Modelling Equidistant Frequency Permutation Arrays: An Application of Constraints to Mathematics

Sophie Huczynska, Paul McKay,
Ian Miguel and Peter Nightingale

Introduction

- We used CP to contribute to work in theoretical mathematics
  - Directly refuting a conjecture and supporting another
- While modelling EFPAs, we developed a model of cycle notation in CP
  - Shows potential (as a modelling pattern), achieving powerful pruning
  - However, slow in its current incarnation

EFPAs

- Equidistant Frequency Permutation Arrays
- A set of codewords such that:
  - each pair is Hamming distance *d* apart;
  - Each symbol 1..*q* appears *λ* times in each codeword.

c1	1	1	2	2	3	3
c2	1	2	1	3	2	3
c3	1	2	3	1	3	2
c4	1	3	2	3	1	2
c5	1	3	3	2	2	1

EFPAs

- q=3, d=4, λ=2
- 5 codewords: v=5

4 differences

2 of each symbol in each codeword

c1	1	1	2	2	3	3
c2	1	2	1	3	2	3
c3	1	2	3	1	3	2
c4	1	3	2	3	1	2
c5	1	3	3	2	2	1

EFPAs

- Of theoretical interest (recent paper in Designs, Codes and Cryptography journal by Sophie Huczynska)
- We supported this work by generating various maximal size EFPAs
  - Refuted a conjecture that EFPAs always have a full column of 1s when $d=q\lambda-\lambda$
  - Provided empirical evidence for the conjecture that particular constructions are maximal

EFPAs

- This theoretical work may apply to powerline communications
  - Each symbol 1..*q* corresponds to a frequency
  - Codewords are sent by transmitting the symbols in the codeword one by one
  - Robust against different types of noise

Powerline Communications

Overlay the symbol frequencies on top of the power transmission

Signal received with symbols missing, extra frequencies added

Powerline Communications: Impulse Noise

- Someone switches on the kettle – POP
- For example, takes out 3 symbols while transmitting c1
- Receiver can still identify c1 (with any 3 symbols missing)

c1	1	~~1~~	~~2~~	~~2~~	3	3
c2	1	2	1	3	2	3
c3	1	2	3	1	3	2
c4	1	3	2	3	1	2
c5	1	3	3	2	2	1

Powerline Communications: Narrow Band Noise

- Some appliances make continuous noise in a narrow frequency range
- For example, adds 1 and 2 everywhere
- Receiver can still distinguish codewords

c1	1,2	1,2	1,2	1,2	1,2,3	1,2,3
c2	1,2	1,2	1,2	1,2,3	1,2	1,2,3
c3	1,2	1,2	1,2,3	1,2	1,2,3	1,2
c4	1,2	1,2,3	1,2	1,2,3	1,2	1,2
c5	1,2	1,2,3	1,2,3	1,2	1,2	1,2

Example follows Han Vinck, *Coded Modulation*, AEU J., 2000

Modelling EFPAs: 1: non-Boolean model

- First model – codewords represented as a sequence of $q\lambda$ non-Boolean variables

c1	{1...q}	{1...q}	....
c2	{1...q}	{1...q}	....
.....			

Modelling EFPAs: 1: non-Boolean model

x1	x2	x3	x4	x5	x6
y1	y2	y3	y4	y5	y6
...					

- For each row, a cardinality constraint ensures that each symbol occurs λ times:

  gcc([x1,..,x6], [1,..,*q*], [λ,..,λ])
- *d* differences between each pair of rows:

  for each i: ri↔(xi≠yi)

  r1+r2+...+r6=*d*

  (where a Boolean variable (e.g. ri) has domain {0,1} and 0=*false* and 1=*true*)

Modelling EFPAs: 1: non-Boolean model

- Symmetry-breaking by lexicographically ordering adjacent rows *e.g.*

  [x1,..,x6] ≤lex [y1,..,y6]
- Same for adjacent columns

  [x1,y1,z1] ≤lex [x2,y2,z2]

x1	x2	x3	x4	x5	x6
y1	y2	y3	y4	y5	y6
z1	z2	z3	z4	z5	z6

Modelling EFPAs: Boolean and Channelled models

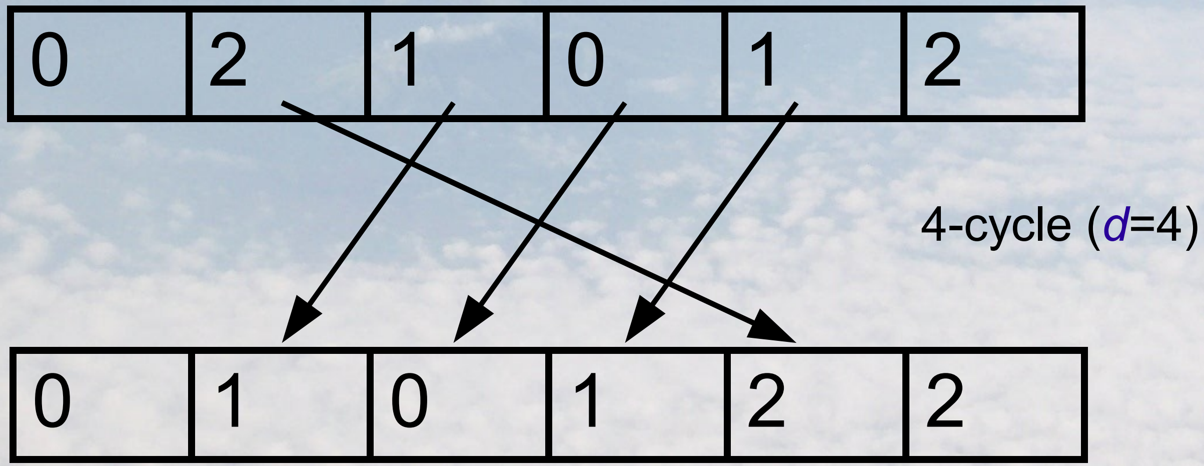
- Boolean model has stronger symmetry-breaking constraints, poor cardinality constraints
- Channelled model combines non-Boolean and Boolean models
- Details in the paper

Modelling EFPAs

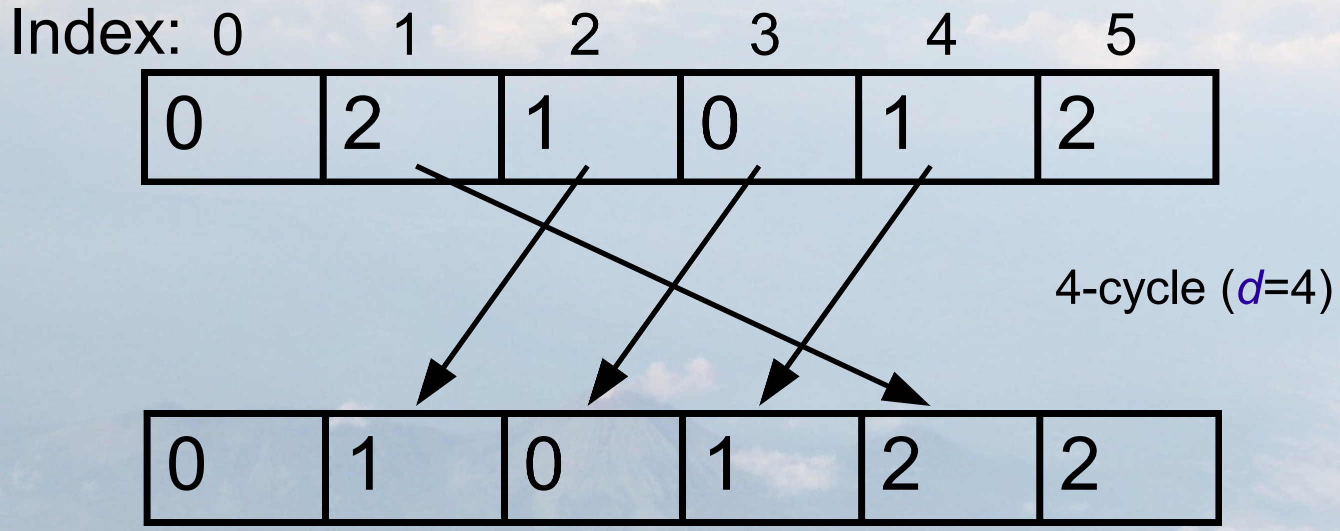
- Now we present two models which extend the non-Boolean model with implied constraint sets
  - **Permutation** (model 2), modelling the permutation between each pair of codewords
  - **Implied** (model 3), exploiting the fact that the first codeword is fixed by symmetry-breaking.

Modelling EFPAs: 2: Permutations

- Modelling permutations
  - Each codeword can be mapped to any other using a permutation with *d* move-points
  - We represent the permutation explicitly

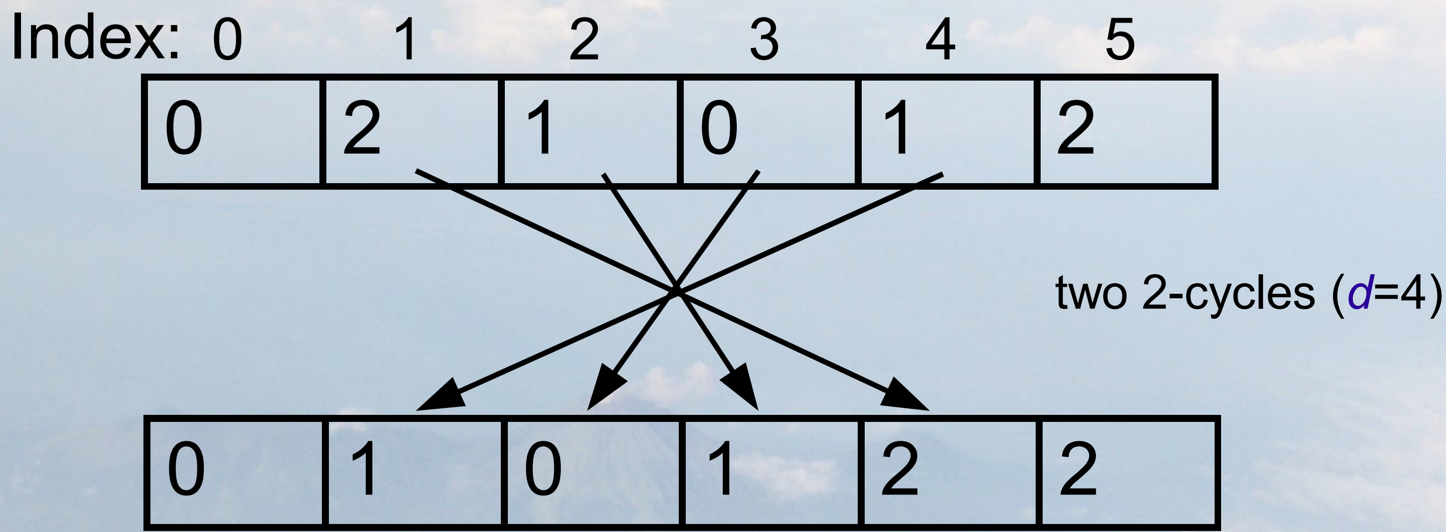


Modelling EFPAs: 2: Permutations



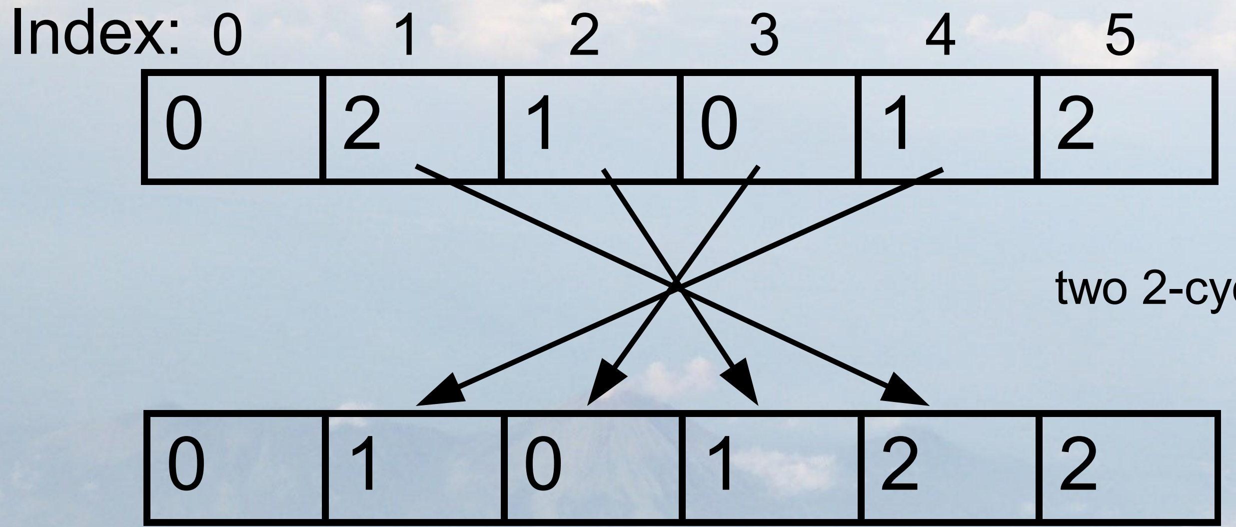
- Represent *cycle notation* in CP
- When d=4, there are two forms of cycle notation:
  - 4-cycle, *e.g.* (1,4,3,2) as shown above
  - two 2-cycles *e.g.* (1,3)(2,5)

Modelling EFPAs: 2: Permutations



- Symmetries arise in cycle notation
- The 4-cycle (1,4,3,2) (on previous slide) is equivalent to (1,4)(2,3) shown above
- (1,4)(2,3) is equivalent to (2,3)(1,4)
- (4,3,2,1) is equivalent to (1,4,3,2)

Modelling EFPAs: 2: Permutations



two 2-cycles (*d*=4)

- Smallest element first in each cycle
- Order cycles by first element
- 4-cycle may only permute distinct symbols (reified allDifferent)

Modelling EFPAs: 2: Permutations

- Somewhat complicated, only implemented for *d*=4
  - *perm* contains the indices to be permuted
  - *cform* is the form of the cycle notation – 0 for 4-cycle, 1 for 2-cycles

perm:	1	4	3	2

cform:	1

Which means: (1,4)(3,2)

Modelling EFPAs: 2: Permutations

- Example (q=3, d=4, λ=3), SAC at root node
- Plain non-Boolean model (first two rows):

1	1	1	2	2	2	3	3	3	4	4	4
1..3	1..4	1..4	1..3	1..4	2..4	1..3	1..4	2..4	1..4	1..4	2..4

- 6 values pruned but nothing assigned

Modelling EFPAs: 2: Permutations

- Example (q=3, d=4, λ=3), SAC at root node
- Permutation model:

1	1	1	2	2	2	3	3	3	4	4	4
1	1..4	1..4	1..3	1..4	2..4	1..3	1..4	2..4	1..4	1..4	4

- An extra 4 values pruned, assigning the first and last variables

Modelling EFPAs: 2: Permutations

- Example (q=3, d=4, λ=3), during search
- Permutation model:

1	1	1	2	2	2	3	3	3	4	4	4
1	1	1	2	2	2	3	4	4	3,4	3,4	3,4

Search decisions

Assigned by perm

- Any permutation must move both remaining 3s

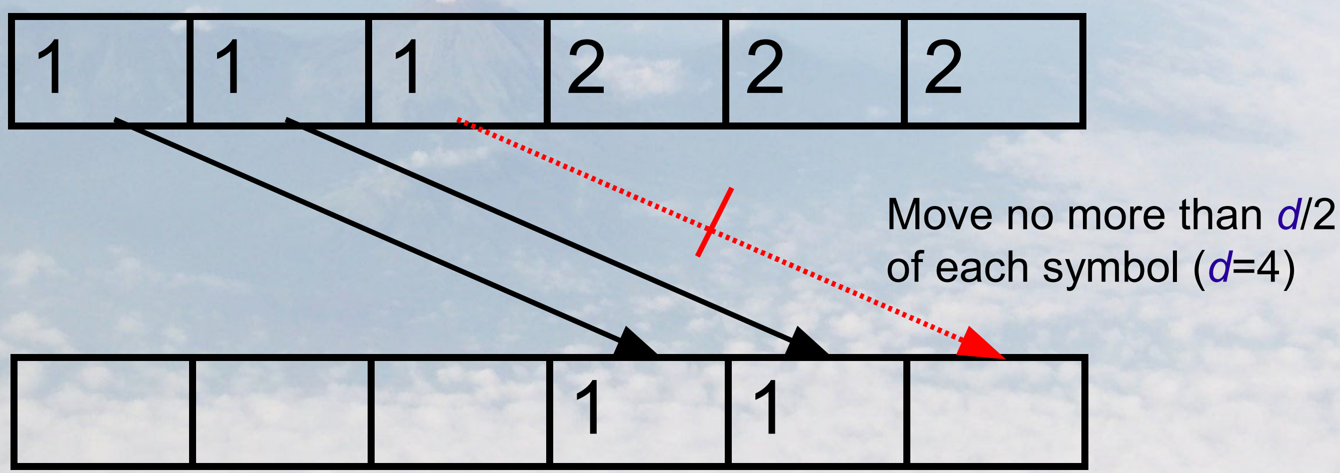
Modelling EFPAs: 3: Implied Constraints

- The first codeword is fixed to: 1,..,1,2,..,2,3,... by column lex ordering constraints
- [x1,y1,z1] ≤lex [x2,y2,z2] implies x1≤x2, and the same applies to every pair of adjacent columns
- The only codeword satisfying x1≤x2≤x3≤... is 1,..,1,2,..,2,3,...

x1	x2	x3	x4	x5	x6
y1	y2	y3	y4	y5	y6
z1	z2	z3	z4	z5	z6

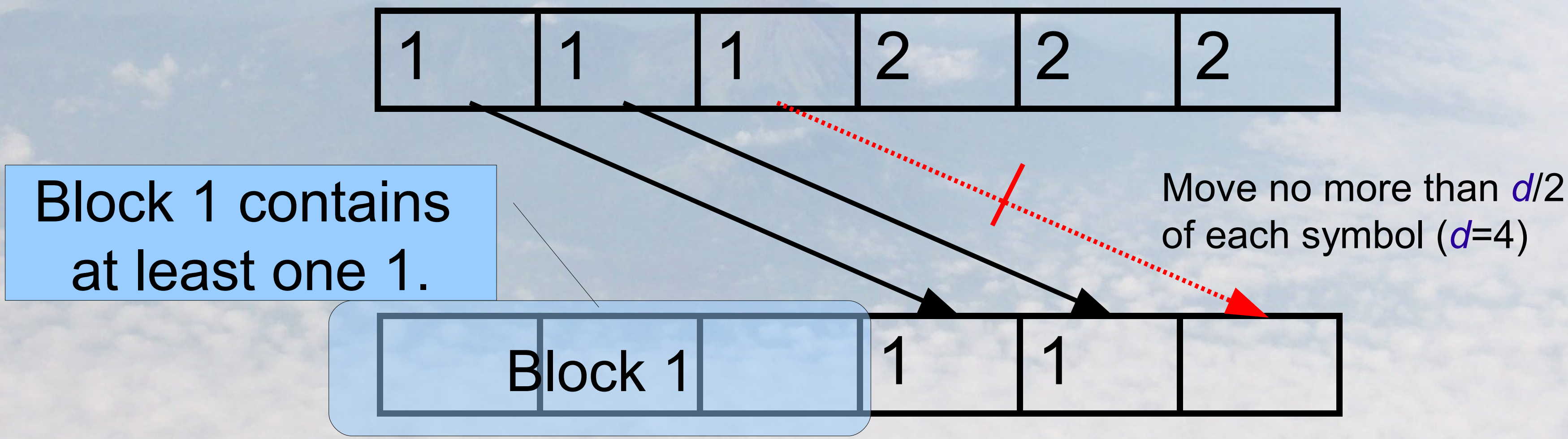
Modelling EFPAs:
3: Implied Constraints

- If more than floor(*d*/2) of any symbol are moved, violates Hamming distance constraint



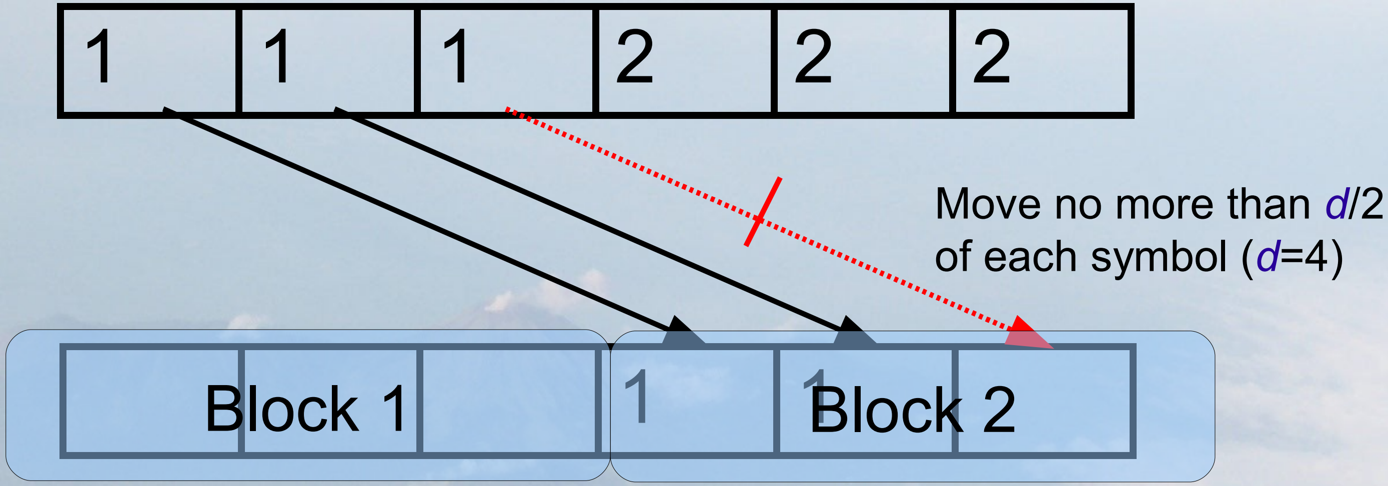
Modelling EFPAs:
3: Implied Constraints

- If more than floor(d/2) of any symbol are moved, violates Hamming distance constraint



- New constraint: Block i has at least λ-floor(d/2) occurrences of i.

Modelling EFPAs: 3: Implied Constraints



- Count occurrences of symbols in blocks using GCC constraints

Tools

- We used the Tailor modelling assistant
  - Translates from Essence' modelling language to Minion input language
  - Provides a small performance improvement by eliminating common subexpressions
- The Minion constraint solver was used

Experiments

- Optimization problem: find largest set of codewords for parameters q, d, λ
- Models all have size parameter v
- We use pairs of values for v, largest satisfiable instance and smallest unsat/unknown
- 24 EFPA instances, 12 satisfiable, 11 unsat, 1 unknown

Experiment 1: non-Boolean, Boolean and Channelled

- Channelled dominates Boolean in both search nodes and time
  - GCC constraint on codewords is valuable
- Non-Boolean and Channelled
  - Neither dominates the other
  - They have different variable/value ordering

Experiment 2: non-Boolean, Perm, Implied

- All based on the non-Boolean model, different sets of additional constraints
- Same variable/value ordering for all
- Implied improves on non-Boolean in most cases, but not dramatically
  - e.g. instance 4-4-4-9, 100s non-Boolean, 80s Implied

Experiment 2: non-Boolean, Perm, Implied

- Perm gives a big improvement in search nodes, but *worse* in solve time
  - Overhead of the extra constraints is too high
  - May have potential (as a modelling pattern) if this issue can be solved

Conclusions

- We used CP to contribute to work in theoretical mathematics
  - Directly refuting a conjecture and supporting another
- While modelling EFPAs, we developed a model of cycle notation in CP
  - Shows potential (as a modelling pattern), achieving powerful pruning
  - However, slow in its current incarnation

Thank You

- Any questions?