ON THE COMPLEXITY OF TIMETABLE AND MULTI-COMMODITY FLOW PROBLEMS

by

S. Even
Department of Computer Science, Technion, Haifa, Israel
On leave of absence from the Weizmann Institute of Science
Rehovot, Israel
and

A. Itai and A. Shamir
Department of Applied Mathematics, The Weizmann Institute of Science
Rehovot, Israel

Abstract

A very primitive version of Gotlieb's timetable problem is shown to be NP-complete, and therefore all the common timetable problems are NP-complete. A polynomial time algorithm, in case all teachers are binary, is shown. The theorem that a meeting function always exists if all teachers and classes have no time constraints is proved. The multi-commodity integral flow problem is shown to be NP-complete even if the number of commodities is two. This is true both in the directed and undirected cases. Finally, the two commodity real flow problem in undirected graphs is shown to be solvable in polynomial time. The time bound is $O(|V|^2|E|)$.

I. The timetable problem is NP-complete

The timetable problem (TT), which we shall discuss here, is a mathematical model of the problem of scheduling the teaching program of a school. In fact, it is a rather naive model since it ignores several factors which definitely play a role in practice [1]. However, we shall show that even a further restriction of the problem still leads to an NP-complete problem [2,3].

Definition of TT : Given the following data:

(1) A finite set H (of hours in the week);

(2) A collection $\{T_1,T_2,\ldots,T_n\}$ where $T_i \subseteq H$; (There are n teachers and T_i is the set of hours during which the i-th teacher is available for teaching.)

(3) A collection $\{C_i,C_2,\ldots,C_m\}$ where $C_j \subseteq H$; (There are m classes and C_j is the set of hours during which the j-th class is available for studying.)

(4) An $n \times m$ matrix R of non-negative integers; (R_{ij} is the number of hours which the i-th teacher is required to teach the j-th class.)

The problem is to determine whether there exists a meeting function

$f(i,j,h) : \{1,\ldots,n\} \times \{1,\ldots,m\} \times H \to \{0,1\}$

(where $f(i,j,h) = 1$ if and only if teacher i teaches class j during hour h) such that:

(a) $f(i,j,h) = 1 \Rightarrow h \in T_i \cap C_j$

(b) $\sum_{h\in H} f(i,j,h) = R_{ij}$ for all $1 \leq i \leq n$ and $1 \leq j \leq m$;

(c) $\sum_{i=1}^{n} f(i,j,h) \leq 1$ for all $1 \leq j \leq m$ and $h \in H$;

(d) $\sum_{j=1}^{m} f(i,j,h) \leq 1$ for all $1 \leq i \leq n$ and $h \in H$.

(a) assures that a meet takes place only when both the teacher and the class are available. (b) assures that the number of meets during the week between teacher i and class j is the required number R_{ij} . (c) assures that no class has more than one teacher at a time and (d) assures that no teacher is teaching two classes simultaneously.

A teacher i is called a k-teacher if $|T_i| = k$; he is called tight if

$$|T_i| = \sum_{j=1}^{m} R_{ij}$$

that is, he must teach whenever he is available.

Definition of RTT : RTT (the restricted timetable problem) is a TT problem with the following restrictions:

(1) $|H| = 3$,

(2) $C_j = H$ for all $1 \leq j \leq m$ (the classes are always available),

(3) each teacher is either a tight 2-teacher or a tight 3-teacher,

(4) $R_{ij} = 0$ or 1 for every $1 \leq i \leq n$ and $1 \leq j \leq m$.

Clearly both the TT and the RTT problem are in the NP class. We want to show that RTT is NP-complete. In that case TT is trivially NP-complete too. We recall that 3-SAT (satisfiability of a conjunctive normal form with 3 literals per clause) is NP-complete where 3-SAT is defined as follows:

Given the data

(1) a set of <u>literals</u>

$X = \{x_1, x_2, \ldots, x_\ell, \overline{x}_1, \overline{x}_2, \ldots, \overline{x}_\ell\}$,

(2) a family of <u>clauses</u> $D_1, D_2, \ldots, D_k$ such that for every $1 \leqslant j \leqslant k$ $|D_j| = 3$ and $D_j \subseteq X$,

the problem is to determine whether there exists an assignment of values "true" and "false" to the literals, such that

(a) exactly one of x_i and $\overline{x}_i$ is assigned "true" while the other is assigned "false",

(b) in each clause D_j there is at least one literal assigned "true".

<u>Theorem 1</u>: 3-SAT $\propto$ RTT

<u>Proof</u>: The proof is by displaying a polynomially bounded reduction of the 3-SAT to RTT. In our construction, certain classes play the role of occurrences of literals x_i or $\overline{x}_i$ in the clauses; the order in which some 2-teachers teach these classes indicates the truth value of the literals. All other classes and teachers are used in order to guarantee that this assignment of truth values satisfies conditions (a) and (b) above, and that all occurrences of a literal are assigned the same truth value.

Let p_i be the number of times the variable x_i appears in the clauses, i.e.

$$p_i = \sum_{j=1}^{k} |D_j \cap \{x_i, \overline{x}_i\}| \quad .$$

For each x_i we construct a set of $5 \cdot p_i$ classes which will be denoted by $C^{(i)}_{ab}$ where $1 \leqslant a \leqslant p_i$ and $1 \leqslant b \leqslant 5$ (we omit the superscript i whenever all classes used in the construction refer to the same i). In order to simplify the exposition we shall use a graphic representation of the classes and teachers (see Figure 1 for the structure corresponding to a single i). In our graphic representation the vertices denote class-hour combinations, where the rows signify the hours and the columns - the classes. The hours are h_1, h_2 and h_3 . Now a

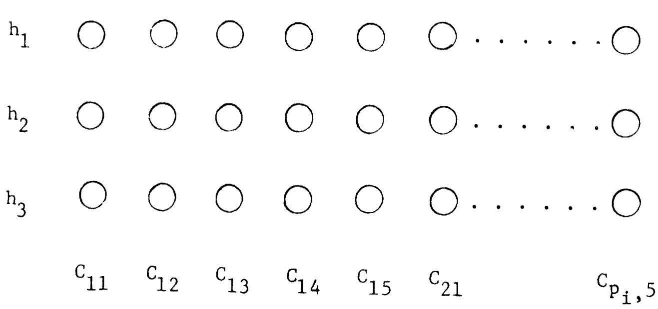

Figure 1

2-teacher who is available in hours h_1 and h_2 and is supposed to meet once with $C_{a_1b_1}$ and once with $C_{a_2b_2}$ will be represented as shown in Figure 2. The two diagonals show the only two ways possible to schedule this teacher. A 3-teacher who has to teach

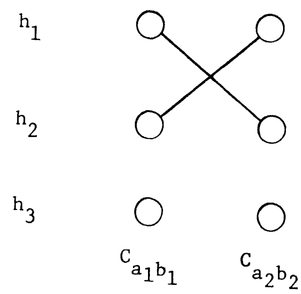

Figure 2

$C_{a_1b_1}$, $C_{a_2b_2}$ and $C_{a_3b_3}$ is denoted by a line with three arrows in the columns corresponding to these classes, as shown in Figure 3. For every $1 \leqslant q \leqslant p_i$ we add two new

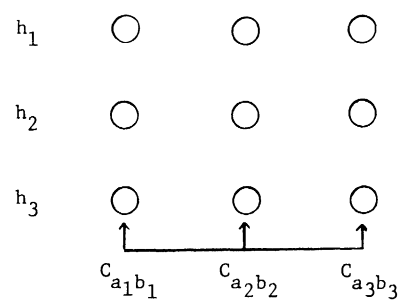

Figure 3

classes, C'_{q1} and C''_{q1} with the structure shown in Figure 4. There are three teachers described in the structure, two are 2-teachers and one 3-teacher. Since all these 3 teachers must teach during h_1 , the top 3 vertices, (h_1, C_{q1}) , (h_1, C'_{q1}) , and (h_1, C''_{q1}) must be utilized.

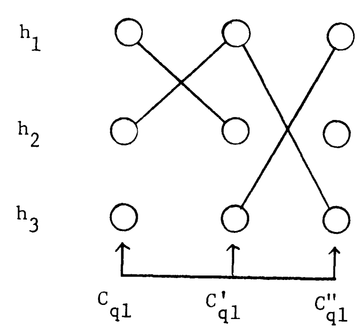

Figure 4

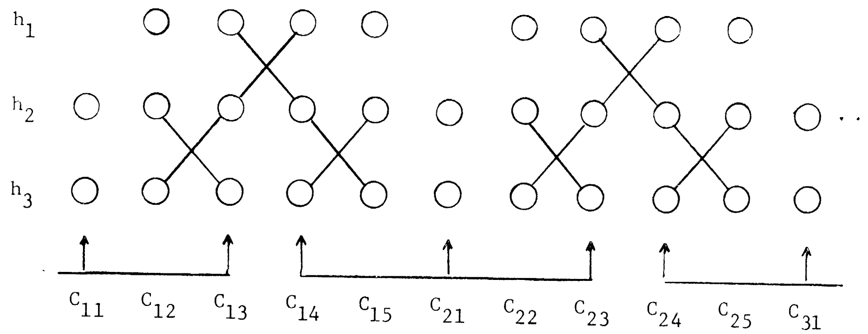

Figure 5

However, we have a choice of utilizing exactly one of the vertices (h_2, C_{q1}) and (h_3, C_{q1}), while leaving the other available; there are several ways to do this, as the reader may verify by himself. As far as the rest of our structure is concerned, the effect of this substruture is as follows: (h_1, C_{q1}) is taken and one of (h_2, C_{q1}) and (h_3, C_{q1}) is taken. Thus, we shall delete (h_1, C_{q1}) from our diagrams.

Consider now the structure of teachers described in Figure 5; it is intended to consistently assign truth values to all occurrences of x_i and $\bar{x}_i$ in the clauses. Clearly, there is a 3-teacher assigned to classes $C_{p_i 4}$, C_{11} and C_{13}; thus, the structure described in Figure 5 is circular. Consider now the p_i 2-teachers who are available during h_1 and h_2, where the q-th such teacher is assigned to classes C_{q3} and C_{q4}. We claim that all these teachers must be scheduled in the same manner; that is, either all of them teach the C_{q3} classes during h_1 and the C_{q4} classes during h_2, or all of them teach the C_{q3} classes during h_2 and C_{q4} classes during h_1. Assume we have a schedule which does not satisfy this consistency condition. Then, there must be a q such that the q-th teacher teaches C_{q3} during h_2 and C_{q4} during h_1, while the (q+1)-st teacher* teaches the $C_{(q+1),3}$ during h_1 and $C_{(q+1),4}$ during h_2. In this case the 3 teacher who must teach C_{q4}, $C_{(q+1),1}$ and $C_{(q+1),3}$ cannot be scheduled during h_1 - a contradiction.

We thus obtain, independently for each i, a uniform scheduling of all the 2-teachers who are available during h_1 and h_2. The order in which these teachers teach C_{q3} and C_{q4} in the i-th structure will be interpreted as the truth value of the variable x_i in the original 3-SAT problem.

We now add a few more 3-teachers, connecting the various i-structures, in order to guarantee that in each clause D_j, at least one literal gets the value "true". For every clause $D_j = \{\xi_1, \xi_2, \xi_3\}$ we assign a 3-teacher in the following way. He is assinged to teach one class for each of the three literals. If $\xi_1 = x_i$ and this is the q-th appearance of this variable, then the corresponding class is $C^{(i)}_{q2}$, while if $\xi_1 = \bar{x}_i$ the corresponding class is $C^{(i)}_{q5}$. The classes corresponding to ξ_2 and ξ_3 are defined analogously.

This completes the definition of the RTT problem. The total number of classes defined is $21 \cdot k$, and the total number of teachers is $22 \cdot k$ ($15 \cdot k$ 2-teachers and $7 \cdot k$ 3-teachers). We claim that the given 3-SAT problem has a positive answer if and only if the RTT problem constructed above has a positive answer.

First, assume the 3-SAT problem has a positive answer. We use, now, the values of the literals in such an assignment to display a schedule for the constructed RTT problem - to prove that its answer is positive too.

If x_i is assigned "true" then for every $1 \leqslant q \leqslant p_i$ the q-th 2-teacher is scheduled to teach $C^{(i)}_{q3}$ during h_1 and he teaches $C^{(i)}_{q4}$ during h_2. Conversely, if x_i is assigned "false" then for every $1 \leqslant q \leqslant p_i$ the q-th 2-teacher is scheduled to teach $C^{(i)}_{q3}$ during h_2 and $C^{(i)}_{q4}$ during h_1.

In every clause D_j there is at least one literal assigned "true"; assume it is ξ. If $\xi = x_i$ and this is the q-th appearance of this variable then the 2-teacher who is supposed to teach C_{q2} and C_{q3} is scheduled to teach C_{q2} during h_3 and C_{q3} during h_2.

* Here $q+1$ should be computed conventionally, except that $p_i+1 = 1$, to fit the circular structure.

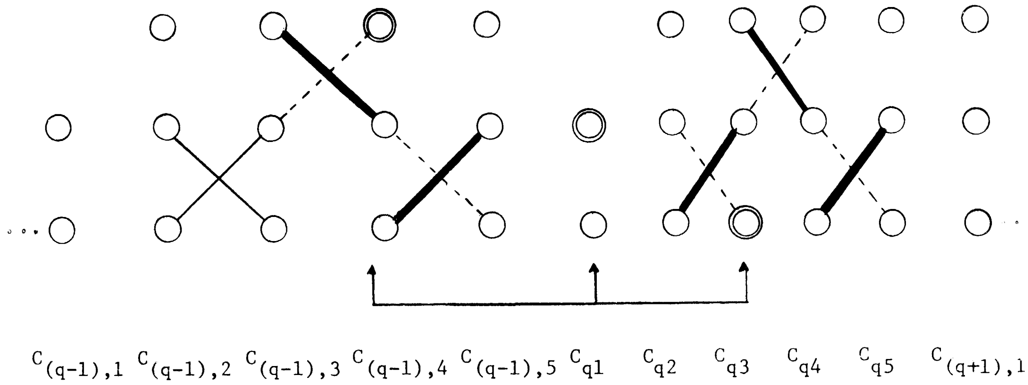

Figure 6

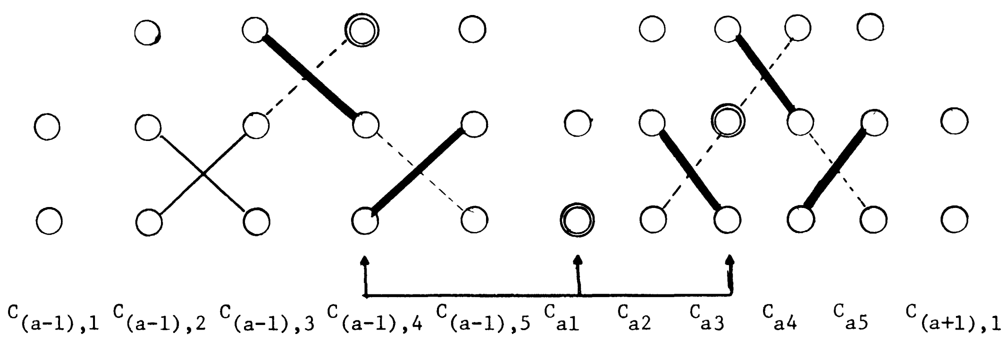

Figure 7

(In our Figure 6 the schedule assigned to each of the 2-teachers discussed so far is shown by a heavy solid line, and the choice we avoided is shown by a dashed line. A light solid line indicates that no choice has been made yet.) The 3-teacher of $C_{(q-1),4}$, C_{q1} and C_{q3} uses h_1 to teach $C_{(q-1),4}$, h_2 to teach C_{q1} and h_3 to teach C_{q3}. (His meets are indicated by the circled vertices.) Finally, the 3-teacher corresponding to D_j uses h_2 to teach C_{q2}. It remains to be shown that he can use h_1 and h_3 to teach the other two classes he is assigned to teach. Clearly, h_1 is never occupied by any other teacher in classes of types C_{a2} and C_{a5}. If $\xi' = x_r$ is another literal in D_j and it is "false", then the corresponding C_{a2} class must be taught during h_2 by the 2-teacher and h_3 remains available. Also if $\xi' = \overline{x}_r$ and it is "false", then C_{a5} must be taught during h_2 by the 2-teacher and again h_3 remains available. Finally, if both remaining literals in D_j are "true" then for one of them we do not follow the scheme used for ξ. For example, if $\xi' = x_r$, it is "true", and this is the a-th appearance of this variable then the 2-teacher teaches C_{a2} during h_2 and C_{a3} during h_3. The 3-teacher teaches $C_{(a-1),4}$ during h_1, C_{a1} during h_3 and C_{a3} during h_2 (as shown in Figure 7). Thus, h_3 remains available to teach C_{a2}, and the scheduling of the 3-teacher corresponding to D_j is now easy. The other cases are similar and the reader may check them out for himself.

Second, assume the answer to the constructed RTT problem is positive, and assume we have a legal scheduling. If in the structure of x_i the 2-teachers assigned to teach C_{q3} and C_{q4} teach C_{q3} during h_1 and C_{q4} during h_2 then x_i is given the value "true", and if they teach C_{q3} during h_2 and C_{q4} during h_1 then x_i is given the value "false". It remains to be shown that each clasue $D_j = \{\xi_1, \xi_2, \xi_3\}$ contains at least one literal which is "true". If $\xi \in D_j$ and it is "false" then h_2 is used for teaching the corresponding class (a C_{a2} if $\xi = x_i$, and a C_{a5} if $\xi = \overline{x}_i$) by the 2-teacher which teaches it and the adjacent class (C_{a3} if $\xi = x_i$ and C_{a4} if $\xi = \overline{x}_i$). Thus, if all three literals are "false" the 3-teacher corresponding to D_j cannot have an assignment to teach its three classes, since it cannot use h_2 .

Q.E.D.

II. The timetable problem with binary teachers is polynomially solvable

Consider the TT problem with the restriction that all teachers are 2-teachers. (A 1-teacher is of no interest.) We shall show that a simple branching procedure solves the problem in polynomial time, since the branching depth is limited.

Our algorithm will determine schedules for the teachers progressively. At a given stage, when part of the teachers have been scheduled we say that a teacher is impossible if he cannot be scheduled consistently; we say that he is implied if there is only one possible way to schedule him consistently with the schedules established so far.

Algorithm:

(1) Set PHASE to 2.

(2) If all teachers have been scheduled, halt with a positive answer.

(3) If there is an unscheduled teacher who is impossible, go to (7).

(4) If there is no unscheduled implied teacher, go to (6).

(5) Let T_i be an unscheduled implied teacher. Temporarily schedule T_i as necessary and go to (2).

(6) Make all temporary schedules permanent. Let T_i be any unscheduled teacher. Arbitrarily choose a schedule for him and record this decision. Set PHASE to 1 and go to (2).

(7) If PHASE = 2, halt with a negative answer.

(8) Reverse the schedule of the recorded teacher and undo all the temporary schedules. Set PHASE to 2 and go to (3).

This algorithm clearly returns a positive answer only if a possible meeting function is constructed. It uses a limited backtracking since only one decision is ever recorded and possibly changed. Thus, the time complexity is polynomial. It is less obvious that this limited backtracking is sufficient to discover a meeting function, if one exists.

Let a component of the evaluation be a set of teachers whose schedules gained permanency simultaneously (in Step (6)). The components may depend on arbitrary choices and on the order in which the teachers are considered. They are numbered consecutively according to their order of occurrence. For completeness, the set of teachers, who are not scheduled or whose schedule had not been made permanent at the time the algorithm terminated, is considered the last component.

Lemma 1: If T_i is a teacher of the last component then none of the class-hours he may use is occupied by a teacher of a previous component.

Proof: New components are started by entering Step (6); but this occurs only when no teacher is implied. Since all teachers are binary, the lemma follows.

Q.E.D.

The lemma implies that whenever the algorithm terminates with a negative answer, after trying both possible schedules for a certain teacher and all the schedules implied by it and failing, we can be sure that all the permanent schedules made before could not have hindered the situation, and thus, the negative answer is conclusive.

It is worth noting here, that the technique of limited branching is applicable in other similar situations, such as the 2-SAT problem (i.e., the satisfiability problem for conjunctive normal forms with at most two literals per clause). Using appropriate data structures in order to find the implications of any decision made, and trying both decisions in Step 6 in parallel (so that the quicker success stops the evaluation of the other possibility), it can be shown that the algorithm has time complexity $O(n)$. Other known algorithms for the 2-SAT problem, such as the Davis and Putnam [4] algorithm (pointed out by Cook [2]) or an algorithm which follows from Quine's work [5] on the concensus (star) operation, have time complexity $O(n^2)$.

III. There is always a meeting function if all teachers and classes have no time constraints

The purpose of this section is to document a theorem which follows from the classical theory of matching in bipartite graphs [6].

We say that a given TT problem has no time constraints if for all $1 \leq i \leq n$ and $1 \leq j \leq m$ $T_i = C_j = H$; we say that it is apparently feasible if neither the teachers nor the classes are overloaded, i.e.:

(1) For all $1 \leq i \leq n$ $\sum_{j=1}^{m} R_{ij} \leq |H|$.

(2) For all $1 \leq j \leq m$ $\sum_{i=1}^{n} R_{ij} \leq |H|$.

Clearly the condition that a TT problem be apparently feasible is necessary for the existence of a meeting function, but is not sufficient.

Our purpose is to prove the following theorem:

Theorem 2: If a TT problem is apparently feasible and has no time constraints then it has a meeting function.

Proof: First let us define the following quantities:

$$r = \sum_{i=1}^{n} \sum_{j=1}^{m} R_{ij} \quad ,$$

$$h = |H| \quad ,$$

$$\nu = m - \lfloor \tfrac{r}{h} \rfloor \quad ,$$

$$\mu = n - \lfloor \tfrac{r}{h} \rfloor \quad .$$

Now, define a bipartite multi-graph $G(X,Y,E)$ in the following way:

$$X = \{x_1,x_2,\ldots,x_n\} \cup \{\xi_1,\xi_2,\ldots,\xi_\nu\} \quad ,$$

$$Y = \{y_1,y_2,\ldots,y_m\} \cup \{\eta_1,\eta_2,\ldots,\eta_\mu\} \quad ,$$

E is a set of edges connecting between vertices of X and vertices of Y constructed as follows. For every $1 \leq i \leq n$ and $1 \leq j \leq m$ we put R_{ij} parallel edges between x_i and y_j. Next, for each $1 \leq i \leq n$ we complete the degree* of x_i to be exactly h by putting $h - \sum_{j=1}^{m} R_{ij}$ edges between x_i and vertices of $\{\eta_1,\eta_2,\ldots,\eta_\mu\}$; it does not matter to which of these vertices these edges are connected provided the degree of each η_k never exceeds h . Also, for each $1 \leq j \leq m$ we complete the degree of y_j to be exactly h by putting $h - \sum_{i=1}^{n} R_{ij}$ edges between y_j and vertices of $\{\xi_1,\xi_2,\ldots,\xi_\nu\}$ again taking care that the degree of each ξ_ℓ never exceeds h . Finally, we complete the degree of the vertices in $\{\xi_1,\xi_2,\ldots,\xi_\nu\}$ and $\{\eta_1,\eta_2,\ldots,\eta_\mu\}$ to be exactly h too by putting edges from any ξ_ℓ to any η_k which both have a lower degree.

It remains to show that this definition is proper in the sense that all the conditions it implies are easily met.

The number of edges we construct in the completion of the degrees of $x_1,x_2,\ldots,x_n$ is $n \cdot h - r$. Thus, we can do this if $\mu \cdot h \geq n \cdot h - r$, and μ satisfies this inequality. Similarly, ν satisfies the condition for the possibility of the completion of the degrees of $y_1,y_2,\ldots,y_m$. Finally, the number of edges required to complete the degrees of $\xi_1,\xi_2,\ldots,\xi_\nu$ is $\nu \cdot h - (m \cdot h - r)$ which is equal to $r - \lfloor \frac{r}{h} \rfloor \cdot h$. (This is the remainder of r upon division by h.) Similarly, the number of edges required to complete the degrees of $\{\eta_1,\eta_2,\ldots,\eta_\mu\}$ is the same. Thus, the last part of the construction raises no difficulties either.

Next, let $\Gamma(A)$, where $A \subseteq X$, be the set of vertices $B \subseteq Y$ such that there is an edge $a \to b \in E$ where $a \in A$ and $b \in B$.

Lemma 2: For every $A \subseteq X$ $|\Gamma(A)| \geq |A|$.

Proof: There are $h \cdot |\Gamma(A)|$ edges incident to $\Gamma(A)$ in G . This includes all the edges which are incident to A . Thus,

$$h \cdot |\Gamma(A)| \geq h \cdot |A| \quad .$$

Q.E.D.

Lemma 2 assures that Hall's condition holds, and thus, by Hall's theorem [6] there is a set of $n + \nu (= m + \mu)$ edges, no two of which have a common end point. We now use this set of edges M , (which is commonly called a complete match of X to Y) to define the meeting function for the first hour $h_1 \in H$; if $x_i \to y_j \in M$ then $f(i,j,h_1) = 1$; otherwise $f(i,j,h_1) = 0$. Clearly conditions (c) and (d) hold for h_1 . Next we remove M from E . The new graph has degree $h - 1$ for all its vertices, and as in Lemma 2, Hall's condition holds again. This assures the existence of another complete match M' of X to Y and we can use it to define $f(i,j,h_2)$ for all $1 \leq i \leq n$ and $1 \leq j \leq m$. We repeat this until by the h-th application all E's edges have been used. This assures that condition (b) holds. Thus, the proof of Theorem 2 is complete.

Q.E.D.

The technique used here is an easy generalization of the one classically used to prove the school dance theorem. (See, for example, reference [7].) Since the proof is constructive and a complete match of X to Y can be obtained in polynomial time (Hopcroft and Karp [8]), this technique can be used in order to find an appropriate scheduling in polynomial time rather than just proving its existence.

IV. The two-commodity integral flow problem is NP-complete

Knuth (see [9]) has shown that the multi-commodity integral flow problem is NP-complete. His reduction, from the satisfiability problem, uses as many commodities as there are clauses.

We present a reduction of the satisfiability problem to the Two-Commodity Integral Flow in Directed graphs (D2CIF), and in turn, a reduction of the D2CIF problem to the U2CIF (the undirected version).

* The degree of a vertex is the number of edges incident to it.

Definition of D2CIF : Given the following data

(1) $G(V,E)$ a directed finite graph. A directed edge from u to v is denoted $u \to v$.

(2) A capacity function $c: E \to N$, where N is the set of non-negative integers.

(3) Vertices s_1 and s_2 (not necessarily distinct) which are called the sources.

(4) Vertices t_1 and t_2 (not necessarily distinct) which are called the terminals.

(5) A positive integer R called the requirement.

The problem is to determine whether there exist two flow functions f_1 and f_2, both $E \to N$, such that

(a) For every $u \to v \in E$, $f_1(u \to v) + f_2(u \to v) \leq c(u \to v)$. Intuitively, the commodities flow along the directed edge $u \to v$ from u to v. The total flow along an edge is bounded from above by the capacity of the edge.

(b) For each commodity $i \in \{1,2\}$ and each vertex $v \in V - \{s_i, t_i\}$

$$\sum_{u \to v \in E} f_i(u \to v) = \sum_{v \to w \in E} f_i(v \to w) .$$

This is the conservation rule which states that for each commodity the amount of flow which enters a vertex equals the flow which emanates from it.

(c) For each commodity $i \in \{1,2\}$ let the total flow be

$$F_i = \sum_{s_i \to v \in E} f_i(s_i \to v) - \sum_{v \to s_i \in E} f_i(v \to s_i) .$$

Then it is required that

$$F_1 + F_2 = R .$$

A flow problem is simple if the capacities of all edges are equal to one.

Theorem 3: Simple D2CIF is NP-Complete.

Proof: It suffices to show satisfiability $\propto$ Simple D2CIF .

Let the clauses of the satisfiability problem be $D_1, \ldots, D_k$ and $x_1, \ldots, x_\ell, \bar{x}_1, \ldots, \bar{x}_\ell$ be the literals. For each variable x_i we construct a lobe as shown in Figure 8. (Here p_i is the number of occurrences of x_i in the clauses and q_i is the number of occurrences of $\bar{x}_i$). The capacity of all edges is 1. The lobes are connected to one another in series: v_t^i is connected to v_j^{i+1} , s_1 is connected to v_s^1 and v_t^ℓ to t_1 . s_2 is connected to all the vertices v_j^i and $\bar{v}_j^i$ where j is odd. In addition there are vertices $D_1, D_2, \ldots, D_k$ and an edge from each to t_2 . For the j-th occurrence of x_i ($\bar{x}_i$) there is an edge from v_{2j}^i ($\bar{v}_{2j}^i$) to the D_r in which it occurs. The requirement is $R = 1 + k$.

(a) Assume that the total flow is R . Since the edges $D_j \to t_2$ $j = 1, \ldots, k$ of total capacity k disconnect all paths from s_2 to t_2 , the maximum flow of the second commodity is k . The requirement can be met if and only if the flow of the first commodity is equal to one (its maximum). This flow must pass through all lobes. Define x_i to be "true" if and only if the first commodity flow passes through the lower path of the i-th lobe. In this case, flow of the second commodity may pass through the upper part of the lobe to all the clauses which contain x_i .

Since the flow of the second commodity is equal to k , through each vertex D_j passes positive flow. Let the flow to D_j originate in the i-th lobe. If it comes from the upper part of the lobe then $x_i \in D_j$ and the first commodity must flow through the lower part of

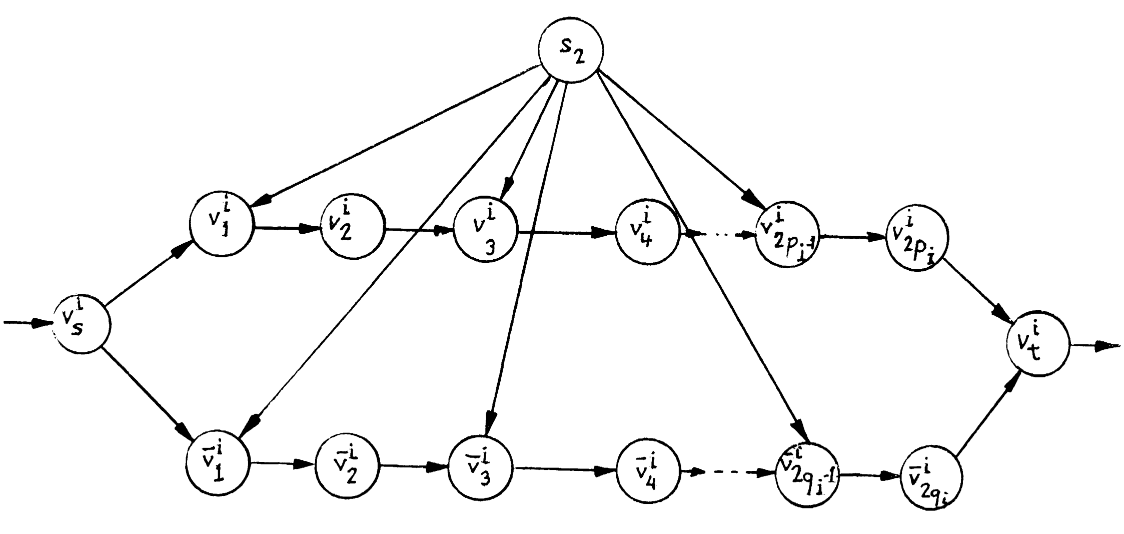

Figure 8

the lobe. Thus x_i is "true" and D_j is satisfied.

If the flow comes from the lower part of the lobe a similar argument holds. This completes the proof that the expression is satisfiable.

(b) If the expression is satisfiable, we send the first commodity flow through the lower path of the i-th lobe if and only if x_i is "true". Since each clause D_j contains at least one literal x_i or $\bar{x}_i$ which is "true", the second commodity passes through the upper or lower path depending on whether x_i or $\bar{x}_i$ is "true".

The total flow thus constructed, $F_1 + F_2 = 1 + k$, meets the requirement.

Q.E.D.

Next, we show that U2CIF [10] is NP-complete too. The definition of U2CIF is similar to that of D2CIF except that the graph is undirected. Denoting an undirected edge between u and v as u-v, its capacity is $c(u\text{-}v)$. However, the flow has a direction. If the flow is from u to v $f_i(u\text{-}v)$ is positive and $f_i(v\text{-}u)$ is its negation. (Note that $c(u\text{-}v) = c(v\text{-}u) \geq 0$.) Condition (a) changes into

$$|f_1(u\text{-}v)| + |f_2(u\text{-}v)| \leq c(u\text{-}v) \qquad u\text{-}v \in E$$

implying that the total flow in both directions is less than the capacity.

As before, condition (b) assures that for each $v \in V - \{s_i, t_i\}$ the total flow of commodity i entering v is equal to the total flow of commodity i emanating from v. I.e.,

$$\sum_{u\text{-}v \in E} f_i(u\text{-}v) = 0 .$$

Let the total i-th commodity flow be $F_i = \sum_{s_i\text{-}u \in E} f_i(s_i\text{-}u)$. Condition (c) states that the total flow equals R: $F_1 + F_2 = R$.

Theorem 4: Simple U2CIF is NP-complete.

Proof: It suffices to show

Simple D2CIF $\propto$ Simple U2CIF.

Each edge u-v of the directed graph is replaced by the construct shown in Figure 9. (u or v may be one of the sources or terminals.) Only the unlabeled vertices of the construct are new and do not appear elsewhere in the graph. It can be shown that R can be met in the D2CIF problem if and only if the requirement $R + 2\cdot|E|$ can be met in the corresponding U2CIF problem.

Clearly the completeness of the above problems imply the completeness of the two commodity integral flow problems with arbitrary capacities for both the directed and the undirected case. Also the completeness of $m \geq 2$ commodity integral flow problem follows.

V. A polynomial algorithm for the two commodity real flow in undirected graphs

The problem of real flow in graphs is similar to the problem of integral flow except that both the capacities and the flows can be real numbers instead of integers. Ford and Fulkerson [11] and Edmonds and Karp [12] have obtained polynomial algorithms for the single commodity case. They have also shown that for a single commodity, integral capacities imply the existence of a maximum flow which is also integral. For more commodities this is no longer true. Hence, the integral and the real flow problems are different.

T.C. Hu [10] devised an algorithm to find maximum real two commodity flow in an undirected graph. This algorithm is based on the fact that the max-flow min-cut theorem holds in this case. He used pairs of augmenting paths to recirculate the first commodity and increase the second. The algorithm has the interesting and useful property that if all capacities are even integers the maximum flow is found in integers. The convergence proof is based on this fact. This approach has two major drawbacks. First, it does not prove

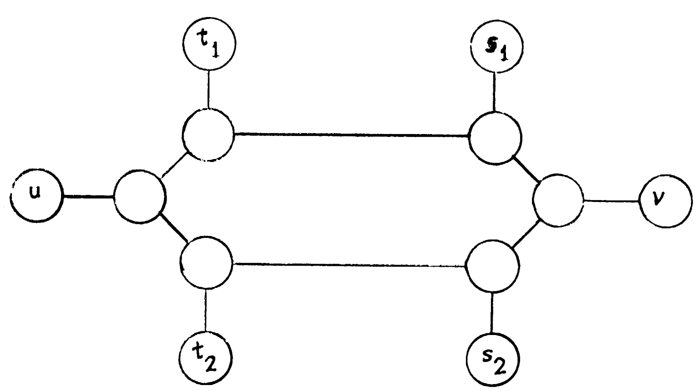

Figure 9

anything about the convergence for arbitrary real capacities. Second, the number of steps of this algorithm is not bounded by a function of the number of edges but rather by the size of the capacities. This leads in some cases to an exponential number of steps (in terms of the length of the input data).

We change the original algorithm in two ways. First, in each augmentation we increase the flow as much as possible, thus eliminating the need to repeat an augmenting path. Second, we find the augmenting path in order of increasing length through a technique similar to that of Dinic [13, 14]. The resulting algorithm has running time $O(|V|^2|E|)$ regardless of the size of the capacities. (We do assume, however, that all arithmetic operations on the capacities are primitive; can be done in one step.) It is interesting to note here that the inclusion of only one of the improvements does not suffice to guarantee a polynomial time algorithm.

The two-commodity flow problem for undirected graphs has then a somewhat unique feature. It is polynomially solvable in reals, and if the capacities are integers, the algorithm produces a maximum flow which may not be integral, but is in halfs. Yet, if one insists on an integral solution, the problem is NP-complete.

We follow Hu's algorithm in some points. To maximize $F_1 + F_2$ we first maximize F_1 then increase F_2 without changing F_1 (although f_1 may change on individual edges). It can be proved that such a scheme maximizes $F_1 + F_2$. The crux of the method is the way F_2 is increased. Towards this end, pairs of augmenting paths are found. A path is <u>simple</u> if no vertex appears more than once.

For every $u\text{-}v \in E$ define

$$\alpha(u\text{-}v) = \frac{1}{2}\,(c(u\text{-}v) - f_1(u\text{-}v) - f_2(u\text{-}v)) .$$

2α is the upper bound on the additional flow that can be pushed from u to v.

An <u>s_2-t_2 path</u> is a simple path from s_2 to t_2 $\pi_1 = (s_2{=}v_0,\ldots,v_r{=}t_2)$ where $\alpha(v_i\text{-}v_{i+1}) > 0$ $i = 0,\ldots,r-1$. The residual capacity of an s_2-t_2 path is

$$\alpha(\pi_1) = \min_{u\text{-}v\in\pi_1} \{\alpha(u\text{-}v)\} .$$

For every $u\text{-}v \in E$ define

$$\beta(u\text{-}v) = \frac{1}{2}\,(c(u\text{-}v) - f_1(u\text{-}v) + f_2(u\text{-}v)) .$$

2β is the upper bound on the additional flow that can be pushed through the edge; where the additional first commodity is pushed from u to v and the additional second commodity is pushed from v to u.

A <u>t_2-s_2 path</u> is a simple path from t_2 to s_2 $\pi_2 = (t_2{=}v_0,\ldots,v_q{=}s_2)$ where $\beta(v_i\text{-}v_{i+1}) > 0$ $i = 0,\ldots,q-1$. The residual capacity of a t_2-s_2 path is

$$\beta(\pi_2) = \min_{u\text{-}v\in\pi_2} \{\beta(u\text{-}v)\} .$$

A <u>pair of augmenting paths</u> is an s_2-t_2 path and a t_2-s_2 path. The residual capacity of a pair of augmenting paths $\gamma(\pi_1,\pi_2) = \min\{\alpha(\pi_1),\beta(\pi_2)\}$. After a maximum flow of the first commodity is found it may block the flow of the second commodity. Possibly, a diversion of the flow of the first commodity would allow an increase of the flow of the second commodity. The algorithm finds pairs of augmenting paths. γ units of the first commodity are recirculated through the paths (π_1,π_2). Thus enabling the increase of the second commodity by γ units from s_2 to t_2 along each path.

<u>Two Commodity Real Flow Algorithm:</u>

(1) Let the initial flow f_2 be zero. Find maximum flow F_1 (from s_1 to t_1).

(2) Find a pair of augmenting paths (π_1,π_2). If no such pair exists - stop: maximum flow has been reached.

(3) For every edge $u\text{-}v$ of π_1 increase the flow of each commodity by γ (in the direction of the path).

(4) For every edge $u\text{-}v$ of π_2 increase the flow of the first commodity by γ and decrease the flow of the second commodity by the same amount.

(5) Go to step 2.

The following facts concerning the algorithm are stated without proof.

(a) Starting with a feasible flow, at the end of each iteration the flow is feasible.

(b) Each iteration increases the flow F_2 by 2γ while F_1 remains unchanged.

(c) When the algorithm halts (in step 2: there are no more pairs of augmenting paths) the flow is maximum. (The proof is very similar to that in Hu's paper.)

(d) After augmenting along a pair of paths, at least one of the paths is saturated and can no longer be used. (In this property our algorithm differs from Hu's algorithm.)

We have not yet specified how to find the pairs of augmenting paths at step 2; if they are not chosen properly the number of iterations may be exponential. Here we can follow Edmonds and Karp [12] choosing the shortest path. However, following Dinic [13, 14] yields a better time bound.

To this end, we construct two types of auxiliary graphs (one for each type of path).

<u>Construction of the s_2-t_2 Auxiliary Graph:</u>

(1) Perform a breadth first search from s_2 to t_2 considering only the edges $u\text{-}v$ for which $\alpha(u\text{-}v) > 0$.

(2) The vertices of the auxiliary graph are the vertices which have been reached by the search. These vertices are divided by the search into levels.

(3) u-v is an edge of the auxiliary graph if and only if $\alpha(u\text{-}v) > 0$ and the level of v is greater by one than that of u.

We obtain s_2-t_2 paths in the original graph by finding paths from s_2 to t_2 in the s_2-t_2 auxiliary graph; here we use depth first search in order to obtain all paths in at most $O(|V|\cdot|E|)$ steps.

The t_2-s_2 auxiliary graph and the t_2-s_2 paths are found similarly. This time we start with t_2, end with s_2 and β replaces α.

When a pair of augmenting paths (π_1,π_2) is found, the capacities along the path must be reduced by $\gamma(\pi_1,\pi_2)$. Whereupon, the residual capacity of some edges is reduced to zero. These edges are deleted from the auxiliary graph.

Augmenting paths are found until there are no more paths in one of the auxiliary graphs. We now construct a new auxiliary graph to replace the exhausted one.

If an auxiliary graph cannot be constructed (the breadth first search fails to reach the end vertex) there exist no more pairs of augmenting paths. The algorithm terminates and maximum flow has been found.

Timing considerations:

All the paths in a new auxiliary graph are longer than those of the exhausted auxiliary graph. Therefore, at most $|V|-1$ auxiliary graph of each type may be constructed. Exhausting the paths in an auxiliary graph may take time proportional to $|V|\cdot|E|$. consequently, finding all the pairs of augmenting paths takes $O(|V|^2|E|)$. This process is the most time consuming part of the algorithm, and $O(|V|^2|E|)$ is the time bound of the two commodity real flow algorithm.

The existence of a finite algorithm for arbitrary capacities enables us to prove two commodity minimum cut maximum flow theorem [10] rigorously.

Directed multi-commodity real flow problem $(m \geq 2)$ and undirected multi-commodity real flow $(m \geq 3)$ cause major difficulties since the max-flow min-cut theorem does not hold. Unlike the real two commodity undirected flow problem, the maximum flow does not necessarily occur when any one of the commodities attains its maximum. Thus, current techniques are not adequate.

On the other hand, multi-commodity flow problems can be solved with the aid of linear programming. (This proves that all multi-commodity flow problems are in NP.) Moreover, some flow problems have been shown to be polynomially equivalent to linear programming [15].

The prospect of demonstrating the difficulty of multi-commodity real flow problems is not too promising since it would imply the difficulty of linear programming, and such attempts have not yet been successful.

References

[1] Gotlieb, C.C., "The Construction of Class-Teacher Time-Tables", Proc. IFIP Congress 62, 1963, pp. 73-77.

[2] Cook, S.A., "The Complexity of Theorem-Proving Procedures", Proc. of the Third Annual ACM Symp. on Th. of Computing, 1971, pp. 151-158.

[3] Karp, R.M., "Reducibility Among Combinational Problems", in Complexity of Computer Computation eds. R.N. Miller and J.W. Thacher, Plenum Press, 1972, pp. 85-104.

[4] Davis, M., and H. Putnam, "A Computing Procedure for Quantification Theory", J. ACM, 1960, pp. 201-215.

[5] Quine, W.V., "On Cores and Prime Implicants of Truth Functions", The Amer. Math. Monthly, Vol. 66, No. 9, 1959, pp. 755-760.

[6] Hall, P., "On Representations of Subsets", J. London Math. Doc., Vol. 10, 1935, pp. 26-30.

[7] Berge, C., The Theory of Graphs and its Applications, John Wiley 1962. Page 92, Example 2.

[8] Hopcroft, J.E., and Karp R.M., "An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs", SIAM J. on Computing, Vol. 2, 1973, pp. 225-231.

[9] Karp, R.M., "On the Complexity of Combinatorial Problems", Networks, Vol. 5, No. 1, 1975, pp. 45-68.

[10] Hu, T.C., "Multi-Commodity Network Flows", J. ORSA, Vol. 11, No. 3, 1963, pp. 344-360.

[11] Ford, L.R. Jr., and Fulkerson, D.R., Flows in Networks, Princeton University Press, 1962.

[12] Edmonds, J., and Karp, R.M., "Theoretical Improvements in Algorithm Efficiency for Network Flow Problems", J. ACM, Vol. 19, No. 2, 1972, pp. 248-264.

[13] Dinic, E.A., "Algorithm for Solution of a Problem of Maximum Flow in a Network with Power Estimation", Sov. Math. Dokl. 11, 1970, pp. 1277-1280.

[14] Even, S. and R.E. Tarjan, "Network Flow and Testing Graph Connectivity", to appear in SIAM J. on Computing.

[15] Itai, A., "Multi-Commodity Flow", Ph.D. dissertion, submitted to the Fienberg Graduate School of the Weizmann Institute of Science, Rehovot, Israel, 1975.