A MONTE CARLO STUDY OF SIMULATION LIST ALGORITHMS

V. L. Kapil and V. G. Ashar
IBM System Products Division
East Fishkill Facility
Hopewell Junction, New York 12533

ABSTRACT

With the increased capability of modern simulation languages, more complex models are being structured to describe highly interactive, dynamic systems. This has resulted in an almost geometric increase in the computer time required to execute these models when the number of scheduled events (or processes) in the future scheduling list becomes abnormally high. This is especially true in the case of process-oriented languages such as SIMPL/I.

This paper presents the results obtained from a comparative study of several well-known scheduling strategies in the context of useful test cases and of the operational environment of a typical large system. The test cases comprise a set of stochastic distributional patterns which are of particular interest in the simulation of queuing and reliability systems. Both the linear and the stratified list types included in the study combine with sequential or binary search techniques to formulate alternative scheduling strategies.

The comparison of various strategies is focused on the empirical distributions of the number of scans required for m insertions over a period of time under different operating conditions.

In this study, SIMPL/I is used as the simulation vehicle, although the conclusions are, in general, expected to be valid for other process-oriented languages. The program below represents a SIMPL/I model for the classical machine-repairman problem which is used as the major instrument of analysis for comparing alternative scheduling strategies.

```
STMT
    PROBLEM: MODEL OPTIONS(MAIN);

       /****************************************************************/
       /*  SIMPL/I MODEL OF MACHINE REPAIRMAN PROBLEM                   */
       /*           WITH M MACHINES AND R REPAIRMEN                     */
       /*                                                               */
       /****************************************************************/

      MACHINES: BEHAVIOR;
         TAKE (NEGEXP(1,60));
         BREAKDOWN_TIME=CLOCK;
         CHECK: IF VOID(FREE_REPAIRMEN) THEN
           DO;
             HOLD UNTIL CHANGE(REPAIRED);
             GOTO CHECK;
           END;
         ELSE
            DO;
              REPAIRMAN = FIRST (FREE_REPAIRMEN);
              REMOVE FIRST FROM (FREE_REPAIRMEN);
              REPAIRMAN -> MACHINE = CURRENT;
              NOTIFY REPAIRMAN;
              HOLD;
              ENTER OBSERVED (CLOCK-BREAKDOWN_TIME) IN (REPAIR_TIME);
            END;
      END MACHINES;

     REPAIRMEN: BEHAVIOR;
         INSERT CURRENT IN (FREE_REPAIRMEN);
         HOLD;
         TAKE (NEGEXP(1,5));
         REPAIRED=REPAIRED+1;
         NOTIFY MACHINE;
    END REPAIRMEN;

       GET DATA ( M, R, SIMULATION_TIME);

       DO I = 1 TO M;
          START MACHINES AFTER TIME(0);
       END;
       DO I = 1 TO R;
          START REPAIRMEN AFTER TIME(0);
       END;

       DO I=1 TO SIMULATION_TIME/100;
          TAKE (SIMULATICN_TIME/10);
          SIMSNAP;
          PLOT REPAIR_TIME;
       END PROBLEM;
```