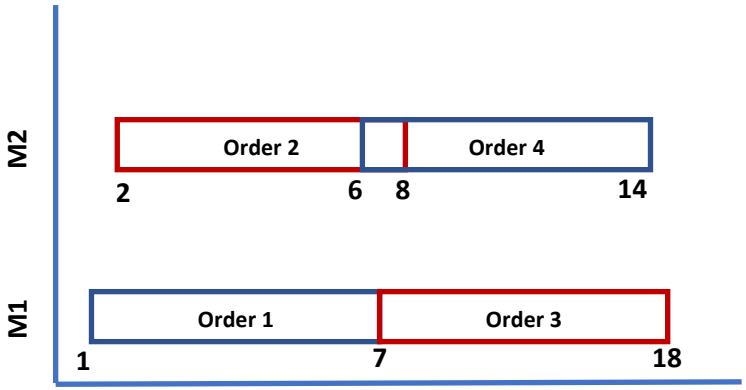
Part a: Provide a mathematical formulation (along with the definition of the variables) for the job shop scheduling problem (discussed in class). Use the notation L to denote assignment between a machine and an order. Use the notation K for sequencing of the orders. Use the indices *p* and *q* for orders and machines respectively. Explain the set of equations in detail [2 marks]. Change of notation is not permitted.

Part b: Given the infeasible solution (due to overlapping of orders on M2), mathematically show that the formulation leads to a set of inconsistent equations [2 marks]. If part b is attempted but is not correctly answered, it will carry a penalty of 2 marks.



Part c: Ignore M2 in the given Gannt Chart. List the values of all the decision variables [1 mark]

Pseudocode

Input: Fitness function, lb, ub, N_p, T , w, c_1 and c_2

p_{best}: N_p x D, f_{pbest}: N_p x 1
g_{best}: 1 x D, f_{gbest}: 1 x 1

1. Initialize a random population (P) and velocity (v) within the bounds
2. Evaluate the objective function value (f) of P ← FE = N_p
3. Assign p_{best} as P and f_{pbest} as f
4. Identify the solution with best fitness and assign that solution as g_{best} and fitness as f_{gbest}

For T iterations
Total FE $= N_p + N_pT$

for t = 1 to T
  for i = 1 to N_p
    Determine the velocity (v_i) of i^{th} particle
    Determine the new position (X_i) of i^{th} particle
    Bound X_i
  } Generation

$$v_i = wv_i + c_1r_1\left(p_{best,i} - X_i\right) + c_2r_2\left(g_{best} - X_i\right)$$

$$X_i = X_i + v_i$$

    Evaluate the objective function value (f_i) of i^{th} particle ← FE =1
    Update the population by including X_i and f_i
    Update $p_{best,\,i}$ and f_{pbest}
    Update g_{best} and f_{gbest}
  } Memorizing
  end
end

$$\left.\begin{aligned} p_{best,i} &= X_i \\ f_{p_{best,i}} &= f_i \end{aligned}\right\} if\ f_i < f_{p_{best,i}}$$

$$\left.\begin{aligned} g_{best} &= p_{best,i} \\ f_{g_{best}} &= f_{p_{best,i}} \end{aligned}\right\} if\ f_{p_{best,i}} < f_{g_{best}}$$

```
function [Xbest,Fbest] = PSOfunc(FITNESSFUN,lb,ub)
Maxiter = 100; Npop = 100; w = 0.7; c1 = 1.5; c2 = 1.5
D = length(lb);
for j = 1:Maxiter
   for i = 1:Npop
            v(j,:) = w*v(j,:) + c1*rand(1,D).*(pbest(j,:)-pop(j,:)) - c2*rand(1,D).*(gbestPop - pop(j,:));
        pop(j,:) = pop(j,:) - v(j,:);
       pop(j,:) = min(pop(j,:),lb);
       pop(j,:) = max(pop(j,:),ub);

       if obj(i)<pbest_obj(i)
          pbest_obj(i) = obj(i);
       end

   end
end
```