Decentralized, Massively Parallel Path Planning and its Application to Process-Control and Multi Robot Systems*

Rumi M. Dubash
(student, rmdubash@cs.uh.edu)
Farokh B. Bastani
(coscfb@cs.uh.edu)

Department of Computer Science
University of Houston
Houston, TX 77204-3475

January 15, 1993

Extended Abstract

1 Earlier Work on Planning

Planning is one of the important cognitive tasks performed by human beings. It forms the basis of every non-trivial action performed by us. A lot of planning (at times subconscious) underlies even simple tasks such as walking through the hallway from one office to another. Thus, it is no surprise that in our effort to mimic/emulate human behavior, planning has received a lot of attention and is at the forefront of the research. A well planned ("thought out") strategy could be a potential winner, while the effects, usually not too pleasant, of a not well planned strategy are quite well known.

Planning as a discipline has been studied in various contexts. For example, planning used in developing task schedules, assembly line operations, etc. [33], planning used in performing simple tasks such as grasping a cup filled with liquid [36], planning used in production systems for firing of rules [6], planning of paths in robot systems [26, 32, 39]. Planning has not been studied as extensively with regard to process-control systems. Planning in this domain has always been couched in terms of optimization [25, 38, 41, 44, 46].

Path planning, one of the fundamental problems in robotics, has over the years received a lot of attention from researchers in various fields. It has been discussed extensively in the literature as *Piano-Movers'* [39] or *Find-Path* [7] problem. The work is based on a wide variety of approaches ranging from ones that make use of the underlying geometrical properties of the problem, to ones based on cell decomposition, from the wave-front approach which makes use of the problem's analogy to the physical world to the potential fields model. The problem has been studied within a restricted definition limiting the kind of obstacles, their shapes and the number of dimensions to more general definitions. [30, 39, 27, 26, 32, 34, 37, 3, 14, 42, 13, 29, 47] and the references contained therein provide a fairly comprehensive and wide ranging reading of the work pursued.

---

*Based on: (a) A Hybrid Architecture for Mobile Robots based on Decentralized, Parallel Path Planning, submitted to International Symposium on Autonomous Decentralized Systems, (b) A Massively Parallel AI-Based Approach to Process-Control, submitted to Conference on Applications of AI, and (c) Fault-Tolerant Process Planning and Control, Proceedings COMPSAC '92

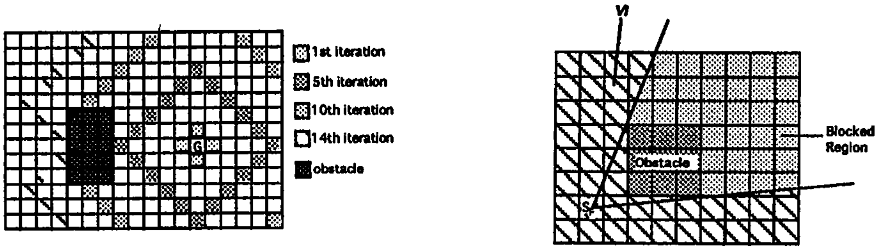

Figure 1: Wave propagation

Depending on the problem's definition, the complexity of the algorithm varies from polynomial to exponential [12, 30, 39]. In most cases, as the degrees of freedom, and thus the dimensionality, rise the computational complexity grows exponentially.

2 Decentralization and Massively Parallel Path Planning

In [19, 20, 21] we presented a massively parallel, decentralized approach to the path planning problem. It is based on the wave-front propagation [15], nearest-neighbor communication (cellular automaton [11, 43]) paradigm. The algorithm for each individual processor is fairly simple and all the processing is based on local information (principle of decentralization as defined in [17, 16]). The configuration-space [27] is mapped on to a multi-dimensional mesh. The wave is initiated by the destination processor in the form of messages to its neighbors. The neighbors pass on the message (propagate the wave) to their neighbors, and so on, Figure 1. To accommodate the necessity for Euclidean distance the messages also contain the distance traveled, the source of the message and the destination.

Besides being able to work with a larger number of degrees of freedom, this approach also possesses certain other desirable properties such as simplicity, reliability, efficiency and inherent fault-tolerance.

In [21] the case when the solution suffered from a digitizing bias was also considered. To arrive at an exact solution it is necessary to know the order in which the obstacles are encountered. This problem is intractable [3, 12]. A by-product of our planning algorithm provides us with this order (in fact more than that), thereby allowing the usage of techniques to derive exact solutions based on this assumption.

3 Why do we need a massively parallel path planning solution

In the present world of automation, process-control programs are usually put into place when the tasks are repetitive, laborious, hazardous, or need a quick response. A lot of strategic planning is involved in the successful manipulation of a control process. This gets carried over when this process is automated.

Planning based on optimization techniques works fine for usual process-control problems as long as the complexity is low enough to allow centralized control. The moment decentralized control entities for achieving the same goal are brought in, the optimization approach breaks down (or gets too complex for efficient, reliable implementation).

One of the approaches is to decompose the problem and solve the problem in a decentralized problem space. This reduces the overwhelming number of cases and avoids case-enumeration-analysis in software design and implementation.

For example, consider the robot-cart shown in Figure 2a [4]. This system is composed of three different controlling entities, each controlling an individual mechanism. There is one component controlling the translation motion and two distinct components controlling each section of the jointed arm on top of the cart. The goal of the system is to reach the ball B with the arm. This problem has been solved in its decomposed form in [4] using coarse-grained decomposition. The idea behind decomposing tasks is to simplify the individual control programs controlling each of the mechanisms. Though the approach is very promising, it has a very critical limitation in the form of the requirement that the *free space* be *convex*. The same system shown in Figure 2b, with the addition of just one obstacle induces a major complexity violating the convexity condition. Using the same approach would require an inordinate amount of coordination and synchronization among the controlling entities, thereby defeating the purpose and benefits that accrue from decomposition.

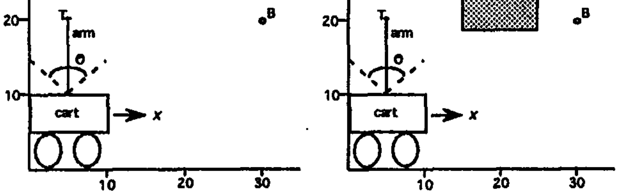

Figure 2: Robot-cart with and without an obstacle in the environment

What is needed is a different approach to tackle this problem. Our conviction is that a non-traditional-AI based approach results in a more comprehensive solution. We have presented a model to reduce process-control problems from the real world to configuration-space [27] to a state space [20]. The process-control problem is then solved as a path planning problem in the state space and the results (the trajectory) translated back as instructions for the controlling mechanisms. The idea is quite similar to the application of fuzzy logic to controllers [40]. Instead of the system being modeled analytically by a set of differential equations, their solutions telling the system what action to take, the fuzzy controller handles these adjustments by a fuzzy rule-based expert system — a model of human thinking process. Figure 3 portrays a solution to the robot-cart problem posed earlier.

Some additional features of this approach include circumventing the issues of reasoning, such as reasoning about the arm position while the cart is in motion, the presence of obstacles, etc. Thus one of the shortcomings in trying to reason within *non-embodied* and *non-situated* systems [9] is eliminated. Diverse process-control problems are reduced to a single unified problem of finding a trajectory for the motion of a point-object avoiding collision with the obstacles that clutter the environment.

But, as pointed out in [8, 9] simply reducing the problem to a regular search problem may not be the most effective form of AI. Also, considering the computability results in path planning and the potential for problems arising with a large number of degrees of freedom, a different, perhaps more non-conventional approach, borrowing from principles of decentralization and using the power of massive parallelism, to path planning is warranted.

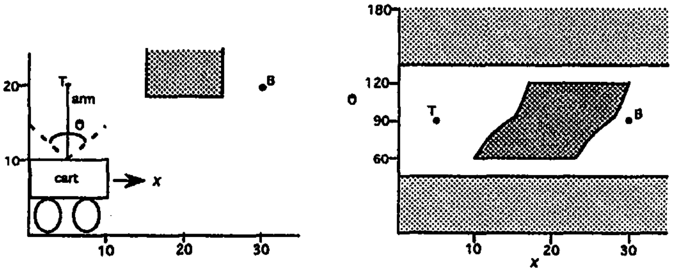

Figure 3: Mapping of the robot-cart

4 Multi Robot Systems — Does massive parallelism provide hope for Centralized/Hybrid models?

Robot systems can be considered to be a special and very important case of control systems. The use of autonomous robots continues to grow with new applications being found with every passing year. In most of the applications, it is not unusual to find more than one robot involved in achieving the goal. [1, 2, 28, 31, 35, 45] is a small selection of the work presented in the current literature. When more than one robot is involved or when the environment in which they are operating is dynamically developing, it is very important that they do not collide with other objects or robots. This requires a much higher level of intelligence than usual. In such cases it is almost impossible to arrive at a global plan/strategy for a robot to achieve its goal. An attempt to capture all the reasoning and intelligence in a central unit is a formidable task. Although this problem has been solved by adopting some simplifying assumptions, for example, prioritization of units [22], the problem still remains open.

One of the intersting approaches is the idea of Distributed or Decentralized Robotic systems. There has been a lot of research lately on this approach [10, 23, 24, 48]. Distributed Robotic Systems have, by their definition, properties, including intelligence that are distributed. Centralized intelligence is considered feasible for systems with approximately 10 units (moving robots). Distributed intelligence is preferred when the number of units is between 10s and low 100s [5].

There are pros and cons for both approaches. In the pros column, the centralized approach is capable of global, long term, (close to) optimal planning. The cons for this approach include concerns of efficiency in the case of a large number of robots, the requirement of homogeneity of robots, assignment of priorities in an effort to reduce the complexity, and the possible repercussions on reliability.

The advantage of the decentralized approach is the fact that a large number of robots can be accommodated. Robots of varying capabilities can coexist. And, an important one is the emergence of behavior. The reasons against this approach include non-optimal paths, communication between robots, cost (considering that each robot is essentially more intelligent thus requiring more hardware and software), and the problem of deadlocks and lockouts.

Considering all the pros and cons, we feel that a hybrid approach presents a better rationale. A hybrid model essentially inherits all the advantages of both the other approaches. Of course the disadvantages step in too. These disadvantages can be offset by lowering the sophistication of the individual units and making the planning more competent by introducing decentralization and massive parallelism. The lowering of sophistication eliminates problems associated with inter-unit communication and planning within the units. Parallelism also contributes positively to the speed and reliability. Thus we still have the advantage of a centralized planner's long range planning capability and at the same time, use distributed intelligence to deal with unpredicted, unforeseen, immediate perturbances in the system's environment. Another major advantage is the possible decrease in cost. Since it is no longer required that each of the agents be highly intelligent, the final cost associated with each is decreased. Applications that involve potential hazards where the agents may not be protected (or their loss is highly probable) can take advantage of a system based on this model. Potential military operations that require the central command to be well protected can also take advantage of this model (re-enforcing the security of the central command). With not so smart agents, the loss incurred when one falls into enemy hands may not be as heavy as in the case of truly distributed, highly intelligent robots.

Indirect applications of such systems in air-traffic control, intelligent highway systems, traffic control and vehicle guidance systems exist. In such systems the agents cannot be granted full autonomity; yet, at the same time a completely centralized approach would also fail. For example, it is quite impossible to expect all pilots in the air and ground to come to an agreement and a plan all by themselves (completely decentralized, negotiating entities). At the same time a completely centralized scheme would fail to take advantage of the vantage point, experience of the human at the controls, and all the unanticipated events.

Also, the desirable feature, stochastic self-organization may not be as simple to achieve with completely decentralized entities. Autonomous entities, though well versed in their tasks may not be able to achieve a goal as easily as when they had a central plan. For example, in a basketball game, though each player is aware of the ultimate goal of shooting a basket, the team may not fair well if there is a lack of a central planning agency, usually the coach or the point-guard.

In [21] we presented a hybrid architecture of robot systems that is reliable and efficient at various levels of execution using the decentralization and parallel computing methods. It is a hybrid model since we continue to adopt the concept of a centralized planning agency, and at the same time adopt a decentralized approach in dealing with unanticipated events. The centralized planning agency is itself based on the a massively parallel cellular automaton model contributing in speed and the possibility of handling several agents at the same time. Figure 4 shows the multiple layers in the hybrid architecture.

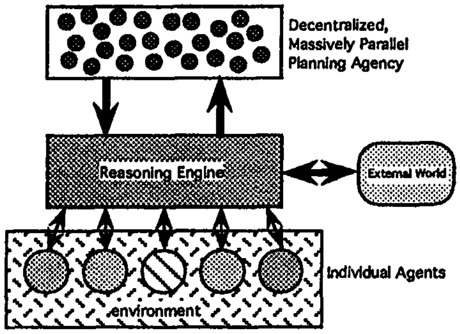

Figure 4: A hybrid architecture

With the massively parallel decentralized path planning system as the core, we outlined and discussed an implementation that is capable of controlling multiple robots, Figure 5 is an example of three interacting robots. The robots themselves could possess varying degrees of autonomity, with the model and solution being flexible enough to accommodate such variety. The implementation is easily extendible and does not suffer the limitations of impracticality of cellular robotic systems.

5 Conclusions

It is our conviction that conventional as well as non-conventional AI can derive a lot from the power of massive parallelism. We have outlined an application researched in contemporary AI, multi-robot

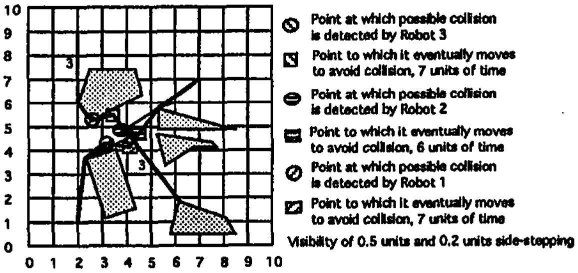

Figure 5: Multiple interacting robots

systems, where this power can be brought to bear to allow tackling of problems that would not have been possible with regular uniprocessing or coarse grained systems. The use of massive parallelism provides renewed hope for centralized and hybrid models.

The application of massive parallelism also opens doors to innovative application of AI techniques to problems that have been not in the realm of traditional AI, the process-control systems. Towards furthering and achieving this end and to support the two applications mentioned, we have developed a massively parallel path finding solution that allows handling of a larger number of dimensions. The path planning problem though traditionally from the field of AI, our solution borrows and builds on the concept of decentralization to simplify the solution.

References

[1] A.R. Arkin, R.R. Murphy, "Autonomous navigation in a manufacturing environment" *Proc. 1990 IEEE Intnl. Conf. on Robotics and Automation*, 1990.

[2] A.R. Arkin "Cooperation without communication: multiagent schema-based robot navigation", *Journal of Robotic Systems*, 1992.

[3] C. Bajaj, "An efficient parallel solution for Euclidean shortest path in three dimensions," *Proc. 1986 IEEE Intnl. Conf. on Robotics and Automation*, 1986.

[4] F.B. Bastani, I.L. Yen, I.R. Chen, "A class of inherently fault tolerant distributed programs," *IEEE TSE*, October 1988.

[5] G. Beni, J. Wang, "Theoretical problems for the realization of distributed robotic systems," *Proc. 1991 IEEE Intnl. Conf. on Robotics and Automation*, 1991.

[6] P. Bonissone, S. Dutta, "Merging Strategic and Tactical Planning in Dynamic, Uncertain Environments," *CAIA*, 1992.

[7] R.A. Brooks, "Solving the find-path problem by good representation of free space," *IEEE TSMC*, 1983.

[8] R.A. Brooks, "Intelligence without representation," *Artificial Intelligence*, 1991.

[9] R.A. Brooks, "Intelligence without reason," *IJCAI*, 1991.

[10] S.J. Buckley, "Fast motion planning for multiple moving robots," *Proc. 1989 IEEE Intnl. Conf. on Robotics and Automation*, 1989.

[11] A.W. Burks (Ed.), *Essays on cellular automaton*, University of Illinois Press, 1970.

[12] J. Canny, J. Reif, "New lower bound techniques for robot motion planning problems," *FOCS*, 1987.

[13] S.C. Chang, J. Ja'Ja', "Parallel algorithms for wiring module pins to frame pads," *Proc. ICPP*, 1989.

[14] F. Dehne, A.L. Hassenklover, J.R. Sack, "Computing the configuration space for a robot on a mesh-of-processors," *Parallel Computing*, Vol. 12, 1989.

[15] E.W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, pp. 269-271, Oct. 1959.

[16] E.W. Dijkstra, "Self-stabilizing systems in spite of distributed control," *CACM*, vol. 17, no. 11, pp. 643-644, 1974.

[17] E.W. Dijkstra, "The solution to a cyclic relaxation problem," in *Selected Writings on Computing: A personal Perspective*, E.W. Dijkstra (ed.), Springer-Verlag, New York, 1982.

[18] R.M. Dubash, I.L. Yen, F.B. Bastani, "Fault tolerant process planning and control," *IEEE COMPSAC Conference*, Chicago, Sept. 1992.

[19] R.M. Dubash, I.L. Yen, F.B. Bastani, "Fault tolerant process planning and control," *Technical Report, UH-CS*, University of Houston, 1992.

[20] R.M. Dubash, F.B. Bastani, "A massively parallel AI-based approach to process-control," *Conference on Applications of Artificial Intelligence*, 1993.

[21] R.M. Dubash, F.B. Bastani, "A Hybrid Architecture for Mobile Robots based on Decentralized, Parallel Path Planning," *International Symposium on Autonomous Decentralized Systems*, 1993.

[22] M. Erdman, T. Lozano-Perez, "On multiple moving objects," *Proc. 1986 IEEE Intnl. Conf. on Robotics and Automation*, 1986.

[23] T. Fukuda, S. Nakagawa, Y. Kawauchi, M. Buss, "Structure decision method for self organising robots based on cell structures-CEBOT," *Proc. 1989 IEEE Intnl. Conf. on Robotics and Automation*, 1989.

[24] S.Y. Harmon, W.A. Aviles, D.W. Gage, "A technique for coordinating autonomous robots," *Proc. 1986 IEEE Intnl. Conf. on Robotics and Automation*, 1986.

[25] R.E. Larson, P.L. McEntire, J.G. O'Reilly, "The theory of real-time control," in *Handbook of Software Engineering, C.R. Vick, C.V. Ramamoorthy (eds.)*, Van Nostrand Reinhold, 1984.

[26] J.C. Latombe, *Robot motion planning*, Kluwer Academic Publishers, Boston, 1991.

[27] T. Lozano-Perez, "Spatial planning: A configuration space approach," *IEEE Trans. Computers*, Feb. 1983.

[28] T. Lozano-Perez, J.L. Jones, E. Mazer, P.A. O'Donnell, "Task-level planning of pick-and-place robot motions", *IEEE Computer*, 1989.

[29] C.Y. Lee, "An algorithm for path connections and its applications," *IRE Trans. Electronic Computers*, Sep. 1961.

[30] D.T. Lee, F.P. Preparata, "Computational geometry - A survey" *IEEE TOC*, Dec. 1984.

[30] D.T. Lee, F.P. Preparata, "Computational geometry - A survey" *IEEE TOC*, Dec. 1984.

[31] L. Missiaen, H. Lecluyse, J.P. Massart, M. Bruynooghe, "Navigation algorithms for a mobile robot using ultrasonic sensors", *Artificial Intelligence and Information Control Systems for Robots*, 1987.

[32] J.S.B. Mitchell, *Planning shortest paths*, Ph.D. Dissertation, Dept. of Operations Research, Stanford University, 1986.

[33] C.L. Pape "A combination of centralized and distributed methods for multi-agent planning and scheduling," *Proc. 1990 IEEE Intnl. Conf. on Robotics and Automation*, 1990.

[34] M.J. Quinn, N. Deo, "Parallel graph algorithms," *Computing Surveys*, Sept. 1984.

[35] N.S.V. Rao, "Algorithmic framework for learned robot navigation in unknown terrains", *IEEE Computer*, 1989.

[36] D.E. Rumelhart, J.L. McClelland, *Parallel Distributed Processing*, MIT Press, 1986.

[37] C. Savage, J. Ja'Ja', "Fast, efficient parallel algorithms for some graph problems," *SIAM J. Comp*, Nov. 1981.

[38] M.G. Singh, *Decentralised control*, North Holland, Amsterdam, 1981.

[39] J.T. Schwartz, M. Sharir, J. Hopcroft, (eds.) *Geometry and complexity of robot motion*, Ablex Publishing Corp. Norwood, NJ, 1987

[40] D.J. Schwartz, G.J. Klir, "Fuzzy logic flowers in Japan," *IEEE Spectrum*, 1992.

[41] J.A. Stankovic, N. Chowdhury, R. Mirchandaney, I. Sidhu, "An evaluation of the applicability of different mathematical approaches to the analysis of decentralized control algorithms," *COMPSAC* 1982.

[42] W.K.Tsai, G.M. Harung, J.K. Antonio, "Fast parallel hierarchical aggregation/disaggregation algorithms for multistage optimization problems and shortest path problems", *Proc. American Control Conference*, 1989.

[43] S. Ulam, "On some mathematical problems connected with patterns of growth of figures", *Proc. Symposia in Applied Mathematics*, Vol 14, 1962.

[44] N.A. Vosbury, "Process Design", in *Handbook of Software Engineering, C.R. Vick, C.V. Ramamoorthy (eds.)*, Van Nostrand Reinhold, 1984.

[45] C.R. Weisbin, G. de Saussure, J.R. Einstein, F.G. Pin, E. Heer, "Autonomous mobile robot navigation and learning", *IEEE Computer*, 1989.

[46] H.S. Witsenhausen, "A counterexample in stochastic optimum control," *SIAM J. of Control*, 1968.

[47] Y. Won, S. Sahni, "Maze routing on a hypercube multiprocessor computer," *Proc. ICPP*, 1987.

[48] D.Y. Yeung, G.A. Bekey, "A decentralized approach to the motion planning problem for multiple mobile robots," *Proc. 1987 IEEE Intnl. Conf. on Robotics and Automation*, 1987.