SASEPA: Simultaneous Allocation and Scheduling with Exclusion and Precedence Relations Algorithm

C. Fernández[1], F. Torres[2], and S.T. Puente[2]

[1] Engineering Department. Systems and Automation Division.
University Miguel Hernandez, Spain.
c.fernandez@umh.es

[2] Physics, Systems Engineering and Signal Theory Department.
University of Alicante, Spain.
{medina, spuente}@disc.ua.es

Abstract. An algorithm for allocating and scheduling tasks in multiprocessor environments is presented. Its main characteristic is its orientation towards machine vision applications. In this sense it deals with the peculiarities of systems which combine generic-type processors with Image Acquisition and Processing Boards. The main goal of the algorithm is total processing time reduction; such are the requirements when we deal with automated industrial inspection applications. By simultaneously tackling the phases of allocation and scheduling, the results obtained are better than those offered by traditional algorithms. The system is applied to a process of citrus fruit inspection, and its performances are also evaluated over randomly generated task graphs.

1 Introduction

Scheduling algorithms for machine vision inspection systems have specific requirements. The goal is to minimize the processing time making use of the available hardware (multiple processors of different kinds). Precedence and exclusion relations for the tasks must also be considered. It is difficult to find algorithms covering simultaneously all those aspects. Let us make a brief review of present techniques:

– EDF based schemes -like [1], [2], or PREC1 [3]- are not directly applicable when there are no deadlines. Apart from that, their main drawback is the lack of allocation strategies, as they are devoted to monoprocessor scheduling.
– Load balancing algorithms -like [4], [5], [6] or U.B.A. [3]- do not consider precedence relations; in the same way algorithms focused on precedence relations -like [7], [8], or [9]- do not perform allocations of tasks.
– There are also algorithms covering both allocation and scheduling, like [10] or A.P.R.A. [3], but they are mainly focused in networking and they are not applicable to most machine vision environments where synchronization tasks between two processors requires both of them to be available at the same time.

– Non-heuristic optimization techniques, like simulated annealing [11], [12], [13] do not consider task preemption, as they are mainly focused on job-shop scheduling.

2 Description of the SASEPA Algorithm

In scheduling problems two kinds of relation can be established between tasks, one related to the instant of their creation (precedence relation), and the other, according to whether they are able to block each other or not (exclusion relation) [14].

Some particularities can be found on machine vision systems regarding exclusion relations: generally, all the low level tasks, which are carried out in the Image Acquisition and Processing Boards (IAPB) are characterized by possessing exclusion relations, while on the other hand, high level tasks are usually carried out in generic processors without exclusion relations.

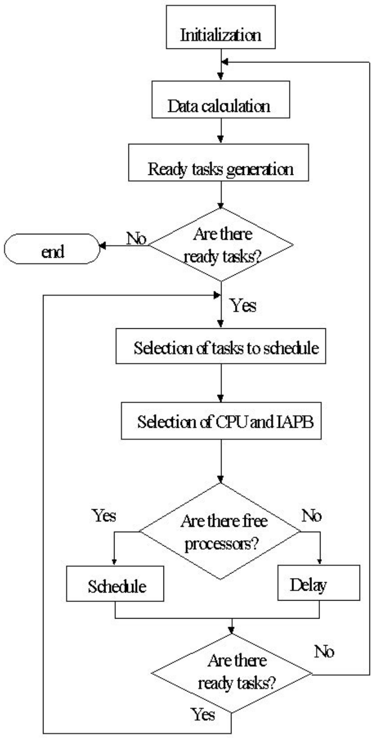

Fig. 1. Flow chart of the SASEPA algorithm

The proposed algorithm performs task scheduling in heterogeneous multiprocessor systems. Three distinct types of tasks have been considered [15]:

- **cpu**: High level, generic processing tasks. They can normally be preempted by other tasks of higher priority.
- **iapb**: Low level tasks carried out by a IAPB, like image acquisition or filtering. Depending on each particular situation, they can be preempted or not.
- **cpu/iapb**: Communication and synchronization tasks between CPU's and IAPB's. These tasks need the simultaneous availability of both processors. Moreover, they must be carried out in the processors implied in the communication (never in others, even though they are the same type). These tasks cannot be preempted.

The system to be scheduled consists of a group of n tasks with their corresponding precedence relations represented by means of a graph G, whose nodes are the distinct tasks and whose edges are the precedence relations existing between them.

The basic idea of the algorithm is to continuously select the task which will delay the system the most in order to schedule it in the first free processor, taking all the precedence relations into account. The steps followed by the algorithm are shown in the flowchart in Fig. 1. A detailed explanation of each module can be found in [16].

3 Performance Evaluation over an Example Application

A real application has been chosen as a benchmark for the proposed algorithm. It is a process of citrus fruit quality inspection by means of machine vision[1]. As Fig. 2 shows, there are 6 different operations to perform: image acquisition by four cameras surrounding the fruit; filtering; color, size and shape analysis; and classification.

The main interest will be to reduce to a minimum the total processing time. More details on this example application can be found in [16].

Fig. 3 shows the schedules obtained using the proposed algorithm and some other classical schemes:

- Critical path [7].
- PREC1 [3] (modified and adapted to deal with multiple processors).
- U.B.A. Utilization Balancing Algorithm [3] (just performs allocation).
- A.P.R.A. Allocation with Precedence Relations Algorithm [3].
- SASEPA.

The proposed algorithm offers the best results in terms of processing time -an optimal schedule is found for this example. Table 1 summarizes the results obtained.

[1] In order to obtain the task graph for the example application, a visual environment devoted to the design of machine vision systems has been used [17]

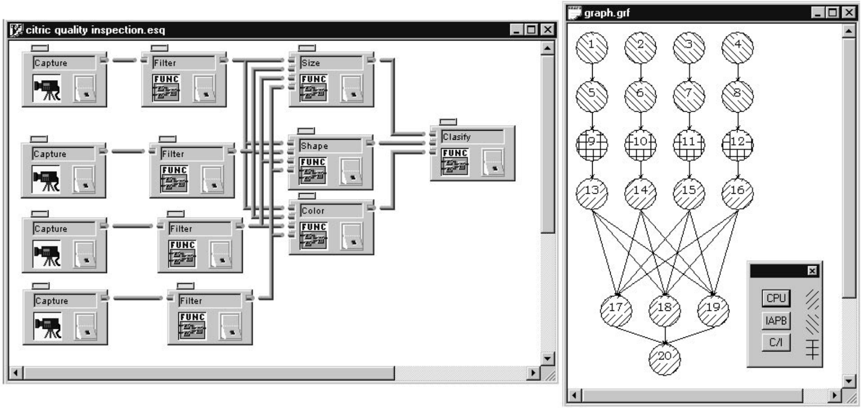

Fig. 2. The example application and its representation as a task graph

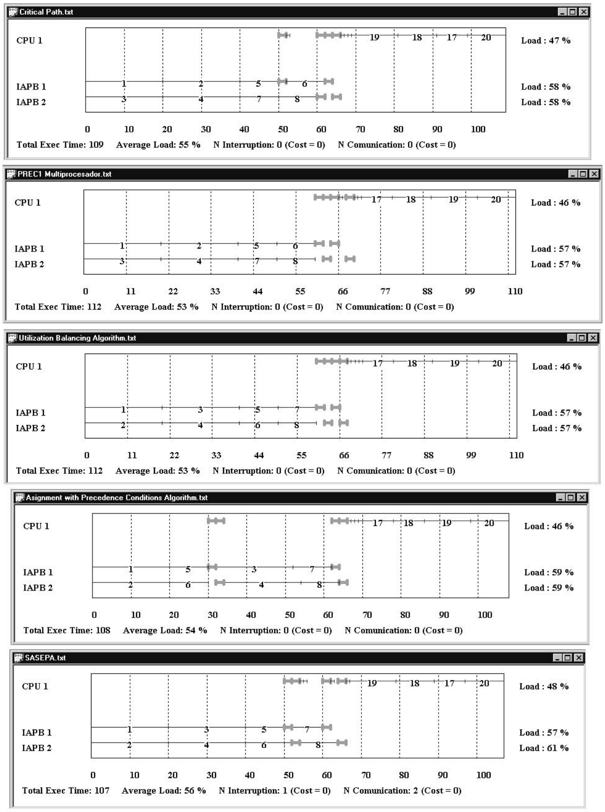

Fig. 3. Schedule comparison

Table 1. Experimental results

Algorithm	SASEPA	Critical Path	PREC1	U.B.A.	A.P.R.A.
Processing time	107	109	112	-	108
Load rate (average)	56%	55%	53%	-	55.5%
Tasks preempted	1	0	-	-	-
Load balance	4.8%	-	-	0%	0%
Communications	2	0	0	0	0

4 Performance Evaluation over Randomly Generated Task Graphs

In order to check the validity of the proposed approach over generic task graphs a second performance evaluation has been carried out. The scheduling capabilities of the SASEPA algorithm have been compared to those of the Critical path [7] and PREC1 [3] schemes over randomly generated examples.

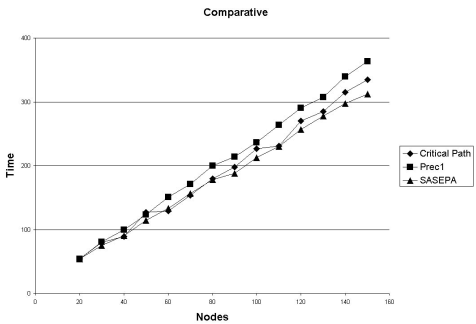

Fig. 4. Results over randomly generated task graphs

The graph complexity varies from 20 to 150 nodes. The averaged results are shown in Fig. 4. It can be seen that the schedules obtained by the proposed algorithm clearly outperform those obtained by the other two schemes.

5 Conclusions

Simultaneously considering the aspects of spatial allocation and time distribution improves scheduling results, as the range of possible solutions increases.

Moreover, the consideration of whether tasks can be preempted or not adds an extra degree of freedom at scheduling. The proposed algorithm covers all these points and is by no means computationally expensive.

The system is mainly set up for machine vision systems composed of generic IAPBs and CPUs and takes into account communications between both types of processors. A previous version of the algorithm has already been implemented in an industrial inspection system devoted to currency verification [17], [7].

References

1. Manabe Y, Aoyagi S: A Feasibility Decision Algorithm for Rate Monotonic and Deadline Monotonic Scheduling. Real-Time Systems. Vol. 14. (1998) 170–181
2. Audsley NC, Burns A.: On fixed priority scheduling, offsets and co-prime task periods. Information Processing Letters. Vol. 67. (1998) 65–69
3. Krishna CM, Shin KG.: Real-time Systems. McGraw-Hill. (1997)
4. Donk-Ik O, Baker TP.: Utilization Bounds for N-Processor Rate Monotone Scheduling with Static Processor Assignment. Real-Time Systems. Vol. 15. (1998) 183–192
5. Bampis E, Delorme C, Konig JC.: Optimal schedules for d-D grid graphs with communication delays. Parallel Computing. Vol. 24. (1998) 1653–1664
6. Campos LM, Scherson ID.: Rate of change load balancing in distributed and parallel systems. Parallel Computing. Vol. 26. (2000) 1213–1230
7. Sebastián JM, Torres F, Aracil R, Reinoso O, Jiménez LM, García D.: Job-Shop Scheduling applied to Computer Vision. Proc. SPIE. Vol. 3166. (1997) 158–169
8. Deogun JS, Kieckhafer RM, Krings AW.: Stability and Performance of List Scheduling With External Process Delays. Real-Time Systems. Vol. 15. (1998) 5–38
9. Kyunghee C, Gihyun J, Taegeun K, Seunhum J.: Real-time scheduling algorithm for minimizing maximum weighted error with O(N logN + cN) complexity. Information Processing Letters. Vol. 67. (1998) 311–315
10. Santos J, Ferro E, Orozco J, Cayssials R.: A Heuristic Approach to the Multitask-Multiprocessor Assignment Problem using the Empty-Slots Method and Rate Monotonic Scheduling. Real-Time Systems. Vol. 13. (1997) 167–199
11. Sadeh NM, Nakakuki Y.: Focused Simulted Annealing Search: An Application to Job Shop Scheduling. Annals of Operation Research. Vol. 60. (1996) 77–103
12. Albrecht A, Steinhofel K, Wong CK.: Distributed Simulated Annealing for Job Shop Scheduling. Schonauer et al. (eds): Parallel Problem Solving from Nature PPSN VI, Vol. 1917. Springer-Verlag (LNCS). (2000) 243–252
13. Takeshi Y, Ryohei N.: Job-Shop Scheduling by Simulated Annealing Combined with Deterministic Local Search. Information Processing Society of Japan (IPSJ) Journal. Vol. 37. nž 04–014 (1996)
14. Xu J, Parnas DL.: On Satisfying Timing Constraints in Hard-Real-Time Systems. IEEE Trans. On Software Engineering. Vol. 19. (1993) 70–84
15. Torres F.: Arquitectura paralela para el procesado de imágenes de alta resolución. Aplicación a la inspección de impresiones en tiempo real. PD, ETSII, Polytechnic University of Madrid. (1995)
16. Fernández C, Torres F, Puente ST.: SASEPA. Simultaneous allocation and scheduling with exclusion and precedence relations algorithm. Internal Report. Systems and Automation Division. Miguel Hernandez University. (2001)
17. Torres F, Candelas FA, Puente ST, Jiménez LM, Fernández C, Agulló RJ.: Computer Vision Systems - Simulation and Scheduling of Real-Time Computer Vision Algorithms. Lecture Notes in Computer Science. Vol 1542. (1999) 98–114