ASH - A Scheduler For HOAs

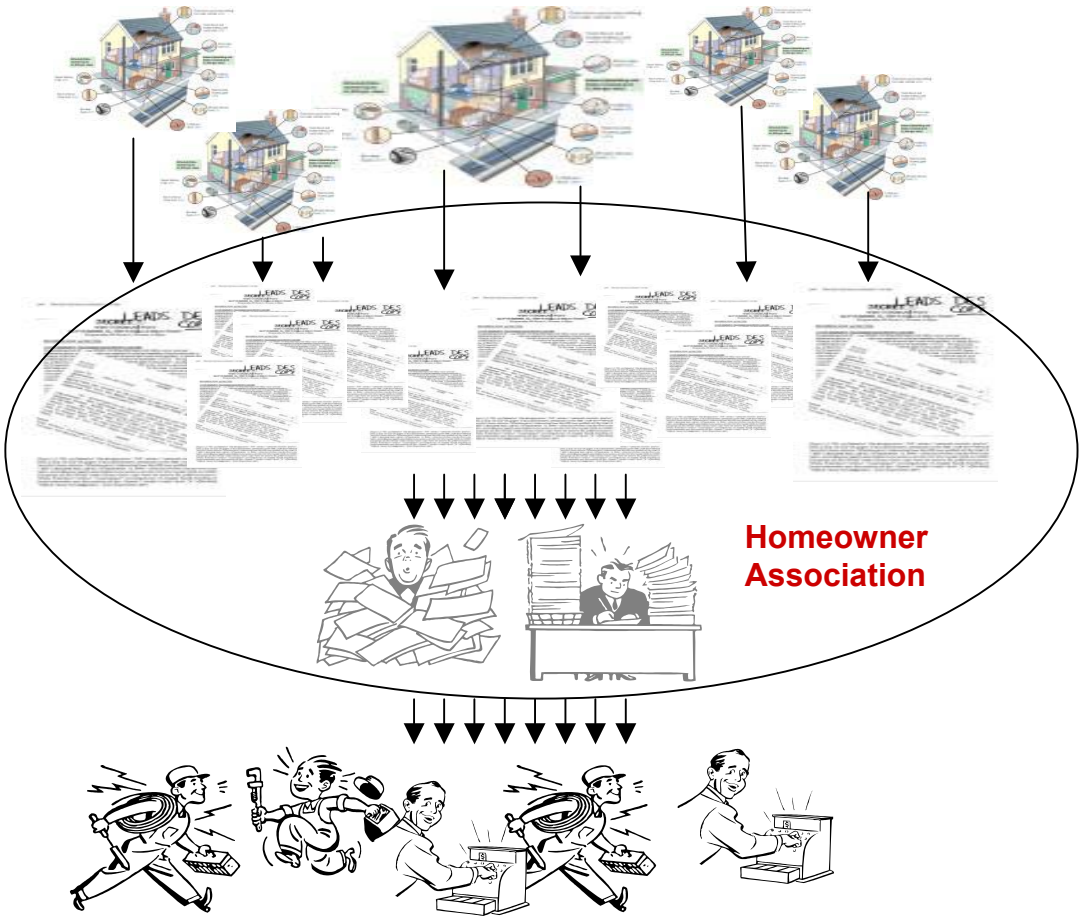
by Qian Li (Sabrina)

Advisor: Dr. Chris Pollett
Committee: Dr. David Taylor
Dr. Sami Khuri

Agenda

- Introduction
- Background and Related Work
- Design and Implementation
  - Design Objectives
  - Super-ASH algorithm
- Experiments and Analysis
- Conclusion and Future Work

Introduction

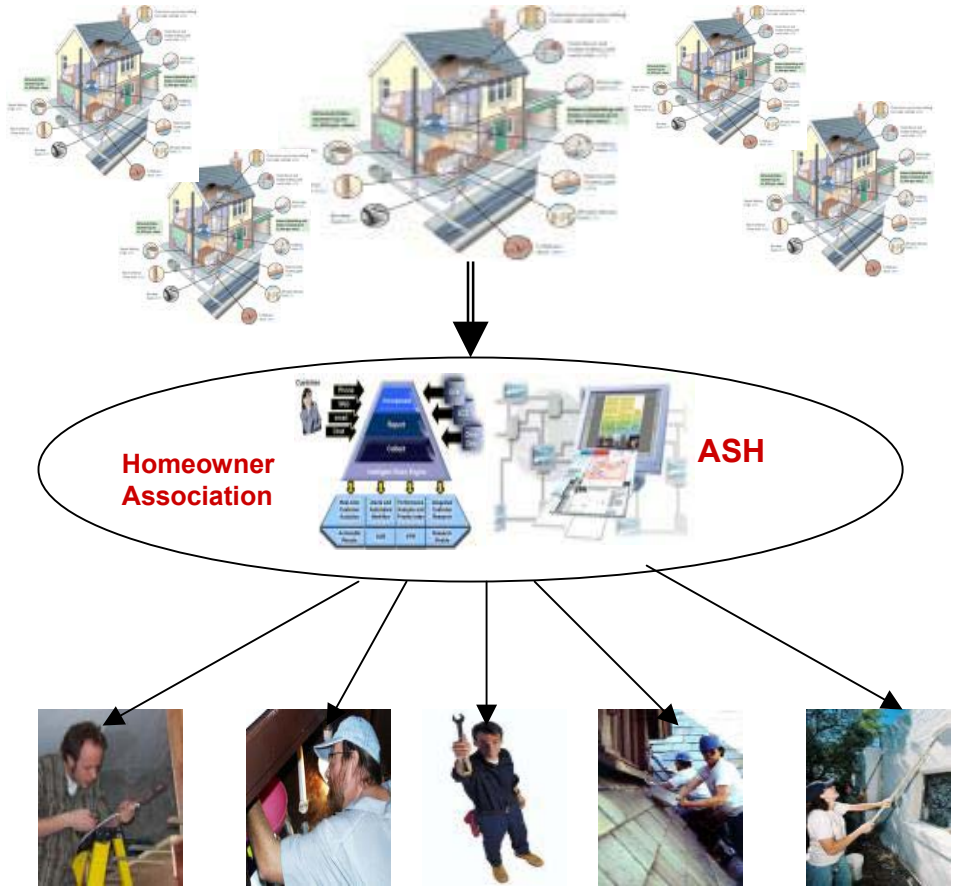


- Problems
  - Request repair jobs with paper forms;
  - Schedule the requested jobs manually;
  - Hard to archive and analyze data.
- Solution

Introduction

Solution:

- Accept online applications;
- Schedule the requested jobs with particular scheduler- ASH.



Introduction

- HOA scheduling problem

	Standard Notation	**HOA Scheduling**
α	Machine environment	One administrator & several contractors
β	Side constraints	Limited resources
γ	Optimized criterion	Customers' ☺

Background & Related Work

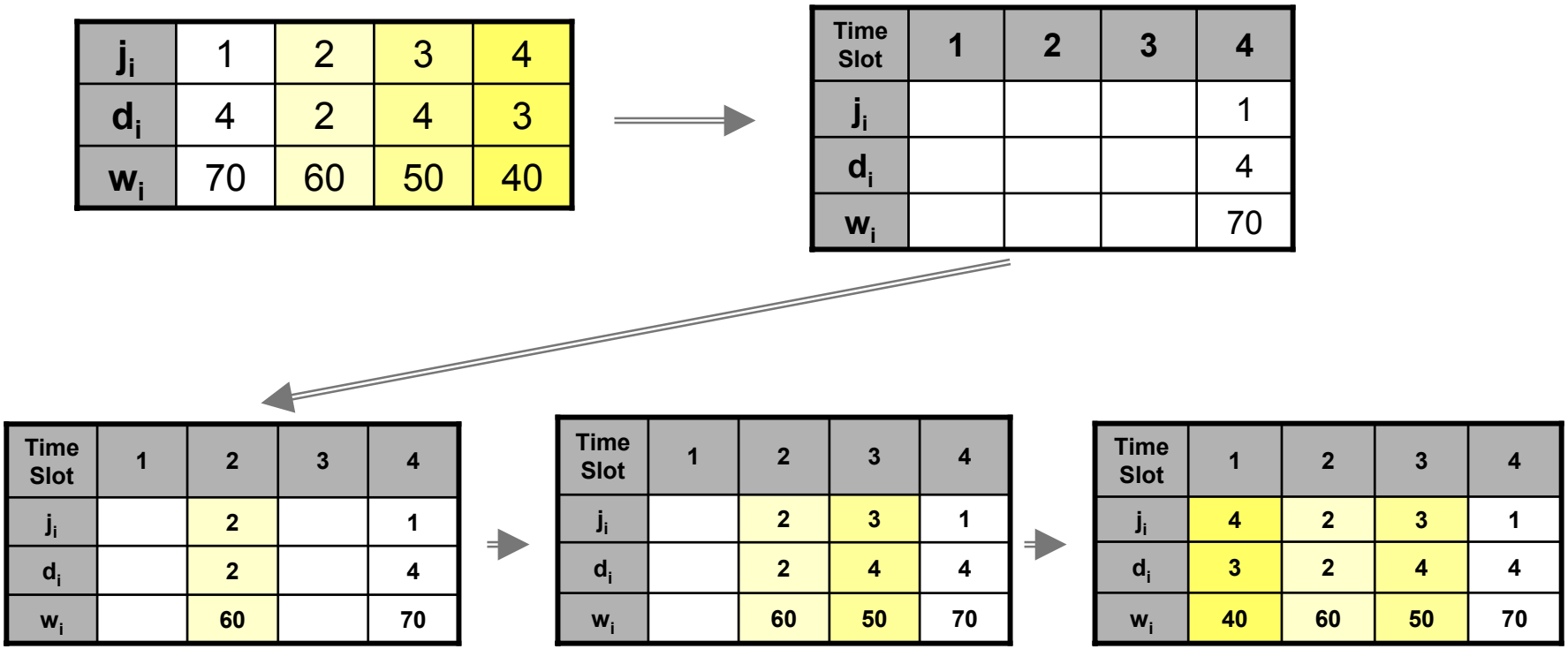
- Original Scheduling Algorithms
  - Greedy Unit Task Scheduling Algorithm
  - First In First Out Algorithm (FIFO)
  - Shortest Processing Time First (SPTF)
  - Semi-clairvoyant R Algorithm (Sc-R)
- Comparisons and Inspirations

Job characteristics

- HOA scheduling problem
  - A set of jobs $J = \{ j_1, j_2, j_3 \ldots j_n \}$;
  - Job property descriptor:
    - p_i represents job j_i process time, d_i represents job j_i deadline and w_i represents job j_i weight;
    - r_i represents job j_i release time;
    - c_i represents job j_i complete time, f_i represents job j_i finish time, *where* $f_i = c_i - r_i$ ..

Original Scheduling Algorithms (1- 4)

- **Greedy Unit Task Scheduling algorithm**



j_i	1	2	3	4
d_i	4	2	4	3
w_i	70	60	50	40

Time Slot	1	2	3	4
j_i				1
d_i				4
w_i				70

Time Slot	1	2	3	4
j_i		2		1
d_i		2		4
w_i		60		70

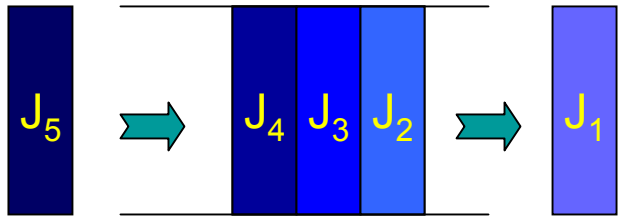
Time Slot	1	2	3	4
j_i		2	3	1
d_i		2	4	4
w_i		60	50	70

Time Slot	1	2	3	4
j_i	4	2	3	1
d_i	3	2	4	4
w_i	40	60	50	70

Original Scheduling Algorithms (1- 4)

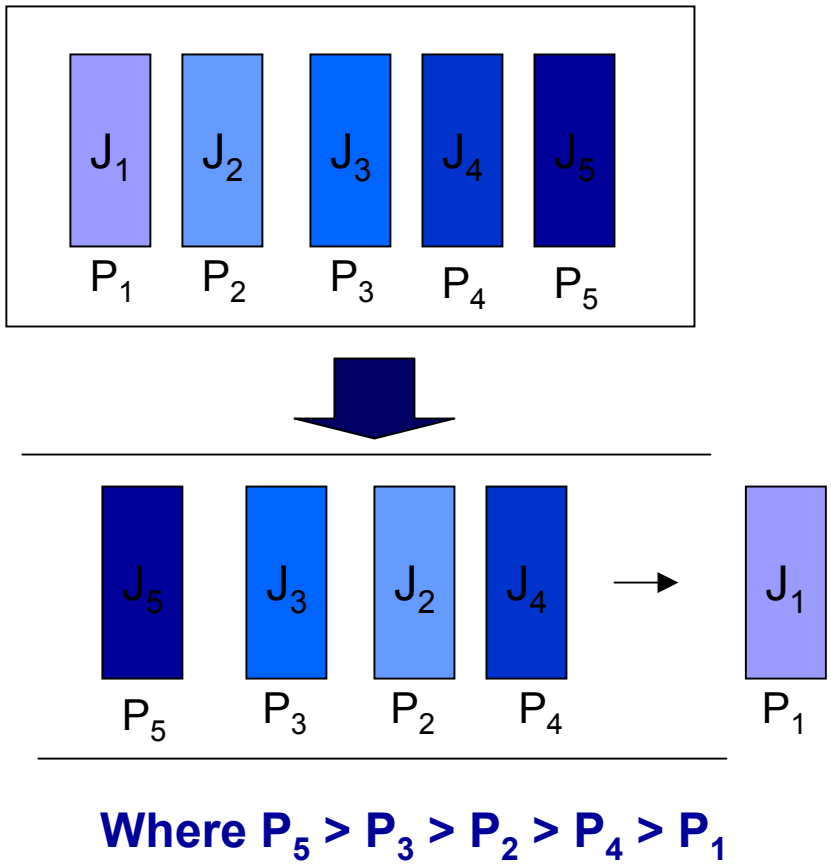
- Greedy Unit Task Scheduling Algorithm
  - Method: make a locally optimal choice that could lead to the final global optimal solution
  - Advantage: conceptual simplicity
  - Disadvantage: rarely find the globally optimal solution

Original Scheduling Algorithms (2- 4)



- First In First Out (FIFO)
  - Method: assigns priority to the jobs in the order in which they request
  - Advantage: simple implementation and fairness
  - Disadvantage: cannot handle different job priorities

Original Scheduling Algorithms (3- 4)



- Shortest Processing Time First (SPTF)
  - Method: choose the job that requires minimum processing time to run first
  - Advantage: obtain the minimum average processing time
  - Disadvantage: starvation - ignore those jobs with long service time requests

Original Scheduling Algorithms (4- 4)

- Semi-clairvoyant R algorithm (Sc-R)

J_i	R_i	P_i	C_i	$F_i = C_i - R_i + 1$
J_1	1	8		
J_2	4	1		
J_3	4	3		

		1	2	3	4	5	6	7	8	9	10	11	12
I	[1,2)												
II													
III	[4,8)		J_1	J_1									
IV	[8,16)	J_1											

Original Scheduling Algorithms (4- 4)

- Semi-clairvoyant R algorithm (Sc-R)

J_i	R_i	P_i	C_i	$F_i = C_i - R_i + 1$
J_1	1	8		
J_2	4	1		
J_3	4	3		

		1	2	3	4	5	6	7	8	9	10	11	12
I	[1,2)				J_2								
II	[2,4)				J_3								
III	[4,8)		J_1	J_1	J_1								
IV	[8,16)	J_1											

Original Scheduling Algorithms (4- 4)

- Semi-clairvoyant R algorithm (Sc-R)

J_i	R_i	P_i	C_i	$F_i = C_i - R_i + 1$
J_1	1	8		
J_2	4	1		
J_3	4	3		

		1	2	3	4	5	6	7	8	9	10	11	12
I	[1,2)				J_2					J_1			
II	[2,4)				J_3	J_3	J_3	J_1 J_3	J_1 J_3	J_3			
III	[4,8)		J_1	J_1	J_1	J_1	J_1						
IV	[8,16)	J_1											

Original Scheduling Algorithms (4- 4)

- Semi-clairvoyant R algorithm (Sc-R)

J_i	R_i	P_i	C_i	$F_i = C_i - R_i + 1$
J_1	1	8		
J_2	4	1		
J_3	4	3		

		1	2	3	4	5	6	7	8	9	10	11	12
I	[1,2)				J_2					J_1			J_3
II	[2,4)				J_3	J_3	J_3	J_1 J_3	J_1 J_3	J_3	J_3	J_3	
III	[4,8)		J_1	J_1	J_1	J_1	J_1						
IV	[8,16)	J_1											

Original Scheduling Algorithms (4- 4)

- Semi-clairvoyant R algorithm (Sc-R)

J_i	R_i	P_i	C_i	$F_i = C_i - R_i + 1$
J_1	1	8	*9*	*9*
J_2	4	1	*4*	*1*
J_3	4	3	*12*	*9*

		1	2	3	4	5	6	7	8	9	10	11	12
I	[1,2)				J_2					J_1			J_3
II	[2,4)				J_3	J_3	J_3	J_1 J_3	J_1 J_3	J_3	J_3	J_3	
III	[4,8)		J_1	J_1	J_1	J_1	J_1						
IV	[8,16)	J_1											

Original Scheduling Algorithms (4- 4)

- Semi-clairvoyant R algorithm (Sc-R)
  - Method: Applied some knowledge under uncertainty and multiple queues with different priorities
  - Advantage:O(1)-competitive with respect to average flow time *or* average stretch
  - Disadvantage: not fully developed and implemented method

Comparisons and Inspirations

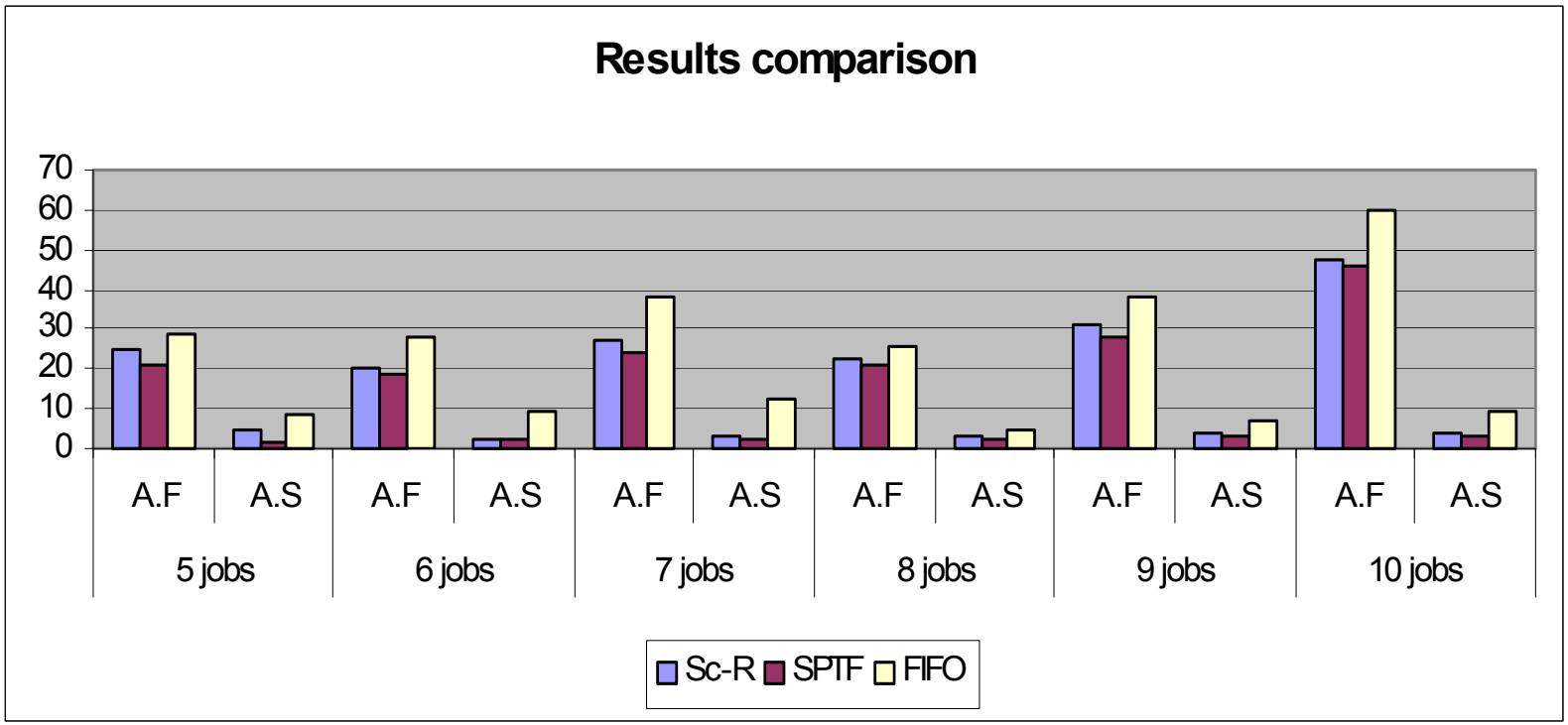
- Comparison measures:
  - Average Flow Time – total processing time divided by the number of jobs:

$$AFT = 1/n \sum_{i=1}^{n} (c_i - r_i + 1)$$

  - Average Stretch (AS) – average proportion between the real process time and minimum process time:

$$AS = 1/n \sum_{i=1}^{n} \{(c_i - r_i + 1) / p_i\}$$

Comparisons and Inspirations



A.F represents Average Flow Time, A.S represents Average Stretch,

Sc-R stands for Semi-clairvoyance R algorithm, SPTF stands for Shortest Processing Time First algorithm, and FIFO stands for First In First Out Algorithm.

Comparisons and Inspirations

- SPTF performs best with respect to AFT and AS, Sc-R stands in the middle, FIFO lags behind both SPTF and Sc-R
- SPTF will cause starvation, and the processing queue is lack of flexibility compared with Sc-R
- Balance the pros and cons in the original scheduling algorithms and the constraints of HOA, apply multi-level queue with multi-level priorities in ASH

The Super-ASH Algorithm (1 of 2)

- Supplemental considerations
  - Budget and Cost
  - One HOA with multiple Contractors – parallel processing
  - Job Type Table – knowledge on the job property, such as the job approximate cost and process time
  - Emergency jobs and normal jobs

The Super-ASH Algorithm (2 of 2)

- Super-ASH flexibility
  - Update job type information

    e.g: add, modify and delete
  - Set budget interval and amount

    e.g: admin. setBudget(30, 10000);
  - Choose scheduling method

    e.g: setQueueModel(int model);

Design and Implementation

- Design Objectives
  - Minimize Actual Average Process Time

    -- the average time that users experience to have their jobs done (QoS) (more details in the next slide)

  - Minimize Actual Average proportion between real process time and minimum process time

    -- the average ratio of a job's actual process time to its expected process time (efficiency) (more details in the next slide)

Design Objectives (1 of 2)

- Minimize Actual Average Flow Time

AAFT = 1/n $\sum_{i=1}^{n} (c_i - r_i' + 1)$

c_i – depends on scheduling results,

r_i' – depends on receive time and budget

- Minimize Actual Average Stretch

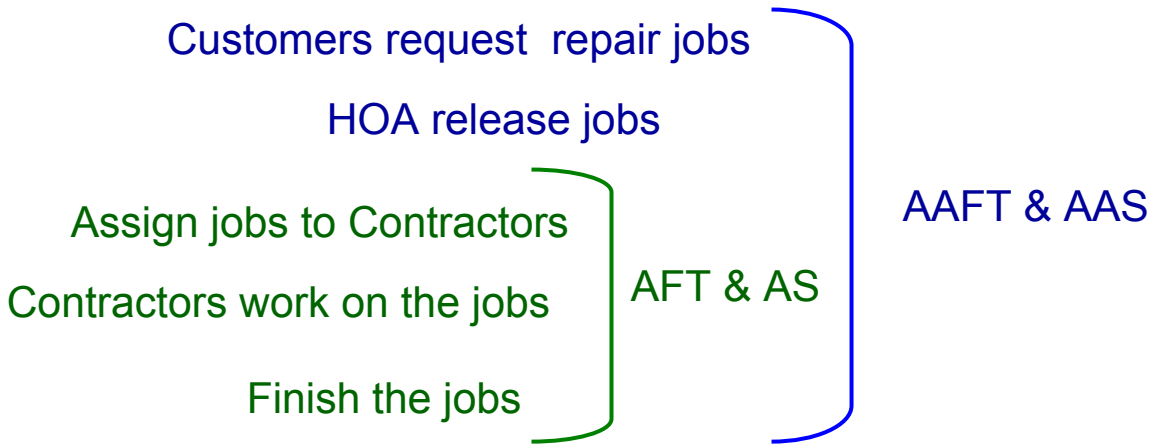
AAS = 1/n $\sum_{i=1}^{n} \{(c_i - r_i' + 1) / p_i\}$

c_i – depends on scheduling results,

r_i' – depends on receive time and budget

Design Objectives (2 of 2)

- Actual Average Flow Time and Actual Average Stretch stands on the Homeowner Association and customers' point of view;
- Average Flow Time and Average Stretch stands on the contractors and schedulers' point of view.

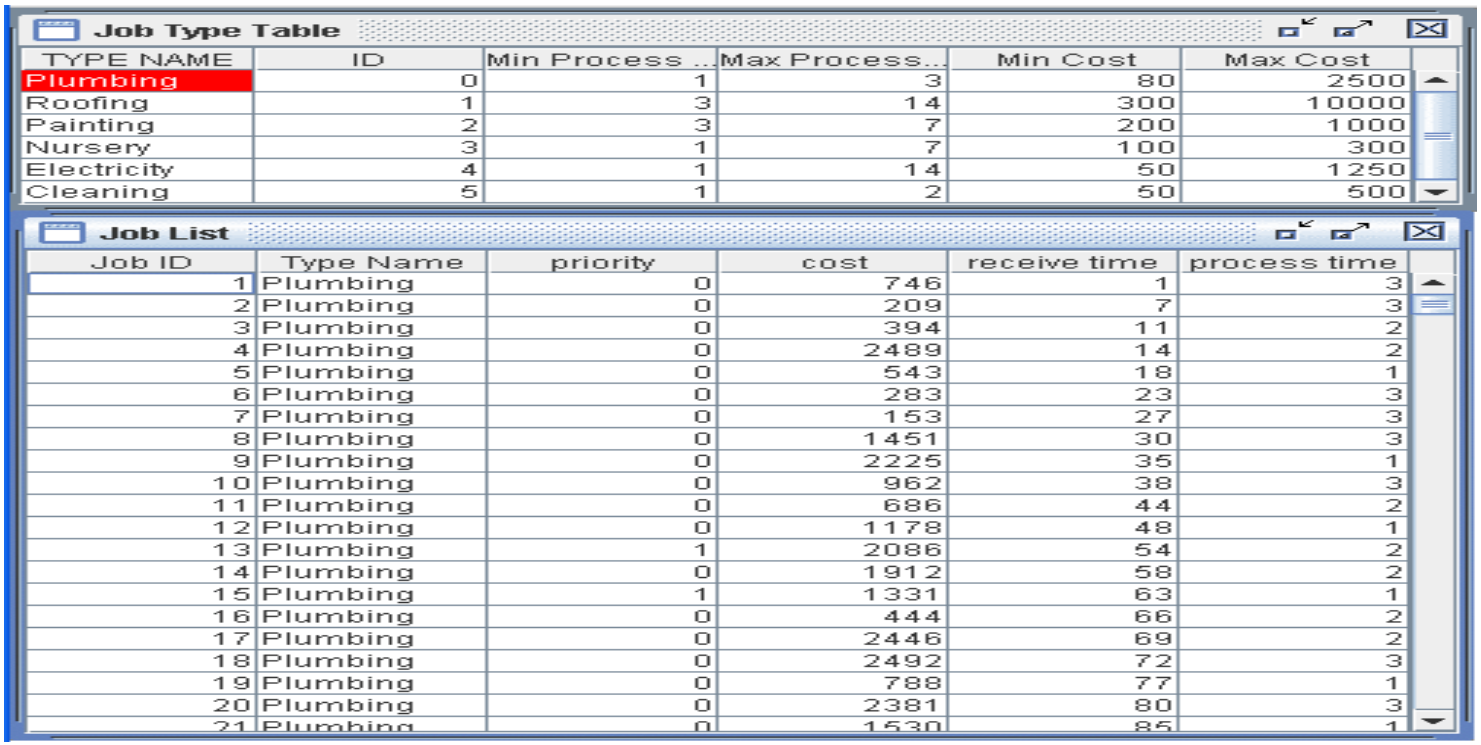


The Super-ASH Algorithm - HOA handling

- Step One:
  - Update the Job Type Table if necessary
    - Add, modify and delete
  - Configure the variables if needed
    - Scheduler (ASH-Power2, ASH-Even or ASH-Tri, FIFO or SPTF)
    - Budget (Interval, amounts)

The Super-ASH Algorithm - HOA handling

- Maintain the Job Type Table and create Job List



Job Type Table

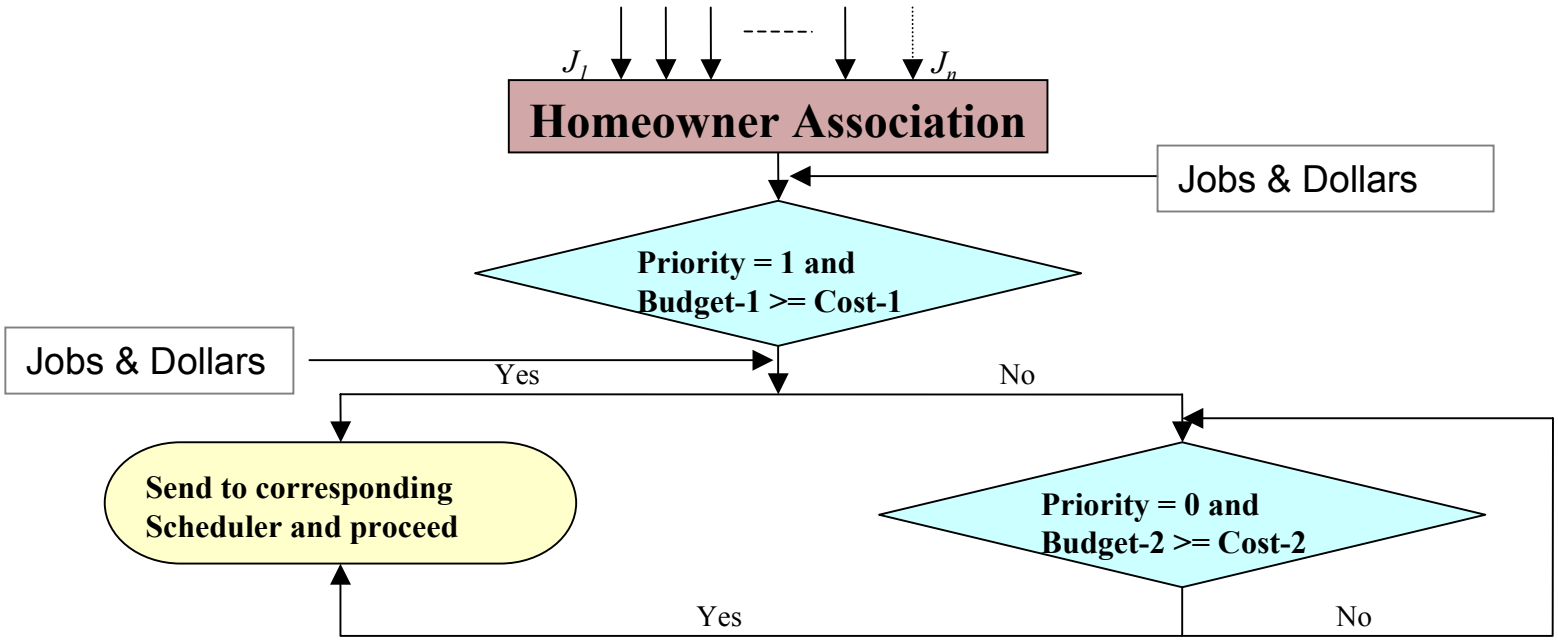
TYPE NAME	ID	Min Process ..	Max Process..	Min Cost	Max Cost
Plumbing	0	1	3	80	2500
Roofing	1	3	14	300	10000
Painting	2	3	7	200	1000
Nursery	3	1	7	100	300
Electricity	4	1	14	50	1250
Cleaning	5	1	2	50	500

Job List

Job ID	Type Name	priority	cost	receive time	process time
1	Plumbing	0	746	1	3
2	Plumbing	0	209	7	3
3	Plumbing	0	394	11	2
4	Plumbing	0	2489	14	2
5	Plumbing	0	543	18	1
6	Plumbing	0	283	23	3
7	Plumbing	0	153	27	3
8	Plumbing	0	1451	30	3
9	Plumbing	0	2225	35	1
10	Plumbing	0	962	38	3
11	Plumbing	0	686	44	2
12	Plumbing	0	1178	48	1
13	Plumbing	1	2086	54	2
14	Plumbing	0	1912	58	2
15	Plumbing	1	1331	63	1
16	Plumbing	0	444	66	2
17	Plumbing	0	2446	69	2
18	Plumbing	0	2492	72	3
19	Plumbing	0	788	77	1
20	Plumbing	0	2381	80	3
21	Plumbing	0	1530	85	1

The Super-ASH Algorithm - HOA handling

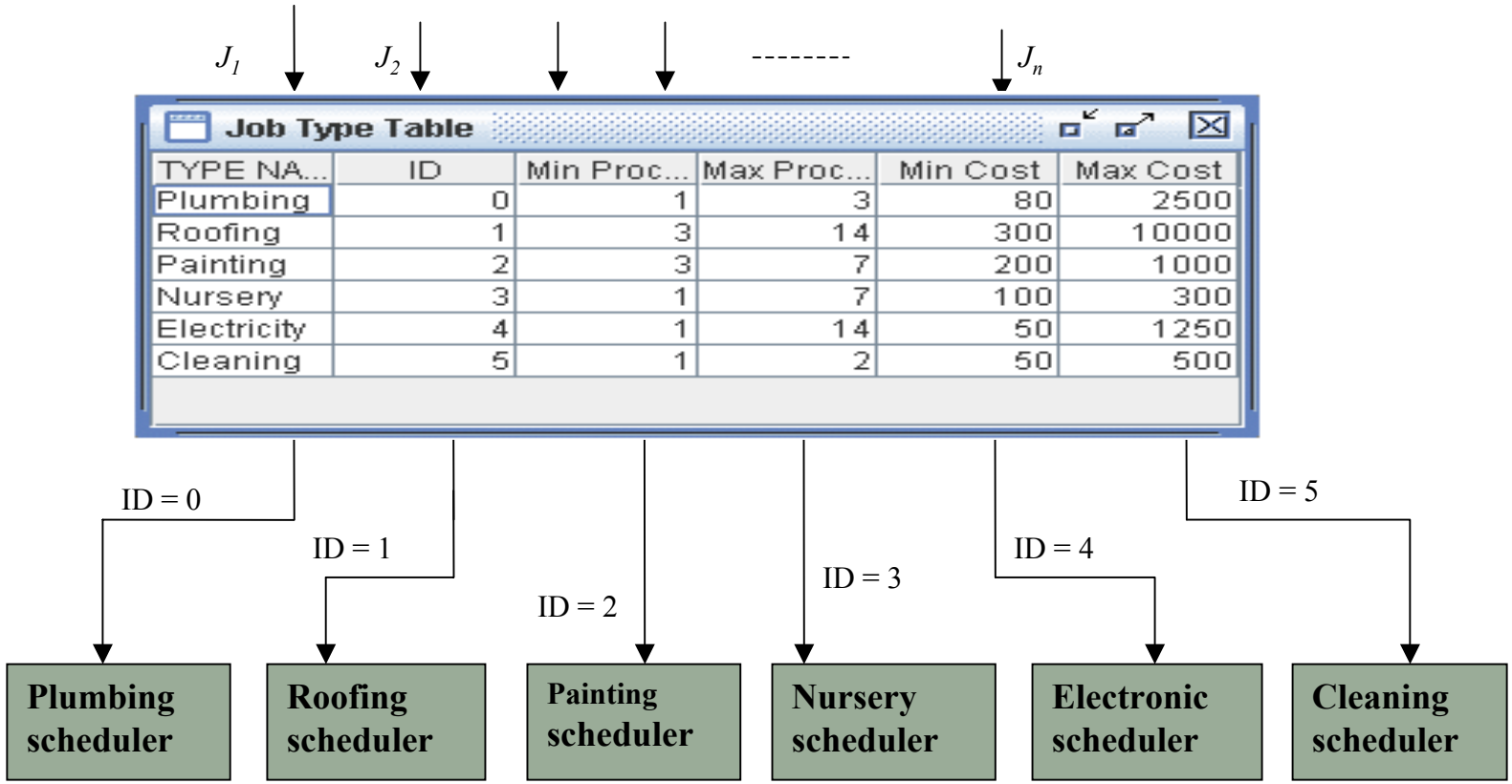
- Step Two: Release jobs



Note: Budget-1 represents the amounts before scheduling all the requested jobs; Budget-2 represents the amounts left after releasing the emergency jobs.Cost-1 equals the costs of to-be-released emergency jobs, Cost-2 equals the costs of to-released normal jobs.

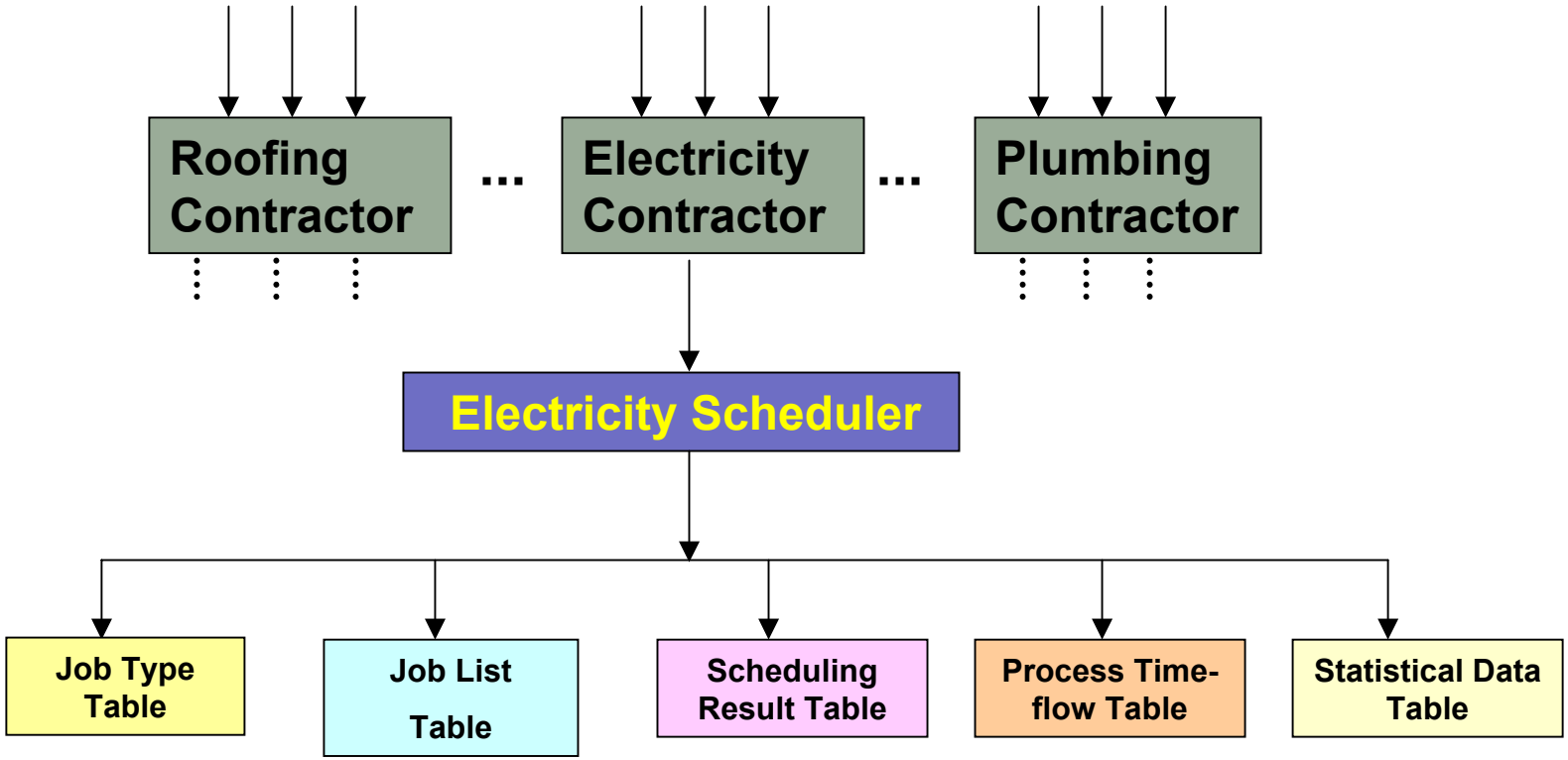
The Super-ASH algorithm

- Step Three: Hand to relative contractors

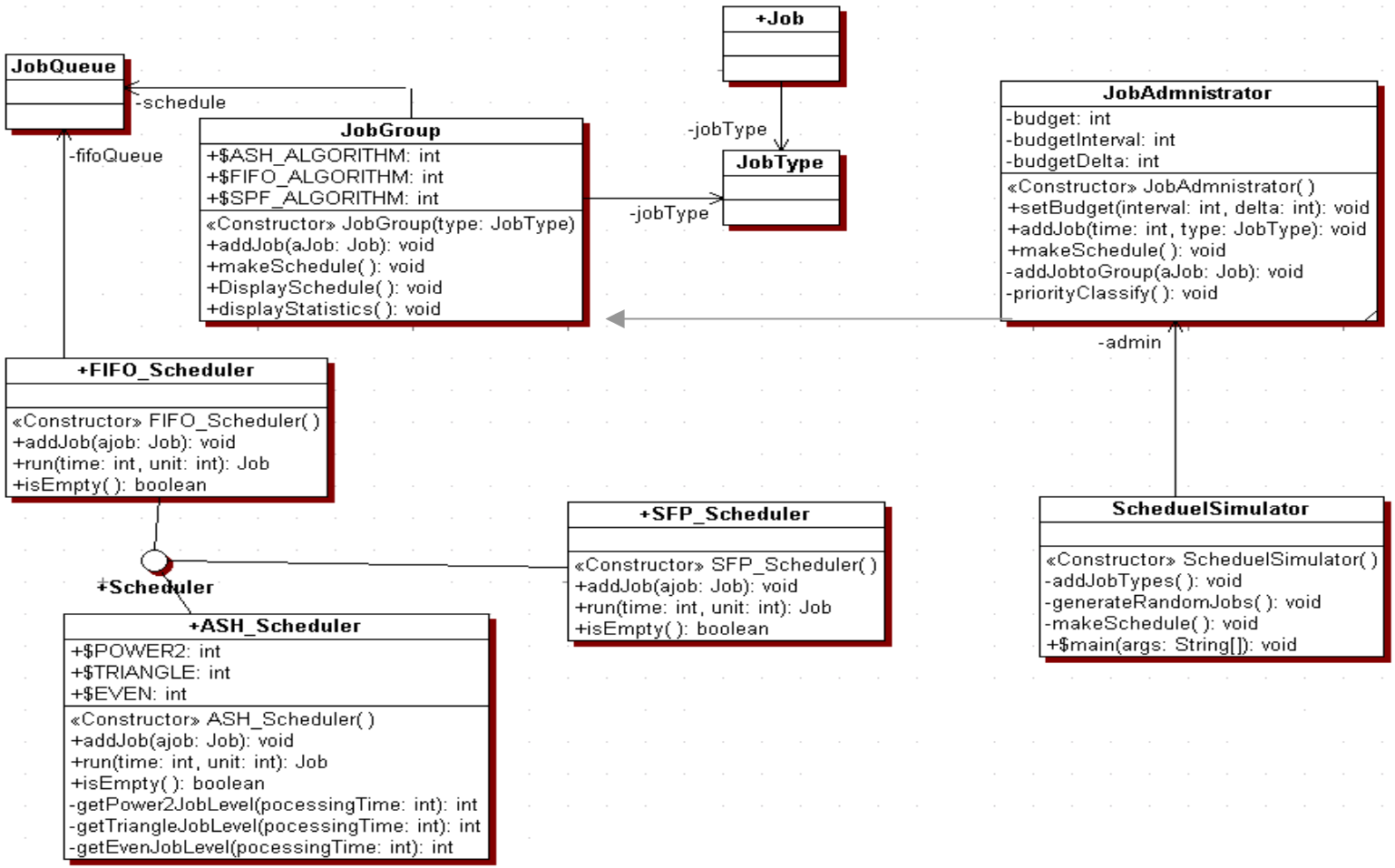


The Super-ASH Algorithm

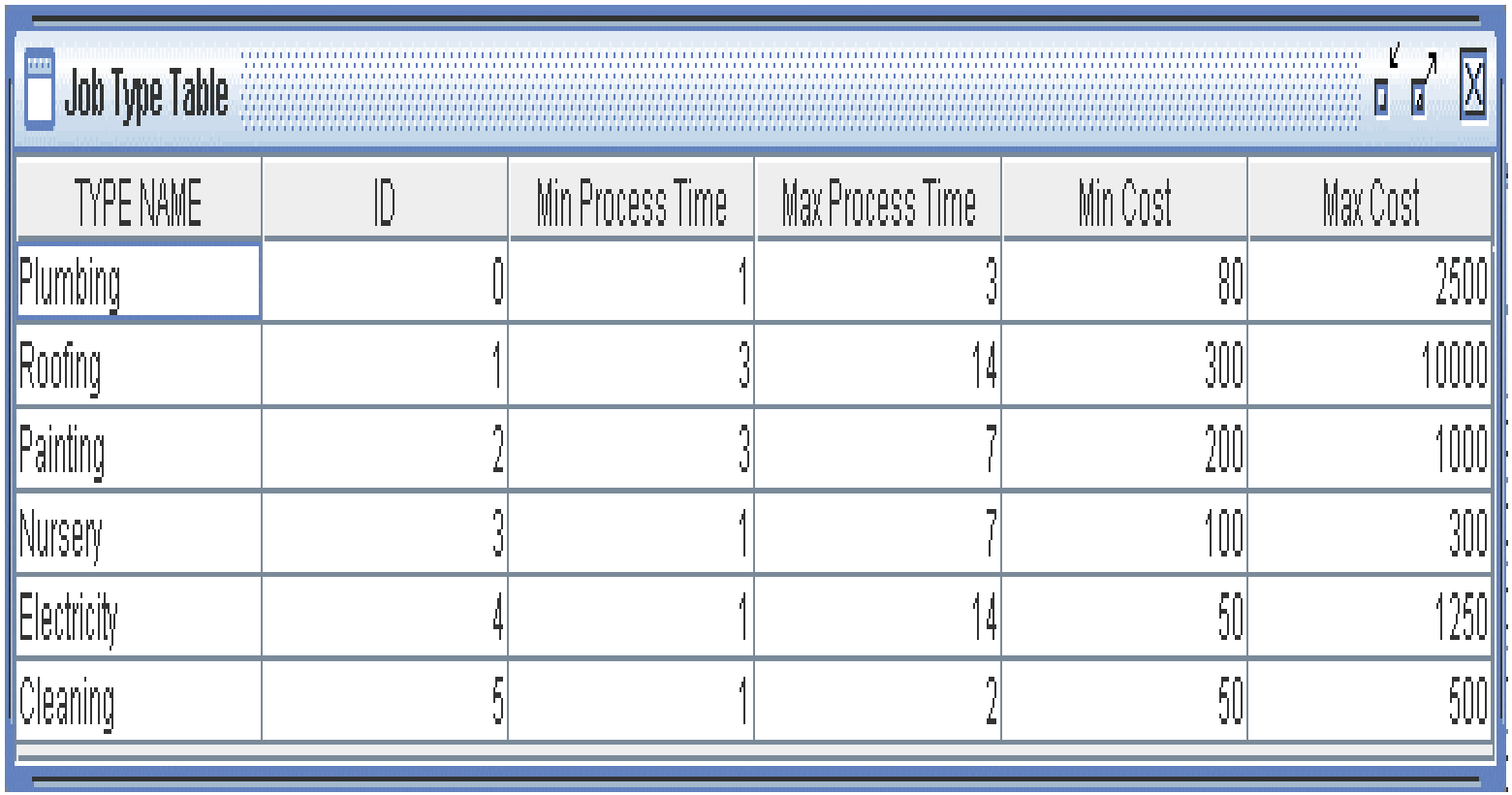
- Step Four: Scheduling



Implementation

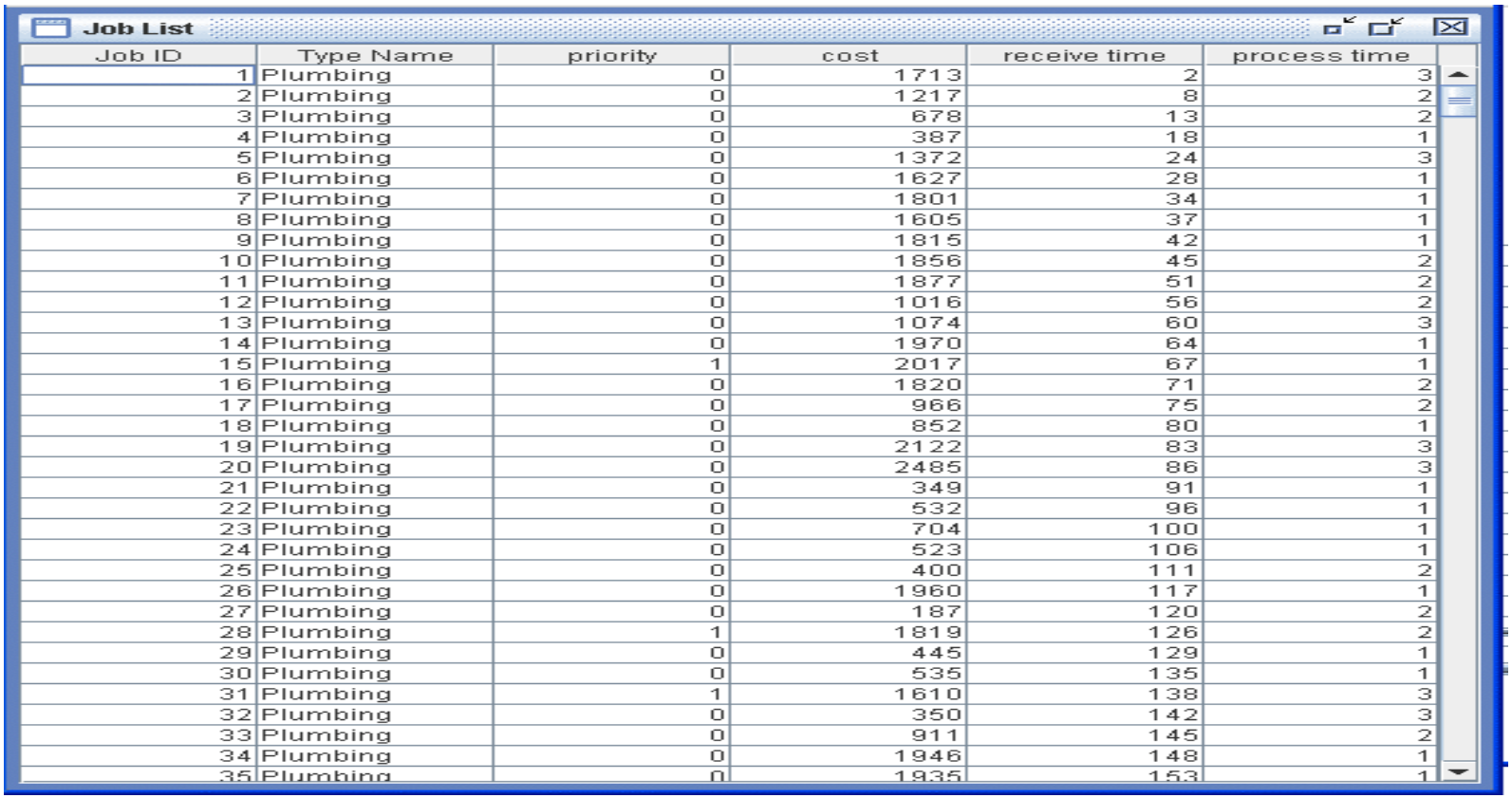


Screen Shot – Job Type Table


Job Type Table

TYPE NAME	ID	Min Process Time	Max Process Time	Min Cost	Max Cost
Plumbing	0	1	3	80	2500
Roofing	1	3	14	300	10000
Painting	2	3	7	200	1000
Nursery	3	1	7	100	300
Electricity	4	1	14	50	1250
Cleaning	5	1	2	50	500

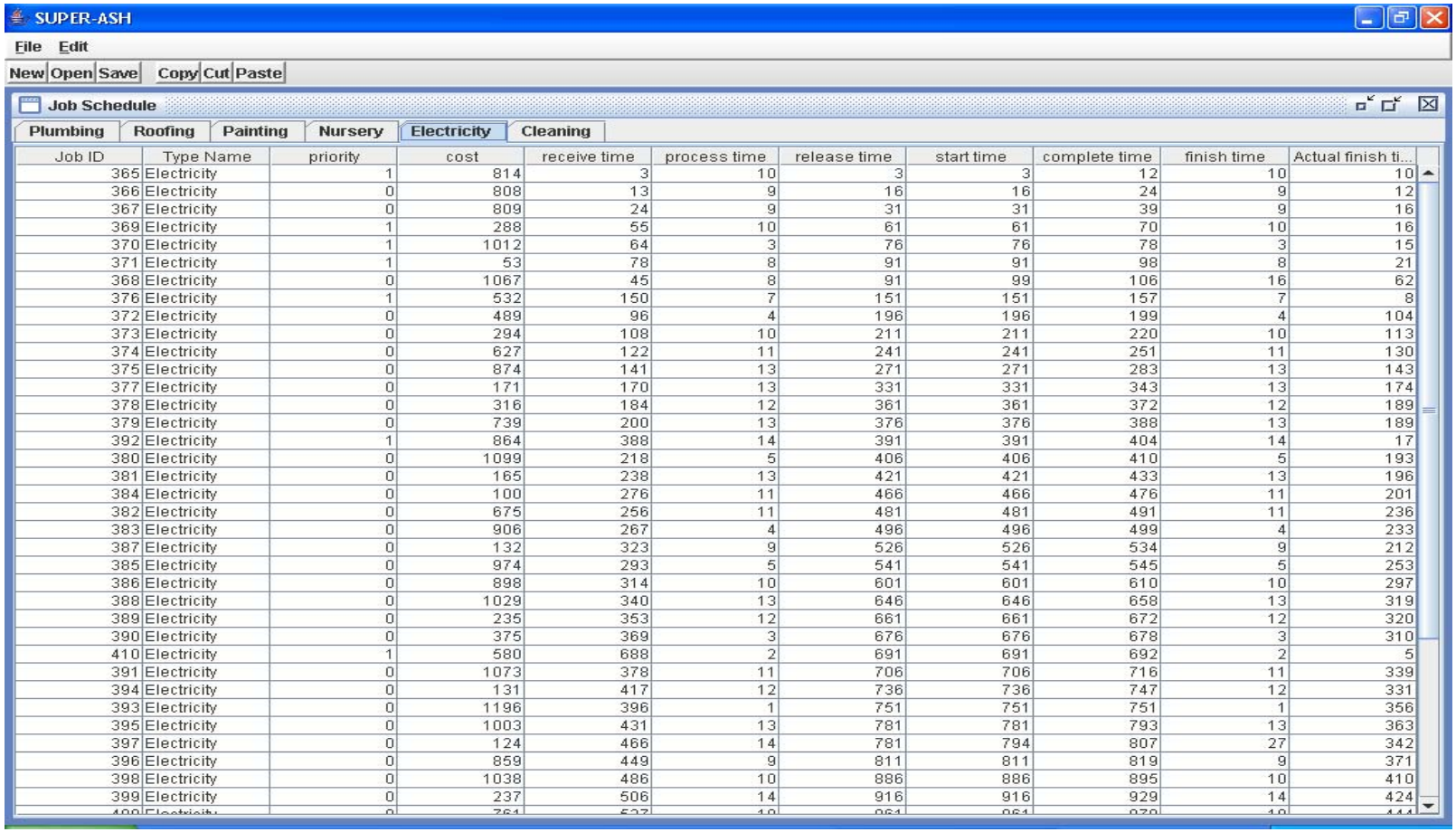
Screen Shot - Job List Table



Job List

Job ID	Type Name	priority	cost	receive time	process time
1	Plumbing	0	1713	2	3
2	Plumbing	0	1217	8	2
3	Plumbing	0	678	13	2
4	Plumbing	0	387	18	1
5	Plumbing	0	1372	24	3
6	Plumbing	0	1627	28	1
7	Plumbing	0	1801	34	1
8	Plumbing	0	1605	37	1
9	Plumbing	0	1815	42	1
10	Plumbing	0	1856	45	2
11	Plumbing	0	1877	51	2
12	Plumbing	0	1016	56	2
13	Plumbing	0	1074	60	3
14	Plumbing	0	1970	64	1
15	Plumbing	1	2017	67	1
16	Plumbing	0	1820	71	2
17	Plumbing	0	966	75	2
18	Plumbing	0	852	80	1
19	Plumbing	0	2122	83	3
20	Plumbing	0	2485	86	3
21	Plumbing	0	349	91	1
22	Plumbing	0	532	96	1
23	Plumbing	0	704	100	1
24	Plumbing	0	523	106	1
25	Plumbing	0	400	111	2
26	Plumbing	0	1960	117	1
27	Plumbing	0	187	120	2
28	Plumbing	1	1819	126	2
29	Plumbing	0	445	129	1
30	Plumbing	0	535	135	1
31	Plumbing	1	1610	138	3
32	Plumbing	0	350	142	3
33	Plumbing	0	911	145	2
34	Plumbing	0	1946	148	1
35	Plumbing	0	1935	153	1

Scheduling Result Table



Job ID	Type Name	priority	cost	receive time	process time	release time	start time	complete time	finish time	Actual finish ti...
365	Electricity	1	814	3	10	3	3	12	10	10
366	Electricity	0	808	13	9	16	16	24	9	12
367	Electricity	0	809	24	9	31	31	39	9	16
369	Electricity	1	288	55	10	61	61	70	10	16
370	Electricity	1	1012	64	3	76	76	78	3	15
371	Electricity	1	53	78	8	91	91	98	8	21
368	Electricity	0	1067	45	8	91	99	106	16	62
376	Electricity	1	532	150	7	151	151	157	7	8
372	Electricity	0	489	96	4	196	196	199	4	104
373	Electricity	0	294	108	10	211	211	220	10	113
374	Electricity	0	627	122	11	241	241	251	11	130
375	Electricity	0	874	141	13	271	271	283	13	143
377	Electricity	0	171	170	13	331	331	343	13	174
378	Electricity	0	316	184	12	361	361	372	12	189
379	Electricity	0	739	200	13	376	376	388	13	189
392	Electricity	1	864	388	14	391	391	404	14	17
380	Electricity	0	1099	218	5	406	406	410	5	193
381	Electricity	0	165	238	13	421	421	433	13	196
384	Electricity	0	100	276	11	466	466	476	11	201
382	Electricity	0	675	256	11	481	481	491	11	236
383	Electricity	0	906	267	4	496	496	499	4	233
387	Electricity	0	132	323	9	526	526	534	9	212
385	Electricity	0	974	293	5	541	541	545	5	253
386	Electricity	0	898	314	10	601	601	610	10	297
388	Electricity	0	1029	340	13	646	646	658	13	319
389	Electricity	0	235	353	12	661	661	672	12	320
390	Electricity	0	375	369	3	676	676	678	3	310
410	Electricity	1	580	688	2	691	691	692	2	5
391	Electricity	0	1073	378	11	706	706	716	11	339
394	Electricity	0	131	417	12	736	736	747	12	331
393	Electricity	0	1196	396	1	751	751	751	1	356
395	Electricity	0	1003	431	13	781	781	793	13	363
397	Electricity	0	124	466	14	781	794	807	27	342
396	Electricity	0	859	449	9	811	811	819	9	371
398	Electricity	0	1038	486	10	886	886	895	10	410
399	Electricity	0	237	506	14	916	916	929	14	424

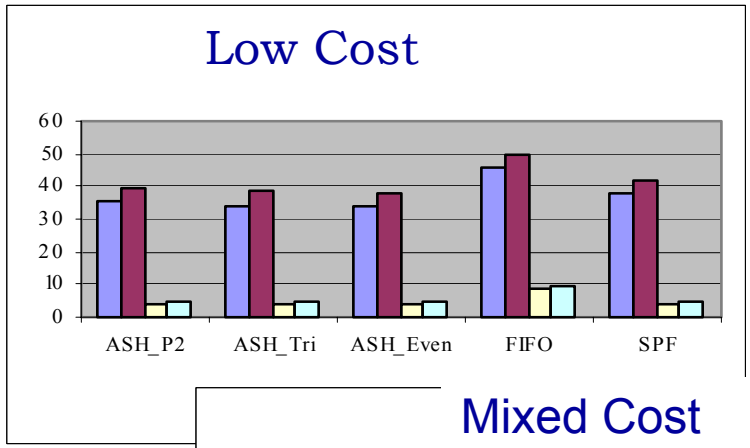
Quick Demo

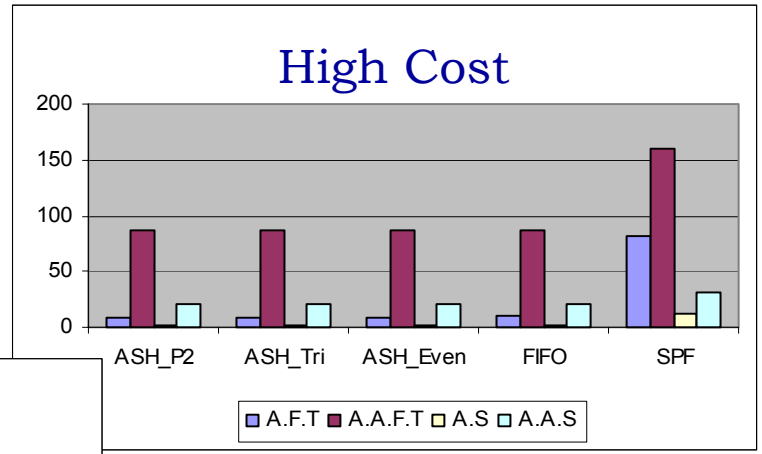
Demo in Process

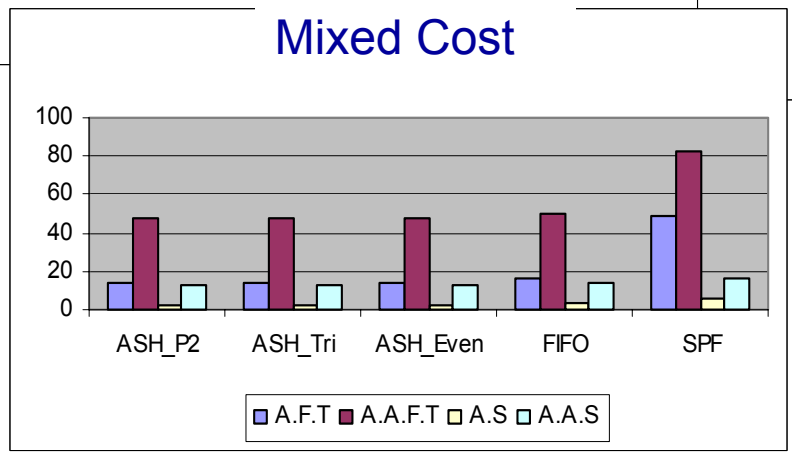
Experiments and Analysis

- Test Case 1: Different Job Costs affects scheduling results
  - Low-Cost
  - High-Cost
  - Mixed-Cost
- Test Case 2: Different Budget setting affects scheduling results
  - Same amounts, Different intervals
  - Same intervals, Different amounts
- Test Case 3: Queue Models result in different scheduling results
  - Power-2
  - Up-down triangle
  - Even

Test Case One – Job Cost







Budget [15, 5000]

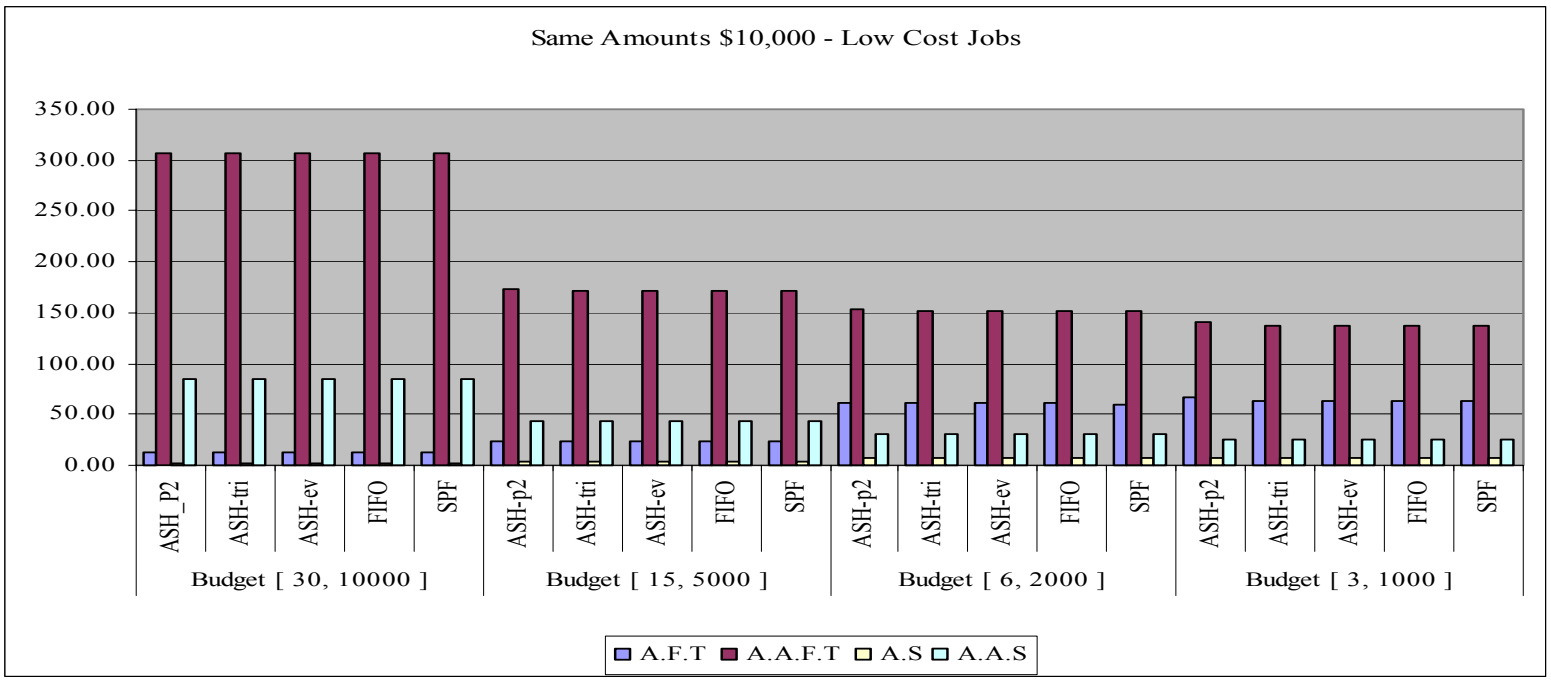
Different Job Set

NOTE: For y-axis, it represents time units per job if they are either AFT or AAFT;

and it represents ratio if they are either AS or AAS.

Test Case Two – Budget and Interval (1 of 4)

- Same Amount, Different Interval (1 of 2)

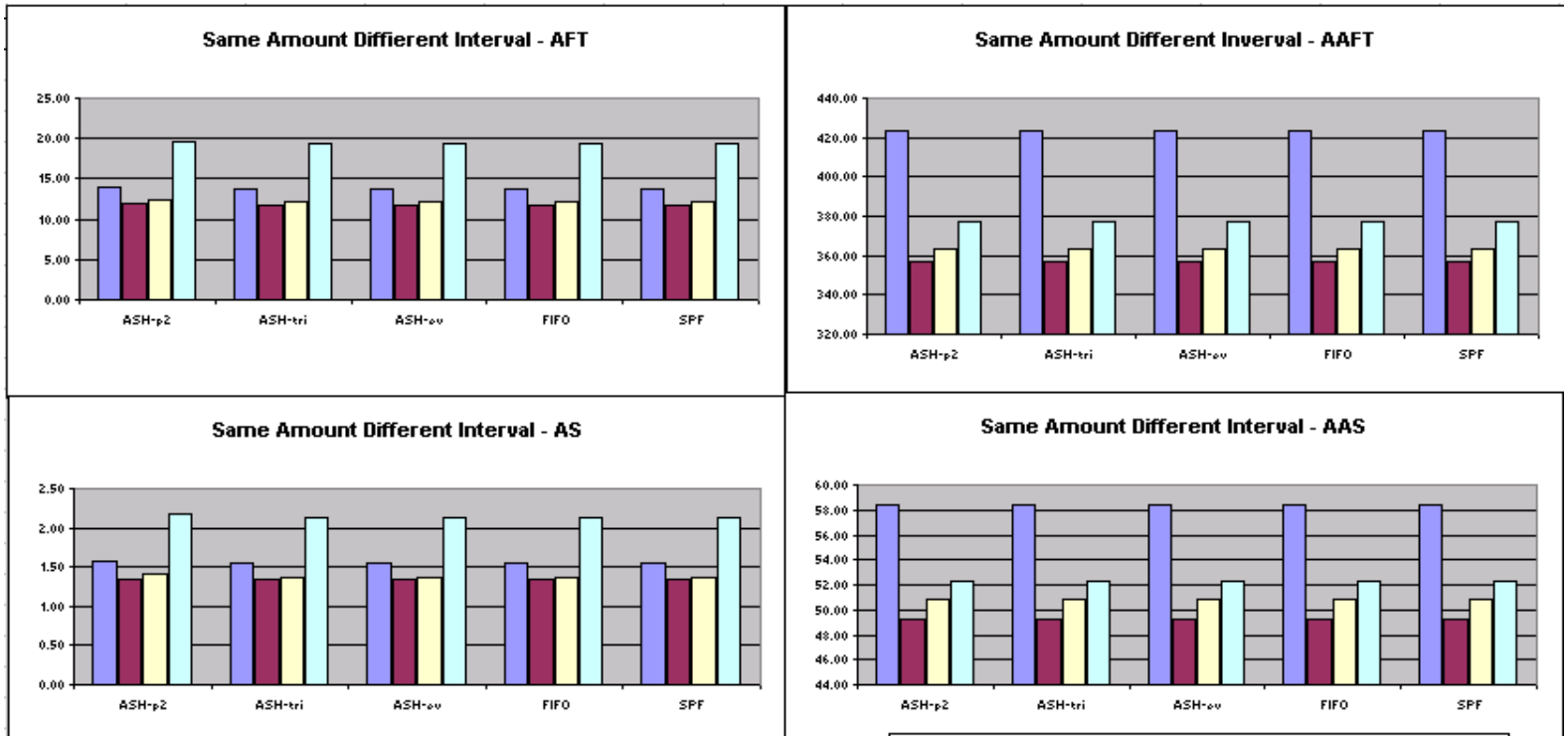


For y-axis, it represents time units per job if they are either AFT or AAFT;

and it represents ratio if they are either AS or AAS.

Test Case Two – Budget and Interval (2 of 4)

- Same Amount, Different Interval (2 of 2)



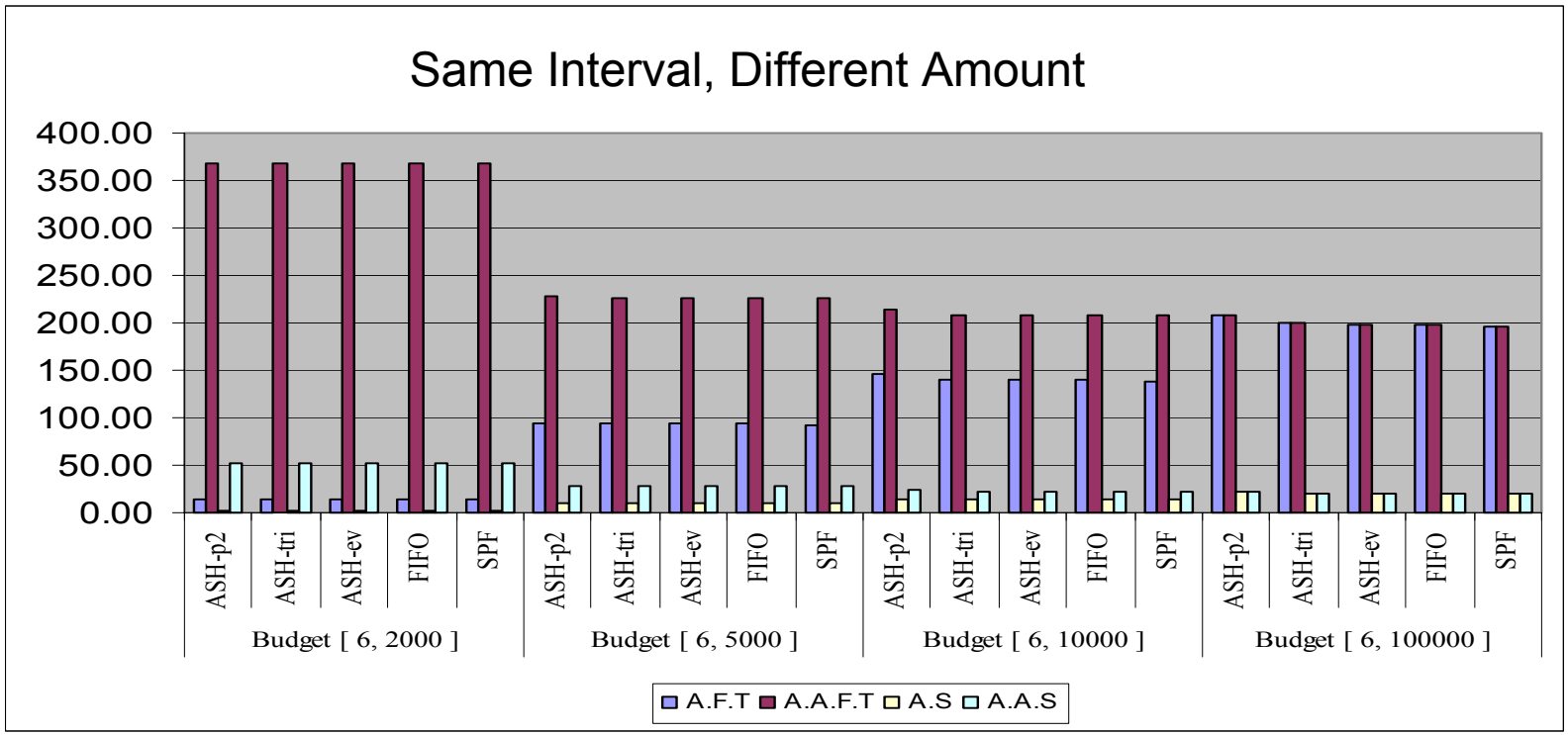
Budget(30,10000) **Budget(15,5000)** **Budget(6,2000)** **Budget(3,1000)**

For y-axis, it represents time units per job if they are either AFT or AAFT;

and it represents ratio if they are either AS or AAS.

Test Case Two – Budget and Interval (3 of 4)

- Same Interval, Different amount

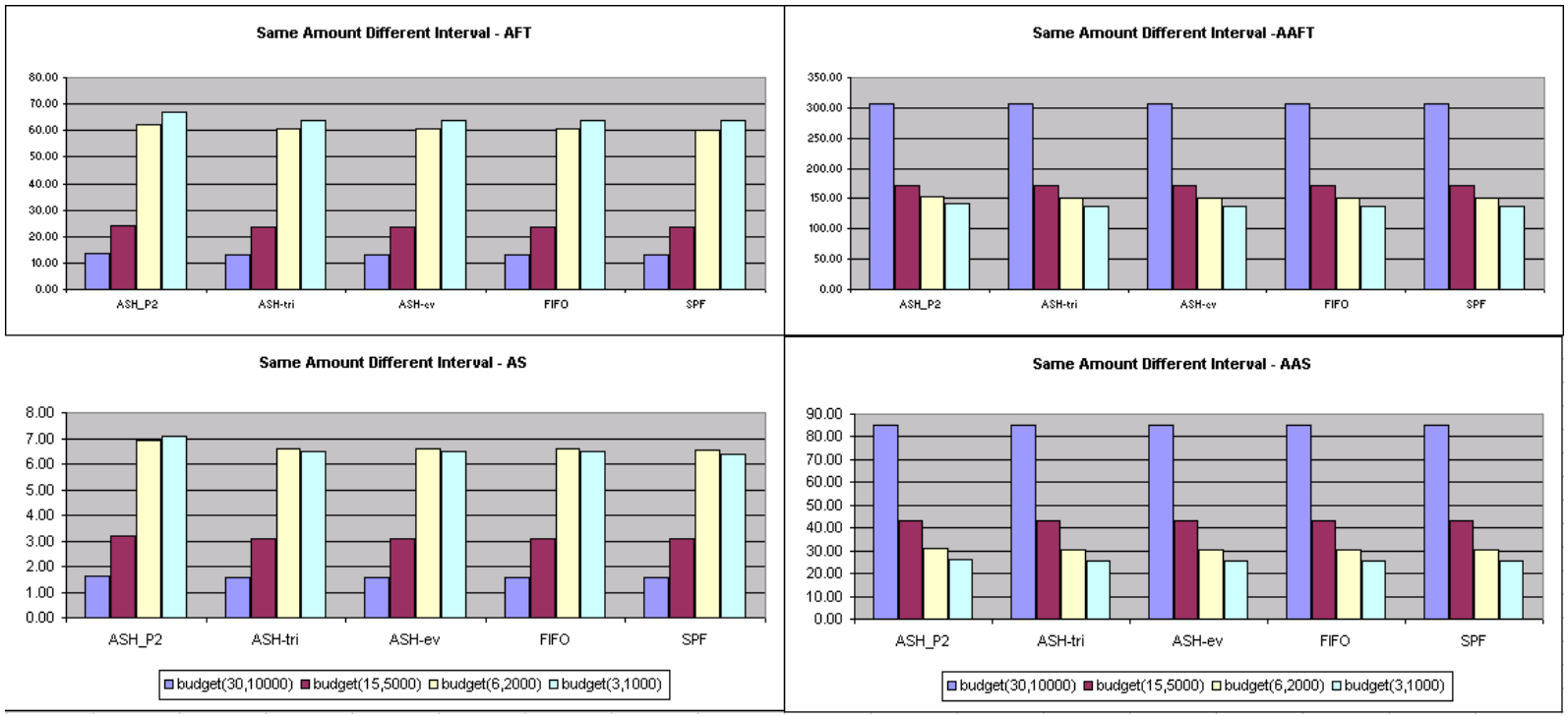


For y-axis, it represents time units per job if they are either AFT or AAFT;

and it represents ratio if they are either AS or AAS.

Test Case Two – Budget and Interval (4 of 4)

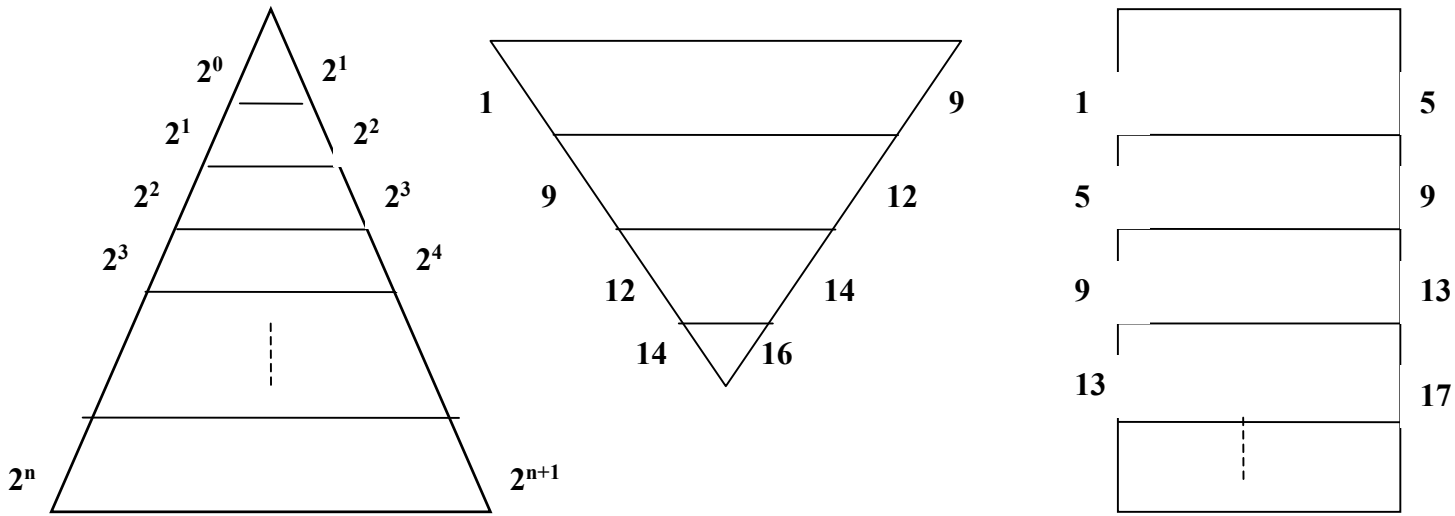
- Same Interval, Different amount



For y-axis, it represents time units per job if they are either AFT or AAFT;

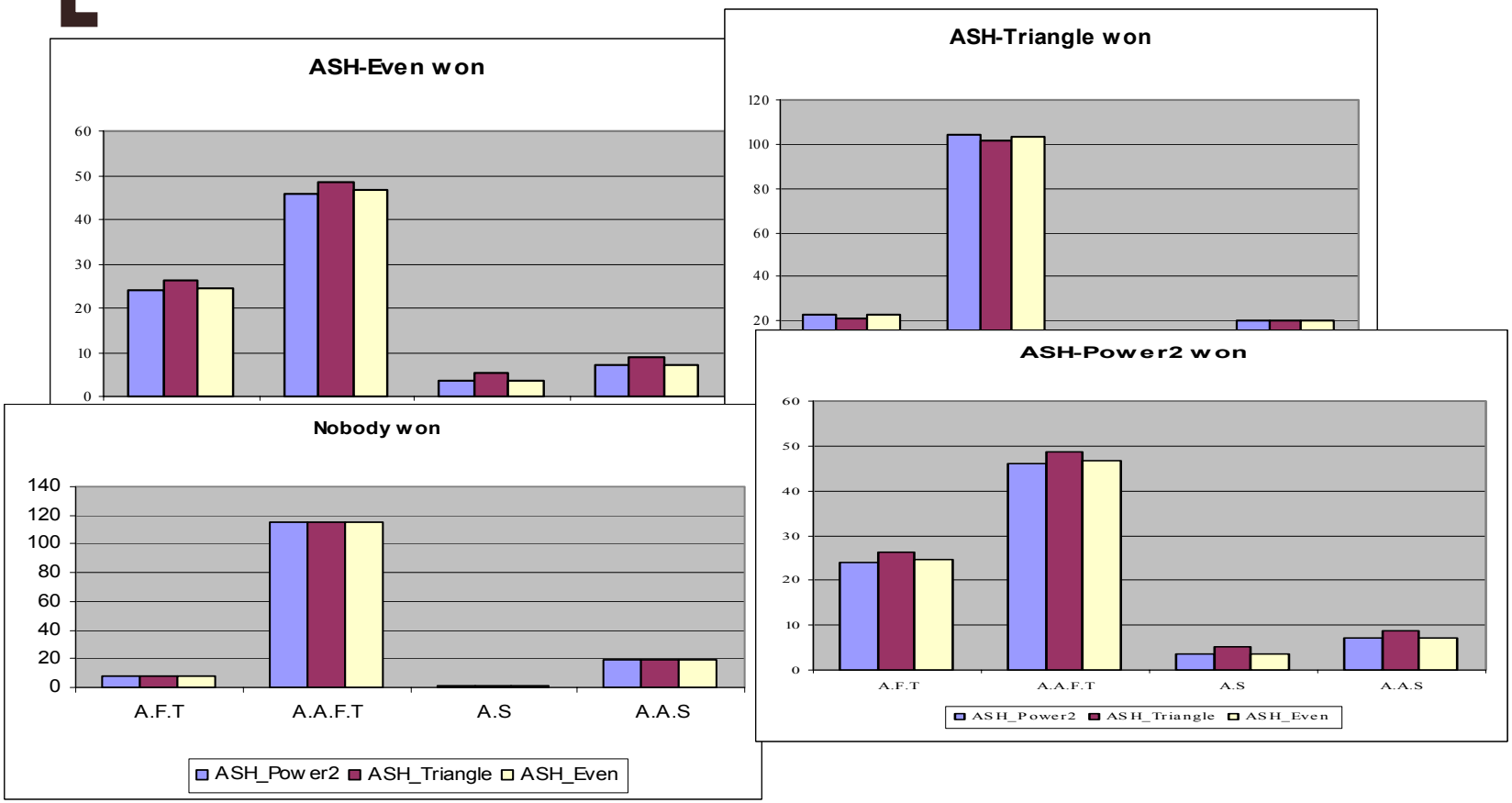
and it represents ratio if they are either AS or AAS.

Test Case Three – Queue Models



Note: From the left to the right, they are Power2 Queue, UpdownTriangle Queue and Even Queue respectively. And the numbers listed on their edges means its width on that layer, for example [2^n , 2^{n+1}).

Test Case Three – Queue Models



For y-axis, it represents time units per job if they are either AFT or AAFT;

and it represents ratio if they are either AS or AAS.

Conclusion

- The Super-ASH algorithm is mostly better than FIFO and SPTF in terms of AAFT and AAS
- The Super-ASH algorithm is flexible for offering HOAs' different configurations
- The Super-ASH algorithm performance is highly dependable to the coming jobs as all the other scheduling algorithms (more)

Conclusion

- More accurate Job Type Table can improve the scheduling performance
- More appropriate budget setting can improve the scheduling performance
- More knowledge about the job property will improve the scheduling performance

Future Work

- Further study on the relationship between job property and queue shape
- Apply heuristic knowledge applied in the Super-ASH algorithm, such as heuristic queue model generator

Acknowledgements

First of all, I would like to thank my advisor Dr. Pollett for your infinite guidance, support and insights. Also, many thanks to Dr. Khuri and Dr. Taylor for sharing your invaluable ideas. Clearly, this thesis would not be possible without you.

Thank you !

Q & A

Thank you