On Time

(Continued from page 1)
new world of scheduling problems, in particular, how to meet the needs of the many users of a time- or resource-constrained computer system.

Mathematics has fundamentally changed scheduling. First, mathematical analysis introduced taxonomy. For example, some scheduling problems are deterministic or static (all necessary information is known), others are dynamic (information changes over time) and may involve stochastic processes (such as customers arriving at a bank). In developing this taxonomy, among the many questions that have been considered are as follows:

1. What tasks have to be accomplished, and by which machines? (E.g., the tasks could be financial transactions, and the "machines" could be either human or automated bank tellers.)
2. Are there restrictions on the order in which tasks must be accomplished? (E.g., you cannot put a roof on a new house before the foundation is laid.)
3. Are there other priorities given for the tasks? (E.g., you want to make sure the guest room is done before Grandma comes for her yearly visit.)
4. Are there penalties when tasks are completed after the due dates (or premiums for tasks completed early, such as in [2])?
5. Once a machine has started work on a task, must it complete the task, or can it interrupt its work if a higher priority task appears?
6. What goals are involved in the scheduling? Minimizing idle time of machines? Completing the tasks as quickly as possible? Using as few machines as possible?

Beyond taxonomy, modern mathematics has offered a broad array of optimization techniques and other tools to assist with the solution and understanding of scheduling problems. One of the early pioneers in the mathematics of scheduling was NASA. Carrying out the Apollo project to land on the moon required careful use and timing of resources. NASA still must schedule carefully for efficient operation of its Shuttle Fleet. Another pioneer was Bell Laboratories (now AT&T Bell Laboratories and Bellcore), where mathematicians such as Ronald Graham, David Johnson, and Edward Coffman made many advances in the mathematical theory of scheduling. Scheduling is now an established sub-area of mathematics within broader areas of mathematics concerned with operations research, management science, and computer science. Tools that have been used to study scheduling come from combinatorics and number theory, as well as other parts of mathematics, and include structures such as undirected and directed graphs and partially-ordered sets, and techniques such as mathematical programming, and packing and coloring algorithms. (See **Critical Path Scheduling**, below, as well as the articles by C. Biehl and K. DeVizia in this newsletter.)

References [see also the **Scheduling Minibibliography** on p. 10 of this newsletter]:

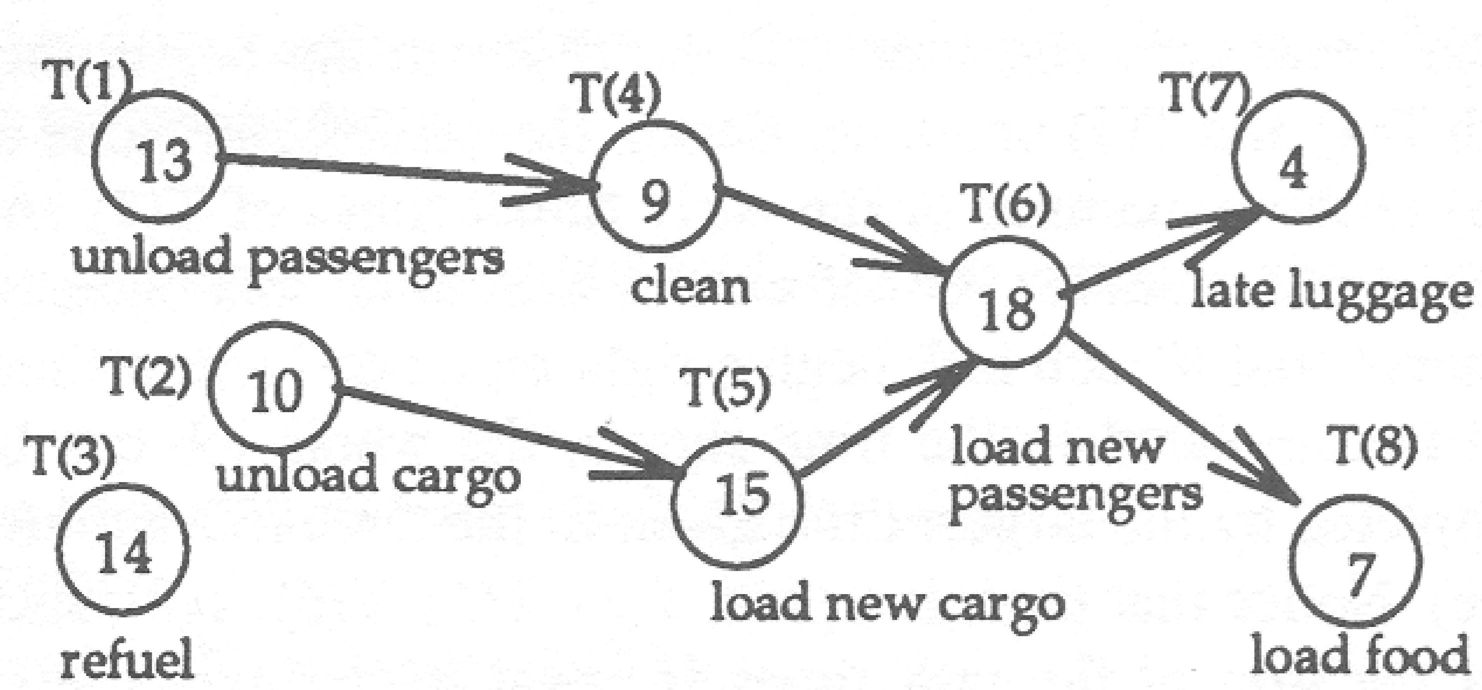
[1] Johnson, Dirk, "Denver May Open Airport in Spite of Glitches", *NY Times*, Wednesday, July 27, 1994, p. A14. This is one of a series of articles appearing over several months about delays in the planned opening of the new Denver airport.

[2] Margolick, David, "Quake-Damaged Freeway Reopening Ahead of Time", *NY Times,* Tuesday, April 12, 1994, p. A12. This article recounts the reconstruction program to reopen the Santa Monica Freeway after it was closed by the Los Angeles earthquake. The contractor responsible for the reconstruction finished the work 74 days ahead of schedule! The reconstruction cost $29.4 million (which includes a $200,000 bonus per day for each of those 74 days, or $14.8 million!)

Critical Path Scheduling

To give you some flavor of a modern scheduling problem, consider the question of efficiently turning around a shuttle plane providing service between two cities. Among the tasks that must be completed to do this are: refueling the plane, putting new drinks and food aboard, cleaning the cabin, unloading current cargo and loading new cargo, unloading current passengers and loading new ones. Clearly, some of these tasks must be done before others; one cannot clean the cabin before the current passengers are deplaned. One can use a "task analysis digraph," i.e., a graph with vertices (circles) representing tasks, and directed edges (arrows) representing precedence constraints (see figure). The length of time (in minutes) to perform a task is indicated inside the circle representing the task. An arrow from Task i to Task j means that Task i must be completed

(Continued on page 10)



Task Analysis Digraph