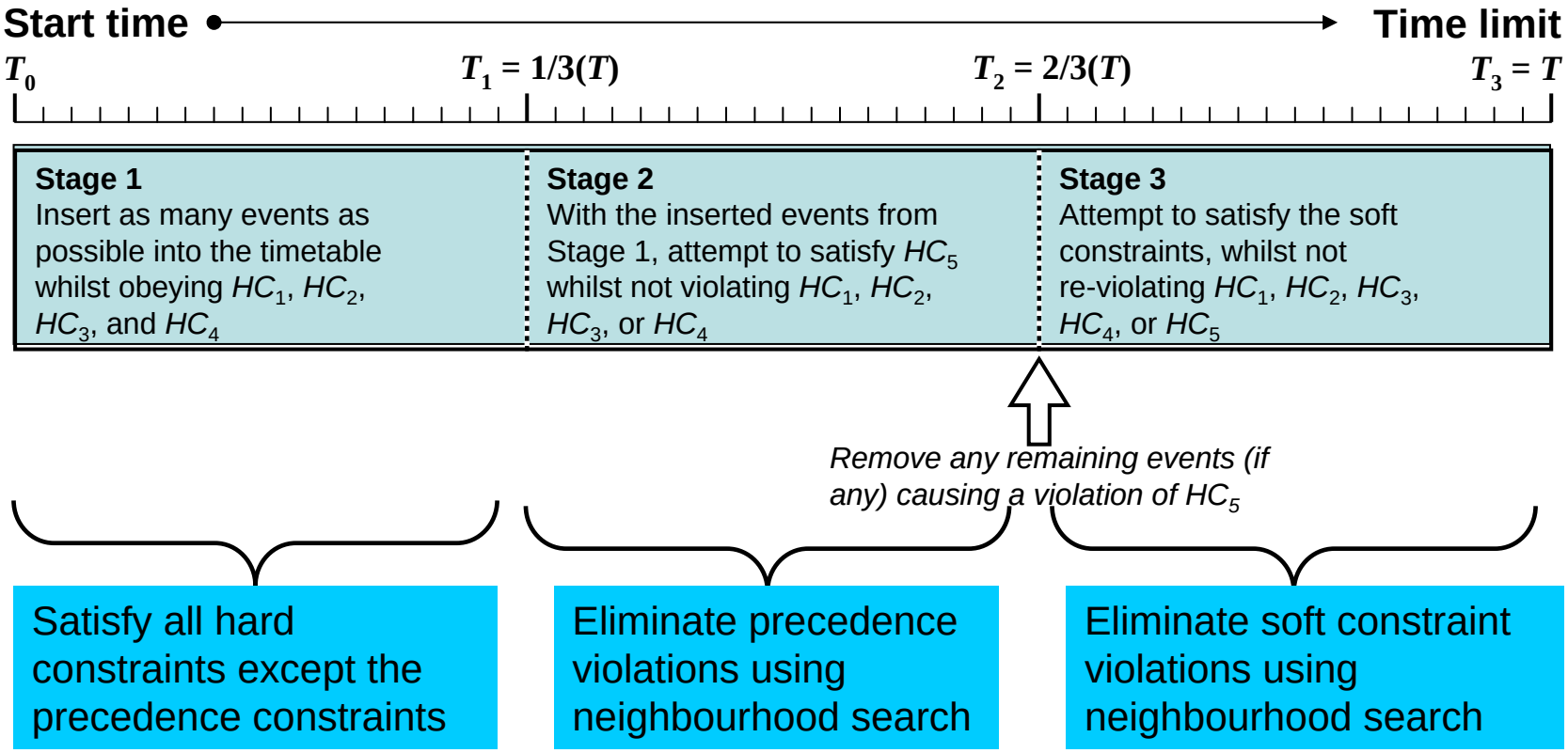
A Time-Dependent Metaheuristic Algorithm for Post Enrolment-based Course timetabling

Rhyd Lewis
Quantitative Methods Research Group
Cardiff Business School
Prifysgol Caerdydd / Cardiff University
Wales

Talk Plan

- Algorithm overview
- Analysis of results gained
- Some comments on ITC2007's method of comparison

Algorithm Overview

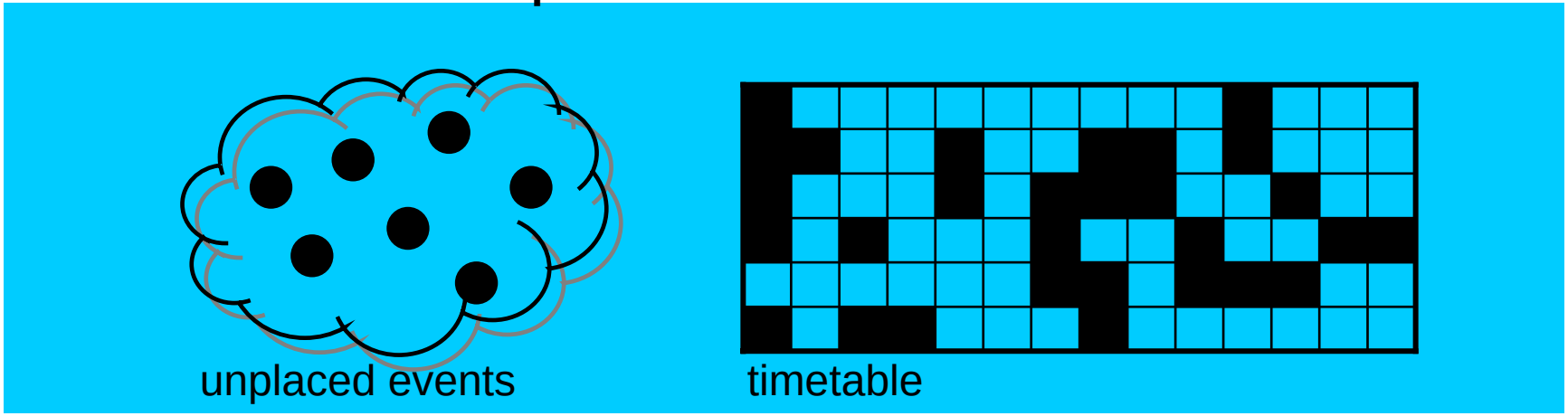


Note: Different time thresholds could have been used in practice (and perhaps should have been…)

Stage 1

- Events are inserted into the timetable using specialised heuristics such that all hard constraints (except the precedence constraints) are obeyed
- Events with no feasible place are left to one side unplaced
- Unplaced events are dealt with via an improvement procedure

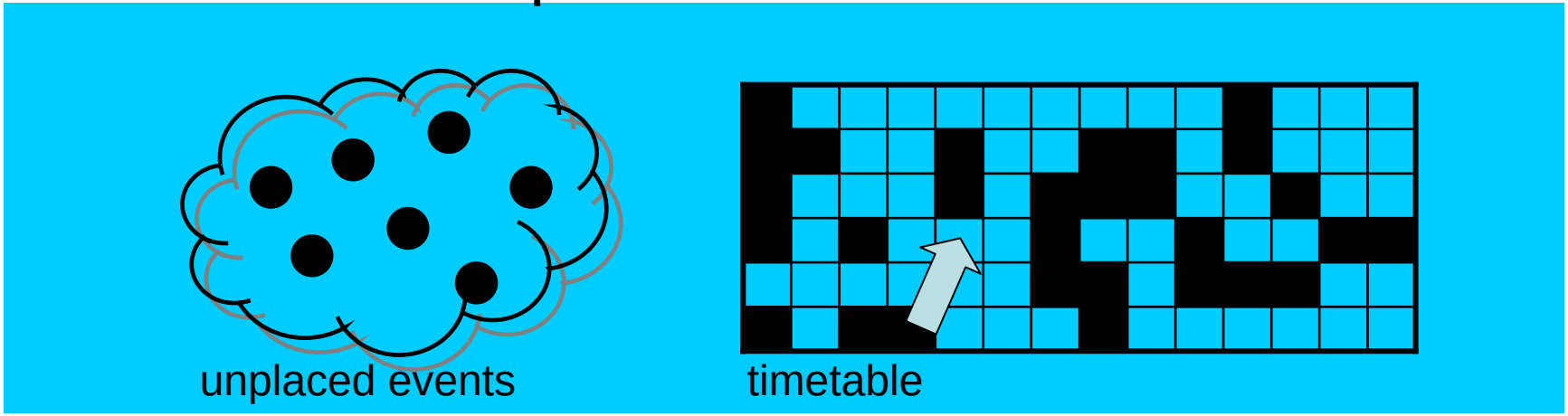
Demonstration of the Improvement Procedure



Stage 1

- Events are inserted into the timetable using specialised heuristics such that all hard constraints (except the precedence constraints) are obeyed
- Events with no feasible place are left to one side unplaced
- Unplaced events are dealt with via an improvement procedure

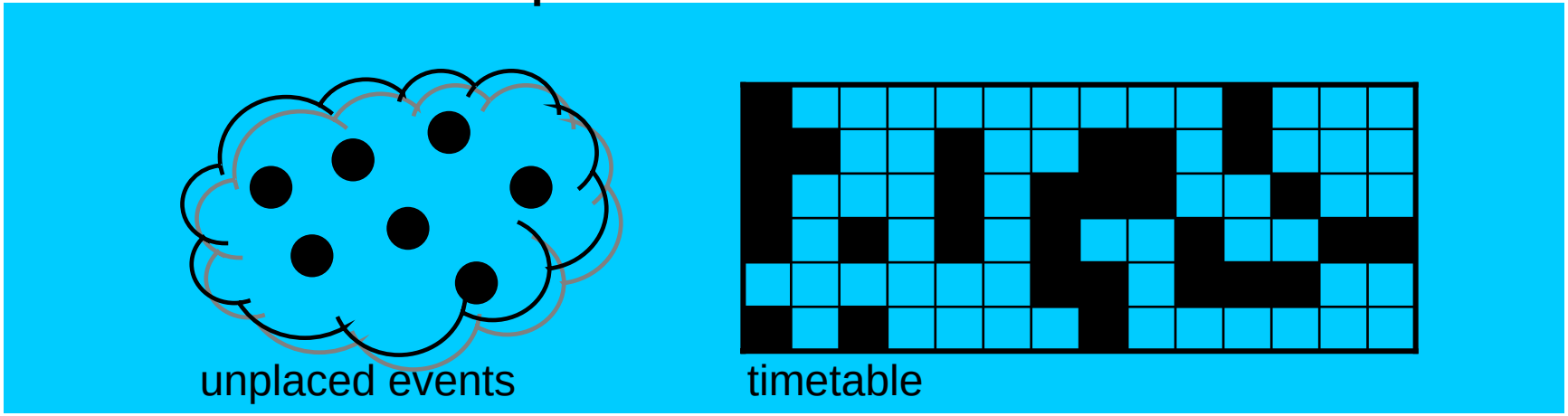
Demonstration of the Improvement Procedure



Stage 1

- Events are inserted into the timetable using specialised heuristics such that all hard constraints (except the precedence constraints) are obeyed
- Events with no feasible place are left to one side unplaced
- Unplaced events are dealt with via an improvement procedure

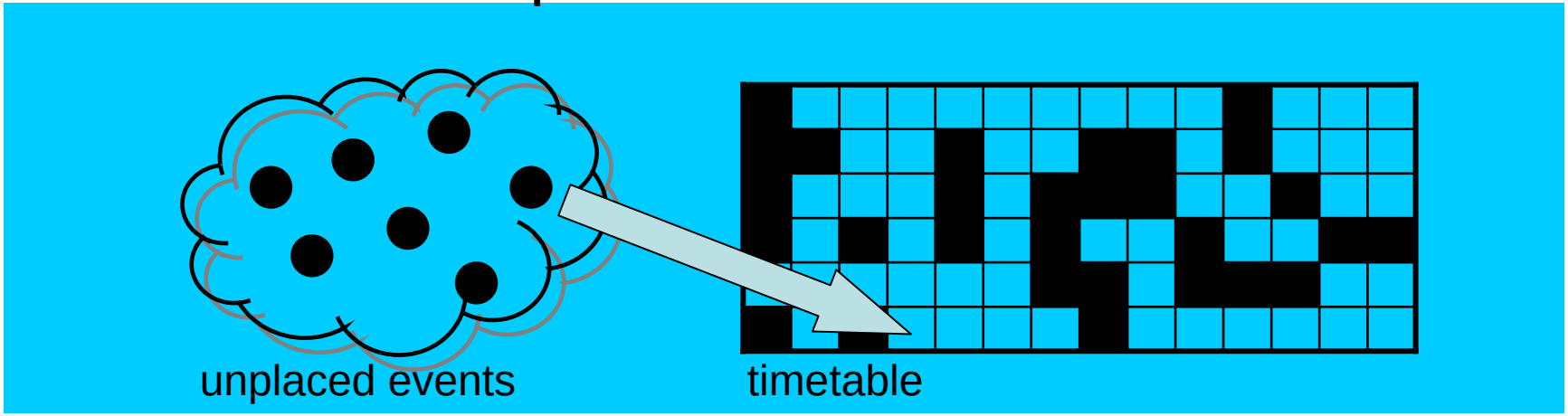
Demonstration of the Improvement Procedure



Stage 1

- Events are inserted into the timetable using specialised heuristics such that all hard constraints (except the precedence constraints) are obeyed
- Events with no feasible place are left to one side unplaced
- Unplaced events are dealt with via an improvement procedure

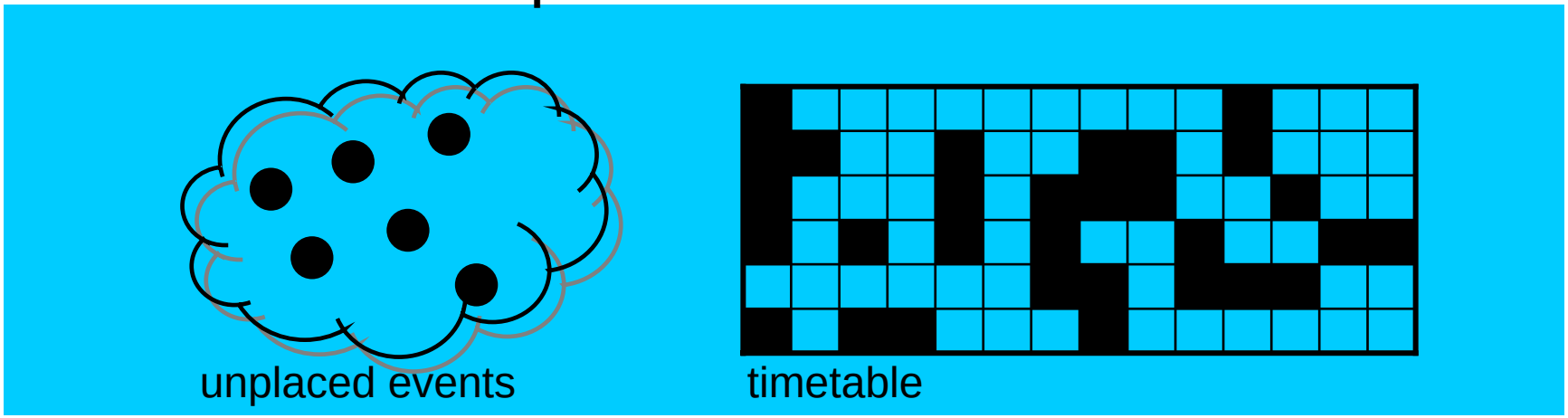
Demonstration of the Improvement Procedure



Stage 1

- Events are inserted into the timetable using specialised heuristics such that all hard constraints (except the precedence constraints) are obeyed
- Events with no feasible place are left to one side unplaced
- Unplaced events are dealt with via an improvement procedure

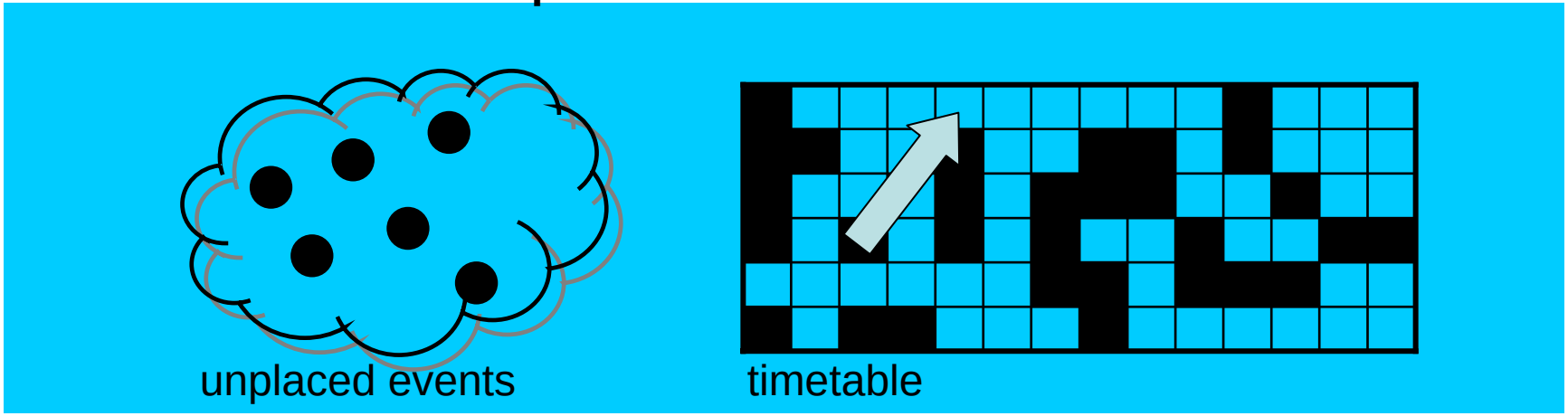
Demonstration of the Improvement Procedure



Stage 1

- Events are inserted into the timetable using specialised heuristics such that all hard constraints (except the precedence constraints) are obeyed
- Events with no feasible place are left to one side unplaced
- Unplaced events are dealt with via an improvement procedure

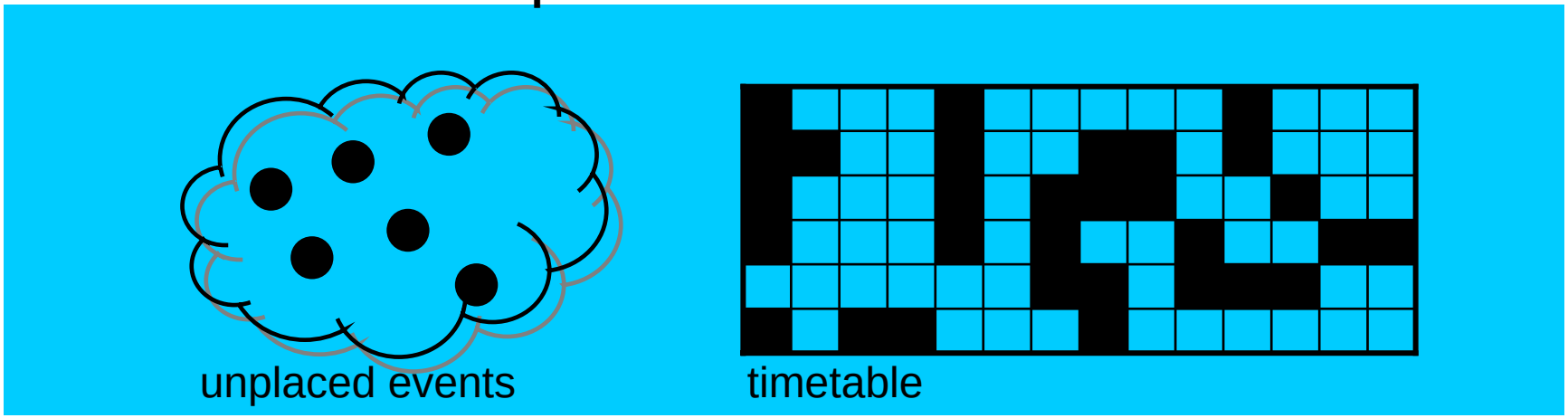
Demonstration of the Improvement Procedure



Stage 1

- Events are inserted into the timetable using specialised heuristics such that all hard constraints (except the precedence constraints) are obeyed
- Events with no feasible place are left to one side unplaced
- Unplaced events are dealt with via an improvement procedure

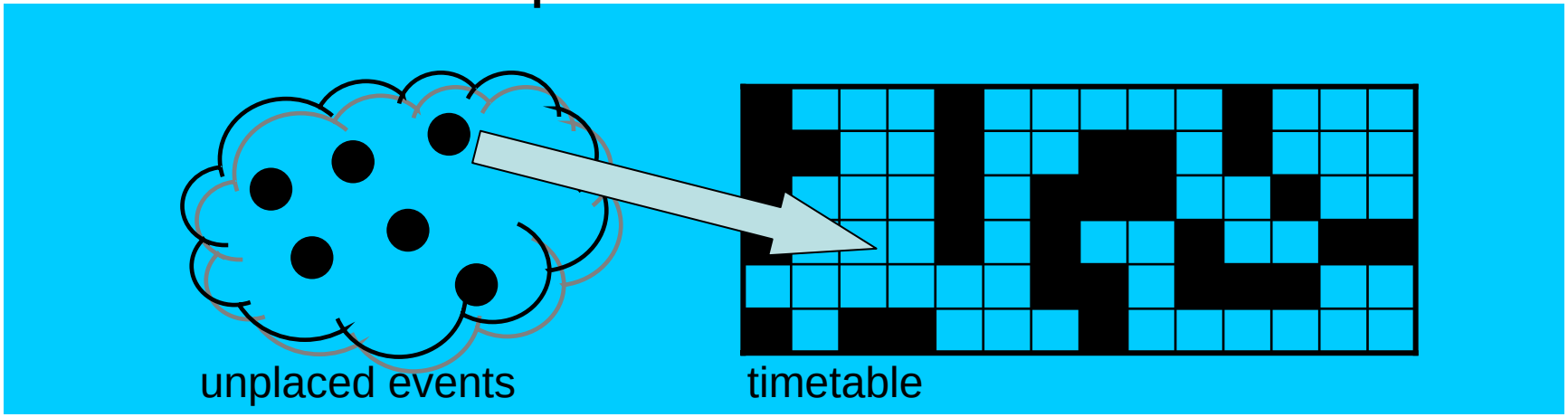
Demonstration of the Improvement Procedure



Stage 1

- Events are inserted into the timetable using specialised heuristics such that all hard constraints (except the precedence constraints) are obeyed
- Events with no feasible place are left to one side unplaced
- Unplaced events are dealt with via an improvement procedure

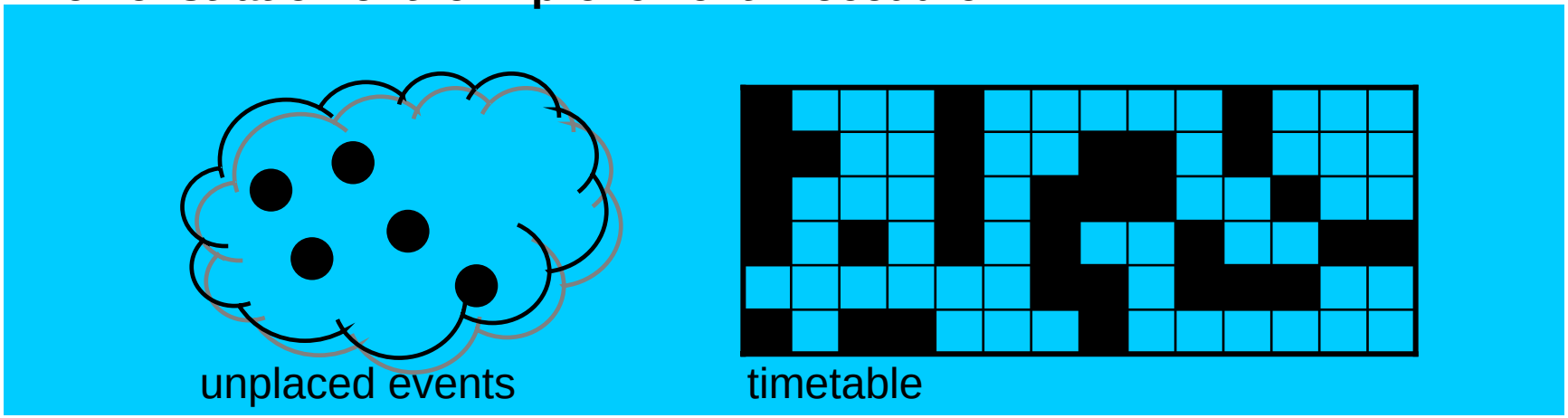
Demonstration of the Improvement Procedure



Stage 1

- Events are inserted into the timetable using specialised heuristics such that all hard constraints (except the precedence constraints) are obeyed
- Events with no feasible place are left to one side unplaced
- Unplaced events are dealt with via an improvement procedure

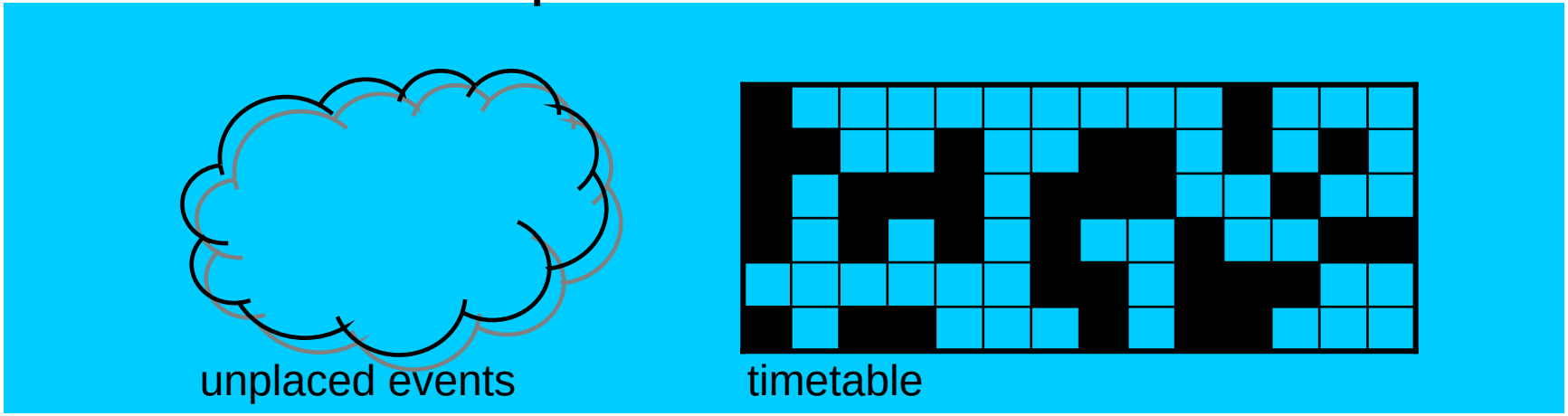
Demonstration of the Improvement Procedure



Stage 1

- Events are inserted into the timetable using specialised heuristics such that all hard constraints (except the precedence constraints) are obeyed
- Events with no feasible place are left to one side unplaced
- **In practice, all events were inserted within the cut-off point of the procedure (with all instances)**

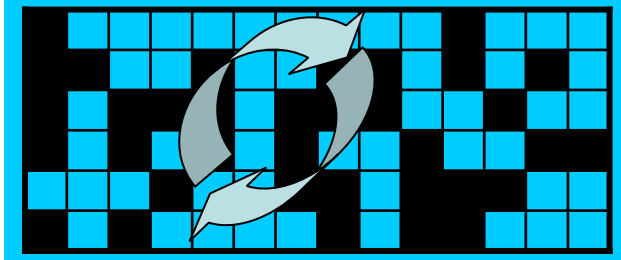
Demonstration of the Improvement Procedure



Stage 2: Eliminating Precedence Constraint Violations

- Achieved using a basic application of simulated annealing
- Neighbourhood operator:

  Select two cells with unequal contents and swap.



- Two actions can occur, either:
  - Two events will be *swapped* OR
  - One event will be *moved*

n: number of events
p: number of places

$$P(\text{swap}) = \frac{n}{p}\left(\frac{n-1}{p-1}\right)$$

$$P(\text{move}) = \frac{p-n}{p} + \frac{n}{p}\left(\frac{p-n}{p-1}\right)$$

- Any proposed neighbourhood move causing a violation of previously satisfied hard constraints is reset immediately

Cooling Schedule Considerations

- Success of Simulated Annealing hinges on an appropriate cooling schedule being used
- Ideally the "cool" should neither be too quick or too slow.
- Let:

  t_0 = initial temperature (determined automatically)

  t_μ = some end temperature (close to zero)

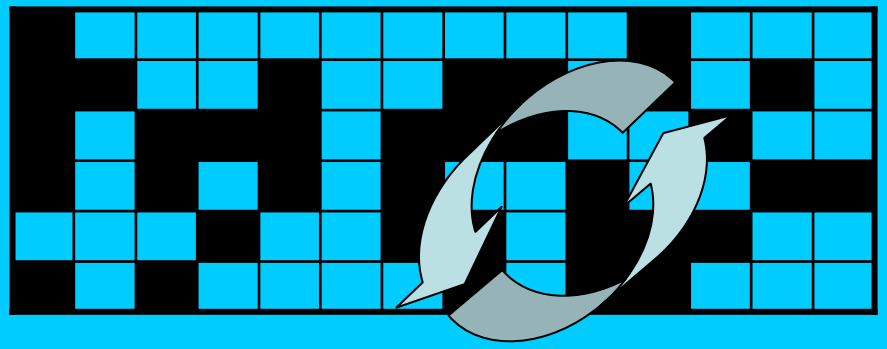
  α = cooling rate, such that $t_{i+1} = \alpha . t_i$

  μ = estimate of the number of fixed-length Markov chains to be completed in the time limit

- Then, for this approach: $\alpha = (t_\mu / t_0)^{1/\mu}$
- Using these conditions, all precedence violations were generally removed from all instances except #1, #2, #9 and #10. These are the "fuller" instances…

Stage 3

- First, any event still causing a violation of (any) hard constraint is removed
- Simulated annealing is then again applied in the same fashion as stage 2, using the *Soft Cost* as a cost function.
- Again, restrictions are added to prohibit violations of any hard constraints



Choose two cells with unequal content and swap.

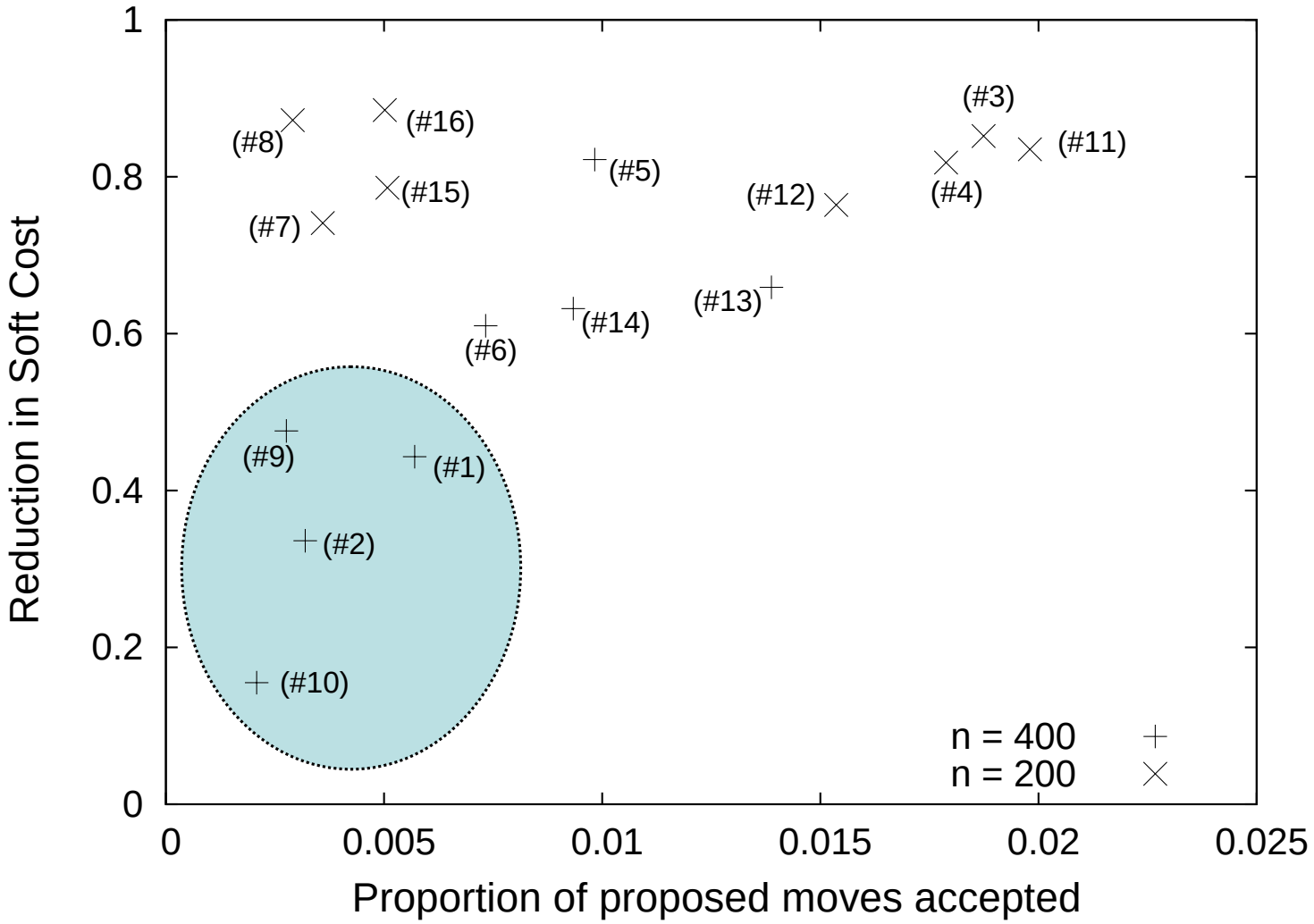
Reject moves causing a violation of the hard constraints

Summary of results from 51 runs on each instance

(Distance To Feasibility and Soft cost (in brackets))

Instance	Best	Q_1	Median	Q_3	Worst
comp-2007-2-1	0 (1294)	0 (1600)	17 (1492)	32 (1693)	105 (1944)
comp-2007-2-2	0 (1599)	18 (1718)	46 (1826)	80 (2016)	213 (2176)
comp-2007-2-3	0 (278)	0 (416)	0 (457)	0 (523)	0 (664)
comp-2007-2-4	0 (388)	0 (538)	0 (589)	0 (644)	0 (761)
comp-2007-2-5	0 (22)	0 (123)	0 (193)	0 (268)	0 (638)
comp-2007-2-6	0 (369)	0 (606)	0 (696)	0 (767)	20 (708)
comp-2007-2-7	0 (74)	0 (300)	0 (421)	0 (529)	0 (890)
comp-2007-2-8	0 (0)	0 (162)	0 (206)	0 (256)	0 (366)
comp-2007-2-9	0 (1582)	59 (1829)	80 (2312)	120 (1864)	214 (1609)
comp-2007-2-10	0 (2380)	83 (2339)	126 (2262)	194 (2303)	372 (2159)
comp-2007-2-11	0 (344)	0 (456)	0 (541)	0 (605)	0 (800)
comp-2007-2-12	0 (486)	0 (660)	0 (741)	0 (852)	125 (710)
comp-2007-2-13	0 (365)	0 (538)	0 (631)	0 (707)	19 (766)
comp-2007-2-14	0 (222)	0 (558)	0 (660)	0 (786)	27 (685)
comp-2007-2-15	0 (266)	0 (301)	0 (344)	0 (366)	0 (455)
comp-2007-2-16	0 (99)	0 (165)	0 (194)	0 (215)	0 (265)

Search space restrictedness Vs Overall reduction in Soft Cost



Ranking and Comparing the Competition Entries

- Fixed time limit was used.
- The competition evaluation criteria make the results a type of *ordinal* data.
- Thus all entries are compared by **ranking** their results for each instance and then averaging
- E.g.

	Results achieved on 5 instances(Distance to feasibility and soft cost (in brackets))					Rank for each instance					Average Rank	Position
Algorithm	#1	#2	#3	#4	#5	#1	#2	#3	#4	#5		
A	5 (3)	0 (0)	20 (34)	0 (0)	0 (0)	2	2	3	2.5	2	2.3	2nd
B	0 (10)	0 (0)	0 (49)	0 (0)	0 (0)	1	2	1	2.5	2	1.7	1st
C	5 (1023)	0 (89)	0 (101)	0 (0)	0 (34)	3	4	2	2.5	4	3.1	4th
D	10 (102)	0 (0)	35 (200)	0 (0)	0 (0)	4	2	4	2.5	2	2.9	3rd

- For added fairness, in the final the top 5 algorithms were compared by looking at results gained from ten runs *on our own machines.*
- Hidden instances were also used.
- **We were not just interested in the best run!!!**

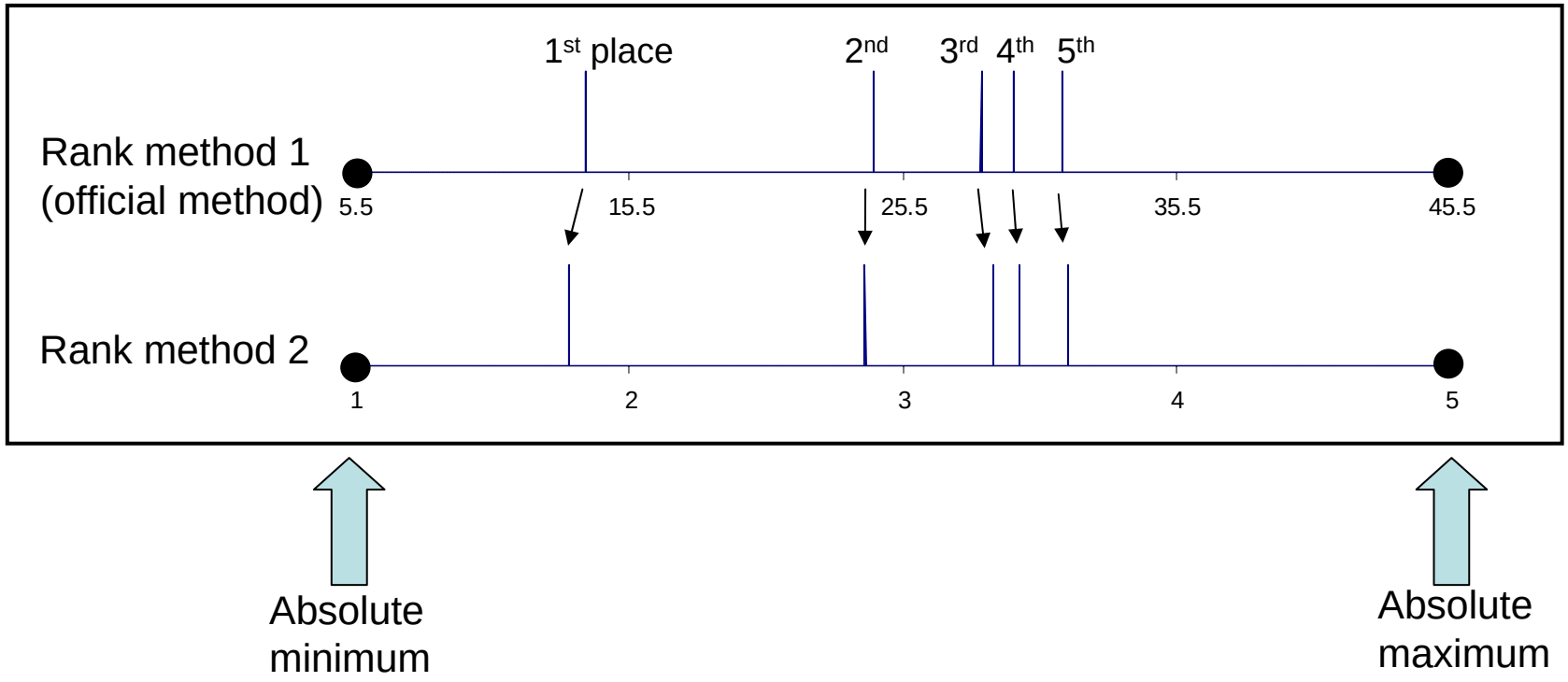
Ranking and Comparing the Competition Entries

- Two methods of ranking were considered in the final, one official, one unofficial
- Given 50 results (i.e. 10 runs, 5 algorithms),
- For each instance:

  - **Method 1(Official):**
    - Rank the 50 results from 1 to 50 (using the median in ties).
    - For each algorithm, average the ranks of its 10 runs
  - **Method 2**
    - Compare the best results of each algorithm on the instance and rank these 1 to 5 (using the median in ties).
    - Do the same with second best, third best,… to the worst results

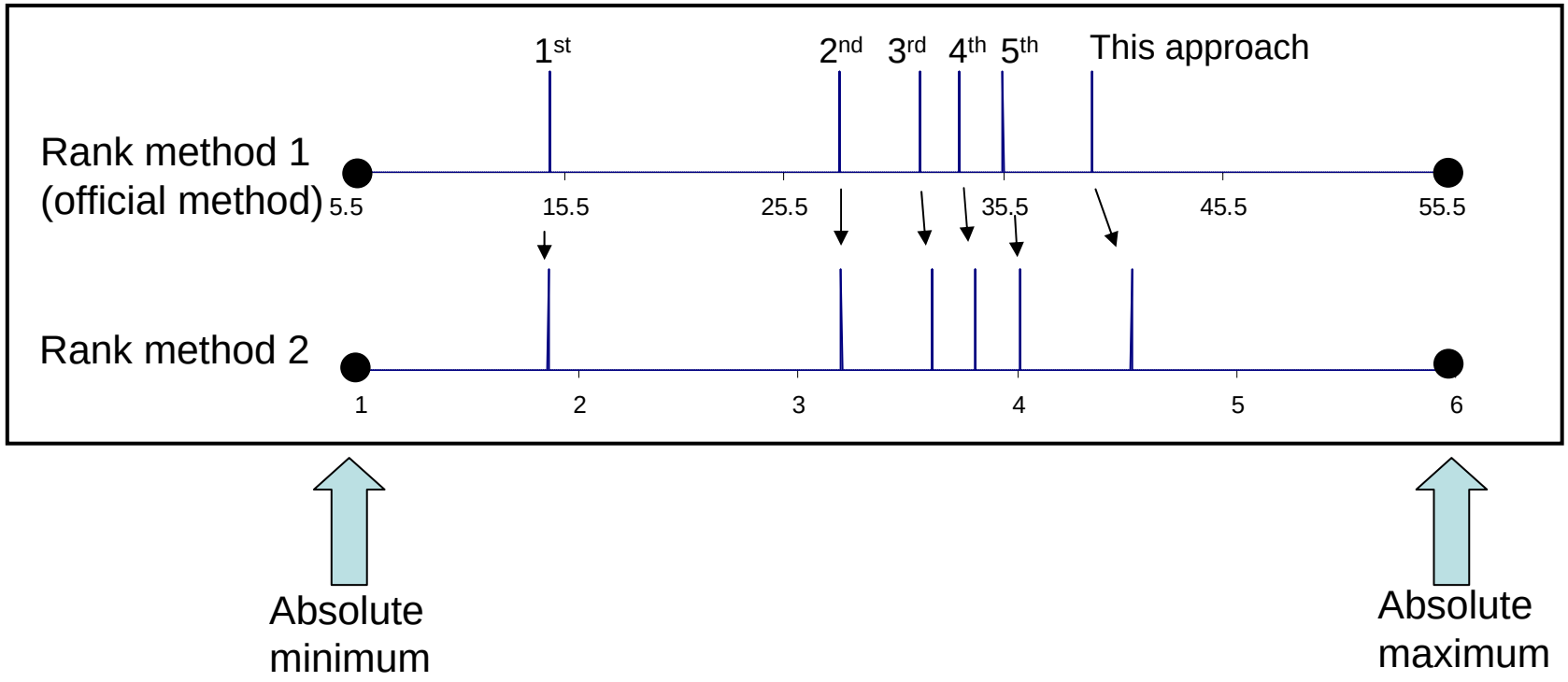
- Finally, average over all instances

Ranking and Comparing the Competition Entries



- Both methods returned the same ranking of the finalists, with similar distances between each

Comparison of final results including this approach



- (all figures adjusted according to additional results)

Comparison of final results including this approach

- (all figures adjusted according to additional results)