Abstract Let $T = \{T_1, T_2, \ldots, T_n\}$ be a set of $n$ independent tasks and $P = \{P_1, P_2, \ldots, P_m\}$ a set of $m$ processors. During each time instant, each processor can be used by a single task at most. A schedule is for each task an allocation of one or more time intervals to one or more processors. A schedule is said to be optimal if it minimizes the maximum completion time. We say a schedule $S$ has the machine saturation property (MS property) if, at any time instant of task execution, all the machines are simultaneously busy. In this paper, we analyze the conditions under which a parallel scheduling system allows a schedule with the MS property. While for some simple models the analytical conditions can be easily stated, a graph model approach is required when conflicts of processor usage are present. For this reason, we define the class of saturated graphs that correspond to scheduling systems with the MS property. We present efficient graph recognition algorithms to verify the MS property directly on some classes of saturated graphs.

Keywords Makespan · Multiprocessor task scheduling · Polynomial algorithms · Intersection graphs

1 Introduction and objectives

Let $T = \{T_1, T_2, \ldots, T_n\}$ be a set of $n$ independent tasks and $P = \{P_1, P_2, \ldots, P_m\}$ a set of $m$ processors. During each time instant, each processor can be used by a
single task at most. A schedule is for each task an allocation of one or more time intervals to one or more processors. A schedule is said to be optimal if it minimizes the maximum completion time. We say a schedule $S$ has the machine saturation property (MS property) if, at any time instant of task execution, all the machines are simultaneously busy. In this paper, we analyze the conditions under which a parallel scheduling system permits a schedule with the MS property. Our objective is to give a combinatorial characterization of the properties of the system in which these schedules are allowed. At first, we present the problem by analyzing some simple cases. Then, we will use a graph model approach for a multiprocessor task scheduling model with prespecified processor allocation. We introduce the class of saturated graphs and give a characterization of them, and polynomial time recognition algorithms for some graph classes. To specify scheduling problems, we use the well-known $\alpha|\beta|\gamma$-notation (see Brucker 2001).

In order to clarify our specific goal in this paper, let us consider the following well-known problem. Consider a set of $m$ identical processors or machines. Tasks are independent, that is there are no precedence constraints on tasks, and they are preemptable, that is the execution of a task can be interrupted and resumed later with no additional cost. Let us denote by $p_j$ the duration of task $T_j$. The problem $P_m|pmtn|C_{\text{max}}$ of finding a schedule that minimizes the maximum completion time can be solved in linear time (Brucker 2001). However, even for this simple case, not all the schedules have the MS property. Indeed, we can state the following two simple conditions:

Condition 1 $n \geq m$;
Condition 2 $\sum_{j} \frac{p_j}{\max} = m$.

Obviously, Condition 1 is necessary for a schedule to satisfy the MS property. Condition 2 is necessary and sufficient and is equivalent to saying that $\sum_{j} \frac{p_j}{m} \geq C_{\text{max}}$; the proof is straightforward.

Clearly, the latter condition is a general one and holds for a number of scheduling models as, for instance, $P_n|\text{max}$. Yet, it is necessary to compute $C_{\text{max}}$ which, for this latter and other cases, is an NP-hard problem. One may wonder when, and for which scheduling models, it is possible to verify a priori the MS property, that is before finding a minimum makespan schedule.

Further on in the paper, we analyze in detail two multiprocessor scheduling systems and show how, by using a graph model approach, it is possible to verify the MS property without finding the optimal schedule. In particular, we consider the two models $P|fix_j, p_j = 1|\text{max}$ and $P|fix_j, pmtn|\text{max}$. The term $fix_j$ is used to denote a scheduling system where each task $T_j$ requires a specific subset $P(T_j)$ of processors to be used simultaneously and, whenever two tasks have one or more processors in common, they cannot be executed simultaneously and must be sequenced. The first scheduling system does not allow preemption and all the tasks have unit processing times, while the second one allows preemption at integer time values and the processing times $p_j$ are assumed to be integer.

The paper is organized as follows. Section 2 recalls the needed notations and recalls the correspondence between scheduling systems and graph models. Saturated graphs, corresponding to scheduling systems with the MS property, are introduced and studied in section 3. Problem $P|fix_j, pmtn|\text{max}$ with preemption allowed at integer interval time, is analyzed in section 3.2 where sufficient