Robotics and Autonomous Systems 80 (2016) 34–42

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods

Kelin Jose, Dilip Kumar Pratihar *

Department of Mechanical Engineering, Indian Institute of Technology Kharagpur, Kharagpur-721 302, West Bengal, India

HIGHLIGHTS

- Centralized multi-robots system for industrial plant inspection.
- Deals with task allocation and collision-free path planning.
- Optimization problem.
- Task allocation problem is solved using genetic algorithm.
- Path planning problem is solved using A* algorithm.

ARTICLE INFO

Article history:
Received 13 March 2015
Received in revised form
10 February 2016
Accepted 22 February 2016
Available online 7 March 2016

Keywords:
Multi-robots system
Centralized system
Plant inspection
Collision-free path planning
Genetic algorithms
A* algorithm

ABSTRACT

Multi-robots systems have been effectively employed in various application domains. This study aimed at developing some heuristic methods for the task allocation and collision-free path planning for three robots working in the common workspace. In an application domain, there were ninety fixed locations in a plant, which were to be inspected by three robots after traveling through the minimum distance. Moreover, overall task completion time was to be as minimum as possible. A genetic algorithm (GA) had been used for the task allocation, and A* algorithm was utilized for path planning. The previous work on the same problem (Liu and Kroll, 2012) did not address the issue of collision avoidance in detail, which had been attempted in this study. Results of this study were found to be better than those of the previous work (Liu and Kroll, 2012). It could happen so, due to the reason that the GA was utilized in this study not only to schedule the tasks but also to assign optimal number of tasks to each robot. Thus, more environmental conditions were encoded in the GA-string.

1. Introduction

In a multi-robots system, multiple robots share the common workspace to perform assigned task(s), which could be difficult to do for a single robot efficiently. A multi-robots system could be either a centralized or a decentralized one. In a centralized multi-robots system, control is done using a central computer, whereas there is no supervisory control in a de-centralized multi-robots system. Multi-robots systems had been used to solve a variety of problems, some of which are discussed in the next section. The present study deals with a centralized multi-robots system.

The problem of task allocation deals with assigning the tasks to multiple robots working in the common workspace. Finding an optimal allocation of tasks is an NP-hard problem. Hence, these kinds of problems could be solved using heuristic search methods.

2. Literature review

Multi-robots systems had been utilized to tackle a variety of problems. Some of those problems are discussed here. Meng and Gan [1] investigated on decentralized coordination for multi-robot system used for cleaning up hazardous waste in dynamic environment. Their approach could achieve good levels of efficiency and robustness. Chakraborty et al. [2] formulated the box-pushing problem using two robots as a multi-objective optimization one and presented Pareto-optimal front of solutions by utilizing a non-dominated sorting genetic algorithm-II (NSGA-II). Both turning and translational motions of the box were allowed, and the

* Corresponding author. Tel.: +91 3222 282992; fax: +91 3222 282278.
E-mail addresses: kelin.jose@gmail.com (K. Jose), dkpra@mech.iitkgp.ernet.in (D.K. Pratihar).

http://dx.doi.org/10.1016/j.robot.2016.02.003
0921-8890/

robots could successfully shift the box to the desired goal position. Chaimowicz et al. [3] proposed an architecture for tightly coupled multi-robot coordination, where multiple robots cooperate to carry an object in an environment containing some obstacles. Fox et al. [4] developed an approach, where multiple robots would be able to efficiently explore unknown environments. Miyata et al. [5] proposed a task-assignment method for multiple cooperative robots in an unknown static environment. Later on, Yamashita et al. [6] developed a motion planning approach for multiple mobile robots transporting a large object after avoiding collisions. Wang and de Silva [7] integrated reinforcement learning with genetic algorithms to design an approach used to solve box-pushing problems by utilizing multiple cooperating robots. In a study performed by Öztürk, and Kuzucuoğlu [8], an iterative approach along with Greedy algorithm had been implemented. The approach although successfully yielded short-term solutions, that is, local minima, it could lead to suboptimal solutions.

"RoboGas Inspector" project [9] dealt with the inspection problem of industrial sites for gas leaks by utilizing multi-robot system. The problems related to the detection of gas and leak localization strategies had been studied by Kroll et al. [10]. The path planning and task allocation for such a system was tackled by Liu and Kroll [11]. However, this study on Multi-Robot Task Allocation (MRTA) for industrial plant inspection did not address the issue of collision avoidance between the robots. The issue of collision avoidance could be dealt using sensors, appropriate motion planner and controllers, but the time delay caused in this would affect the complete task allocation. Thus, it could be a better option to develop a scheme for collision-free task allocation.

In this study, an optimizer named genetic algorithm (GA) was utilized to obtain the task allocation and A* algorithm was used to solve the path planning problems of multiple robots working in the common workspace for carrying out industrial plant inspection in a centralized way. Thus, the optimal path between each location and the next was obtained. Moreover, a collision-avoidance scheme had been adopted to ensure collision-free path for each robot. Thus, two heuristic methods, namely GA and A* were utilized in this study to tackle the problems of task allocation and collision-free path planning of multiple robots carrying out industrial plant inspection.

The remaining part of this paper has been organized as follows: The tools and techniques used in the present study have been discussed in Section 3. Section 4 deals with mathematical formulation of the problem. The developed algorithm has been explained in Section 5. Results are stated and discussed in Section 6. Some concluding remarks are made in Section 7.

3. Tools and techniques used

In this study, A* algorithm and GA had been used for path planning and task allocation, respectively, whose working principles are briefly discussed below.

3.1. A* *algorithm*

A* algorithm is a graph search technique used to find a path from a given initial node to the pre-specified goal node. A heuristic estimate is employed here that ranks each node by an estimate of the best route that goes through that node. The algorithm begins at a start node. It then estimates the distance to the goal node from the current node. This estimate and the associated cost constitute the heuristics, which are assigned to the path leading to this node. The node is then included into a priority queue, and put into an open set. The algorithm then removes the next node from the priority queue. If the queue is seen to be empty, there is no path from the initial node to the goal node, and the algorithm stops. If the node is the goal node, A* reconstructs and outputs the successful path and stops. This path is reconstructed from the stored closed nodes. If this node is not the goal node, new nodes are created for all admissible adjoining nodes; the exact way of implementation depends on the problem to be solved. For each successive node, A* determines the cost of entering the node and saves it with the node. This cost is calculated from the cumulative sum of costs stored with its ancestors, and by adding the cost of the operation to it.

The algorithm also maintains a closed list of nodes, which have been checked. If a newly generated node is found to be already in this list with an equal or lower cost, no further processing is done on that node or with the path associated with it. If a node of the closed list matches with the new one, but has been stored with a higher cost, it is deleted from the closed list, and processing continues on the new node. Next, an estimate of the new node's distance to the goal is added to the cost to form the heuristic for that node. This is then included in the open priority queue, unless an identical node with less or equal heuristic is found there. Interested readers may refer to Russel and Norvig [12] for a more detailed description of this algorithm.

3.2. *Genetic algorithm*

Genetic Algorithm (GA) is a population-based probabilistic search and optimization technique. It works based on Darwin's principle of natural selection [13]. The process starts with a population of randomly generated solutions. This population of solutions undergoes various operations to form the next generation. In each generation, the fitness of each individual in the population is evaluated. The fitness is usually the value of the objective function in an optimization (maximization) problem to be solved. The better individuals in terms of their fitness values are stochastically selected from the current population, and each individual's genome is modified to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. The algorithm terminates, when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached by the solution(s).

A typical GA requires a genetic representation of the solution domain, and a fitness function value (that is, objective function value) to evaluate the solution. The solution domain will be represented usually by an array of bits (in binary-coded GA) or that of real numbers (in real-coded GA). The main aim of the GA will be to improve the fitness values of the population of solutions using different operators like reproduction, crossover, mutation, and others, and finally to determine the optimal solution(s). Interested readers may refer to Pratihar [14] for a detailed description of the working principle of GA.

4. Mathematical formulation of the problem

The following assumptions are made:

- Each robot can execute only one task at a time.
- Only one robot is required to execute each task.
- Each task is executed only once.
- All the tasks are to be executed.
- All the robots start from the depots at the same time.

Let us consider an optimization problem, where a group of m robots $R = \{R_1, R_2, \ldots, R_i, \ldots, R_m\}$ are to be assigned inspection task at n locations $T = \{T_1, T_2, \ldots, T_j, \ldots, T_n\}$ in an optimal sense, such that either the completion time (that is, maximum of the traveling times taken by m robots) or total fuel consumption (representative of total traveled distance) becomes the minimum. Let C_{ij} be the cost (expressed in terms of the sum of traveling time and inspection