Drum Scheduling: A Special Case of the Traveling Salesman Problem

Michael Gabilondo
COP 5537
November 25, 2009

Outline

- Introduction, History, Known Results
- Drum Scheduling problem description
- Drum Scheduling as a graph problem
- Optimal Drum Scheduling Algorithm from (Fuller, 1972)
- Program Sreenshot
- References

- TSP: Find a Hamiltonian Cycle with smallest cost
  - NP-Hard

- 1857: Hamilton invented “icosian game”, a pegboard with a dodecahedron graph embedding – find a hamiltonian cycle
- 1970/80s: instances with up to 2392 cities
- 2005: 33,810 cities
- Approximation algorithms – millions of cities

- 1964 – Gilmore and Gomory, sequencing a one-state variable machine: A solvable case of TSP
- 1972 – Fuller, applied 1964 paper to drum scheduling – optimal O(nlogn) algorithm

From “An Optimal Drum Scheduling Algorithm”, Fuller (1972)

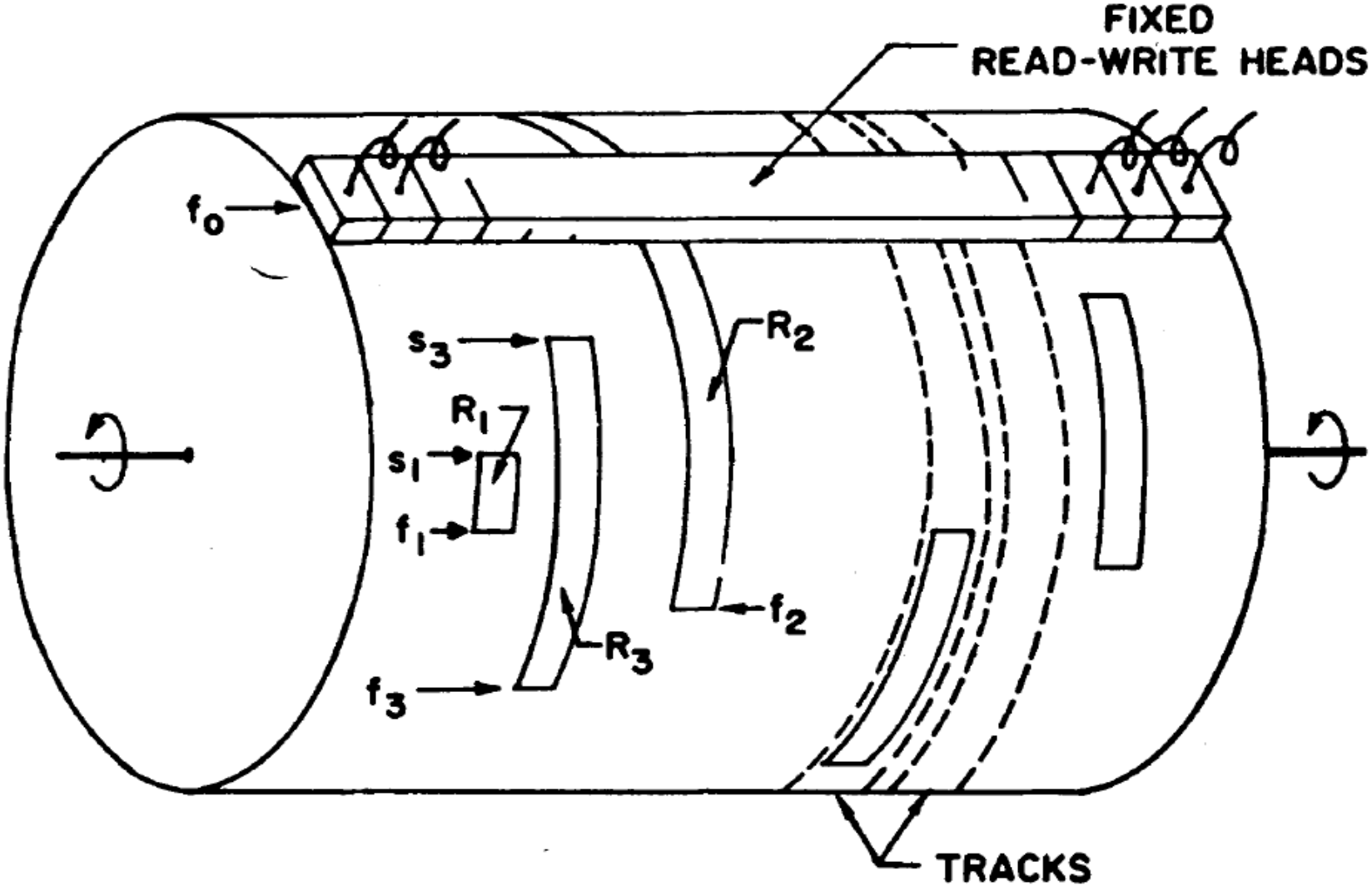


Fig. 1. Drum storage unit.

From “An Optimal Drum Scheduling Algorithm”, Fuller (1972)

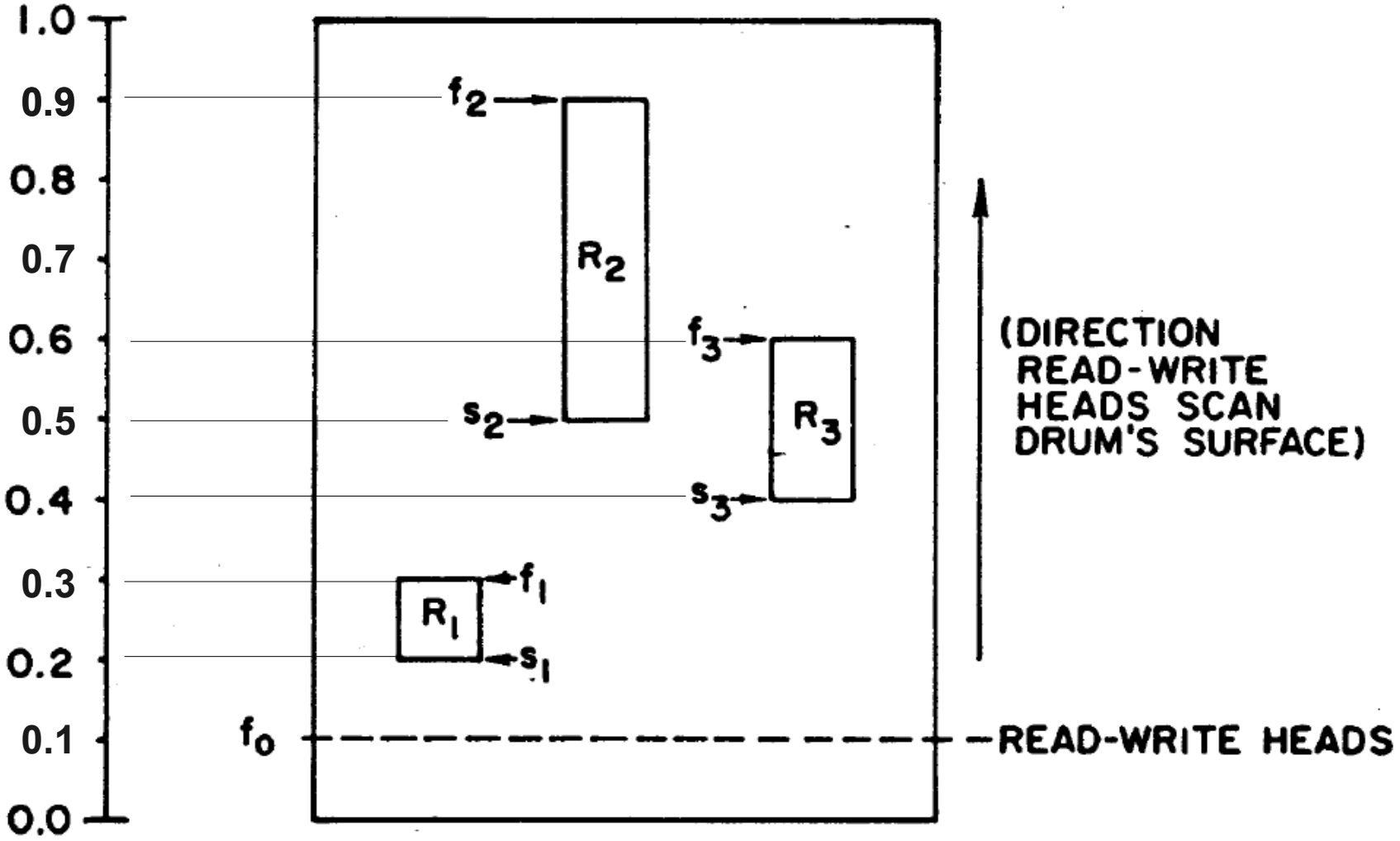
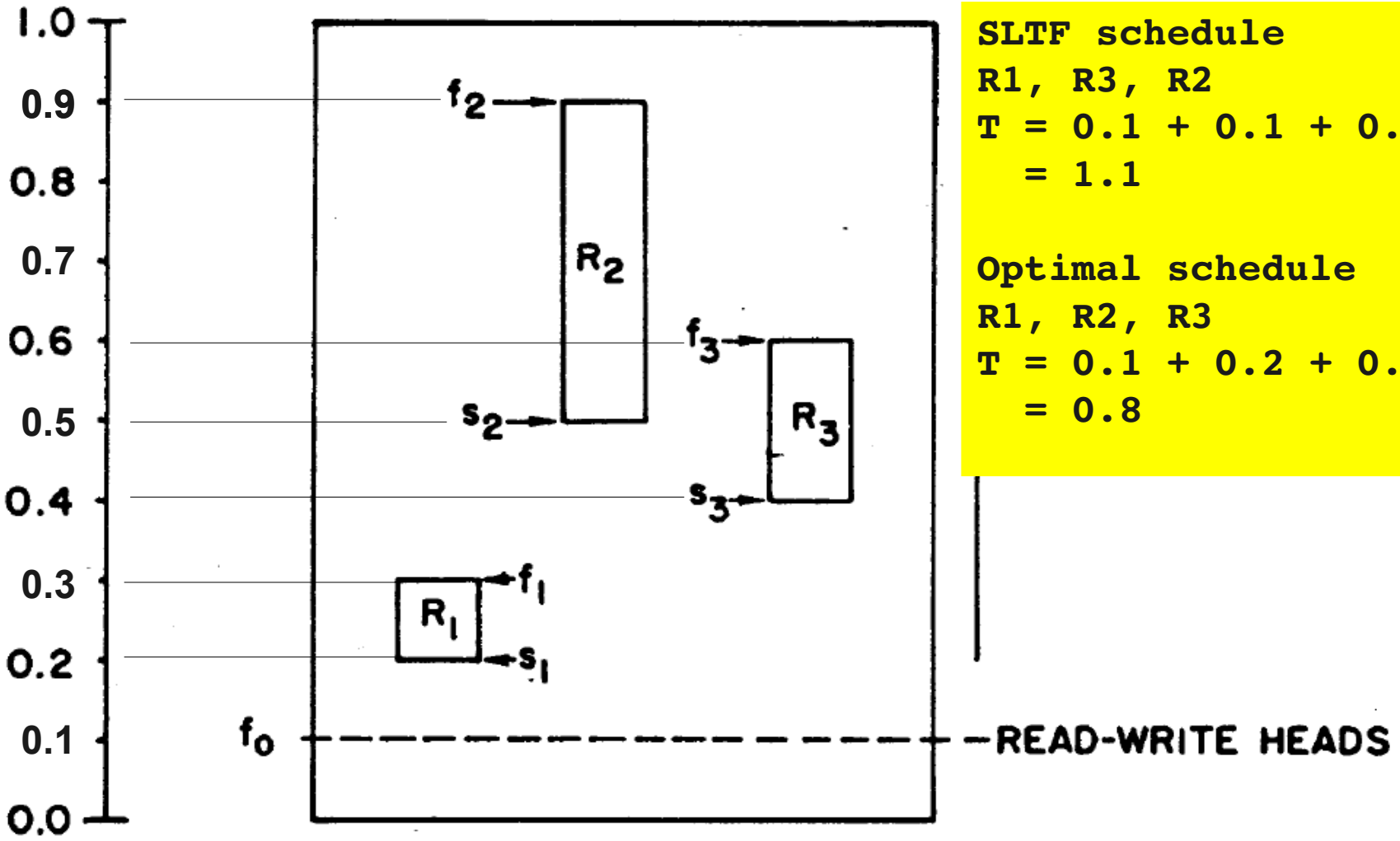


Fig. 2. Location on the drum's surface of the three records and the initial position of the READ–WRITE heads.

From “An Optimal Drum Scheduling Algorithm”, Fuller (1972)

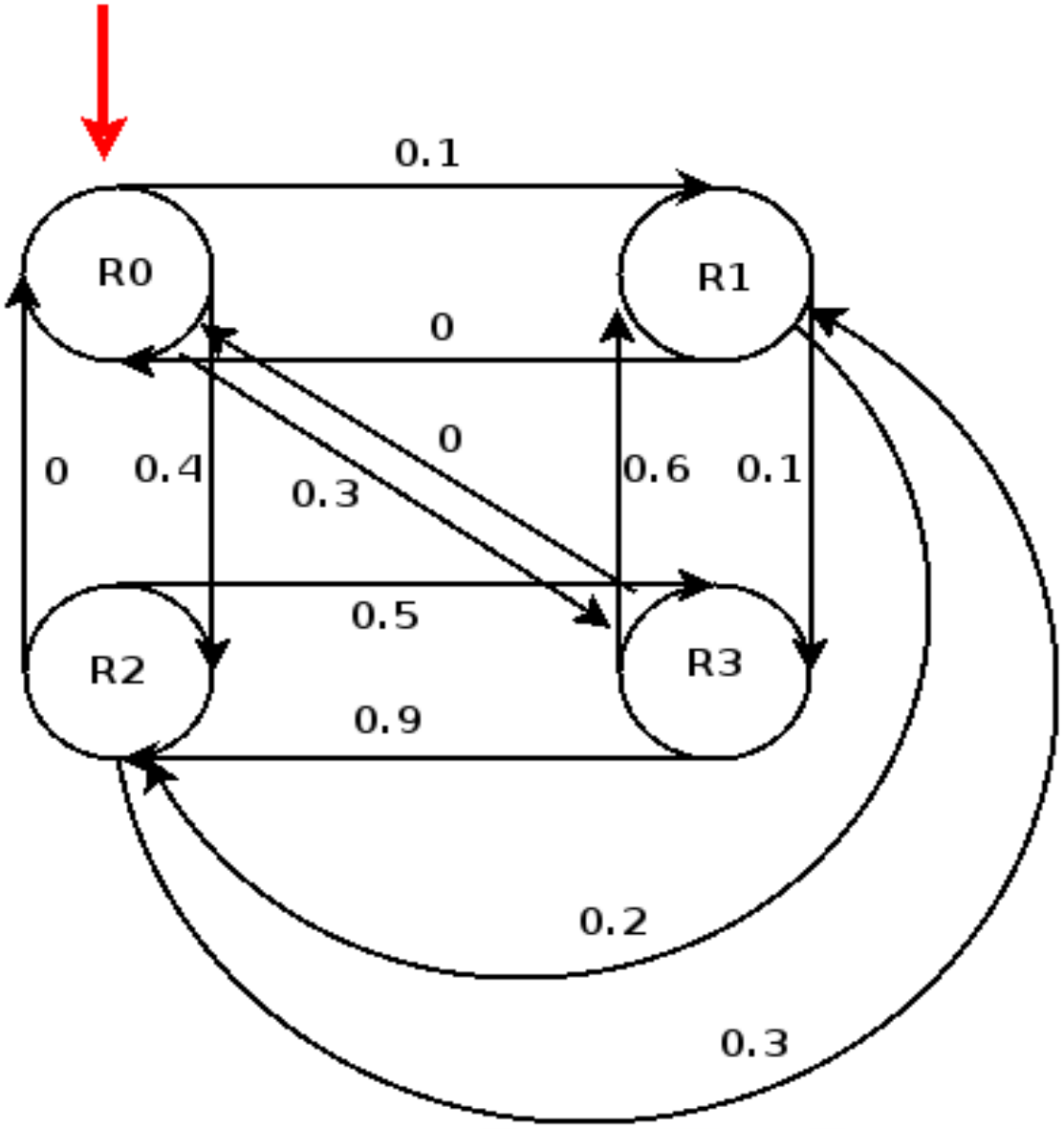



```
SLTF schedule
R1, R3, R2
T = 0.1 + 0.1 + 0.9
  = 1.1

Optimal schedule
R1, R2, R3
T = 0.1 + 0.2 + 0.5
  = 0.8
```

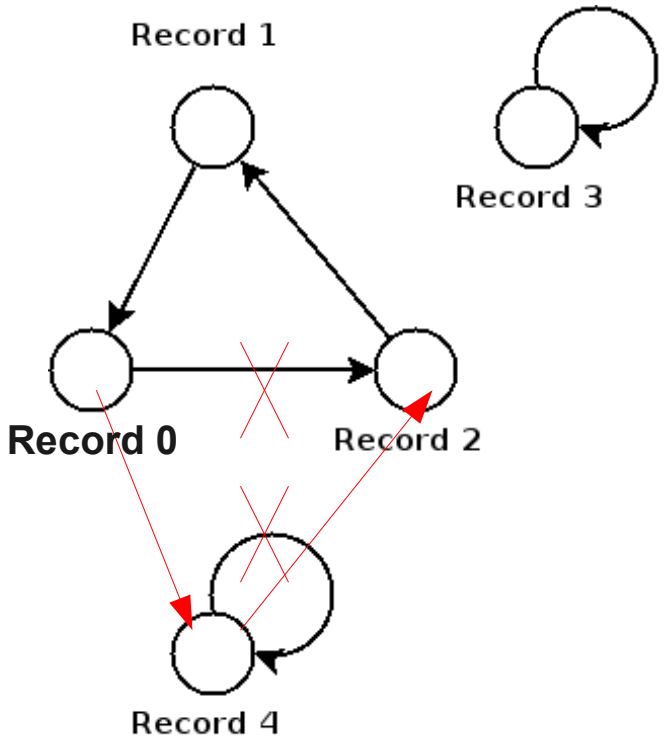
Fig. 2. Location on the drum's surface of the three records and the initial position of the READ–WRITE heads.

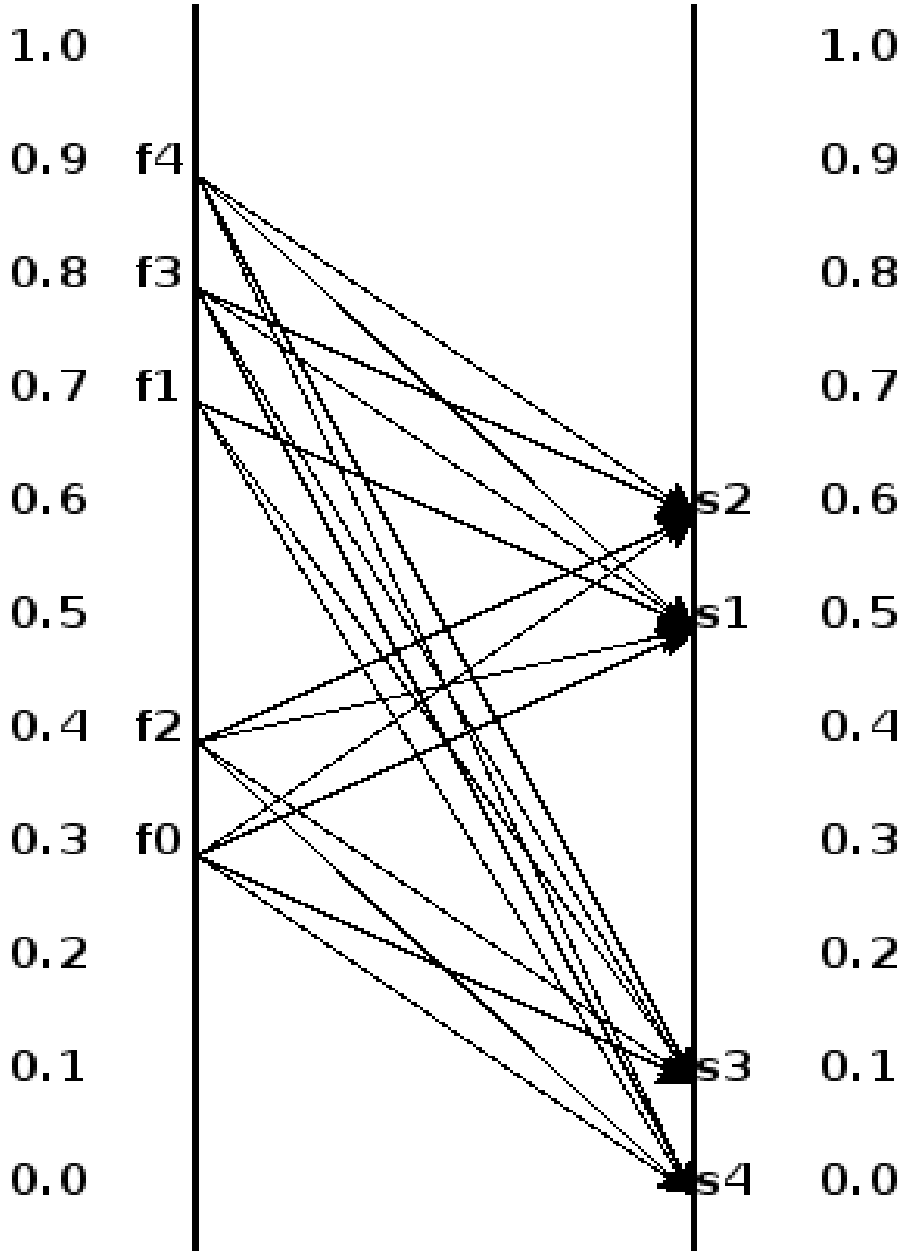
Pseudorecord
Ends at initial position of read-write heads
Starts at the end of any record



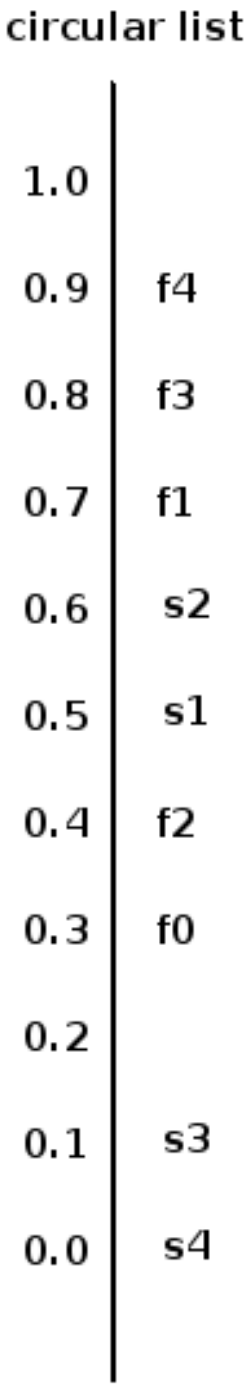
General Approach

- Construct a **minimal cost permutation**
  - Ignoring the “single cycle” constraint
  - (0, 2, 1) (3) (4)
- Join cycles using “edge interchanges”
  - Increase cost by minimum amount
  - (0, 2, 1, 4, 3)





matching problem



Minimal Matching Procedure
for Ordered Bipartite Graphs:

Greedy: choose the edge with the smallest cost

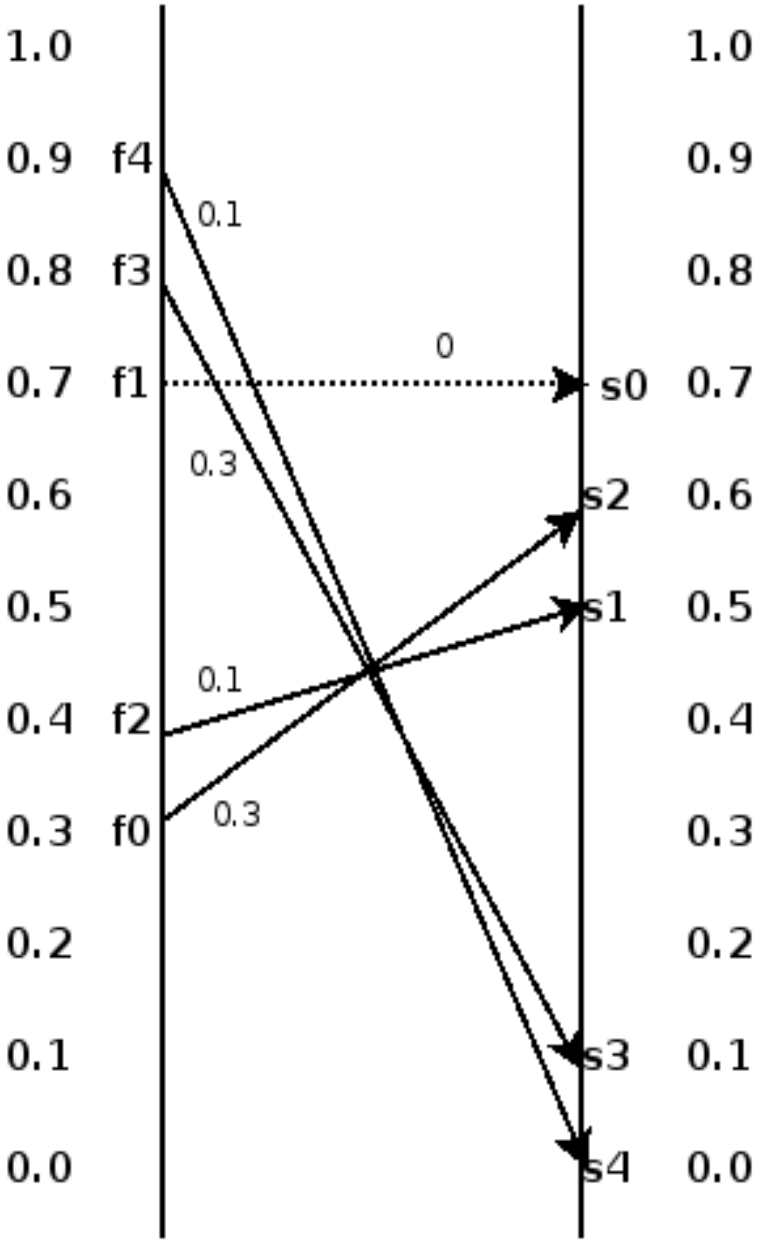
f2 ------ s1

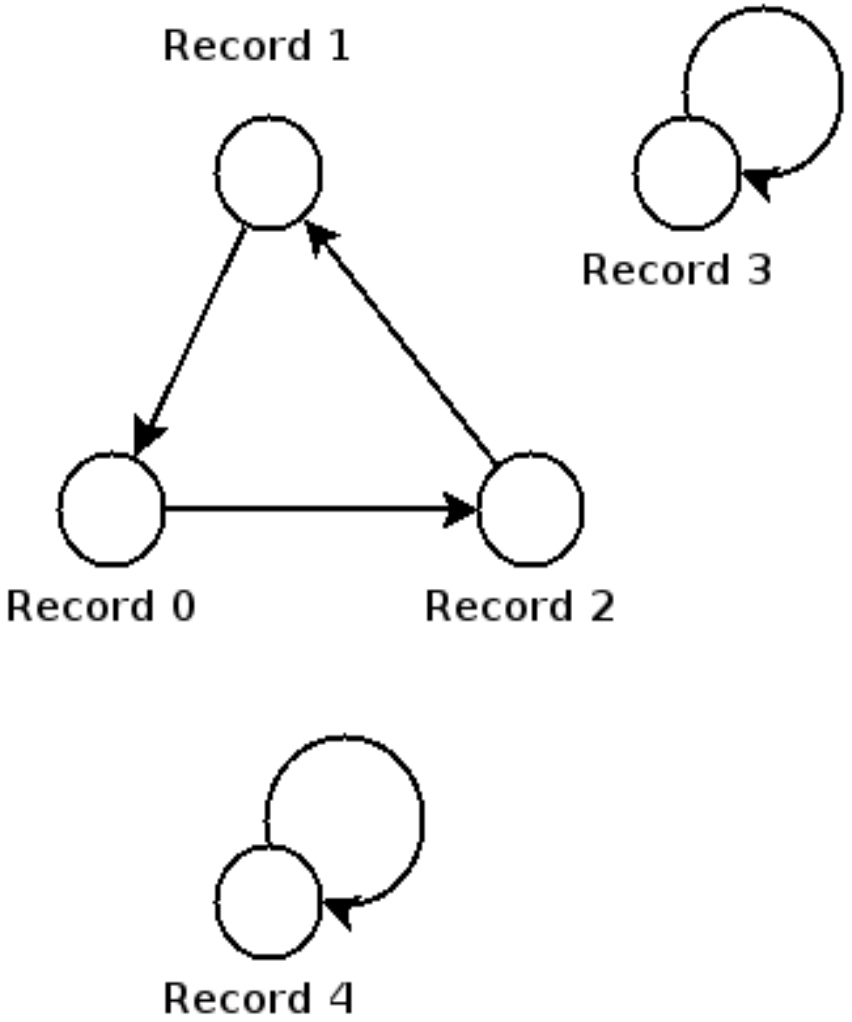
f0 ------ s2

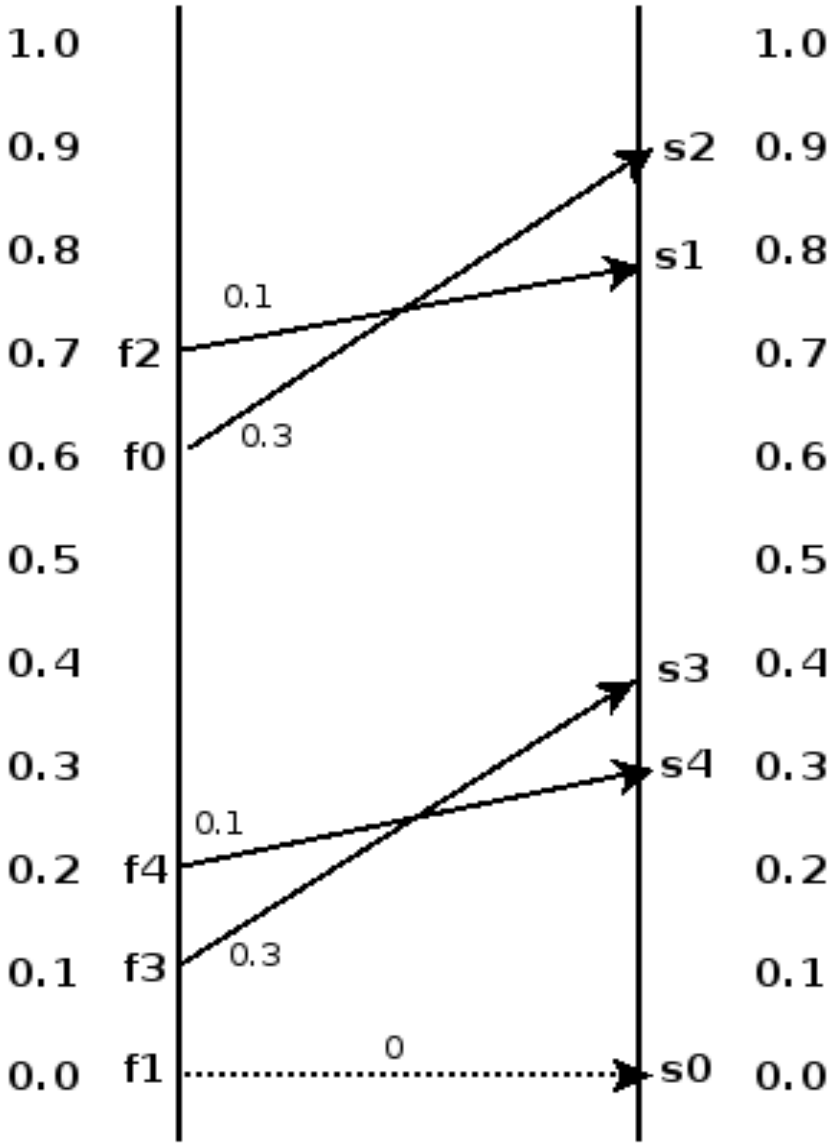
f4 ------ s4

f3 ------ s3

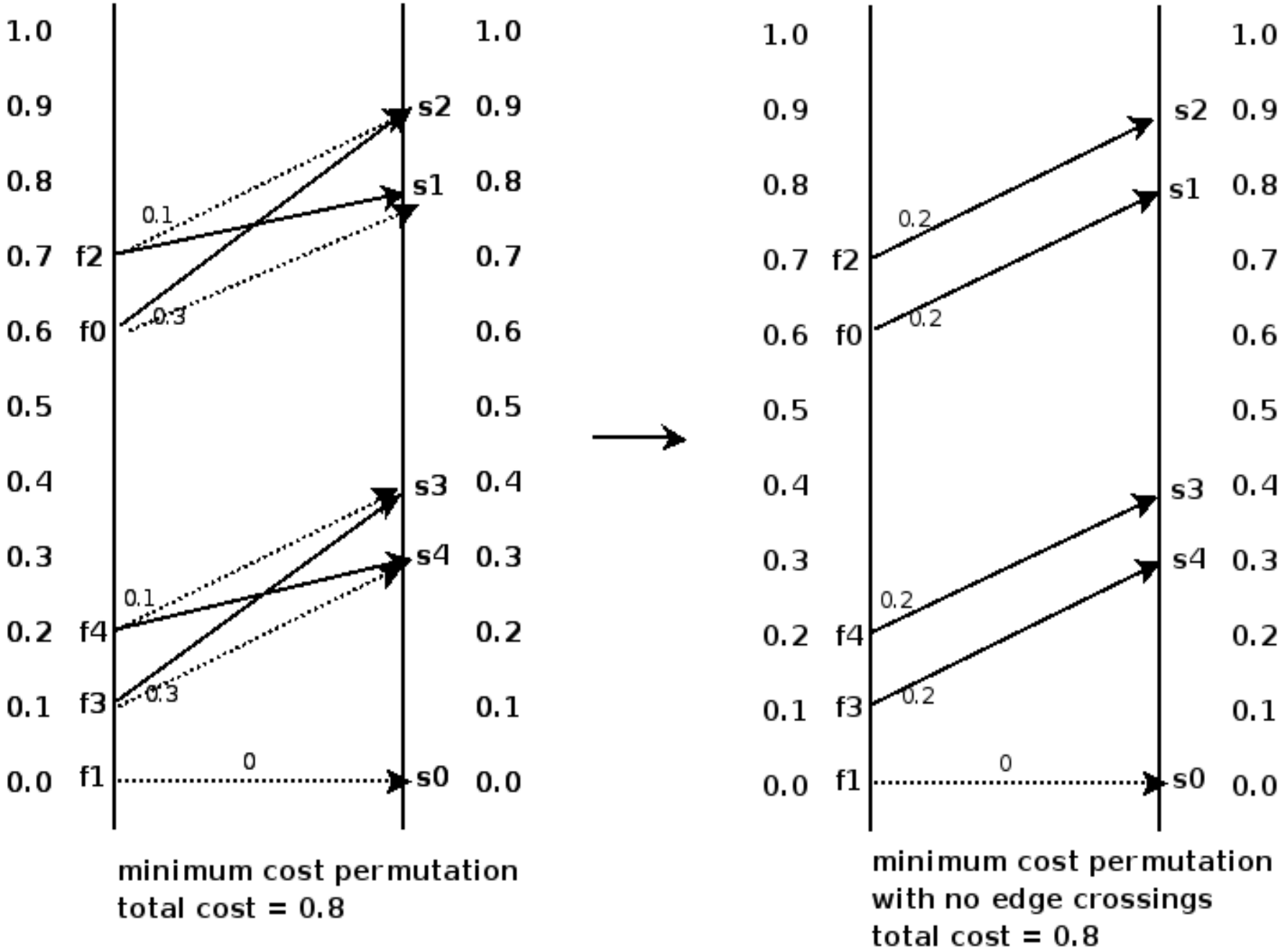
f1 = f_delta ------ s0

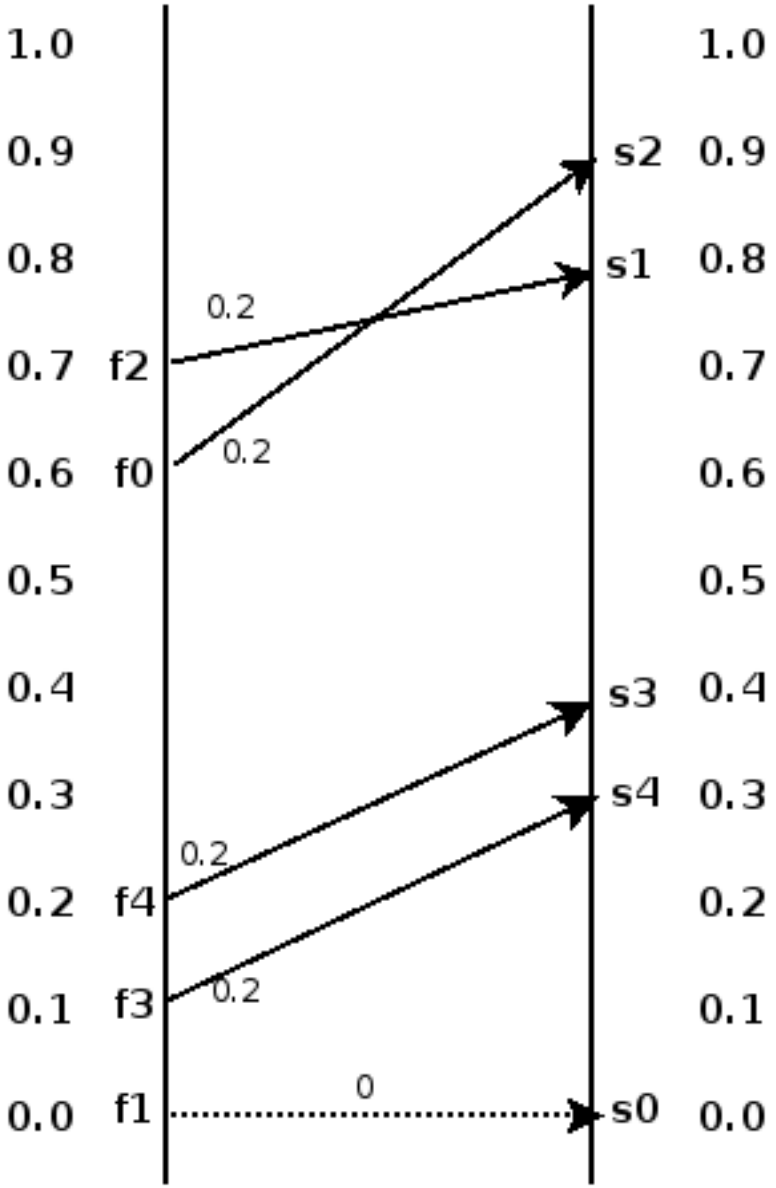

1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
f4
f3
f1
f2
f0
0.1
0
0.3
0.1
0.3
s0
s2
s1
s3
s4
minimum cost permutation
total cost = 0.8

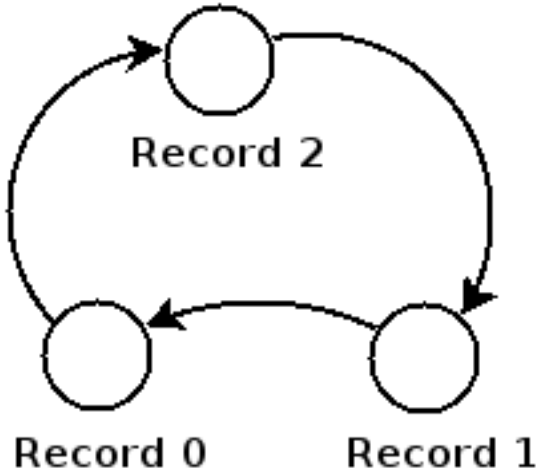

Record 1
Record 3
Record 0
Record 2
Record 4

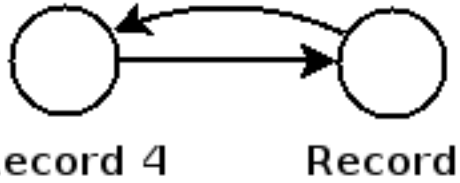


minimum cost permutation
with point of reference on the drum shifted to f1 = f_delta
total cost = 0.8

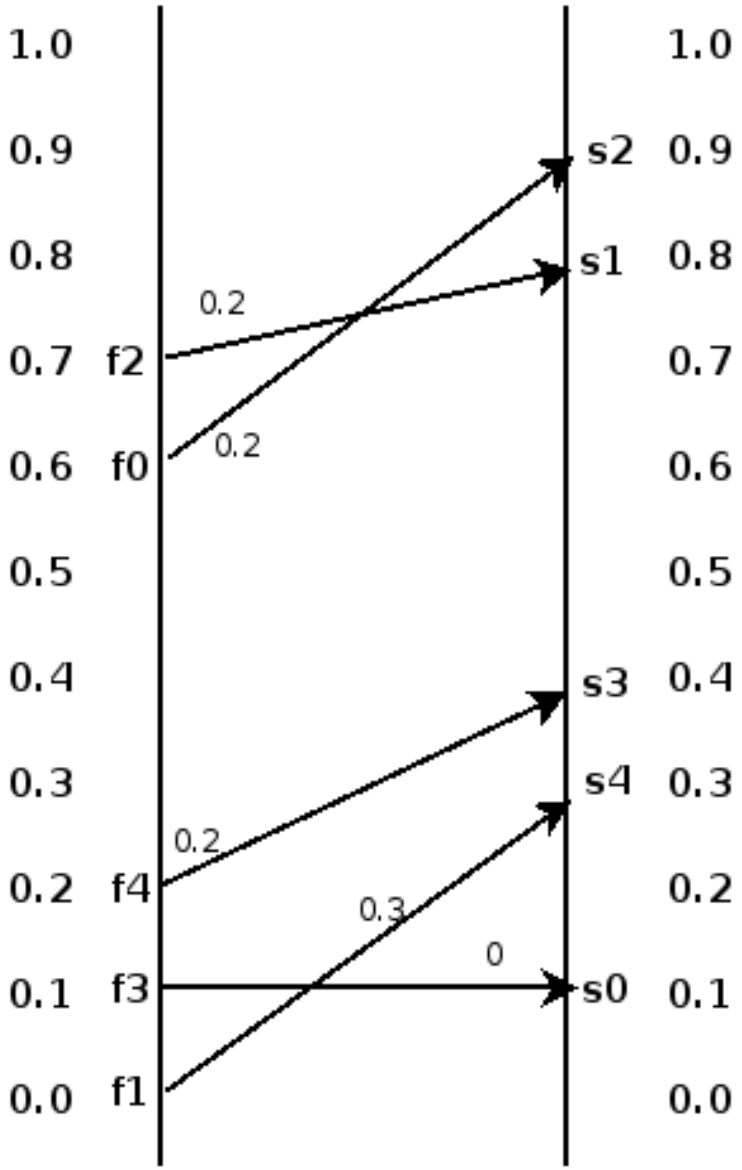

1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
f2
f0
f4
f3
f1
s2
s1
s3
s4
s0
0.1
0.3
0.1
0.3
0
minimum cost permutation
total cost = 0.8
0.2
0.2
0.2
0.2
0
minimum cost permutation
with no edge crossings
total cost = 0.8

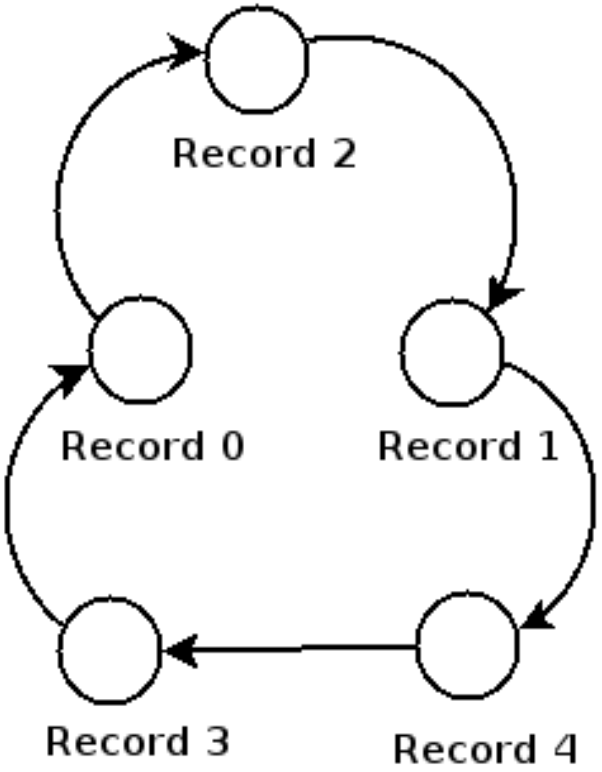






minimum cost, minimum cycle permutation
total cost = 0.8





minimum cost permutation with the restriction that it contains only one cycle
total cost = 0.9

```
# number of records
4

# starting position of read/write heads
0.3

# record 1 start and ending position
0.5 0.7

# record 2
0.6 0.4

# record 3
0.1 0.8

# record 4
0.0 0.9
```

Console

```
<terminated> Schedule [Java Application] /usr/lib/jvm/java-6-su
The minimum latency schedule is
2  1  4  3
with cost 0.90
```

References

- Sequencing a one-state variable machine: A solvable case of TSP (Gilmore and Gomory, 1964)

- An Optimal Drum Scheduling Algorithm (Fuller, 1972)

- Traveling Salesman Problem (Hoffman et al)|

- On the Near-Optimality of the STLF Drum Scheduling Discipline (Stone and Fuller, 1973)

- Wikipedia (traveling salesman problem)