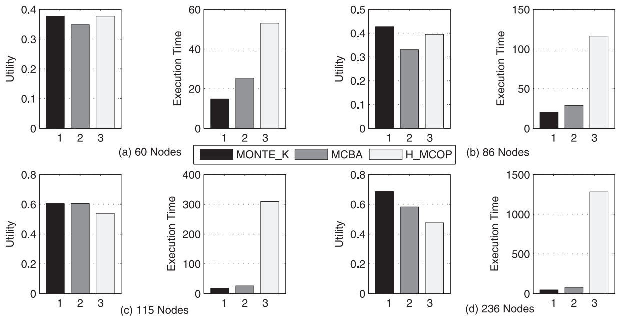
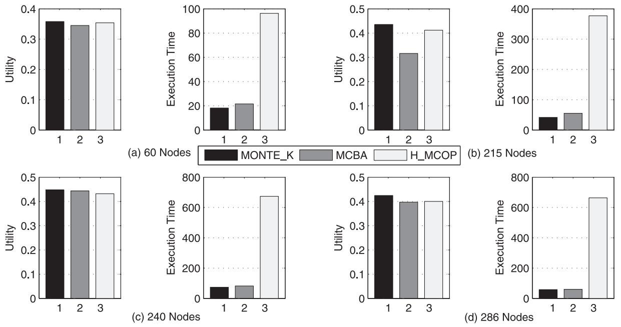


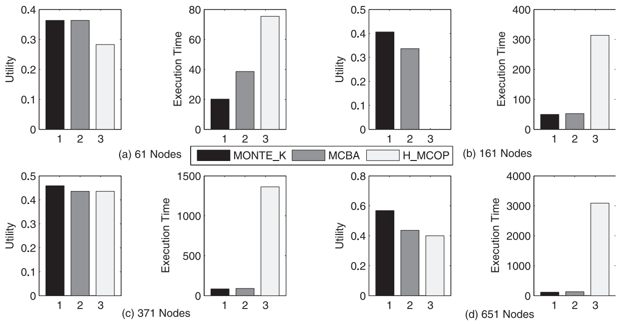
(a) 60 Nodes (b) 86 Nodes (c) 115 Nodes (d) 236 Nodes

Figure 4: Maximal length of paths is 4 hops



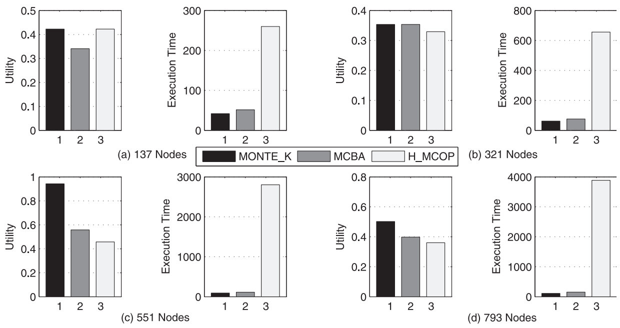
(a) 60 Nodes (b) 215 Nodes (c) 240 Nodes (d) 286 Nodes

Figure 5: Maximal length of paths is 5 hops



(a) 61 Nodes (b) 161 Nodes (c) 371 Nodes (d) 651 Nodes

Figure 6: Maximal length of paths is 6 hops



(a) 137 Nodes (b) 321 Nodes (c) 551 Nodes (d) 793 Nodes

Figure 7: Maximal length of paths is 7 hops

form 500 repeated experiments for MONTE_K and MCBA in each sub-network and record the utilities of the identified social trust paths in each experiment. The maximal utilities of the social trust paths identified in all 500 experiments by MONTE_K and MCBA are selected for the comparison with that yielded by H_MCOP. The average execution time of each of MONTE_K and MCBA in each sub-network is recorded based on 500 repeated experiments. The execution time of H_MCOP is averaged based on 5 independent executions. The results are plotted in Fig. 4 to Fig. 7.

Table 1: Properties of different social networks

ID	Max Hops	Number of Nodes	Number of Links	Max Outdegree	Max Indegree	Simulation Times
1	4	60	113	16	15	100
2	4	86	192	20	32	100
3	4	115	257	41	82	150
4	4	236	1321	74	91	200
5	5	60	107	9	18	100
6	5	215	528	34	48	200
7	5	240	655	32	49	250
8	5	286	749	31	84	300
9	6	61	124	17	32	100
10	6	161	355	43	46	200
11	6	371	1623	56	48	350
12	6	651	2475	173	151	450
13	7	137	373	48	18	200
14	7	321	860	39	38	350
15	7	551	3265	122	91	400
16	7	793	3411	83	89	500

Utility: We can see that in any of 16 cases, MONTE_K does not yield any utility worse than that of H_MCOP while in most sub-networks, the utilities of social trust paths identified by MONTE_K are better than those of H_MCOP (*see Fig. 4(a, c, d), Fig. 5(a) to (d), Fig. 6 (a) to (d) and Fig. 7 (b) to (d)*). The sum of utilities computed by MONTE_K is 12.23% more than that of H_MCOP in 4 hops sub-networks, 4.27% more in 5 hops, 60.62% more in 6 hops and 41.51% more in 7 hops. This is because when a trust path with the maximal utility is a feasible solution, H_MCOP can identify it as the optimal solution. However, when the identified trust path is not a feasible solution, H_MCOP can hardly find a near-optimal solution and some times yields an infeasible one even when a feasible solution exists (*see Fig. 6(b) where the utility computed by H_MCOP is 0*).

Regarding the utility of identified paths, MONTE_K also outperforms MCBA in most cases and is no worse than MCBA in all cases. The sum of utilities computed by MONTE_K is 17.25% more than that of MCBA in 4 hops sub-networks, 10.89% more in 5 hops, 14.30% more in 6 hops and 34.60% more in 7 hops. This is because *Strategy 2* in MONTE_K guarantees that the solutions identified by later simulations will be no worse than the current one.

Execution Time: From Fig. 4 to Fig. 7, we can observe that the execution time of MONTE_K is significantly less than that of H_MCOP in all sub-networks. The total execution time of MONTE_K is only 5.92% of that of H_MCOP in 4 hops sub-networks, 10.58% in 5 hops, 5.63% in 6 hops and 4.05% in 7 hops. In particular, in the most complex sub-network with 793 nodes, 3411 links and 7 hops (*see the last row of Table 1*), the execution time of MONTE_K is only 2.88% of that of H_MCOP (*see Fig. 7 (d)*). From the above results, we can see that MONTE_K is much more efficient than H_MCOP for identifying the optimal social trust path,