On the Problem of Multiple Mobile Robots Cluttering a Workspace*

B. J. OOMMEN

and

I. REICHSTEIN

School of Computer Science, Carleton University, Ottawa K1S 5B6, Canada

ABSTRACT

We consider the problem of multiple mobile robots navigating in a common workspace. Initially, based on the premise that the costs of operation of the robots can be ignored, results regarding the probability of collision of two point robots in the workspace are derived. Obviously, these results depend on the geometry of the workspace, and hence they are derived for workspaces with various geometries. The results are then extended for two robots of finite dimensions. The question "Are $k+1$ robots better than k?" is then considered. Based on two modes of operation, namely, the batch-scheduled mode and the list-scheduled mode, various theoretical results are derived. If the costs of operation of the robots can be ignored, based on various computational results obtained, we conjecture that the answer to the question is in the affirmative in both modes of operation. Finally, by using the model of computation which includes the costs of operation of the robots, the problem is revisited. We have shown that for various geometries and revenue to cost ratios, there are optimal finite values for the number of robots that can operate within the workspace. Expressions for these optimal values have been derived.

INTRODUCTION

The area of robotics is one of the most fascinating and interesting areas of computer science. Not only is it an area of great importance economically, but as a research area, robotics encompasses such fields as kinematics, mechanics, computational geometry, control, and language design.

One of the most interesting areas in robotics is the study of the problem of navigating a robot (or a manipulator) within a workspace. When the robot has

*Partially supported by the Natural Sciences and Engineering Research Council of Canada. A preliminary version of some of these results was presented at the Tenth IASTED International Symposium on Robotics and Automation, Lugano, June 1987.

655 Avenue of the Americas, New York, NY 10010 0020-0255/92/$5.00

no obstacles to avoid, the problem is essentially a control problem. Solutions usually involve joint interpolated motions, when the trajectory is not necessarily linear, and motions computed using recursive algorithms (such as Taylor's algorithm) if the path desired is linear. However, the problem is far more complex if the robot (or manipulator) has to plan its motion when there are obstacles in its workspace.

Udupa [31] was probably the initiator of the work on motion planning amidst obstacles. He introduced the concept of working in configuration space, and later this was studied extensively by many researchers, including Brooks and Lozano-Perez [3], Lozano-Perez and Wesley [19, 20], and Oommen and Reichstein [23]. Simultaneously, a variety of results concerning the theoretical issues of motion planning were presented by researchers, among whom are Schwartz, Sharir, and Yap. Since the literature in this field is so extensive, we refer the reader to a comprehensive survey by Whitesides of the papers and results in the area [33]. We strongly recommend this survey for a researcher who is just embarking on working in this rather extensive field.

Whereas the problem of moving a single robot in known terrains was almost completely investigated, the problem of navigating in unknown terrains was almost completely untouched till the last few years. The study of the problem of navigating mobile robots sparked a whole series of very interesting results. Learned paths were suggested by Iyengar et al. [15], who also suggested using learned spatial graphs [14] to compute the path of the robot. Oommen et al. presented a formal approach to tackling the problem using learned visibility graphs [22]. Although their solution was suitable only for a point robot, we believe that it is conceptually an ideal working model, inasmuch as the learned visibility graph actually converges to the actual visibility graph. Alternate solutions which exist for *practical* robot navigation systems have been proposed, which involve map making [2], environment learning [5], discretization of the terrain [7], and the use of multilevel planning [10]. Other papers involving mobile robot navigation are those of Gouzenes [11] and Chattergy [6], as well as Moravec's report on the CMU Rover [21], Ozguner et al.'s approach to the navigation of a hexapod [24], and Thompson's paper on the navigation of the JPL robot [29].

A subsequent question of much importance has been that of navigating (or moving) multiple objects. Schwartz and Sharir [27] presented an analytic solution for the special case in which the objects to be moved were circular and the obstacles were polygonal. This problem was also studied by Ramanathan and Alagar [25]. Yap [34] discussed the theoretical issues involved in coordinating the motion of several disks. The general problem of coordinating the motion of multiple independent objects was shown by Hopcroft, Schwartz, and Sharir [13] to be PSpace-hard. Erdmann and Lozano-Perez [8] described an efficient technique (in configuration space) to move multiple objects and in particular to move multiple manipulator arms. Using a certain mechanical construction, Shih [28] derived the elliptic nature of the path taken by two robots moving in antiparallel directions.

In this paper we consider a problem that has been addressed only marginally in the literature. The motivating result which triggered this research was due to Grossman et al. [12, 36] of IBM. In their paper [12], they considered the value of using multiple independent robot arms. Clearly this "value" depends on the criterion function used to evaluate the performance of the multiple robots. The reader will probably be best motivated, for this present paper, by reading the following extract from [12]:

> In a given time interval, if one arm can perform one unit of work, then n independent arms all working in parallel could, in principal, perform n units of work. In reality, n is only an upper bound, since for most applications there would be spatial contention, and the arms will occasionally interfere with one another.
>
> If there were no provision for avoiding such interferences, multiple independent robot arms might become so deadlocked as to render unfeasible a task that can be done with a single arm. The simplest solution is to ensure that each arm other than the first has a parked position that leaves the first arm unencumbered. With this provision, the work W accomplished by n arms must satisfy the trivial inequality
>
> $$1 < W(n) < n$$
>
> where $W(1)$ is assumed normalized to 1.

Based on a particular criterion function, Grossman et al. actually proved a fascinating result. They proved that in both one and two dimensions, there is "little merit" in having more than *two* arms. In [12], the authors alluded to the more general question of this problem being related to the classical Buffon's needle problem, which deals with the probability of a randomly dropped needle landing across parallel lines. However, the problem of studying the "value" of multiple robots navigating in arbitrary workspaces was left open.

In this paper, we consider the general problem of moving k mobile robots in a workspace of any given geometry. The ultimate intention is to address the question "Are $k+1$ robots better than k?" To solve this problem, we first consider the problem of *two* robots working inside a workspace, and an index termed the *collision index*, p, is defined. The value of this index, which clearly depends on the geometry of the workspace, has been derived for two point robots operating in various workspaces. The problem has also been solved approximately for two robots with *finite* dimensions operating within a circular workspace.

Using these results for the collision index p, the problem of evaluating the use of multiple robots in the workspace has also been studied. Based on two simplifying assumptions, a variety of results have been derived for the cases when the robots operate in either a batch-scheduled or a list-scheduled mode of operation. It has first been shown that the upper bound for the time complexity of the decision problem "Are $k+1$ robots better than k?" is exponential in both modes of operation. In fact, it consists of computing $O(2^{k^2})$ decision problems, *each of which is NP-complete.* However, based on the results we have obtained (for $k \leqslant 7$ and $k \leqslant 6$ for the list and batch modes respectively), we conjecture that the answer to the latter decision problem is in the affirmative. Suffice it to say that since the computations involved for $k = 7$ and 8 require almost a whole year of dedicated computing facilities on a Sun system [37, 38] for the batch and list cases respectively, we have extrapolated our results for one extra robot in each case. The extrapolated results seem to confirm the conjecture. However, even with an elementary model of computation we shall show the result that the index of productivity per robot *decreases* with increasing number of robots.

The model of computation is then changed, by changing the criterion function to one in which the costs and revenues involved in operating a robot are considered. For both the above modes of operation it is shown that for a fixed revenue to cost ratio, there is a maximum finite value for the number of robots which can work profitably inside the workspace. Employing more robots than this value is actually detrimental, in the sense that it would cause the user to incur a loss.

Notation I.1. If S and T are any points in the plane, $S \to T$ is the straight line originating at S and terminating at T. We also define $C(N, M)$ to be the binomial coefficient $N!/[(M!)(N-M)!]$.

II. THE COLLISION INDEX

Let two robots R_1 and R_2 be working inside a workspace $\mathfrak{W}$. We define a *job* to be the action of a single robot R_i navigating from its beginning point B_i to its end point E_i, $i = 1, 2$. In practice, the motion of R_i from B_i to E_i could involve the transportation of some object or (hazardous) material. By definition, we say that a single robot working by itself executes a single job in one unit of time.

Suppose the robots R_1 and R_2 are commanded to move from B_1 to E_1 and B_2 to E_2 respectively. The assumption made throughout this paper is that the robots move along straight line paths. Clearly, if the paths $B_1 \to E_1$ and $B_2 \to E_2$ are nonintersecting, the robots can execute their motions in parallel.

However, whenever the paths intersect, we assume that the robots are scheduled in sequence so as to avoid a collision.

It is clear that both the assumption of having straight line paths and the assumption of scheduling robots in sequence if their straight line paths collide are quite restrictive. One could easily envisage a scheme in which the robots avoid the paths of their "colleagues" by taking circuitous routes. However, the strategy for planning such routes is not obvious even for robots and workspaces with simple geometries. Alternatively, one could also envisage schemes in which robots used multiple stops (or controlled velocities) so as to avoid collisions. In the latter scenario, if R_1 were to go from B$_1$ to E_1 and R_2 from B_2 to E_2, and if $B_1 \rightarrow E_1$ intersected $B_2 \rightarrow E_2$, the path planning would be arranged so that B_1 went to an intermediate midpoint M_1 and waited until R_2 crossed the intersection points of the paths. As this juncture we suggest that the above two strategies are problems open for research, but we note in passing that for a given multirobot path pattern, even for the straightforward path planning strategy which we have employed, the supervisory task for the robots is an NP-complete problem.

We define the collision index, p to be the probability of the paths $B_1 \rightarrow E_1$ and $B_2 \rightarrow E_2$ intersecting. Clearly, this index p depends on the geometry of $\mathfrak{W}$, the dimensions of the robots, and the constraints on the extrema of the paths within the workspace. We shall now derive the collision indices for point robots for a variety of workspaces and then extend these results for robots of finite dimensions.

THEOREM I. *The collision index for two point robots moving between uniformly distributed random points on the circumference of a circular workspace is* $\frac{1}{3}$.

Proof. Consider Figure 1 in which robot R_i moves from B_i to E_i, where $i = 1$, 2, and further, where B_i and E_i are on the circumference of the circle. With no loss of generality, let the radius of the circular workspace be unity. Let θ_{1b} and θ_{2b} be the polar angles of the points B_1 and B_2 respectively. With no loss of generality the measurement of these polar angles can be from B_1, thus setting $\theta_{1b} = 0$. Let θ_{1e} and θ_{2e} be the polar angles of the points E_1 and E_2 respectively. Clearly, θ_{2b}, θ_{1e}, and θ_{2e} are uniformly distributed in [0, 2π]. For the rest of this theorem, we shall refer to the extrema of the robot's paths in terms of their polar angles.

Since R_1 is moving inside the circle along $\theta_{1b} \rightarrow \theta_{1e}$, and R_2 is moving along $\theta_{2b} \rightarrow \theta_{2e}$, the paths will *not* intersect if and only if

$$\mathfrak{C} = \mathfrak{C}_1 \vee \mathfrak{C}_2,$$

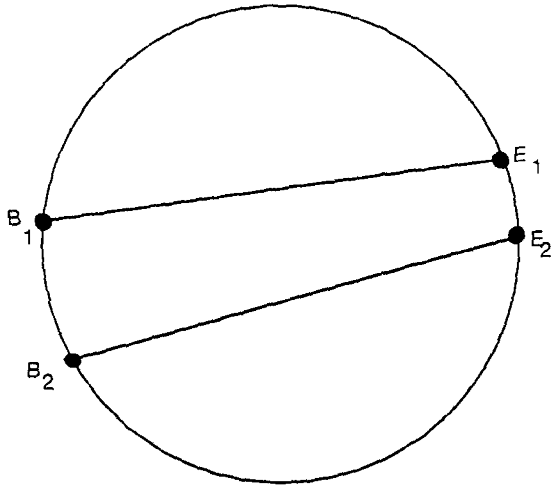


Fig. 1. Two point robots moving between uniformly distributed random points on the circumferance of a circular workspace. Robot R_i moves from B_i to E_i for $i = 1, 2$.

is satisfied, where

$$\mathfrak{E}_1 = \theta_{2b} \in [0, \theta_{1e}] \wedge \theta_{2e} \in [0, \theta_{1e}],$$

$$\mathfrak{E}_2 = \theta_{2b} \in [\theta_{1e}, 2\pi] \wedge \theta_{2e} \in [\theta_{1e}, 2\pi].$$

Consider the value of $\Pr[\mathfrak{E}_1 \mid \theta_{1e}]$. Since θ_{2b} and θ_{2e} are independently drawn uniform random variables,

$$\Pr[\mathfrak{E}_1 \mid \theta_{1e}] = (\theta_{1e}/2\pi)^2.$$

Hence the unconditional (total) probability of event $\mathfrak{E}_1$ is

$$\Pr[\mathfrak{E}_1] = \frac{1}{4\pi^2} \cdot \frac{1}{2\pi} \int_0^{2\pi} (\theta_{1e})^2 \, d\theta_{1e}$$

$$= \frac{1}{8\pi^3} \frac{\theta_{1e}^3}{3} \bigg|_0^{2\pi} = \frac{1}{3}.$$

Similarly, $\Pr[\mathfrak{E}_2 \mid \theta_{1e}] = [(2\pi - \theta_e)/2\pi]^2$, whence,

$$\Pr[\mathfrak{E}_2] = \frac{1}{4\pi^2} \cdot \frac{1}{2\pi} \int_0^{2\pi} (2\pi - \theta_{1e})^2 \, d\theta_{1e} = \frac{1}{3}.$$

Hence, Pr[nonintersecting paths for R_1 and R_2] $= \Pr[\mathfrak{E}_1] + \Pr[\mathfrak{E}_2] = \frac{2}{3}$. Thus, the collision index p has the value,

$$p = 1 - (\Pr[\mathfrak{E}_1] + \Pr[\mathfrak{E}_2]) = \frac{1}{3}. \qquad \blacksquare$$

Remark. As pointed out by an anonymous referee whose assistance we have recognized in the acknowledgements, this result can be alternatively derived using a discrete probability approach. Choose any four points on the circle. Observe that these points can be labeled as the starting and the destination points of the robots in question. For any set of points thus chosen, if the points B_1, E_1, B_2 and E_2 are permuted, there are twenty-four possible permutations, eight of which involve intersections. Therefore $p = \frac{1}{3}$. This argument is very nice in that it would apply equally well to a large class of closed convex curves. However, one should be careful not to generalize this argument uncritically, as one could select a closed convex curve consisting of two parts, one of which is linear. The argument would fail in the latter case because paths involving points on the linear boundary would not be internal paths, and would thus be nonintersecting. However, the fundamental argument using the continuous probability approach described in the above proof would still be consistent.

Corollary. *Let ξ_2 be the mean time to execute a job with two robots at work. Then, for two point robots moving between random points on the circumference of a circular workspace, ξ_2 is $\frac{2}{3}$.*

Proof. Let T_2 be the mean time for both the robots to execute their jobs. Then

$$\begin{aligned} T_2 &= 1 \times \Pr[\text{paths don't intersect}] + 2 \times \Pr[\text{paths do intersect}] \\ &= 1(1-p) + 2p = 1 + p. \end{aligned}$$

However, observe that in T_2 units of time two jobs have been executed. Hence,

$$\xi_2 = \frac{T_2}{2} = \frac{1+p}{2} = \frac{2}{3}. \qquad \blacksquare$$

The expressions for the value of p are obtained in a rather straightforward way in the case when the workspace is circular. In the case when the workspace is polygonal, however, the expressions are a little more complicated. Indeed, in this situation, we take into account the principle that two robots may not be at the same place at the same time, thus excluding certain paths that the robots might traverse. We first consider the case when two robots are operating *between the vertices* of a regular n-gon. We will restrict the motions of the robots in this case by insisting that the robots may not have either common starting points or common destination points, i.e., $B_1 \neq B_2$, and $E_1 \neq E_2$. We do, however, allow the motion of R_1 from $B_1 \rightarrow E_1$ at the same time that R_2 is moving from $E_1 \rightarrow B_1$. In other words, we permit two robots to be moving in antiparallel directions between the same extrema. This model is *not* unrealistic, since the robots are but points and the actual implementation of the paths can be achieved by perturbing the robots by a small quantity from the straight line joining the vertices.

THEOREM II. *The collision index p for two point robots moving between uniformly distributed random vertices of a regular N-gon is*

$$p = \frac{N^2 - 5N + 6}{3(N^2 - 3N + 3)}$$

whenever the point robots are permitted to move in antiparallel directions.

Proof. Consider Figure 2. With no loss of generality, let the starting vertex of R_1 be the vertex whose index we call 0. Let R_1 be commanded to do the job $0 \rightarrow E_1$, where $E_1 \in \{1, 2, 3, \ldots, N-1\}$. Clearly, E_1 can take any of these values with probability $1/(N-1)$.

For a given path $0 \rightarrow E_1$ for R_1, R_2 may originate at B_2, which can be any one of $N-1$ vertices, namely $\{1, 2, \ldots, N-1\}$. Further, whenever $B_2 \neq E_1$, R_2 can terminate at E_2, which can be any one of $N-2$ vertices, namely $\{0, 1, 2, \ldots, N-1\} - \{E_1\} - \{B_2\}$. But when $B_2 = E_1$, there are $N-1$ possible points for E_2. Thus the total number of legal paths which R_2 can take is

$$\#_{\text{legal}} = (N-2)(N-2) + (1)(N-1) = N^2 - 3N + 3.$$

However, for a given value of E_1, the paths intersect wherever $\mathfrak{F}_1$ or $\mathfrak{F}_2$ is true, where

$$\mathfrak{F}_1 = B_2 \in \{1, 2, \ldots, E_1 - 1\} \wedge E_2 \in \{E_1 + 1, \ldots, N-1\}$$
$$\mathfrak{F}_2 = B_2 \in \{E_1 + 1, \ldots, N-1\} \wedge E_2 \in \{1, 2, \ldots, E_1 - 1\}$$

Due to the symmetry of the situation, it can be seen that the number of intersecting paths is $2(E_1 - 1)(N - E_1 - 1)$. Thus, the probability of the robot

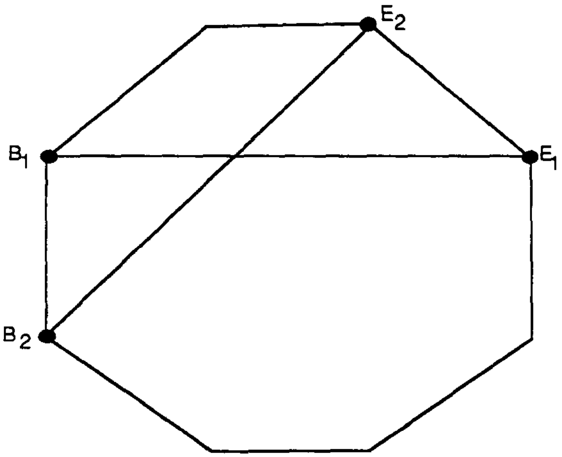


Fig. 2. Two point robots moving between uniformly distributed points on the vertices of a polygonal workspace. Robot R_i moves from B_i to E_i for $i = 1, 2$.

paths intersecting is

$$p = \frac{\sum_{E_1=1}^{N-1} 2(E_1 - 1)(N - E_1 - 1)}{(N-1)(N^2 - 3N + 3)}.$$

This can be simplified and evaluated to yield the desired result. ■

REMARKS.

1. One might have expected that as the number of vertices increased the probability of intersection of the paths would decrease, since the number of possible paths for the robots increases quadratically. However, it is the contrary which is true. The value increases monotonically with N from 0.095 for a square, to 0.154 for a pentagon, and to 0.191 for a regular hexagon. The *worst* case is for the circle, when $p = \frac{1}{3}$.

2. The mean time per job in the case of two robots operating in a regular N-gon is

$$\xi_2 = 0.5 + \frac{N^2 - 5N + 6}{6(N^2 - 3N + 3)}.$$

This time increases monotonically with the number of vertices in the N-gon.

3. In the case when the robots are *not* permitted to move in antiparallel directions, it can be seen that the number of legal paths for R_2 is $(N-1)(N-2)$, and thus the collision index p in this case has the value $(N-3)/3(N-1)$.

THEOREM III. *The value of p, the collision index for two point robots operating randomly between uniformly distributed points inside a circular workspace of radius R and with extrema of the paths not necessarily on the circumference, is approximately* 0.21.

Proof. Consider Fig. 3. Let B_1 and E_1 be the beginning and end points of the path that R_1 is commanded to take. Let B_2 be the beginning point for R_2. Clearly, the path of R_2 intersects the path of R_1 if the point E_2 lies in the region of the circle obstructed by the line $B_1 \rightarrow E_1$, which is the region shaded in Fig. 3.

Let C_1 and C_2 be the intersections of $B_2 \rightarrow B_1$ and $B_2 \rightarrow E_1$ and the circle respectively. Clearly, the probability of intersection of the paths given B_1, E_1, and B_2 is proportional to the shaded area $C_1B_1E_1C_2$. By using analytic geometry the following results can be derived.

Let $B_1 = (b_{1x}, b_{1y})$, $E_1 = (e_{1x}, e_{1y})$, and $B_2 = (b_{2x}, b_{2y})$. Let the points C_1 and C_2 be (c_{1x}, c_{1y}) and (c_{2x}, c_{2y}) respectively. Let α be the positive root of

$$\alpha^2\left[(b_{2x} - b_{1x})^2 + (b_{2y} - b_{1y})^2\right] + \alpha\left[2(b_{2x}b_{1x} + b_{2y}b_{1y}) - 2(b_{2x}^2 + b_{2y}^2)\right] + \left[b_{2x}^2 + b_{2y}^2 - R^2\right] = 0.$$

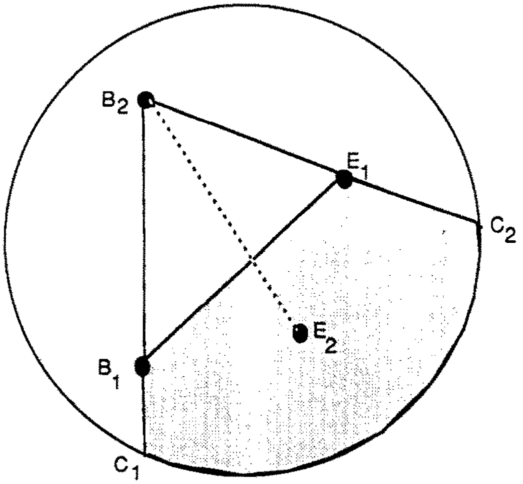


Fig. 3. Two point robots moving within a circular workspace. Robot R_i moves from B_i to E_i for $i = 1, 2$. Observe that the paths intersect if B_2 is outside the region $C_1B_1E_1C_2$ and E_2 is inside that region.

Then $c_{1x} = (1-\alpha)b_{2x} + \alpha b_{1x}$ and $c_{1y} = (1-\alpha)b_{2y} + \alpha b_{1y}$. Similarly, let β be the positive root of

$$\beta^2\left[(b_{2x} - e_{1x})^2 + (b_{2y} - e_{1y})^2\right]$$
$$+ \beta\left[2(b_{2x}e_{1x} + b_{2y}e_{1y}) - 2(b_{2x}^2 + b_{2y}^2)\right] + \left[b_{2x}^2 + b_{2y}^2 - R^2\right] = 0.$$

Then $c_{2x} = (1-\beta)b_{2x} + \beta e_{1x}$ and $c_{2y} = (1-\beta)b_{2y} + \beta e_{1y}$. Thus,

$$\Pr\left[\text{intersection} \mid B_1, E_1, B_2\right] = \frac{\text{Area}(C_1\, B_1\, E_1\, C_2)}{\pi R^2}. \tag{1}$$

This tells us that p, the collision index, is obtained by integrating (1) over all possible values if B_1, B_2, and E_1. Thus,

$$p = \frac{1}{(\pi R^2)^4}, \int_{B_1}\int_{E_1}\int_{B_2} \text{Area}(C_1\, B_1\, E_1\, C_2)\, dB_1\, dE_1\, dB_2 \tag{2}$$

where the above dB_1, dE_1, and dB_2 are vector differentials. The value of the above integral is approximately 0.21. ■

Till now we have assumed that the robots within the workspace are point robots. To render the problem more physically meaningful, we shall conclude this section by evaluating the collision index for a robot with fixed dimensions.

THEOREM IV. *Let $\mathfrak{W}$ be a circular workspace of unit radius. Let the robots R_1 and R_2 be such that the angle subtended at the center by the robots is 2Δ whenever $\Delta \leqslant \pi/2$. Then, if the robots move between uniformly distributed points on the circumference of $\mathfrak{W}$, the collision index p is*

$$p = 1 - \frac{2(\pi^2 - 6\Delta\pi + 12\Delta^2)}{3(\pi - 2\Delta)^2}.$$

Proof. Consider Figure 4. With no loss of generality, let one tip of robot R_1 be at the point whose polar angle is 0 with respect to the center of $\mathfrak{W}$. R_1 can now be commanded to move to a point where the position of this tip is at an angular position θ_{1e}. Clearly, θ_{1e} can take any value in $[\Delta, 2\pi - \Delta]$.

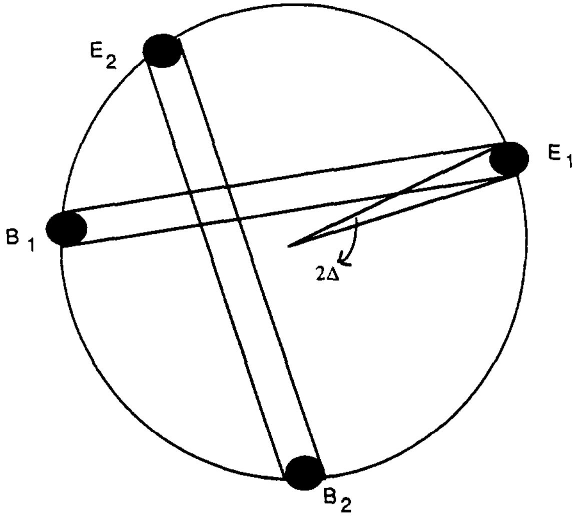


Fig. 4. Two robots of finite dimension moving inside a circular workspace. Robot R_i moves from B_i to E_i for $i = 1, 2$. The angle subtended at the center by each robot is 2Δ.

Since the robots are of finite dimensions, the initial position θ_{2b} of R_2 can take any random value in $[2\Delta, 2\pi - 2\Delta]$. Similarly, θ_{2e} can take any random value in $[0, 2\pi] - [\theta_{1e} - 2\Delta, \theta_{1e} + 2\Delta]$. Clearly, this model is true only if $\Delta \ll \pi/2$.

The paths of the robots will not intersect if θ_{2b} and θ_{2e} are on the same side of the arc $(0, \theta_{1e})$. By including the thicknesses of the robots, it can be seen that for the upper segment

$$\Pr\left[\text{not intersecting paths in upper segment} \mid \theta_{1e}\right] = \frac{(\theta_{1e} - 4\Delta)^2}{(2\pi - 4\Delta)^2}.$$

Hence the probability of having nonintersecting paths in the upper segment is

$$q_1 = \frac{1}{2\pi} \int_0^{2\pi} \frac{(\theta_{1e} - 4\Delta)^2}{(2\pi - 4\Delta)^2}\, d\theta_{1e},$$

which simplifies to $(\pi^2 - 6\Delta\pi + 12\Delta^2)/3(\pi - 2\Delta)^2$.

The probability of having nonintersecting paths in the lower segment has exactly the same value. Hence, the collision index p for this case is

$$p = 1 - 2q_1,$$

which yields the desired result. ■

We shall now proceed to consider the multirobot case.

III. MULTIROBOT MOTIONS

Until now we have considered the problem of two mobile robots navigating in a workspace $\mathfrak{W}$, and we obtained a quantitative measure for the efficiency of their traversals. Indeed, if the collision index of robots in $\mathfrak{W}$ is p, the mean time taken for the two robots to execute a job was shown to be

$$\xi_2 = \frac{1+p}{2}.$$

The problem of multiple robots working in a common workspace would be far too extensive if all possible geometries of robots and workspaces were to be considered. Rather than tackle the problem from this perspective, we shall disengage ourselves from the actual geometries of the robots and the workspace, and concentrate on the cluttering problem as a function of the collision index p. Thus, the question which we shall consider for the rest of the paper can be phrased: "If the collision index of two robots in a workspace is p, how quickly can k robots do a job?" Additionally, the productivity of the robots in this scenario can be investigated.

Extending these concepts to finite-dimension robots is not so direct, because one must consider the possibility of deadlock. Thus, for example, in the case of finite-dimension robots, if the area of the robot is R and the area of the workspace is A, then one might expect that when $k = A/R$ the whole situation would be deadlocked. Indeed, in Theorem 4 this situation is conceptually taken into account by the collision index, because the probability of intersection increases with the size of the robot. Observe that in Theorem 4, the condition for the calculation of the collision index (i.e., $\Delta \ll \pi/2$) is that the area of the robots is small compared with the area of the workspace. It is for this approximation that the arguments below would be useful for robots of finite dimension.

We shall now describe the two scheduling modes of operation that we consider in this paper.

(i) *Batch-scheduled mode.* In this mode we assume that there is a continuing stream (i.e. an infinite number) of jobs to be done. A scheduler assigns to each robot $R_i (i = 1$ to $k)$ a job which is essentially one of going from where it is, at position B_i, to a destination point E_i. A *Supervisor* commands the individual robot R_j to go whenever the path $B_j \rightarrow E_j$ is free. Obviously the task of the supervisor is to command the various robots to go in such a way that all the k jobs are done as quickly as possible. Once all the k jobs are completed, the *scheduler* assigns k new jobs to the robots.

(ii) *List-scheduled mode.* In this mode too we assume that there is a continuing stream of jobs to be done. Each robot $R_i (i = 1$ to $k)$ is assigned a job by the scheduler, and the supervisor tries to get the maximum number done at *that* time instant. The scheduler comes into operation at the next time instant and assigns new jobs only to those robots with have just finished their jobs.

It may seem intuitively clear that the list-scheduled mode of operation is the superior mode because the robots are given no ''resting'' time. However, whereas the list-scheduled mode of operation requires the scheduler to be invoked at every time instant, the batch-scheduled mode requires the scheduler to be invoked only when all the k jobs currently assigned to the robots are accomplished. The latter may be a desirable mode of operation, especially if the scheduler involves the intervention of a human being.

We would emphasize to the reader the differences in operation between the scheduler and the supervisor. Once the scheduler assigns the task to the individual robots (be they operating in a batch-scheduled mode or in a list-scheduled mode), the supervisor takes over and specifies the sequence in which the robots should do their jobs. At the risk of being repetitive we reiterate that whereas in the batch-scheduled mode of operation the scheduler comes back on the scene only after all k current jobs have been accomplished, in the list-scheduled mode the scheduler is invoked at the beginning of every time unit to assign tasks to the robots that have just finished their previously assigned tasks.

Let us suppose k robots are required to execute k jobs. We shall define T_k to be the mean time for the k robots to execute all k jobs in the batch-scheduled mode of operation. Let ξ_k be the mean time per job in this mode with k robots at work. Clearly, $\xi_k = T_k / k$.

Let N_k be the mean number of tasks that can be done per unit time in the list-scheduled mode. Then μ_k, the mean time per job in the list-scheduled mode with k robots at work, is given by $\mu_k = 1/N_k$. Observe that whereas T_k and N_k are quite unrelated, ξ_k and μ_k are analogous quantities for the two distinct scheduling modes of operation.

We also define the productivity index of k robots, λ_k, as the number of jobs done per robot per unit time. Clearly,

$$\lambda_k = \begin{cases} 1/k\xi_k & \text{in the batch-scheduled mode} \\ 1/k\mu_k & \text{in the list-scheduled mode.} \end{cases}$$

Note that the ratio of the productivity in the list-scheduled mode to the productivity in the batch-scheduled mode is just ξ_k/μ_k. This ratio will be used to compare the efficiencies of the two modes.

One of the important features of our approach is that, as we mentioned earlier, we have aimed to disengage ourselves from the actual geometries of the workspace and the robots, primarily because the problem would be intractable if all possible geometries were to be considered. However, to render even this approach tractable we must make two approximating assumptions, which we shall refer to as assumptions A1 and A2:

Al. All path patterns of k robots involving all possible intersections of robot pairs are geometrically realizable.

A2. For any path pattern, the intersection of a pair of robot paths is independent of the intersection of another pair of robot paths.

Assumption A1 is clearly not true for all values of k, because for certain values of k and for certain geometries, all combinations of patterns do not exist. For example, let $F = \{l_1, l_2, l_3, l_4, l_5\}$ be a collection of 5 line segments, and $G = \{m_1, \ldots, m_{10}\}$ be a collection of 10 line segments. Let H be an embedding of the 15 lines in the plane such that all elements of F are disjoint and all elements of G are disjoint, but each element of G intersects a distinct pair of elements of F. Then H cannot be realized. The proof of the claim is quite straightforward. Suppose that H exists. For this embedding, pick a point in each line $l \in F$ and gradually contract l to this point, pulling the lines that intersect l into this point. We can do this continuously for all elements of F without causing any lines to cross, because this contraction is a continuous map. Then the result is a planar graph of five vertices such that every vertex is joined to every other vertex (by the deformed lines in G). But this is a planar embedding of the complete graph on five vertices, which is not planar. Thus no such embedding H exists. Therefore, in the case of 15 robots working in an infinite workspace (the entire plane) there is at least one path pattern which cannot be realized. This counterexample is due to the anonymous referee mentioned in the acknowledgements.

More generally, for any nonplanar graph of V vertices and E edges, there is a combination of $V+E$ line segments which represents the graph and cannot be realized in the plane.

The validity of assumption A1 depends on the geometry of the workspace as well. For example, if five robots operate between points on the circumference of a circle, moving in the interior, we find by enumeration that one of the distinct path patterns is not physically realizable.

Although assumptions A1 and A2 are theoretically unrealistic, these approximating assumptions are made because the explicit expressions encountered in the analysis of the scheduling are significantly simplified if they are assumed valid. Furthermore, computer simulation results show that for all computationally realistic values for the number of robots, the actual experimental results obtained reproduce quite well the features of the analytic results obtained using the approximating assumptions A1 and A2.

Since the problem completely changes for every value of k, we shall first treat the cases of $k=3$ and $k=4$ separately. The case of the general value of k will be considered subsequently. We shall then conclude this section by stating some experimental results and conjecturing that the decision problem "Are $k+1$ robots better than k?" has an affirmative answer.

For the rest of the paper we shall assume that if p is the collision index, then $q=1-p$.

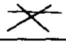
THEOREM V. *Let the collision index for two robots operating in a random workspace be p, and let $q=1-p$. Then, given assumptions $A1$ and $A2$, for three robots operating in a batch-scheduled mode, ξ_3, the mean time per job, is*

$$\xi_3=\frac{q^3+6pq^2+6p^2q+3p^3}{3}.$$

Proof. The result follows by enumerating the various ways by which the paths of the three robots can intersect. In Table 1, these distinct possible path patterns are enumerated, and the probability of the pattern occurring and the time for execution of such jobs is given. Observe that when all the paths intersect [this happens with probability p^3, since all the C(3, 2) pairs must intersect], the jobs must be executed serially, i.e., in three time units. Similarly, if none of the paths intersect (this happens with probability q^3), the jobs can be done in parallel in a single time unit. By enumerating all the possibilities we observe that T_3, the mean time for executing three jobs, is $T_3=q^3+6pq^2+6p^2q+3p^3$, whence the result follows. ■

TABLE 1

All Distinct Intersection Patterns for Three Robots Operating in the Batch-Scheduled Mode, Along with the Probability of the Pattern Occurring, the Number of Intersections, and the Execution Time Associated with Each Pattern

Pattern	Probability	Intersections	Execution time
	q^3	0	1
	$3pq^2$	1	2
	$3p^2q$	2	2
	p^3	3	3

REMARK. Let the collision index for two robots operating in a random workspace be p, and let $q = 1 - p$. Then, for three robots operating in a *list-scheduled* mode, μ_3, the mean time per job, can analogously proved to be

$$\mu_3 = \frac{1}{3q^3 + 6pq^2 + 6p^2q + p^3}.$$

In an analogous way we shall now derive the expression for ξ_4.

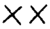
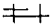
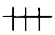
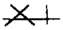
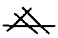
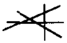
THEOREM VI. *Let the collision index for two robots operating in a random workspace be p, and let $q = 1 - p$. Then, if assumptions $A1$ and $A2$ are satisfied, for four robots working in a batch-scheduled mode of operation, ξ_4, the mean time per job is*

$$\xi_4 = \frac{q^6 + 12pq^5 + 30p^2q^4 + 44p^3q^3 + 42p^4q^2 + 18p^5q + 4p^6}{4}.$$

Proof. Consider Table 2, in which we have enumerated all the possible distinct ways by which four robots' paths could intersect or not intersect. Clearly A1 is satisfied. In the table, apart from the possible path patterns and their probability of occurring, we have also listed the execution times for these jobs. As before, observe that when none of the paths intersect (this happens with probability q^6), the execution time is unity, since all the jobs can be done in parallel. Similarly, when all the paths intersect (this happens with probability p^6), the execution time is 4. In between these is the whole spectrum of possibilities in which the execution times are 2 or 3, and the coefficients range from 4 to 44. By enumerating all these possibilities one obtains the result. ■

TABLE 2

All the Distinct Intersection Patterns for Four Robots Along with the Probability of the Pattern Occurring, the Number of Intersections, and the Execution Time Associated with Each Pattern

Pattern	Probability	Intersections	Execution time
	q^6	0	1
	$6pq^5$	1	2
	$12p^2q^4$	2	2
	$3p^2q^4$	2	2
	$12p^3q^3$	3	2
	$4p^3q^3$	3	2
	$4p^3q^3$	3	3
	$12p^4q^2$	4	3
	$3p^4q^2$	4	2
	$6p^5q$	5	3
	p^6	6	4

REMARKS.

1. There are two reasons why we have painstakingly enumerated all the possibilities to derive Theorem VI. First of all, we would like to highlight the sixth and seventh rows of the table, in which the number of intersecting paths is three. Whereas the time for execution for the jobs in row 6 is two time units, the time for execution for the jobs in row 7 is three time units—primarily because three of the robot paths are completely interwoven.

2. In the case of list-scheduled robots, the corresponding quantity μ_4, which is the mean time per job for four robots, is

$$\mu_4 = \frac{1}{4q^6 + 18pq^5 + 42p^2q^4 + 44p^3q^3 + 30p^4q^2 + 12p^5q + p^6}$$

if assumptions A1 and A2 are satisfied. Clearly, in this case, A1 is satisfied.

We shall now consider the general case of k robots at work in a common workspace.

THEOREM VII. *Let k robots be working in a workspace in a batch-scheduled mode of operation, and let the collision index be p. Further, let $q = 1 - p$. Then, if assumption $A1$ and $A2$ are valid, an upper bound on the time complexity of any algorithm that computes ξ_k is exponential, because it could involve evaluating an exponential number of decision problems each of which is NP-complete.*

Proof. We shall prove this result by first showing that if k robots are operating in a workspace, the number of distinct possible path patterns is exponential in k^2. We shall then show that the computation of the execution time for a *single* path pattern is essentially an NP-complete problem.

Suppose k robots are operating in a common workspace. Clearly, given assumptions A1 and A2, the path of any robot R_i can either intersect or not intersect the path of any other robot R_j. Since there are $C(k, 2)$ such pairs, and since each pair is either intersecting or nonintersecting, the total number of path patterns that can be encountered is $2^{C(k,2)}$, where $C(k, 2) = k(k-1)/2$, which is $O(2^{k^2})$.

We shall now compute ξ_k by individually examining every one of these path patterns, and shall prove that the evaluation of the time for an arbitrary path pattern is the solution of an NP-complete problem. Let ω be any path pattern. We create $G_\omega = (V, E)$, the intersection graph from ω, as follows:

(a) V, the vertices of the graph, are the indices $\{1, 2, \dots, k\}$.

(b) An undirected edge $(i, j) \in E$ if the path of R_i intersects the path of R_j.

Consider the coloration of the graph G_ω. A *coloration* of a graph is an assignment of a color to each vertex such that no two vertices joined by an edge get the same color. A k-coloration is a coloration using k colors. Observe that we can schedule the robots in k steps if and only if the adjacency graph has a k-coloration. We do this by identifying each color with a distinct time step. Clearly, if there is a k-coloration, we can move the robots in k steps, because at each step we are moving robots of the same color, none of which are joined by an edge. Conversely, any scheduling of the robots in k steps must also be a k-coloration of the graph. The theorem follows, since computing the minimum coloration (the chromatic number) of a general graph is an NP-complete problem [35]. ■

REMARK. Observe that we have resorted to a brute force enumeration technique to evaluate the quantity ξ_k, namely that of computing the time taken for each path pattern individually and subsequently computing ξ_k as a weighted combination of these time durations. However, apart from the fact that the above bound is an upper bound for the time required to compute ξ_k, we also believe that it gives both an upper and a lower bound for the time. Thus, we conjecture that the best algorithm is the brute force algorithm suggested above. The rationale for the conjecture is as follows.

From the theory of elementary probability and expectations it is known that if $\mathfrak{X}$ is any discrete random variable,

$$E[\mathfrak{X}] = \sum_{X} X \Pr[\mathfrak{X} = X],$$

where the summation is over all possible values of the random variable $\mathfrak{X}$. In this case, we intend to find the expected time per job. To show that the summation has to be treated individually for every path pattern, we shall show that there exist values of execution times (not necessarily distinct) which have *distinct* associated probability measures. Conversely, and more importantly, there are possible path patterns which have the same associated probability measures, but which have very different execution times.

The first of these claims is easily proven, because when the paths of none of the robots intersect, the execution time is unity, and obviously the probability of this event is $q^{C(k,2)}$. Similarly, if all the paths intersect, the execution time is k, and the probability associated with this event is $p^{C(k,2)}$. Thus, distinct execution times possess distinct associated probability measures.

To prove the second of the claims, let ω_1 be a path pattern in which there are H intersection points. If ω_1 is such that every robot path intersects *at most* one other robot path, the execution time is exactly 2. Thus the total contribution due to ω_1 in the computation of ξ_k is *exactly* $2p^Hq^{C(k,2)-H}$. Let ω_2 be another pattern in which the number of intersections is H, but in which the H intersections are due to a subset of λ_H robots, in which *subset* every robot path intersects every other robot path. Clearly, ω_2 is executed in time λ_H. Thus the contribution to the computation of ξ_k due to ω_2 is exactly $\lambda_H p^H q^{C(k,2)-H}$. (See Figure 5 for the specific case when $k = 6$ and $H = 3$.) Thus ω_1 and ω_2 are two path patterns with exactly the same associated probability measure but very different execution times.

Although the above remarks are true, they do not constitute a rigorous proof of the conjecture. The reason for this is that there may be a way by which patterns with "similar" (though not identical) probability measures may be clustered together to yield a partial mean, and this partial mean could ultimately yield the actual resultant ξ_k. However, by considering a variety of cases, we believe that no such clustering exists for general arbitrary path patterns.

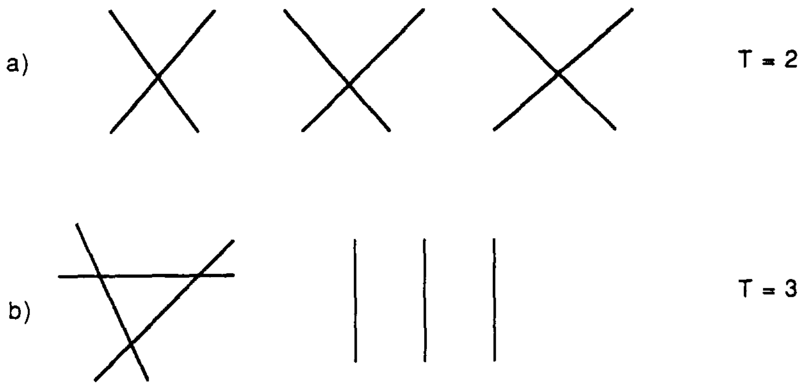


Fig. 5. Two cases for six robots: (a) three intersections with execution time of two units in the batch-scheduled mode; (b) three intersections with execution time of three units.

Remark 2. It is clear that if the assumption A1 is relaxed, the number of path patterns will definitely be less than what is claimed in Theorem VII. However, if we consider a set of k lines where each of the $k/2$ odd lines is infinite and horizontal and they are evenly spaced, and each of the $k/2$ even lines is finite and vertical, then each vertical line segment can intersect the horizontal lines in roughly $k^2/8$ ways. Thus there are $\Omega(k^k)$ [i.e., $\Omega(2^k \log {}^k)$] distinct intersection patterns for the k lines, which is also exponentially many. However, the results obtained by assuming A1 do not differ drastically from actual simulations that have been conducted for computationally feasible values of k.

For the sake of completeness, we shall state (without a detailed proof) the analogous theorem to Theorem VII for the list-scheduled robots.

Theorem VIII. *Let k robots be working in a workspace with list-scheduled jobs, and let the collision index be p. Further, let $q = 1 - p$. Then, if assumptions $A1$ and $A2$ are valid, an upper bound on the time complexity of any algorithm that computes μ_k is exponential, because it could involve evaluating an exponential number of decision problems each of which is NP-complete.*

Proof. The proof of the theorem is identical to that of Theorem VII except that the NP-complete problem encountered in the scenario is that of finding the size of the *largest* clique in G_ω, the complementary graph of G_ω defined in Theorem VII.

The problems which we have studied have been implemented on the SUN microsystem in order to get a feeling for the actual path planning scenario [37, 38]. They were programmed with p (the collision index) and k (the number of robots involved) as input parameters. Indeed, the number of computations involved grew exponentially as expected. For example, the time to computer μ_k in the list-scheduled case grew from a few tenths of a second when $k = 3$ to a few seconds when $k = 4$. It rose to about a half a minute when $k = 5$. The case for $k = 6$ took about 50 minutes, and the case for $k = 7$, which has 2,097,152 path patterns, required about 80 hours. We estimate that the case for $k = 8$ would require about a whole year of computing time on the Sun system. In the batch-scheduling mode, the case $k = 7$ proved infeasible.

The results of the computed values of ξ_k are plotted in Figure 6 for various values of p. The value is unity for all k whenever $p = 1$, and it is $1/k$ whenever $p = 0$. Clearly these two special cases report limiting behavior. When $p = 0$, the robots will never have intersecting paths, and so $k + 1$ robots will do better than k robots. When $p = 1$, the paths always intersect, and so it is *never* better to have more than one robot. The values of ξ_k are also tabulated in Table 3 for various other values of p. For example, when $p = \frac{1}{3}$, the value of ξ_k falls from 0.5802 when $k = 3$ to 0.5103 when $k = 4$ and to 0.4529 when $k = 5$. It attains the value of 0.4130 when $k = 6$. These values are typical, since the curves are monotonically decreasing for all computed values and seem to reach their asymptotes fairly rapidly. A value for $k = 7$ was also calculated using five point Lagrangian extrapolation, which results in a value of 0.4044. Based on these results, if ξ_k is the criterion, we conjecture that $k + 1$ robots is in general better than k robots for $p \in (0, 1)$.

Similarly, for the list-scheduled case, the values of μ_k are also tabulated in Table 4 for various values of p in (0, 1). For example, when $p = \frac{1}{3}$, the value of μ_k moves from 0.4426 when $k = 3$ to 0.3643 when $k = 4$, and to 0.3131 when $k = 5$. It attains the value 0.2787 when $k = 6$ and 0.2531 when $k = 7$. Again based on these results, if μ_k is the criterion, we conjecture that $k + 1$ robots is better than k robots for $p \in (0, 1)$ for the list-scheduled multiple robot scenario.

IV. ECONOMIC ANALYSIS OF THE EFFICIENCY OF MULTIPLE ROBOTS

Suppose k robots are to execute k jobs. Clearly, the average time taken per job with k batch-scheduled robots at work being ξ_k, the average time taken by the robot to accomplish all the k jobs is $k\xi_k$. Hence, the number of jobs done per unit time is $k/(k\xi_k)$, which is $1/\xi_k$. Similarly, for the list-scheduled case, the average time taken per job by k robots being μ_k, the average time taken by the robots to accomplish all the k jobs is $k\mu_k$. Furthermore, the average number of jobs done per unit time is N_k, which is $1/\mu_k$.

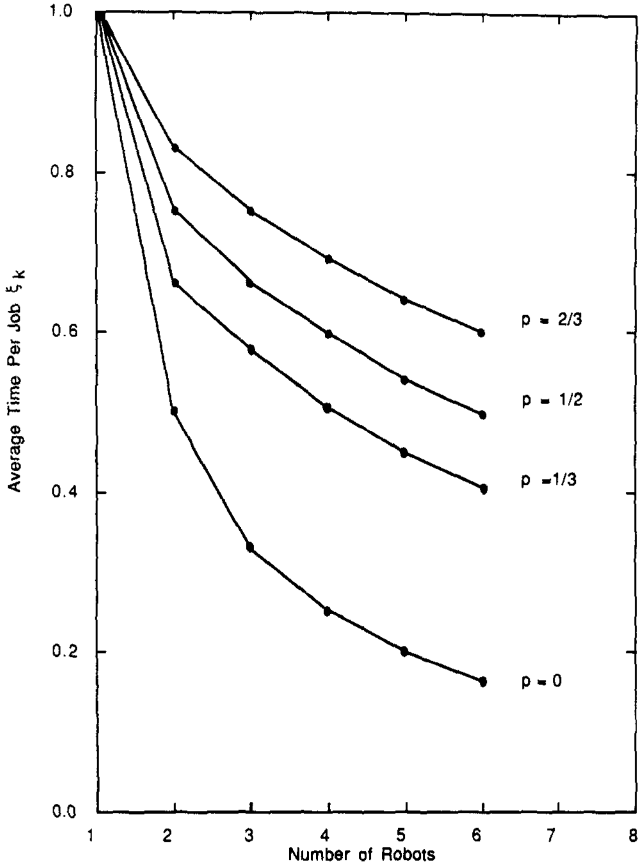


Fig. 6. Plot of average time per job, ξ_k versus number of robots, k, in the batch-scheduled mode. Note that for $p = 0$, $\xi_k = 1/k$.

TABLE 3

Results in the Batch-Scheduled Mode for the Average Time Taken by k Robots to Do a Job, ξ_k, for Various Collision Index Values p

	ξ_k				
k	$p=0$	$p=\frac{1}{3}$	$p=\frac{1}{2}$	$p=\frac{2}{3}$	$p=1$
1	1.0000	1.0000	1.0000	1.0000	1.0000
2	0.5000	0.6667	0.7500	0.8333	1.0000
3	0.3333	0.5802	0.6667	0.7531	1.0000
4	0.2500	0.5103	0.5898	0.6862	1.0000
5	0.2000	0.4529	0.5395	0.6401	1.0000
6	0.1667	0.4130	0.5011	0.5995	1.0000

TABLE 4

Results in the List-Scheduled Mode for the Average Time Taken by k Robots to Do a Job, μ_k, for Various Collision Index Values p

	μ_k				
k	$p=0$	$p=\frac{1}{3}$	$p=\frac{1}{2}$	$p=\frac{2}{3}$	$p=1$
1	1.0000	1.0000	1.0000	1.0000	1.0000
2	0.5000	0.6000	0.6667	0.7500	1.0000
3	0.3333	0.4426	0.5000	0.5745	1.0000
4	0.2500	0.3643	0.4238	0.4892	1.0000
5	0.2000	0.3131	0.3724	0.4429	1.0000
6	0.1667	0.2787	0.3350	0.4070	1.0000
7	0.1428	0.2531	0.3083	0.3768	1.0000

We define the productivity of each robot as the number of jobs done per robot per unit time. Clearly this is a function of p, the collision index, and has the value $1/(k\xi_k)$ in the batch-scheduled mode, with the corresponding value for the list-scheduled mode being $1/(k\mu_k)$. The productivity of each robot in the batch-scheduled mode is tabulated in Table 5 for $p=\frac{1}{3}$, while that for the list-scheduled mode is tabulated in Table 6 for the same value of p. The productivity is seen to *decrease* with increasing number of robots for both modes.

The ratio of the productivities for the robots operating in the list-scheduled mode to that in the batch-scheduled mode is tabulated in Table 7 for several values of k and $p \in [0, 1]$. The ratio is computationally always greater than

TABLE 5

Results in the Batch-Scheduled Mode for Collision Index $p=\frac{1}{3}$, Showing the Number of Robots, k; the Average Time Taken to Do a Job by k Robots, ξ_k; the Productivity Factor, $1/(k\xi_k)$; and the Profit per Unit Time (in Units of E) for Three Values of the Profit Factor f[a].

		Productivity	Profit/(time E)		
k	ξ_k	$1/(k\xi_k)$	$f=1$	$f=2.0$	$f=2.5$
1	1.0000	1.000	0.000	1.000	1.500
2	0.6667	0.750	−0.500	1.000	1.750
3	0.5802	0.575	—	0.447	1.309
4	0.5103	0.490	—	−0.081	0.899
5	0.4529	0.442	—	0.584	0.520
6	0.4130	0.404	—	—	0.053

[a] A negative value indicates a loss. Only the first loss in each column is noted.

TABLE 6

Results in the List-Scheduled Mode for Collision Index $p = \frac{1}{3}$, Showing the Number of Robots, k; the Average Time Taken to Do a Job by k Robots, μ_k; the Productivity Factor, $1/(k\mu_k)$; and the Profit per Unit Time (in Units of E) for Three Values of the Profit Factor f[a].

		Productivity	Profit/(time E)		
k	μ_k	$1/(k\mu_k)$	$f = 1$	$f = 1.5$	$f = 2.0$
1	1.0000	1.000	0.000	0.500	1.000
2	0.6000	0.833	−0.333	0.500	1.333
3	0.4426	0.753	—	0.389	1.519
4	0.3643	0.686	—	−0.117	1.490
5	0.3131	0.639	—	−0.209	1.388
6	0.2787	0.598	—	—	1.176
7	0.2531	0.565	—	—	0.903

[a] A negative value indicates a loss. Only the first loss in each column is noted.

unity whenever $k > 1$ and $p \in (0, 1)$. Based on these results, we conjecture that it is always better to operate within a list-scheduled mode.

If the actual costs of operating the robots are not taken into account, then the decline in productivity is not in and of itself a problem. This is because employing a greater number of robots implies a greater total production as long as ξ_k or μ_k continues to decline. However, if we make some simple economic assumptions about the costs and revenues involved, the question "Are $k + 1$ robots more profitable then k?" has some interesting answers.

Consider the following economic model. Let E be the expense involved for the robot plant to operate a single robot for a single time unit. Further, let C be the charge for a single job to be executed. Furthermore, let $C = fE$, where f is the profit factor. Clearly, if $f = 1$, the plant pays E units per robot time

TABLE 7

Ratio ξ_k/μ_k of the productivities in the List-Scheduled Mode to the Batch-Scheduled Mode for Five Collision Index Values p

	ξ_μ/k_μ				
k	$p = 0$	$p = \frac{1}{3}$	$p = \frac{1}{2}$	$p = \frac{2}{3}$	$p = 1$
1	1.00	1.00	1.00	1.00	1.00
2	1.00	1.11	1.12	1.11	1.00
3	1.00	1.31	1.33	1.31	1.00
4	1.00	1.40	1.39	1.40	1.00
5	1.00	1.45	1.45	1.45	1.00
6	1.00	1.48	1.50	1.47	1.00

unit and charges $C = E$ units for each job. If the plant has a certain profit policy (i.e., it charges according to a fixed profit factor f), how beneficial is it for the plant to get $k+1$ robots on the shop floor as opposed to having k robots?

Let us assume that k batch-scheduled robots are assigned to do the jobs. The number of jobs done by the k robots per unit time is $k/(k\xi_k) = 1/\xi_k$. The expense incurred by the plant for this period is kE. The tariff (or revenue) that has come in this unit of time is $C\cdot(1/\xi_k)$, essentially, because $1/\xi_k$ was the number of jobs done. Hence the net profit (gain) G made per unit time is

$$G = \frac{C}{\xi_k - kE}\,.$$

If f is the profit factor, we observe that, $G = fE/\xi_k - kE$, whence $G = E(f - k\xi_k)/\xi_k$. Clearly, if the plant is to remain operational, $G \geq 0$.

For the extreme case of $p = 0$ (no possibility of any of the paths intersecting), since the average time per job is $\xi_k = 1/k$, the profit per unit time is given by

$$G = E(f-1)k.$$

Thus for any fixed profit factor $f > 1.0$, the profit per unit time increases indefinitely with k. In this case, then, the more robots used, the more profit earned. Note that if $f = 1$, we will have a breakeven situation.

For the other extreme case of $p = 1$ (each path intersects every other path) the average time per job is $\xi_k = 1$, independent of k. Hence the profit per unit time is given by

$$G = E(f-k).$$

In this case, for any fixed f, the profit is always maximized with one robot, with a loss situation occurring at $k = f$. So there is no reason to have more than one robot. The situation for more realistic values of p is illustrated for $p = \frac{1}{3}$ (the case of the circular workspace) in Figure 7, where G/E is plotted for various values of the profit factor f. If $f = 1$, one robot is the optimum number (at the breakeven point), with any more robots leading to a loss. If $f = 2$ (i.e., if the policy is to charge per job twice the amount that the plant pays per robot unit time), then it is detrimental to have more than three robots. In this case, the optimum number of robots is either one or two. Similarly, if $f = 2.5$, it is detrimental to have more than six robots. The optimum number in

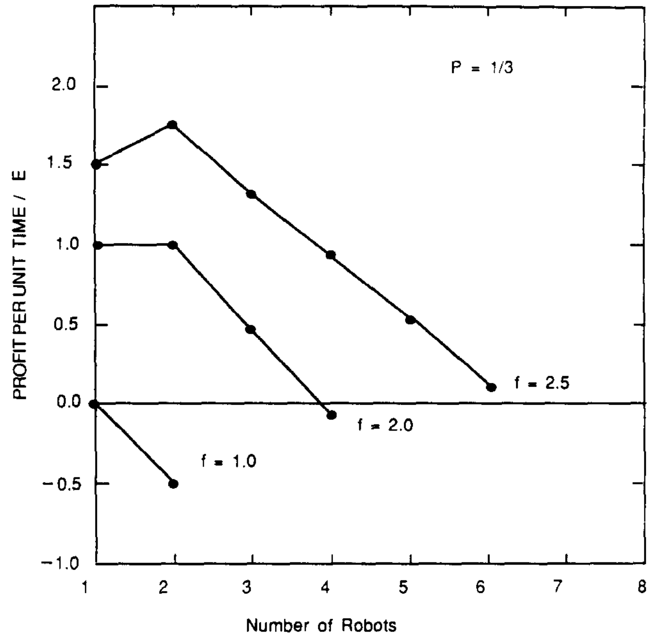


Fig. 7. Plot of profit per unit time (in units of the cost per robot per unit time, E) as a function of the number of batch-scheduled robots for profit factors of $f = 1.0$, 2.0, and 2.5. The collision index p is $\frac{1}{3}$. Negative values indicate a loss.

this case is exactly two. Such results are typical. A tabulation of these results is given in Table 5.

Analogously, in the list-scheduled case the net profit G made per unit time is

$$G = CN_k - kE = \frac{C}{\mu_k} - kE,$$

where N_k is the average number of jobs done per unit time with k robots at work. Again if f is the profit factor,

$$G = E(fN_k - k) = E\frac{f - k\mu_k}{\mu_k}.$$

As in the batch-scheduled case, for the extreme case of $p = 1$ the optimum number of robots is 1, and if $p = 0$, the profit earned increases indefinitely with k. For a more realistic value of p (say $p = \frac{1}{3}$, the case of the circular workspace), we have plotted in Figure 8 the ratio of G/E as a function of f. If $f = 1$, one robot is the optimum number (at the breakeven point), with any more robots leading to a loss. If $f = 1.5$ (i.e., if the policy is to charge per job one and one-half times the amount that the plant pays per robot time unit), then it is detrimental to have more than four robots. In this case, the optimum number of robots is either one or two. Similarly, if $f = 2$, it is possibly detrimental to have more than nine robots. The optimum number in this case is exactly three. A collection of these results is tabulated in Table 6. In every case the results strength our conjecture that the list-scheduled mode is superior to the batch-scheduled mode. However, if the invoking of the scheduler at every time instant is an expensive operation, it may be advantageous to use the batch-scheduled mode of operation, especially if k is small.

We conclude this section by making some observations about assumptions A1 and A2. Clearly, as shown earlier, these assumptions are not always valid. However, for practical purposes they are reasonable in that the results derived using them do not vary significantly from actual simulation results. To check this we have done a simulation where the robot paths are generated randomly

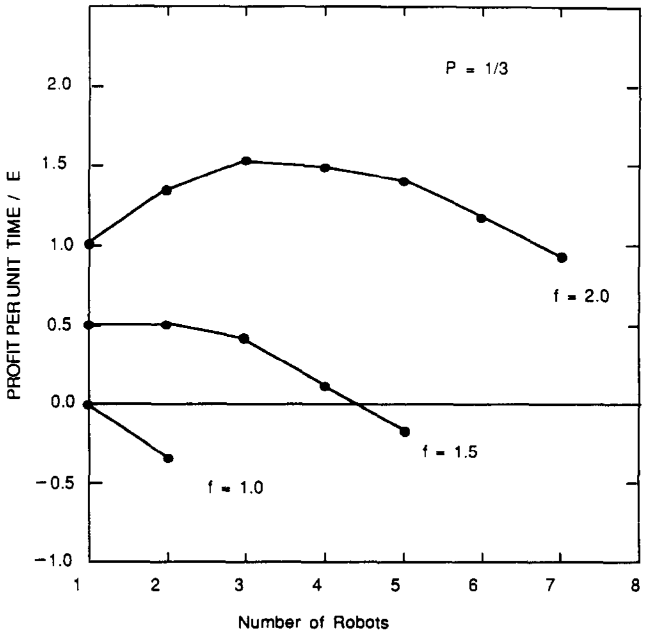


Fig. 8. Plot of profit per unit time (in units of the cost per robot per unit time, E) as a function of the number of list-scheduled robots for profit factors of $f = 1.0$, 1.5, and 2.0. The collision index p is $\frac{1}{3}$. Negative values indicate a loss.

between points on the circumferance of a circle. This corresponds to a workspace with collision index p of $\frac{1}{3}$. In this case we can see by enumeration that A1 is not fully justified, because, as we have noted previously, some of the path patterns are not geometrically realizable for $k = 5$ and greater. However, in practice, although our assumptions are not exact, the degree of approximation is warranted. To be more specific, using batch scheduling, the values for ξ_k obtained in the simulation are 0.668, 0.579, 0.518, 0.474, and 0.439 for $k = 2$, 3, 4, 5, and 6 robots respectively, based on 100,000 trials; these results agree reasonably well with the case $p = \frac{1}{3}$ tabulated in Table 3. Of course, the agreement becomes worse with increased k. Similar results obtain for the list-scheduled mode.

CONCLUSION

In this paper we have considered the general problem of moving k robots in a workspace. Using a general model in which the probability of two robots paths intersecting is a quantity p, the decision problem "Are $k+1$ robots better than k?" has been studied using two approximating assumptions. The problem has been studied for the cases when the robots are operating in either a batch-scheduled or a list-scheduled mode of operation. We have shown that the upper bound time complexity of this problem requires the solution of an exponential number of NP-complete problems. In the batch-scheduled case the NP-complete problem is the computation of the chromatic number of a graph G_ω, and in the list-scheduled case the NP-complete problem involves the computation of the maximal clique of the complimentary graph of G_ω.

We have concluded the paper by considering the economic issues at stake. We have shown that if the plant charges $C = fE$ per job, where E is the cost per robot time unit, then, for a given profit factor f, there is an optimal number of robots to be used. This optimal number has been obtained for various values of f. In this scenario, it is clear that $k+1$ robots need not be better than k for either mode of operation.

Finally we have shown the outcome of a simulation to indicate that the effect of our approximations need not seriously affect the nature of our results.

The authors are extremely grateful to anonymous reviewers who reviewed an earlier version of the paper and pointed our various technical drawbacks. The quality of this version was significantly enhanced by their comments.

REFERENCES

1. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Mass., 1974.

2. R. A. Brooks, Visual map making for a mobile robot, in *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, pp. 824-829.
3. R. A. Brooks and T. Lozano-Perez, A subdivision algorithm in configuration space for findpath with rotation,'' *IEEE Trans. Systems Man Cybernet.* SMC-15: 224-233, (Mar./Apr. 1985).
4. J. F. Canny, Collision Detection for Moving Polyhedra, A. I. Memo 806, M.I.T. Artificial Intelligence Lab., Oct. 1984.
5. R. Chatilla, Path planning and environment learning in a mobile robot system, in *Proceedings of the European Conference Artificial Intelligence*, Orsay, France, 1982.
6. R. Chattergy, Some Heuristics for the navigation of a robot, *Internat. J. Robotics Res.* 4:59-66 (Spring 1985).
7. J. L. Crowley, Navigation of intelligent mobile robot, *IEEE J. Robotics and Automation* RA-1:31-41 (Mar. 1985).
8. M. Erdmann and T. Lozano-Perez, On multiple moving objects, in *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, 1986, Vol. 3, pp. 1419-1424.
9. P. C. Gilmore, R. E. Gomory, A linear programming approach to the cutting stock problem, *Oper. Res.* 9:849-887 (1961).
10. G. Giralt, R. Sobek, and R. Chatila, A multilevel planning and navigation system for a mobile robot, in *Proceedings of the 6th International Joint Conference on Artificial Intelligence*, Tokyo, 1979, pp. 335-338.
11. L. Gouzenes, Strategies for solving collision-free trajectory problems for mobile and manipulator robots, *Internat. J. Robotics Res.* 3:51-65 (Winter 1984).
12. D. D. Grossman, R. C. Evans and P. D. Summers, The value of multiple independent robot arms, *Robotics and Comput. Integrated Mfg.* 2:135-142 (1985).
13. J. E. Hopcroft, J. T. Schwartz, and M. Sharir, On the complexity of motion planning for multiple independent objects; PSPACE-hardness of the ''waterhouseman's problem,'' *Internat. J. Robotics Res.*, 1984, pp. 76-88.
14. S. S. Iyengar, C. C. Jorgensen, S. V. N. Rao, and C. R. Weisbin, Robot Navigation Algorithms Using Learned Spatial Graphs, ORNL Technical Report TM-9782, Oak Ridge National Lab., Oak Ridge, Tenn., Aug. 1985.
15. S. S. Iyengar, C. C. Jorgensen, S. V. N. Rao, and C. R. Weisbin, Learned navigation paths for a robot in unexplored terrain, in *Proceedings of the 2nd Conference Artificial Intelligence Applications and Engineering of Knowledge Based Systems*, Miami Beach, Fla., 1985, pp. 148-155.
16. K. Kant and S. W. Zucker, Trajectory Planning in Time-Varying Environments, 1: TPP = PPP + VPP, TR-84-7R, Computer Vision and Robotics Lab., McGill Univ., Montreal, 1984.
17. J. Laumond, Model structuring and concept recognition: Two aspects of learning for a mobile robot, in *Proceedings of the 8th Conference Artificial Intelligence*, Karlsruhe, West Germany, 1983, p. 839.
18. D. T. Lee and F. P. Preparata, Computational Geometry—a survey, *IEEE Trans. Comput.* C-33, (12):1072-1101 (Dec. 1984).
19. T. Lozano-Perez, Spatial planning: A configuration space approach, *IEEE Trans. Comput.*, C-32:108-120 (Feb. 1983).
20. T. Lozano-Perez and M. A. Wesley, An algorithm for planning collision-free paths among polyhedral obstacles, *Comm. ACM* 22 (10):560-570 (Oct. 1979).
21. H. P. Moravec, The CMU Rover, in *Proceedings of the National Conference Artificial Intelligence*, Aug. 1982, pp. 377-380.

22. B. J. Oommen, S. S. Iyengar, N. S. V. Rao, and R. L. Kashyap, Robot navigation in unknown terrains using learned visibility graphs. Part 1: The disjoint convex obstacle case, *IEEE Trans. Robotics and Automation*, 1987, pp. 672-681.
23. B. J. Oommen and I. Reichstein, On the problem of translating an elliptic object through a workspace of elliptic obstacles, *Robotica* 5:187-196 (1987).
24. F. Ozguner, S. J. Tsai, and R. B. McGhee, An approach to the use of terrain-preview information in rough-terrain locomotion by a hexapod walking machine, *Internat. J. Robotics Res.* 3:134-146 (1984).
25. G. Ramanathan and V. S. Alagar, Algorithmic motion planning in robotics: coordinated motion of several disks amidst polygonal obstacles, in *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, 1985, pp. 514-522.
26. S. V. N. Rao, S. S. Iyengar, C. C. Jorgensen, and C. R Weisbin, Concurrent Algorithms for Autonomous Robot Navigation in an Unexplored Terrain, Tech. Rep. 85-048, Dept. of Computer Science, Louisiana State Univ., 1985.
27. J. T. Schwartz and M. Sharir, On the piano movers' problem: III. Coordinating the motion of several independent bodies: The special case of circular bodies amidst polygonal barriers, *Internat. J. Robotics Res.* 2:46-75 (1983).
28. L. Y. Shih, Automation guidance of mobile robots in two way traffic, *Automatica* 21:193-198 (1985).
29. A. M. Thompson, The navigation system of the JPL robot, in *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Cambridge, Mass. 1977, pp. 749-757.
30. M. P. Turchen and A. K. C. Wong, Low level learning for a mobile robot: Environmental model acquisition," in *Proceedings of the 2nd International Conference on Artificial Intelligence Applications*, 1985, pp. 156-161.
31. S. M. Udupa, Collision Detection and avoidance in computer controlled manipulators, in *Proceedings of the 5th International Conference Artificial Intelligence*, M.I.T., Cambridge, Mass., Aug. 1977, pp. 737-748.
32. C. R. Weisbin, J. Barhen, G. DesSaussure, W. R. Hamel, C. Jorgensen, E. M. Oblow, and R. E. Ricks, Machine intelligence for robotics applications, in *Proceedings of the 1985 Conference on Intelligent Systems and Machines*, 1985.
33. S. Whitesides, Computational geometry and motion planning, in *Computational Geometry* (G. Toussaint, Ed.), North Holland, 1985.
34. C. K. Yap, Coordinating the Motion of Several Disks, Tech. Report 105, New York Univ. Computer Science Dept., Courant Inst. of Mathematical Sciences, 1984.
35. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.
36. D. D. Grossman, Traffic Control of Multiple Robot Vehicles, IBM Report RC 12028, July 1986.
37. I. Reichstein and B. J. Oommen, Computational issues in determining the optimal number of mobile robots operating in a common workspace, in *Proceedings of the Eighteenth Annual Pittsburgh Conference on Modelling and Simulation*, Pittsburgh, Apr. 1987, pp. 845-854.
38. B. J. Oommen and I. Reichstein, On batch scheduled multiple mobile robots cluttering a workspace, in *Proceedings of the 26th IEEE Conference on Decision and Control*, Los Angeles, Dec. 1987, pp. 645-648.

Received 10 August 1989; revised 5 February 1990