Fixed-size Minimax for Committee Elections: Approximation and Local Search Heuristics

COMSOC '06

6 December 2006

Rob LeGrand

Washington University in St. Louis

legrand@cse.wustl.edu

Evangelos Markakis

University of Toronto

vangelis@cs.toronto.edu

Aranyak Mehta

IBM Almaden Research Center

mehtaa@us.ibm.com

Electing a committee from approval ballots

n = 5 candidates

m = 6 ballots

11110

00011 ← approves of candidates 4 and 5

01111

00111

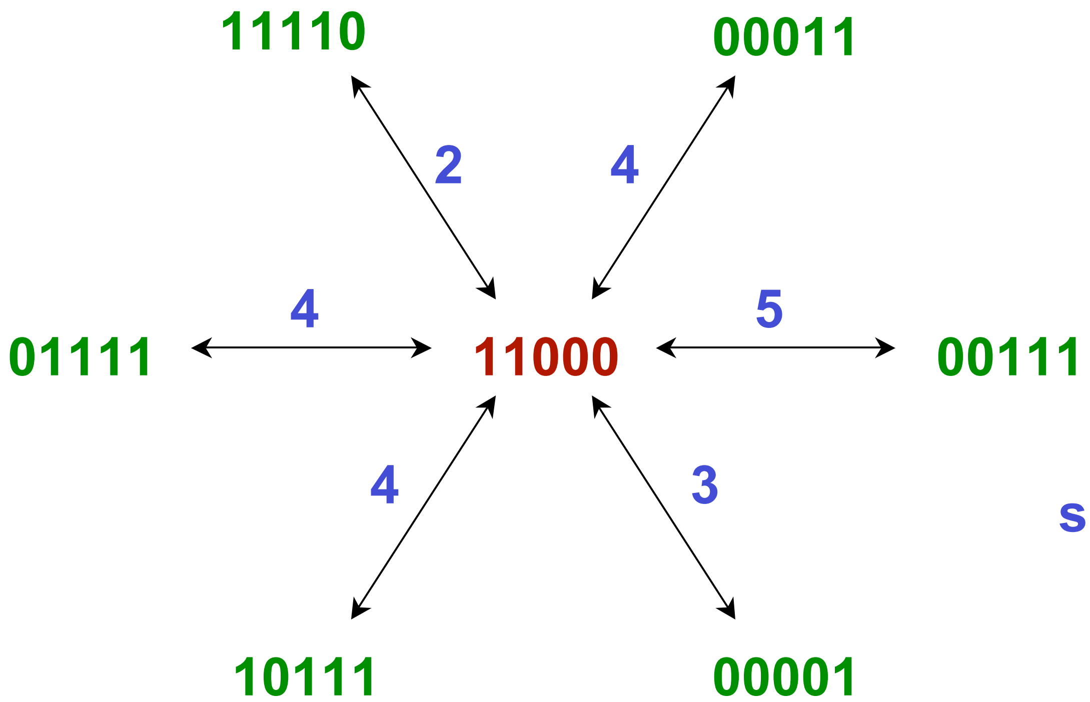
10111

00001

• What's the best committee of size k = 2?

Sum of Hamming distances

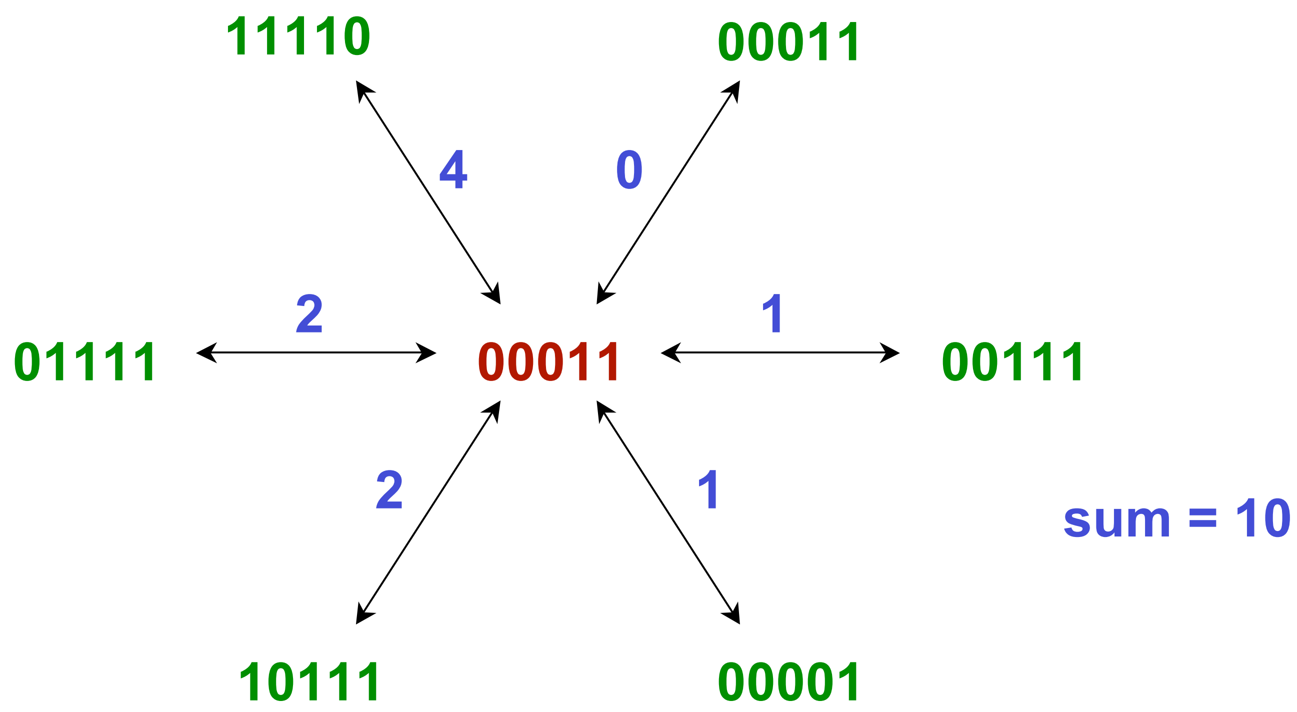
k = 2 winners



sum = 22

Fixed-size minisum

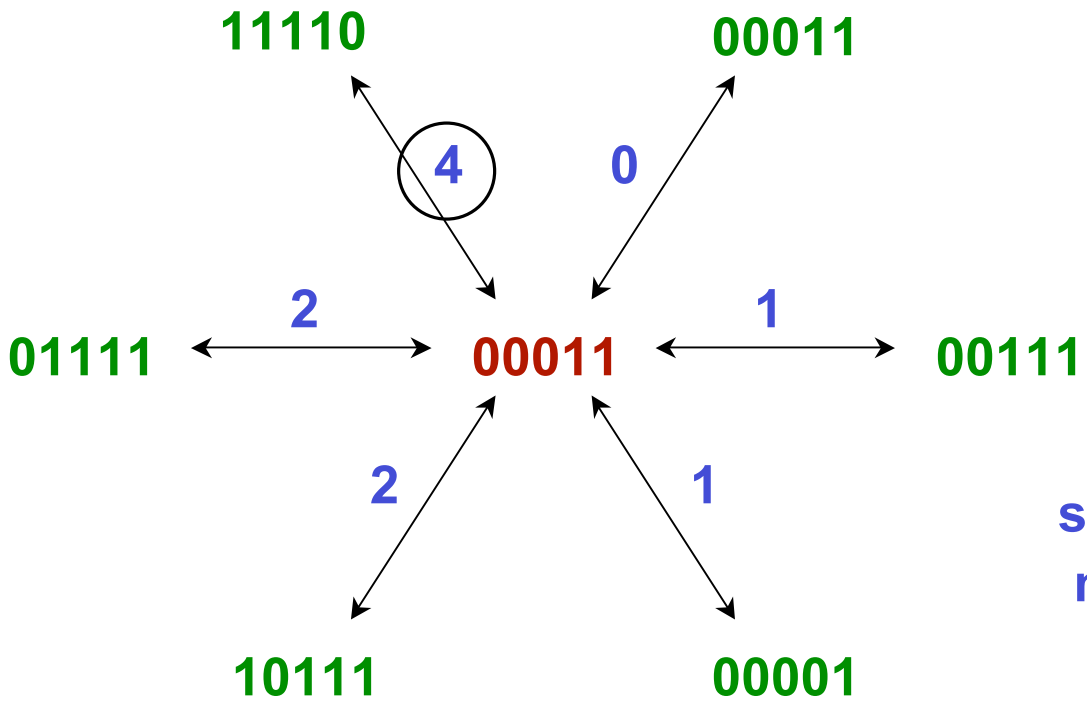
k = 2 winners



- Minisum elects winner set with smallest sumscore
- Easy to compute (pick candidates with most approvals)

Maximum Hamming distance

k = 2 winners

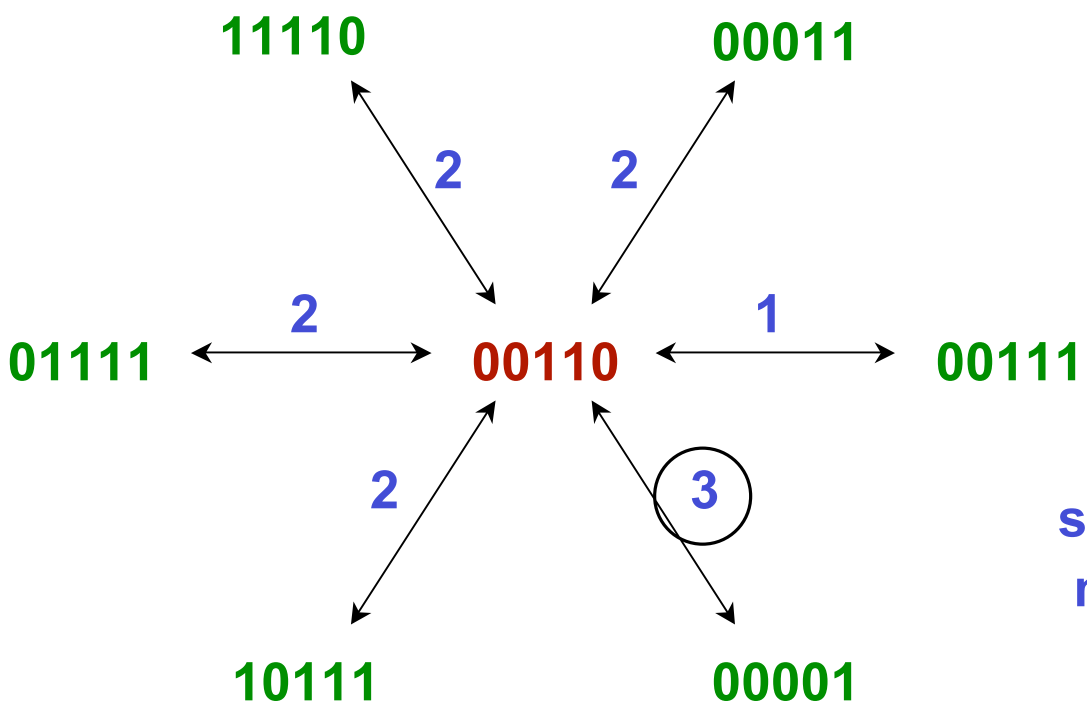


sum = 10
max = 4

Fixed-size minimax

[Brams, Kilgour & Sanver, '04]

k = 2 winners



sum = 12
max = 3

- Minimax elects winner set with smallest maxscore
- Harder to compute?

Complexity

Endogenous minimax = EM = BSM(0, n)	Bounded-size minimax = BSM(k_1, k_2)	Fixed-size minimax = FSM(k) = BSM(k, k)
NP-hard [Frances & Litman, '97]	NP-hard (generalization of EM)	?

Complexity

Endogenous minimax = EM = BSM(0, n)	Bounded-size minimax = BSM(k_1, k_2)	Fixed-size minimax = FSM(k) = BSM(k, k)
NP-hard [Frances & Litman, '97]	NP-hard (generalization of EM)	NP-hard (this paper)

Approximability

Endogenous minimax = EM = BSM(0, n)	Bounded-size minimax = BSM(k_1, k_2)	Fixed-size minimax = FSM(k) = BSM(k, k)
has a PTAS* [Li, Ma & Wang, '99]	no known PTAS; no known constant-factor approx.	no known PTAS; no known constant-factor approx.

* Polynomial-Time Approximation Scheme: algorithm with approx. ratio 1 + ε that runs in time polynomial in the input and exponential in 1/ε

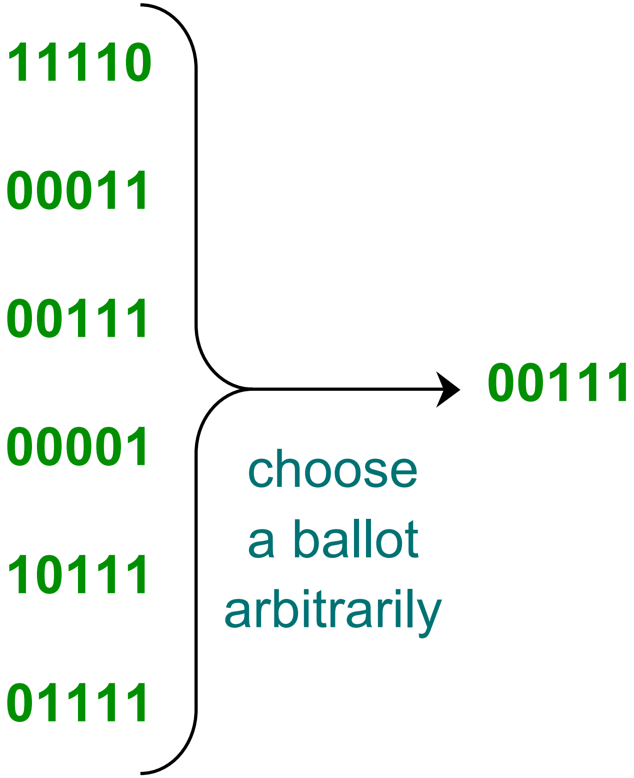
Approximability

Endogenous minimax = EM = BSM(0, n)	Bounded-size minimax = BSM(k_1, k_2)	Fixed-size minimax = FSM(k) = BSM(k, k)
has a PTAS* [Li, Ma & Wang, '99]	no known PTAS; has a 3-approx. (this paper)	no known PTAS; has a 3-approx. (this paper)

* Polynomial-Time Approximation Scheme: algorithm with approx. ratio 1 + ε that runs in time polynomial in the input and exponential in 1/ε

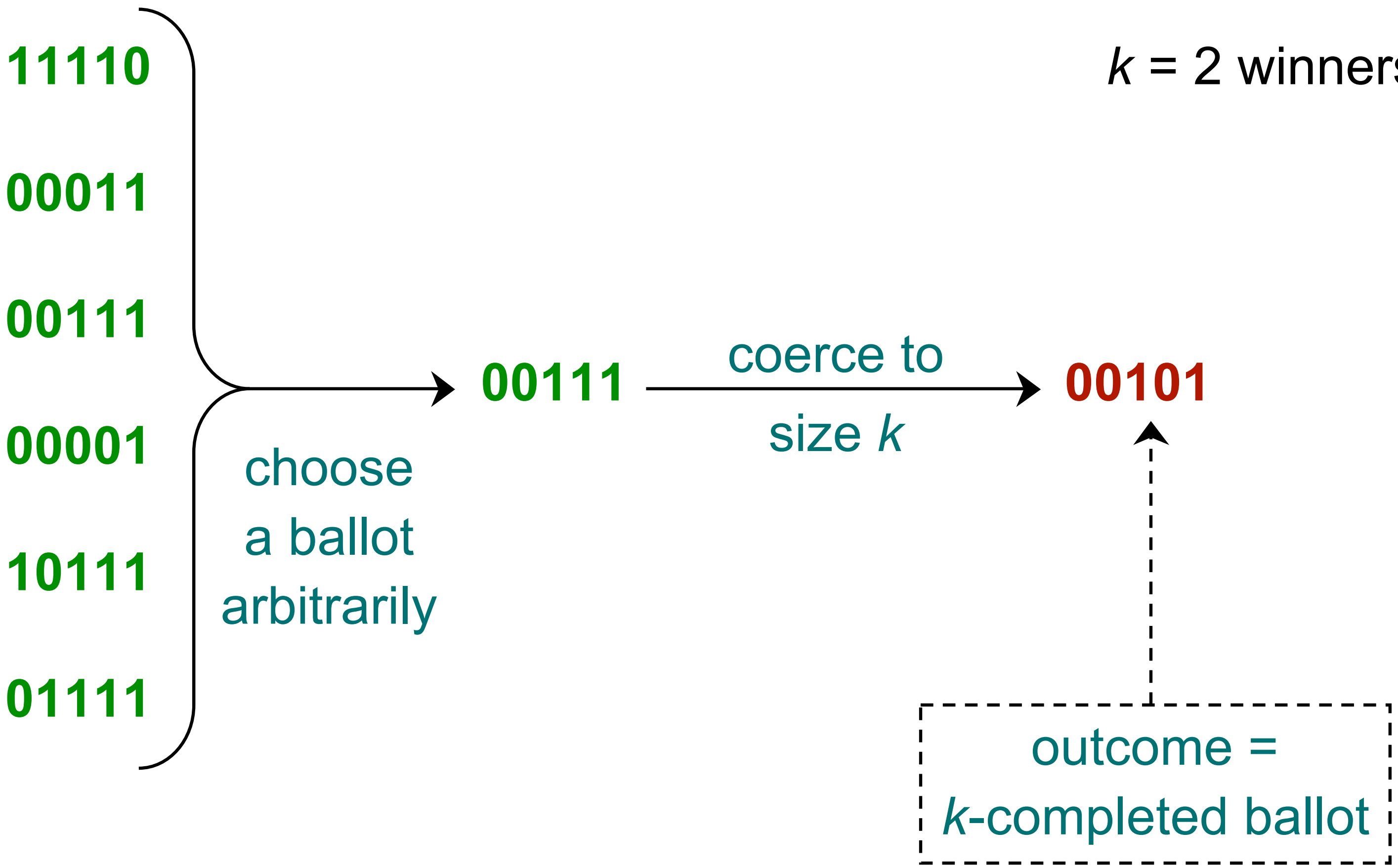
Approximating FSM

k = 2 winners

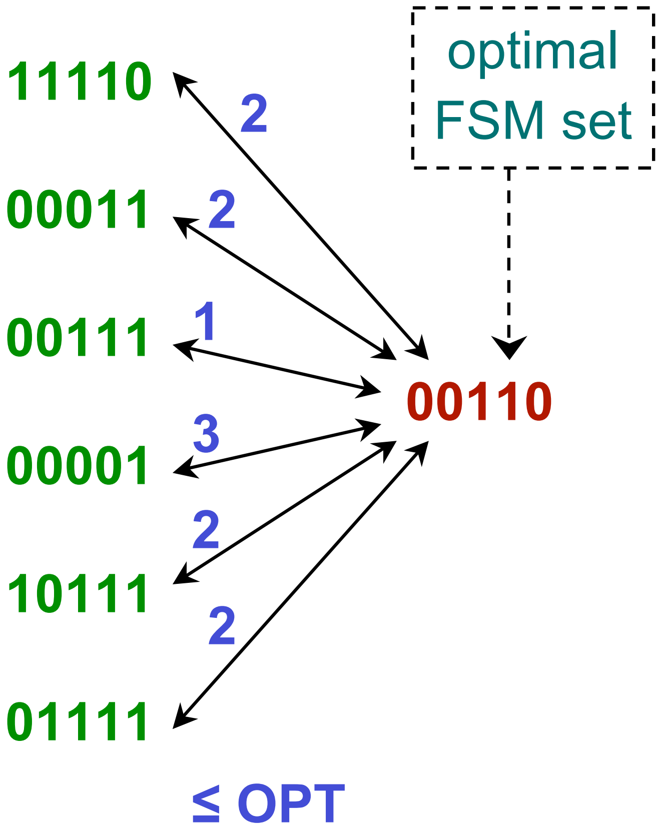


Approximating FSM

k = 2 winners

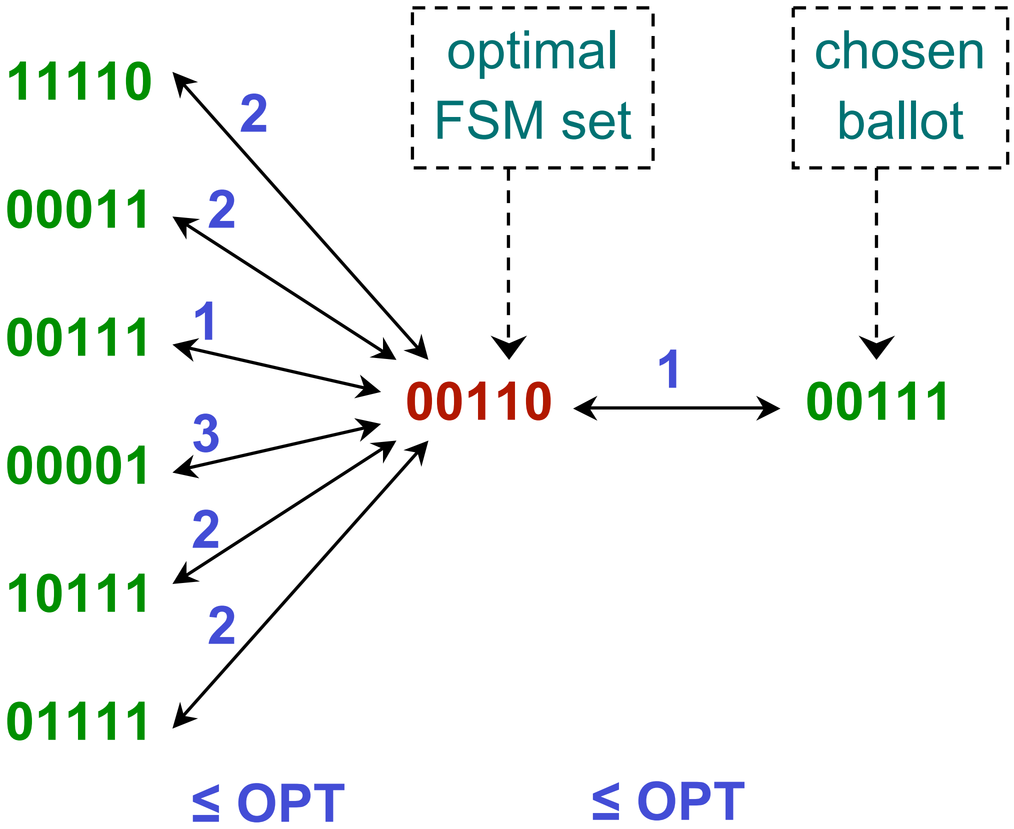


Approximation ratio ≤ 3



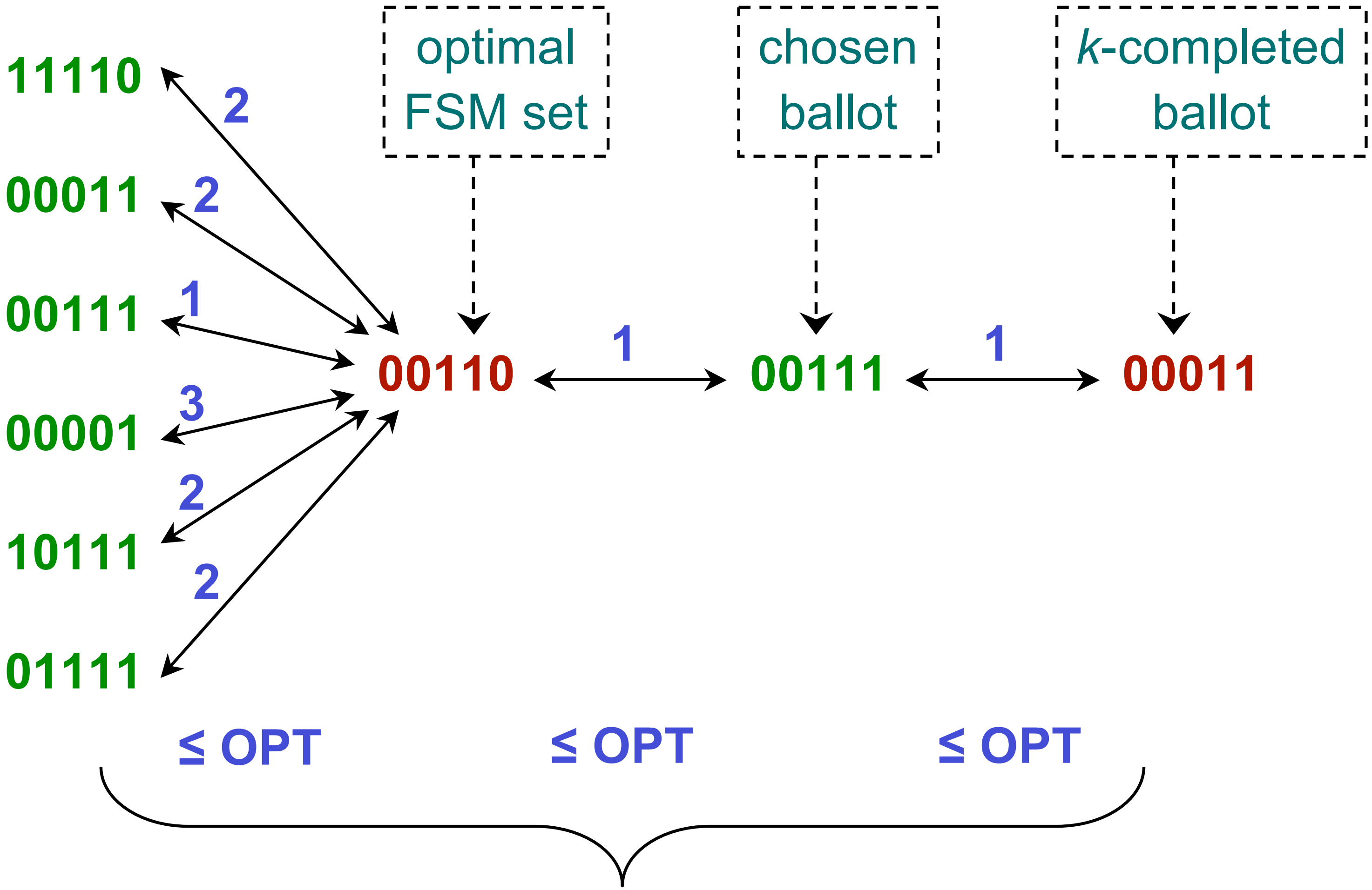
OPT = optimal maxscore

Approximation ratio ≤ 3



OPT = optimal maxscore

Approximation ratio ≤ 3



OPT = optimal maxscore

≤ 3·OPT

(by triangle inequality)

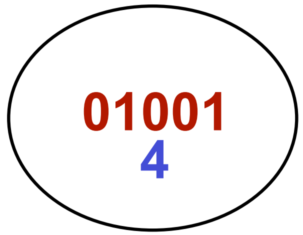
Better in practice?

- So far, we can guarantee a winner set no more than 3 times as bad as the optimal.
  - Nice in theory . . .

- How can we do better in practice?
  - Try local search

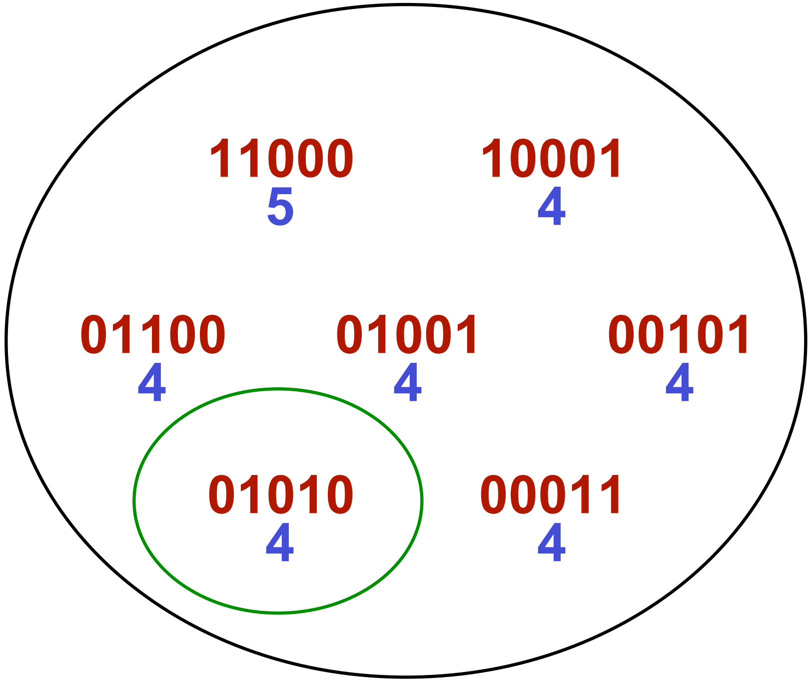
Local search approach for FSM

1. Start with some $c \in \{0,1\}^n$ of weight k



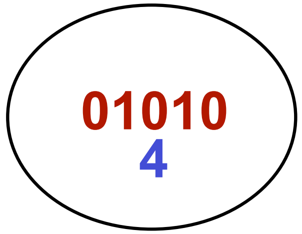
Local search approach for FSM

1. Start with some $c \in \{0,1\}^n$ of weight k
2. In c, swap up to r 0-bits with 1-bits in such a way that minimizes the maxscore of the result



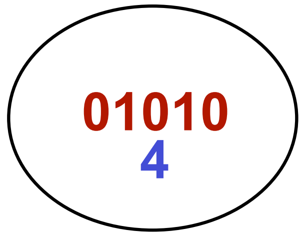
Local search approach for FSM

1. Start with some $c \in \{0,1\}^n$ of weight k
2. In c, swap up to r 0-bits with 1-bits in such a way that minimizes the maxscore of the result



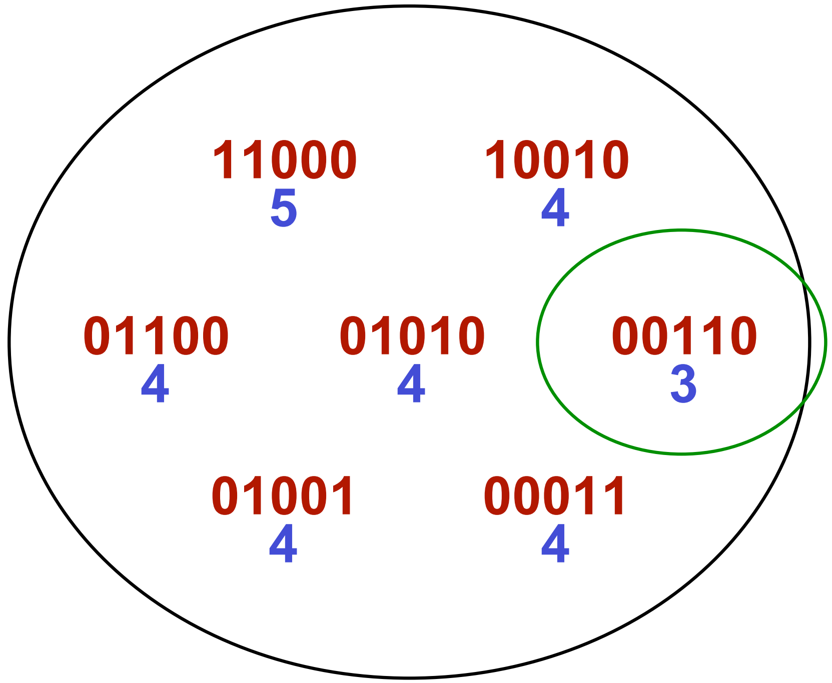
Local search approach for FSM

1. Start with some $c \in \{0,1\}^n$ of weight k
2. In c, swap up to r 0-bits with 1-bits in such a way that minimizes the maxscore of the result



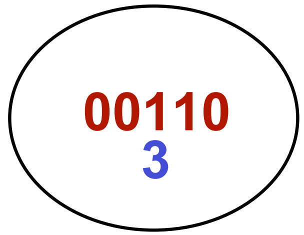
Local search approach for FSM

1. Start with some $c \in \{0,1\}^n$ of weight k
2. In c, swap up to r 0-bits with 1-bits in such a way that minimizes the maxscore of the result
3. Repeat step 2 until maxscore(c) is unchanged n times
4. Take c as the solution



Local search approach for FSM

1. Start with some $c \in \{0,1\}^n$ of weight k
2. In c, swap up to r 0-bits with 1-bits in such a way that minimizes the maxscore of the result
3. Repeat step 2 until maxscore(c) is unchanged n times
4. Take c as the solution



Specific FSM heuristics

- Two parameters:
  - where to start vector *c*:
    1. a fixed-size-minisum solution
    2. a *k*-completion of a ballot (3-approx.)
    3. a random set of *k* candidates
    4. a *k*-completion of a ballot with highest maxscore
  - radius of neighborhood *r*: 1 and 2

Heuristic evaluation

- Real-world ballots from GTS 2003 council election
- Found exact minimax solution
- Ran each heuristic 5000 times
- Compared exact minimax solution with heuristics to find realized approximation ratios
  - example: 15/14 = 1.0714
    - maxscore of solution found = 15
    - maxscore of exact solution = 14

- We also performed experiments using ballots generated according to random distributions (see paper)

Average approx. ratios found

	radius = 1	radius = 2
fixed-size minimax	1.0012	1.0000
3-approx.	1.0017	1.0000
random set	1.0057	1.0000
highest-maxscore	1.0059	1.0000

performance on GTS ’03 election data

n = 24 candidates, k = 12 winners, m = 161 ballots

Largest approx. ratios found

	radius = 1	radius = 2
fixed-size minimax	1.0714	1.0000
3-approx.	1.0714	1.0000
random set	1.0714	1.0000
highest-maxscore	1.0714	1.0000

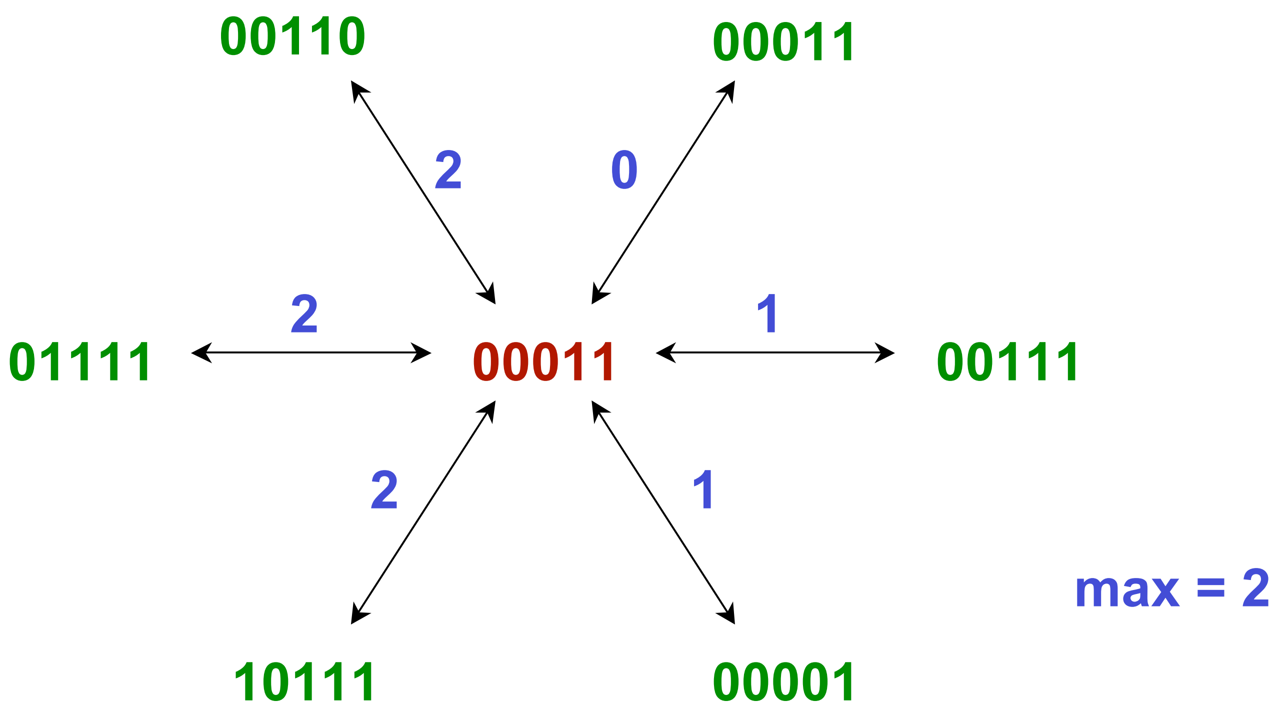
performance on GTS ’03 election data

n = 24 candidates, k = 12 winners, m = 161 ballots

Conclusions from all experiments

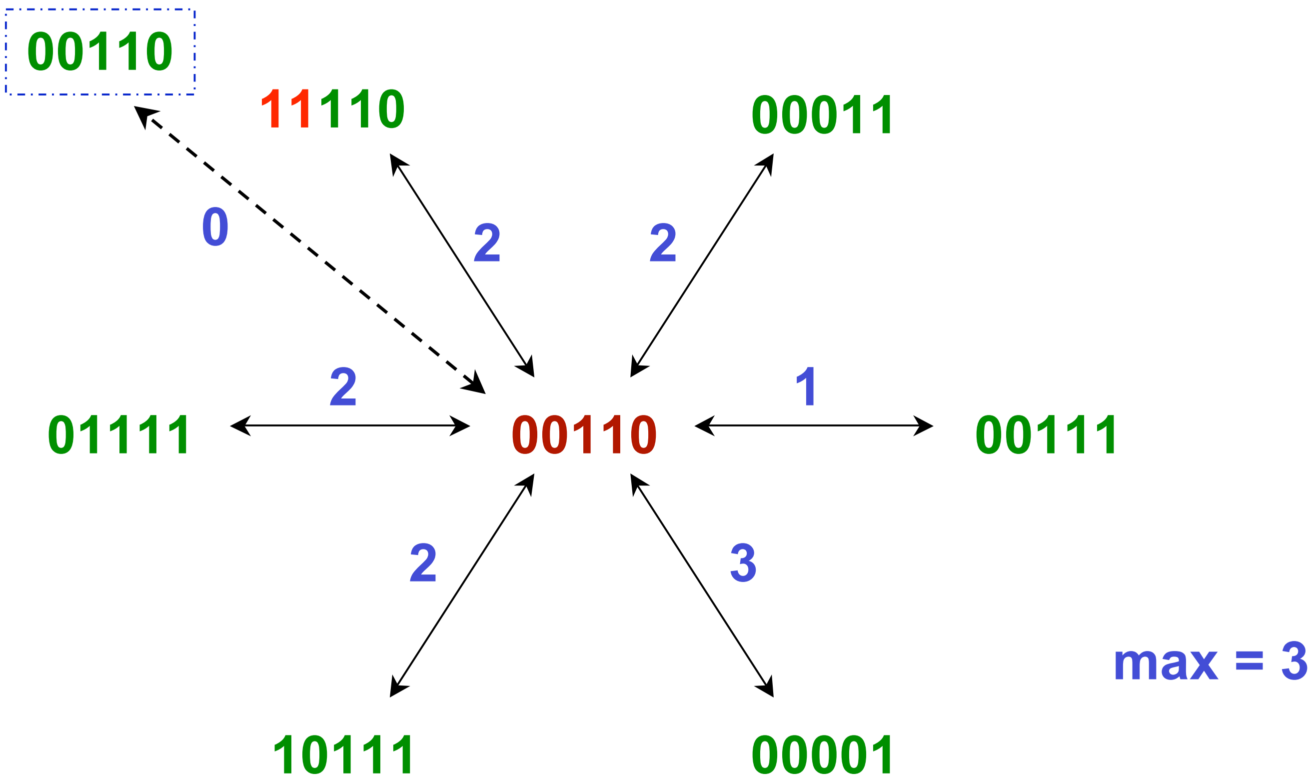
- All heuristics perform near-optimally
  - highest ratio found: 1.2
  - highest average ratio < 1.04
- When radius is larger, performance improves and running time increases
- The fixed-size-minisum starting point performs best overall (with our 3-approx. a close second)

Manipulating FSM



- Voters are sincere
- Another optimal solution: **00101**

Manipulating FSM



•A voter manipulates and realizes ideal outcome

Nonmanipulable "FSM"?

Electing a set found using our 3-approximation for FSM gives a nonmanipulable procedure:

- For the voters whose ballots are *not* chosen, voting insincerely cannot affect the outcome
- For the voter whose ballot *is* chosen, the outcome will be one of the sets of size *k* closest to the voter's wishes

Conclusions

- BSM and FSM are NP-hard
- Both can be approximated with ratio 3
- Polynomial-time local search heuristics perform well in practice
  - some retain ratio-3 guarantee
- Exact FSM can be manipulated
- Our 3-approximation for FSM is nonmanipulable

Future work

- Investigate weighted version of minimax [Brams, Kilgour & Sanver, '06]
- What is the best approximation ratio for FSM achievable in polynomial time? (Is there a PTAS?)
- What is the nonmanipulable FSM approximation algorithm with the best ratio?

Thanks!