University of Waterloo
Department of Electrical and Computer Engineering
Spring 2014 – ECE457A Cooperative and Adaptive Algorithms

27. Solving Complex Task Allocation using Metaheuristic Algorithms

Complex Task Allocation

We have done a comparative analysis of five metaheuristic algorithms in solving a complex task allocation problem. Complex task allocation is a situation in which multiple agents are working cooperatively to solve a complex problem with many discrete tasks, optimizing the time it takes to complete all tasks and the quality with which they are completed. Often the problem is used to describe multi-robot systems.

Problem Formulation

Given a set of tasks and a set of robots capable of accomplishing them, we want to find the optimal plan for completing all tasks. We measure optimality in terms time to completion, quality of task completion relative to task priority, and utilization of a finite energy store by each robot.

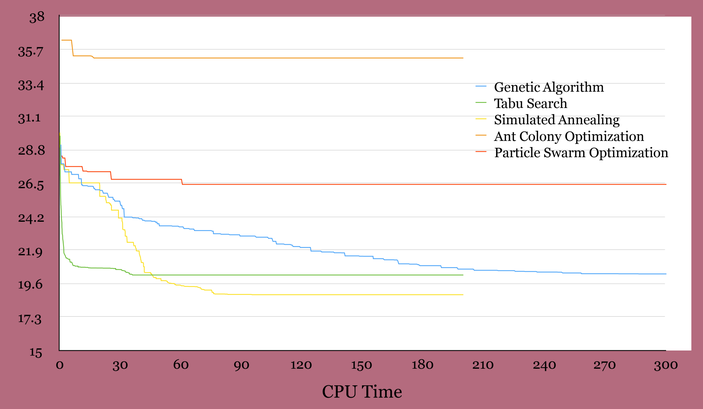
Problem Modeling

We base our model of this problem upon the multiple traveling salesmen problem (mTSP). A graph is constructed with nodes for each task and special nodes which switch agents. Unlike mTSP, the weights of edges and nodes depend on the agent traveling them. An additional constraint is added to ensure no robot exceeds its energy budget.

Metaheuristics

- **Tabu Search –** We used a variant of tabu search capable of working on permutation problems, with tabu tenure and the number of considered neighbors tuned to each data set.
- **Simulated Annealing -** We use simulated annealing on a geometric cooling schedule, with cooling factor and iterations per temperature tuned to each data set.
- **GA –** We use a permutation variation of GA with the order 1 crossover algorithm, the swap mutation and an elitism based survivor selection process
- **PSO –** We adapted PSO with new operators to give meaning to the concept of velocity and position within a permutation problem
- **ACO** – We use a variation of ACS on the problem graph, treating the ant's visit to a home node as a switch between robots and the graph's associated cost function.

Experimental Results



	ACO	PSO	GA	TS	SA
Iterations per second	1.21	2.30	6.22	58.18	503.42
Best cost after 1 CPU second	36.32	28.37	27.83	24.23	27.76
Iterations after 1 CPU second	1	2	5	35	356
Best value after 10 CPU seconds	35.25	27.64[1]	26.80	20.74	26.52
Iterations after 10 CPU seconds	11	23	55	507	4879
Best value after 100 CPU seconds	35.10	26.42	22.79	20.19	18.84
Iterations after 100 CPU seconds	121	229	591	5818	50342

Conclusion

- Tabu search worked very quickly to obtain good solutions but was prone to stagnation
- Simulated annealing found the most optimal solutions, though it converged more slowly than tabu search.
- Ant colony optimization, genetic algorithms, and particle swarm optimization did not perform well at this task.

Dane Carr, Austin Dobrik, Alexander Maguire, Ariel Weingarten, Lucas Wojciechowski