TECHNICAL REPORT
of UEC, UEC-TR-C5(1), 1988

A Simple Algorithm for Finding a Maximum Clique*

Etsuji Tomita   Yasuhiro Kohata   Haruhisa Takahashi

Department of
Communications and Systems

THE UNIVERSITY
OF
ELECTRO-COMMUNICATIONS

Chofugaoka, Chofu, TOKYO 182 JAPAN

1. Introduction.

We present a very simple branch and bound recursive algorithm for finding a maximum clique in a given graph G=(V,E) with vertex set V and edge set E. The approach that we describe in this paper uses a prunning method based on "NUMBERING" in which the upper bound of a maximal clique under expansion is evaluated. Efficiency of the algorithm is experimentally evaluated for several graphs with up to 560 vertices. Earlier works on finding a maximum independent set or a maximum clique include Tarjan and Trojanowski [TT], and Balas and Yu [BY]. We implemented and tested both of them.

The readers are advised to refer to [TTT, Section 2] for our definitions and notation. In addition, we denote by $\omega(G)$ the number of vertices of a maximum clique in graph G.

2. An algorithm.

The *basic* framework of our algorithm is as in [TTT, Section 3]. In addition to a global variable Q of a set of vertices which constitute a complete subgraph as in [TTT], we introduce another global variable Qmax which is a maximum clique having been found.

We begin the algorithm MCLIQ by letting $Q:=\phi$; $Qmax:=\phi$, and extend Q step by step by applying a recursive procedure EXTEND to V and its succeeding induced subgraphs searching for larger and larger complete subgraphs until they reach maximals.

Let $Q=\{p_1, p_2, \cdots, p_d\}$ be found to be a complete subgraph at some stage, and consider a subgraph G(SUBG) which is induced by a set of vertices

$$SUBG=V\cap\Gamma(p_1)\cap\Gamma(p_2)\cap\cdots\cap\Gamma(p_d)$$

where SUBG=V when $Q=\phi$ at the initial stage. Then apply the procedure EXTEND to SUBG searching for larger complete subgraphs. If $SUBG=\phi$ then Q is clearly a *maximal* complete subgraph, i.e., a clique. So if $|Qmax| < |Q|$ then we let Qmax:=Q. If $SUBG\neq\phi$, $Q\cup\{q\}$ becomes a larger complete subgraph for every $q\in SUBG$. Then consider smaller subgraphs $G(SUBG_q)$ which are induced by new sets of vertices

$$SUBG_q = SUBG\cap\Gamma(q)$$

for all $q\in SUBG$, and apply recursively the same procedure EXTEND to $SUBG_q$ to find larger complete subgraphs containing $Q\cup\{q\}$.

In principle, we can obtain a maximum clique Qmax of the given graph G=(V,E) after completing all the searchings stated above. Now the problem of efficiency consideration is how to prune unnecessary searchings that cannot lead to a maximum clique.

Note here for $Q=\{p_1, p_2, \cdots, p_d\}$ and for $SUBG=V\cap\Gamma(P_1)\cap\Gamma(P_2)\cap\cdots\cap\Gamma(P_d)$ given above, that

$$\omega(Q\cup SUBG)=|Q|+\omega(SUBG).$$

In addition, note that if we can confirm in advance that $|Q|+\omega(SUBG)\leqq|Qmax|$ holds for $SUBG\neq\phi$, then it is no longer necessary to continue searching for any $q\in SUBG$.

Then in order to know an upper bound of $\omega(SUBG)$, we give in advance for each $p\in SUBG$ a positive integer "Number" No[p] with the following property :

(i) If $\Gamma(p)\ni r$ then $No[p]\neq No[r]$; and

(ii) No[p]=1 or else if No[p]=k>1, then there must exist vertices $p_1\in\Gamma(p)$, $p_2\in\Gamma(p)$, $\cdots$, $p_{k-1}\in\Gamma(p)$ with $No[p_1]=1$, $No[p_2]=2$, $\cdots$, $No[p_{k-1}]=k-1$.

(Such "Numbers" as above were first applied to the maximum clique finding problem in [TY].)

Consequently, we know that

$$\omega\,(\mathrm{SUBG}) \leqq \mathrm{Max}\{\mathrm{No}[p] \mid p \in \mathrm{SUBG}\},$$

and hence if $|Q| + \mathrm{Max}\{\mathrm{No}[p] \mid p \in \mathrm{SUBG}\} \leqq |Q\mathrm{max}|$ holds then we can disregard such SUBG.

The above mentioned No[p] for every $p \in$ SUBG can be easily given step by step by a so-called following greedy coloring algorithm: Assume the vertices of SUBG=$\{p_1, p_2, \cdots, p_m\}$ are arranged in this order. First let No[p_1]=1. Subsequently, let No[p_2]=2 if $p_2 \in \Gamma[p_1]$ else No[p_2]=1, $\cdots$ and so on. After Numbers are given to all vertices of SUBG, we arrange these vertices in the nondecreasing order with respect to their just given Numbers. We call such a numbering procedure as NUMBERING(SUBG, No). This algorithm can be implemented to work in $O(|\mathrm{SUBG}|^2)$ time.

Notice that $\mathrm{Max}\{\mathrm{No}[p] \mid p \in \mathrm{SUBG}\}$ is not unique and deppends on the order of verteces in SUBG. It is important especially at the initial stage in which SUBG=V to select the order of vertices in SUBG so as to make $\mathrm{Max}\{\mathrm{No}[p] \mid p \in V\}$ as small as possible. Then prior to applying NUMBERING(V, No), we arrange vertices of V in the nonincreasing order with respect to their degrees.

The above numbering method is used in our algorithm as follows : When SUBG has been numbered by NUMBERING(SUBG, No), the vertices of $\mathrm{SUBG}_q = \mathrm{SUBG} \cap \Gamma(q) (\subseteq \mathrm{SUBG})$ are arranged in the same order as in SUBG. Then new "Numbers" are given to $\mathrm{SUBG}_q (\neq \phi)$ by NUMBERING(SUBG_q, NewNo) according to this ordering.

Now the whole algorithm is described as follows.

```
procedure MCLIQ(G)
        {Graph G=(V,E)}
begin
  Q:=$\phi$   {global variable Q is to constitute a complete subgraph}
  Qmax:=$\phi$   {global variable Qmax is a maximum clique having been found}
  arrange the vertices of V in nonincreasing order
          with respect to their degrees
  NUMBERING(V,No)
  EXTEND(V, No)
  procedure EXTEND(SUBG, No)
  begin
    while SUBG$\neq \phi$
       do q:=vertex in SUBG such that No(q) is the greatest
          if $|Q|$+No(q)$\leqq |Qmax|$
             then {$\omega$ (Q$\cup$SUBG)$\leqq |Qmax|$ } exit fi
          Q:=Q$\cup$ {q}
          SUBG$_q$:=SUBG$\cap \Gamma$ (q)
          if SUBG$_q$ =$\phi$
             then {Q is a clique}
                  if $|Qmax| < |Q|$ then Qmax:=Q fi
             else NUMBERING(SUBG$_q$, NewNo)
                  EXTEND(SUBG$_q$, NewNo)
          fi
          SUBG:=SUBG-{q}
          Q:=Q-{q}
       od
  end {of EXTEND}
end {of MCLIQ}
```

3. Experimental evaluation.

The algorithm has been implemented in Pascal and run on SONY NEWS-830 (68020 ,16MHz). For purposes of comparison, we have also implemented the elaborate algorithm of Balas and Yu [BY] (TC4) which is shown to be very efficient. The computational results listed below consist of two parts. First, the CPU-times are tested on randomly generated graphs with densities ξ (i.e. probability of presence of a given edge) 0.15, 0.25, 0.50, and 0.75 with the number of verteces ranging from 40 to 560. The results are shown in Fig.1. Next, the CPU-times are tested also on randomly generated graphs with 120 and 240 verteces, of density ranging from 0.10 to 0.90 and 0.1 to 0.75, respectively. The results are shown in Fig.2.

The experimental results show that our much simpler algorithm is comparalbe to or more efficient than Balas and Yu [BY], especially for graphs with smaller density. The experiments (Fig.2) show that the smaller density, the better efficiency in CPU times. We also tentatively compared our algorithm with that of Tarjan and Trojanowski [TT] coded in Pascal on a large computer HITAC M-260D and confirmed the better efficiency of our algorithm for graphs in our experiments.

4. A concluding remark.

This *simple* algorithm MCLIQ can be made more efficient by appropriately controlling the application of NUMBERING. A part of results in this direction appears in [NTTT].

Acknowledgements. The authors wish to express their gratitude to Mikio Shindo, Masaya Nagai, and Suguru Tabuchi for their contributions in an early stage of this work. They also would like to acknowledge useful discussions with Prof. Takao Nishizaki of Tohoku University.

REFERENCES

[BY] E.BALAS AND C.S.YU, "Finding a maximum clique in an arbitrary graph", SIAM J.Comput.,15(1986),pp.1054-1068.

[NTTT]M.NAGAI,T.TABUCHI,E.TOMITA AND H.TAKAHASHI, "An experimental evaluation of some algorithms for finding a maximum clique",Nat. Conv. Rec. IEICE Japan D-348(1988) (in Japanese)

[TT] R.E.TARJAN AND A.E.TROJANOWSKI, "Finding a maximum independent set", SIAM J.COMPUT., 6(1977), pp.537-546.

[TTT] E.TOMITA, A.TANAKA AND H.TAKAHASHI, "The worst-case time complexity for finding all the cliques", Technical Report,UEC-TR-C5,1988.

[TY] E.TOMITA AND M.YAMADA, "An algorithm for finding a maximum complete subgraph", Nat. Conv. Rec, IECE Japan 8 (1978) (in Japanese)

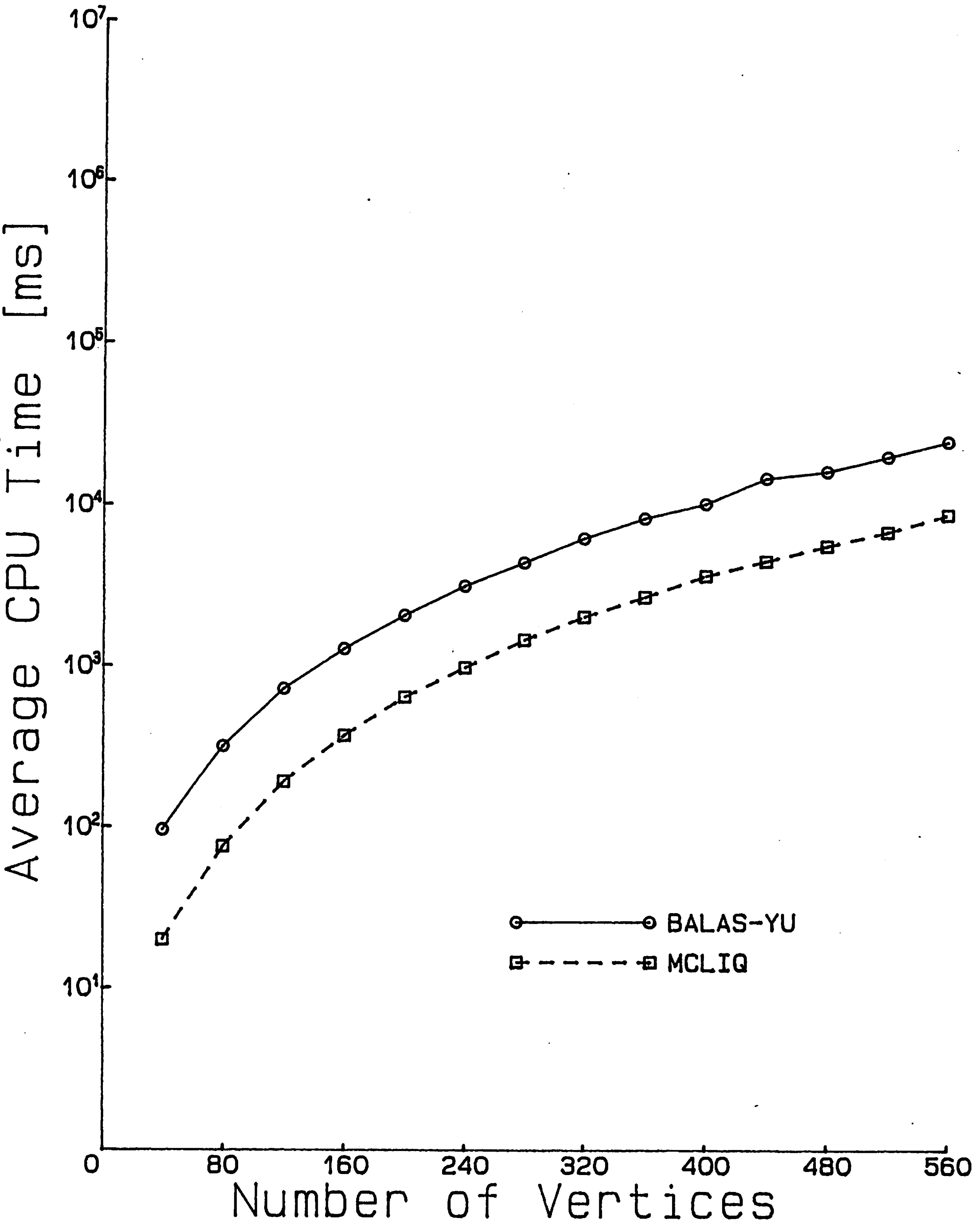

Fig. 1 (A) Random Graphs of Density ξ = 0.15

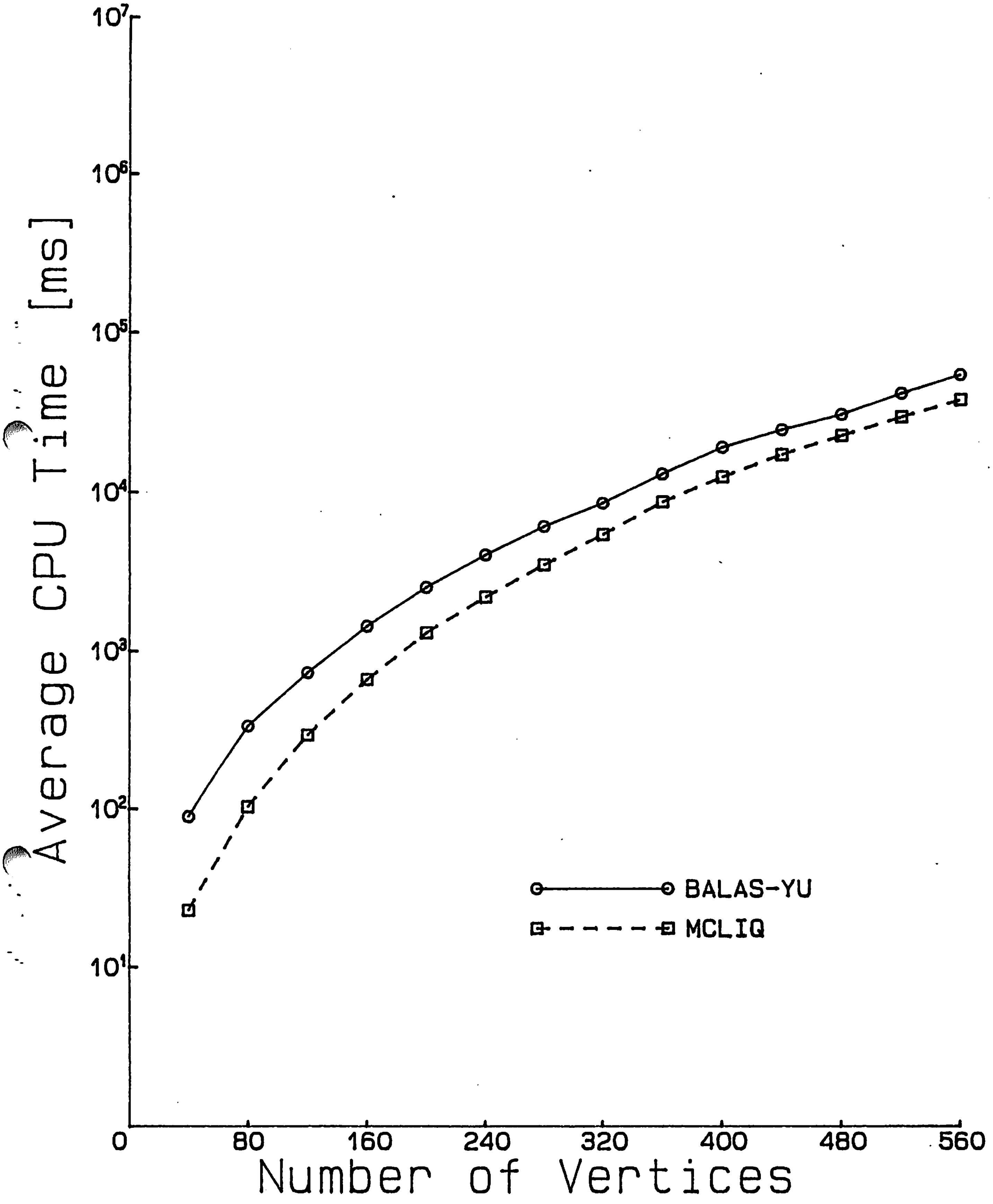

Fig. 1 (B) Random Graphs of Density δ = 0.25

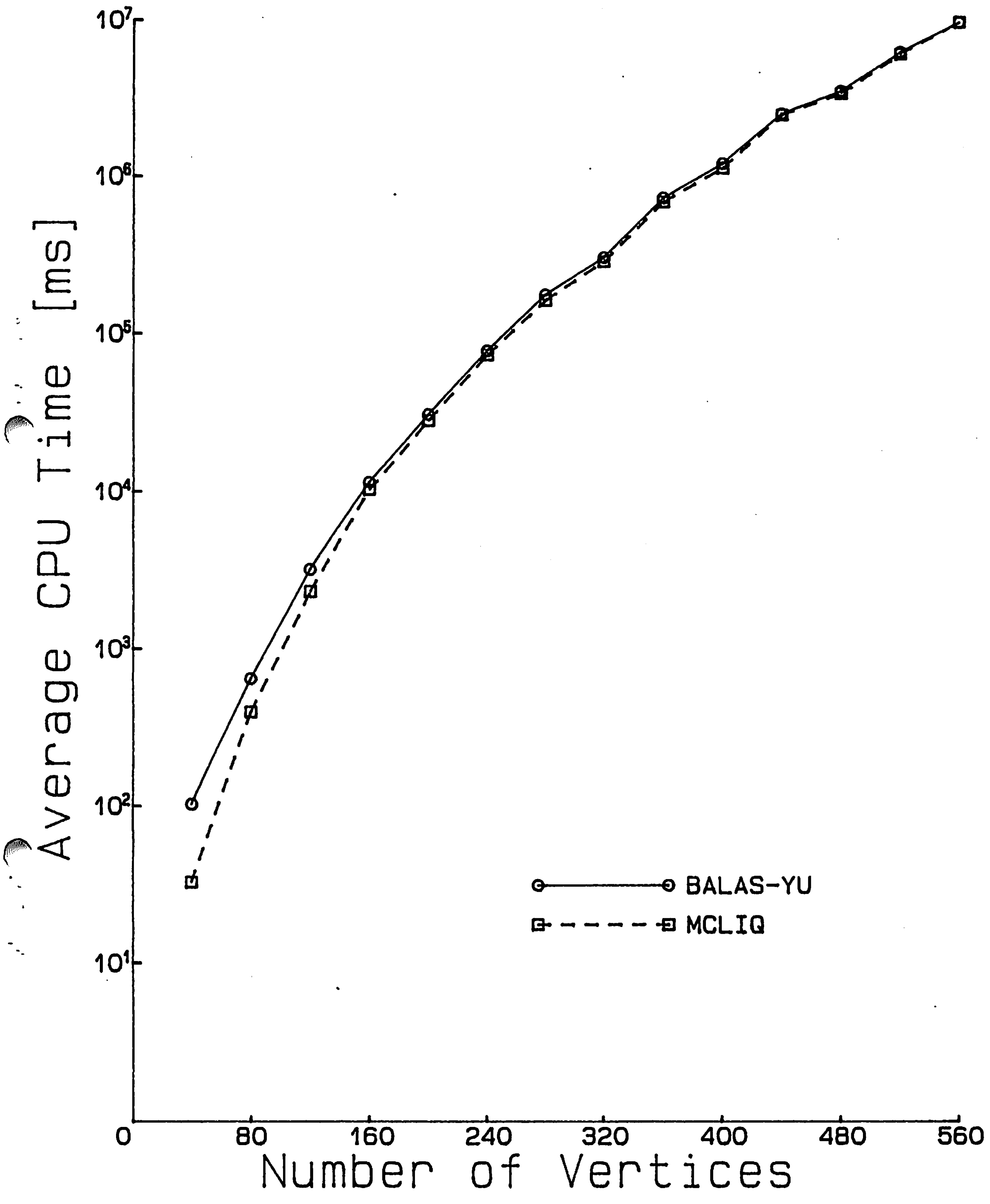

Fig. 1 (C) Random Graphs of Density ξ = 0.50

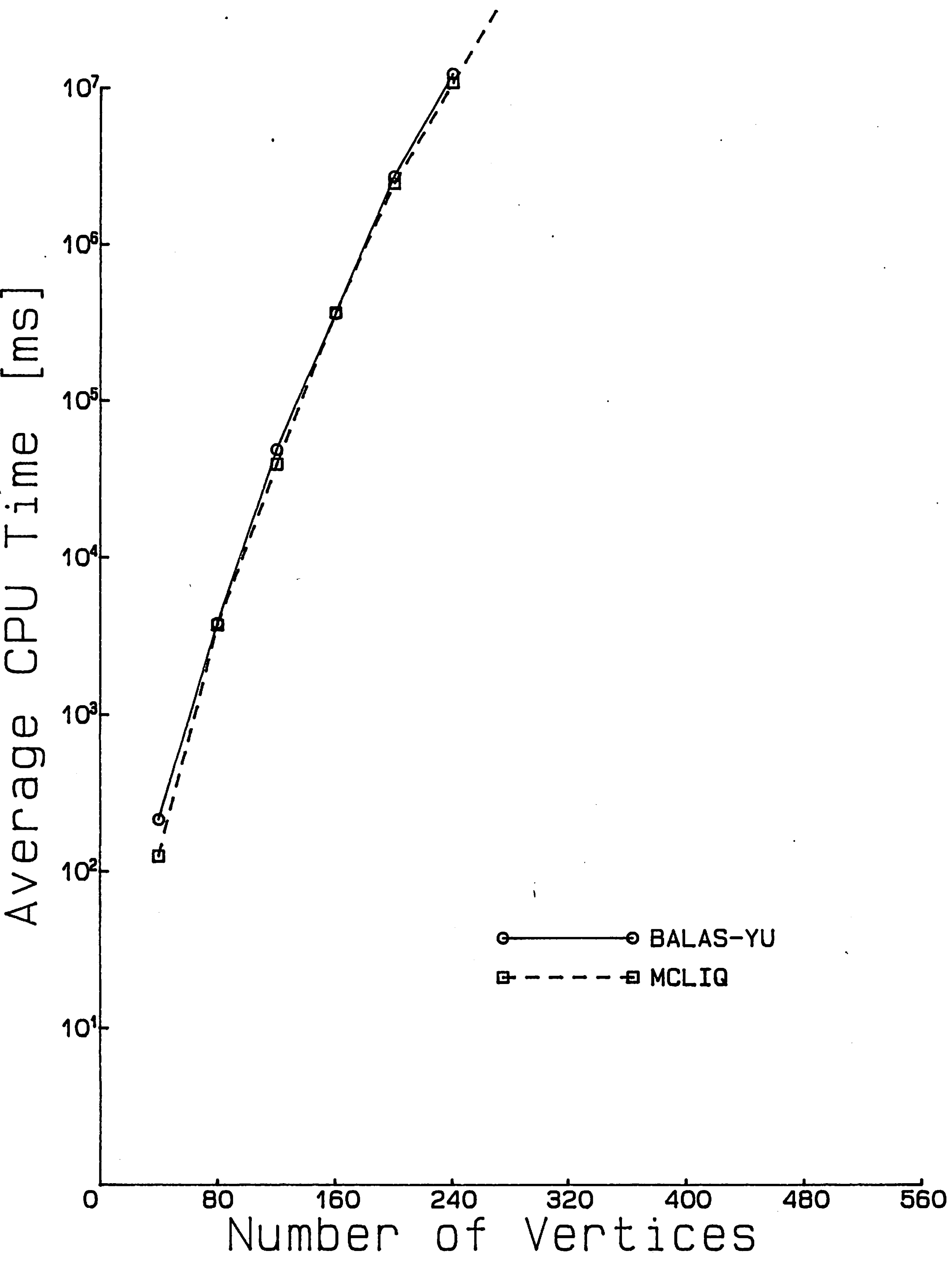

Fig.1(D) Random Graphs of Density ξ = 0.75

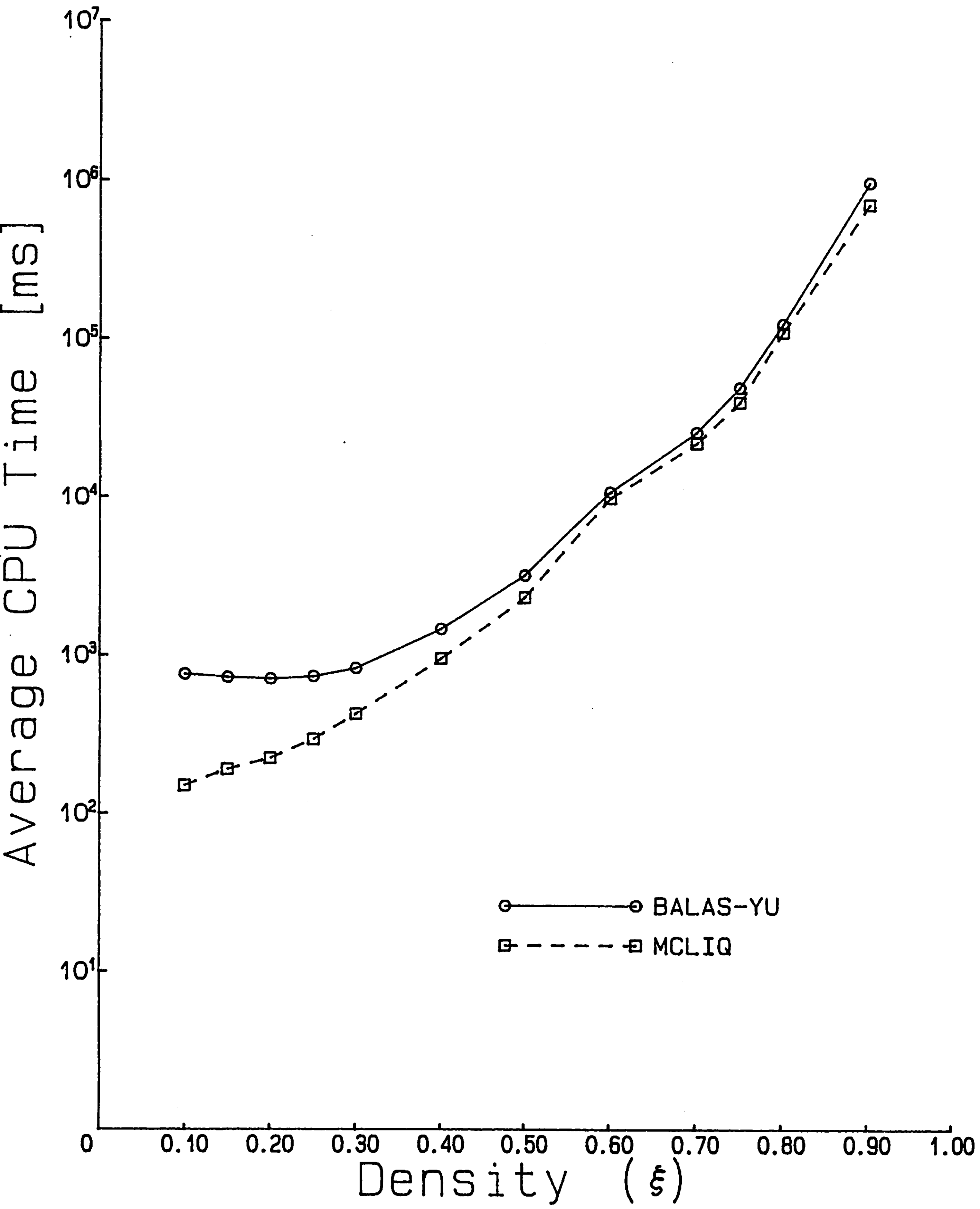

Fig.2 (A) Random Graphs of Vertices = 120

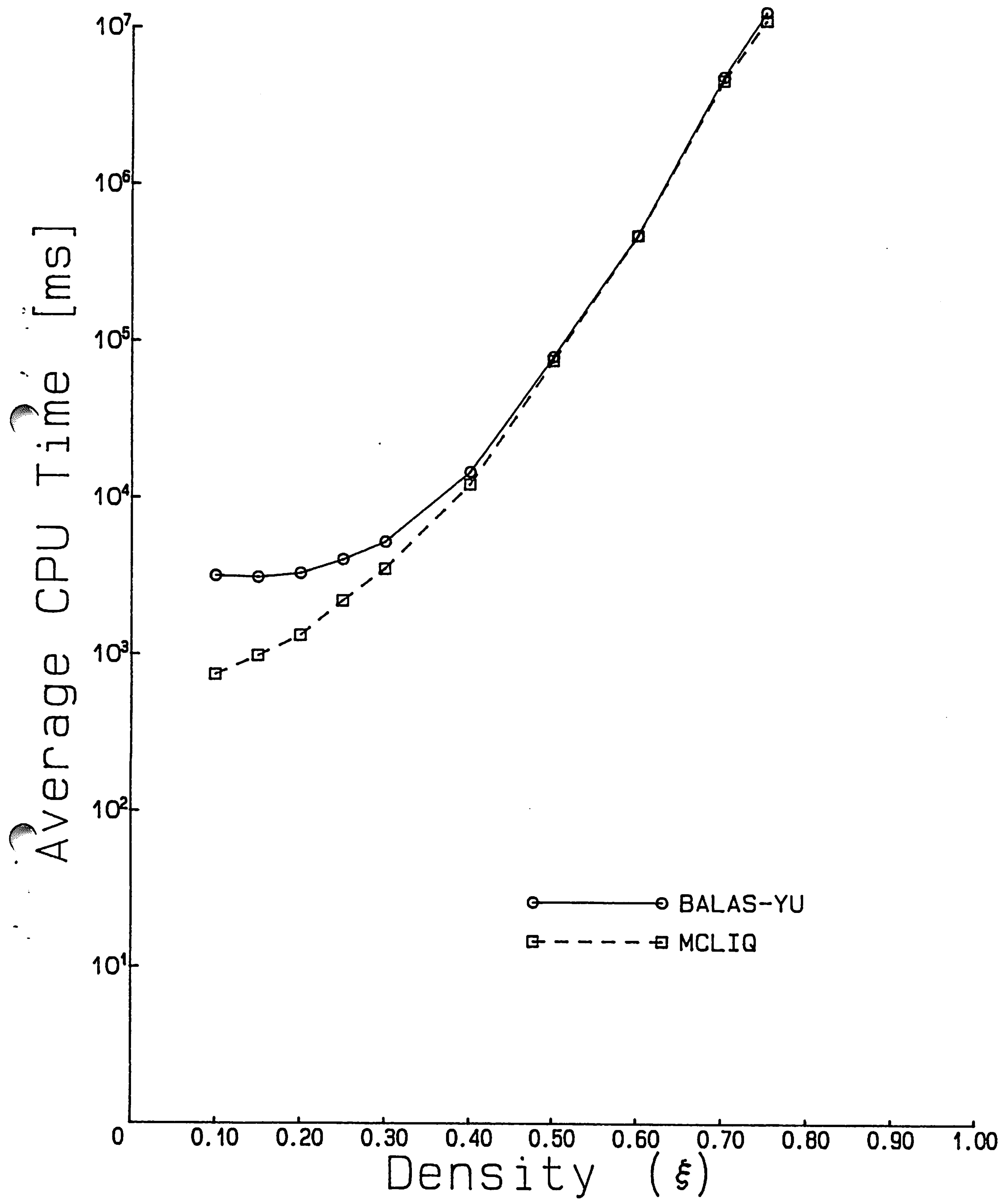

Fig.2(B) Random Graphs of Vertices = 240