Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Energy Saving Task Scheduling for Heterogeneous CMP System Based on Multi-objective Fuzzy Genetic Algorithm

Lei Miao[1], Yong Qi[1] , Di Hou[1], Chang-li Wu[2] , Yue-hua Dai[1]
1. School of Electronics and Information Engineering Xi'an Jiaotong University
2. Chinese Society of Astronautics
Xi'an, China

***Abstract*—With the chip multi-processor (CMP) being more and more widespread used in the laptop, desktop and data center area, the power-performance scheduling issues are becoming challenges to the researchers. In this paper, we propose a multi-objective fuzzy genetic algorithm to optimize the energy saving scheduling tasks on heterogeneous CMP system. According to the characteristic of heterogeneous CMP system, we present a novel encoding and decoding scheme of genetic algorithm, improve the crossover operator and the mutation operator. Based on that, we improve the genetic algorithm architecture by using the relative fuzzy membership grade fitness and the elitist strategy. Simulation results demonstrate that using our algorithm can save both the execution time and system energy cost at the same time.**

***Keywords*—task scheduling, multi-objective optimal, heterogeneous chip multi-processor**

I. INTRODUCTION

Multi-processor on chip (CMP) has shown significant performance and power advantages over single cores in commercial systems. Applying a corollary of Moore's Law for CMP, integration of more cores on one chip is becoming a tendency in predictable future [1]. However, with the increment number of cores on chip and the processors in system, CMP systems encounter with significant power efficiency challenges. In order to handle trade-off between energy and the performance demand, we must maximize the efficiency of global energy utilization and minimize the system performance degradation.

Dynamic voltage scaling (DVS) [2] has been employed as an available technique to reduce the energy consumption of CMOS microprocessor system. In recent years, there were studies on DVS-based scheduling for CMP system energy reduction [3,4]. Allan Rae [3] discussed the method to allocate sub-tasks on processors with optimal voltage in heterogeneous multiprocessor system. Yuan Cai [4] proposed a dynamic voltage scaling procedure that changed load current profile by concerning the current workload. Most of the methods are not designed to reduce the execution time of task set by increasing the parallelism, while saving the system energy.

Genetic algorithm has been used by many researchers as an efficient method to solve the problem of task scheduling [5-7]. Hou , Ansari and Ren, [5] proposed pure genetic algorithm to schedule tasks in the multiprocessor system. To overcome the shortages of HRA, Correa R.C. presented FSG and CGL algorithm to improve the schedule [6]. Moreover, Yi Wen Zhong and Jian Gang Yang [7] used a task execution order list (TEOL) to solve the problem. Peng Yang and Chun Wong proposed a two-phase scheduling method to deal with the energy aware scheduling problem of heterogeneous CMP embedded system by using a parallel genetic algorithm (PGA) [8]. According to the characteristic of energy-aware scheduling task set on heterogeneous CMP system, we propose a novel coding scheme which can describe both the information of task allocating and operating voltage. We use relative fuzzy membership grade fitness to optimize both the execution time and energy consumption of task scheduling. We also improve the crossover and mutation operator of the genetic algorithm. We represent a multi-objective fuzzy genetic algorithm (MOFGA) which can minimize the execution time of tasks while reduce the system power consumption. Simulation results demonstrate that our algorithm can increase the efficiency of task scheduling on heterogeneous CMP, decrease the execution time and energy consumption of system.

The rest of this paper is organized as follows. The system model of task scheduling on heterogeneous CMP system is defined in section 2. In section 3, we introduce the coding scheme and the procedure of generating initial population. The genetic operators are presented in section 4. In section 5, we propose the multi-objective fussy genetic algorithm. The results, comparison and analysis of simulation experiment are shown in section 6. The conclusion is given in section 7.

II. MODEL AND DEFINITIONS

We will discuss the energy saving scheduling issue based on a heterogeneous CMP system in which each cores has its own L1 cache and all of them share one L2 cache. We assume the CMP system with n cores is represented as a set $P = \{ P_i \mid i \in [1,n] \}$. We suppose that an application task can be partitioned into a set of communicating sub-tasks $T = \{t_1, t_2, \ldots, t_m\}$, and the estimated execution time of each sub-task and the dependencies between them are known.

A. System Energy Model

In CMOS circuits [9], the operating frequency and the energy consumption are characterized by following functions.

$$f = \xi \cdot \frac{(V_{dd} - V_{th})^2}{V_{dd}} \quad (1)$$

$$E \approx N_{swt} \cdot C_o \cdot V_{dd}^{\ 2} \cdot N_c \quad (2)$$

Where ξ is a constant, V_{dd} is the supply voltage and V_{th} represents the system threshold voltage which is the lowest voltage supply, N_{swt} represents the number of switches per clock, C_o is the output capacitance, and f is the processor clock frequency.

As shown by the above two equations, the reduction in frequency combined with a linear reduction of operating voltage result in an approximately quadratic reduction of the energy consumption. As well known, the reduction of processor speed means the linear increase of time cost.

So we can reduce the system energy consumption by controlling the operating voltage of processor cores. We assume that the operating frequency can be scaled discretely at $r \in N$ levels, while $\mu_j \in [1, r]$ represents the operate voltage level of the core when sub-task t_j was processed on it. The set $\Lambda = \{\mu_1, \mu_2, \ldots, \mu_m\}$ represents the relative operating voltage of sub-task set T.

The total energy consumption of the system can be defined as the sum of energy cost for each sub-task:

$$E\_total = \sum_{j=1}^{m} E_j \quad (3)$$

Where E_j represents the energy cost of sub-task t_j.

B. Task Scheduling Model

According to [5], the task can be represented by a DAG $D = (T, E)$. The vertices denote the set of sub-tasks T and directed edges indicate the data dependencies between two sub-tasks. The direct edge $e_{ij} \in E$, $e_{ij} = (t_i, t_j)$ represents that sub-task t_j can't be executed until sub-task t_i finished. And the value of e_{ij} represents the amount of data transferred between these sub-tasks. We define sub-task t_i as the direct predecessor of t_j, and sub-task t_j as the direct successor of t_i. The sub-task without predecessor can be defined as start vertex and the sub-task without successor can be defined as end vertex. The element ε_{ij} belonging to $\Psi = \{\varepsilon_{11}, \varepsilon_{12}, \cdots, \varepsilon_{m(m-1)}, \varepsilon_{mm}\}$, represented the estimated execution time of sub-task t_j on processor core P_i. The set $\Omega = \{\sigma_{ij} \in \{1,0\} \mid i \in [1,n], j \in [1,m]\}$ represents the allocation relationship between the processor cores and the sub-tasks, $\sigma_{ij} = 1$ represents sub-task t_j will be executed on core P_i.

The beginning and finishing time of each sub-task is defined as

$$Finish[j] = \mu_j \varepsilon_{ij} + Begin[j] \quad (4)$$

$$Begin[j] = \max_{t_p \in Pre[t_j]} \{Finish[p] + (1 - \sigma_{ij}) e_{pj}\} \quad (5)$$

Where $Pre[t_j]$ represents the set of predecessor sub-task of t_j, $\mu_j \in [1, r]$ represents the operating voltage level of the core when sub-task t_j was processed on it.

In this paper we need to optimize the scheduling problem on heterogeneous CMP system to decrease both the execution time and energy consumption of system at the same time. We assume $OM1 = E_{total}$ and $OM2 = \max_{i=1}^{m} \{Finish[t_i]\}$. The energy saving scheduling issue can be as the following two-objective optimization problem.

$$Min\{E\_total, \max_{i=1}^{m} \{Finish[t_i]\}\}$$

III. Generate the Initial Population

A. Coding scheme

We propose a genetic coding scheme that denotes both the task allocating strategy and the information of voltage scaling. The coding scheme is indicated by a linear array *Chrom* with *2m* elements which is shown in the following figure 1.

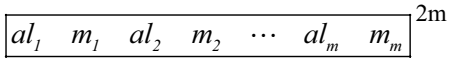

Figure 1. The coding scheme

Where the odd element of *Chrom* $al_i \in [1, n]$ represents the sub-task t_i is allocated on processor core P_{ali} and the even element of *Chrom* $\mu_i \in [1, r]$ represents the operating voltage level of the core P_{ali} when sub-task t_i is processed on it.

B. Decoding Scheme

We propose an effective decoding scheme to mapping the chromosome to the scheduling strategy. We can get the subtask set L_i the element of which is the subtask allocated to the processor core P_i. However the subtask set L_i can not satisfy the restrictions between sub-tasks that any sub-task would not run until every predecessors of it have been finished. We use the following method to create a sub-tasks sequence $List_i$ of P_i which satisfied the restrictions.

We describe the restrictions between sub-tasks by using the method of calculate the height value of every sub-tasks [5]. The height value of a sub-task in a DAG is defined as a random number range from $hp(t_i)$ to $hs(t_i)$.

$$Height(t_i) = Random(hp(t_i), hs(t_i)) \quad (6)$$

$$hp(t_i) = \begin{cases} 1, & if\ Pre(t_i) = \phi, \\ 1 + \max\limits_{t_j \in Pre(t_i)} hp(t_j), & otherwise. \end{cases} \quad (7)$$

$$hs(t_i) = \begin{cases} \max\limits_{k=1}^{m} hp(t_k), & if\ Succ(t_i) = \phi, \\ \min\limits_{t_j \in Succ(t_i)} hs(t_j) - 1, & otherwise. \end{cases} \quad (8)$$

Where $Pre(t_i)$ represents the set of predecessor sub-task of t_i and $Succ(t_i)$ represents the set of successor sub-task of t_i.

According to the ascending order of height value we arrange the every sub-tasks belonging to L_i into a sequence $List_i$

allocated to P_i. That is to say each subtask $t_{ij} \in List_i$ satisfy the following constraint.

$Height(t_{ij-1})) \leqslant Height(t_{ij}) \leqslant Height(t_{ij+1})$

C. Generate the Initial Population

The initial population is generated by the following step.

Algorithm 1.

1) Evaluate all the odd elements belonging to chromosome Chrom

al_i =Random(*1*,*n*) ;

2) Evaluate all the even elements belonging to chromosome Chrom

μ_i=Random(*1*, *r*)

3) Based on step 1 and step 2, the algorithm results in a new chromosome.

4) If not reaching the size of population, go to the step 1.

Else the initialization finished.

IV. Genetic Operator

A. Relative Fuzzy Membership Grade Fitness

During the procedure of genetic algorithm evolution, we need to calculate fitness value for every chromosome belonging to the new generation. Usually the fitness value of each chromosome is the only factor to evaluate the quality of it. In order to optimize both the execution time and energy consumption of task scheduling, we take the relative fuzzy membership grade as the fitness in this paper. By using the relative fuzzy membership grade we can obtain a fuzzy evaluation sequence of chromosomes in the population [10]. The chromosomes with the higher relative fuzzy membership grade would have high possibility to be selected into the next generation.

We calculate the expected execution time and energy consumption of each task scheduling solutions (i.e., the chromosomes) in the population. The objective of CMP system energy consumption of the solution $Chrom_i$ is as following shown:

$$OM_1^j = \max_{k=1}^{G} E\_total^k - E\_total^j , \quad j = 1,2,\cdots,G \tag{9}$$

The objective of task execution time of the solution $Chrom_i$ is as following shown:

$$OM_2^j = \max_{k=1}^{G} FT^k - FT^j , \quad j = 1,2,\cdots,G \tag{10}$$

Where $FT^j = \max_{s=1}^{m}\{ Finish[t_s]\}^j$ represents the task execution time of the solution $Chrom_i$. The minimum and maximum values of these two objectives are as following:

$$\begin{cases} OM_{imax} = \max\limits_{i=1}^{G} OM_{ji}^j & i = 1,2 \quad j = 1,2,\cdots,G \\ OM_{imin} = \min\limits_{i=1}^{G} OM_i^j & i = 1,2 \quad j = 1,2,\cdots,G \end{cases} \tag{11}$$

According to these minimum and maximum values, we can calculate the linear interpolation evaluate function for solution $Chrom_i$. We assume that the minimum values of these two objectives correspond to *N* and the maximum values of these two objectives correspond to *1*.

$$\begin{aligned} FMG_i^j &= N\frac{OM_i^j - OM_{imax}}{OM_{imin} - OM_{imax}} + \frac{OM_i^j - OM_{imin}}{OM_{imax} - OM_{imin}} \\ &= N - (N-1)\frac{OM_i^j - OM_{imin}}{OM_{imax} - OM_{imin}} \end{aligned} \tag{12}$$

Where $i = 1,2$ and $j = 1,2,\ldots,G$.

The relative fuzzy membership grade fitness for solution $Chrom_i$ is shown as the following equation.

$$FMG^j = \frac{1}{2N}\sum_{i=1}^{2} w_i FMG_i^j \tag{13}$$

Where w_1 and w_2 are random weight values determined by the following function [11].

$$w_k = \frac{r_k}{r_1 + r_2} \quad k = 1,2 \tag{14}$$

Where r_1 and r_2 are random natural numbers which range from 1 to 100.

B. Crossover Operator

Our crossover operator is summarized in the following algorithm 2:

Algorithm 2.

1) Selects one pair of chromosomes as parents from the population with the possibility P_{cro};

2) Selects the crossover point $Point_{cro}$ stochastically;

3) Apply a point crossover operator to the parents: exchange the elements after the crossover point belonging to the parents.

4) Based on above steps, the crossover operator results in a new chromosome.

C. Mutation operator

Our mutation operator is summarized in the following algorithm3:

Algorithm 3.

1) Selects one chromosome as parent from the population with the possibility P_{mut};

2) Selects the mutation point $Point_{mut}$ stochastically;

3) if $Point_{mut}$ is a even number

$\mu_j = \overline{\mu_j}$, where $\overline{\mu_j} \in$ VL - $\{\mu_j\}$, $VL = \{1,2, \ldots ,r\}$;

Else

$al_j = \overline{al_j}$, where $\overline{al_j} = AL - \{al_j\}$, $AL = \{1,2, \ldots ,n\}$;

3) Based on above steps, the mutation operator produces a new chromosome.

V. The Genetic Algorithm Structure

According to the elitist strategy from reference [11], we transform some chromosomes into the next generation. During the procedure of evolution we keep a Pareto set and update it for every generation. The Pareto set includes chromosomes with the shortest task execution time, lowest system energy

cost chromosomes and chromosomes randomly selected from generation. We assume that the population size is G and $3N_e$ is the number of elite solutions. The main steps of our multi-objective fuzzy genetic algorithm (MOFGA) can be summarized as the following algorithm 4.

Algorithm 4.

1) Calculate the height value for each sub-task, generate an initial population *Group(0)* with *G* chromosomes, select *3Ne* chromosomes randomly from generation into the elite group *Ge(0)*, *i=0*

2) select *G* pair of chromosomes randomly from *Group(i)* to apply crossover operation, from which obtain *G* successor chromosomes; add them into Group(i).

3）Calculate the values of *OM1*, *OM2*and *FMG* for each chromosome in the current population.

4）Transfer *G-3Ne* chromosomes with the larger *FMG* belonging to *Group(i)* and all chromosomes belonging to *Ge(i)* into the next generation *Group(i+1)*

5) Select *Ne* chromosomes with the larger *OM1*, *Ne* chromosomes with the larger *OM2* and *Ne* chromosomes randomly from generation into the elite group *Ge(i+1)*;

6）Apply mutation operation in the *Group(i+1)* , *i++*;

7) If one of the following conditions is satisfied, end the algorithm output the best solution:

a) $i > N_{max}$, where N_{max} is the maximum iteration number.

b) There has no change about the best individual during the nearest 10 iterations.

Else, return to 2).

VI. SIMULATION RESULTS

We established a simulation environment to verify and evaluate our scheduling optimal algorithm. The simulator is built up with two parts: the task set generator and the implementation of our multi-objective fuzzy genetic algorithm.

In this paper, we design a task generator which randomly generates DAG in accordance with the following conditions:

1) The number of sub-task is a random natural number from 20 to 200.

2) The estimated execution time of sub-tasks is a random natural number from 10 to 100.

3) Each vertex has 0-3 successors.

4) The weight of direct edge is a random natural number from 10 to 40.

We simulate the implement of the same task set which produced from the generator using following two methods: PAG [5], and MOFGA algorithm. The simulation results for task execution time of PAG, MOFGA and normal genetic algorithm (GA for short) scheduling 10 kinds of task set on CMP system having 8 heterogeneous processor cores are shown in table 1.

TABLE I. COMPARE THE EXECUTION TIME AND ENERGY COST OF OPTIMAL SCHEDULING FOR THREE METHODS

Task Set	Execution Time				Energy Cost			
	GA	PAG	MOF GA	TSR (%)	GA	PAG	MOF GA	ESR (%)
1	926	874	736	27.2	4656	4176	3964	31.4
2	1862	1747	1582	31.7	8567	7852	7143	34.1
3	2857	2594	2287	32.3	15359	13847	12754	37.6
4	1583	1274	1256	34.8	7851	7369	6783	35.7
5	1163	1037	964	25.6	6372	5844	5278	33.5
6	1635	1485	1263	34.5	8249	7835	6915	35.2
7	1748	1498	1382	30.9	8936	8134	7437	32.9
8	2167	1982	1734	32.2	11638	10549	9672	36.5

Where $TSR=(T_{NONE}-T_{MOFGA})/T_{NONE}$ and $ESR=(E_{NONE}-E_{MOFGA})/E_{NONE}$ separately represent the ratio of time and the ratio of energy reduced by MOFGA compared to the scheduling without any optimal method.

Table 1 summarized our key results of task execution time and the system energy cost of GA, PAG and MOFGA. The MOFGA has the shortest execution time of task set. Compared with the scheduling without optimal, our MOFGA can reduce 25.5-34.8% execution time of task set. Moreover, the heterogeneous CMP system would consume least energy by using our method. Compared with the scheduling without optimal, our algorithm saves 31.4-36.5% system energy consumption at the same condition.

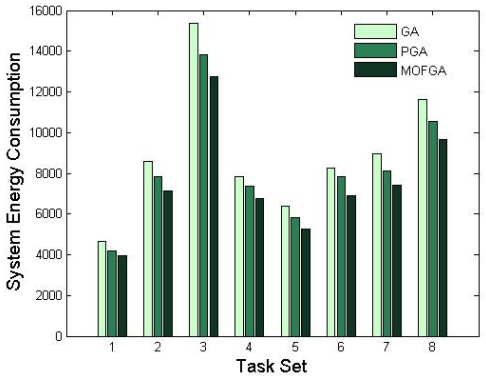

Figure 2. Execution time of GA, PGA and MOFGA

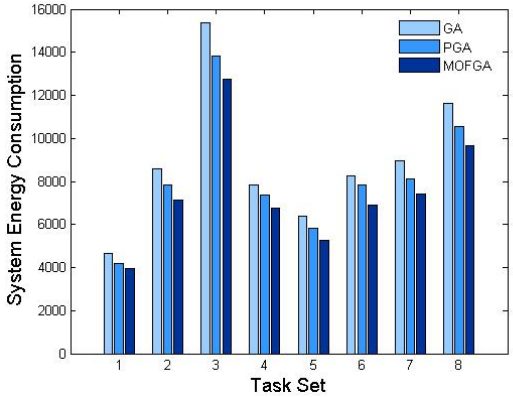

Figure 3. System energy consumption of GA, PGA and MOFGA

Figure 2 and figure 3 show the comparison for the execution time of task set and the system energy consumption of using GA, PAG and MOFGA to schedule various task set on a heterogeneous CMP system with 8 cores.

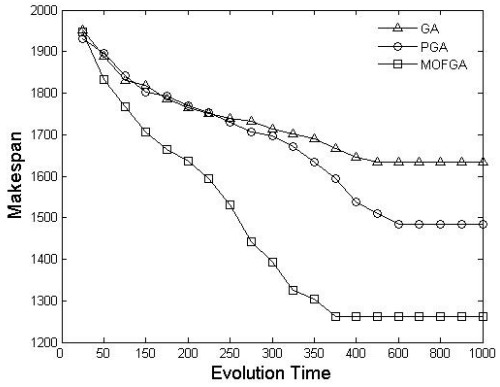

Figure 4. Compare of evolution convergence for GA, PGA and MOFGA

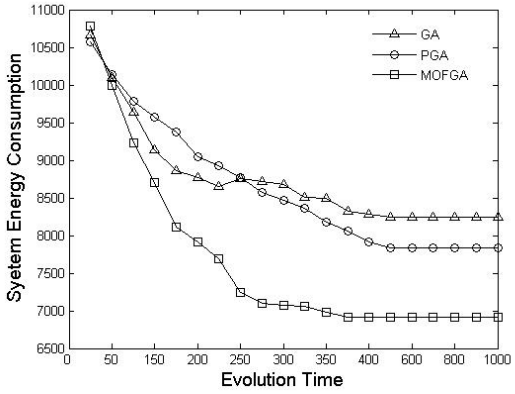

Figure 5. Compare of evolution convergence for GA, PGA and MOFGA

Figure 4 and figure 5 show the execution time and the energy cost of Task Set 5 for the best solution of GA, PAG and MOFGA, during the evolution iteration range from 50 to 1000. From these two graphs, we can see that MOFGA has the quickest velocity of convergence and the best results of both execution time and energy cost.

From the simulation results given above we can draw the conclusion that compared with the GA and PAG, MOFGA have remarkable advantages in solving the problem of power-performance in heterogeneous CMP systems. MOFGA cut the total execution time of task set down through the usage of coding scheme, improved genetic operators and the structure of algorithm. MOFGA reduce the system energy cost by adjusting the operating voltage of the active cores. Simulation results demonstrate that using our algorithm can decrease the execution time and energy consumption of system efficiently.

VII. CONCLUSION

This paper proposes a multi-objective fussy genetic algorithm to optimize the problem of energy saving task scheduling on heterogeneous CMP system. According to the analyzing of heterogeneous CMP system characteristics, we represent a novel coding scheme of genetic algorithm. Based on that, we improve the crossover operator, mutation operator and the structure of genetic algorithm. We introduce the relative fuzzy membership grade fitness and the elitist strategy in genetic algorithm to avoid local convergence. The experiment results show that our algorithm can save energy for 31.4-36.5% and reduce 25.5-34.8% execution time of task set at the same time. In short, using our algorithm can increase the efficiency of energy saving task scheduling on heterogeneous CMP system.

REFERENCES

[1] A. Agarwal and M. Levy. Thousand-core chips: the kill rule for multicore. In Proceedings of the 44th Annual Conference on Design Automation, pages 750–753, 2007.

[2] T. D. Burd, T. A. Perimg, A. J. Stratakos, and R. W. Brodersen, “A Dynamic Voltage Scaled Microprocessor System”, IEEE Journal of solid-state Circuits, pp.1571- 1580, November 2000.

[3] Rae A. and Parameswaran S., “Voltage reduction of application-specific heterogeneous multiprocessor systems for power minimisation”, IEEE Proceedings of the ASP-DAC 2000. Asia and South Pacific, pp. 147 – 152, 2000.

[4] Yuan Cai, Reddy, S.M., Pomeranz, I., Al-Hashimi, B.M.. Battery-aware dynamic voltage scaling in multiprocessor embedded system Circuits and Systems, IEEE Proceedings of ISCAS 2005, pp. 616 – 619, May 2005.

[5] E Hou, N Ansari, and H Ren., A Genetic Algorithm for Multiprocessor Scheduling, IEEE Trans Parallel and Distributed Systems, pp. 113-120, February 1994.

[6] Correa, R.C. Ferreira, A. Rebreyend, P., Scheduling Multiprocessor Tasks With Genetic Algorithms, IEEE Transactions on Parallel and Distributed Systems, Volume 10, Issue 8, pp. 825 – 837, August 1999.

[7] Yi Wen, Zhong Jian and Gang Yang, A Genetic Algorithm for Tasks Scheduling in Parallel Multiprocessor Systems, IEEE Proceedings of Machine Learning and Cybernetics, Volume 3, pp. 1785-1790, 2003.

[8] Peng Yang, Chun Wong, Energy-Aware Runtime Scheduling for Embedded Multiprocessor SOCs. IEEE Design&Test of Computers, pp. 46-58, 2001.

[9] Sung-Mo Kang and Yusuf Leblebici, CMOS Digital Integrated Circuits: Analysis and Design, McGraw-Hill, 2002.

[10] Sakawa, Masatoshi, “Genetic Algorithms and Fuzzy Multiobjective Optimization”, Operations Research/Computer Science Interfaces Series , Vol. 14, Springer, 2001

[11] Ishibuchi, H. and Murata, T., “A multi-objective genetic local search algorithm and its application to flowshop scheduling”, IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, IEEE, Volume 28, Issue 3, pp. 392 – 403, Aug 1998.