MCTS for FJSP

Reporter : Cheng-Wei Chou
Advisor : Professor Shi-Jim Yen
National Dong Hwa University

Outline

- FJSP
- MCTS
- MCTS for FJSP
- Results
- Conclusion & Future Work

FJSP

- FJSP : Flexible Job-shop Scheduling Problems
- Multi-objective
- Optimization

Problem 8×8 with 27 operation (partial flexibility)

		M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
J1	$O_{1,1}$	5	3	5	3	3	X	10	9
	$O_{1,2}$	10	X	5	8	3	9	9	6
	$O_{1,3}$	X	10	X	5	6	2	4	5
J2	$O_{2,1}$	5	7	3	9	8	X	9	X
	$O_{2,2}$	X	8	5	2	6	7	10	9
	$O_{2,3}$	X	10	X	5	6	4	1	7
	$O_{2,4}$	10	8	9	6	4	7	X	X
J3	$O_{3,1}$	10	X	X	7	6	5	2	4
	$O_{3,2}$	X	10	6	4	8	9	10	X
	$O_{3,3}$	1	4	5	6	X	10	X	7
J4	$O_{4,1}$	3	1	6	5	9	7	8	4
	$O_{4,2}$	12	11	7	8	10	5	6	9
	$O_{4,3}$	4	6	2	10	3	9	5	7
J5	$O_{5,1}$	3	6	7	8	9	X	10	X
	$O_{5,2}$	10	X	7	4	9	8	6	X
	$O_{5,3}$	X	9	8	7	4	2	7	X
	$O_{5,4}$	11	9	X	6	7	5	3	6
J6	$O_{6,1}$	6	7	1	4	6	9	X	10
	$O_{6,2}$	11	X	9	9	9	7	6	4
	$O_{6,3}$	10	5	9	10	11	X	10	X
J7	$O_{7,1}$	5	4	2	6	7	X	10	X
	$O_{7,2}$	X	9	X	9	11	9	10	5
	$O_{7,3}$	X	8	9	3	8	6	X	10
J8	$O_{8,1}$	2	8	5	9	X	4	X	10
	$O_{8,2}$	7	4	7	8	9	X	10	X
	$O_{8,3}$	9	9	X	8	5	6	7	1
	$O_{8,4}$	9	X	3	7	1	5	8	X

FJSP

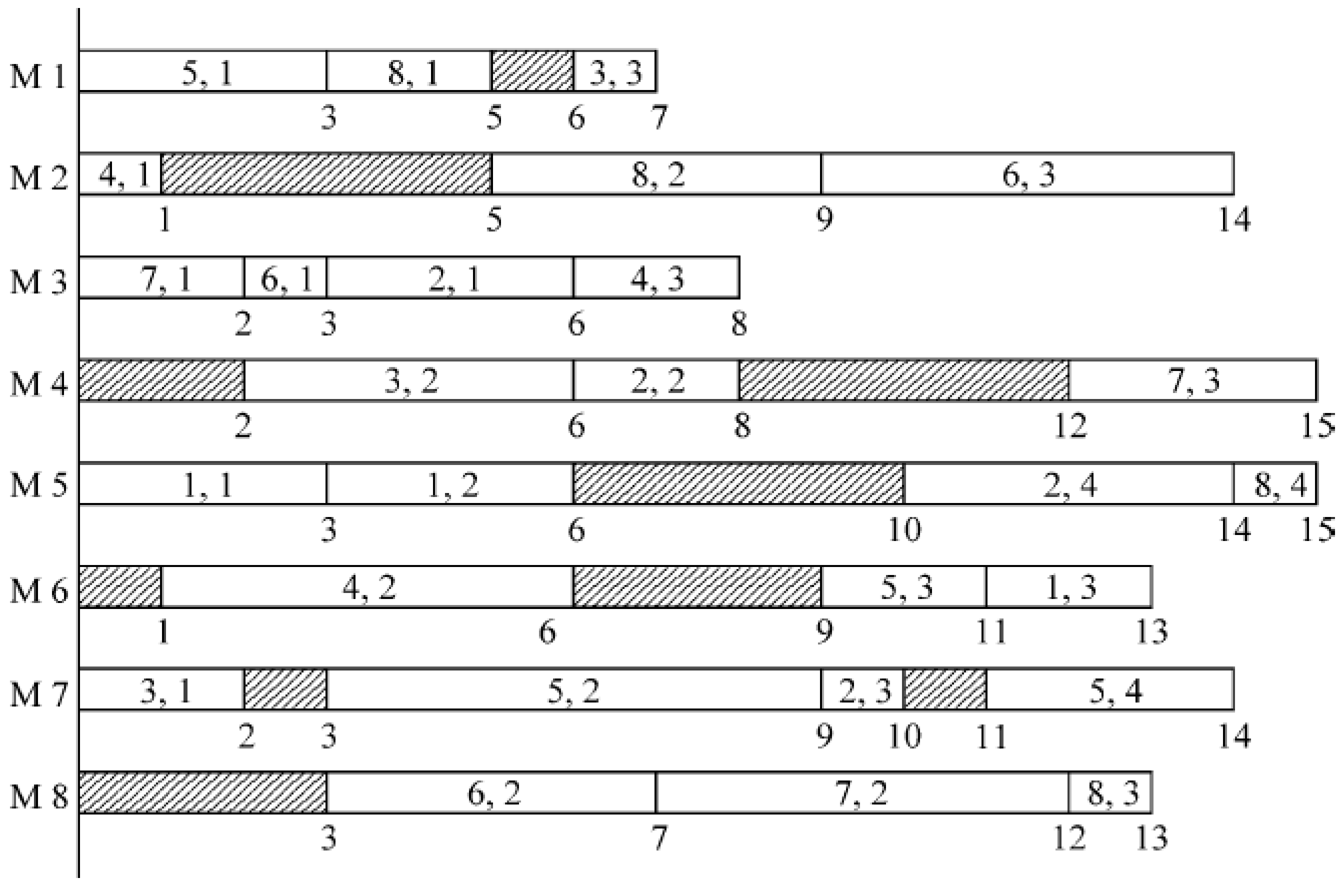


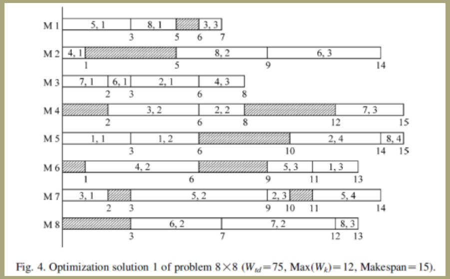
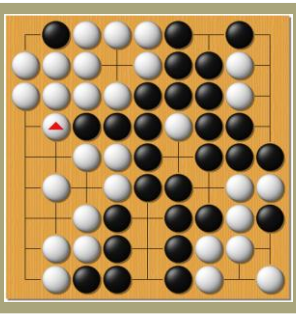
Fig. 4. Optimization solution 1 of problem 8×8 (W_{td}=75, Max(W_k)=12, Makespan=15).

FJSP

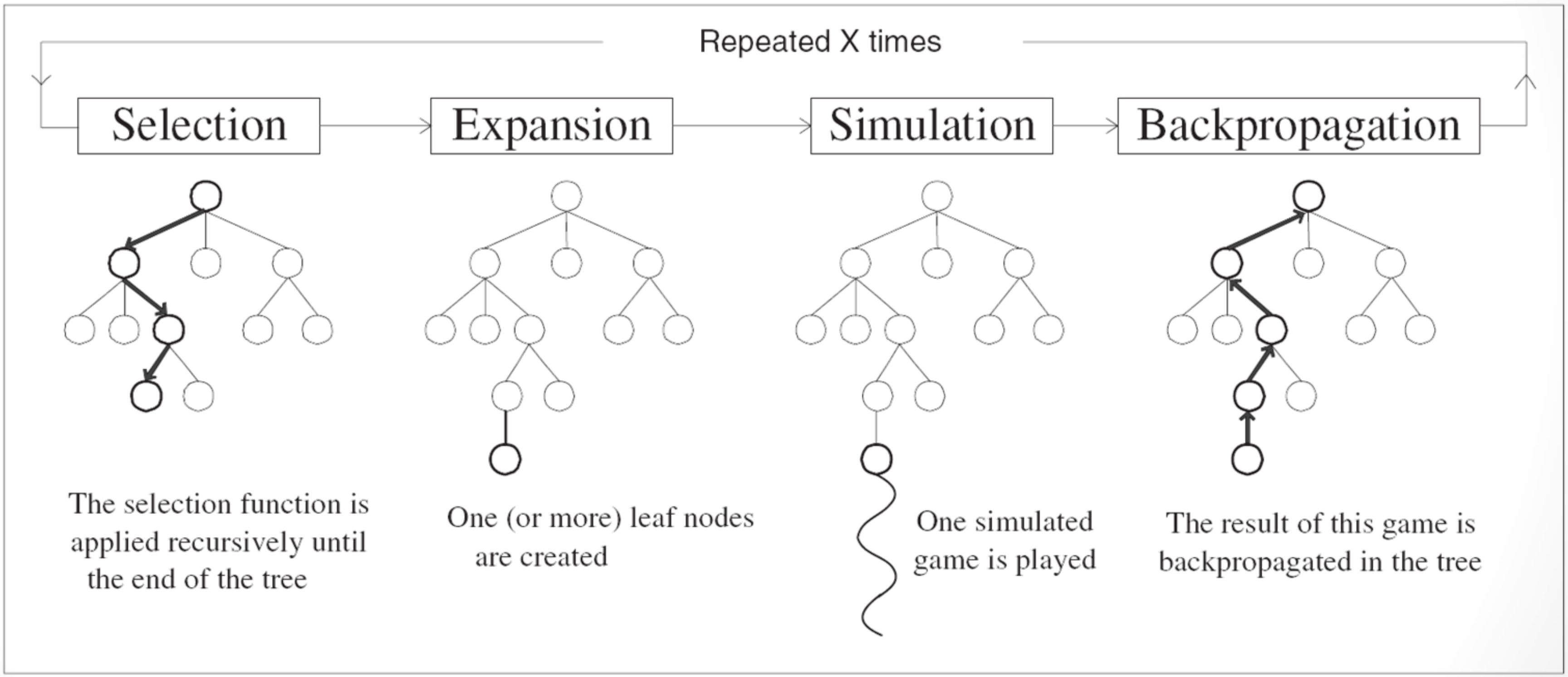
- Our goal : find a non-dominate set of solutions
- What is the concept of dominate?

Solution	W_{td}	Max(W_k)	Makespan
A	75	12	15
B	76	13	17
C	73	13	16

FJSP

FJSP 	**Go** 
Multi-objective	Single-objective
Single Agent	Multi Agent
Find Best Scheduling	Find One Move

MCTS


Repeated X times
Selection
Expansion
Simulation
Backpropagation
The selection function is applied recursively until the end of the tree
One (or more) leaf nodes are created
One simulated game is played
The result of this game is backpropagated in the tree

Simulation

- Steps of simulation：
- 1：Random select next executable job
- 2：Select machine by probability
- 3：Repeat step 1 and 2 until the end of scheduling
- Simulation one time :
  - 1.1(3)
  - 1.1(3) > 2.1(5)
  - 1.1(3) > 2.1(5) > 7.1(6)
  - 1.1(3) > 2.1(5) > 7.1(6) > 2.2(4)
  - 1.1(3) > 2.1(5) > 7.1(6) > 2.2(4) > 3.1(1)
  - 1.1(3) > 2.1(5) > 7.1(6) > 2.2(4) > 3.1(1) > 8.1(2) > ……

Backpropagation

- Update the Wtd, Max(Wk), and Makespan of every node of best sequence
- Update global non-dominate set

(75, 12, 17)				**(75, 12, 16)**
(73 , 13 , 16)	⇨	(75 , 12 , 16)	⇨	(73 , 13 , 16)
(71 , 11 , 18)				(71 , 11 , 18)

Selection

- For unvisited nodes：Select one node by probability distribution

- For visited nodes : $$U_i = v_i + c\sqrt{\frac{\ln N}{n_i}}$$

- Problem : How to judge “past performance” in multi-objective?

Selection

$$U_i = v_i + c\sqrt{\frac{\ln N}{n_i}}$$

- Hydra method：Find best node for every target and randomly select one of them

Expansion

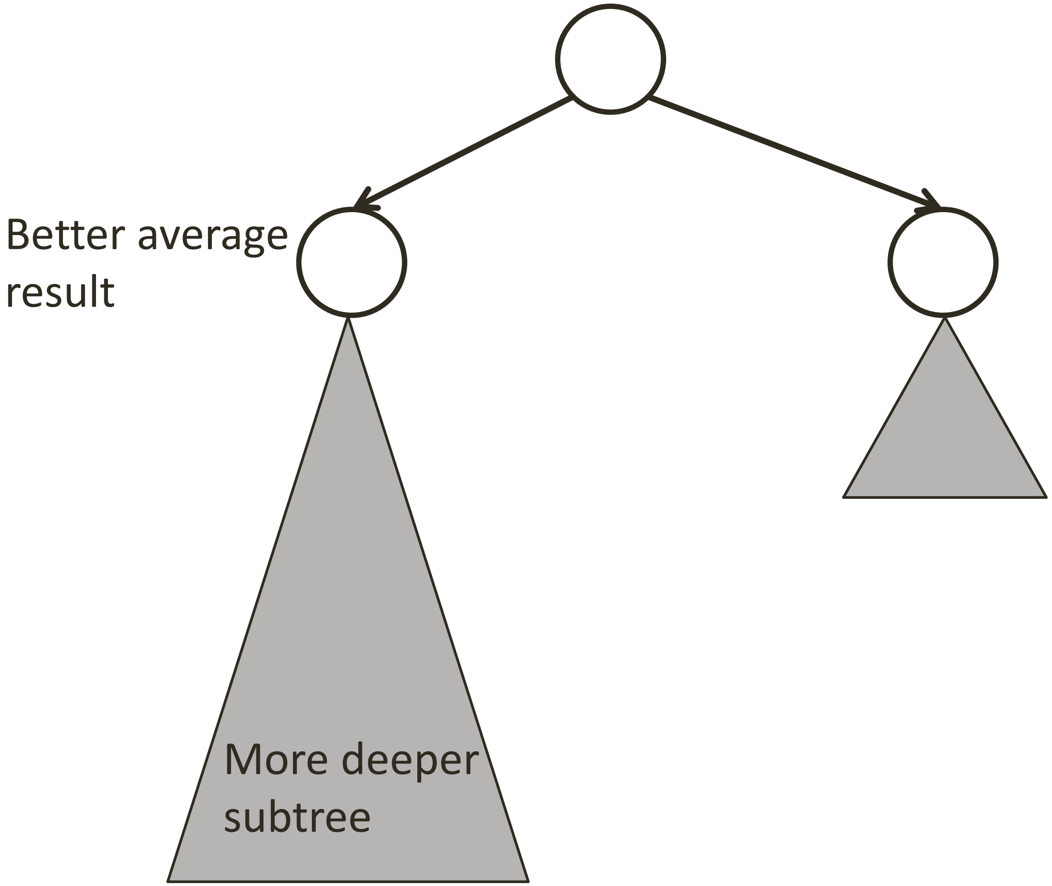
- Create nodes for the best N machines of every executable job.
- If N = 2, and now is start of scheduling, then create below nodes

- J1.1, M4
- J1.1, M3
- J2.1, M1
- J2.1, M2

J \ M	M1	M2	M3	M4
J1.1	10	9	8	7
J1.2	6	7	5	4
J2.1	1	2	3	4
J2.2	7	10	3	2

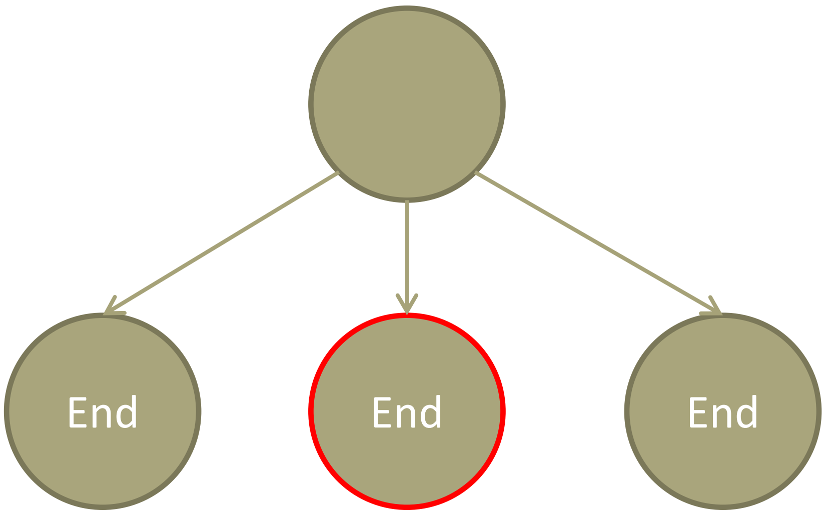
The "Cannot Back" Problem

- Single agent : deeper search, better average result



The Condition of End Search

- When the expansion step cannot be completed, end search



Results

- Our best result :

	8x8				10x10				15x10	
W_{td}	73	75	77	77	41	42	42			
$Max(W_k)$	13	12	11	12	7	5	6			
Makespan	16	15	16	14	8	8	7			

- Now best result of other papers:

	8x8				10x10				15x10	
W_{td}	73	75	77	77	41	42	42	43	91	93
$Max(W_k)$	13	12	11	12	7	5	6	5	11	10
Makespan	16	15	16	14	8	8	7	7	11	11

Conclusion & Future Work

- MCTS can be used to solve the multi-objective flexible job-shop scheduling problems
- The property of single agent will cause the “cannot back” problem
- Next Step : using machine learning method to improve the quality of simulation

Thank you!!

Q&A