1 Multiprocessor Scheduling using network flow models

Most common model in the scheduling theory literature is based on network flow models. We shall present a scheduling algorithm based on the popular **Max-flow Min-cut** theorem that determines an optimal schedule for a *two* processor system in scheduling modularly divisible tasks. To understand this algorithm, we shall digress a little and introduce the concept of a *commodity flow network* and the required terminology.

Commodity flow network. Consider a flow network shown in Fig. 1. The max-

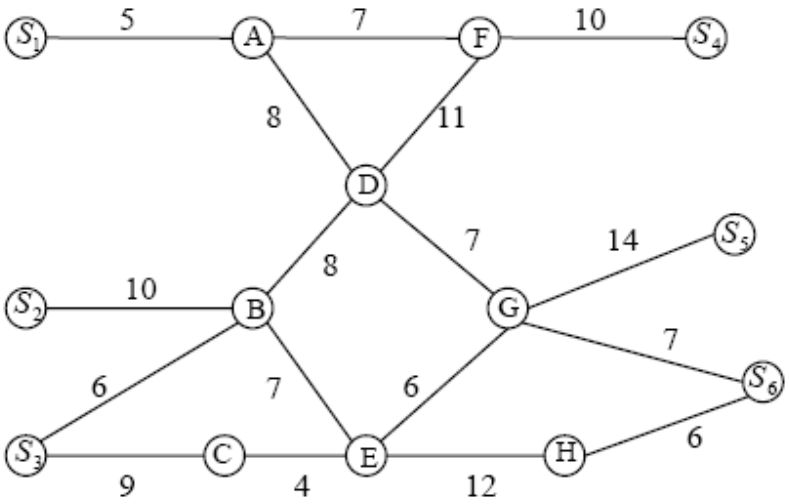

Figure 1: Commodity flow network

imum flow problem is a problem involving a commodity network graph. The nodes labeled S1,S2, and S3 are **source nodes** and the nodes labled S4,S5, and S6 are **sink nodes**. Between these nodes lie several interior nodes linked by weighted branches. Source nodes represent production centres, each theoretically capable of producing infinite amount of specific commodity. Sink nodes represent demand centres, each of which can consume infinite amount of commodity, theoretically. The branches represent commodity transport linkages, with the weight of a branch indicating the capacity of the corresponding link.

A commodity flow in such a network is represented by weighted directed arrows along the branches of the network. The directed arrows represent the direction of flow. Each

arrow in Fig. 2 has a pair of numbers. The first one represents the **capacity** of the link and the second one represents the actual flow on that branch. A **feasible commodity flow** is such that:

1. at each intermediate node, the sum of out-flow = sum of in-flow.
2. at each sink node, the net flow into the nodes in nonnegative, and so at the out flow at the source nodes.
3.net flow on a branch does not exceed the capacity of that branch.

We will now introduce some definitions that will used frequently. The value of the

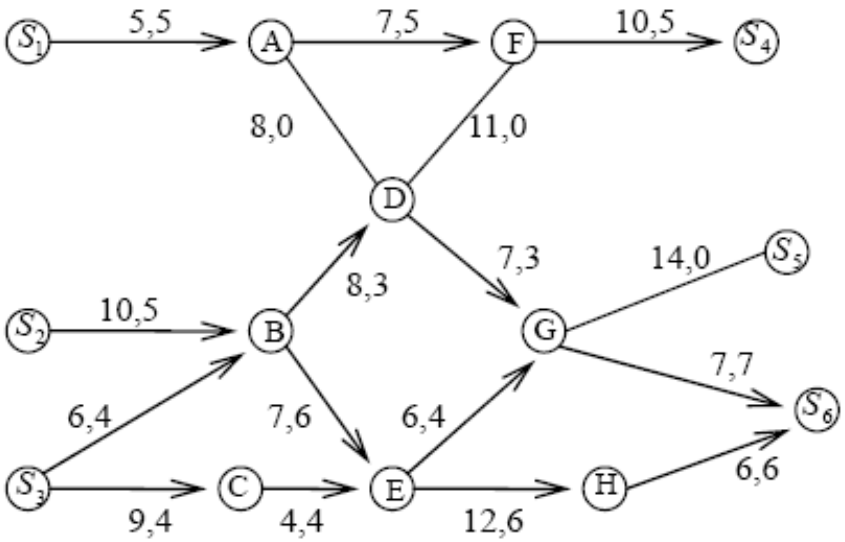

Figure 2: Flow in a commodity flow network

commodity flow is the sum of the net flows out of the source nodes of the network. However, because the net flow into the network must also be equal to the net flow out of the network, the **value** of the commodity flow is defined as equal to the sum of the net flow into the sink nodes. Accordingly, the value of the flow for the Fig. 2 is 18. Now, a **maximum flow** is a feasible flow whose value is maximum among all the feasible flows. Fig. 3 shows a maximum flow for this example. A **cutset** of a commodity flow network is a set of edges which when removed disconnects the source nodes from the sink nodes. Note that no proper subset of a cutset is a cutset. In Fig. 3, the branches (S1,A),(B,E),(B,D), and (C,E) form a cutset. The **weight of the cutset** is equal to the sum of the capacities of the branches in the cutset.

In Fig. 3, the weight of the cutset shown in 24. It is important to realize that this is

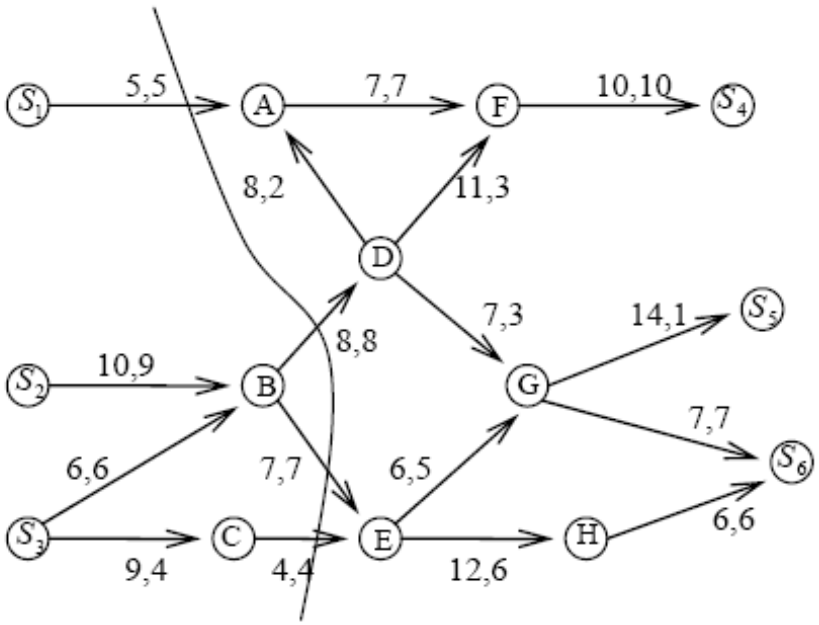

Figure 3: Maximum flow and minimum cut in a commodity flow network

also equal to the maximum flow!! This is merely not a coincidence, as this fact is a well established following theorem.

Max-flow, Min-cut Theorem: The value of a maximum flow in a commodity network is equal to the weight of a minimum weight cutset of that network.

We will omit the proof, and use this result in developing a scheduling strategy that solves a two processor assignment problem.

Two-processor assignment problem and its solution. Consider a modularly divisible load as shown by a intermodule-connection graph in Fig. 4. The modules are to be scheduled on a two processor network in such a way that the total completion time (sum of communication and computation) is a minimum. We are given the execution times of the individual modules on each of the processors is given below. The symbol ∞ in the table denotes that the respective module cannot run on that processor. The weights on the edges in the intermodule-connection graph represents the communication cost when respective modules communicate. We use (Module,execution time) to denote the module and its execution time on a particular processor. Thus, on P1 - (A,5), (B,2), (C,4), (D,6), (E,5), (F,∞) and on P2 - (A,10), (B,∞), (C,4), (D,3), (E,2),

(F,4), respectively. Now,

(i) suppose that it is a requirement to assign module B to P1 and F to P2, then the

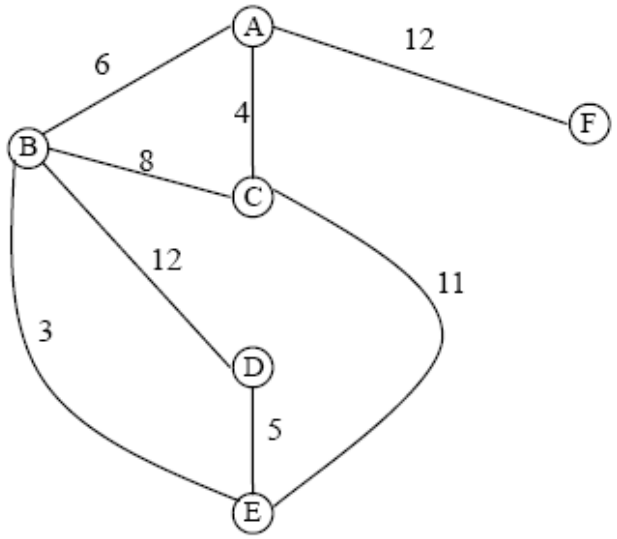

Figure 4: Intermodule-connection graph

optimum way of minimizing the intermodule cost alone is to view B and F as source and sink nodes, respectively, of a commodity flow graph. the minimum cut is the minimum intermodule cost cut, and this assigns modules B,C,D and E to P1 and F to P2.

(ii). suppose if we desire to minimize the running time, then the assignment A,B and C to P1 and D,E, and F to P2 meets our objective.

However, neither of the above attempts minimize the total completion time. We are in the search of a feasible schedule that minimizes the total completion time of the entire task. To generate such a schedule, we modify the intermodule-connection graph as follows. Then, we apply the Max-flow Min-cut theorem to determine the optimal schedule. Thus, *we have to modify the intermodule-connection graph in such a way that every cutset generates a feasible schedule and reflects the total cost for that assignment.* Then we can apply Max-flow,Min-cut theorem to choose the minimum cost schedule. We do the modification as follows.

1. Add nodes S1 and S2 that represent processors P1 and P2, respectively. S1 is the unique source node and S2 is the unique sink node.
2. For each node other than S1 and S2, add a branch from that node to each of S1 and S2. The weight of the branch to S1 carries the cost of executing the corresponding

module on P2, and the weight of the branch from that node to S2 carries the cost of executing the corresponding module on P1. Note that there is a reversal of subscripts and this is intentional.

The modified graph is a commodity flow network of the general type. Each cutset partitions the graph into disjoint subsets, with S1 and S2 in the distinct subsets. We associate a module assignment with each cutset such that if the cutset partitions a node into subset containing S1, then the corresponding module is assigned to processor P1. Thus, following this notion, for the cut shown in Fig. 5, the assignment of modules is such that A, B, and C to P1 and D,E, and F to P2. With this assignment, we se

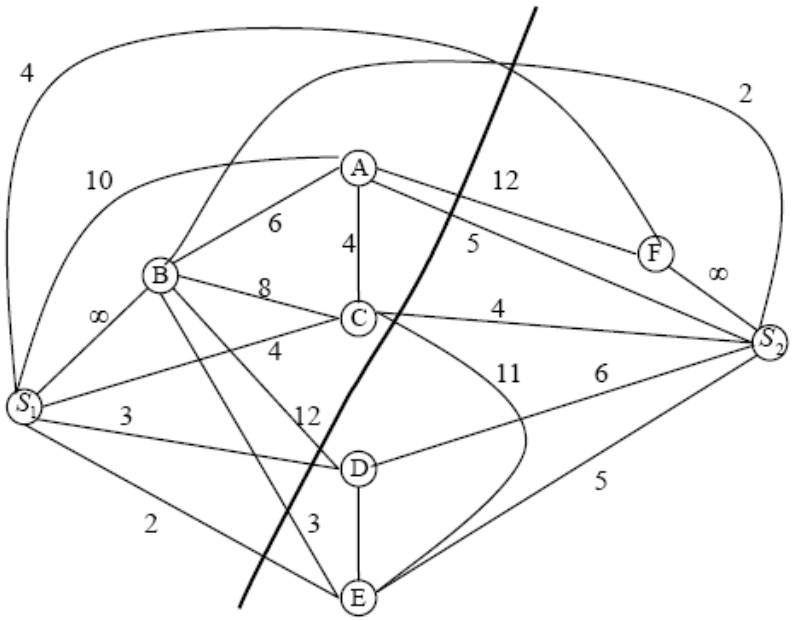

Figure 5: Modified module-interconnection graph and a cut that determines a module assignment

that there is a one-to-one correspondence between module assignments and cutsets of a commodity flow graph. Following theorem proves this claim.

Theorem 3.1. The weight of a cutset of the modified graph is equal to the cost of the corresponding module assignment.

Proof: A module assignment incurs two types of cost. One is from intermodule references between processors. The other cost is the running time cost. on a specific processor. Now, the cutset corresponding to a module assignment problem has two types of branches. One type of branch represents the cost of intermodule references.

All costs due to such references contribute to the weight of the cutset, and no other intermodule references contribute to the weight of the cutset. The second type of branch in the cutset is a branch from an internal node to the source or the sink nodes. If an assignment places a module in processor P1, then the branch between that node and S2 is in the cutset, and this contributes a cost equal to the cost of running that module on processor P1, because the branch to S2 carries the P1 running time cost. Similarly, we can acount for the running time costs on processor P2.
Thus, the weight of a cutset accounts for all the costs due to its corresponding assignment, and no other costs contribute to the weight. Hence the proof. **Q.E.D**

Exercise 3.1. In the example presented in this section, show the minimum cost cut and what is the cost of such an assignment?.

It may be noted that what we have solved is for a two-processor system. It has been shown in the literature that this technique can be extended to three or more processors. It is beyond the scope of this course to deal with this aspect.