Scene from an imaginary dinghy race

Constructing Sailing Match Race Schedules
Round-Robin Pairing Lists

Craig Macdonald, Ciaran McCreesh, Alice Miller, Patrick Prosser

All @ Computing Science Glasgow

Craig

Ciaran

Patrick

Alice

Sailing Match Races

In a match race, skippers races against each other in pairs. There are **several hundred events** per year, each involving up to ten of the 1,500 internationally registered skippers.

Ceilidh Cup/Scottish Student Sailing Match Racing at the Royal Northern & Clyde Yacht Club

The Royal Northern & Clyde Yacht Club (RNCYC) and Scottish Student Sailing (SSS) again join forces for their annual match racing competition, held in RNCYC's Sonar at Rhu on 4th & 5th October. This event combines a match racing event for Scottish universities with additional non-qualifying entrants, with the Ceilidh Cup being awarded to the best overall team, as well as a separate prize for the best University team.

With the full nine teams, many helms at the event are returning to the Sonars, including top seed Theo Hoole (Strathclyde University) will be looking to build upon his foreign match racing trips to The Netherlands this year. RNCYC member Nicole McPherson should also be familiar in her home waters, having participated in recent events including team racing versus the Seawanhaka Corinthian Yacht Club in the Sonars. Meanwhile while unranked RNCYC member Calum Underwood who is also helming for Strathclyde University will be hoping that his summer Sonar campaign will pay off among those helms & crews less familiar with the Sonars. Finally, with the revival of the Scottish Hunter 707 fleet at Port Edgar, its not surprising that many helms have experience from the 707s, such as Emily Robertson (Glasgow University), and ringer Craig Paul (sailing for Edinburgh University).

With more than 36 races scheduled over the 2 days, adjudicated by a team of 7 umpires, and with a strengthening forecast, racing should be tight and interesting. The winner of the Ceilidh Cup will be invited to the RYA's National Match Racing Championship Grand Finals on 14-16th November.

Skippers & Clubs:

Theo Hoole, University of Strathclyde Sailing Club
Connor Simms, Dundee University Sailing Club
Nicole McPherson*, Royal Northern & Clyde Yacht Club
Alison Morrish*, University of Strathclyde Sailing Club
Craig Paul*, Edinburgh University Sailing Club
Josh Bather, University of St Andrews Sailing Club
Scott McWhirter, Glasgow University Sailing Club
Emily Robertson, Glasgow University Sailing Club
Calum Underwood, University of Strathclyde Sailing Club
* denotes teams ineligible for the SSS prize.

Photo credit: Copyright Neil Ross

By Craig Macdonald

Ceilidh Cup/Scottish Student Sailing Match Racing at the Royal Northern & Clyde Yacht Club

The Royal Northern & Clyde Yacht Club (RNCYC) and Scottish Student Sailing (SSS) again join forces for their annual match racing competition, held in RNCYC's Sonar at Rhu on 4th & 5th October. This event combines a match racing event for Scottish universities with additional non-qualifying entrants, with the Ceilidh Cup being awarded to the best overall team, as well as a separate prize for the best University team.

With the full nine teams, many helms at the event are returning to the Sonars, including top seed Theo Hoole (Strathclyde University) will be looking to build upon his foreign match racing trips to The Netherlands this year. RNCYC member Nicole McPherson should also be familiar in her home waters, having participated in recent events including team racing versus the Seawanhaka Corinthian Yacht Club in the Sonars. Meanwhile while unranked RNCYC member Calum Underwood who is also helming for Strathclyde University will be hoping that his summer Sonar campaign will pay off among those helms & crews less familiar with the Sonars. Finally, with the revival of the Scottish Hunter 707 fleet at Port Edgar, its not surprising that many helms have experience from the 707s, such as Emily Robertson (Glasgow University), and ringer Craig Paul (sailing for Edinburgh University).

With more than 36 races scheduled over the 2 days, adjudicated by a team of 7 umpires, and with a strengthening forecast, racing should be tight and interesting. The winner of the Ceilidh Cup will be invited to the RYA's National Match Racing Championship Grand Finals on 14-16th November.

Skippers & Clubs:
Theo Hoole, University of Strathclyde Sailing Club
Connor Simms, Dundee University Sailing Club
Nicole McPherson*, Royal Northern & Clyde Yacht Club
Alison Morrish*, University of Strathclyde Sailing Club
Craig Paul*, Edinburgh University Sailing Club
Josh Bather, University of St Andrews Sailing Club
Scott McWhirter, Glasgow University Sailing Club
Emily Robertson, Glasgow University Sailing Club
Calum Underwood, University of Strathclyde Sailing Club
* denotes teams ineligible for the SSS prize.

Photo credit: Copyright Neil Ross

By Craig Macdonald

Ceilidh Cup/Scottish Student Sailing Match Racing at the Royal Northern & Clyde Yacht Club

The Royal Northern & Clyde Yacht Club (RNCYC) and Scottish Student Sailing (SSS) again join forces for their annual match racing competition, held in RNCYC's Sonar at Rhu on 4th & 5th October. This event combines a match racing event for Scottish universities with additional non-qualifying entrants, with the Ceilidh Cup being awarded to the best overall team, as well as a separate prize for the best University team.

With the full nine teams, many helms at the event are returning to the Sonars, including top seed Theo Hoole (Strathclyde University) will be looking to build upon his foreign match racing trips to The Netherlands this year. RNCYC member Nicole McPherson should also be familiar in her home waters, having participated in recent events including team racing versus the Seawanhaka Corinthian Yacht Club in the Sonars. Meanwhile while unranked RNCYC member Calum Underwood who is also helming for Strathclyde University will be hoping that his summer Sonar campaign will pay off among those helms & crews less familiar with the Sonars. Finally, with the revival of the Scottish Hunter 707 fleet at Port Edgar, its not surprising that many helms have experience from the 707s, such as Emily Robertson (Glasgow University), and ringer Craig Paul (sailing for Edinburgh University).

With more than 36 races scheduled over the 2 days, adjudicated by a team of 7 umpires, and with a strengthening forecast, racing should be tight and interesting. The winner of the Ceilidh Cup will be invited to the RYA's National Match Racing Championship Grand Finals on 14-16th November.

Skippers & Clubs:
Theo Hoole, University of Strathclyde Sailing Club
Connor Simms, Dundee University Sailing Club
Nicole McPherson*, Royal Northern & Clyde Yacht Club
Alison Morrish*, University of Strathclyde Sailing Club
Craig Paul*, Edinburgh University Sailing Club
Josh Bather, University of St Andrews Sailing Club
Scott McWhirter, Glasgow University Sailing Club
Emily Robertson, Glasgow University Sailing Club
Calum Underwood, University of Strathclyde Sailing Club
* denotes teams ineligible for the SSS prize.

Photo credit: Copyright Neil Ross

By Craig Macdonald

Rhu

This

by Fiona Holland June 8, 2015, 02:32:00 (GMT Standard Time)

Old Pulteney IRC Scottish Championship and Mudhook Regatta © Neill Ross Photography

Not this

Tanjung Rhu Resort Langkawi

Ceilidh Cup/Scottish Student Sailing Match Racing at the Royal Northern & Clyde Yacht Club

The Royal Northern & Clyde Yacht Club (RNCYC) and Scottish Student Sailing (SSS) again join forces for their annual match racing competition, held in RNCYC's Sonar at Rhu on 4th & 5th October. This event combines a match racing event for Scottish universities with additional non-qualifying entrants, with the Ceilidh Cup being awarded to the best overall team, as well as a separate prize for the best University team.

With the full nine teams, many helms at the event are returning to the Sonars, including top seed Theo Hoole (Strathclyde University) will be looking to build upon his foreign match racing trips to The Netherlands this year. RNCYC member Nicole McPherson should also be familiar in her home waters, having participated in recent events including team racing versus the Seawanhaka Corinthian Yacht Club in the Sonars. Meanwhile while unranked RNCYC member Calum Underwood who is also helming for Strathclyde University will be hoping that his summer Sonar campaign will pay off among those helms & crews less familiar with the Sonars. Finally, with the revival of the Scottish Hunter 707 fleet at Port Edgar, its not surprising that many helms have experience from the 707s, such as Emily Robertson (Glasgow University), and ringer Craig Paul (sailing for Edinburgh University).

With more than 36 races scheduled over the 2 days, adjudicated by a team of 7 umpires, and with a strengthening forecast, racing should be tight and interesting. The winner of the Ceilidh Cup will be invited to the RYA's National Match Racing Championship Grand Finals on 14-16th November.

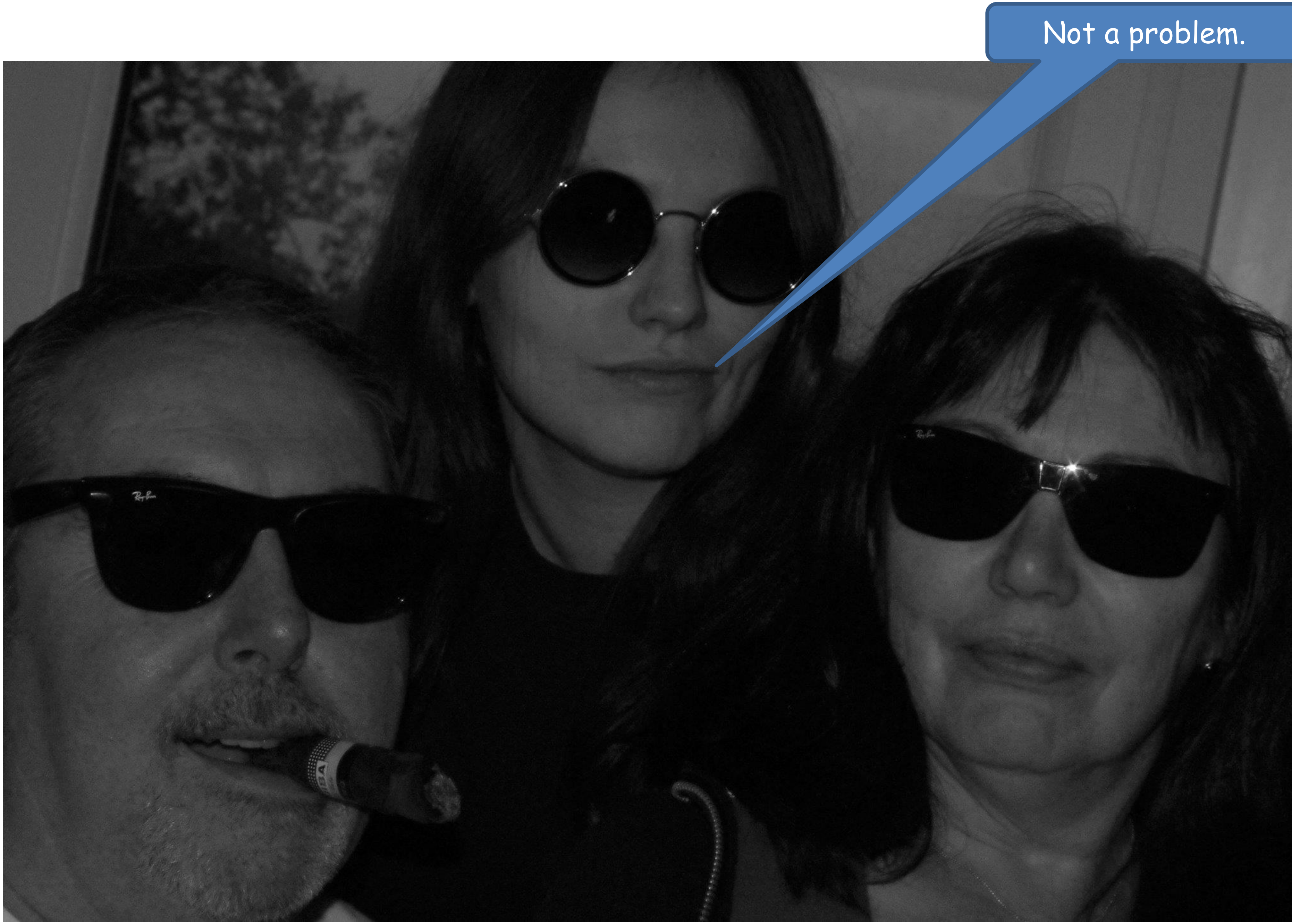
Skippers & Clubs:
Theo Hoole, University of Strathclyde Sailing Club
Connor Simms, Dundee University Sailing Club
Nicole McPherson*, Royal Northern & Clyde Yacht Club
Alison Morrish*, University of Strathclyde Sailing Club
Craig Paul*, Edinburgh University Sailing Club
Josh Bather, University of St Andrews Sailing Club
Scott McWhirter, Glasgow University Sailing Club
Emily Robertson, Glasgow University Sailing Club
Calum Underwood, University of Strathclyde Sailing Club
* denotes teams ineligible for the SSS prize.

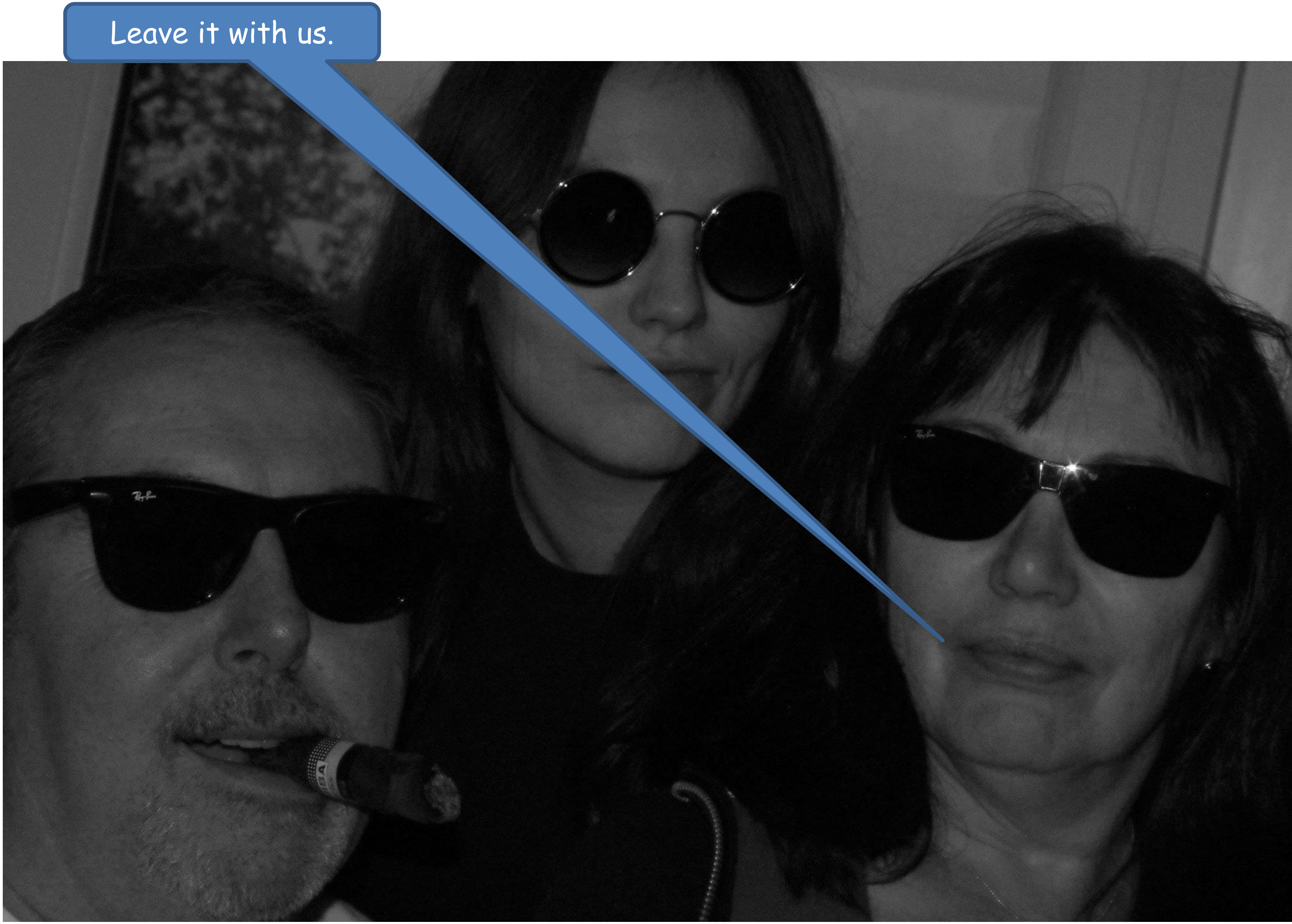
Photo credit: Copyright Neil Ross

By Craig Macdonald


What’s the problem Craig?

- We have n ***skippers*** and m ***boats*** ($m \leq n$)
- Each skipper has to compete against every other skipper once and once only.
- We can send m skippers out together, in pairs (m/2 pairs)
- We call this a ***flight***
- There are some restrictions to take into consideration


Not a problem.


Leave it with us.


It's a round robin.

1st Stab

Consider 7 skippers and 6 boats

1st Stab

Consider 7 skippers and 6 boats

- We have 7.6/2 = 21 matches

1st Stab

Consider 7 skippers and 6 boats

- We have 7.6/2 = 21 matches
- Given 6 boats we have 3 matches in each flight

1st Stab

Consider 7 skippers and 6 boats

- We have 7.6/2 = 21 matches
- Given 6 boats we have 3 matches in each flight
- We have 7 flights

1st Stab

Consider 7 skippers and 6 boats

- We have 7.6/2 = 21 matches
- Given 6 boats we have 3 matches in each flight
- We have 7 flights

Flight	Matches		
0	(0,1)	(2,3)	(4,5)
1	(0,2)	(4,6)	(1,5)
2	(2,6)	(0,5)	(1,3)
3	(5,6)	(0,3)	(1,4)
4	(3,5)	(1,6)	(2,4)
5	(3,6)	(0,4)	(2,5)
6	(3,4)	(1,2)	(0,6)

1st Stab

Consider 7 skippers and 6 boats

- We have 7.6/2 = 21 matches
- Given 6 boats we have 3 matches in each flight
- We have 7 flights

Skipper 0 is in a match with skipper 1 in flight 0

Flight	Matches		
0	(0,1)	(2,3)	(4,5)
1	(0,2)	(4,6)	(1,5)
2	(2,6)	(0,5)	(1,3)
3	(5,6)	(0,3)	(1,4)
4	(3,5)	(1,6)	(2,4)
5	(3,6)	(0,4)	(2,5)
6	(3,4)	(1,2)	(0,6)

1st Stab

Consider 7 skippers and 6 boats

- We have 7.6/2 = 21 matches
- Given 6 boats we have 3 matches in each flight
- We have 7 flights

Skipper 1 is in a match with skipper 6 in flight 4

Flight	Matches		
0	(0,1)	(2,3)	(4,5)
1	(0,2)	(4,6)	(1,5)
2	(2,6)	(0,5)	(1,3)
3	(5,6)	(0,3)	(1,4)
4	(3,5)	(1,6)	(2,4)
5	(3,6)	(0,4)	(2,5)
6	(3,4)	(1,2)	(0,6)

1st Stab

	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							

Flight	Matches		
0	(0,1)	(2,3)	(4,5)
1	(0,2)	(4,6)	(1,5)
2	(2,6)	(0,5)	(1,3)
3	(5,6)	(0,3)	(1,4)
4	(3,5)	(1,6)	(2,4)
5	(3,6)	(0,4)	(2,5)
6	(3,4)	(1,2)	(0,6)

1st Stab

	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							

M[i][j] = flight that match (i,j) takes place

Flight	Matches		
0	(0,1)	(2,3)	(4,5)
1	(0,2)	(4,6)	(1,5)
2	(2,6)	(0,5)	(1,3)
3	(5,6)	(0,3)	(1,4)
4	(3,5)	(1,6)	(2,4)
5	(3,6)	(0,4)	(2,5)
6	(3,4)	(1,2)	(0,6)

1st Stab

	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							

M[i][j] = flight that match (i,j) takes place

M[i][j] ≡ M[j][i]

Flight	Matches		
0	(0,1)	(2,3)	(4,5)
1	(0,2)	(4,6)	(1,5)
2	(2,6)	(0,5)	(1,3)
3	(5,6)	(0,3)	(1,4)
4	(3,5)	(1,6)	(2,4)
5	(3,6)	(0,4)	(2,5)
6	(3,4)	(1,2)	(0,6)

1st Stab

	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							

M[i][j] = flight that match (i,j) takes place
M[i][j] ≡ M[j][i]
allDifferent(M[i])

Flight	Matches		
0	(0,1)	(2,3)	(4,5)
1	(0,2)	(4,6)	(1,5)
2	(2,6)	(0,5)	(1,3)
3	(5,6)	(0,3)	(1,4)
4	(3,5)	(1,6)	(2,4)
5	(3,6)	(0,4)	(2,5)
6	(3,4)	(1,2)	(0,6)

1st Stab

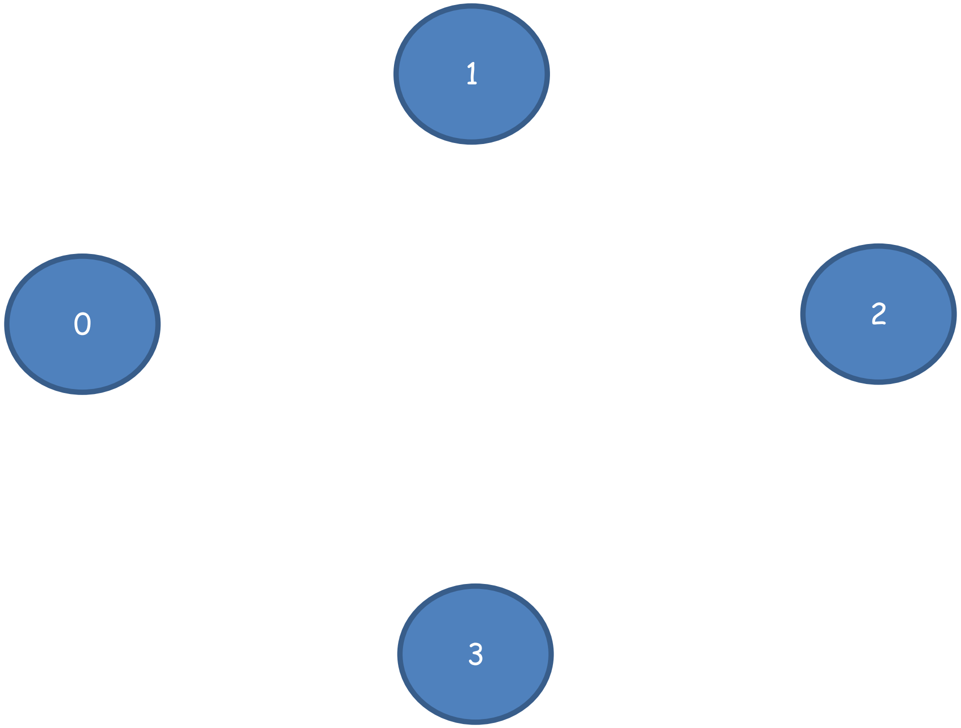
	0	1	2	3	4	5	6
0		0	1	3	5	2	6
1			6	2	3	1	4
2				0	4	5	2
3					6	4	5
4						0	1
5							3
6							

M[i][j] = flight that match (i,j) takes place
M[i][j] ≡ M[j][i]
allDifferent(M[i])

Flight	Matches		
0	(0,1)	(2,3)	(4,5)
1	(0,2)	(4,6)	(1,5)
2	(2,6)	(0,5)	(1,3)
3	(5,6)	(0,3)	(1,4)
4	(3,5)	(1,6)	(2,4)
5	(3,6)	(0,4)	(2,5)
6	(3,4)	(1,2)	(0,6)

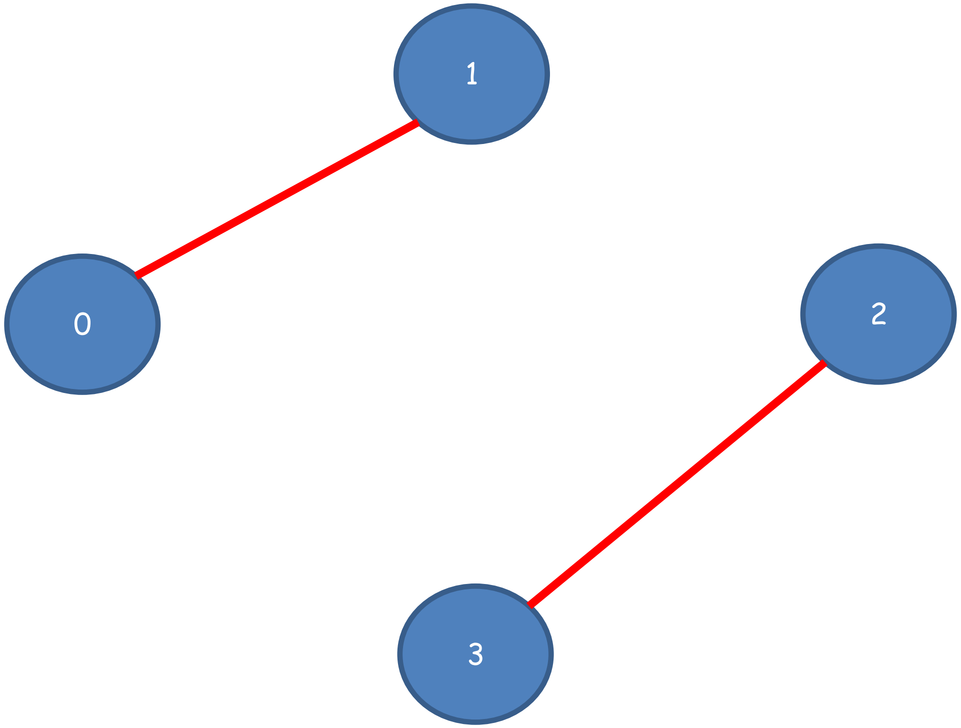
Another view: vertices are skippers
coloured edge is time of match
4 skippers & 4 boats & 6 matches

1st Stab



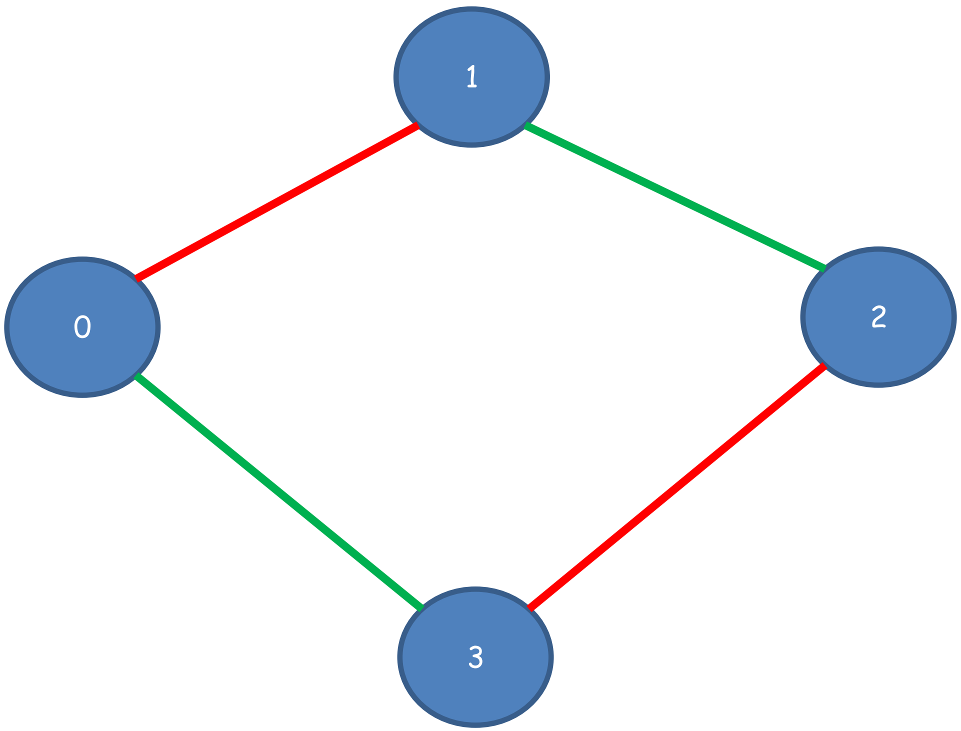
Another view: vertices are skippers
coloured edge is time of match
4 skippers & 4 boats

1st Stab



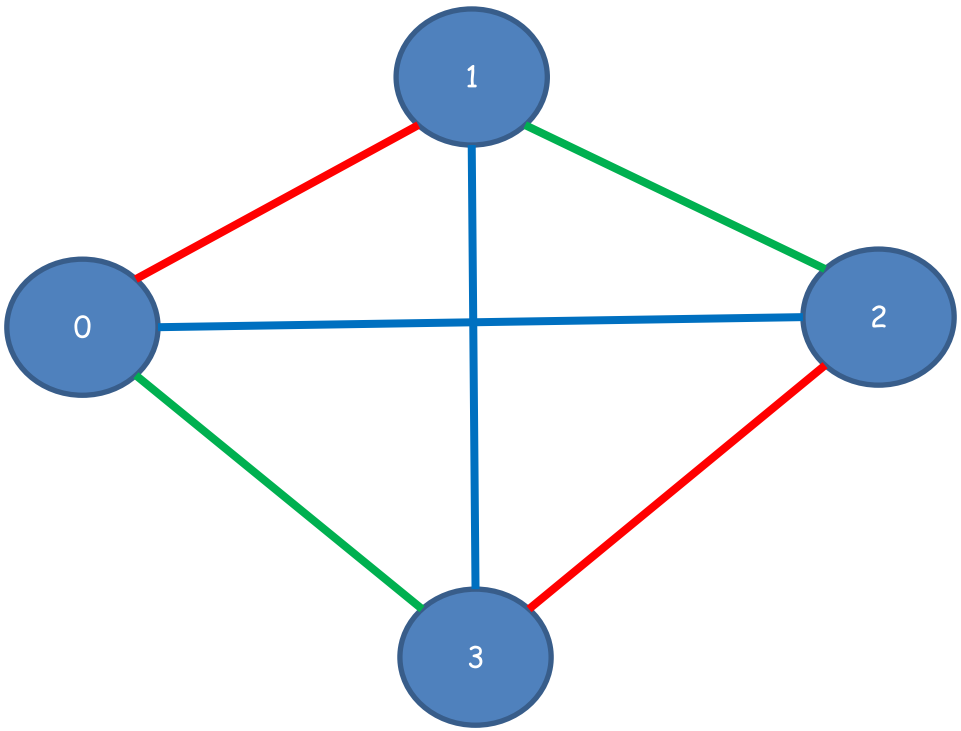
Another view: vertices are skippers
coloured edge is time of match
4 skippers & 4 boats

1st Stab



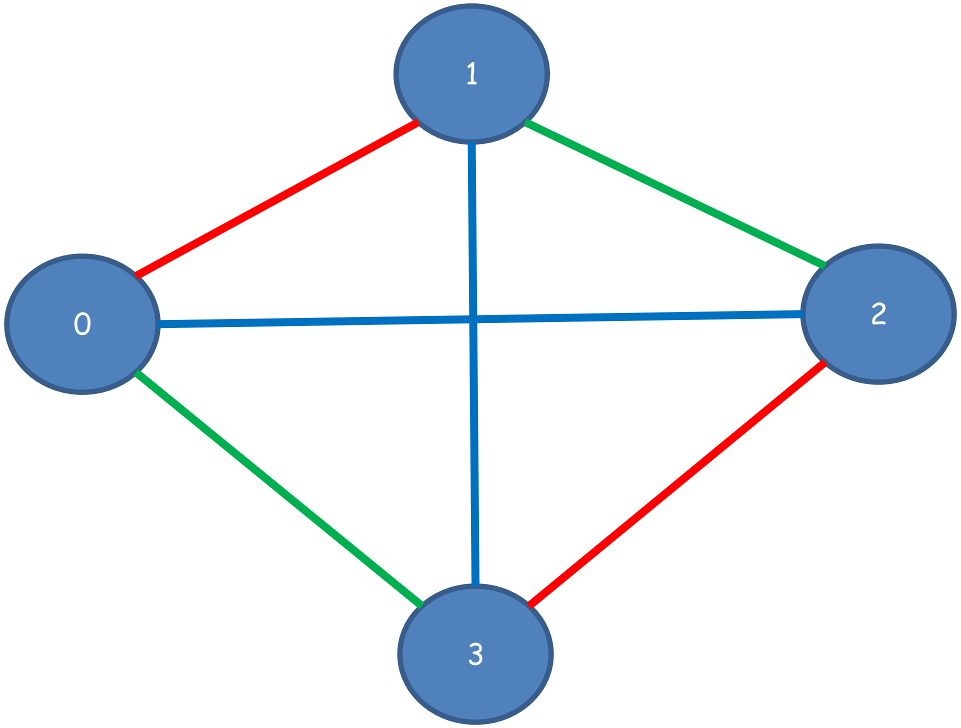
Another view: vertices are skippers
coloured edge is time of match
4 skippers & 4 boats

1st Stab



Another view: vertices are skippers
coloured edge is time of match
4 skippers & 4 boats

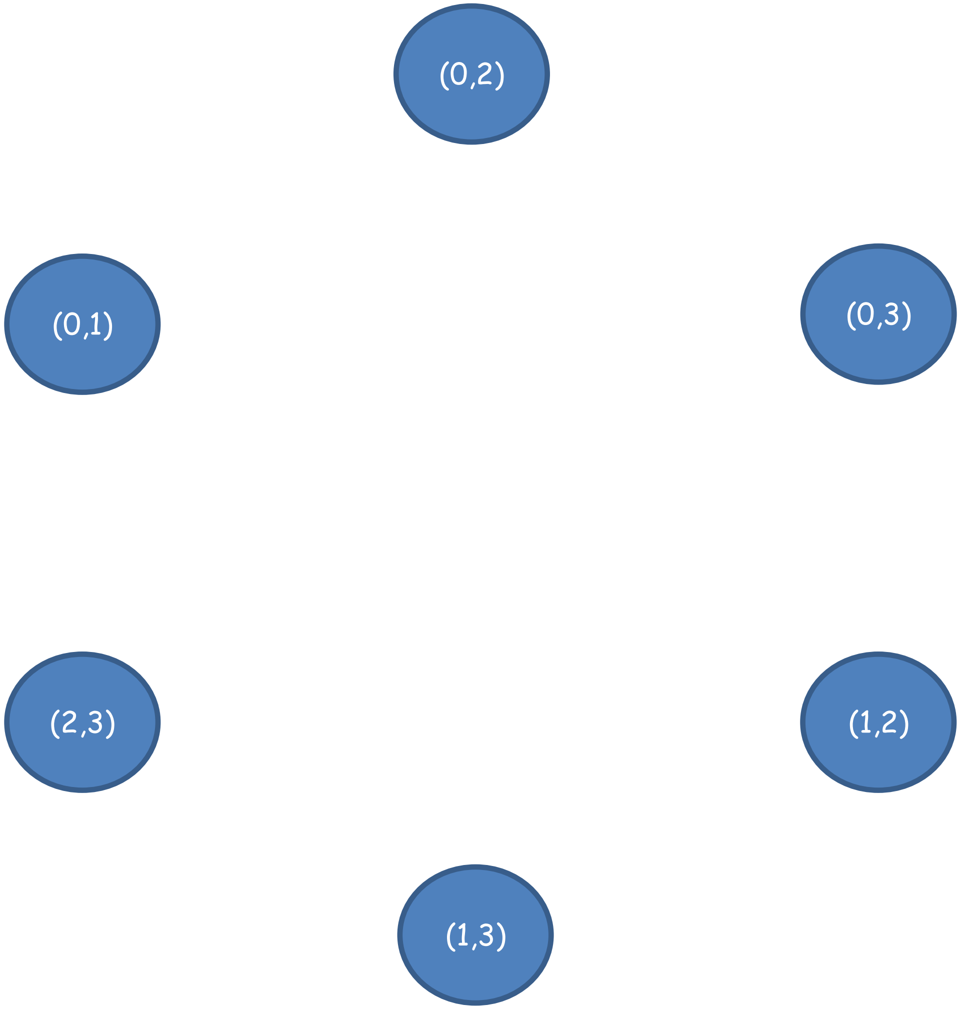
1st Stab



1-factorisation

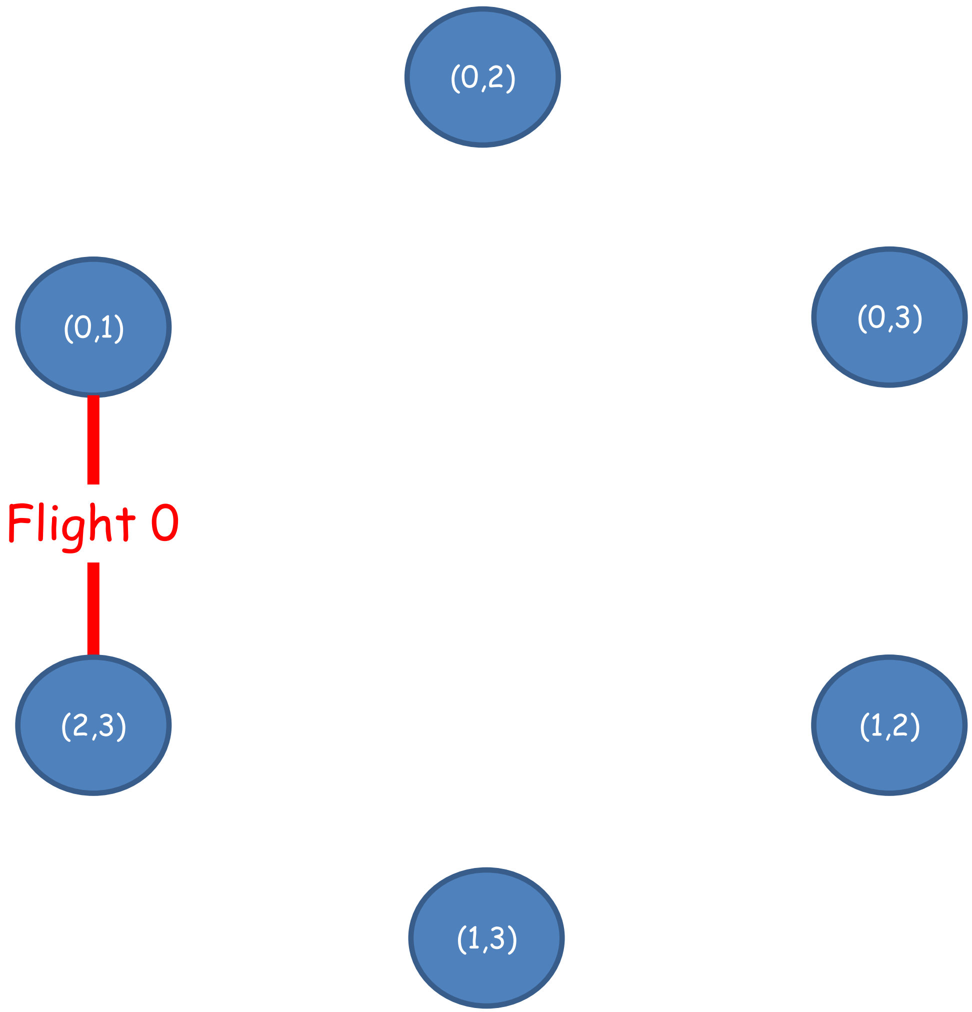
Another view: vertices are matches
edge if matches at same time
4 skippers & 4 boats

1st Stab



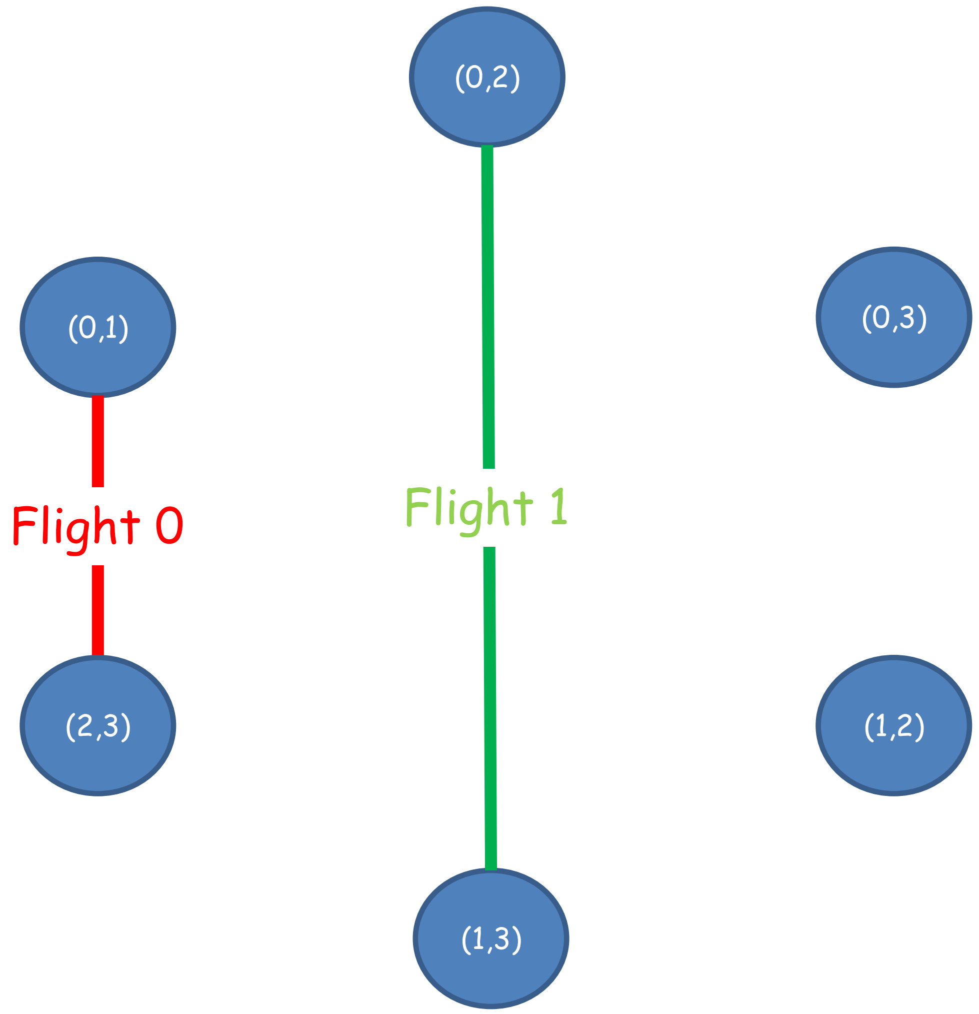
Another view: vertices are matches
edge if matches at same time
4 skippers & 4 boats

1^{st} Stab



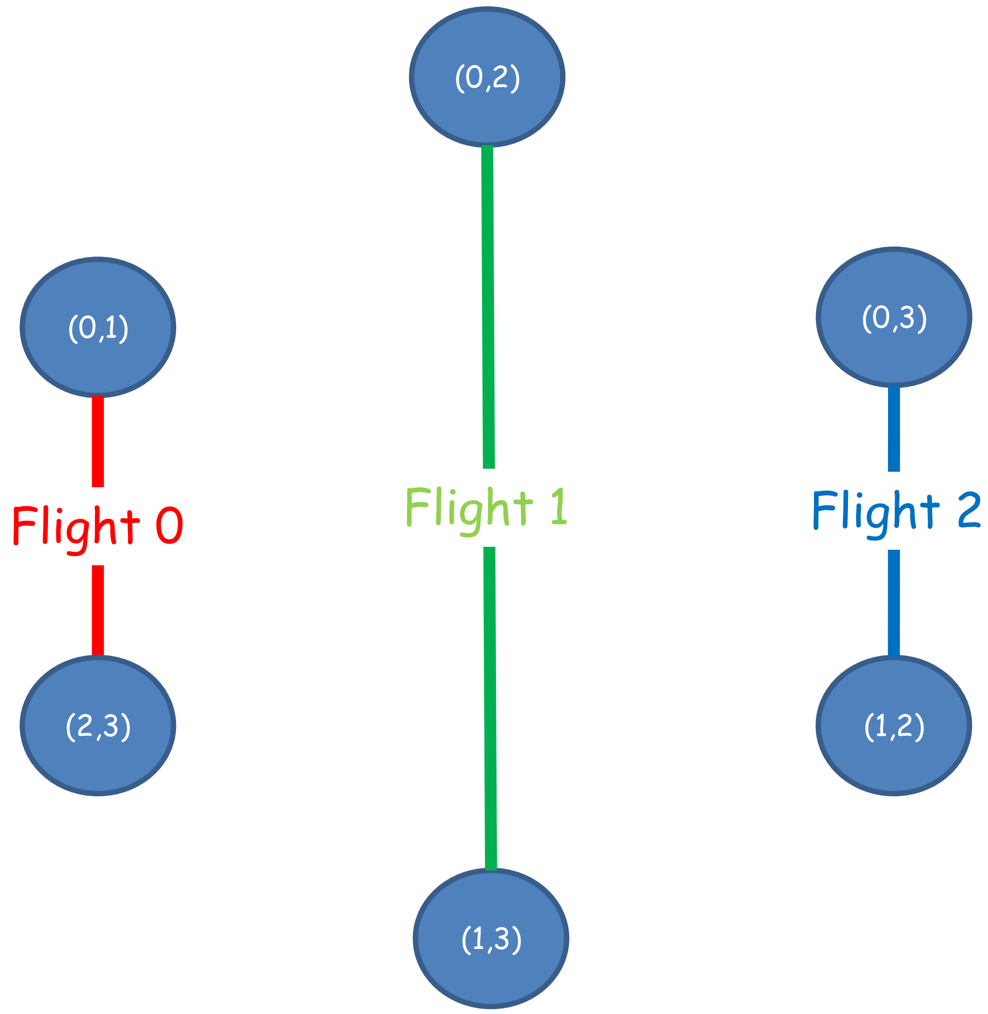
Another view: vertices are matches
edge if matches at same time
4 skippers & 4 boats

1st Stab



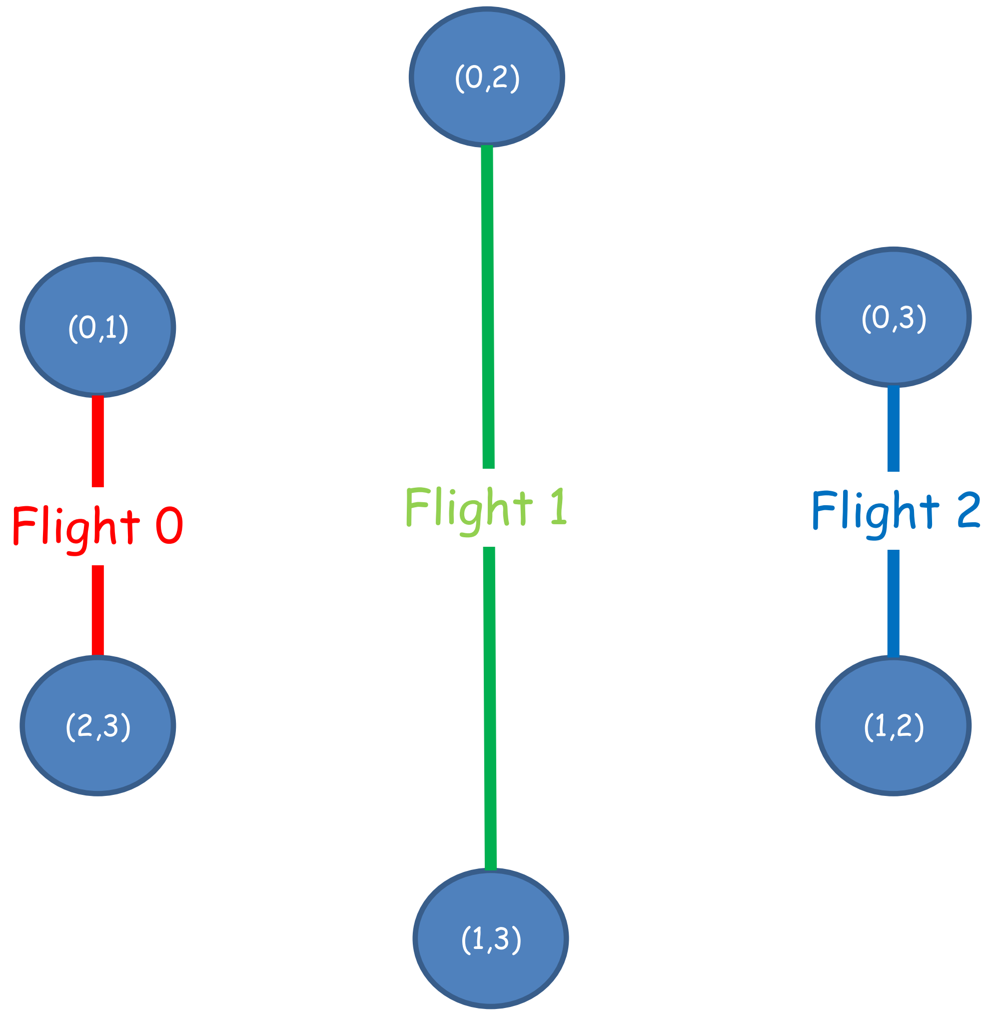
Another view: vertices are matches
edge if matches at same time
4 skippers & 4 boats

1st Stab

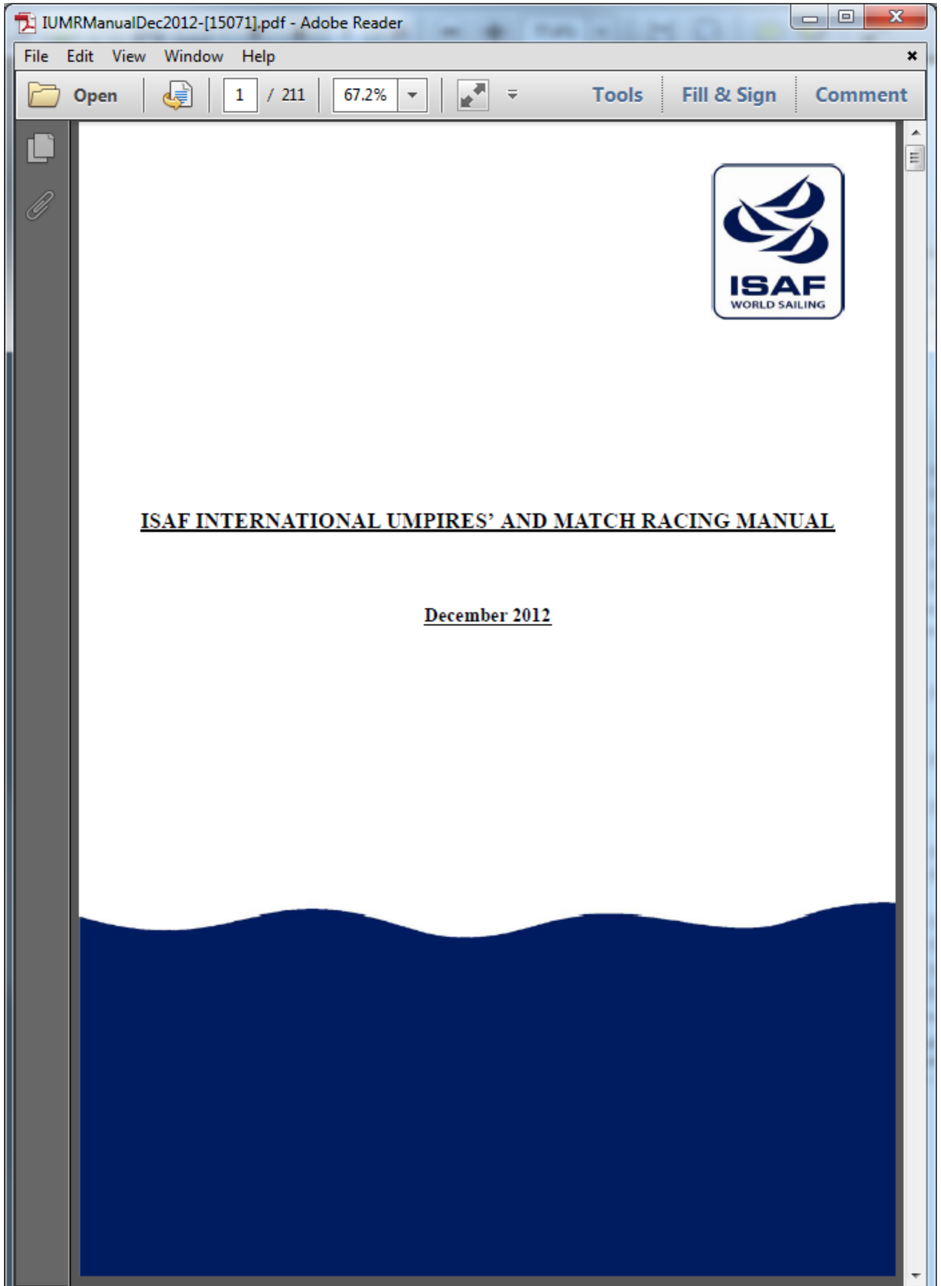


Another view: vertices are matches
edge if matches at same time
4 skippers & 4 boats

1st Stab



Colouring of the line graph



There are some restrictions to take into consideration

Principal Criteria in Order of Priority:

1. Each skipper sails against each other skipper once.

2*. When skippers have an even number of matches, they have the same number of port and starboard assignments.

3*. When skippers have an odd number of matches, the first half of the skippers will have one more starboard assignment.

4. No skipper in the last match of a flight should be in the first match of the next flight.
5. No skipper should have more than two consecutive port or starboard assignments.
6. Each skipper should be assigned to match 1, match 2, etc. in a flight as equally as possible.
7. In flights with five or more matches, no skipper should be in the next-to-last match in a flight and then in the first match of the next flight.
8. If possible, a skipper should be starboard when meeting the nearest lowest ranked skipper (i.e. #1 will be starboard against #2, #3 will be starboard against #4).
9. Close-ranked skippers meet in the last flight.
10. Minimize the number of boat changes.
11. Skippers in the last match of a flight do not change boats.
12. Skippers in new boats do not sail in the first match of the next flight.
13. Skippers have a reasonable sequence of matches and blanks.

Note that criteria 10, 11 and 12 only apply when there are fewer boats than skippers; 11 and 12 override 6 when changes are required, and 13 applies when there are more boats than skippers.

A copy of these variables and constraints is created for each skipper σ:

$$\forall \tau \in \{0 \dots t-1\} : \mathit{timeSlot}[\tau] \in \{-2 \dots t-1\} \setminus \{\sigma\} \quad \text{(V1)}$$

$$\forall i \in \{0 \dots f-1\} : \mathit{state}[i] \in \{\textsc{First}, \textsc{Mid}, \textsc{Last}, \textsc{Bye}, \textsc{End}\} \quad \text{(V2)}$$

$$\forall i \in \{0 \dots f-1\} : \quad \text{(C1)}$$

$$\begin{aligned}
\mathit{state}[i] &= \textsc{First} \Leftrightarrow \mathit{timeSlot}[m \cdot i] \geq 0\\
\mathit{state}[i] &= \textsc{Mid} \Leftrightarrow \exists j \in \{m \cdot i + 1 \dots m \cdot (i+1) - 2\} : \mathit{timeSlot}[j] \geq 0\\
\mathit{state}[i] &= \textsc{Last} \Leftrightarrow \mathit{timeSlot}[m \cdot (i+1) - 1] \geq 0\\
\mathit{state}[i] &= \textsc{Bye} \Leftrightarrow \forall j \in \{m \cdot i \dots m \cdot (i+1) - 1\} : \mathit{timeSlot}[j] = -1\\
\mathit{state}[i] &= \textsc{End} \Leftrightarrow \forall j \in \{m \cdot i \dots m \cdot (i+1) - 1\} : \mathit{timeSlot}[j] = -2
\end{aligned}$$

$$\mathrm{eachOccursExactlyOnce}(\mathit{timeSlot}, \{0, \dots, n-1\} \setminus \{\sigma\}) \quad \text{(C2)}$$

$$\mathrm{regular}(\mathit{state}, \text{Figure 2}) \quad \text{(C3)}$$

$$\forall i \in \{0 \dots f-2\} : \mathit{change}[i] \in \{0, 1\} \quad \text{(V3)}$$

$$\mathit{totalChanges} \in \mathbb{N} \quad \text{(V4)}$$

$$\forall i \in \{0 \dots f-2\} : \mathit{change}[i] = 1 \Leftrightarrow \mathit{state}[i] = \textsc{Bye} \wedge \mathit{state}[i+1] \neq \textsc{Bye} \quad \text{(C4)}$$

$$\mathit{totalChanges} = \sum \mathit{change} \quad \text{(C5)}$$

Match and temporal perspectives:

$$\forall i \in \{0 \dots n-2\} : \forall j \in \{i+1 \dots n-1\} :$$

$$\mathit{match}[i,j] \in \{0 \dots t-1\} \quad \text{(V5)}$$

$$\mathit{match}[j,i] \equiv \mathit{match}[i,j]$$

$$\forall k \in \{0 \dots t-1\} :$$

$$\mathit{match}[i,j] = k \Leftrightarrow \sigma[i].\mathit{timeSlot}[k] = j \wedge \sigma[j].\mathit{timeSlot}[k] = i \quad \text{(C6)}$$

$$\forall i \in \{0 \dots n-2\} : \forall j \in \{i+1 \dots n-1\} :$$

$$\mathit{modMatch}[i,j] \in \{0 \dots f-1\} \quad \text{(V6)}$$

$$\mathit{modMatch}[j,i] \equiv \mathit{modMatch}[i,j]$$

$$\mathit{modMatch}[i,j] = \mathit{match}[i,j]/m \quad \text{(C7)}$$

$$\forall i \in \{0 \dots n-1\} : \mathrm{allDifferent}(\mathit{modMatch}[i]) \quad \text{(C8)}$$

$$\forall \tau \in \{0 \dots t-1\} :$$

$$\mathit{time}[\tau] \in \{(0,1) \dots (n-2, n-1)\} \quad \text{(V7)}$$

$$\forall i \in \{0 \dots n-2\} : \forall j \in \{i+1 \dots n-1\} : \mathit{time}[\tau] = (i,j) \Leftrightarrow \mathit{match}[i,j] = \tau \quad \text{(C9)}$$

$$\mathit{totalBoatChanges} = \sum \sigma.\mathit{totalChanges} \quad \text{(V8)}$$

$$\mathrm{minimise}(\mathit{totalBoatChanges}) \quad \text{(C10)}$$

Fig. 1. Our constraint model, from a skipper perspective (top) and a match and temporal perspective (below). The number of skippers is n, and m is the number of matches in a flight. The number of flights is $f = \lceil n(n-1)/m \rceil$, and there are $t = n(n-1)/2$ matches (and time slots) in total. We define $\mathbb{N}$ to include zero.

The objective value from stage 1 is used as a hard constraint:

$$\mathit{totalBoatChanges} \leq \beta \quad \text{(C11)}$$

A copy of these variables and constraints is created for each skipper σ:

$$\forall i \in \{0 \dots m-1\} : \forall j \in \{0 \dots f-1\} :$$

$$\mathit{position}[i,j] \in \{0,1\} \quad \text{(V9)}$$

$$\mathit{position}[i,j] = 1 \Leftrightarrow \mathit{timeSlot}[m.j + i] \geq 0 \quad \text{(C12)}$$

$$\forall i \in \{0 \dots m-1\} :$$

$$\mathit{imbalance}[i] \in \mathbb{N} \quad \text{(V10)}$$

$$\mathit{imbalance}[i] = \left| \frac{n-1}{m} - \sum_{j=0}^{f-1} \mathit{position}[i,j] \right| \quad \text{(C13)}$$

$$\mathit{maxImbalance} \in \mathbb{N} \quad \text{(V11)}$$

$$\mathit{maxImbalance} = \mathrm{maximum}(\mathit{imbalance}) \quad \text{(C14)}$$

We minimize the maximum imbalance over all skippers:

$$\forall i \in \{0 \dots n-1\} : \mathit{imbalance}[i] \equiv \sigma[i].\mathit{maxImbalance} \quad \text{(V12)}$$

$$\mathit{maxImbalance} \in \mathbb{N} \quad \text{(V13)}$$

$$\mathit{maxImbalance} = \mathrm{maximum}(\mathit{imbalance}) \quad \text{(C15)}$$

$$\mathrm{minimise}(\mathit{maxImbalance}) \quad \text{(C16)}$$

A copy of these variables and constraints is created for each skipper σ:

$\forall \tau \in \{0 \dots t{-}1\} : \mathit{timeSlot}[\tau] \in \{-2 \dots t{-}1\} \setminus \{\sigma\}$ (V1)

$\forall i \in \{0 \dots f{-}1\} : \mathit{state}[i] \in \{\textsc{First}, \textsc{Mid}, \textsc{Last}, \textsc{Bye}, \textsc{End}\}$ (V2)

$\forall i \in \{0 \dots f{-}1\}$: (C1)

$$\begin{aligned}
\mathit{state}[i] = \textsc{First} &\Leftrightarrow \mathit{timeSlot}[m \cdot i] \geq 0 \\
\mathit{state}[i] = \textsc{Mid} &\Leftrightarrow \exists j \in \{m \cdot i + 1 \dots m \cdot (i+1) - 2\} : \mathit{timeSlot}[j] \geq 0 \\
\mathit{state}[i] = \textsc{Last} &\Leftrightarrow \mathit{timeSlot}[m \cdot (i+1) - 1] \geq 0 \\
\mathit{state}[i] = \textsc{Bye} &\Leftrightarrow \forall j \in \{m \cdot i \dots m \cdot (i+1) - 1\} : \mathit{timeSlot}[j] = -1 \\
\mathit{state}[i] = \textsc{End} &\Leftrightarrow \forall j \in \{m \cdot i \dots m \cdot (i+1) - 1\} : \mathit{timeSlot}[j] = -2
\end{aligned}$$

eachOccursExactlyOnce($\mathit{timeSlot}$, $\{0, \dots, n{-}1\} \setminus \{\sigma\}$) (C2)

regular(state, Figure 2) (C3)

$\forall i \in \{0 \dots f{-}2\} : \mathit{change}[i] \in \{0, 1\}$ (V3)

$\mathit{totalChanges} \in \mathbb{N}$ (V4)

$\forall i \in \{0 \dots f{-}2\} : \mathit{change}[i] = 1 \Leftrightarrow \mathit{state}[i] = \textsc{Bye} \wedge \mathit{state}[i+1] \neq \textsc{Bye}$ (C4)

$\mathit{totalChanges} = \sum \mathit{change}$ (C5)

Match and temporal perspectives:

$\forall i \in \{0 \dots n{-}2\} : \forall j \in \{i{+}1 \dots n{-}1\}$:

$\mathit{match}[i,j] \in \{0 \dots t{-}1\}$ ɔ)

$\mathit{match}[j,i] \equiv \mathit{match}[i,j]$

$\forall k \in \{0 \dots t{-}1\}$:

$\mathit{match}[i,j] = k \Leftrightarrow \sigma[i].\mathit{timeSlot}[k] = j \wedge \sigma[j]$ ... (C6)

$\forall i \in \{0 \dots n{-}2\} : \forall j \in \{i{+}1 \dots n{-}1\}$:

$\mathit{modMatch}[i,j] \in \{0 \dots f{-}1\}$ (V6)

$\mathit{modMatch}[j,i] \equiv \mathit{modMatch}[i,j]$

$\mathit{modMatch}[i,j] = \mathit{match}[i,j]/$ ... (C7)

$\forall i \in \{0 \dots n{-}1\}$: allDiffer... (C8)

$\forall \tau \in \{0 \dots t{-}1\}$:

$\mathit{time}[\tau] \in \{(0,1)$ ... (V7)

$\forall i \in \{0 \dots n{-}$ ... $\dots n{-}1\} : \mathit{time}[\tau] = (i,j) \Leftrightarrow \mathit{match}[i,j] = \tau$ (C9)

$\mathit{totalBoatChang}$... $.\mathit{totalChanges}$ (V8)

minimise($\mathit{totalBoa}$...anges) (C10)

Fig. 1. Our constraint model, from a skipper perspective (top) and a match and temporal perspective (below). The number of skippers is n, and m is the number of matches in a flight. The number of flights is $f = \lceil n(n-1)/m \rceil$, and there are $t = n(n-1)/2$ matches (and time slots) in total. We define $\mathbb{N}$ to include zero.

The objectiv... stage 1 is used as a hard constraint:

totalF... $\leq \beta$ (C11)

...these variables and constraints is created for each skipper σ:

... $\in \{0 \dots m{-}1\} : \forall j \in \{0 \dots f{-}1\}$:

$\mathit{position}[i,j] \in \{0, 1\}$ (V9)

$\mathit{position}[i,j] = 1 \Leftrightarrow \mathit{timeSlot}[m.j + i] \geq 0$ (C12)

$\forall i \in \{0 \dots m{-}1\}$:

$\mathit{imbalance}[i] \in \mathbb{N}$ (V10)

$$\mathit{imbalance}[i] = \left| \frac{n-1}{m} - \sum_{j=0}^{f-1} \mathit{position}[i,j] \right| \quad \text{(C13)}$$

$\mathit{maxImbalance} \in \mathbb{N}$ (V11)

$\mathit{maxImbalance} = \text{maximum}(\mathit{imbalance})$ (C14)

We minimize the maximum imbalance over all skippers:

$\forall i \in \{0 \dots n{-}1\} : \mathit{imbalance}[i] \equiv \sigma[i].\mathit{maxImbalance}$ (V12)

$\mathit{maxImbalance} \in \mathbb{N}$ (V13)

$\mathit{maxImbalance} = \text{maximum}(\mathit{imbalance})$ (C15)

minimise($\mathit{maxImbalance}$) (C16)

For an explanation at this level of detail, read the paper

A copy of these variables and constraints is created for each skipper σ:

$\forall \tau \in \{0 \dots t{-}1\} : \mathit{timeSlot}[\tau] \in \{-2 \dots t{-}1\} \setminus \{\sigma\}$ (V1)

$\forall i \in \{0 \dots f{-}1\} : \mathit{state}[i] \in \{\text{FIRST}, \text{MID}, \text{LAST}, \text{BYE}, \text{END}\}$ (V2)

$\forall i \in \{0 \dots f{-}1\} :$ (C1)

$\mathit{state}[i] = \text{FIRST} \Leftrightarrow \mathit{timeSlot}[m \cdot i] \geq 0$

$\mathit{state}[i] = \text{MID} \Leftrightarrow \exists j \in \{m \cdot i + 1 \dots m \cdot (i+1) - 2\} : \mathit{timeSlot}[j] \geq 0$

$\mathit{state}[i] = \text{LAST} \Leftrightarrow \mathit{timeSlot}[m \cdot (i+1) - 1] \geq 0$

$\mathit{state}[i] = \text{BYE} \Leftrightarrow \forall j \in \{m \cdot i \dots m \cdot (i+1) - 1\} : \mathit{timeSlot}[j] = -1$

$\mathit{state}[i] = \text{END} \Leftrightarrow \forall j \in \{m \cdot i \dots m \cdot (i+1) - 1\} : \mathit{timeSlot}[j] = -2$

$\text{eachOccursExactlyOnce}(\mathit{timeSlot}, \{0, \dots, n{-}1\} \setminus \{\sigma\})$ (C2)

$\text{regular}(\mathit{state}, \text{Figure 2})$ (C3)

$\forall i \in \{0 \dots f{-}2\} : \mathit{change}[i] \in \{0, 1\}$ (V3)

$\mathit{totalChanges} \in \mathbb{N}$ (V4)

$\forall i \in \{0 \dots f{-}2\} : \mathit{change}[i] = 1 \Leftrightarrow \mathit{state}[i] = \text{BYE} \wedge \mathit{state}[i+1] \neq \text{BYE}$ (C4)

$\mathit{totalChanges} = \sum \mathit{change}$ (C5)

Match and temporal perspectives:

$\forall i \in \{0 \dots n{-}2\} : \forall j \in \{i{+}1 \dots n{-}1\} :$

$\mathit{match}[i,j] \in \{0 \dots t{-}1\}$ 5)

$\mathit{match}[j,i] \equiv \mathit{match}[i,j]$

$\forall k \in \{0 \dots t{-}1\} :$

$\mathit{match}[i,j] = k \Leftrightarrow \sigma[i].\mathit{timeSlot}[k] = j \wedge \sigma[j]$ (C6)

$\forall i \in \{0 \dots n{-}2\} : \forall j \in \{i{+}1 \dots n{-}1\} :$

$\mathit{modMatch}[i,j] \in \{0 \dots f{-}1\}$ (V6)

$\mathit{modMatch}[j,i] \equiv \mathit{modMatch}[i,j]$

$\mathit{modMatch}[i,j] = \mathit{match}[i,j]/$ (C7)

$\forall i \in \{0 \dots n{-}1\} :$ allDiffere (C8)

$\forall \tau \in \{0 \dots t{-}1\} :$

$\mathit{time}[\tau] \in \{(0,1)$ (V7)

$\forall i \in \{0 \dots n$- $\dots n{-}1\} : \mathit{time}[\tau] = (i,j) \Leftrightarrow \mathit{match}[i,j] = \tau$ (C9)

$\mathit{totalBoatChang}$ $.\mathit{totalChanges}$ (V8)

$\text{minimise}(\mathit{totalBoa}$ $\mathit{anges})$ (C10)

Fig. 1. Our constraint model, from a skipper perspective (top) and a match and temporal perspective (below). The number of skippers is n, and m is the number of matches in a flight. The number of flights is $f = \lceil n(n-1)/m \rceil$, and there are $t = n(n-1)/2$ matches (and time slots) in total. We define $\mathbb{N}$ to include zero.

For an explanation at this level of detail, read the paper

The objecti stage 1 is used as a hard constraint:

total $\leq \beta$ (C11)

these variables and constraints is created for each skipper σ:

$\in \{0 \dots m{-}1\} : \forall j \in \{0 \dots f{-}1\} :$

$\mathit{position}[i,j] \in \{0,1\}$ (V9)

$\mathit{position}[i,j] = 1 \Leftrightarrow \mathit{timeSlot}[m.j + i] \geq 0$ (C12)

$\forall i \in \{0 \dots m{-}1\} :$

$\mathit{imbalance}[i] \in \mathbb{N}$ (V10)

$$\mathit{imbalance}[i] = \left| \frac{n-1}{m} - \sum_{j=0}^{f-1} \mathit{position}[i,j] \right| \quad \text{(C13)}$$

$\mathit{maxImbalance} \in \mathbb{N}$ (V11)

$\mathit{maxImbalance} = \text{maximum}(\mathit{imbalance})$ (C14)

We minimize the maximum imbalance over all skippers:

$\forall i \in \{0 \dots n{-}1\} : \mathit{imbalance}[i] \equiv \sigma[i].\mathit{maxImbalance}$ (V12)

$\mathit{maxImbalance} \in \mathbb{N}$ (V13)

$\mathit{maxImbalance} = \text{maximum}(\mathit{imbalance})$ (C15)

$\text{minimise}(\mathit{maxImbalance})$

Taking Mark Drummond's point of view (IJCAI93) this presentation is an advertisement for the paper

What follows is a sketch

Phase 1

Principal Criteria in Order of Priority:

1. Each skipper sails against each other skipper once.
2*. When skippers have an even number of matches, they have the same number of port and starboard assignments.
3*. When skippers have an odd number of matches, the first half of the skippers will have one more starboard assignment.
4. No skipper in the last match of a flight should be in the first match of the next flight.
5. No skipper should have more than two consecutive port or starboard assignments.
6. Each skipper should be assigned to match 1, match 2, etc. in a flight as equally as possible.
7. In flights with five or more matches, no skipper should be in the next-to-last match in a flight and then in the first match of the next flight.
8. If possible, a skipper should be starboard when meeting the nearest lowest ranked skipper (i.e. #1 will be starboard against #2, #3 will be starboard against #4).
9. Close-ranked skippers meet in the last flight.
10. Minimize the number of boat changes.
11. Skippers in the last match of a flight do not change boats.
12. Skippers in new boats do not sail in the first match of the next flight.
13. Skippers have a reasonable sequence of matches and blanks.

Note that criteria 10, 11 and 12 only apply when there are fewer boats than skippers; 11 and 12 override 6 when changes are required, and 13 applies when there are more boats than skippers.

Stage 1

What is a "boat change"?

10. Minimize the number of boat changes.
11. Skippers in the last match of a flight do not change boats.
12. Skippers in new boats do not sail in the first match of the next flight.

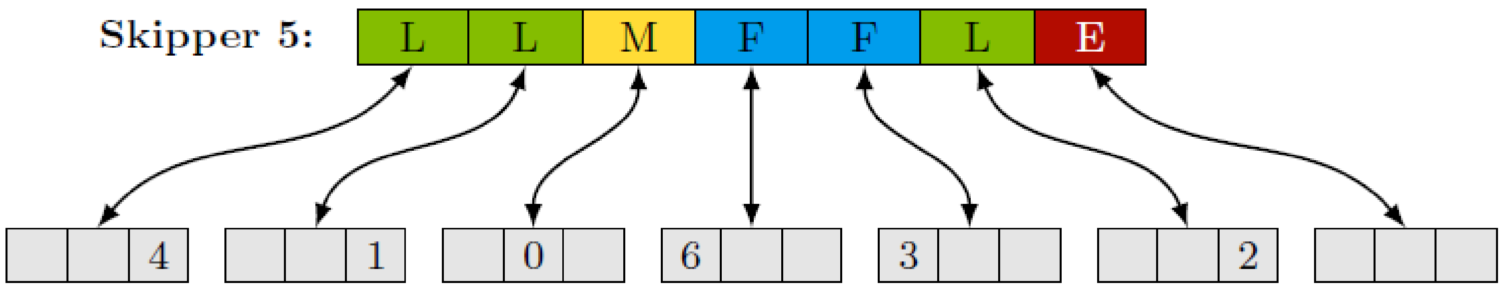
What is a "boat change"?

There are fewer boats than skippers and
A skipper gets into a boat for the 1st time ever (and is in a "new" boat) or …
A skipper has had a "bye" and then goes out into a flight (and is in a "new" boat)

10. Minimize the number of boat changes.
11. Skippers in the last match of a flight do not change boats.
12. Skippers in new boats do not sail in the first match of the next flight.

A skipper has
- a temporal view of his schedule (who do I race at time t?) and
- a state view (where am I racing in flight f?)

Modelling a Skipper



Flight	Matches		
0	(0,1)	(2,3)	**(4,5)**
1	(0,2)	(4,6)	**(1,5)**
2	(2,6)	**(0,5)**	(1,3)
3	**(5,6)**	(0,3)	(1,4)
4	**(3,5)**	(1,6)	(2,4)
5	(3,6)	(0,4)	**(2,5)**
6	(3,4)	(1,2)	(0,6)

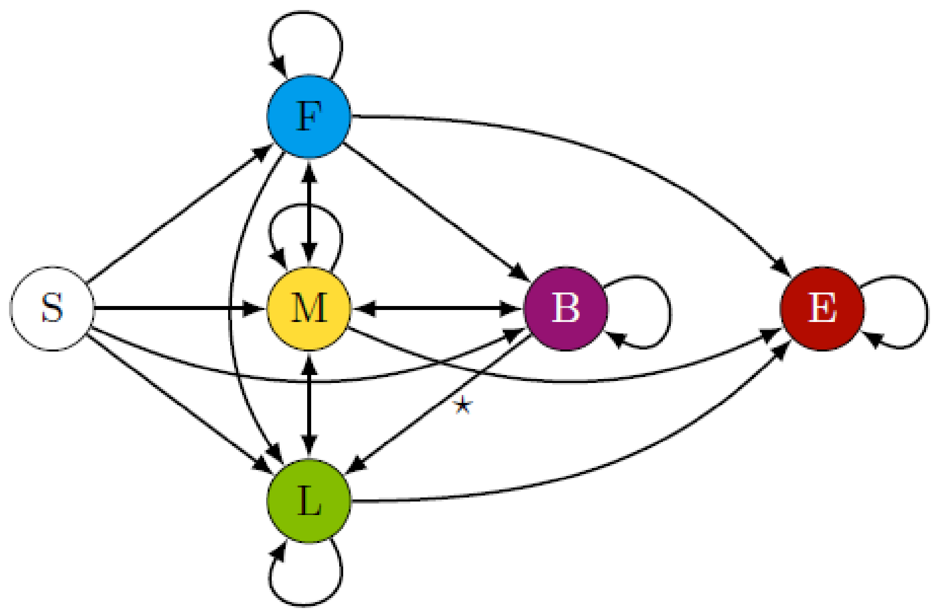
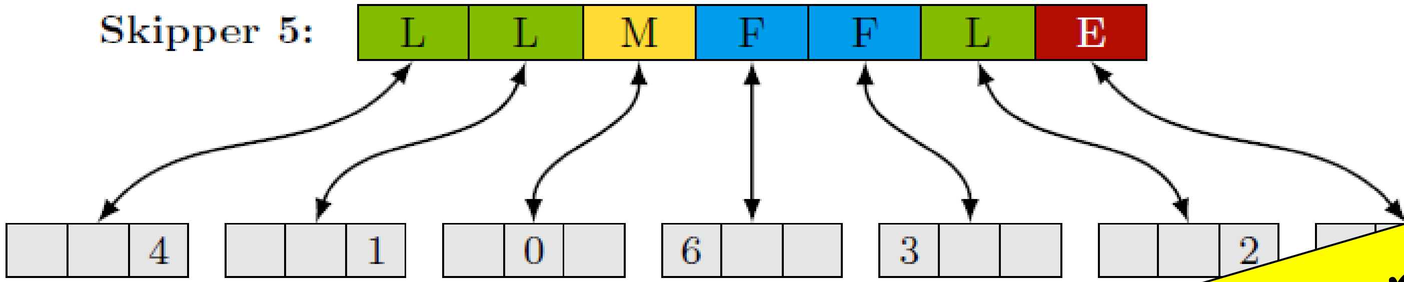


Fig. 2. A pictorial representation of a skipper (skipper 5) with multicoloured state and grey timeslots. The schedule for skipper 5 is in bold and the DFA for criteria 4, 11 and 12 is drawn with state START in white, FIRST in blue, MID in yellow, LAST in green, BYE in pink and END in red. The edge marked ⋆ is explained in the text.

Modelling a Skipper



Schedule for skipper 5

Flight	Matches		
0	(0,1)	(2,3)	**(4,5)**
1	(0,2)	(4,6)	**(1,5)**
2	(2,6)	**(0,5)**	(1,3)
3	**(5,6)**	(0,3)	(1,4)
4	**(3,5)**	(1,6)	(2,4)
5	(3,6)	(0,4)	**(2,5)**
6	(3,4)	(1,2)	(0,6)

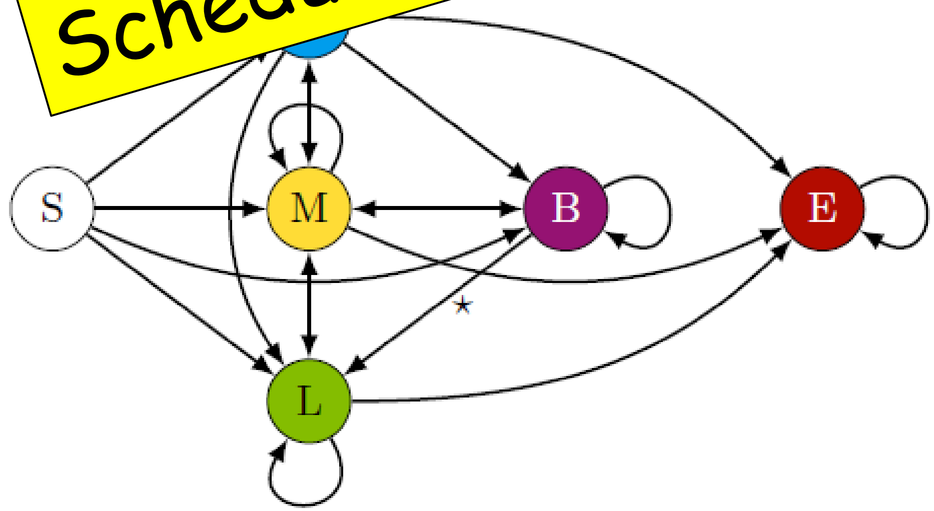


Fig. 2. A pictorial representation of a skipper (skipper 5) with multicoloured state and grey timeslots. The schedule for skipper 5 is in bold and the DFA for criteria 4, 11 and 12 is drawn with state START in white, FIRST in blue, MID in yellow, LAST in green, BYE in pink and END in red. The edge marked ⋆ is explained in the text.

Flight	Matches		
0	(0,1)	(2,3)	**(4,5)**
1	(0,2)	(4,6)	**(1,5)**
2	(2,6)	**(0,5)**	(1,3)
3	**(5,6)**	(0,3)	(1,4)
4	**(3,5)**	(1,6)	(2,4)
5	(3,6)	(0,4)	**(2,5)**
6	(3,4)	(1,2)	(0,6)

Schedule for skipper 5

Modelling a Skipper

Flight	Matches		
0	(0,1)	(2,3)	**(4,5)**
1	(0,2)	(4,6)	**(1,5)**
2	(2,6)	**(0,5)**	(1,3)
3	**(5,6)**	(0,3)	(1,4)
4	**(3,5)**	(1,6)	(2,4)
5	(3,6)	(0,4)	**(2,5)**
6	(3,4)	(1,2)	(0,6)

In flight 0
Competes against skipper 4 in last position

Modelling a Skipper

Flight	Matches		
0	(0,1)	(2,3)	**(4,5)**
1	(0,2)	(4,6)	**(1,5)**
2	(2,6)	**(0,5)**	(1,3)
3	**(5,6)**	(0,3)	(1,4)
4	**(3,5)**	(1,6)	(2,4)
5	(3,6)	(0,4)	**(2,5)**
6	(3,4)	(1,2)	(0,6)

In flight 1
Competes against skipper 1 in last position

Modelling a Skipper

Flight	Matches		
0	(0,1)	(2,3)	**(4,5)**
1	(0,2)	(4,6)	**(1,5)**
2	(2,6)	**(0,5)**	(1,3)
3	**(5,6)**	(0,3)	(1,4)
4	**(3,5)**	(1,6)	(2,4)
5	(3,6)	(0,4)	**(2,5)**
6	(3,4)	(1,2)	(0,6)

In flight 2
Competes against skipper 0 in middle position

Modelling a Skipper

Flight	Matches		
0	(0,1)	(2,3)	**(4,5)**
1	(0,2)	(4,6)	**(1,5)**
2	(2,6)	**(0,5)**	(1,3)
3	**(5,6)**	(0,3)	(1,4)
4	**(3,5)**	(1,6)	(2,4)
5	(3,6)	(0,4)	**(2,5)**
6	(3,4)	(1,2)	(0,6)

In flight 3
Competes against skipper 6 in first position

Modelling a Skipper

Flight	Matches		
0	(0,1)	(2,3)	**(4,5)**
1	(0,2)	(4,6)	**(1,5)**
2	(2,6)	**(0,5)**	(1,3)
3	**(5,6)**	(0,3)	(1,4)
4	**(3,5)**	(1,6)	(2,4)
5	(3,6)	(0,4)	**(2,5)**
6	(3,4)	(1,2)	(0,6)

In flight 4
Competes against skipper 3 in first position

Modelling a Skipper

Flight	Matches		
0	(0,1)	(2,3)	$\boxed{\mathbf{(4,5)}}$
1	(0,2)	(4,6)	$\boxed{\mathbf{(1,5)}}$
2	(2,6)	$\boxed{\mathbf{(0,5)}}$	(1,3)
3	$\boxed{\mathbf{(5,6)}}$	(0,3)	(1,4)
4	$\boxed{\mathbf{(3,5)}}$	(1,6)	(2,4)
5	(3,6)	(0,4)	$\boxed{\mathbf{(2,5)}}$
6	(3,4)	(1,2)	(0,6)

In flight 5
Competes against skipper 2 in last position

Modelling a Skipper

Flight	Matches		
0	(0,1)	(2,3)	**(4,5)**
1	(0,2)	(4,6)	**(1,5)**
2	(2,6)	**(0,5)**	(1,3)
3	**(5,6)**	(0,3)	(1,4)
4	**(3,5)**	(1,6)	(2,4)
5	(3,6)	(0,4)	**(2,5)**
6	(3,4)	(1,2)	(0,6)

Not in flight 6
This is the end

Modelling a Skipper

Flight	Matches		
0	(0,1)	(2,3)	**(4,5)**
1	(0,2)	(4,6)	**(1,5)**
2	(2,6)	**(0,5)**	(1,3)
3	**(5,6)**	(0,3)	(1,4)
4	**(3,5)**	(1,6)	(2,4)
5	(3,6)	(0,4)	**(2,5)**
6	(3,4)	(1,2)	(0,6)

And yes … (x,y) is different from (y,x)!

(x,y) x is port

(x,y) y is starboard

Modelling a Skipper

Flight	Matches		
0	(0,1)	(2,3)	**(4,5)**
1	(0,2)	(4,6)	**(1,5)**
2	(2,6)	**(0,5)**	(1,3)
3	**(5,6)**	(0,3)	(1,4)
4	**(3,5)**	(1,6)	(2,4)
5	(3,6)	(0,4)	**(2,5)**
6	(3,4)	(1,2)	(0,6)

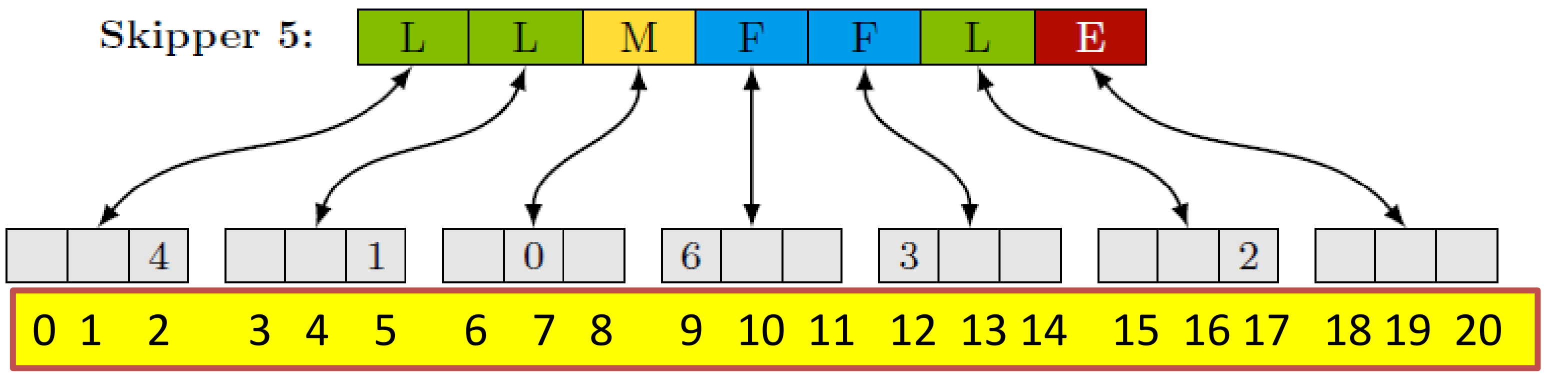
And yes … (x,y) is different from (y,x)!

(x,y) x is port

(x,y) y is starboard

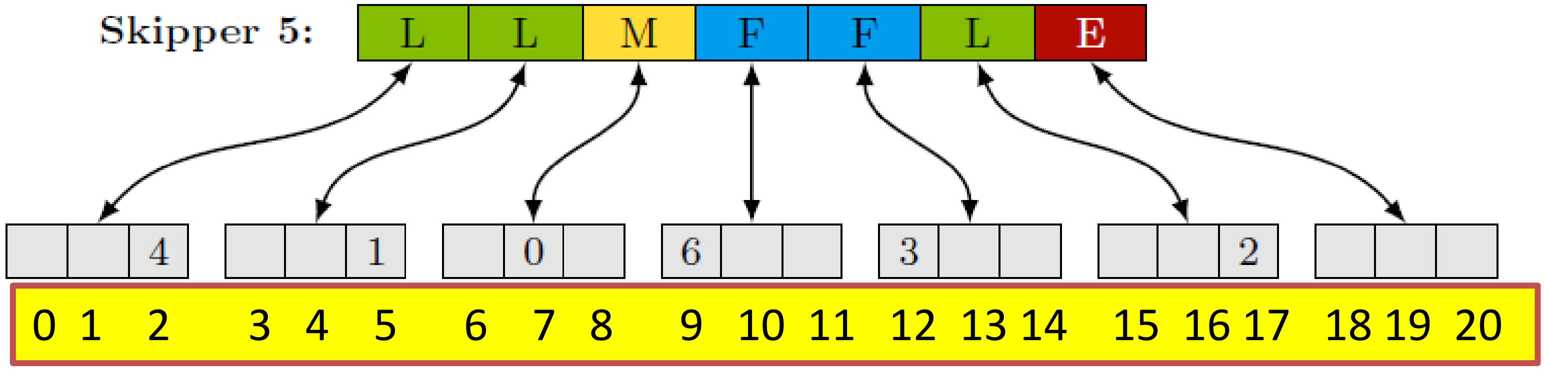
There's no red Port left in the bottle

Modelling a Skipper



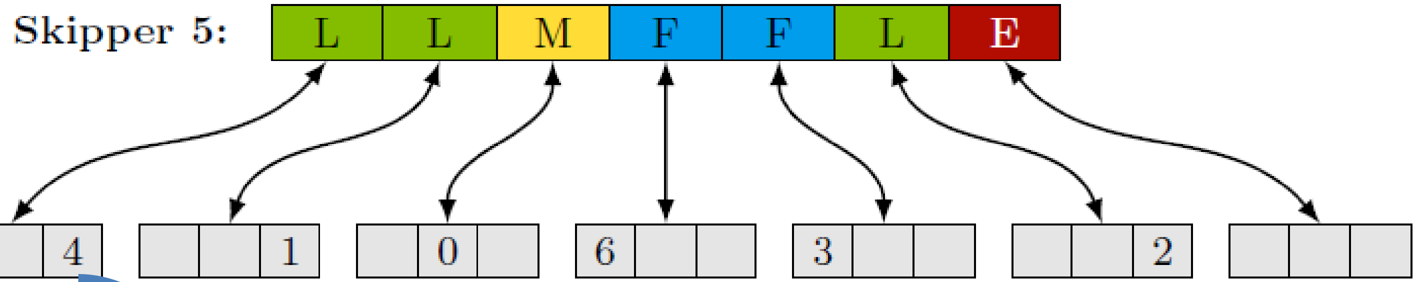
Time slots, in this case divide by 3 to find out flight

Modelling a Skipper



Value in timeslot is the skipper we compete against

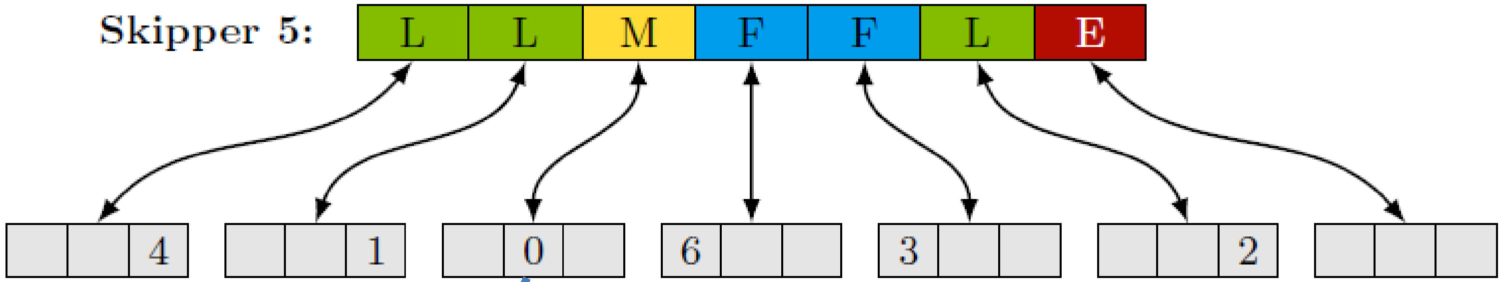
Modelling a Skipper



Skipper 5 is in Last place in flight 0 and competes with skipper 4

Flight	Matches		
0	(0,1)	(2,3)	**(4,5)**
1	(0,2)	(4,6)	**(1,5)**
2	(2,6)	**(0,5)**	(1,3)
3	**(5,6)**	(0,3)	(1,4)
4	**(3,5)**	(1,6)	(2,4)
5	(3,6)	(0,4)	**(2,5)**
6	(3,4)	(1,2)	(0,6)

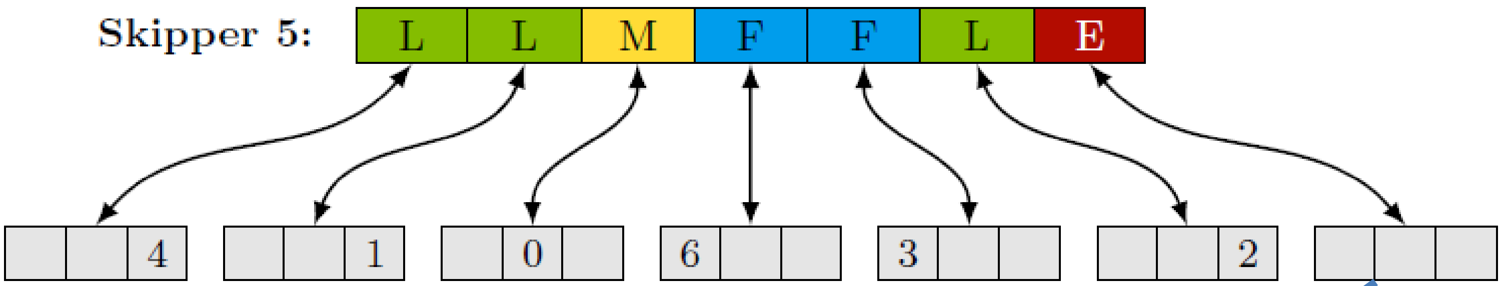
Modelling a Skipper



Skipper 5 is in Middle place in flight 2 and competes with skipper 0

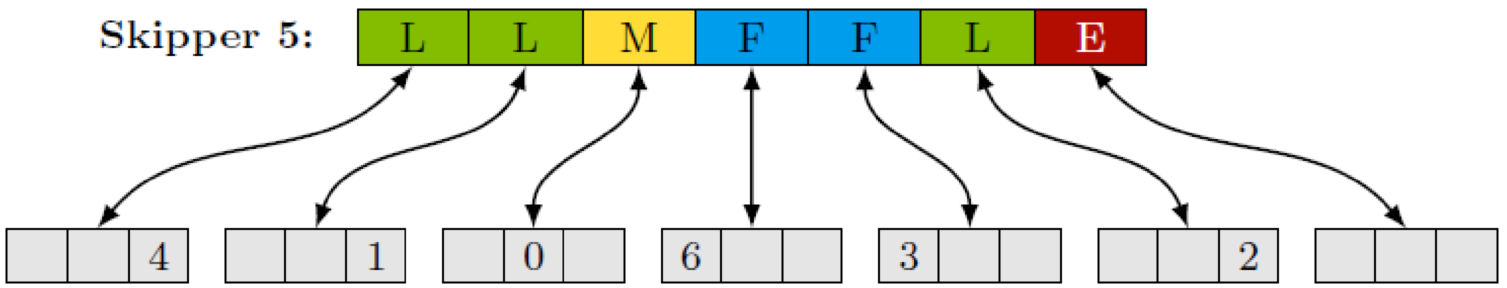
Flight	Matches		
0	(0,1)	(2,3)	**(4,5)**
1	(0,2)	(4,6)	**(1,5)**
2	(2,6)	**(0,5)**	(1,3)
3	**(5,6)**	(0,3)	(1,4)
4	**(3,5)**	(1,6)	(2,4)
5	(3,6)	(0,4)	**(2,5)**
6	(3,4)	(1,2)	(0,6)

Modelling a Skipper



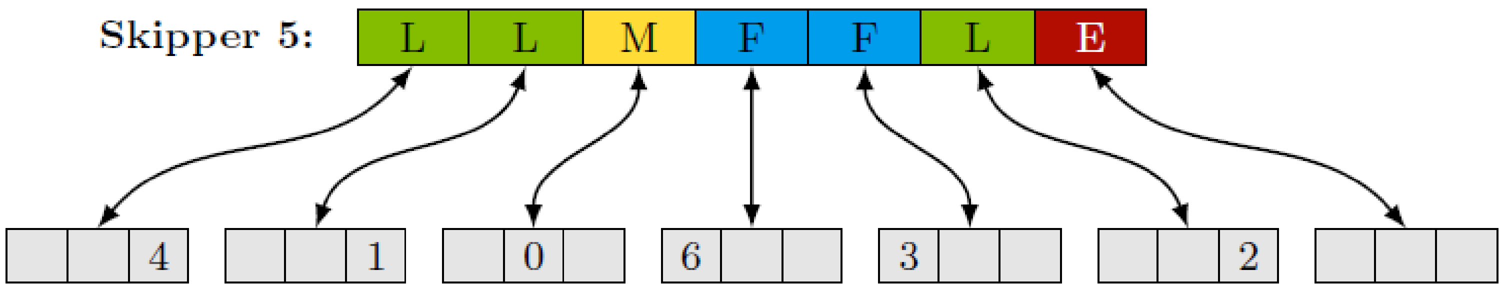
Flight	Matches		
0	(0,1)	(2,3)	**(4,5)**
1	(0,2)	(4,6)	**(1,5)**
2	(2,6)	**(0,5)**	(1,3)
3	**(5,6)**	(0,3)	(1,4)
4	**(3,5)**	(1,6)	(2,4)
5	(3,6)	(0,4)	**(2,5)**
6	(3,4)	(1,2)	(0,6)

Skipper 5 is in the End state in flight 6



Channel between timeslots and state!

Modelling a Skipper



Channel between timeslots and state!

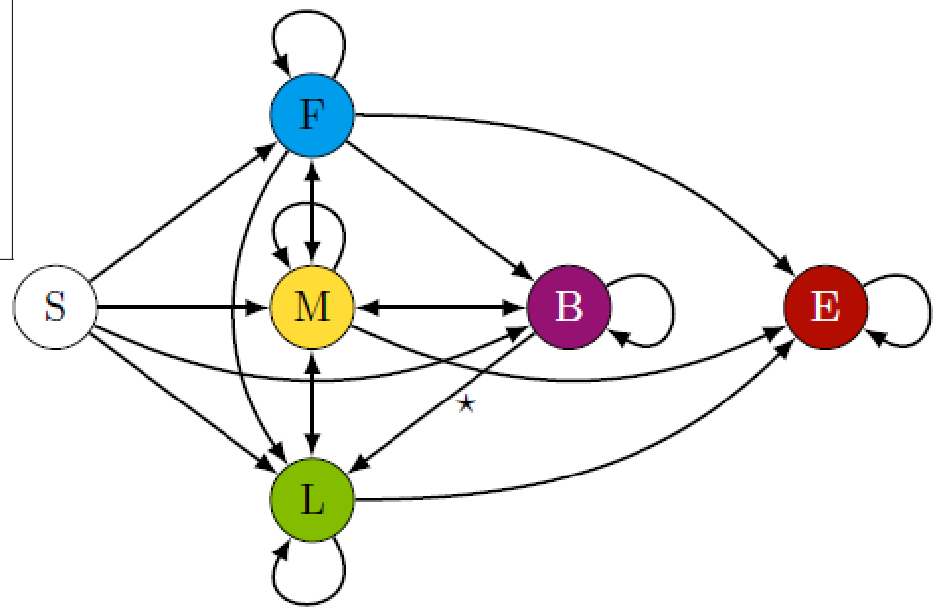
state is Last, First, Mid, Bye or End, i.e. position in flight

Modelling a Skipper

Principal Criteria in Order of Priority:

1. Each skipper sails against each other skipper once.
2*. When skippers have an even number of matches, they have the same number of port and starboard assignments.
3*. When skippers have an odd number of matches, the first half of the skippers will have one more starboard assignment.
4. No skipper in the last match of a flight should be in the first match of the next flight.
5. No skipper should have more than two consecutive port or starboard assignments.
6. Each skipper should be assigned to match 1, match 2, etc. in a flight as equally as possible.
7. In flights with five or more matches, no skipper should be in the next-to-last match in a flight and then in the first match of the next flight.
8. If possible, a skipper should be starboard when meeting the nearest lowest ranked skipper (i.e. #1 will be starboard against #2, #3 will be starboard against #4).
9. Close-ranked skippers meet in the last flight.
10. Minimize the number of boat changes.
11. Skippers in the last match of a flight do not change boats.
12. Skippers in new boats do not sail in the first match of the next flight.
13. Skippers have a reasonable sequence of matches and blanks.

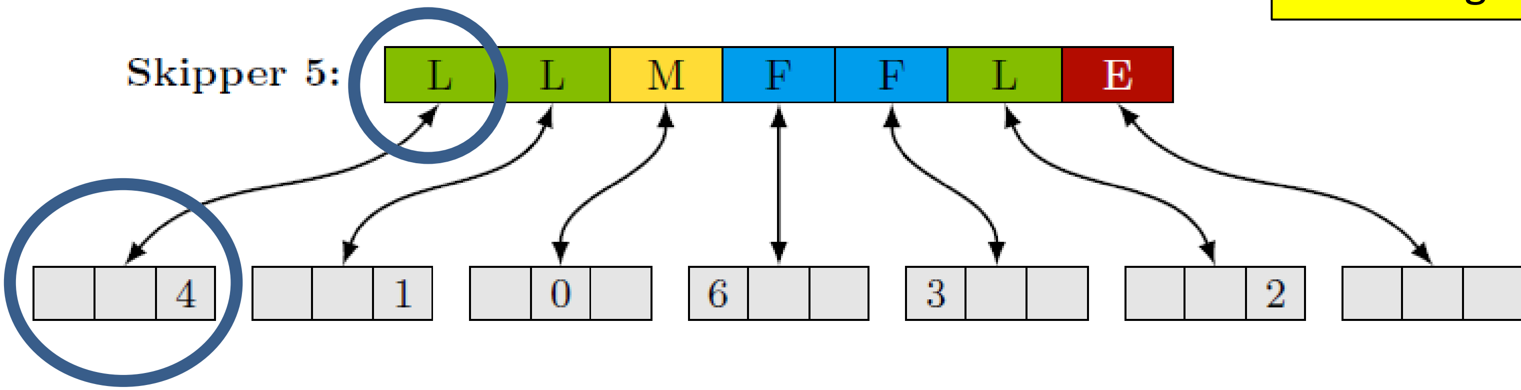
Note that criteria 10, 11 and 12 only apply when there are fewer boats than skippers; 11 and 12 override 6 when changes are required, and 13 applies when there are more boats than skippers.



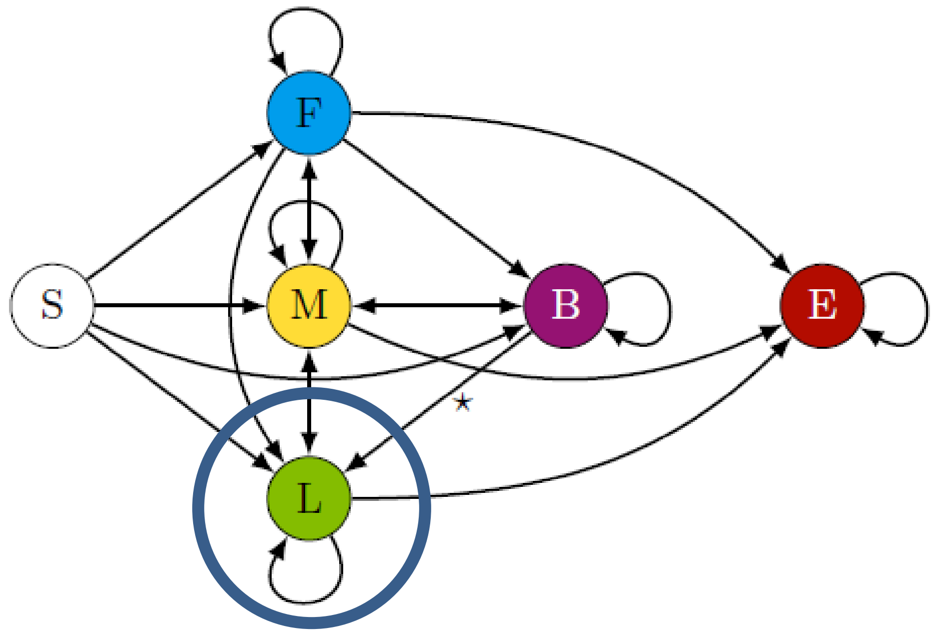
Finite Automata to describe criteria 4, 11 and 12

Regular **constraint acts on skipper's state**

Modelling a Skipper



Flight	Matches		
0	(0,1)	(2,3)	**(4,5)**
1	(0,2)	(4,6)	(1,5)
2	(2,6)	**(0,5)**	(1,3)
3	**(5,6)**	(0,3)	(1,4)
4	**(3,5)**	(1,6)	(2,4)
5	(3,6)	(0,4)	**(2,5)**
6	(3,4)	(1,2)	(0,6)



Skipper is last in flight 0 (green) and can move in next flight to states Last, Mid or End i.e. cannot go into a Bye or go First

We also have

- array match[i][j] = timeslot for match (i,j)
- time[t] is a pair (i,j), where match (i,j) is in timeslot t

These are channelled into skippers
Time is decision variable (what do we do now?)

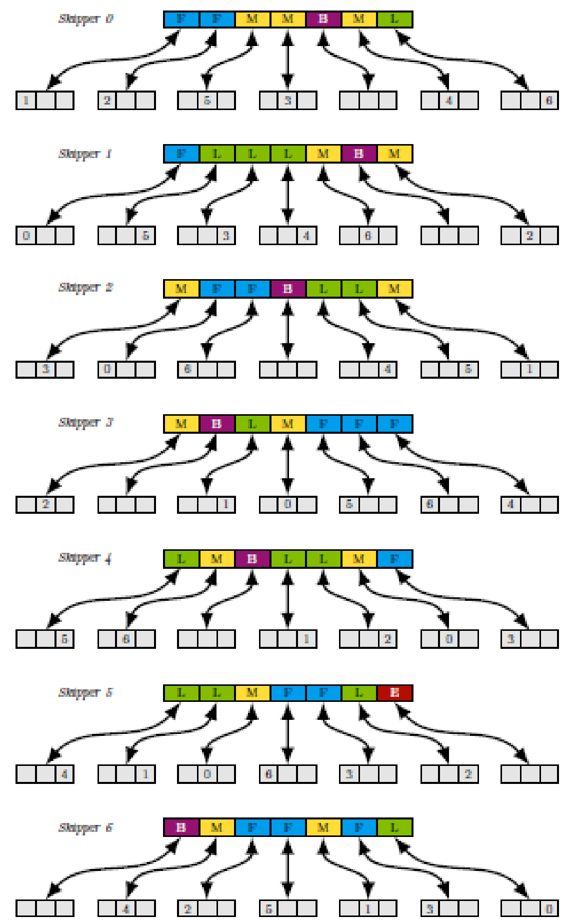
7 Skippers and 6 Boats: phase 1

modMatch

	0	1	2	3	4	5	6
0	-	0	1	3	5	2	6
1	0	-	6	2	3	1	4
2	1	6	-	0	4	5	2
3	3	2	0	-	6	4	5
4	5	3	4	6	-	0	1
5	2	1	5	4	0	-	3
6	6	4	2	5	1	3	-

match

	0	1	2	3	4	5	6
0	-	0	3	10	16	7	20
1		-	19	8	11	5	13
2			-	1	14	17	6
3				-	18	12	15
4					-	2	4
5						-	9
6							-



schedule

Flight	Matches		
0	(0,1)	(2,3)	(4,5)
1	(0,2)	(4,6)	(1,5)
2	(2,6)	(0,5)	(1,3)
3	(5,6)	(0,3)	(1,4)
4	(3,5)	(1,6)	(2,4)
5	(3,6)	(0,4)	(2,5)
6	(3,4)	(1,2)	(0,6)

time

(0,1)	(2,3)	(4,5)

(0,2)	(4,6)	(1,5)

(2,6)	(0,5)	(1,3)

(5,6)	(0,3)	(1,4)

(3,5)	(1,6)	(2,4)

(3,6)	(0,4)	(2,5)

(3,4)	(1,2)	(0,6)

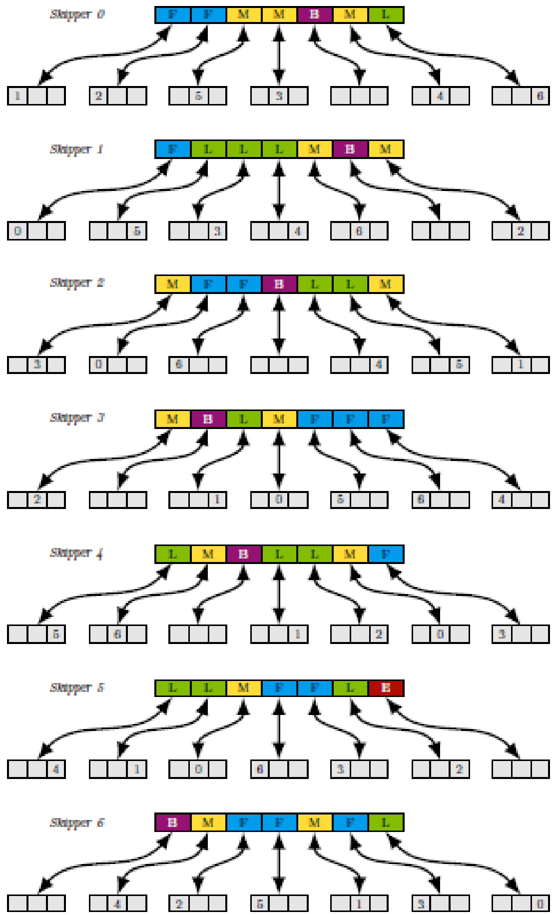
7 Skippers and 6 Boats: phase 1

modMatch

	0	1	2	3	4	5	6
0	-	0	1	3	5	2	6
1	0	-	6	2	3	1	4
2	1	6	-	0	4	5	2
3	3	2	0	-	6	4	5
4	5	3	4	6	-	0	1
5	2	1	5	4	0	-	3
6	6	4	2	5	1	3	-

match

	0	1	2	3	4	5	6
0	-	0	3	10	16	7	20
1		-	19	8	11	5	13
2			-	1	14	17	6
3				-	18	12	15
4					-	2	4
5						-	9
6							-



schedule

Flight	Matches		
0	(0,1)	(2,3)	(4,5)
1	(0,2)	(4,6)	(1,5)
2	(2,6)	(0,5)	(1,3)
3	(5,6)	(0,3)	(1,4)
4	(3,5)	(1,6)	(2,4)
5	(3,6)	(0,4)	(2,5)
6	(3,4)	(1,2)	(0,6)

time

(0,1) (2,3) (4,5)	(0,2) (4,6) (1,5)	(2,6) (0,5) (1,3)	(5,6) (0,3) (1,4)	(3,5) (1,6) (2,4)	(3,6) (0,4) (2,5)	(3,4) (1,2) (0,6)

modMatch converts timeslots in match into flights, forces skipper's matches into different flights

7 Skippers and 6 Boats: phase 1

Symmetry break at top of search … first flight contains matches (0,1), (2,3), (4,5)

Phase 1, minimise boat changes (total or worst case skipper)

Output the integer … number of boat changes, as input to Phase 2

Phase 2

Minimise imbalance

Phase 2

Given the number of boat changes, produce a schedule that is "balanced" and has at most the given number of boat changes.

Balance of a skipper is balance between … number of times first, number of times middle, number of times last

Balance for schedule, is balance of worst skipper

Phase 2

Given the number of boat changes, produce a schedule that is "balanced" and has at most the given number of boat changes.

Balance of a skipper is balance between … number of times first, number of times middle, number of times last

Balance for schedule, is balance of worst skipper

Read the paper for details (I'm running out of time)

Phase 3

Renumbering

Phase 3

Principal Criteria in Order of Priority:

1. Each skipper sails against each other skipper once.
2*. When skippers have an even number of matches, they have the same number of port and starboard assignments.
3*. When skippers have an odd number of matches, the first half of the skippers will have one more starboard assignment.
4. No skipper in the last match of a flight should be in the first match of the next flight.
5. No skipper should have more than two consecutive port or starboard assignments.
6. Each skipper should be assigned to match 1, match 2, etc. in a flight as equally as possible.
7. In flights with five or more matches, no skipper should be in the next-to-last match in a flight and then in the first match of the next flight.
8. If possible, a skipper should be starboard when meeting the nearest lowest ranked skipper (i.e. #1 will be starboard against #2, #3 will be starboard against #4).
9. Close-ranked skippers meet in the last flight.
10. Minimize the number of boat changes.
11. Skippers in the last match of a flight do not change boats.
12. Skippers in new boats do not sail in the first match of the next flight.
13. Skippers have a reasonable sequence of matches and blanks.

Note that criteria 10, 11 and 12 only apply when there are fewer boats than skippers; 11 and 12 override 6 when changes are required, and 13 applies when there are more boats than skippers.

Phase 3

Skippers 0 and 1 are in a match in the last flight

Flight	Matches		
0	(0,1)	(2,3)	(4,5)
1	(0,2)	(4,6)	(1,5)
2	(2,6)	(0,5)	(1,3)
3	(5,6)	(0,3)	(1,4)
4	(3,5)	(1,6)	(2,4)
5	(3,6)	(0,4)	(2,5)
6	(3,4)	(1,2)	(0,6)

Flight	Matches		
0	(0,6)	(2,3)	(4,5)
1	(0,2)	(4,1)	(6,5)
2	(2,1)	(0,5)	(6,3)
3	(5,1)	(0,3)	(6,4)
4	(3,5)	(6,1)	(2,4)
5	(3,1)	(0,4)	(2,5)
6	(3,4)	(6,2)	(0,1)

In the above, swap all 0's with 0's and swap all 6's with 1's so last match is (0,1)

9. Close-ranked skippers meet in the last flight.

NOT A CP STEP

Phase 4

Orientation
(port/starboard)

Ortientation

Phase 4

Principal Criteria in Order of Priority:

1. Each skipper sails against each other skipper once.

2*. When skippers have an even number of matches, they have the same number of port and starboard assignments.

3*. When skippers have an odd number of matches, the first half of the skippers will have one more starboard assignment.

4. No skipper in the last match of a flight should be in the first match of the next flight.

5. No skipper should have more than two consecutive port or starboard assignments.

6. Each skipper should be assigned to match 1, match 2, etc. in a flight as equally as possible.

7. In flights with five or more matches, no skipper should be in the next-to-last match in a flight and then in the first match of the next flight.

8. If possible, a skipper should be starboard when meeting the nearest lowest ranked skipper (i.e. #1 will be starboard against #2, #3 will be starboard against #4).

9. Close-ranked skippers meet in the last flight.

10. Minimize the number of boat changes.

11. Skippers in the last match of a flight do not change boats.

12. Skippers in new boats do not sail in the first match of the next flight.

13. Skippers have a reasonable sequence of matches and blanks.

Note that criteria 10, 11 and 12 only apply when there are fewer boats than skippers; 11 and 12 override 6 when changes are required, and 13 applies when there are more boats than skippers.

Phase 4

Reads in the renumbered schedule and …

orient[i][j] = 0 iff skipper i is on port side in his jth match
orient[i][j] = 1 iff skipper i is on starboard side in his jth match

Phase 4

orient[i][j] = 0 iff skipper i is on port side in his jth match
orient[i][j] = 1 iff skipper i is on starboard side in his jth match

Flight	Matches		
0	(0,6)	(2,3)	(4,5)
1	(0,2)	(4,1)	(6,5)
2	(2,1)	(0,5)	(6,3)
3	(5,1)	(0,3)	(6,4)
4	(3,5)	(6,1)	(2,4)
5	(3,1)	(0,4)	(2,5)
6	(3,4)	(6,2)	(0,1)

Skipper 2 meets skippers in the order 3,0,1,4,5,6
Skipper 3 meets skippers in the order 2,6,0,5,1,4
Skipper 5 meets skippers in the order 4,6,0,1,3,2

Phase 4

orient[i][j] = 0 iff skipper i is on port side in his jth match
orient[i][j] = 1 iff skipper i is on starboard side in his jth match

Flight	Matches		
0	(0,6)	(2,3)	(4,5)
1	(0,2)	(4,1)	(6,5)
2	(2,1)	(0,5)	(6,3)
3	(5,1)	(0,3)	(6,4)
4	(3,5)	(6,1)	(2,4)
5	(3,1)	(0,4)	(2,5)
6	(3,4)	(6,2)	(0,1)

Skipper 2 meets skippers in the order 3,0,1,4,5,6
Skipper 3 meets skippers in the order 2,6,0,5,1,4
Skipper 5 meets skippers in the order 4,6,0,1,3,2

Phase 4

orient[i][j] = 0 iff skipper i is on port side in his jth match
orient[i][j] = 1 iff skipper i is on starboard side in his jth match

Flight	Matches		
0	(0,6)	(2,3)	(4,5)
1	(0,2)	(4,1)	(6,5)
2	(2,1)	(0,5)	(6,3)
3	(5,1)	(0,3)	(6,4)
4	(3,5)	(6,1)	(2,4)
5	(3,1)	(0,4)	(2,5)
6	(3,4)	(6,2)	(0,1)

Skipper 2 meets skippers in the order 3,0,1,4,5,6
Skipper 3 meets skippers in the order 2,6,0,5,1,4
Skipper 5 meets skippers in the order 4,6,0,1,3,2

orient[i][j] = 0 iff skipper i is on port side in his jth match
orient[i][j] = 1 iff skipper i is on starboard side in his jth match

Flight	Matches		
0	(0,6)	(2,3)	(4,5)
1	(0,2)	(4,1)	(6,5)
2	(2,1)	(0,5)	(6,3)
3	(5,1)	(0,3)	(6,4)
4	(3,5)	(6,1)	(2,4)
5	(3,1)	(0,4)	(2,5)
6	(3,4)	(6,2)	(0,1)

Skipper 2 meets skippers in the order 3,0,1,4,5,6
Skipper 3 meets skippers in the order 2,6,0,5,1,4
Skipper 5 meets skippers in the order 4,6,0,1,3,2

Phase 4

orient[i][j] = 0 iff skipper i is on port side in his jth match
orient[i][j] = 1 iff skipper i is on starboard side in his jth match

Flight	Matches		
0	(0,6)	(2,3)	(4,5)
1	(0,2)	(4,1)	(6,5)
2	(2,1)	(0,5)	(6,3)
3	(5,1)	(0,3)	(6,4)
4	(3,5)	(6,1)	(2,4)
5	(3,1)	(0,4)	(2,5)
6	(3,4)	(6,2)	(0,1)

Skipper 2 meets skippers in the order 3,0,1,4,5,6
Skipper 3 meets skippers in the order 2,6,0,5,1,4
Skipper 5 meets skippers in the order 4,6,0,1,3,2

Phase 4

orient[i][j] = 0 iff skipper i is on port side in his jth match
orient[i][j] = 1 iff skipper i is on starboard side in his jth match

Flight	Matches		
0	(0,6)	(2,3)	(4,5)
1	(0,2)	(4,1)	(6,5)
2	(2,1)	(0,5)	(6,3)
3	(5,1)	(0,3)	(6,4)
4	(3,5)	(6,1)	(2,4)
5	(3,1)	(0,4)	(2,5)
6	(3,4)	(6,2)	(0,1)

Skipper 2 meets skippers in the order 3,0,1,4,5,6
Skipper 3 meets skippers in the order 2,6,0,5,1,4
Skipper 5 meets skippers in the order 4,6,0,1,3,2

Phase 4

orient[i][j] = 0 iff skipper i is on port side in his jth match
orient[i][j] = 1 iff skipper i is on starboard side in his jth match

Flight	Matches		
0	(0,6)	(2,3)	(4,5)
1	(0,2)	(4,1)	(6,5)
2	(2,1)	(0,5)	(6,3)
3	(5,1)	(0,3)	(6,4)
4	(3,5)	(6,1)	(2,4)
5	(3,1)	(0,4)	(2,5)
6	(3,4)	(6,2)	(0,1)

Skipper 2 meets skippers in the order 3,0,1,4,5,6
Skipper 3 meets skippers in the order 2,6,0,5,1,4
Skipper 5 meets skippers in the order 4,6,0,1,3,2

Phase 4

Reads in the renumbered schedule

orient[i][j] = 0 iff skipper i is on port side in his jth match
orient[i][j] = 1 iff skipper i is on starboard side in his jth match

Flight	Matches		
0	(0,6)	(2,3)	(4,5)
1	(0,2)	(4,1)	(6,5)
2	(2,1)	(0,5)	(6,3)
3	(5,1)	(0,3)	(6,4)
4	(3,5)	(6,1)	(2,4)
5	(3,1)	(0,4)	(2,5)
6	(3,4)	(6,2)	(0,1)

Skipper 2 meets skippers in the order 3,0,1,4,5,6
Skipper 3 meets skippers in the order 2,6,0,5,1,4
Skipper 5 meets skippers in the order 4,6,0,1,3,2

Therefore
orient[2][0] = 1 and orient[3][0] = 0 (starboard when meeting nearest lowest ranked skipper)
orient[2][4] ≠ orient[5][5] (match between 2 and 5)
orient[3][3] ≠ orient[5][4] (match between 3 and 5)

Phase 4

Reads in the renumbered schedule

orient[i][j] = 0 iff skipper i is on port side in his jth match
orient[i][j] = 1 iff skipper i is on starboard side in his jth match

Flight	Matches		
0	(0,6)	(2,3)	(4,5)
1	(0,2)	(4,1)	(6,5)
2	(2,1)	(0,5)	(6,3)
3	(5,1)	(0,3)	(6,4)
4	(3,5)	(6,1)	(2,4)
5	(3,1)	(0,4)	(2,5)
6	(3,4)	(6,2)	(0,1)

Skipper 2 meets skippers in the order 3,0,1,4,5,6
Skipper 3 meets skippers in the order 2,6,0,5,1,4
Skipper 5 meets skippers in the order 4,6,0,1,3,2

Therefore
orient[2][0] = 1 and orient[3][0] = 0 (starboard when meeting nearest lowest ranked skipper)
orient[2][4] ≠ orient[5][5] (match between 2 and 5)
orient[3][3] ≠ orient[5][4] (match between 3 and 5)

Phase 4

Reads in the renumbered schedule

orient[i][j] = 0 iff skipper i is on port side in his jth match
orient[i][j] = 1 iff skipper i is on starboard side in his jth match

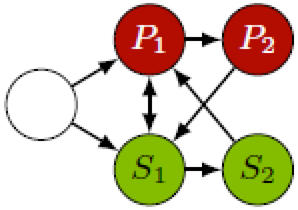
Flight	Matches		
0	(0,6)	(2,3)	(4,5)
1	(0,2)	(4,1)	(6,5)
2	(2,1)	(0,5)	(6,3)
3	(5,1)	(0,3)	(6,4)
4	(3,5)	(6,1)	(2,4)
5	(3,1)	(0,4)	(2,5)
6	(3,4)	(6,2)	(0,1)

Skipper 2 meets skippers in the order 3,0,1,4,5,6
Skipper 3 meets skippers in the order 2,6,0,5,1,4
Skipper 5 meets skippers in the order 4,6,0,1,3,2

Therefore
orient[2][0] = 1 and orient[3][0] = 0 (starboard when meeting nearest lowest ranked skipper)
orient[2][4] ≠ orient[5][5] (match between 2 and 5)
<u>orient[3][3] ≠ orient[5][4] (match between 3 and 5)</u>

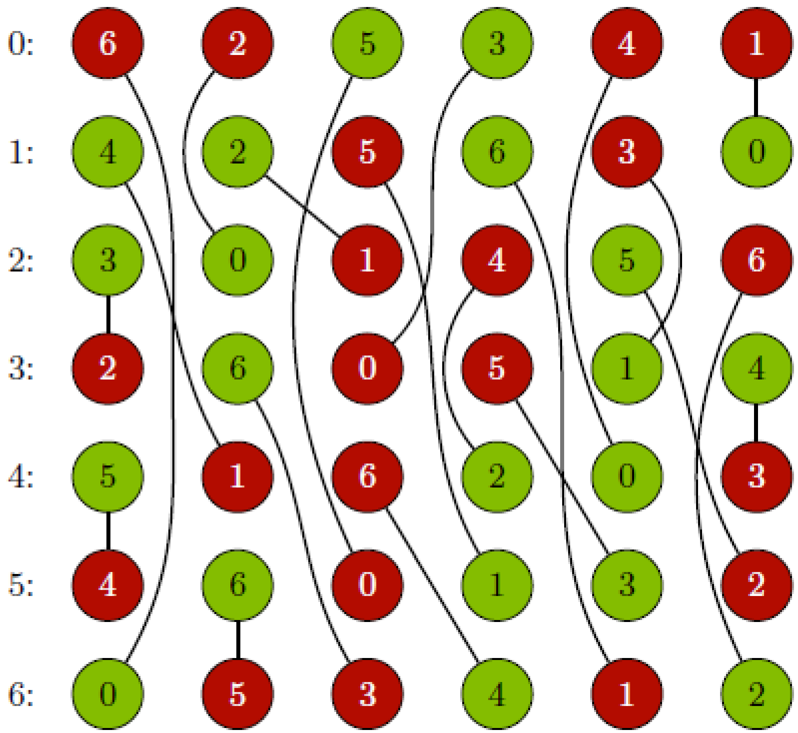
Phase 4

Regular constraint for a skipper (row of array orient)

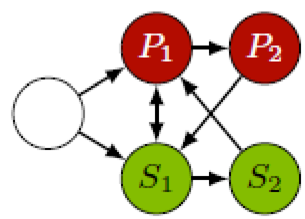


5. No skipper should have more than two consecutive port or starboard assignments.

Phase 4



Flight	Matches		
0	(0,6)	(3,2)	(5,4)
1	(0,2)	(4,1)	(6,5)
2	(2,1)	(5,0)	(6,3)
3	(1,5)	(3,0)	(4,6)
4	(3,5)	(6,1)	(2,4)
5	(1,3)	(0,4)	(5,2)
6	(4,3)	(2,6)	(1,0)



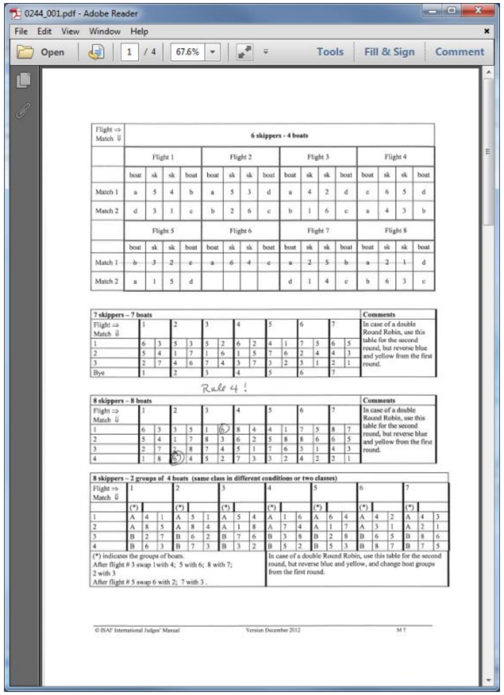
(x,y) x is port

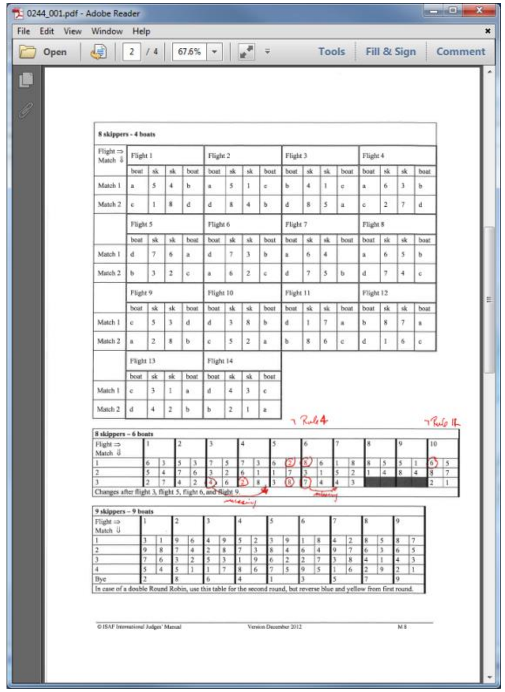
(x,y) y is starboard

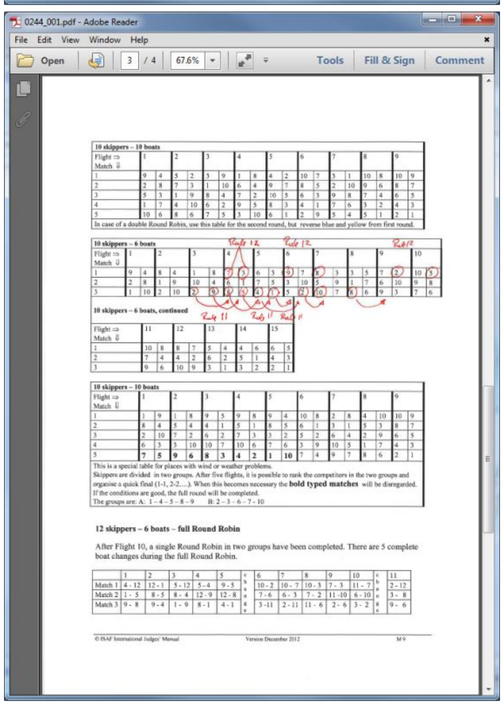
So?

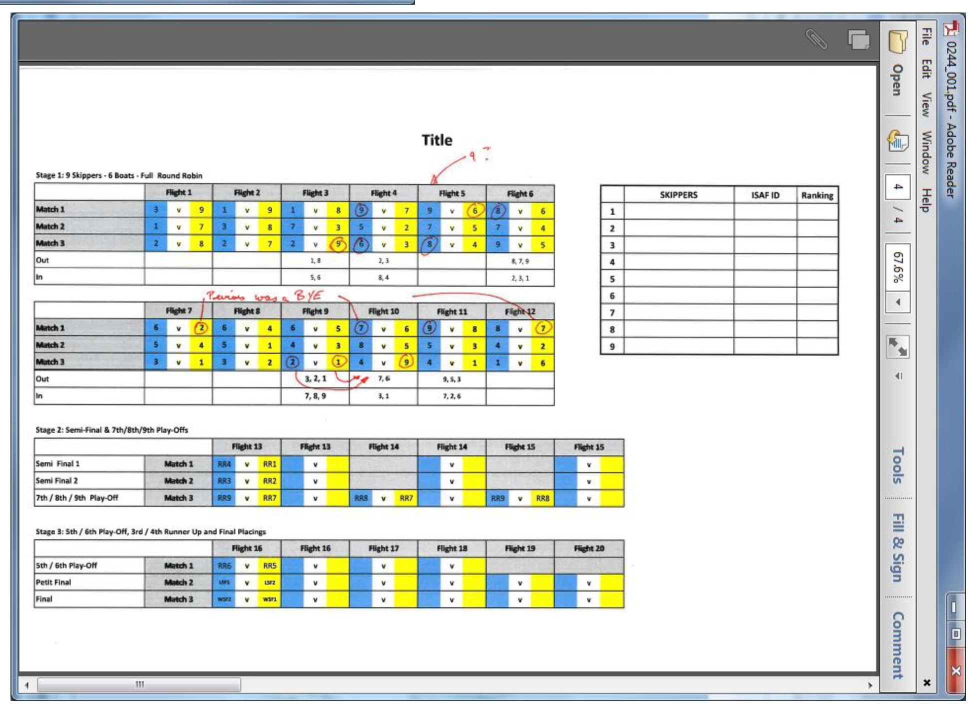
Published Schedules

So?









Many schedules violate the published criteria.
Many schedules are missing.
Many schedules are very poor (wrt boat changes)

So?

Our schedules are

- Good (generally better than published)
- Legal!
- Take a long time to produce (allow days for optimisation, why not)

So?

Our schedules are

- Good (generally better than published)
- Legal!
- Take a long time to produce (allow days for optimisation, why not)

Technical Issues

- Modelling was not easy, we had communications problems, diagrams and DFA helped greatly
- CP allowed nice incremental development
- Phase 4 (orientation) is always easy and we don’t know why
- Implemented a separate system to validate schedules, recognising criteria violations
- Implemented non-CP for phase 1 to validate CP and test out heuristics and symmetry breaking

So?

Our schedules are

- Good (generally better than published)
- Legal!
- Take a long time to produce (allow days for optimisation, why not)

Technical Issues

- Modelling was not easy, we had communications problems, diagrams and DFA helped greatly
- CP allowed nice incremental development
- Phase 4 (orientation) is always easy and we don't know why
- Implemented a separate system to validate schedules, recognising criteria violations
- Implemented non-CP for phase 1 to validate CP and test out heuristics and symmetry breaking

Practical Application

- Schedules will be used in West of Scotland
- Liaising with international umpires, accept schedules nationally and internationally
- May be able to investigate interaction between criteria

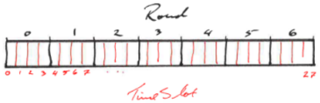
SKIPPER

TIMESLOT

n = 8   m = 4   boats = 8

$\frac{8.7}{2}$ = 28 matches

7 rounds



One of these for each skipper

Rows

skipper[i][k] = x

x ∈ {-2, -1, 0.. n-1} \ {i}

x ≥ 0 ⟷ skipper i is in match with x at time slot k

x = -2   done

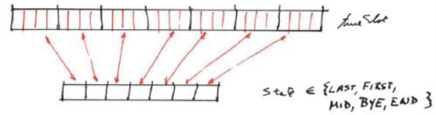
x = -1   nothing happening in this time slot for this skipper.

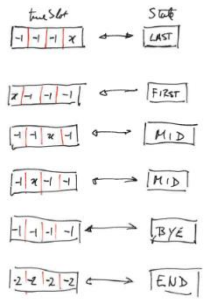
For a single skipper

n = 8 m = 4

Skipper

STATE



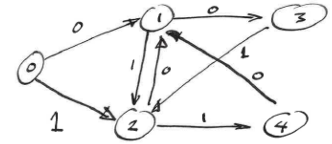


Global cardinality to ensure skipper has assignments {0..n-1} \ {i}

DFA applied to state

Deterministic Finite Automaton

a	b	c
0	0	1
0	1	2
1	0	3
1	1	2
2	1	4
2	0	1
3	1	2
4	0	1



accepting states {1, 2, 3, 4}

do not allow 111 or 000

a: state

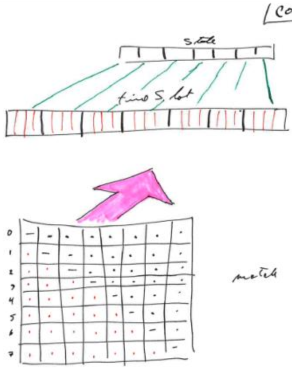
b: input

c: new state

ORIENTING

0: PORT   1: STARBOARD

COMBINED



match[i][j] = k ⟷ timeSlot[i][k] = j

for all i, j, k

match — decision variables

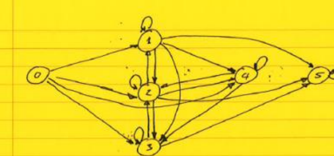
DVO — sdf

Valo — random.

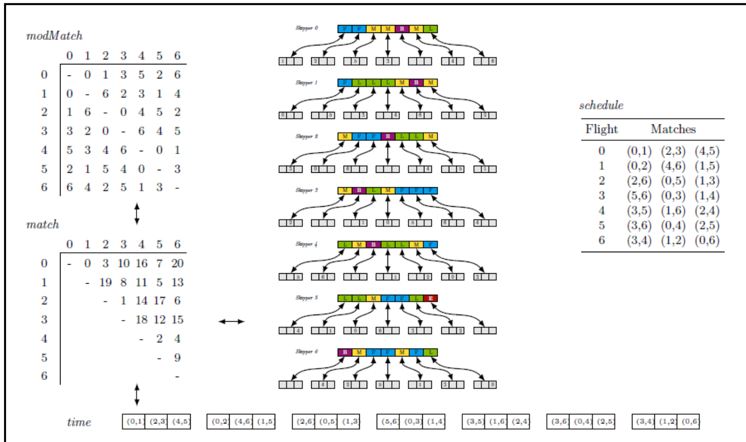
01	23	45
02	14	35
04	13	25
03	15	26
05	27	68
07	16	48
06	18	47
08	24	37
28	34	67
38	46	57
36	58	17
56	78	12

	0	1	2	3	4	5	6	7	8
0	—	0	3	9	6	12	18	15	21
1		—	35	7	4	10	16	32	19
2			—	1	22	8	11	13	24
3				—	25	5	30	23	27
4					—	2	28	20	17
5						—	33	29	31
6							—	26	14
7								—	34
8									—

DFA



		1st	mid	last	bye	end
		1	2	3	4	5
start	0	1	2	3	4	—
1st	1	1	2	3	4	5
mid	2	1	2	3	4	5
last	3	—	2	3	—	5
bye	4	—	2	3	4	—
end	5	—	—	—	—	5



A copy of these variables and constraints is created for each skipper σ:

$\forall \tau \in \{0 \dots t{-}1\} : timeSlot[\tau] \in \{-2 \dots t{-}1\} \setminus \{\sigma\}$ (V1)

$\forall i \in \{0 \dots f{-}1\} : state[i] \in \{\textsc{First}, \textsc{Mid}, \textsc{Last}, \textsc{Bye}, \textsc{End}\}$ (V2)

$\forall i \in \{0 \dots f{-}1\} :$ (C1)

$state[i] = \textsc{First} \Leftrightarrow timeSlot[m \cdot i] \geq 0$

$state[i] = \textsc{Mid} \Leftrightarrow \exists j \in \{m \cdot i + 1 \dots m \cdot (i+1) - 2\} : timeSlot[j] \geq 0$

$neSlot[m \cdot (i+1) - 1] \geq 0$

$\in \{m \cdot i \dots m \cdot (i+1) - 1\} : timeSlot[j] = -1$

$\in \{m \cdot i \dots m \cdot (i+1) - 1\} : timeSlot[j] = -2$

$timeSlot, \{0, \dots, n{-}1\} \setminus \{\sigma\})$ (C2)

(C3)

$] \in \{0, 1\}$ (V3)

(V4)

$] = 1 \Leftrightarrow state[i] = \textsc{Bye} \wedge state[i+1] \neq \textsc{Bye}$ (C4)

(C5)

bjective value from stage 1 is used as a hard constraint:

$lBoatChanges \leq \beta$ (C11)

y of these variables and constraints is created for each skipper σ:

$\{0 \dots m{-}1\} : \forall j \in \{0 \dots f{-}1\} :$

$osition[i,j] \in \{0,1\}$ (V9)

$osition[i,j] = 1 \Leftrightarrow timeSlot[m.j + i] \geq 0$ (C12)

$\{0 \dots m{-}1\} :$

$imbalance[i] \in \mathbb{N}$ (V10)

$$imbalance[i] = \left| \frac{n-1}{m} - \sum_{j=0}^{f-1} position[i,j] \right| \quad \text{(C13)}$$

$maxImbalance \in \mathbb{N}$ (V11)

$maxImbalance = \text{maximum}(imbalance)$ (C14)

ctives:

$\text{-}1 \dots n{-}1\} :$

(V5)

$].timeSlot[k] = j \wedge \sigma[j].timeSlot[k] = i$ (C6)

$\text{-}1 \dots n{-}1\} :$

$'{-}1\}$ (V6)

$Iatch[i,j]$

$i[i,j]/m$ (C7)

$ent(modMatch[i])$ (C8)

$2, n{-}1)\}$ (V7)

$i{+}1 \dots n{-}1\} : time[\tau] = (i,j) \Leftrightarrow match[i,j] = \tau$ (C9)

$totalChanges$ (V8)

$es)$ (C10)

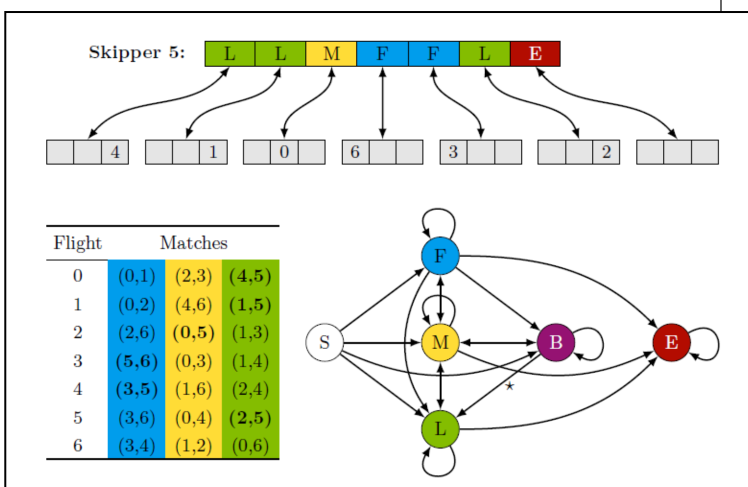
We minimize the maximum imbalance over all skippers:

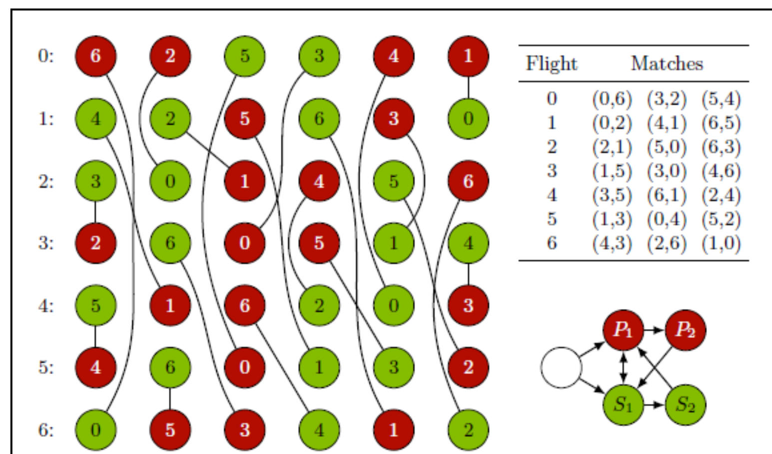
$\forall i \in \{0 \dots n{-}1\} : imbalance[i] \equiv \sigma[i].maxImbalance$ (V12)

$maxImbalance \in \mathbb{N}$ (V13)

$maxImbalance = \text{maximum}(imbalance)$ (C15)

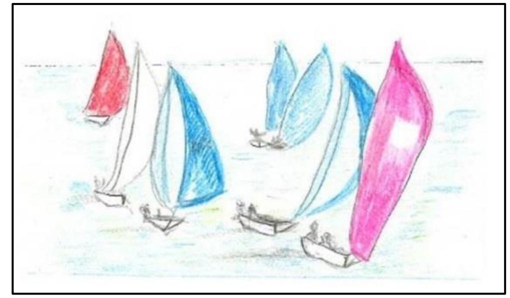
$\text{minimise}(maxImbalance)$ (C16)





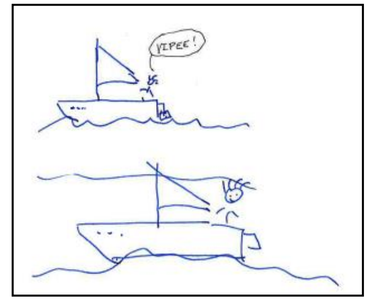
A MATCHING PROBLEM

A



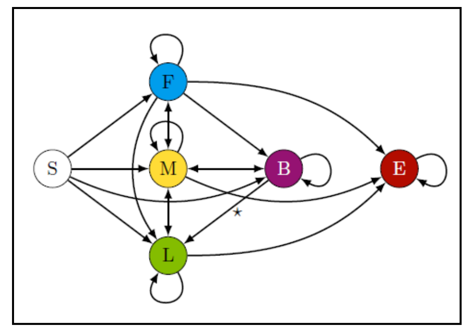
1

B



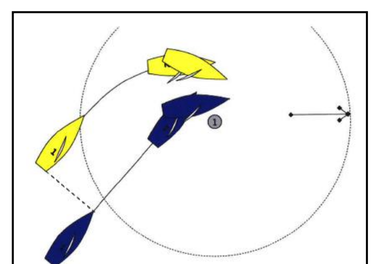
2

C



3

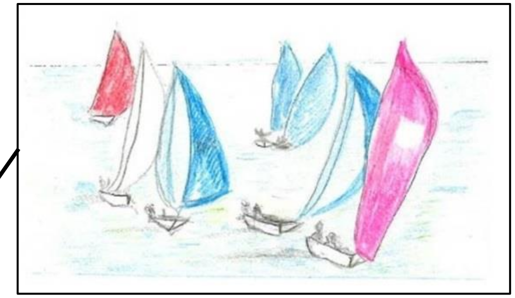
D



4

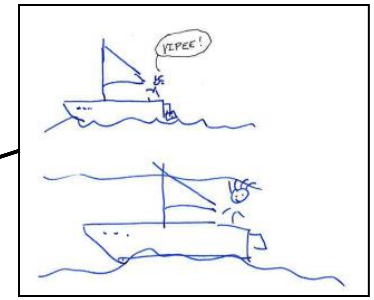
A MATCHING PROBLEM

A



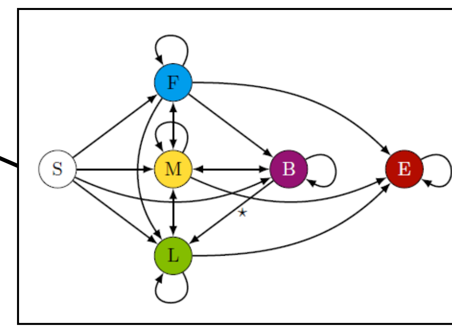
1

B



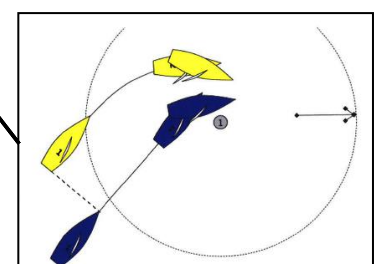
2

C



3

D



4

(A,4) (B,3) (C,2) (D,1)

Scene from an imaginary dinghy race