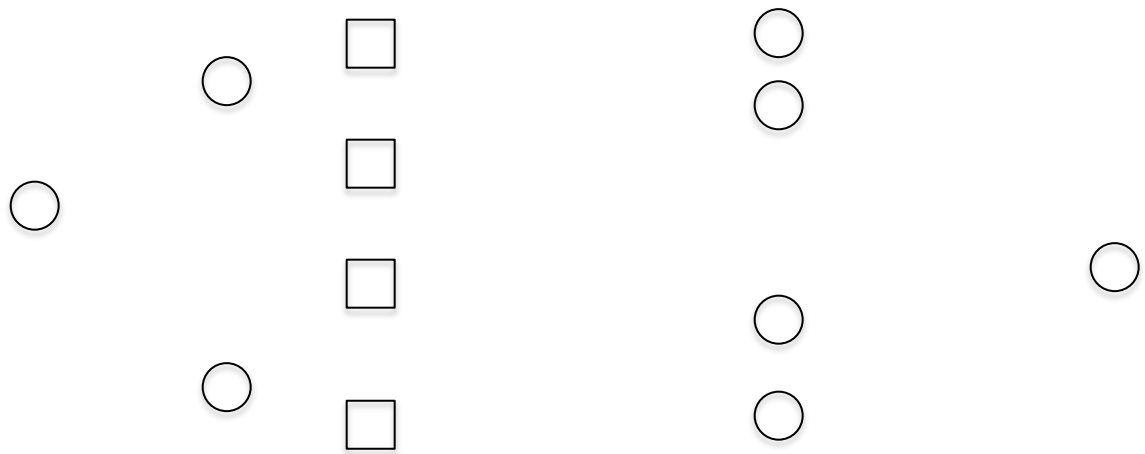
Francisco Rocha
CMPS 102
3/12/15

Homework #4 (Network Flow)

1. We can observe that given a list of both a child's preference of party favors and snacks both of size n (S, F), we can create a flow graph in which we draw edges between the n children to their respective choices in order to determine whether we can do this sort of matching. Furthermore let us construct the graph in the following way:
   Draw arcs each of capacity 2 from super source S to the nodes of the n children, then from those nodes each gets an arc of capacity 1 to two "gadget" nodes which will connect to the set lists of desirable snacks and party favors that the children's parents submitted, then we draw arcs to a super sink T with flow of 1 since only one choice is allowed for any given snack or favor (S_i, F_i).
   Pictorially the graph described will be as follows:



   As you can see we are interested in finding the MaxFlow of the given graph and more importantly want to see if the flow coming our of the source S is equal to the flow going into our sink T, $ie\ a\ MaxFlow\ of\ 2n$. This will account for each of the n children getting matched with their respective preference choice of snacks and party favors, which are given to each be of size n as well. Moreover if the MaxFlow does not satisfy the given condition above $(ie. flow\ out\ of\ S \neq flow\ into\ T)$ we are assured that a matching is not possible.

   Algorithm and Runtime:

   Algorithm is as follows:

Build graph G as described above, Run Ford-Fulkerson to find maximal $S - T$ flow If flow of $2n$ exists return true else return false

We can observe that building the graph G would account for a runtime of $O(n^2)$, running FF would account for a runtime of $O(mF)$ where m is the number of edges in graph and F is the MaxFlow therefore our total runtime will be $O(n^2 \bullet 2n) = O(n^3)$.

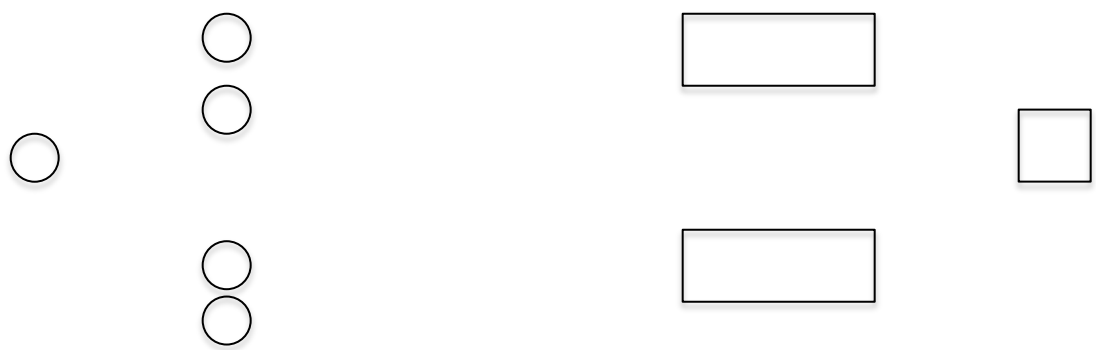
<u>Proof of Correctness:</u>
We will prove that our graphical representation of the given problem yields same result as solving the given problem. In other words, we will show that if there exits such an assignment of children and snacks and party favors in A (where A is the original problem), then a MaxFlow of $2n$ exits in our graphical representation G. We also need to prove that if there is a MaxFlow of $2n$ then there exists a valid assignment in A:

$i. (Yes\ in\ A => Yes\ in\ G)$
Claim: If a valid assignment in A exists, now we shall prove that it also exists in G.
<u>Proof:</u>
For each pair of (C_i, S_i) and (C_i, F_i) that classifies a suitable choice of snack and favor for any given child in A, saturate the arc $C_i \rightarrow G_1\ and\ C_i \rightarrow G_2$ then we saturate the arc $G_1 \rightarrow S_i$ and $G_2 \rightarrow F_i$ in G. Also saturate all the arcs going out of S and all the arcs going into T. Observe that we respect capacity of each arc since we are only saturating each arc by flow of 2 from super source to children since they have two choices $(one\ snack\ one\ party\ favor)$ which we then restrict using the "gadget" nodes so that each kid gets exactly one snack and one favor and by pushing flow of one from gadget nodes to every other node in G we also respect capacity.

$ii. (Yes\ in\ G => Yes\ in\ A)$
Claim: If there is a MaxFlow of n in G, then we shall prove that there exists a valid assignment in A.
<u>Proof:</u>
We observe that by the integrality theorem, an integral flow of $2n$ exists therefor we must prove that each child will get matched with one snack and one favor. This is shown with our gadget nodes, which add arcs to the list of snacks they want and to the list of favors they want. Each gadget has a leaving edge of capacity one and will receive flow of one so by our integrality theorem they each get exactly one choice. Furthermore if an $G_1 \rightarrow S_i$ and $G_2 \rightarrow F_i$ matching's exist then we can see that an equivalent assignment in A is possible since flow of exactly 2 is being pushed to any given C_i and flow of exactly 1 is being pushed from any given $G_1 \rightarrow S_i$ and $G_2 \rightarrow F_i$ pairing to T which will give us a overall flow of $2n$ if matching is feasible.

2. The problem states that we are to be given a matching of all the n childrens' s_i unique working spots and we must assign a matching to each cabin such that all the

kids attending camp are evenly spread within the cabins
$(ie\ each\ cabin\ has\ \left\lceil \frac{n}{c} \right\rceil\ kids\ in\ it)$.
We shall construct the graph in the following way:
Draw arcs from super source S to each of the s_i working spots with edge capacity of 1 indicating that only one child should be assigned to each working spot. Then using our given function we made edges from working spots s_i to every cabin that it determines not to be "too far" with a capacity of ∞ since we have accounted for the upper bound in the previous edge, in other words we cannot send more than flow of 1 through this edge at any point because s_i node will only have a flow of 1 going though it at any given point. Furthermore let us now connect the cabin nodes to a sink T where the edge capacity is $\left\lceil \frac{n}{c} \right\rceil$ in order to assure that the most children that any cabin will hold is an even partition of the kids at the camp.



<u>Algorithm and Runtime:</u>

Algorithm is as follows:

Build graph G as described above, Run Ford-Fulkerson to find maximal $S - T$ flow If flow of n exists return true else return false

We can observe that building the graph G would account for a runtime of $O(n \bullet c)$ since in a worst case every child is close enough to every cabin, running FF would account for a runtime of $O(mF)$ where m is the number of edges in graph and F is the MaxFlow therefore our total runtime will be $O(nc \bullet n) = O(cn^2)$.

<u>Proof of Correctness:</u>
We again must show that our graph G has a solution then its solution is equivalent to A and also if a solution of A exists then we must also be able to find an equivalent solution in G.

$i. Yes\ in\ A => Yes\ in\ G$
<u>Proof:</u>
Claim: If a valid assignment in A exists then there is a valid flow of n in G.

Since a valid assignment exists, then for each pairing of appropriate childrens' work spot s_i to cabin, we can push flow of 1 from $s_i \to c_j$. Then saturate $c_j \to T$ edge fully, followed by also saturating $S \to s_i$, as required. Then we observe that $MaxFlow\ v(F) = n$ since all the edge going into T will be at most $\left\lceil \frac{n}{c} \right\rceil$ therefore the sum of the flow of all the edges will be n. Flow will be conserved since by definition flow of 1 is sent from $S \to s_i$ allowing for only flow of one to leave from $s_i \to c_j$ which will decide what cabin they are assigned too based on weather work spot is close enough to any given cabin, therefore each c_j will have at most $\left\lceil \frac{n}{c} \right\rceil$ flow coming out and in since it was a valid assignment in A. Therefore flow is conserved and capacity is respected.

$ii. Yes\ in\ G => Yes\ in\ A$

<u>Proof:</u>

Claim: If a flow of n exists then we can find a valid assignment of children and cabins in A. Since a valid flow of n exists then we can see that is a saturated edge from $s_i \to c_j$ has flow then student s_i is assigned to cabin c_j. We must prove that children only get assigned to cabins that are close enough to their working spot:

- This follows from the fact that when constructing the graph we only draw a $s_i \to c_j$ edge if a child's working spot is close enough to a given cabin, therefore flow can be only from children to an acceptable cabin

We must also prove that each cabin has equal amount of kids.

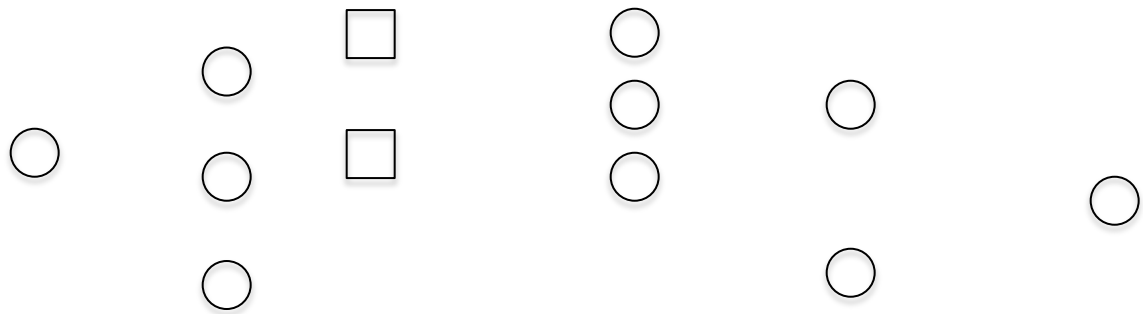
- This follows from the fact that the $MaxFlow\ v(F)$ was n since each cabin can have a total of $\left\lceil \frac{n}{c} \right\rceil$ flow coming out by conservation of flow.
- Therefore each cabin node must have had at most $\left\lceil \frac{n}{c} \right\rceil$ flow coming in and $\left\lceil \frac{n}{c} \right\rceil$ coming out since each day has at least $\left\lceil \frac{n}{c} \right\rceil$ individual edges coming in from s_i nodes and $\left\lceil \frac{n}{c} \right\rceil$ kids were assigned to cabin c_i, which was needed, by definition of valid assignment.

3. In this problem we wish to help with the assignment of collaborative design projects for Paris Fashion Week where we have the following restrictions:
   - Each design committee must consist of exactly k designers
   - Each design committee cannot consist of only people from the same design school $(P, M, and\ F)$
   - There are n articles of clothing to be designed, each by one committee
   - There are m designers, who can each work on up to 3 articles of clothing.
   - You know the set C_i of articles each designer i has the necessary skills to work on.

   Knowing this we construct the graph as follows:

   We start at super source S and draw edges from it to each of the m designers with each having a capacity of 3 since they can work on 3 articles of clothing. After that we draw a edge from designer nodes to each articles of clothing from C_i which they can work on with capacity 1. After we draw an edge of capacity ∞ from each

article of clothing to each of the three schools and we will push flow in regards to how many people from each school are able to work on said article of clothing, then from the school nodes we will draw and edge of capacity $k - 1$, to account for the restriction that each item must not consists of a committee of entirely one school, to the committee nodes. From there we connect each committee node to super sink T and if there is a flow of kn then solution is possible.
Pictorially it would look as follows:



<u>Algorithm and Runtime</u>

Algorithm is as follows:

Build graph G as described above, Run Ford-Fulkerson to find maximal $S - T$ flow If flow of kn exists return true else return false

We can observe that building the graph G would account for a runtime of $O(m + m + mn + n)$, which simplifies to $O(mn)$, since in a worst case every designer has necessary skills for every article of clothing, running FF would account for a runtime of $O(mF)$ where m is the number of edges in graph and F is the MaxFlow therefore our total runtime will be $O(mn \bullet kn) = O(mkn^2)$.

<u>Proof of Correctness:</u>

We again must show that our graph G has a solution then its solution is equivalent to A and also if a solution of A exists then we must also be able to find an equivalent solution in G.

$i. Yes\ in\ A => Yes\ in\ G$

<u>Proof:</u>

Claim: If a valid assignment of designers and articles of clothing exists in A then there is a flow of kn in G.

Since there is a valid grouping of committees given the restrictions we can push flow of 1 to each ($of\ up\ to$ 3) article of clothing that each designer is working on as needed ($ie\ up\ to$ 3 $from\ S \rightarrow d_i \rightarrow s_k$). Then we will push flow as needed from school node to committee node($s_k \rightarrow c_j$), depending on how many people from each school are working on the said article of clothing following this we see that there is a flow of k from committee to sink T($c_j \rightarrow T$), classifying it as an appropriate grouping. Then the $MaxFlow\ v(F)$ will be equal to kn since all the edge from committee to T will be equal to k. Flow will be conserved since by definition we only draw nodes from students to articles of clothing and push flow if they can work and are working on them, then we calculate how many people from each school are working on any given project and push flow to the given committee with respect to how much was pushed in and is capable of being pushed out ($ie\ at\ most\ k - 1\ can\ go\ from\ school\ to\ committee\ and\ committee$ $and\ committee\ will\ have\ k\ flow\ going\ in\ from\ different\ school\ nodes, and\ k$ $coming\ out\ since\ we\ push\ flow\ as\ needed\ from\ clothing\ to\ school\ nodes$). We also observe that capacity is respected since we do not push more out of any designer node will have at most flow coming in and 3 flow going out ($ie\ no\ designer\ will\ be\ assigned\ to\ more\ jobs\ than\ he\ can\ work\ on$) to account for them being able to work on at most 3 items and we will push that flow as needed. ($ie\ they\ can\ possible\ work\ on\ less\ articles\ of\ clothing$).

$ii. Yes\ in\ G => Yes\ in\ A$

<u>Proof:</u>

Claim: If a flow of kn exists in G, then a valid assignment of designers to committees exits in A.

Since a valid flow of kn exists in G then we can see that saturated edges from designers, to respective committees will assign designer d_i to committee c_j. Now we must show that this is a <u>valid</u> assignment. So fist let us show that designers only get assigned to up to 3 items, therefore are only on 3 committees.

- The flow from $S \rightarrow d_i$ is at most 3 and we push as much as needed from ($d_i \rightarrow s_k$). Since we only draw edges from ($s_k \rightarrow c_j$) if a designer from any said school has the skills necessary to be on said committee therefor flow can only go from designer to acceptable committee.

Also we need to show that each committee not only has k designers but that every committee is made up of designers from not entirely one school.

- This is true since the $MaxFlow\ v(F)$ is equal to kn since each committee to sink edge ($c_j \rightarrow T$) is fully saturated and incoming edges ($s_k \rightarrow c_j$) have capacity $k - 1$ to ensure that our multi-school restriction is satisfied. Therefore each

incoming flow is k and from at least two different schools and flow coming out is k by conservation of flow. Therefore k designers from at least two different schools will be working on one article of clothing.

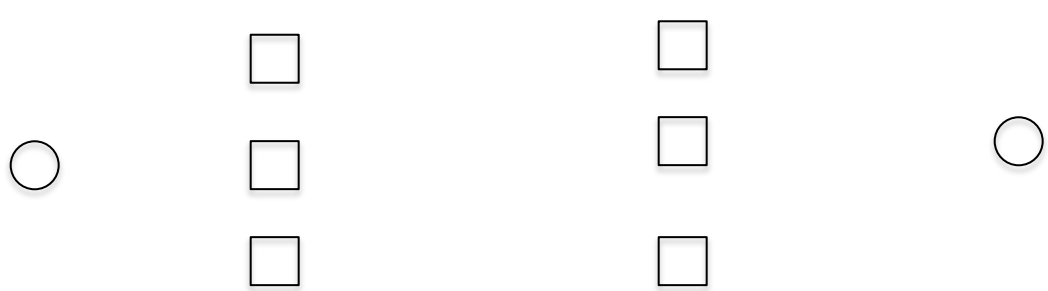
4. Given the specifications of the problem at hand, we must first show a $n \times n$ matrix that is not $rearrangeable$ (a) which is a matrix whose diagonal can not be made to all 1's by swapping some pairs of rows and some pairs of columns.
(a)

1	0	1	0
1	0	0	0
1	0	0	0
1	1	1	1

Notice that regardless how we move around the rows and columns we cannot make the diagonal consist of only 1's.

(b) now we wish to give a polynomial time flow algorithm in which we decide if a given $n \times n$ matrix is rearrangeable .
The formulation of the graph will be as follows:
Let each of the rows be a set of nodes and each of the columns be a set of nodes. We will draw and edge from row r_i to column c_j nodes if said entry $e_{i,j} = 1$ and each capacity of these edges $r_i \to c_j$ will be 1. Furthermore each row node is connected to a super source S and each column node is connected to super sink T, with each of these edges having flow of one. If $MaxFlow$ $v(f)$ of n exists then matrix is rearrangeable.

Pictorially, graph looks as follows:



<u>Algorithm and Runtime</u>

Algorithm is as follows:

Build graph G as described above, Run Ford-Fulkerson to find maximal $S - T$ flow If flow of n exists return true else return false

We can observe that building the graph G would account for a runtime of $O(n^2)$, since in a worst case every entry will have a 1, running FF would account for a runtime of $O(mF)$ where m is the number of edges in graph and F is the MaxFlow therefore our total runtime will be $O(n^2 \bullet n) = O(n^3)$.

<u>Proof of Correctness:</u>
We again must show that our graph G has a solution then its solution is equivalent to A and also if a solution of A exists then we must also be able to find an equivalent solution in G.

$i. Yes\ in\ A => Yes\ in\ G$
<u>Proof:</u>
Claim: If matrix M is rearrangeable, then there exists a MaxFlow of n in G.
Since M is rearrangeable then we can observe that there are n positions of 1's such that two such entries $e_{i,j}$ and $e_{l,k}$ satisfy that $i \neq l$ and $j \neq k$. We run flow from $r_i \rightarrow c_j$ if said $e_{i,j}$ entry contains a 1 and we saturate all edges leaving S and all edges going into T. This will account for a flow that is n since n amount of flow will be pushed out of S and there will be at least n number of edges such that $e_{i,j}$ and $e_{l,k}$ where $i \neq l$ and $j \neq k$, and the total number of edges going into T will be $|C| = n$ therefore $MaxFlow\ v(f) = n$. Furthermore capacity and conservation is respected since each edge has a capacity of 1 so at must they can revive and give 1 flow and we have an integral flow and all edges going into T are saturated.

$ii. Yes\ in\ G => Yes\ in\ A$
<u>Proof:</u>
Claim: If a MaxFlow of n exists in G, then matrix M is rearrangeable. Since there is a flow of n in G then we know that there is an edge fully saturated between $r_i \rightarrow c_j$ therefore any other nodes $r_i \rightarrow c_k (where\ k \neq j)$ cannot be saturated by the integrality theorem and the capacity of all edges in this constructed graph is 1 and in a valid flow conservation and capacity must be conserved by definition and construction of this graph G. Therefore the flow value of n implied that there are n entries of value 1 in matrix M such that two such entries $e_{i,j}$ and $e_{l,k}$ can be rearranged and $i \neq l$ and $j \neq k$ therefore M is rearrangeable since any two 1's in the same row or column cannot be rearranged to other rows or columns by any number or exchange operations unless there are n number of ones that satisfy the identity given above.

5. We wish to find a fair carpool schedule between a group of friends in which no one gets overloaded with too much driving. We shall define fairness to be the obligation Δ_j of the jth person where Δ_j isn't necessarily an integer:

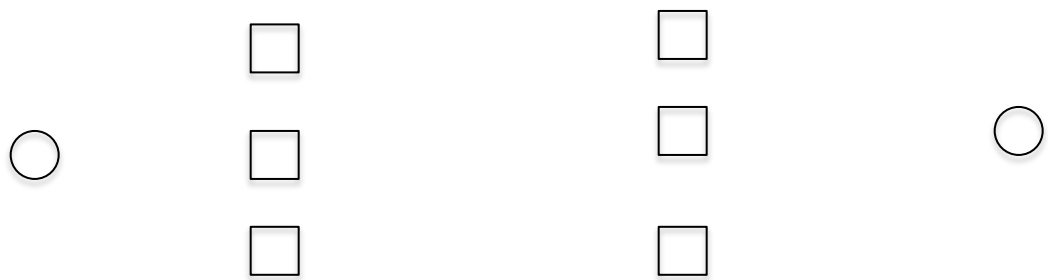
$$\Delta_j = \sum_{i:p_j \in S_i} \frac{1}{|S_i|}$$

Where $S = \{p_1, \dots, p_k\}$, and is the set of all of the people who are part of this carpool schedule. Also let S_i is the subset of the people who go to work on the ith day. We

want to show that a fair driving schedule exists for sets $S_1, \dots, S_d$, let us do this by constructing a graph in the following way:

We will list all of the people as nodes p_j for the jth diver and draw an edge to the d_i if they can drive on that i^{th} day with a capacity of 1. We will connect all the p nodes to a super source where each arc going from S to p_i will have capacity $\lceil \Delta_j \rceil$ denoting the most days that they are able to drive. Furthermore we will then connect all the day nodes to a super sink T will a capacity 1. If G has flow of value d then we know the acceptable arrangement in A.

Pictorially, this is how the graph G will look:



<u>Algorithm and Runtime</u>

Algorithm is as follows:

Build graph G as described above, Run Ford-Fulkerson to find maximal $S - T$ flow If flow of p exists return true else return false

We can observe that building the graph G would account for a runtime of $O(p + pd + d)$, which simplifies to $O(pd)$, since in a worst case every driver will be in the car everyday, running FF would account for a runtime of $O(mF)$ where m is the number of edges in graph and F is the MaxFlow therefore our total runtime will be $O(pd \bullet d) = O(pd^2)$.

<u>Proof of Correctness:</u>

We again must show that our graph G has a solution then its solution is equivalent to A and also if a solution of A exists then we must also be able to find an equivalent solution in G.

$i. Yes\ in\ A => Yes\ in\ G$

Proof:

Claim: If there is a valid driving schedule then there is a valid flow of d in G.
Since a valid assignment exists in A then there is a fair driving schedule. Observe we push flow of 1 from the jth driver on day i if he will be driving that day ($ie. p_j \to d_i$) then observe that all of the day edges to the super sink $d_i \to T$ will be fully saturated, and the $S \to p_j$ will be filled as needed according to how much is fair for them to drive. Therefore the $MaxFlow\ v(f) = d$ and we can see that:

- **Capacity is respected** since for $S \to p_j$, they were filled as needed, and each $p_j \to d_i$ has a flow of one which will not exceed what they received and flow is an integral flow. Furthermore $d_i \to T$ has flow of one since for each day one person is driving therefore flow of 1 will be coming out therefore T will receive d amount of flow.
- **Flow is conserved** since flow from $S \to p_j$ is Δ_j, and the flow from $p_j \to d_i$ will be $\frac{1}{|S_i|}$, and the flow from $d_i \to T$ will be 1. Since $\Delta_j = \sum_{i:p_j \in S_i} \frac{1}{|S_i|}$ the flow from $p_j \to d_i$ will be 1 since $|S_i| \cdot \frac{1}{|S_i|} = 1$ and the flow at T will be equal to $\sum_i (d_i) = d$ as is needed.

$ii. Yes\ in\ G => Yes\ in\ A$

Proof:

Claim: If a flow of d exists in G, then a fair driving schedule exists. We will consider and integral $MaxFlow\ v(f) = d$ and say that driver p_j drives on day i if and only if the flow from $p_j \to d_i$ is 1. This is a fair driving schedule since the sum of all edges going from days to T is d and every $d_i \to T$ edge is saturated. Since $v(f)$ is an integral flow exactly 1 person will drive on each day d_i due to flow conservation and by how we constructed our graph no person will be assigned to drive on a day they will not be in the car. Also note that the days in which p_j drives will be equal to the flow coming in from $S \to p_j$ and leaving from $p_j \to d_i$ indicating what days he is driving, which is no greater than $\lceil \Delta_j \rceil$, again by our construction of graph. Therefore this matching from $p_j \to d_i$ is our driving schedule.