Robotics and Autonomous Systems 57 (2009) 129–133

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

An experimental study of distributed robot coordination

Stefano Carpin [a,*], Enrico Pagello [b]

[a] School of Engineering, University of California, Merced, USA
[b] Department of Information Engineering, University of Padova, Italy

ARTICLE INFO

Article history:
Available online 10 July 2008

Keywords:
Multi-robot systems
Intelligent mobility
Randomized algorithms
Motion planning

ABSTRACT

Coordinating the path of multiple robots along assigned paths is a computationally hard problem with great potential for applications. We here provide a detailed experimental study of a randomized algorithm for scheduling priorities that we have developed, and also compare it with exact and approximated solutions. It turns out that for problems of reasonable size our algorithm exhibits an appealing compromise between speed and quality.

1. Introduction

Planning the motion of multiple robot systems has been a task investigated since the early days of mobile robotics. While the problem is interesting in itself, because of the inherent computational complexity it exhibits, it has to be acknowledged that few applications have been presented up to now in the context of autonomous mobile robots. This is partly due to the fact that systems with a remarkable number of robots have not been deployed yet. Moreover, when multiple robots are to operate in a shared unstructured environment, one of the holy grails of multi-robot research, their motion is commonly governed by reactive navigation modules rather than by precisely planned paths. That said, it is not implied that the problem in itself has lost interest from the applicative point of view. The demand for systems capable of governing the motion of multiple objects in shared environments is instead ever increasing. Applications include, for example, luggage handling systems at airports, storage systems in factories, moving containers in harbors, and more. The theme of *intelligent mobility* is envisioned to play a major role in the foreseeable future. Possibly, one of the major differences that will be seen will be a decrease in individual robots' motion freedom. Sticking to the storage systems in factories example, mobile carts are not free to wander wherever they want, but are rather constrained to proceed along predefined paths, usually hard wired in hardware. Given a set of predefined paths, coordinating the motion of vehicles along these routes will be asked for more and more often. And, of course, it will be necessary to find solutions that optimize certain performance indices, like for example, consumed energy, time needed to complete the task, and the like. From a computational point of view these problems have been studied for quite some time, and their inherent complexity has soon been detected. Approximated and heuristic-based solutions are therefore a must when one is required to deal with multiple moving objects. In this paper we offer an experimental assessment of a simple randomized approach to solve the coordination problem we have proposed in the past. Section 2 discusses related literature, while the problem is formalized in Section 3. The solving algorithm is illustrated in Section 4, and the experimental framework and results are shown in Section 5. Conclusions are finally provided in Section 6.

2. Related work

The problem of multi-robot motion planning has been continuously studied in the past. The first major distinction concerns *centralized* versus *decentralized* approaches. When a centralized motion planner is used, one process has to plan the motion for the whole set of robots. The obvious drawback is in the high dimensionality of the composite configuration space to be searched. In a decentralized approach every robot plans its own motion, and then has to undergo a stage of negotiations to solve possible collisions with other robots. Decentralized approaches are inherently incomplete, but much faster. Sánchez and Latombe [1] speculated that decentralized approaches are likely to show their incompleteness often when used in industrial production plants. In the case of mobile robot systems, however, the environment is likely to be less cluttered and hence these problems are less likely to occur. Efficient methods to solve the single robot motion planning are available [2–4] and will not be further discussed here (recent books on motion planning like [5,6] provide extensive and up to date coverage of the topic). A common approach

* Corresponding address: School of Engineering, University of California, 5200 North Lake Road, PO BOX 2039, 95344 Merced, CA, USA. Tel.: +1 228 209 4152.
E-mail address: carpin@ieee.org (S. Carpin).

0921-8890/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2008.07.001

to solve the multi-robot motion planning problem consists in assigning priorities to robots and planning their motion according to them [7]. Paths for robots are computed one after the other, according to their priority. When planning the motion of a robot with priority p_j, it is necessary to take into consideration the already planned motions for robots with priority p_i, where $p_i < p_j$. Finding a good priority schema is a hard problem in itself [8]. A related problem that we address in this paper consists in coordinating the path of a set of robots along given specified paths. As the paths may intersect with each other, it might be necessary to stop certain robots when approaching an intersection point in order to give way to other robots and avoid collisions. In this context one is generally interested in minimizing certain parameters, like, for example, the time needed by all robots to reach their final destination. This rules out certain trivial coordination schemas, like, for example, the one where just one robot moves and all the others remain stationary, since the overall time would be too high. LaValle and Hutchinson solved the problem using a game-theoretic approach based on multi-objective optimization [9,10]. The approach allows tuning of the algorithm behavior between centralized planning and complete decentralized planning. The authors show that optimal results can be found, but a significant amount of time is needed. Simèon et al. [11] solved the path coordination problem using a resolution complete algorithm. They show how it is possible to split a given path in segments where the robot will not collide with any other robot, and segments where paths intersect. The authors illustrate results involving up to 150 robots, but where no more than 10 robots are intersecting each other's path. Computation time is in the order of minutes. Akella and Hutchinson [12] solve the problem of robot coordination along predefined routes by simply varying the start time of each robot. Once a robot starts to move, it never stops until it reaches its target position. Peng and Akella recently extended these ideas addressing the case of robots with kinodynamic constraints [13].

3. Problem formulation

The problem we aim to study is the following: given n robots, assume that n paths, one for each robot, are provided. We suppose that each path has been subdivided into *free* and *occupied* segments. Given a path, an occupied segment is a part of the path such that the robot can collide with other robots while it is moving along that part of the path. Occupied segments arise when different paths intersect each other, or are very close (see Fig. 1 for an example). Any segment that is not occupied is declared free. In light of results presented in [11,13], free and occupied segments can be efficiently determined. Hence, a path p_i can be seen as a sequence $p_i^1 \dots p_i^{s(i)}$, where each p_i^j is either a free or an occupied segment, and $s(i)$ is the number of segments composing path p_i. The task is to find a coordination schema, i.e. a mapping:

$$C : [0, T] \to \{1 \dots s(1)\} \times \{1 \dots s(2)\} \times \dots \times \{1 \dots s(n)\} \quad (1)$$

such that for each time $0 \leq t \leq T$ no two or more robots are moving along path segments that collide with each other. In the beginning, robot i is positioned at segment p_i^1, and in the end it has to reach $p_i^{s(i)}$. While moving through the different segments, a certain amount of time will be spent to traverse each of them. Let $t(p_i^j)$ be the time spent by robot i to traverse segment p_i^j ($1 \leq j \leq s(i)$). Throughout the paper we assume that robots only move forward along their paths, though they can stop at certain points to give way to other robots. However, they never backtrack along their route. The goal is to find a coordination schema that

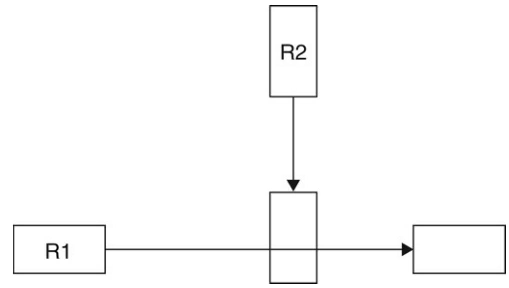

Fig. 1. The simplest case of robot coordination, involving two robots only. R1's path can be divided into three segments, free, occupied and free respectively. R2's segment can be divided into two segments. If R2 is given a priority higher than R1, and both robots travel at the same speed, R1 will not be able to reach its final destination, because of R2 stopping on its path.

minimizes the time needed by all robots to complete their motion. Formally we aim to minimize the following quantity

$$z = \max_{1 \leq i \leq n} \sum_{j=1}^{s(i)} t(p_i^j). \quad (2)$$

It can be shown that this problem is equivalent to the Job Shop Scheduling (JSS) Problem, which is known to be NP-hard [14]. The JSS problem asks how to schedule n jobs that have to be processed through m machines in such a way that the overall required time is minimized. The constraints are, that no machine can process more than one job at the same time, and that each job has to be processed by the machines in a given order. In the path coordination problem, each robot is a job, and each free or occupied segment is a machine. The reader should note that while reducing the robot motion planning coordination problem to an instance of the JSS, not every job needs to be processed by every machine (i.e. not every robot has to travel through all the possible segments). Under the assumption that $P \neq NP$, the search for a coordination schema that minimizes the time needed to complete the motion task is doomed to take exponential time. This motivates the great number of approximated and heuristic approaches that have been proposed throughout the years.

4. Random rearrangements

In our former work [15] we proposed a simple distributed schema to solve the multi-robot motion planning problem. Similar ideas were later used in [16]. The idea is slightly modified here to describe how the various robots can operate to find a valid coordination schema, and is depicted in Algorithm 1. The algorithm assumes that a data structure *SpaceTime* is available. *SpaceTime* records which part of the space is occupied or free at a given time. The SpaceTime data structure can be accessed by providing two indices, one for the space and one for the time. The algorithm picks a random priority schema (line 1), and then schedules the robot motions according to the selected priority schema. The first considered robot will be scheduled to move straight along its path with no stops, and SpaceTime, accordingly, will be updated. When scheduling successive robot motions, it is necessary to check whether the robot can move to its next path segment or if that is already occupied (line 6). If it is possible, the robot moves to its next segment (line 7), or a delay is inserted (line 11). In both cases SpaceTime is updated to record the robots' position (line 12) while time evolves (lines 8 and 11).

Here we stick to the hypothesis formulated in [15], i.e. that each robot will apply this procedure to compute a coordination schema, and that in the end the one leading the best value for the variable z, formerly defined, will be used to execute the real motion. So, when