Student Selection Scientific Algorithm

Method 1

Let ***G*** be the GPA for each student, ***E*** be the extra activity mark cumulative, ***A*** be the student interested area and ***O*** be the organization

So we can define S where

$$S = \sum_{m}^{j=1} \sum_{n}^{i=1} G_i, E_i, (A, O)_j$$

Here ***n*** is the number of students and ***m*** is number of organizations combined with interested areas.

Given the constrains

$|A_i| \leq 3$

$\frac{G_i.W_G + E_i.W_E}{2} \leq 4$ Where W_G and W_E are the weights given for the GPA and Extra activities respectively.

The Steps of the algorithm

Group students in to their interested areas and organization couple by the priority of their choice

1. Select students in to groups by the priority of the interested area and organization combination.

   Priority(1) => IntrestedArea(Software Engineering) , Organization (WSO2)

Id	$P = \frac{G.W_G + E.W_E}{2}$
ICT/2008/09/012	3.2
ICT/2008/09/052	3.9
…..	…..

   Priority(2) => IntrestedArea(Software Engineering) , Organization (WSO2)

Id	$P = \frac{G.W_G + E.W_E}{2}$
ICT/2008/09/025	3.4
ICT/2008/09/036	2.8
…..	…..

   Priority(3) => IntrestedArea(Software Engineering) , Organization (WSO2)

Id	$P = \frac{G.W_G + E.W_E}{2}$
ICT/2008/09/025	2.1
ICT/2008/09/036	2.7
…..	…..

2. For every table, sort the students according to ***P***
   Example

```
for I = 1 to N-1
        J = I
        do while (J > 0) and (P(J) < P(J - 1)
          Temp = P(J)
          P(J) = P (J - 1)
          P (J - 1) = Temp
          J = J - 1
        end_do
end_for
```

Id	Sort_$ASC\left(P = \frac{G.W_G+E.W_E}{2}\right)$
ICT/2008/09/052	3.9
ICT/2008/09/012	3.2
…..	…..

3. Fill the opportunities granted from organizations combining the interested area with these three tables starting from Priority(1) table of the group.

```
do while (opportunity(Software Engineering + WSO2)>0)
       student.selected = true
       opportunity(Software Engineering + WSO2) = opportunity(Software Engineering + WSO2) – 1
end do
```

4. Iterate this procedure for every priority group.

Problems Encountered

- **More concentration on students' priority then the total mark obtained.**
  Because first we consider who has selected the particular interested area and org pair, then sort according to the marks obtained and fill the slots available.
- **Problem here is there may be students who has obtained high marks and not selected to a company of his 2nd priority while low mark students may select due to their 1st priority.**

Method 2

Sort the students according to the mark(T value) they have obtained then ,consider respectively 1^{st} priority ,2^{nd} and 3^{rd} .

Try to give them the 1^{st} else 2^{nd} or 3^{rd} . This option is most suitable because consider students' abilities first then their interest. There may be students who have not selected to any while having high marks because all their choices are already filled(may be some available opportunities which they have not requested , therefore give them a chance to update their choices) .

The Steps of the algorithm

1. Select all students in to a groups by the priority of the interested area and organization combination.

   Example.

   Priority(1) => IntrestedArea(Software Engineering) , Organization (WSO2)

Id	Priority1		Priority2		Priority3	
	Area	Org	Area	Org	Area	Org
ICT/2008/09/012						
ICT/2008/09/052						
…..						

2. sort the students according to ***P***

   Example

```
for I = 1 to N-1
        J = I
        do while (J > 0) and (P(J) > P(J - 1)
          Temp = P(J-1)
          P(J) = P (J)
          P (J - 1) = Temp
          J = J - 1
        end_do
end_for
```

Id	Sort_$ASC\left(P=\frac{G.W_G+E.W_E}{2}\right)$
ICT/2008/09/052	3.9
ICT/2008/09/012	3.2
…..	…..

3. Fill the opportunities granted from organizations combining the interested area .

```
do while (opportunity(Software Engineering + WSO2)>0)
       student.selected = true
       opportunity(Software Engineering + WSO2) = opportunity(Software Engineering + WSO2) – 1
end do

A→area
O→organization


  for I = 1 to N-1
          J = 3
          do while (J > 0)
          if(opportunity(Aij,Oij)>0)
          opportunity(Aij,Oij)-=1
          student.selected=true
          J=0
          if(student.selected=false)
          //send student to not selected pool
          end_do
  end_for
```

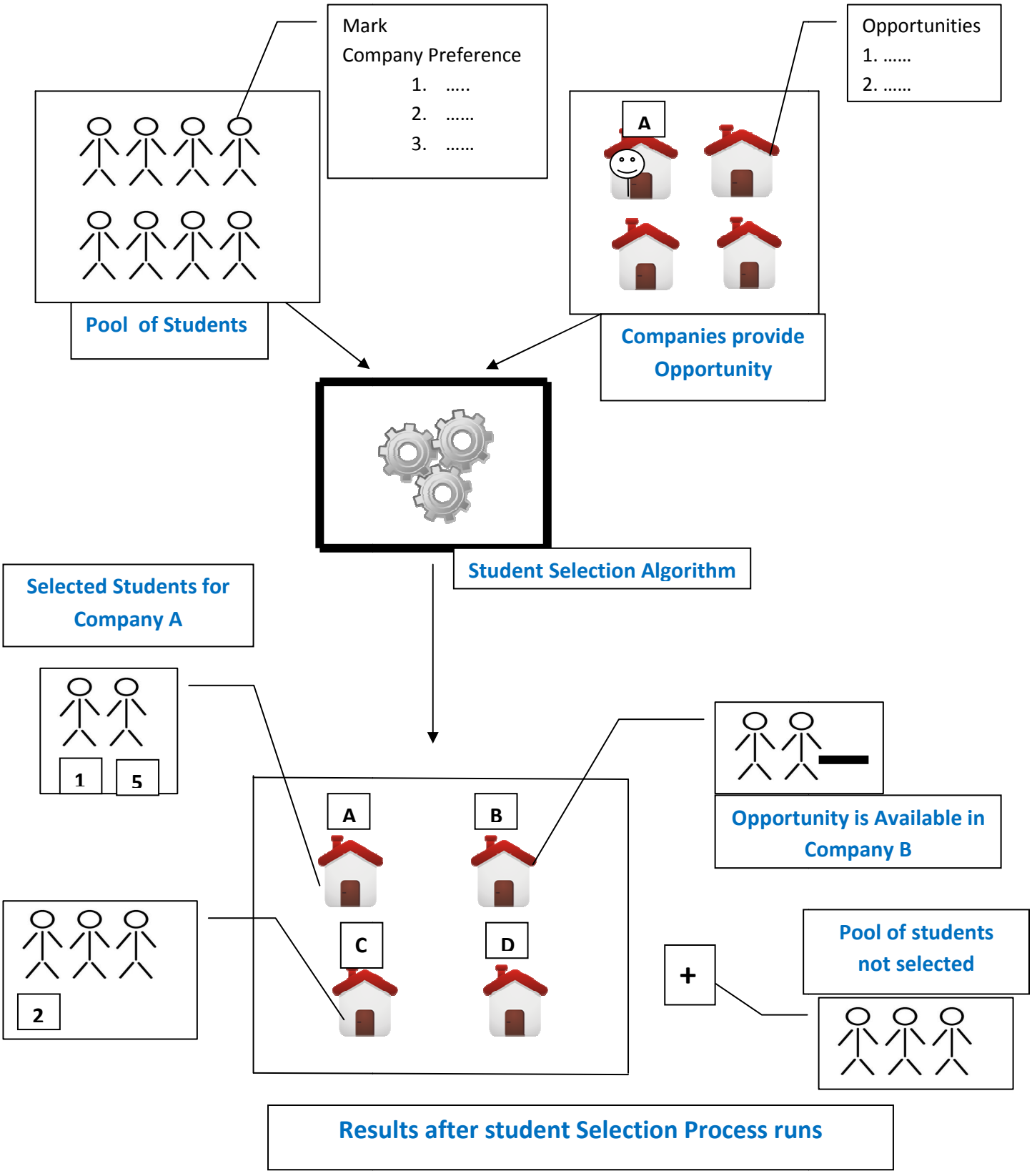
Algorithm Optimization

To select a maximum amount of students and the most appropriate students, Algorithm can be optimized in the context of W_G and W_E as below

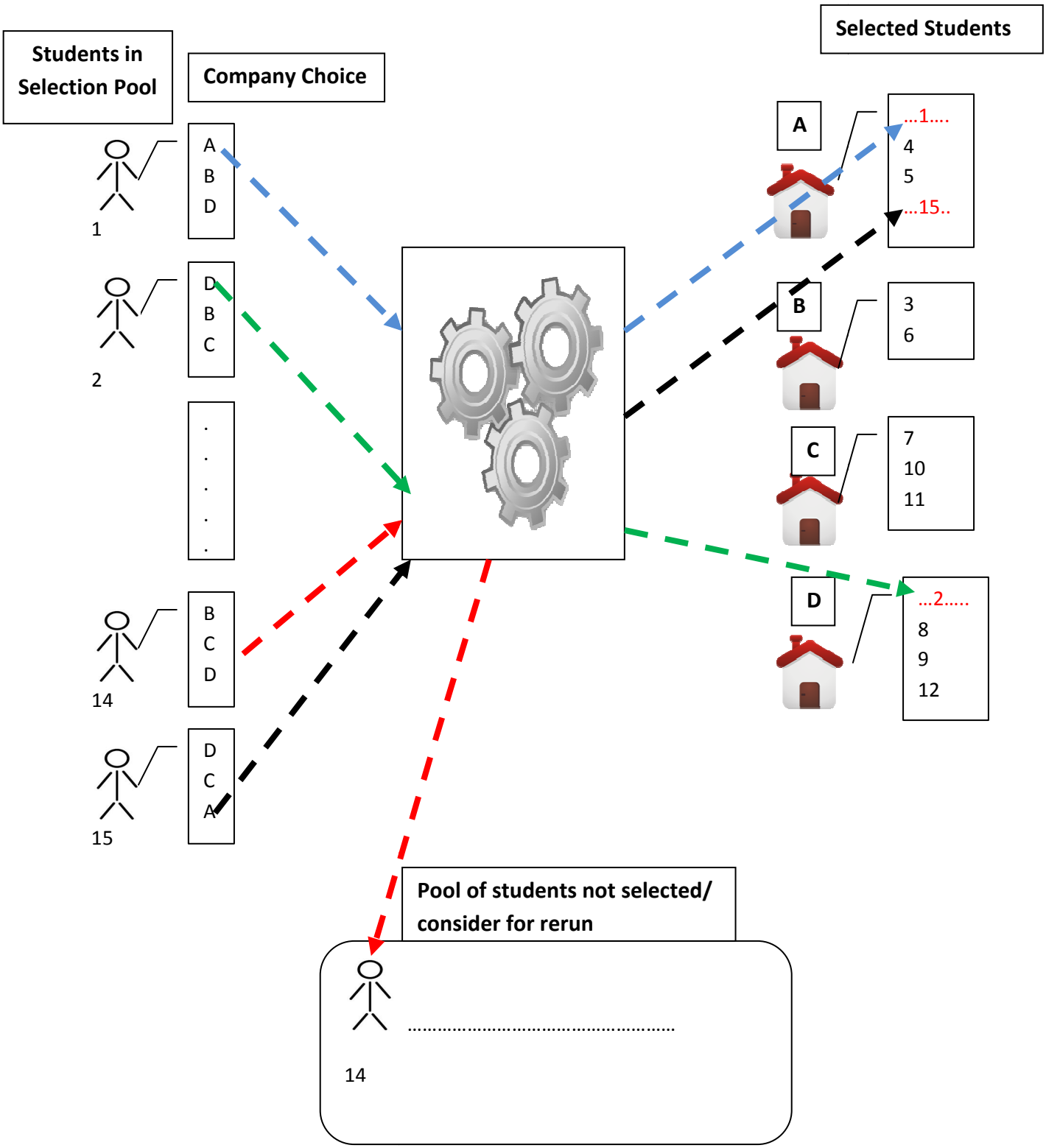
W_G	W_E	S_i
trivial	*trivial*	-
trivial	*trivial*	-
….	….	….
0.6	0.4	S_x
0.7	0.3	S_y
0.8	0.2	S_z
….	….	….
trivial	*trivial*	-
trivial	*trivial*	-

Where S is final selection outcome of the algorithm, $W_G + W_E = 1$ and
Out of S_i the maximum would be

$$\sum S_{max} = W_{G(max)} + W_{E(max)}$$



- Student may not select, No enough opportunities
  Companies they have selected are already filled
- Some companies may have request a particular student, then he is removed from the pool and opportunities are reduce by 1(Company A have requested no 1 student)



Student selection method