# COMP4418, 2018 – Assignment 3

ZID: 5097690 Name: Baixiang Guan

## Question 1:

### (a) the uncovered set:

| Alternative | Alternatives can be reached in at most two steps |
|---|---|
| a | b, c, d, e, f, g |
| b | a, c, d, e, f, g |
| c | a, b, d, e, f, g |
| d | a, b, c, e, f, g |
| e | a, b, d, f, g |
| f | b, c, d, e, g |
| g | a, c, d, e, f |

The uncovered set is {a, b, c, d}.

### (b) the top cycle

| Alternative | One path to visit all other Alternatives |
|---|---|
| a | a→b→d→e→f→g→c |
| b | b→d→e→f→g→c→a |
| c | c→a→b→d→e→f→g |
| d | d→e→f→g→c→a→b |
| e | e→f→g→c→a→b→d |
| f | f→g→c→a→b→d→e |
| g | g→c→a→b→d→e→f |

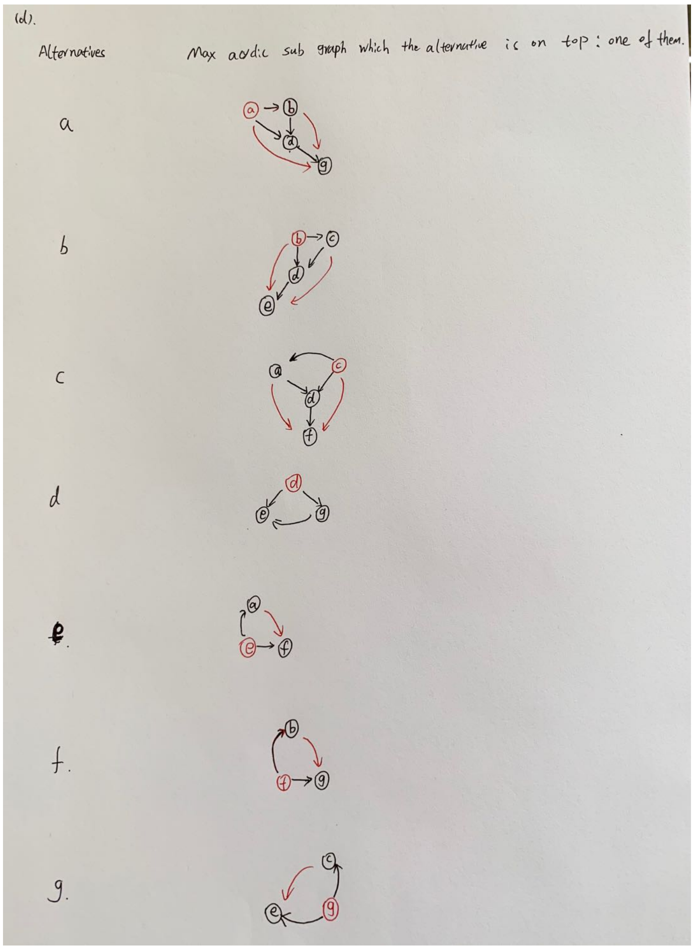
**The top cycle is {a, b, c, d, e, f, g},** because these Alternatives can reach all others by a path.

### (c) the set of Copeland winners:

| Alternative | Dominates | Copeland score |
|---|---|---|
| a | (a, b), (a, d), (a, f), (a, g) | 4 |
| b | (b, c), (b, d), (b, e), (b, g) | 4 |
| c | (c, a), (c, d), (b, e), (c, f) | 4 |
| d | (d, e), (d, f), (d, g) | 3 |
| e | (e, a), (e, f) | 2 |
| f | (f, b), (f, g) | 2 |
| g | (g, c), (g, e) | 2 |

**The set of Copeland winners is {a, b, c},** because the Copeland score of these Alternative are 4 which is highest.

## (d) the set of Banks winners:




**The set of Banks winners is {a, b, c},** because we can see in the picture such that the max sub graphs which contain the max number of alternatives are a, b and c.

## (e) the set of Condorcet winners:

**The set of Condorcet winners is $\emptyset$.** The main reason is that there is a cycle in the graph which is **a→b→d→e→f→g→c→a.** Hence, because Condorcet winner does not allow parents, but, in this cycle, every alternative has at least one parent. **Therefore, there is no Condorcet winner.**

## Question 2:

**(1).**

**Prove or disprove that the Condorcet winner always has the maximum Borda score among all the alternatives.**

**The Condorcet winner is not always the maximum Borda score among all the alternatives.**

Assume there is a set of agents N = {1,2,3} and there is a set of alternatives A={a1, a2⋯, am}, m>3.

Then if a1 is the Condorcet winner which means that the majority agents prefer a1 than the other alternatives. Hence, we can assume that agents 1 and 2 prefer a1 than other alternatives, and agents 3 prefer other alternatives than a1 which means than a1 is at the end of agents 3 preference. So, if there exist one alternative such as a2 is lower than a1 just one ranking in the agents 1 and 2 preference, which means that such as a1 get m-1 Borda score and a2 get m-2 Borda score in the agents 1 and 2 preference. However, in agents 3 preference, if the Borda score that a2 get more than a1 three score, for example, a2 get m-1 Borda score and a get 0 Borda score. Then, the the Borda score of a2 among all the alternatives is m-2+m-2+m-1 = 3m-5, and the Borda score of a1 is m-1+m-1 = 2m-2 Hence, the score of a2-a1 = m-3, and because m>3, Therefore, a1 cannot be the maximum Borda score among all the alternatives. So, The Condorcet winner is **not** always the maximum Borda score among all the alternatives.

There is a example, N={1,2,3} A = {a,b,c,d}

1: a>b>c>d

2: a>b>d>c

3: b>c>d>a

So, a is the Condorcet winner, and b has the maximum Borda score among all the alternatives. But a is not b. Hence,The Condorcet winner is **not** always the maximum Borda score among all the alternatives.

**(2).**

**Prove or disprove that the Condorcet winner has at least half of the Borda score of the Borda winner.**

**The Condorcet winner has at least half of the Borda score of the Borda winner.**

It is assumed that Condorcet winner exists, and there exist agents and alternatives. Because if not, it is no meaning.

Then, we can assume that there is a set of agents N = {1, 2, ⋯⋯, n} and there is a set of alternatives A = {a1, a2⋯, am}, $m \geq 1$. And a1 is the Condorcet winner, a2 is the Borda winner.

And suppose that S1 is the Borda score of a1, and S2 is the Borda score of a2.

So, if the Condorcet winner has at least half of the Borda score of the Borda winner, it means that $\mathrm{S2} - \mathrm{S1} \leq \mathrm{S2}/2$ which is $S2 \leq 2S1$. Hence, if we can proof that the if the largest S2 is smaller the smallest S1, then we can proof the Condorcet winner has at least half of the Borda score of the Borda winner. Because a1 is Condorcet winner which means that there are at least $\lceil n/2 \rceil$ agents prefer a1 than a2. And because we want to the largest S2 and the smallest S1, So, we can let a2 is the second preference in $\lceil n/2 \rceil$ agents which is the amount of smallest agents that prefer a1 than a2, and the top preference in the rest $n - \lceil n/2 \rceil$ agents. Hence,

$$S2 = (m-2) * \lceil n/2 \rceil + (m-1) * (n - \lceil n/2 \rceil) = (m-1) * n - \lceil n/2 \rceil$$

And we can also simply let the Borda score of a1 is the top preference in $\lceil n/2 \rceil$ agents because a1 is needed to preferred than a2 and a2 is the second preference in $\lceil n/2 \rceil$ agents, which means that a1 need the top preference in $\lceil n/2 \rceil$ agents, and also the last preference in the $n - \lceil n/2 \rceil$ agent's preference list can ensure that S1 can get the smallest value. (The main reason is that if a1 is Condorcet winner, it needs to compare to any other alternatives, and we can easily assume that a1 is the top preference of at $\lceil n/2 \rceil$ agents. And the rest are at the end. For example, if a1 is not the top preference in the $\lceil n/2 \rceil$ agents, then this means there exist other alternative that the agent preferred than a1 and Borda socre is lareger at least 1. So, in the rest $n - \lceil n/2 \rceil$ agent, a1 must preferred than that alternative and the Borda socre is larger at least 1 because a1 is Condorcet winner. We can assume all alternatives compare to the a1 have the situation. Hence, this means that if a1 in the $\lceil n/2 \rceil$ agents is the top preference, then it can be the last at the rest of agents.) So,

$$S1 = (m-1) * \lceil n/2 \rceil$$

And $S2 - 2S1 = (m-1) * (n - 2 * \lceil n/2 \rceil) - \lceil n/2 \rceil$

Because of $m \geq 1$ $and$ $n - 2 * \lceil n/2 \rceil \leq 0$, hence $S2 - 2S1 \leq 0$ which is $S2 \leq 2S1$.

**Therefore, the Condorcet winner has at least half of the Borda score of the Borda winner.**

## Question 3:

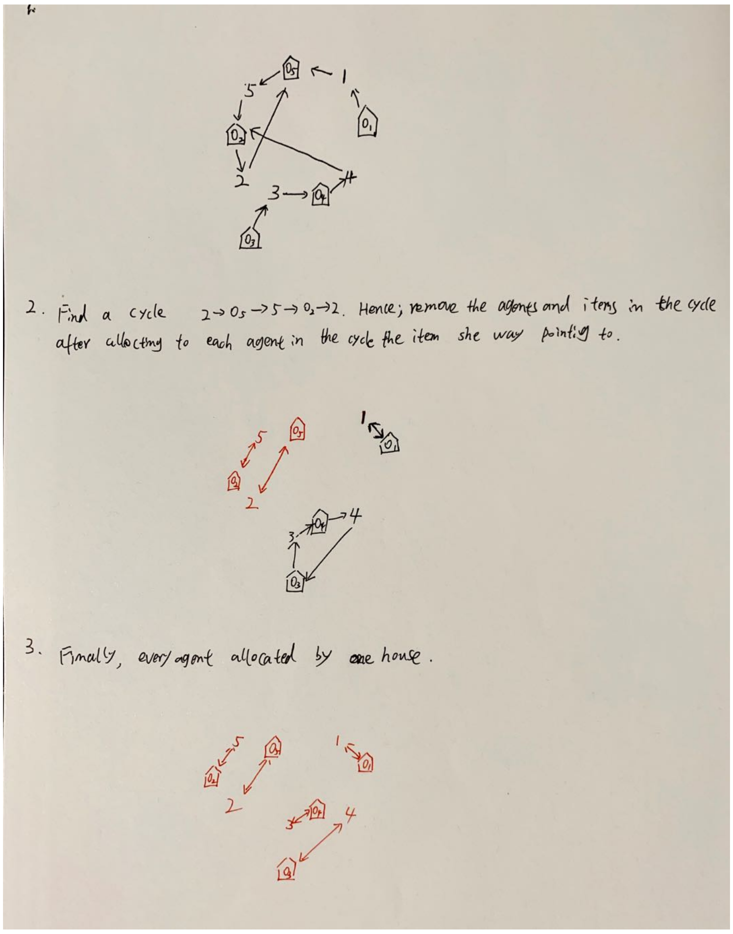
(1).

Because the set of agents $N = \{1,2,3,4,5\}$, a set of items $O = \{o1,o2,o3,o4,o5\}$, and endowment function $\omega$ : such that $\omega(i) = \{oi\}$.

The preference table as below:

| Agent / Preference | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| | O5 | O5 | O4 | O2 | O2 |
| | O2 | O4 | O2 | O1 | O4 |
| | O1 | O3 | O3 | O5 | O1 |
| | O3 | O1 | O5 | O3 | O5 |
| | O4 | O2 | O1 | O4 | O3 |

And we can use the TTC (top trading cycles) algorithm find the outcome which is **1↔O1, 2↔O5, 3↔O4, 4↔O3, 5↔O2**. Just as the picture shows at below:



(2).

**Agent 4 cannot misreport her preference to get a more preferred allocation.**

The main reason is that because the O4 is not appeared in the most preference of all other agents. This means that if just agent 4 change her preference, then agent 4 will also cannot in a cycle especially in the cycle of agent 2 and 5. Hence, even she change her preference, agent 4 cannot get the O2 and O5 which she more preferred. Because agent 2 and 5 in a cycle and agent 2 choose the house before agent 4. In addition, for the O1, because after the cycle of agent 2 and 5 are removed, then because agent 1 choose the house before agent 4 and agent 1 will choose the O1 after O2 and O5 removed. Therefore, when agent 4 choose the house, she just can choose the O3 which is similar with the original.

Overall, agent 4 cannot misreport her preference to get a more preferred allocation.

## Question 4:

**(1).**

Due to the outcome is from the algorithm which consider the student proposing deferred acceptance.

At first: {{1, e}, {2, b}, {3, a}, {4, a}, {5, d}}. And because student 3 and 4 have the same priority which is school a. Hence, we consider the school a proposing which is that the school a more prefer student 4 than 3. So, student 3 has to change to the second preference which is school b.

Secondly, {{1, e}, {2, b}, {3, b}, {4, a}, {5, d}}. And the same reason with before, student 2 and 3 both prefer school b. Then, we consider the school b and change the results which are at below:

Thirdly, {{1, e}, {2, a}, {3, b}, {4, a}, {5, d}}. And repeat again.

Fourthly, {{1, e}, {2, a}, {3, b}, {4, b}, {5, d}}.

Finally, {{1, e}, {2, a}, {3, b}, {4, c}, {5, d}}.

Because every student has a school and every school has a student and they are all differently.

**Hence, the final results are {{1, e}, {2, a}, {3, b}, {4, c}, {5, d}}.**

And the table shows at below:

**The outcome is {{1, e}, {2, a}, {3, b}, {4, c}, {5, d}}.**

| Student / School | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| | e | b | a | a | d |
| | e | b | b | a | d |
| | e | a | b | a | d |
| | e | a | b | b | d |
| | e | a | b | c | d |

**(2).**

**It is not the Pareto optimal for the students.** According to the 10 lecture notes which is that the student Proposing DA is strategyproof. The resultant allocation Pareto dominates

(wrt to students) all allocations that satisfy justified envy-freeness. This means that the results of student Proposing DA satisfy justified envy-freeness which is not fully envy-freeness and there may exist a better allocation just for the student.
In this case, there may be an allocation that can be better for students 2 and 3. The allocation is that **{{1, e}, {2, b}, {3, a}, {4, c}, {5, d}}.** Students 1, 4 and 5 are the same as before, however, students 2 and 3 are more satisfied than before. And there are not the conflicts among the students and schools.
Hence, for the students, the original results are not the Pareto optimal.

## Question 5:

The design code at below as the picture shows:

```
Input: n agents , m items, and u_i(o_j) ≥ 0 for each i∈[n] and j∈O
output: EF1 allocation X
1. Initialize allocation X = (X(1) ··· X(n)) with X(i)=φ , Incoming(i)=0, mark(a_i)(a_j)=0,
   a_i is the i's allocation  a_j is j's allocation
2. if the number of n ≥ m then
      for i=1 to n do
         X(i)=0;
      end for.
   else : n (n>n)
      for j=1 to m do
         for i=1 to n do
            if Incoming(i)=0 then
               end = i
               X(i)=0;
               break
            end if
         end for
         for h=1 to n do
            if h envy end then
               mark(a_h)(a_end) = 1
               Incoming(end)=1
            end if
         end for
         for h=1 to n do
            if end envy h then
               Incoming(h)=1
               mark(a_end)(a_h) = 1
            end if
         end for
         for h=1 to n do
            if mark(a_end)(a_h)=1 and mark(a_h)(a_end)=1, then
               exchange the allocation a_end and a_h, which means h get end's allocation
               and end get h's allocation.
               a_end = a_h , a_h = a_end.
               mark(a_end)(a_h)=0, mark(a_h)(a_end)=0.
            end if
         for k=1 to n do
            if mark(a_h)(a_end)=1 then
               mark(a_k)(a_h)=1
            elif mark(a_k)(a_h)=1 then
               mark(a_h)(a_end)=1
            end if.
         end for
      end for
   end if
   return X
```

**(1).**

We can see from the picture that the worst complexity of the algorithm is that for j=1 to m times (for i = 1 to n plus 3*(for h =1 to n) plus for k =1 to n) and plus some if sentences. Hence, the worst complexity of the algorithm is (5mn+a), a is constant, which means that this algorithm will take time the O(nm).

**(2).**

we can see from the algorithm, at the beginning, we initialize some arguments and allocation. then we estimate the n and m, if $n \geq m$, which means that we can allocate the m items for each agent at most 1 items. and after allocated, there may some agents get the items and some agents get nothing, but no agent get more than one items. Hence, It satisfy the EF1, because if any agent remove her items, then she will get nothing. and other agent will not envy she.

And if $n < m$, we can allocate every items one by one,

Firstly, we need find no incoming arcs which means that the agent(i) get the allocation which is others not envy. And we allocate the new item to this agent(i). So, after this step, we know even though we allocate the new item to i, and some others envy she, However, if we remove the new items from X(i), then no one envy i. Hence, it satisfy EF1.

Secondly, we mark the allocation which two agent one prefer another one's allocation, and she want change the allocation, such as mark(i)(j)=1, one agent i want to use her allocation exchange other agent's allocation j. This ensure that if mark(i)(j)=1 and mar(j)(i)=1, then they can exchange they preferred allocation. And when every one agent after exchanged, they whether satisfy the allocation or just envy one more items from other agents who they envy.

For example:

agent: 1 2 3 4 5 use $a_i$ denote i's allocation

$O_1$

after compare, then mark($a_2$)($a_1$)=1 mark($a_3$)($a_1$)=1.. mark($a_5$)($a_1$)=1

1 2 3 4 5.

$O_1$ $O_2$.

if there 2 prefer $O_1$ then, we do not change. because.

mark($a_1$)($a_2$) = 0, and also add mark($a_3$, $a_2$)=1 mar($a_4$, $a_2$)=1 ..

1 2 3 4 5

$O_1$ $O_2$ $O_3$

if 1 prefer $O_3$, 2 prefer $O_1$, 3 prefer $O_2$ then because we have done compare. that mark($a_1$)($a_3$) = 1, and mark($a_3$)($a_1$) = 1 then we change them and reset mark($a_1$)($a_3$)=0, mark($a_3$)($a_1$)=0. and $a_3=a_1$, $a_1=a_3$.

1 2 3 4 5

$O_3$ $O_2$ $O_1$.

However, mark($a_2$)($a_3$)=1, mark($a_3$)($a_2$) =1, which means that $a_2$ is also prefer $a_3$ and $a_3$ change to $a_1$ which is $O_1$, So this is can remain the relation between agents and items. So, chang $a_2$ and $a_3$ which is before $O_2$ and $O_1$. Then,

1 2 3 4 5

$O_3$ $O_1$ $O_2$

and $a_2=a_3$, $a_3=a_2$ which is $a_3=O_2$, $a_2=O_1$

overall, this algorithm is like the Lipton et all (2004) that find cycle, but use more short time to find cycle, and one by one changed. After, all items allocated and checked, Finally, It will be the EF1 just like Lipton algorithm.

**(3).**

In my opinion, the advantages of algorithm of Lipton et al. better than the new algorithm are that when the number of agents is not so large, then the algorithm of Lipton et al may more efficiency, because it not need like use iteration to remove the cycle, it just removes the all cycle by once while operate. And in addition, the allocation of algorithm of Lipton et al. may better than the new algorithm, because the new algorithm just guarantees the EF1 allocation, but it not ensures the allocation is balance or suitable. And the algorithm of Lipton et al will check the whole graph when the new items allocated. And update the data. Hence, for the quantity of allocation for each agent, the algorithm of Lipton et al. may better than new algorithm.