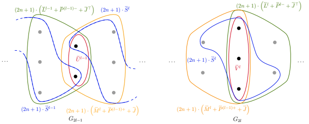
STEP 1: 2-by-3 VCSP with long ordered ascent.


1
M
$\cdots$
$m_k M$
$2k$
$(m_k+1)L$
$2k+1$
$\overbrace{(2m_k+3)}^{m_{k+1}} M$
$\cdots$
$m_h M$
$2h$
$\overbrace{\begin{pmatrix} 0 \\ m_h+1 \\ 2m_h+2 \end{pmatrix}}^{2(m_h+1)L(-,0)}$
A
B
C

STEP 2: "Padded" VCSP with large domains and long steepest ascent.



STEP 3: "Encoded" VCSP with Boolean domains and long steepest ascent.


$(2n+1)\cdot\left(\tilde{\tilde{L}}^{l-1}+\tilde{\tilde{P}}^{(l-1)-}+\tilde{\tilde{J}}^{\top}\right)$
$(2n+1)\cdot\tilde{\tilde{S}}^l$
$\cdots$
$\tilde{\tilde{U}}^{l-1}$
$\cdots$
$(2n+1)\cdot\tilde{\tilde{S}}^{l-1}$
$(2n+1)\cdot\left(\tilde{\tilde{M}}^l+\tilde{\tilde{P}}^{(l-1)+}+\tilde{\tilde{J}}\right)$
G_{2l-1}
$(2n+1)\cdot\left(\tilde{\tilde{L}}^l+\tilde{\tilde{P}}^{l-}+\tilde{\tilde{J}}^{\top}\right)$
$\cdots$
$(2n+1)\cdot\tilde{\tilde{S}}^l$
$\tilde{\tilde{V}}^l$
$\cdots$
$(2n+1)\cdot\left(\tilde{\tilde{M}}^l+\tilde{\tilde{P}}^{(l-1)+}+\tilde{\tilde{J}}\right)$
G_{2l}

STEP 1: 2-by-3 VCSP instance

Assignments in $(\{0,1\} \times \{A,B,C\})^h$

STEP 1: 2-by-3 VCSP instance

Assignments in $(\{0,1\} \times \{A,B,C\})^h$

Constraints $L = \begin{array}{c@{}c} \begin{matrix} A & B \end{matrix} & \\ \begin{pmatrix} 0 & 2 \\ 1 & 1 \\ 2 & 0 \end{pmatrix} & \begin{matrix} A \\ B \\ C \end{matrix} \end{array}$, $M = \begin{array}{c@{}c} \begin{matrix} A & B & C \end{matrix} & \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} & \begin{matrix} A \\ B \end{matrix} \end{array}$

STEP 1: 2-by-3 VCSP instance

Assignments in $(\{0,1\} \times \{A,B,C\})^h$

Constraints $L = \begin{array}{cc} & \\ \begin{pmatrix} \overset{A}{0} & \overset{B}{2} \\ 1 & 1 \\ 2 & 0 \end{pmatrix} & \begin{array}{c} A \\ B \\ C \end{array} \end{array}$, $M = \begin{array}{cc} \begin{pmatrix} \overset{A}{0} & \overset{B}{1} & \overset{C}{0} \\ 1 & 0 & 1 \end{pmatrix} & \begin{array}{c} A \\ B \end{array} \end{array}$

Weights: $m_1 = 1$, $m_{k+1} = 2m_k + 3$

STEP 1: 2-by-3 VCSP instance

Assignments in $(\{0,1\} \times \{A,B,C\})^h$

Constraints $L = \begin{array}{cc} A & B \\ \begin{pmatrix} 0 & 2 \\ 1 & 1 \\ 2 & 0 \end{pmatrix} & \begin{array}{c} A \\ B \\ C \end{array} \end{array}$, $M = \begin{array}{cc} \begin{array}{ccc} A & B & C \end{array} & \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} & \begin{array}{c} A \\ B \end{array} \end{array}$

Weights: $m_1 = 1$, $m_{k+1} = 2m_k + 3$

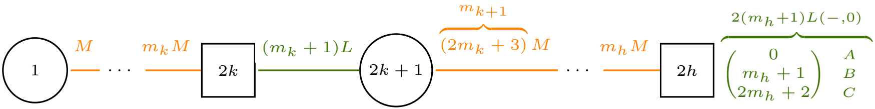
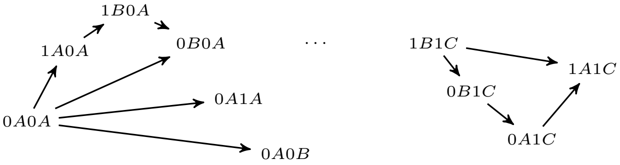


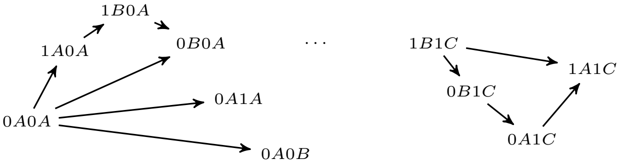
Figure 3: The 2-by-3 VCSP instance on $n = 2h$ variables.

STEP 1: long ordered ascent



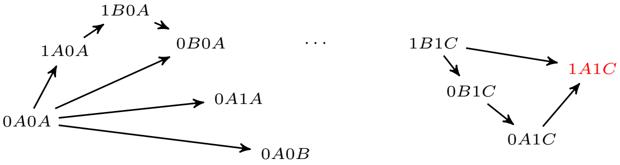
Claim: The 2-by-3 VCSP instance has a single local peak $1A1A \ldots 1A1C$.

STEP 1: long ordered ascent



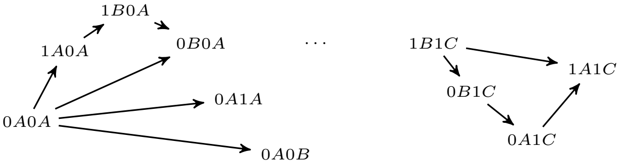
Claim: The 2-by-3 VCSP instance has a single local peak $1A1A \dots 1A1C$.

STEP 1: long ordered ascent



Claim: The 2-by-3 VCSP instance has a single local peak $1A1A\ldots1A1C$.

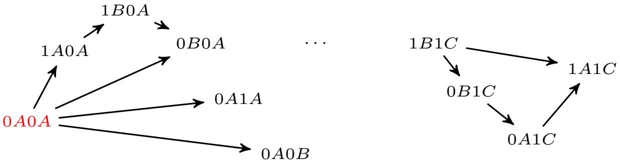
STEP 1: long ordered ascent



Claim: The 2-by-3 VCSP instance has a single local peak $1A1A\ldots1A1C$.

We consider the $<$-ordered ascent starting at $0A0A\ldots0A$.

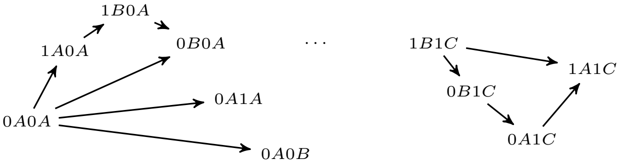
STEP 1: long ordered ascent



Claim: The 2-by-3 VCSP instance has a single local peak $1A1A\ldots1A1C$.

We consider the $<$-ordered ascent starting at $0A0A\ldots0A$.

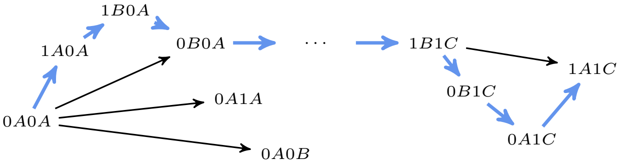
STEP 1: long ordered ascent



Claim: The 2-by-3 VCSP instance has a single local peak $1A1A \dots 1A1C$.

We consider the $<$-ordered ascent starting at $0A0A \dots 0A$.

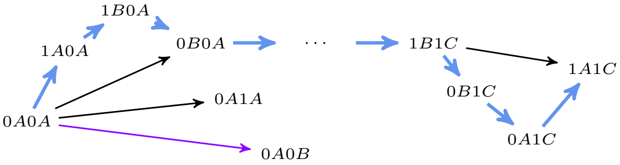
STEP 1: long ordered ascent



Claim: The 2-by-3 VCSP instance has a single local peak $1A1A \ldots 1A1C$.

We consider the $<$-ordered ascent starting at $0A0A \ldots 0A$.
Claim: Every step in this ascent increases fitness by 1.

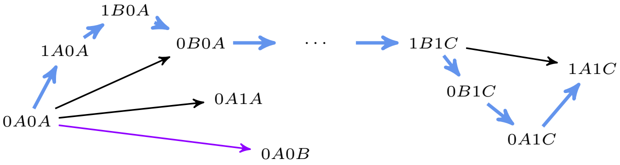
STEP 1: long ordered ascent



Claim: The 2-by-3 VCSP instance has a single local peak $1A1A \ldots 1A1C$.

We consider the $<$-ordered ascent starting at $0A0A \ldots 0A$.
Claim: Every step in this ascent increases fitness by 1.

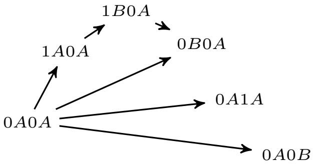
STEP 1: long ordered ascent

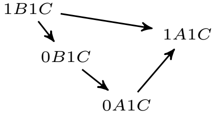


Claim: The 2-by-3 VCSP instance has a single local peak $1A1A \dots 1A1C$.

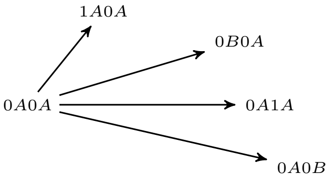
We consider the $<$-ordered ascent starting at $0A0A \dots 0A$.
Claim: Every step in this ascent increases fitness by 1.
Ordered ascent takes an exponential number of steps.

STEP 2: Padded VCSP with long steepest ascent

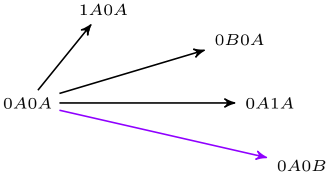




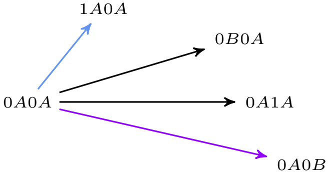
STEP 2: Padded VCSP with long steepest ascent



STEP 2: Padded VCSP with long steepest ascent

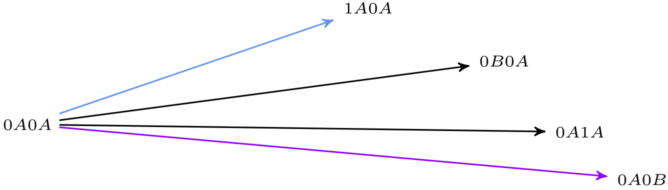


STEP 2: Padded VCSP with long steepest ascent



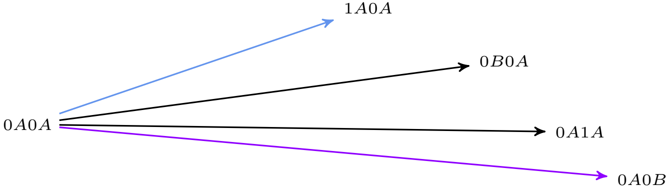
$\widehat{f}(x) =$

STEP 2: Padded VCSP with long steepest ascent



$$\widehat{f}(x) = \begin{cases} (2n+1)f(x), & x \in \prod_{i=1}^{n} D_i \\ & \\ & \end{cases}$$

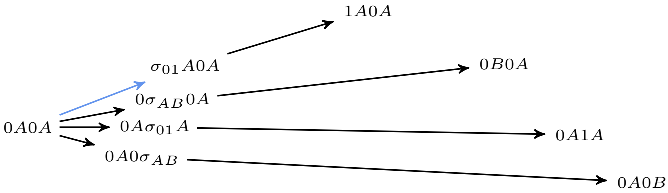
STEP 2: Padded VCSP with long steepest ascent



Increase the size of each domain D_i by adding *intermediate states* $\sigma_{u,v}$ for $u, v \in D_i$.

$$\widehat{f}(x) = \begin{cases} (2n+1)f(x), & x \in \prod_{i=1}^{n} D_i \\ & \\ & \end{cases}$$

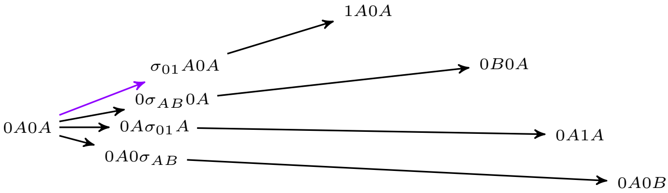
STEP 2: Padded VCSP with long steepest ascent



Increase the size of each domain D_i by adding *intermediate states* $\sigma_{u,v}$ for $u, v \in D_i$.

$$\widehat{f}(x) = \begin{cases} (2n+1)f(x), & x \in \prod_{i=1}^{n} D_i \\ n-k+1+(2n+1)\min_{h\in\{u,v\}} f(x[k:h]), & \text{x contains exactly one} \\ & \text{intermediate state $\sigma_{u,v}$} \\ & \text{ at position k.} \\ \text{"Small enough"}, & \text{Otherwise} \end{cases}$$

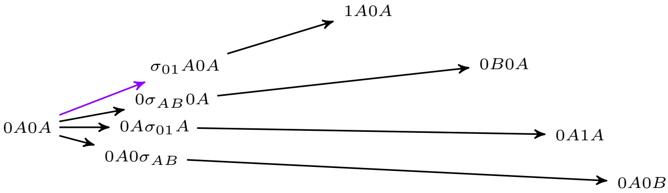
STEP 2: Padded VCSP with long steepest ascent



Increase the size of each domain D_i by adding *intermediate states* $\sigma_{u,v}$ for $u, v \in D_i$.

$$\widehat{f}(x) = \begin{cases} (2n+1)f(x), & x \in \prod_{i=1}^{n} D_i \\ n-k+1+(2n+1)\min_{h\in\{u,v\}} f(x[k:h]), & x \text{ contains exactly one} \\ & \text{intermediate state } \sigma_{u,v} \\ & \text{ at position } k. \\ \text{"Small enough"}, & \text{Otherwise} \end{cases}$$

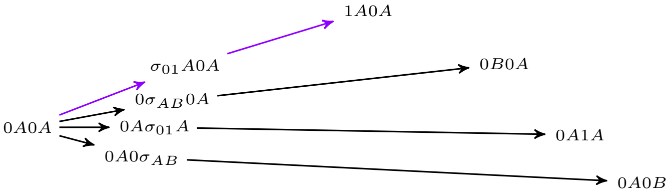
STEP 2: Padded VCSP with long steepest ascent



Increase the size of each domain D_i by adding *intermediate states* $\sigma_{u,v}$ for $u, v \in D_i$.

$$\widehat{f}(x) = \begin{cases} (2n+1)f(x), & x \in \prod_{i=1}^{n} D_i \\ n-k+1+(2n+1)\min_{h \in \{u,v\}} f(x[k:h]), & x \text{ contains exactly one} \\ & \text{intermediate state } \sigma_{u,v} \\ & \text{ at position } k. \\ \text{"Small enough"}, & \text{Otherwise} \end{cases}$$

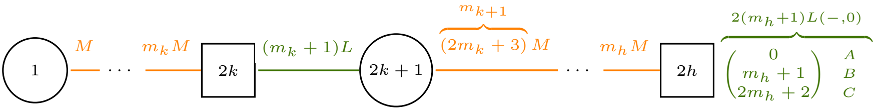
STEP 2: Padded VCSP with long steepest ascent



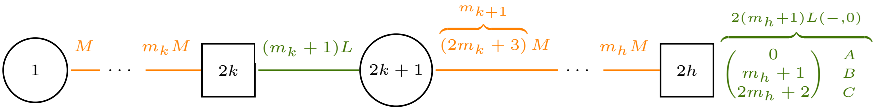
Increase the size of each domain D_i by adding *intermediate states* $\sigma_{u,v}$ for $u, v \in D_i$.

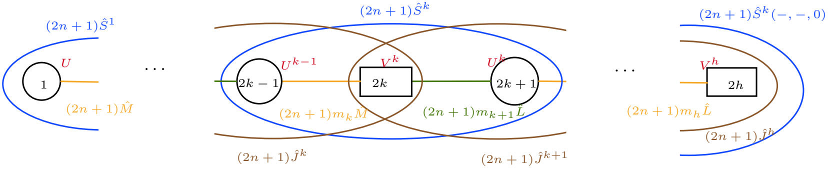
$$\widehat{f}(x) = \begin{cases} (2n+1)f(x), & x \in \prod_{i=1}^{n} D_i \\ n-k+1+(2n+1)\min_{h\in\{u,v\}} f(x[k:h]), & x \text{ contains exactly one} \\ & \text{intermediate state } \sigma_{u,v} \\ & \text{ at position } k. \\ \text{"Small enough"}, & \text{Otherwise} \end{cases}$$

STEP 2: Padded VCSP with long steepest ascent

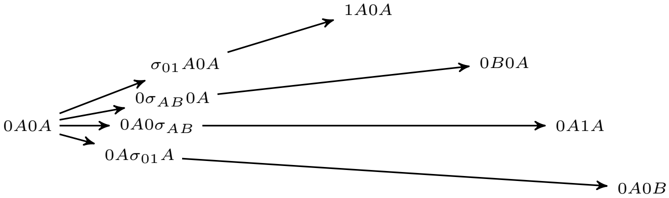

1
M
$\cdots$
$m_k M$
$2k$
$(m_k + 1)L$
$2k + 1$
$\overbrace{(2m_k + 3)}^{m_{k+1}} M$
$\cdots$
$m_h M$
$2h$
$\overbrace{\begin{pmatrix} 0 \\ m_h + 1 \\ 2m_h + 2 \end{pmatrix} \begin{matrix} A \\ B \\ C \end{matrix}}^{2(m_h+1)L(-,0)}$

STEP 2: Padded VCSP with long steepest ascent





STEP 3: Boolean Encoded VCSP



$1A0A$

$\sigma_{01}A0A$

$0B0A$

$0\sigma_{AB}0A$

$0A0A$

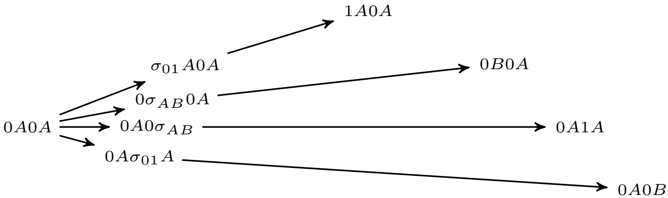
$0A0\sigma_{AB}$

$0A1A$

$0A\sigma_{01}A$

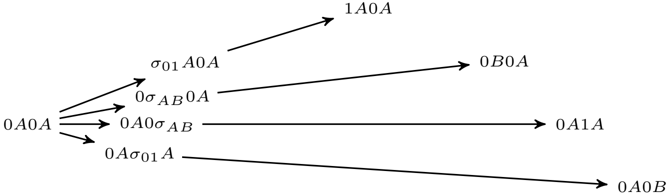
$0A0B$

STEP 3: Boolean Encoded VCSP



We encode the higher valence domains by Boolean domains.

STEP 3: Boolean Encoded VCSP

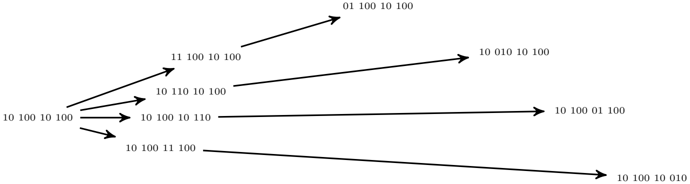


We encode the higher valence domains by Boolean domains.

$$\begin{aligned} 0 &\mapsto 10 \\ \sigma_{01} &\mapsto 11 \\ 1 &\mapsto 01 \end{aligned}$$

$$\begin{aligned} A &\mapsto 100 \\ \sigma_{AB} &\mapsto 110 \\ B &\mapsto 010 \\ \sigma_{BC} &\mapsto 011 \\ C &\mapsto 001 \end{aligned}$$

STEP 3: Boolean Encoded VCSP



01 100 10 100

11 100 10 100

10 010 10 100

10 110 10 100

10 100 10 100

10 100 10 110

10 100 01 100

10 100 11 100

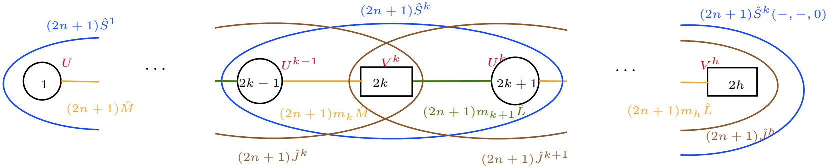
10 100 10 010

We encode the higher valence domains by Boolean domains.

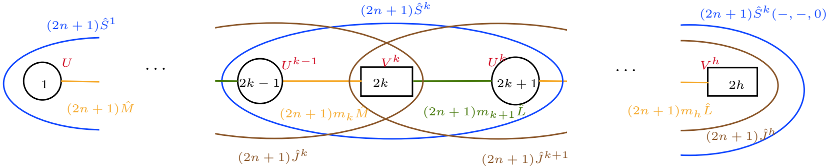
$$\begin{aligned} 0 &\mapsto 10 \\ \sigma_{01} &\mapsto 11 \\ 1 &\mapsto 01 \end{aligned}$$

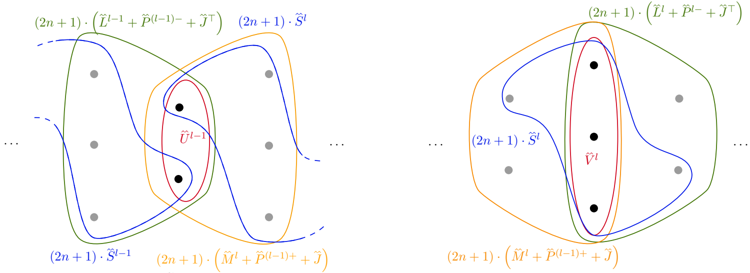
$$\begin{aligned} A &\mapsto 100 \\ \sigma_{AB} &\mapsto 110 \\ B &\mapsto 010 \\ \sigma_{BC} &\mapsto 011 \\ C &\mapsto 001 \end{aligned}$$

STEP 3: Boolean Encoded VCSP



STEP 3: Boolean Encoded VCSP

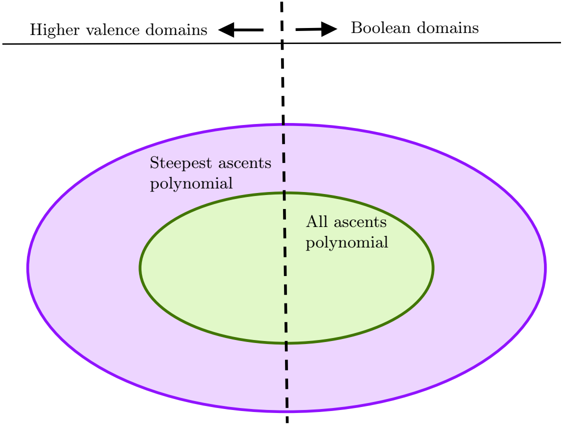



$(2n+1)\cdot\left(\widehat{\widetilde{L}}^{l-1}+\widehat{\widetilde{P}}^{(l-1)-}+\widehat{\widetilde{J}}^{\top}\right)$
$(2n+1)\cdot\widehat{\widetilde{S}}^{l}$
$\cdots$
$\widehat{\widetilde{U}}^{l-1}$
$\cdots$
$(2n+1)\cdot\widehat{\widetilde{S}}^{l-1}$
$(2n+1)\cdot\left(\widehat{\widetilde{M}}^{l}+\widehat{\widetilde{P}}^{(l-1)+}+\widehat{\widetilde{J}}\right)$
$(2n+1)\cdot\left(\widehat{\widetilde{L}}^{l}+\widehat{\widetilde{P}}^{l-}+\widehat{\widetilde{J}}^{\top}\right)$
$\cdots$
$(2n+1)\cdot\widehat{\widetilde{S}}^{l}$
$\widehat{\widetilde{V}}^{l}$
$\cdots$
$(2n+1)\cdot\left(\widehat{\widetilde{M}}^{l}+\widehat{\widetilde{P}}^{(l-1)+}+\widehat{\widetilde{J}}\right)$

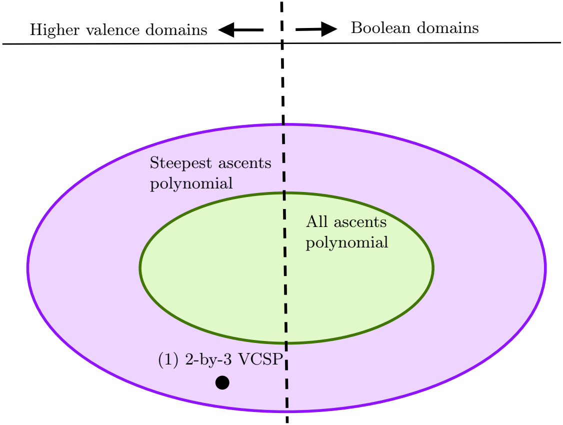
G_{2l-1}

G_{2l}

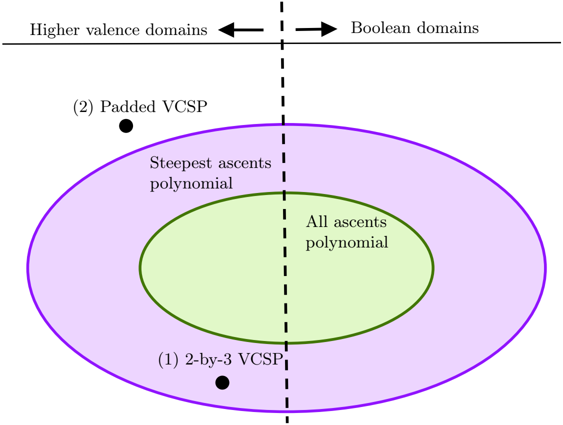
Summary


Higher valence domains
Boolean domains
Steepest ascents polynomial
All ascents polynomial

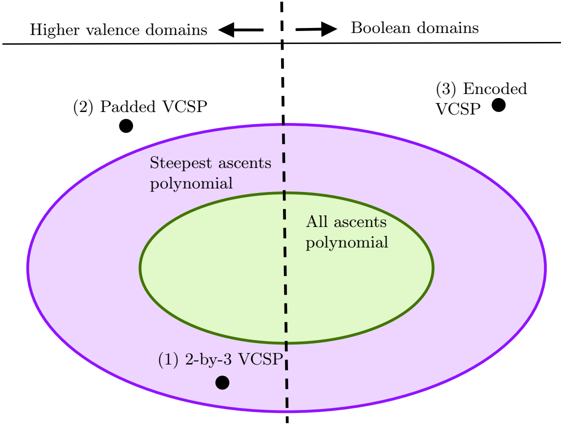
Summary


Higher valence domains
Boolean domains
Steepest ascents polynomial
All ascents polynomial
(1) 2-by-3 VCSP

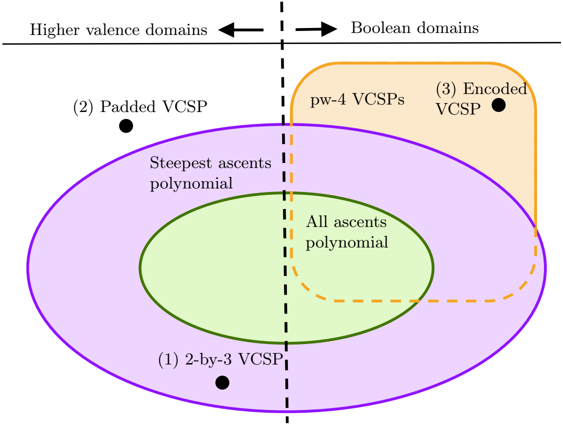
Summary


Higher valence domains
Boolean domains
(2) Padded VCSP
Steepest ascents polynomial
All ascents polynomial
(1) 2-by-3 VCSP

Summary

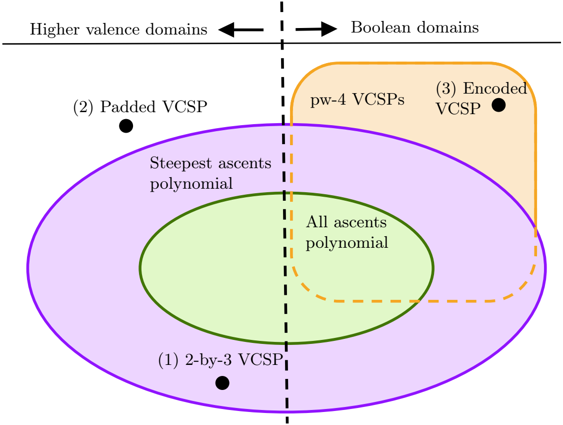

Higher valence domains
Boolean domains
(2) Padded VCSP
(3) Encoded VCSP
Steepest ascents polynomial
All ascents polynomial
(1) 2-by-3 VCSP

Summary


Higher valence domains
Boolean domains
(2) Padded VCSP
pw-4 VCSPs
(3) Encoded VCSP
Steepest ascents polynomial
All ascents polynomial
(1) 2-by-3 VCSP

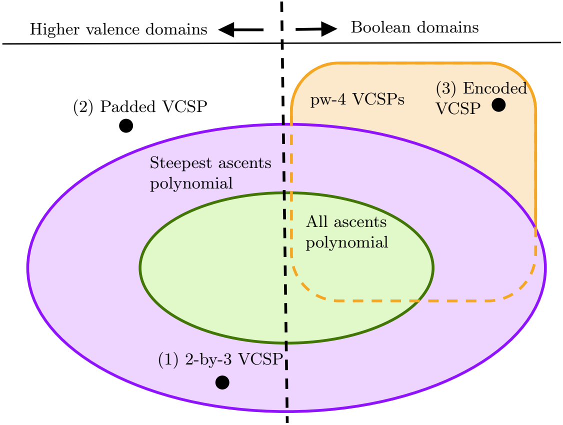
Summary

- **Conjecture 12.** *There exists a polynomial $p(n)$ such that for any Boolean VCSP instance $\mathcal{C}$ on n variables if the constraint graph of $\mathcal{C}$ has pathwidth ≤ 2, then any steepest ascent in the associated fitness landscape has length at most $p(n)$.*



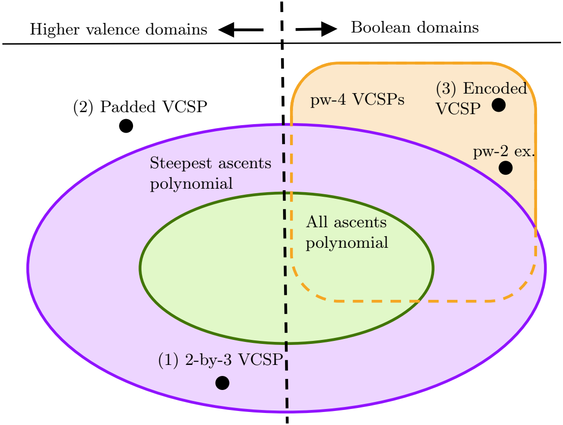
Summary

- **Conjecture 12.** ~~*There exists a polynomial $p(n)$ such that for any Boolean VCSP instance $\mathcal{C}$ on n variables if the constraint graph of $\mathcal{C}$ has pathwidth ≤ 2, then any steepest ascent in the associated fitness landscape has length at most $p(n)$.*~~



Summary

- **Conjecture 12.** ~~*There exists a polynomial $p(n)$ such that for any Boolean VCSP instance $\mathcal{C}$ on n variables if the constraint graph of $\mathcal{C}$ has pathwidth ≤ 2, then any steepest ascent in the associated fitness landscape has length at most $p(n)$.*~~



References

Bertelè, U., & Brioschi, F. (1973).On non-serial dynamic programming. *Journal of Combinatorial Theory, Series A*, *14*(2), 137–148.

Cohen, D. A., Cooper, M. C., Kaznatcheev, A., & Wallace, M. (2020).Steepest ascent can be exponential in bounded treewidth problems. *Operations Research Letters*, *48*, 217–224.

Kaznatcheev, A., Cohen, D. A., & Jeavons, P. (2020).Representing fitness landscapes by valued constraints to understand the complexity of local search. *Journal of Artificial Intelligence Research*, *69*, 1077–1102.