Multiple path coordination for mobile robots: a geometric algorithm

S. Leroy, J.P. Laumond, T. Simeon
LAAS-CNRS
7, avenue du Colonel Roche
31077 Toulouse Cedex 4 - France
{sleroyjpl,nic}@laas.fr

Abstract

This paper presents a geometric based approach for multiple mobile robot motion coordination. All the robot paths being computed independently, we address the problem of coordinating the motion of the robots along their own path in such a way they do not collide each other. The proposed algorithm is based on a bounding box representation of the obstacles in the so-called coordination diagram. The algorithm is resolution-complete. Its efficiency is illustrated by examples involving more than 100 robots.

1 Introduction: Path coordination

This paper addresses the following problem: consider *n* mobile robots sharing the same workspace and planning their paths independently; *n* such paths being given we want to devise an algorithm deciding whether coordinated motions exist for the mobile robots along their own paths, so that each robot can reach its own goal without colliding the other ones. The problem is known as the multiple robot path coordination problem [Latombe, 1991b].

Path coordination *versus* Path planning Multiple robot path coordination and path planning are two related issues in robot motion planning. In multiple robot path planning the robot paths are not *a priori* computed. A solution to the multiple robot path planning problem is a collision-free path in the cartesian product of the configuration spaces of all the robots. A solution to the problem exists *iff* the start and goal configurations belong to a same connected component of the global collision-free configuration space. Searching such a space is a highly combinatorial problem [Hopcroft *et* a/., 1984].

To face this complexity several authors have investigated decoupled schemes[1]. The decoupled approach has been introduced in [Kant and Zucker, 1986]: the method first plans the paths of the robots independently and then computes the velocity profiles so that the robots do not collide. The approach has been further revisited in [Erdmann and Lozano-Perez, 1986; Buckley, 1989; Warren, 1990; Alami *et* a/., 1995].

The path coordination problem as such has been addressed in [O'Donnell and Lozano-Perez, 1989] where the notion of coordination diagram has been first introduced. It dealt with two robots, a case which has been also addressed in [Bien and Lee, 1992; Chang *et* a/., 1994]. A strategy based on dynamic programming was proposed more recently in [La Valle and Hutchinson, 1996] to address problems involving more than two robots.

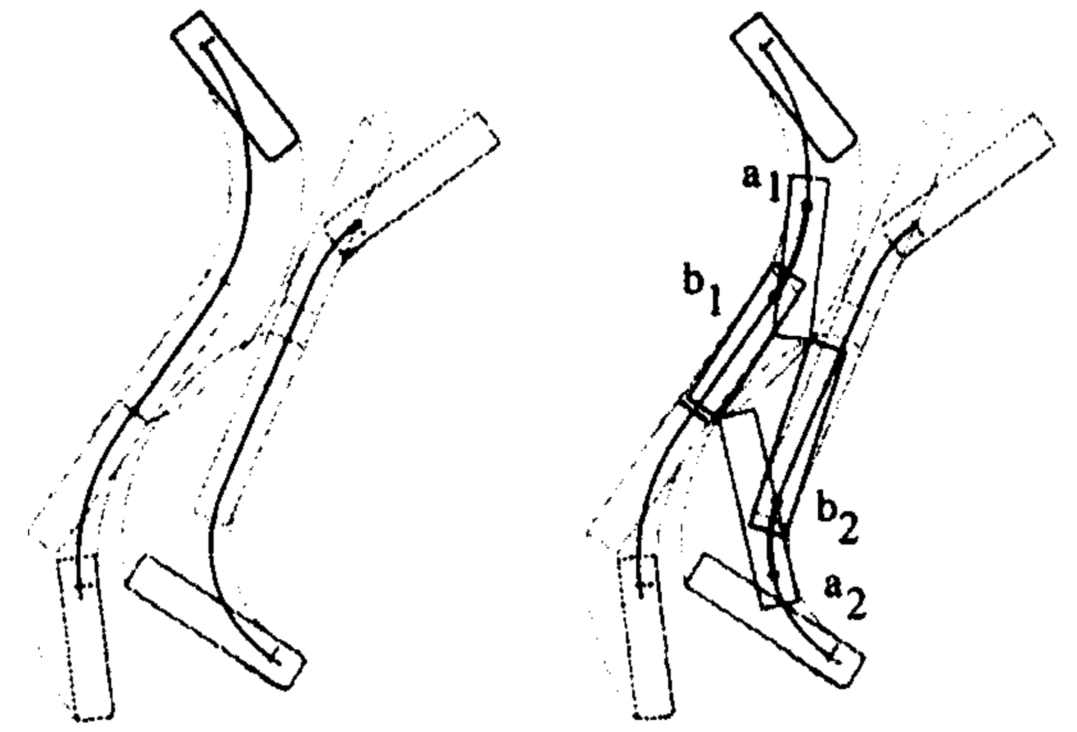
Objective, approach and contribution We want to solve problems involving more than 100 robots in realistic situations. The algorithm consists in searching a n-dimensional coordination diagram. The main contribution is to propose a bounding box representation of the diagram obstacles. With respect to the previous works above we do not use any regular grid representation. The algorithm is resolution complete and it is complete for a large class of inputs. Its efficiency inherits from the efficiency of simple geometric operations giving rise to a collision-checker dedicated to mobile robot coordination and summarized in Section 2. After having introduced a cell decomposition of the coordination diagram for the case of two robots (Section 3), we extend the algorithm to the general case (Section 4).

2 Paths *SA* and geometric tools

Paths *SA* The geometric tools we use are based on the following assumption: the robot paths are sequences of straight line segments (S) and arcs of a circle (A). Such sequences are denoted by 5.4 . This assumption is supported by both theoretical and practical considerations.

[1] Other schemes for multiple robot path planning have been proposed. For instance some centralized approaches aim at facing the problem complexity with probabilistic algorithms (see [Svestka and Overmars, 1995] and references therein). Prom another point of view, cooperation-oriented approaches are based on local informations (potential methods): see for instance [Reif and Wang, 1995] and [Cao *et ai,* 1997] for a recent overview. Techniques for path coordination are out of the scope of all these methods.

First of all, it has been proved that a collision-free admissible path exists *iff* there exists a collision-free admissible path of type *SA* [Laumond, 1986]. Moreover, most of the existing complete motion planners for mobile robots provide solution paths of the type *SA* (e.g., [Laumond *et a*/., 1994; Latombe, 1991a; Svestka and Overmars, 1995; Mirtich and Canny, 1992]). Finally geometric algorithms like boolean operations or swept volume computations are simple and computationally efficient when dealing with arcs of circle and straight line segments.



(a) (b)

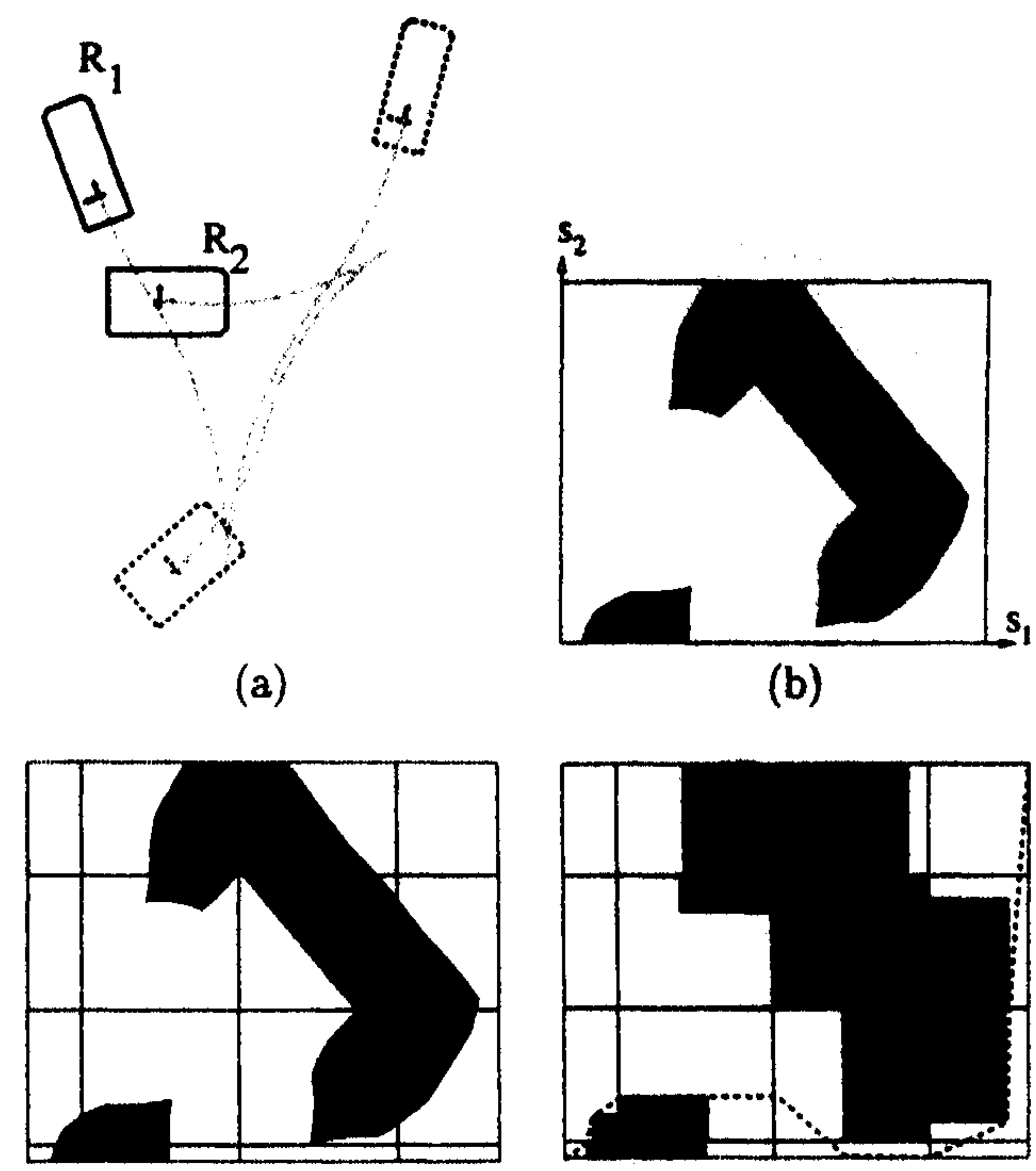
Figure 1: Two intersecting robot traces

Traces A mobile robot path being given, a *trace* is the volume swept by the robot when moving along the path. Assuming that the robot is a polygon, the trace of a path of type *SA* is a generalized polygon whose boundary is a sequence of straight line segments and arcs of a circle. [Simeon *et a*/., 1998] have shown how to compute such traces efficiently (Figure 1(a)).

Coordination configurations To coordinate the motions of two robots along their own path, it is necessary to compute the intersection of their trace. Figure 1 shows two traces. The bold sub-path [a_1,6i] (resp. [a_2,b_2]) gathers the configurations at which the first (resp. second) robot intersects the trace of the second (resp. first) one. The endpoints of such sub-paths are called *coordination configurations.* [Simeon *et a*/., 1998] have proposed a geometric algorithm to compute them when the robots are convex polygons and move along *SA* paths. In this paper we keep the same assumptions.

3 Coordination for two robots

Coordination diagram Coordinating the motion of two robots along two given paths is a classical problem. Its solution consists in exploring the so-called *coordination diagram* [O'Donnell and Lozano-Perez, 1989]. Let us consider the two paths $\gamma_1(s_1)$ and $\gamma_2(s_2)$ m Figure 2(a). Both coordinates s_1 and s_2 are assumed to vary from 0 to 1. Figure 2(b) shows the corresponding coordination diagram (s_1, s_2): the black domains represent the set of configuration pairs (s_1, s_2) such that the robots collide when they are respectively at configurations $\gamma_1(s_1)$ a $\gamma_2(s_2)$. Black domains are obstacles to avoid. A coordinated motion exists *iff* there is a collision-free path in the diagram linking the point (0,0) (the robots are both at the beginning of their own path) to the point (1,1) (the robots are both at the end of their path).



(c) (d)

Figure 2: Two *SA* paths (a), the coordination diagram (b), the partition of the diagram induced by the path decomposition (c), the bounding box representation of the obstacles and a solution path (d).

A bounding box representation Our contribution is to propose an algorithm to explore the diagram without computing the exact shape of the obstacles[2]. We use a bounding box representation based on the following property: *the (minimal) box bounding an obstacle in a coordination diagram is a rectangle whose endpoint coordinates are the coordination configurations*[3]. Let us consider the case in Figure 1. The coordinates of four points defining the rectangle in the coordination diagram are respectively (a_1, b_1), (a_1, b_2), (a_2, b_1) and

[2]The obstacles in Figure 2(b) have been computed with a brute force discretization approach used only for display purpose.

[3] In our context the coordinate of a configuration on a path γ is its curvilinear abscissa *s* on γ.

(a_2, b_2). The computation of the boxes is then done by computing the coordination configurations (see above).

Path decomposition Let us now consider two *SA* paths γ_1 and γ_2. Instead of applying the bounding box representation directly in the coordination diagram of γ_1 and γ_2, we first apply a path decomposition. Each path is decomposed into its elementary pieces consisting of either straight line segments, or arcs of a circle. Let $(\gamma_{1,i})$ and $(\gamma_{2,j})$ the pieces sequences of γ_1 and γ_2 respectively. The coordination diagram for γ_1 and γ_2 then appear as the union of the coordination diagrams of the various pairs $(\gamma_{1,i}, \gamma_{2,j})$. For instance, the two paths in Figure 2(a) both consist of 4 arcs of a circle. Therefore the coordination diagram appears as the union of 16 elementary coordination diagrams (Figure 2(c)). Then, for each elementary coordination diagram, we compute a bounding box representation of the obstacles. Figure 2(d) shows the bounding box representation of the diagram in Figure 2(b).

Search Such a representation induces a cell decomposition of the coordination diagram into rectangles. Any classical search algorithm may be used to compute a collision-free path from the origin (0,0) to the goal (1,1). Figure 2(d) shows a solution path. For this example, note that the widthest robot *R2* (corresponding to the vertical coordinate in the diagram) should *necessarily* move forward, backward and then forward along the first two pieces of its path.

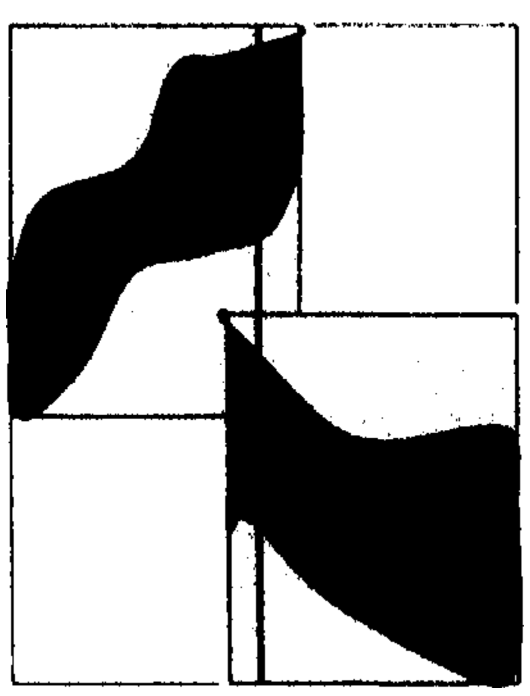


Figure 3: This case cannot appear when at least one robot moves along a straight line segment.

Completeness The algorithm is complete *iffit* is complete when applied to the elementary diagrams corresponding respectively to three cases: S||S, S||A, A||A.

For the first two cases the algorithm is complete. The only way for the bounding box approach to loose a solution is that there exist two vertical *and* horizontal lines intersecting two obstacles (Figure 3). This is however not possible since at least one robot moves along a straight line segment: indeed, the robot moving along the straight line cannot intersect *twice* the other (convex) robot remaining at a *fixed* position. Then the bounding box approximation does not affect the completeness of the algorithm for these first two cases.

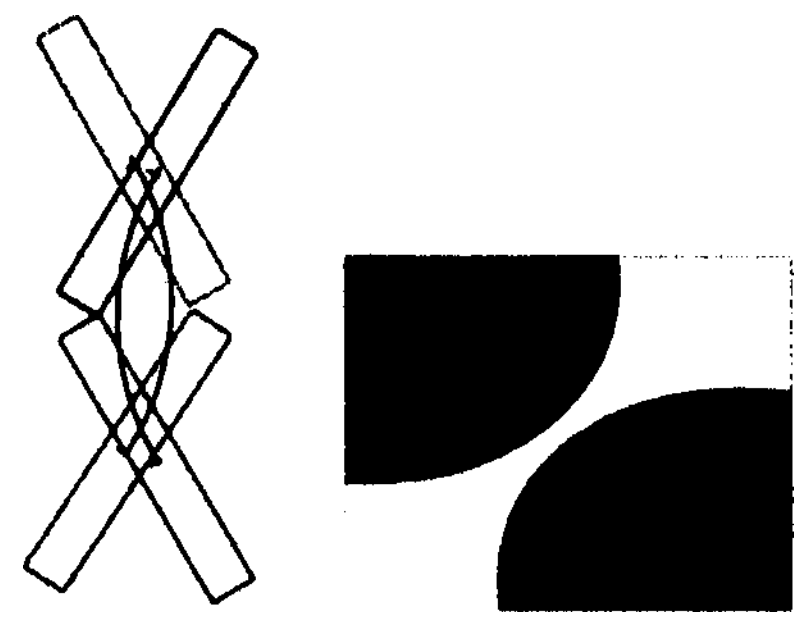


Figure 4: **A||A** special case: bounding boxes would fill the space.

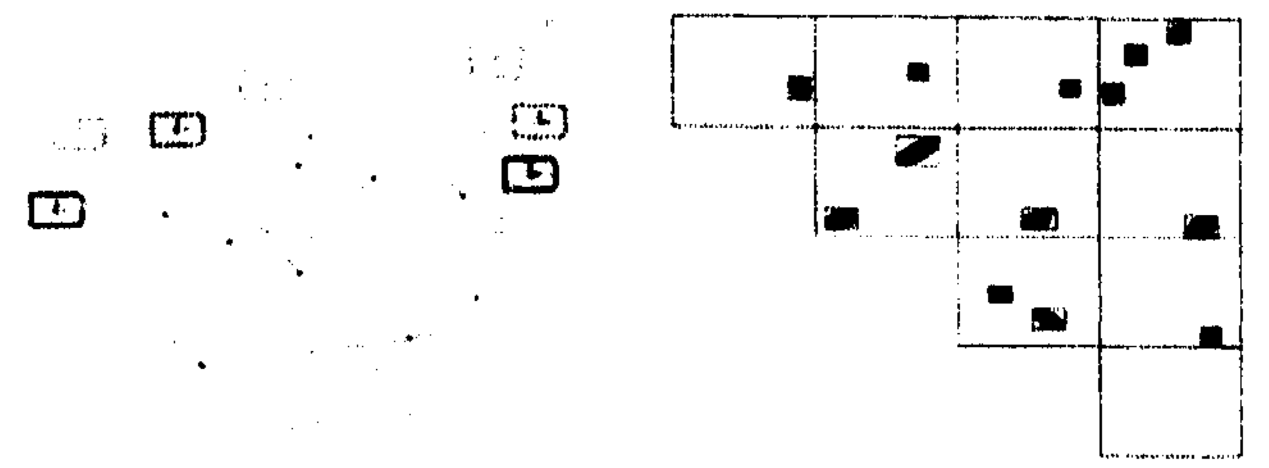
Completeness is not necessarily guaranteed in the third case **A||A**: we may find counterexamples where the bounding box approximation of the obstacles may split the free space into two connected components. Figure 4 shows an example where the bounding box transforms the full space into an obstacle. However such cases can be solved by the following resolution complete procedure: both arcs of a circle are recursively split into smaller arcs and each pair of the new elementary pieces is processed with the bounding box approach. Moreover such cases are easily identified in the path decomposition step above. This means that, according to the inputs, the algorithm may or not activate the recursive subdivision. The activation condition is a function dedicated to the case **A||A** and checking the existence of a collision-free vertical or horizontal line in the diagram. The activation cases are seldom seen. For instance they do not appear in the examples displayed in Figures 5, 7 and 8.

4 Coordination for *n* robots

Generalized coordination diagram Let us now consider n robot paths γ_i. The cartesian product of all the $\frac{n(n-1)}{2}$ elementary (γ_i, γ_j) coordination diagrams is a n-dimensional cube called *generalized coordination diagram.* A point in the n-cube belongs to an obstacle *iff* at least two robots collide. Therefore, the obstacles in the generalized coordination diagram have a cylindrical shape[4]. As a consequence the topology of the generalized coordination diagram is fully characterized by the topology of the elementary 2-dimensional diagrams. Figure 5(b) shows the 10 elementary diagrams for the path coordination problem of Figure 5(a).

A solution to the coordination problem is a collision-free path between $(0,\dots 0)$ to $(1,\dots 1)$.

[4]This property has been already noticed in [La Valle and Hutchinson, 1996]



(a) (b)

Figure 5: The 10 elementary diagrams (b) of the generalized coordination diagram of 5 paths (a).

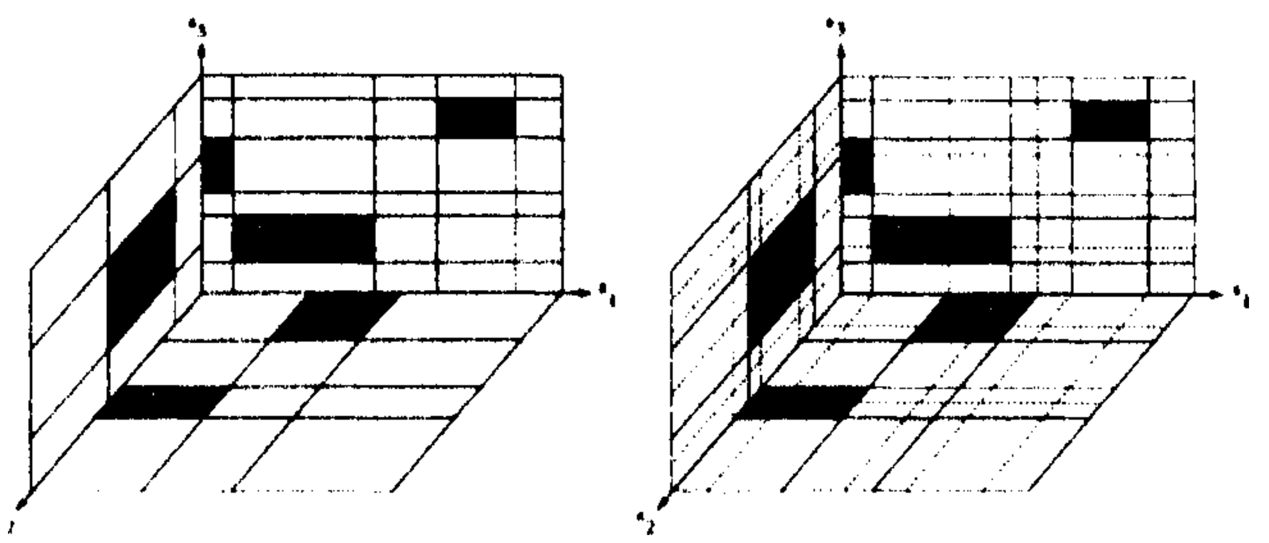
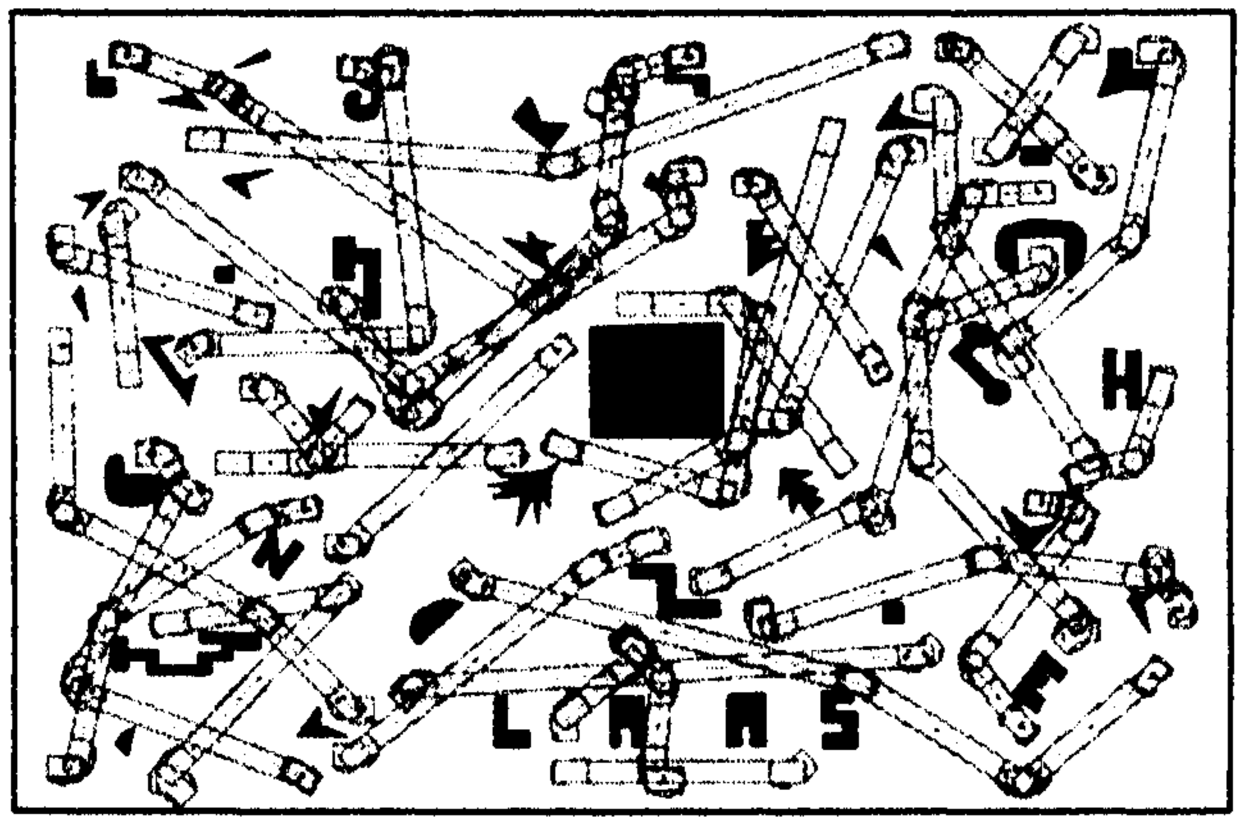


Figure 6: The cell decomposition of a diagram refines the cell decomposition of other diagrams.

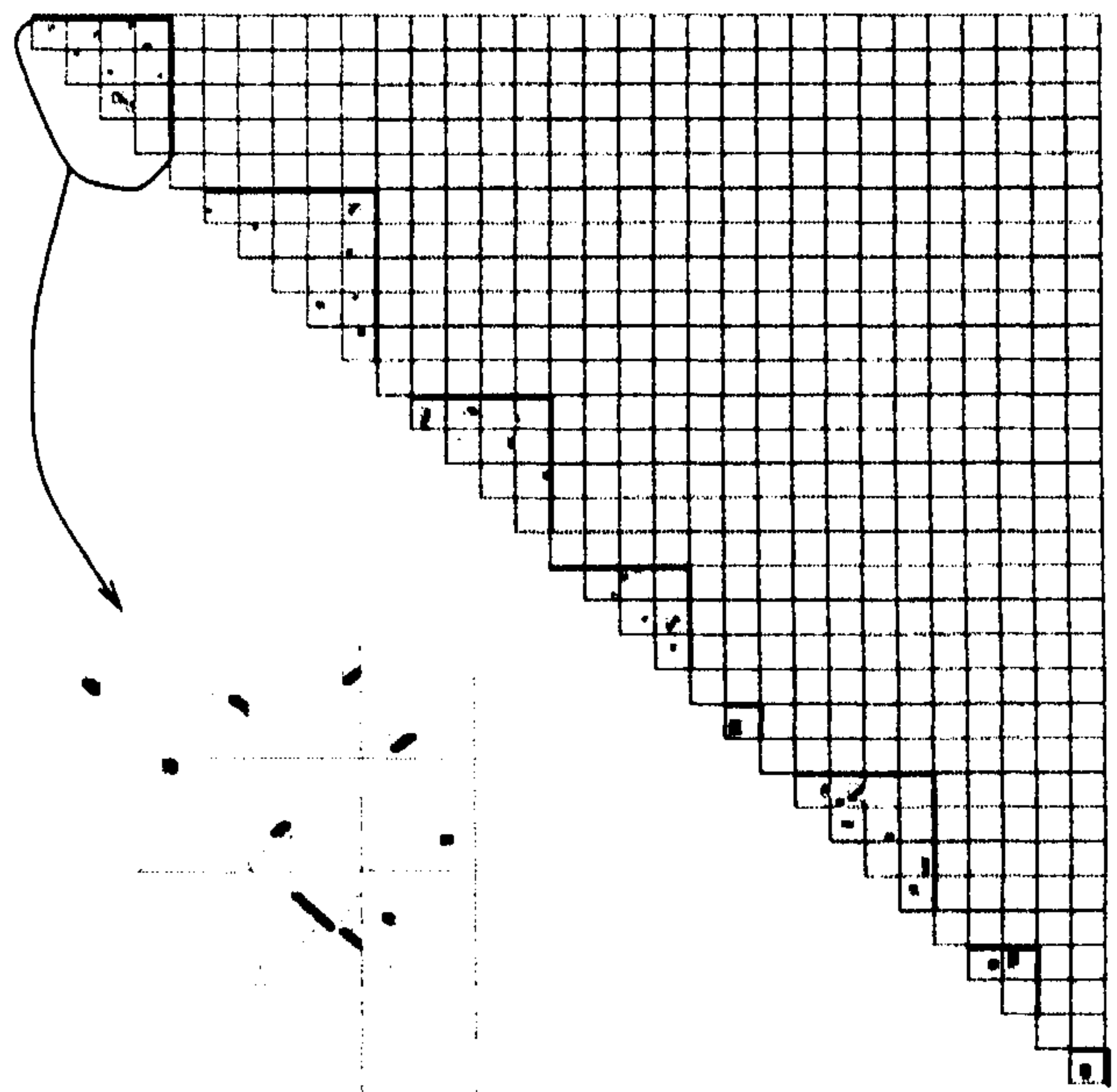
Generalized coordination diagram modeling and searching We have seen that the bounding box representation of the coordination diagram for two robots induces a decomposition of the diagram into rectangles. Let us consider three paths γ_1, γ_2 and' γ_3. The cell decomposition of (γ_1, γ_2) coordination diagram induces a partition of the axis s_2.. Then the cell decomposition of the (γ_2, γ_3) diagram is refined according to this partition. More generally, the cell decomposition of a (γ_i, γ_j) diagram induces a refinement of the cell decompositions of the 2(n - 1) diagrams (γ_i, γ_k) and (γ_k, γ_j) (see Figure 6). We denote by (i, j)-cell a cell of the (γ_i, γ_j) diagram after refinement. The 2-dimensional (i, j)-cells of all the (γ_i, γ_j) diagrams induce a cell decomposition of the n-cube. The cells of then-cube are denoted by n-cells. The main advantage of the following search is that it does not require an explicit representation of the n-cube.

Let us consider a (collision-free) n-cell reached at a current step of the search. The strategy consists in moving *only one robot* at once at each step. To do that the algorithm generates the 2n cells adjacent to the n-cell through a (n - l)-dimensional hyper-plane. Let us consider a n-cell cell, adjacent to the current collision-free n-cell and corresponding to an elementary motion of robot *i.* Due to the cylindrical shape of the obstacles, testing if cell is collision-free is easily performed: each of the (n-1) projections of cell onto the elementary $(\gamma_i, \gamma_.)$ diagrams should be a collision-free (i, .)-cell.

The search is performed by an A^* algorithm whose heuristic function is the shortest Euclidean path to the goal point (1,... 1) of the n-cube. Our algorithm computes coordination paths which are Manhattan paths: only one robot moves at once. If needed, we may overcome this fact by "smoothing" the computed path with the help of optimization techniques as in [Svestka and Overmars, 1995].



(a)



(b)

Figure 7: A case with 32 robots: the robots traces (a) and the 496 elementary diagrams. The partition into the 8 robot subgroups is illustrated by the 8 bold triangles.

Completeness Due to the cylindrical shape of the obstacles in the generalized coordination diagram, the algorithm above inherits from the completeness property of the coordination procedure for two robots presented in Section 3.

Interaction graph The final extension we propose is supported by a practical assumption. When a high number of robots plan their paths independently the path coordination problems are in general localized in different domains of the environment and only concern robot *subsets.* To reduce the combinatorial complexity of the global problem in practice we first identify which robot traces intersect another trace. We then build an *internetion graph* whose nodes are the robots; two robot-nodes are adjacent *iff* both corresponding traces intersect. A simple decomposition of the graph into connected components identifies automatically the various subgroups of robots requiring motion coordination. Then the algorithm above is applied to each subgroup.

Results Figure 7(a) shows an example of 32 mobile robots paths (including the traces). The 8 connected components of the interaction graph have been computed automatically. The global coordination diagram appears in Figure 7(b) showing clearly the structure induced by the 8 connected components. A detailed view of the coordination diagram involving a subgroup of 5 robots appears; it includes a display of the computed solution path for this group.

All the steps of the algorithm have been implemented in C++ and run on Sparc Ultra-1. The following table presents the computation times of each step of the algorithm for the examples in the figure 7 and the figure 8 that involves 150 robots[5]. A more complete analysis appears in [Leroy, 1998].

	32 rob.	150 rob.
Interaction graph computation and bounding box representation of the diagrams	$30s$	$240s$
Diagram refinement	$6s$	$13s$
Search	$3.7s$	$1.5s$

5 Conclusion

The proposed approach permits to solve problems for more than 100 robots in a reasonable time. The key points of the method are the efficiency of computation of the coordination configurations and the bounding box representation of the obstacles in the elementary coordination diagrams.

Nevertheless we should notice that the performance depends on the decomposition of the interaction graph into connected components. The worst case appears when the interaction graph has only one component (e.g., when the trace of some robot intersects *all* the other traces). In fact, the complexity of the approach is dominated by the highest dimension of the considered n-cubes. In practice the algorithm may explore efficiently n-cubes of dimension up to ten (i.e., involving 10 robots).

[5]The motion planner computing an admissible collision-free path for each robot is based on the algorithm presented in [Laumond *et a*l., 1994]. It is not possible to display the "effective" motions on pictures; animations related to this work may be seen at http://www.laas.fr/~sleroy.

We just argue that this limitation is not critical in practice. Moreover we do not know any alternative approach allowing to solve the case of Figure 8.

References

[Alami *et al,* 1995] R. Alami, F. Robert, F. F. Ingrand, and S. Suzuki. Multi-robot cooperation through incremental plan-merging. In *IEEE International Conference on Robotics and Automation, Nagoya, Aichi (Japan),* pages 2573-2578, 1995.

[Bien and Lee, 1992] Z. Bien and J. Lee. A minimum-time trajectory planning method for two robots. *IEEE Transactions on Robotics and Automation,* 8(3):414-418, June 1992.

[Buckley, 1989] S.J. Buckley. Fast motion planning for multiple moving robots. In *IEEE International Conference on Robotics and Automation, Scottsdale (USA),* pages 322-326, May 1989.

[Cao *et al,* 1997] Y. Cao, A.S. Fukunaga, and A.B. Kahng. Cooperative mobile robotics: Antecedents and directions. In *Autonomous robots,* volume 4, pages 7-27, 1997.

[Chang *et al,* 1994] C. Chang, M.J. Chung, and B.H. Lee. Collision avoidance of two robot manipulators by minimum delay time. *IEEE Transaction on Systems, Man and Cybernetics,* 24(3):517-522, 1994.

[Erdmann and Lozano-P6rez, 1986] M. Erdmann and T. Lozano-Perez. On multiple moving objects. In *IEEE International Conference on Robotics and Automation, San Francisco (USA),* pages 1419-1424, 1986.

[Hopcroft *et al.,* 1984] J. E. Hopcroft, J. T. Schwartz, and M. Sharir. On the complexity of motion planning for multiple independant objects: Pspace-hardness of the warehouseman's problem. *The International Journal of Robotics Research,* 3:76-88, 1984.

[Kant and Zucker, 1986] K. Kant and S. W. Zucker. Toward efficient trajectory planning: The path-velocity decomposition. *The International Journal of Robotics Research,* 5(3):72-89, 1986.

[La Valle and Hutchinson, 1996] S. La Valle and S. Hutchinson. Optimal motion planning for multiple robots having independants goals. In *IEEE International Conference on Robotics and Automation, Minneapolis (USA),* pages 1847-1852, May 1996.

[Latombe, 1991a] J.C. Latombe. A fast path planner for a car-like indoor mobile robot. In *9th Nat Conf on Artificial Intelligence, AAAI, Anaheim, CA,* pages 659-665, July 1991.

[Latombe, 1991b] J.C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, 1991.

[Laumond *et al,* 1994] J.P. Laumond, P. Jacobs, M. Taix, and R. Murray. A motion planner

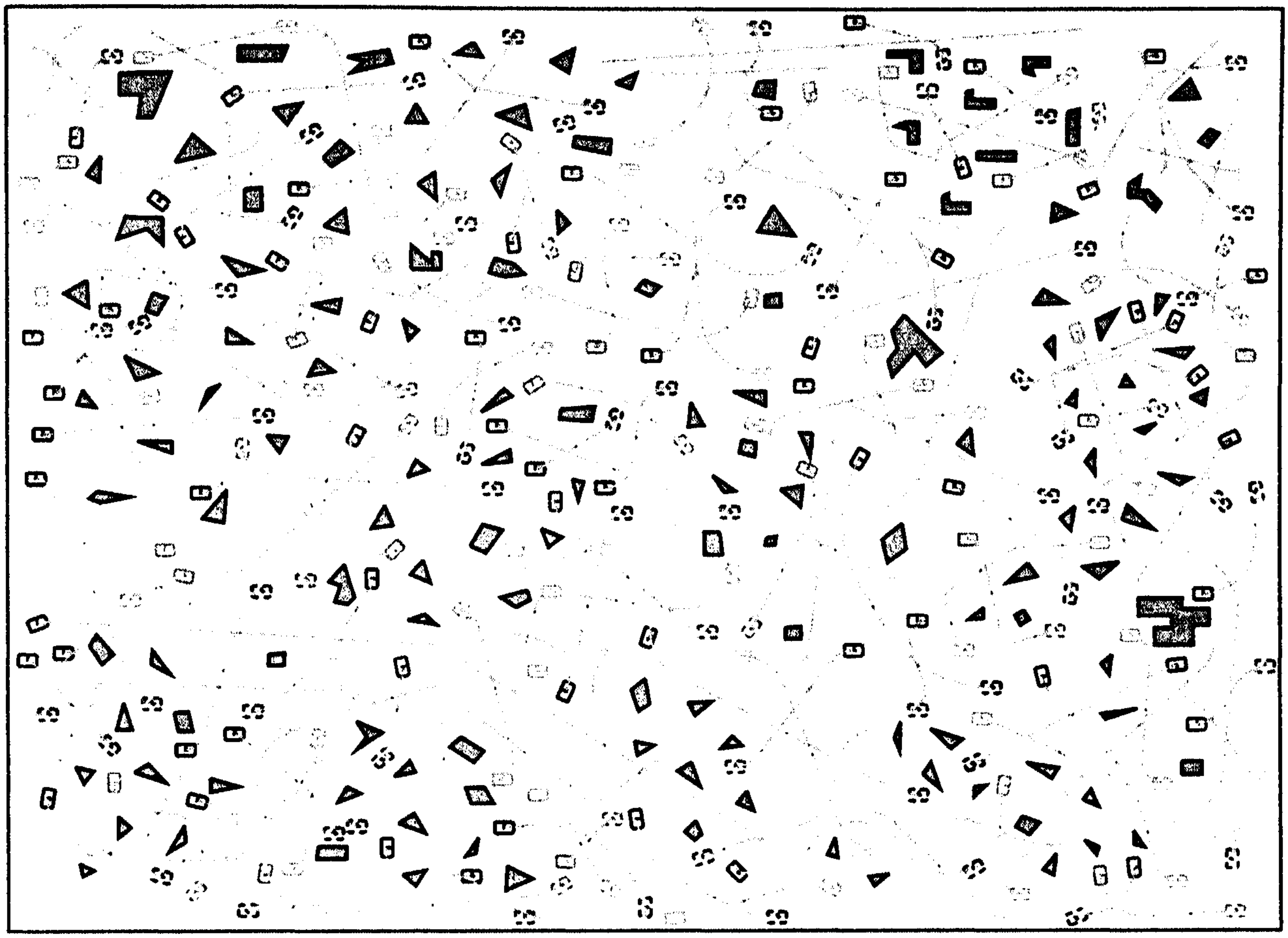


Figure 8: A case with 150 robots

for non holonomic mobile robots. *IEEE Transactions on Robotics and Automation,* 10(5):577-593, 1994.

[Laumond, 1986] J.P Laumond. Feasible trajectories for mobile robots with kinematic and environment constraints. In *International Conference on Intelligent Autonomous Systems, Amsterdam,* 1986.

[Leroy, 1998] S. Leroy. *Outils geometriques pour la planification de chemins de robots mobiles non holonomes.* PhD thesis, Universite Paul Sabatier, Toulouse (France), Decembre 1998.

[Mirtich and Canny, 1992] B. Mirtich and J. Canny. Using skeletons for nonholonomic motion planning among obstacles. In *IEEE International Conference on Robotics and Automation, Nice (France),* pages 2533-2540, 1992.

[O'Donnell and Lozano-Perez, 1989] P. O'Donnell and T. Lozano-Perez. Deadlock-free and collision-free coordination of two robot manipulators. In *IEEE International Conference on Robotics and Automation, Scottsdale (USA),* pages 484-489, 1989.

[Reif and Wang, 1995] J. Reif and H. Wang. Social potential fields : a distributed behavioral control for autonomous robots. In K. Goldberg *et al,* editor, *Algorithmic Foundations of Robotics,* pages 331-345. A. K. Peters, 1995.

[Simeon *et al,* 1998] T. Simeon, S. Leroy, and J.P. Laumond. A collision checker for car-like robots coordination. In *IEEE International Conference on Robotics and Automation, Leuven (Belgium),* volume 1, pages 46-51, 1998.

[Svestka and Overmars, 1995] P. Svestka and M. Overmars. Coordinated motion planning for multiple car-like robots using probabilistic roadmaps. In *IEEE International Conference on Robotics and Automation, Nagoya, Aichi (Japan),* pages 1631-1636, 1995.

[Warren, 1990] C.W. Warren. Multiple robot path coordination using artificial potential fields. In *IEEE International Conference on Robotics and Automation, Cincinnati (USA),* volume 1, pages 500-505, 1990.