IT3708: Project 4

Magnus Moan

May 2, 2017

Task 1

Particle Swarm Optimization and Bees Algorithm

I use the same representation for my Particle Swam Optimization (PSO) and Bees Algorithm (BA). The representation is an indirect representation. Each individual (particle in PSO and bee in BA) is represented as a one dimensional array of length equal to the number of operations ($nOperations$) to be scheduled in the problem. The number of operations is equal to the number of jobs multiplied with the number of machines:

$$nOperations = nJobs \times nMachines$$

The values in the array are floating point numbers in the range $[0, nOperations]$. The floating point numbers are randomly initialized. This representation is usually refered to as a vector in RK space in the literature. To go from this representation to a feasible schedule I perform the following steps:

1. Go from the continuous space to a discrete space by creating a new array with integers in the range $[0, nOperations]$. Sort these integers according to the rank of the floating point numbers. See Figure 1 for example (line with "an integer series").

2. Transform the integer array to another integer array representing the different jobs by taking the modulus of the integer number in each position of the vector. See Figure 1 for example. The resulting vector will consist of $nJobs$ unique numbers which are repeated exactly $nMachines$ times.

3. A feasible schedule can be created from this representation. This is done by traversing the vector of jobs and for each integer (which represents a job) do the following:

   i) Find the next machine this specific job needs to prosseced by.

   ii) Check the next available time for this specific job (it might by occupied by some other machine) as well as the next available job for the machine needed (the machine might be occupied by another job)

   iii) Set the starting time of the current job at the current machine equal to the maximum of the two times found in step ii).

   iv) Update the next available time for both the machine and the job.

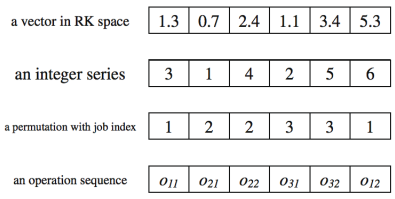


Figure 1: The different steps from individual representation to operation sequence

Ant Colony Optimization

For the Ant Colony Optimization (ACO) I use a direct representation. Each ant have an array consisting of tuples. Each tuple represents an operation. An operation consist of a job number, a machine number and a duration. My algorithm create this array in an iterative fashion for each ant in each iteration. During the creation of the array I make sure that it's only to add operations that keeps the resulting schedule feasible. This is done by always picking between operations where all precessor operations have alredy been added to the array. A precessor operation is an operation that is earlier than the current operation in same jobs technological sequence. When I first have this array I already know that it's feasible. Creating a schedule from it is straight forward. This can be done by iterating over the array in the same fashion as in steps ii)-iv) above. Step i) is not neseccary since I in this representation already know the machine. See Table 1 for an example of an array with operations.

Table 1: Ant representation w/ corresponding problem instance

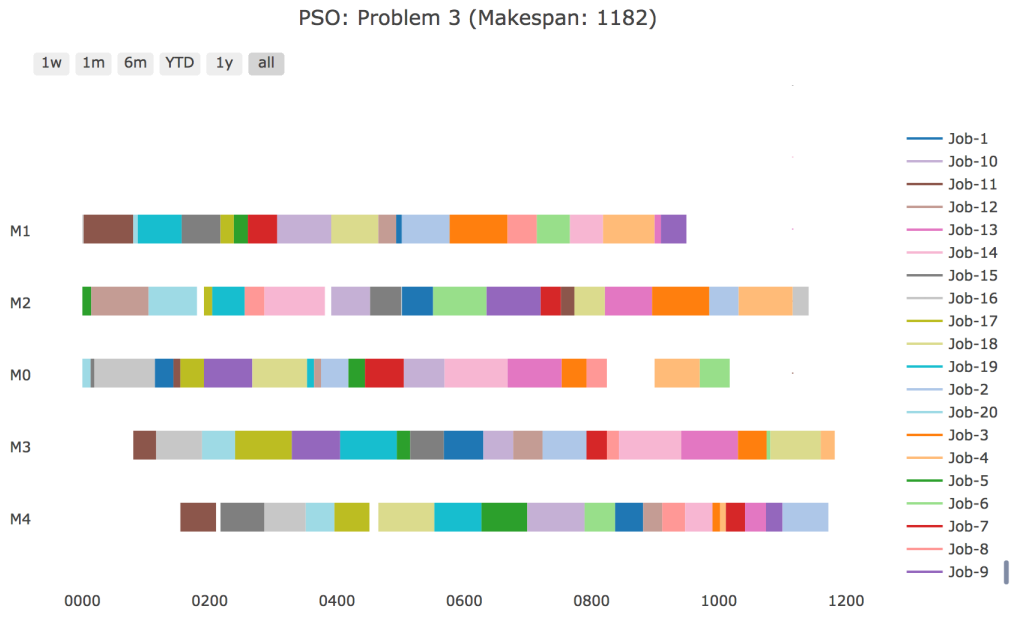
Ant representation

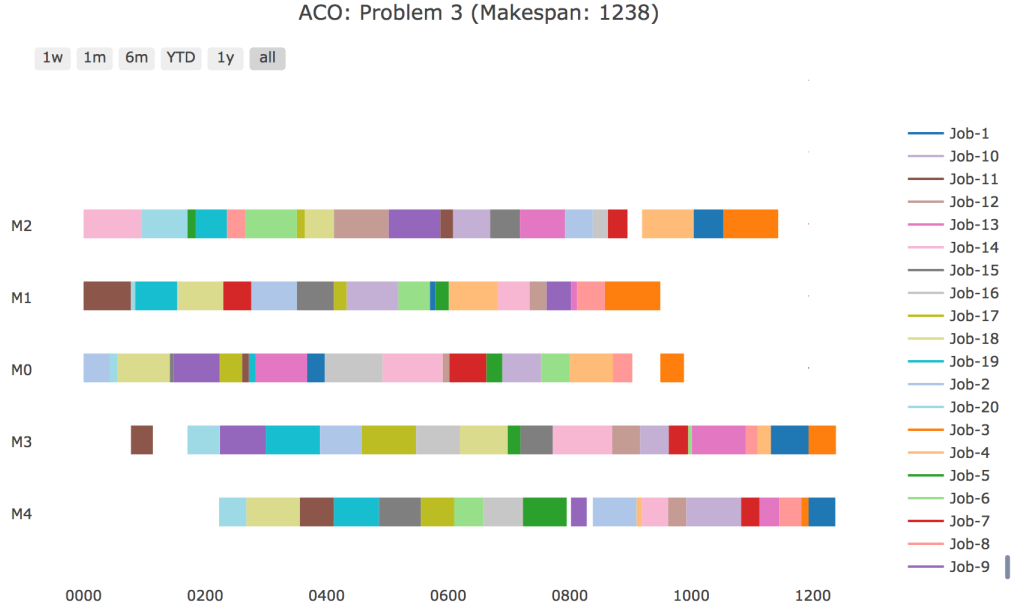
(2,1,1)	(2,0,1)	(1,1,3)	(0,0,2)	(1,0,1)	(1,0,1)

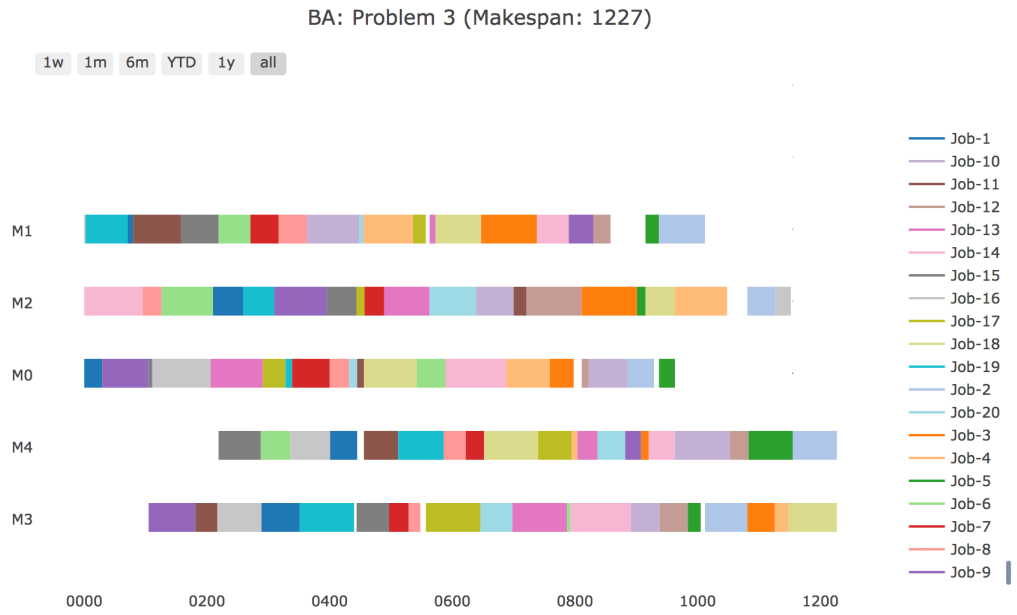
Problem instance. m = machine, t = time

Job-n	(m, t)	(m, t)
Job-0	(0,2)	(1,2)
Job-1	(1,3)	(0,1)
Job-2	(1,1)	(0,1)

Task 2


PSO: Problem 3 (Makespan: 1182)
1w 1m 6m YTD 1y all
M1
M2
M0
M3
M4
0000 0200 0400 0600 0800 1000 1200
Job-1 Job-10 Job-11 Job-12 Job-13 Job-14 Job-15 Job-16 Job-17 Job-18 Job-19 Job-2 Job-20 Job-3 Job-4 Job-5 Job-6 Job-7 Job-8 Job-9


ACO: Problem 3 (Makespan: 1238)
1w 1m 6m YTD 1y all
M2
M1
M0
M3
M4
0000 0200 0400 0600 0800 1000 1200
Job-1 Job-10 Job-11 Job-12 Job-13 Job-14 Job-15 Job-16 Job-17 Job-18 Job-19 Job-2 Job-20 Job-3 Job-4 Job-5 Job-6 Job-7 Job-8 Job-9


BA: Problem 3 (Makespan: 1227)
1w
1m
6m
YTD
1y
all
M1
M2
M0
M4
M3
0000
0200
0400
0600
0800
1000
1200
Job-1
Job-10
Job-11
Job-12
Job-13
Job-14
Job-15
Job-16
Job-17
Job-18
Job-19
Job-2
Job-20
Job-3
Job-4
Job-5
Job-6
Job-7
Job-8
Job-9