

Adding Memory to the Evolutionary Planner/Navigator

Krzysztof Trojanowski*, Zbigniew Michalewicz^{o*}, Jing Xiao^o

Abstract—The integration of evolutionary approaches with adaptive memory processes is emerging as a promising new area for research and practical applications. In this paper, we report our study on adding memory to the Evolutionary Planner/Navigator (EP/N), which is an adaptive planning/navigation system for mobile robots based on evolutionary computation. Preliminary results from our experiments demonstrate the potential of such extension to the EP/N in improving planning effectiveness in partially-known environments.

Keywords—path planning and navigation, evolutionary algorithm, memory chromosomes, memory strategies, adaptation to changes in an environment.

I. INTRODUCTION

MEMORY serves as a mechanism for storing previous experience and knowledge to aid problem solving strategies. In the problem of robot navigation, the knowledge from the robot's past exploration of the environment together with the knowledge from previous navigation tasks, can be used to facilitate more efficient and effective planning. Thus, we are motivated to study the issue of adding memory to the memoryless Evolutionary Planner/Navigator (EP/N) [9], [13], [14], [15], [16] for mobile robots.

The approaches to memory extension in evolutionary systems could be classified in different categories. One possibility is to distinguish between

- local memory approaches, where single individuals are extended by their own memory structures, and
- global memory approaches, where a memory is employed for the whole population, as in tabu search [6], [7].

The above categories can be further divided on the basis of additional attributes, such as whether memory size is static or changeable over time, whether 'genetic' operators operate on memory chromosomes, and whether, in the case of local memory, individuals exchange information stored in their memories. General issues involved in using memory include (1) the memory structure, (2) the rules for remembering (i.e., storing information in the memory), and (3) the rules for recalling (i.e., accessing the memory information), etc.

In this paper we examine one particular local memory strategy, which is to equip each individual in a population with a memory buffer in addition to its active chromosome. We apply this strategy to the EP/N to enhance its capa-

bility of finding suitable paths in partially-known environments. The next section reviews the EP/N briefly. Section 3 describes in detail the memory strategy we use, whereas Section 4 presents preliminary results of various experiments and compares the results of the extended EP/N to those obtained from the original, memoryless EP/N. Section 5 concludes the paper.

II. DESCRIPTION OF THE EP/N

The EP/N is an adaptive path planner/navigator for guiding a mobile robot from some initial location to some goal location following a collision-free and near-optimal path. It represents paths as chromosomes in uniquely designed data structures, measures path feasibility (i.e., whether a path is collision-free) and quality (i.e., near-optimality) by an evaluation function, and evolve paths to better ones via special genetic operators incorporating problem-specific knowledge. The EP/N is particularly characterized by unifying off-line planning and on-line planning/navigation with the same evolutionary algorithm. For on-line operation, the evolutionary algorithm enables simultaneous planning and robot movement to achieve high efficiency and effective adaptation of the robot to changes or uncertainties in an environment.

Specifically, the on-line EP/N runs two processes in parallel:

1. navigation of the robot along the current best path while sensing the environment to detect unknown objects, and
2. continuation of the evolution process in search for further path improvements, taking into account new location of the robot and newly sensed objects (if any).

The two processes are related in the following way: while the robot moves along the current best path p_c , the best new path p emerged from the evolution process is checked every n generations for feasibility: if p is feasible, the robot starts moving along p ; otherwise the robot continues to move along p_c while the evolution process also continues. Note that during such on-line navigation, the starting location of each path (chromosome) in a population is constantly updated to reflect the current location of the robot as it moves. By letting the robot to follow the *current* best path from the continuing evolution, the EP/N is able to constantly improve the robot motion between the current location of the robot and the goal, even if the robot is not approaching any obstacles. A discovery of a new obstacle by sensors during the navigation process results in changes in fitness values for all paths in the current population, but evolution continues with the current population. In

*Institute of Computer Science, Polish Academy of Sciences, ul. Ordona 21, 01-237 Warsaw, Poland

^oDepartment of Computer Science, University of North Carolina, Charlotte, NC 28223, USA

other words, unlike some traditional planners, the EP/N does not discard the planning process and re-start it *from scratch* (or with an equal effort) whenever there are new objects which may block the best path. Rather, it takes advantage of the information accumulated so far by continuously keeping the evolution process of paths. The process *self adapts* to changes in real-time.

Note that this characteristic of the EP/N also distinguishes the system most significantly from many on-line reactive planners, which do not try to optimize (i.e., globally plan) paths. Unlike the existing reactive planners, the EP/N does not just react to newly sensed information but acts on the basis of the known information, as well as the knowledge accumulated so far about the environment. When we compare this approach to traditional methods we cannot classify the evolutionary method neither as a global planning approach like the roadmap or the cell decomposition [8] nor as a local one. It is a general strategy where the planner makes use of both accumulated and newly sensed information and is flexible to such aspects of planning as changes in the environment or multiple optimisation goals. And that is why this method seems to be so competitive.

It is for the sake of better utilization of the known information or past experience that we study the addition of local memories to the on-line EP/N. In this study, we build memory structures upon the latest version of the EP/N (as described in [16]), except that we drop the consideration on path clearance in the path evaluation function to make it easier for us to compare results from different experiments.

III. USE OF MEMORY

As mentioned in Section 1, there are three major issues related to the use of memory: structure and content of memory, how to remember information, and how to recall information. We now discuss them in turn.

Memory structure and content. We have implemented the following local memory strategy for the EP/N: each individual in a population has its own memory structure, and there is no exchange of information among individuals in the population. An individual consists of an *active chromosome*, which represents a path, and a *memory buffer*, which may contain several chromosomes (i.e., paths) inherited from the individual's ancestors. The size of the memory buffer is constant during the time of an evolutionary process.

Process of remembering. The first generation of populations have empty memory buffers. Then, each time after a new individual is generated, if it is good enough to be included in the next generation of population, the active chromosome of its parent (or better parent — in case there are two parents) is added to its memory buffer. In addition, it will inherit the chromosomes in the memory buffer of its parent or better parent. Thus, what is remembered (i.e. the content of memory buffers) increases as the generation number increases. Each memory buffer is a FIFO queue such that when it is full, the oldest path (chromosome) is deleted to make room for a new one.

Process of recalling. Recalling memory can occur at different times, such as

- at the beginning of every generation,
- after every modification of a chromosome,
- before every step of the robot motion, or
- after the discovery of every previously unknown obstacle.

Also, there can be a stochastic variable to make the recalling of memory based on some probability at those times.

In our current implementation with the EP/N, memory is recalled every time the robot encounters a previously unknown obstacle. Note that this is the time when all paths in the current population are re-evaluated (to take into account the effect of the newly discovered obstacle). During this re-evaluation process, the chromosomes in the memory buffer of an individual become active and are also re-evaluated together with the active chromosome of the individual. Since active chromosomes represent paths starting from the *current* location of the robot, whereas chromosomes in a memory buffer represent paths from some (likely different) *previous* locations of the robot, it is necessary to adjust the starting points of the remembered paths to the current location before re-evaluation. After the re-evaluation, if any of the remembered paths is better than the currently active chromosome of the individual, then it is swapped with the current one to become active, while the latter becomes inactive and is remembered in the memory.

Figure 1 illustrates the potential usefulness of the described idea.

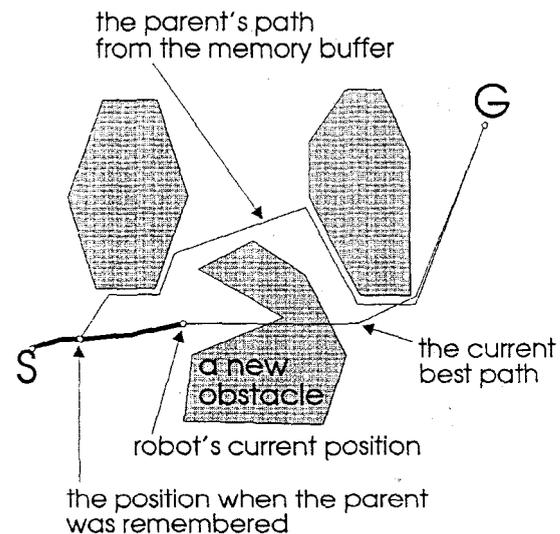


Fig. 1. Usefulness of memory

IV. EXPERIMENTAL STUDY

We have performed a few experiments by running the extended EP/N with memory and compare the results with the results obtained under the memoryless EP/N.

A. Comparison on path quality

As explained in [16], in a partially-known environment where a robot encounters unknown obstacles during a navigation task, a reasonable measure of the quality of the actual path that the robot takes is to divide the path into so-called fragments: the cut point between fragments is the location where the robot senses a new obstacle. There are as many cut points as the number of new obstacles sensed during the robot's movement (therefore the number of fragments, f , is by one greater than the number of cut points). Then each fragment is compared to an ideal path generated to connect the fragment's start and goal locations¹, which results in a relative error e_i in path cost for the segment.

We use e_i s to compare the qualities of two alternative real paths.

B. Experiments and results

We have done experiments in the environments shown in Figures 2, 3, and 4, where the unknown obstacles are indicated by the boundaries only and the rest are known obstacles. The sizes of those environments are 600×400 , 540×450 and 630×450 , respectively. The navigation task of a robot in each environment is indicated by a dot at the lower-left corner as the starting location and a dot at the upper-right corner as the goal location of the robot.

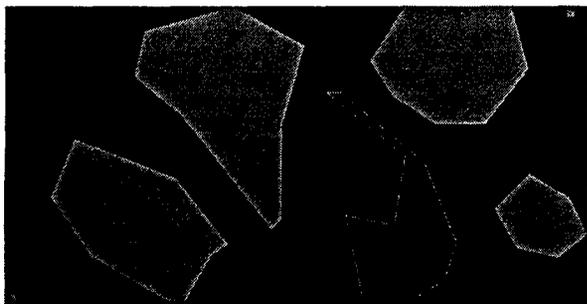


Fig. 2. Environment 1 with one unknown obstacle



Fig. 3. Environment 2 with one unknown obstacle

For the navigation task in each environment, we have run four sets of simulations to test the behavior of the

¹Such an ideal path can be generated by running the EP/N off-line in the same environment but with all obstacles known for a sufficient number of generations.

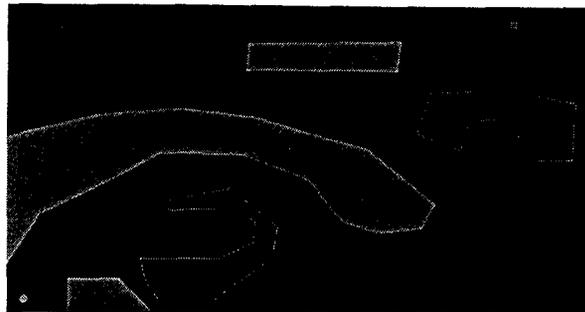


Fig. 4. Environment 3 with two unknown obstacles

EP/N without or with memory. We first ran the memory-less EP/N, i.e., the memory buffer size $m = 0$, and next — the extended EP/N with different memory buffer sizes: $m = 1, 2$, and 5 , respectively. In each run, the population size was set to 20, and the number of generations n between the robot's (two adjacent) steps was set to 10.

Figures 5, 6, and 7 display sample paths traveled by the robot under the guidance of EP/N in these environments, where the end points of fragments are clearly marked.

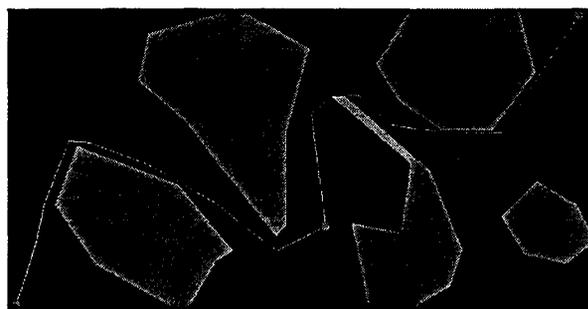


Fig. 5. A sample path traveled (in environment 1)

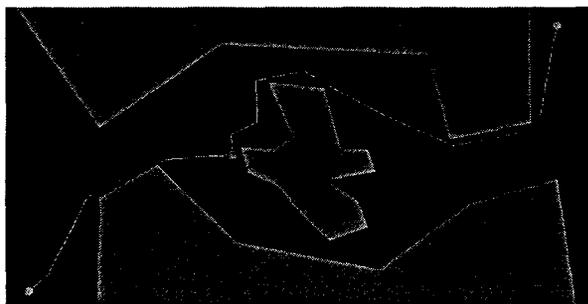


Fig. 6. A sample path traveled (in environment 2)

Tables 1–4 show the results obtained from the four sets of experiments in each of the three environments, where the symbols are:

- m — the size of a memory buffer;
- g — the average number of generations elapsed during the traversal of an entire path;
- c_i — the average cost of the i -th fragment of the path;
- \bar{e}_i — the average error of the i -th fragment of the path;

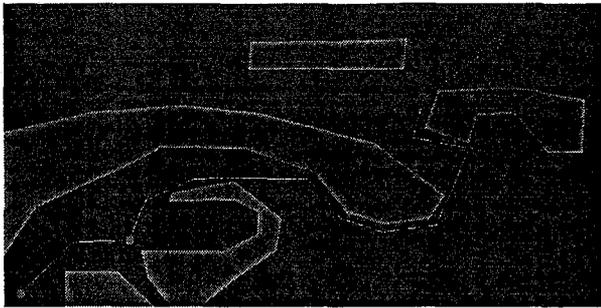


Fig. 7. A sample path traveled (in environment 3)

σ_i — the standard deviation of e_i .

The statistics were obtained by repeating each experiment 100 times. Note that only those runs which resulted in end points of a corresponding fragment close together (within a circle of the radius 3) were used in the statistics to make the results fair. Moreover, the first fragment in any situation was not considered since its creation was not affected by whether there was a memory (due to that the first recalling of a memory occurs when an obstacle is sensed, which is when the first fragment ends).

TABLE I
EXPERIMENTAL RESULTS FOR ENVIRONMENT 1

m	c_2	\bar{e}_2	σ_2	g
0	628.43	0.1269	0.0953	619.56
1	580.67	0.0634	0.0383	600.67
2	575.10	0.0564	0.0449	607.22
5	576.46	0.0580	0.0299	598.21

TABLE II
EXPERIMENTAL RESULTS FOR ENVIRONMENT 2

m	c_2	\bar{e}_2	σ_2	g
0	620.86	0.0973	0.0518	450.00
1	611.94	0.0851	0.0387	447.45
2	612.60	0.0859	0.0384	442.73
5	607.83	0.0787	0.0438	445.71

TABLE III
EXPERIMENTAL RESULTS FOR ENVIRONMENT 3

m	c_2	c_3	\bar{e}_2	\bar{e}_3
0	602.37	382.22	0.1376	0.6705
1	582.09	287.37	0.1031	0.1870
2	576.32	308.22	0.0935	0.2992
5	563.83	289.82	0.0724	0.1976

The results confirmed our intuitions. Individuals with memory structures have potentially more chances to generate better paths to deal with changing environmental

TABLE IV
EXPERIMENTAL RESULTS FOR ENVIRONMENT 3 (CONT'D)

m	σ_2	σ_3	g
0	0.0999	1.0766	294.80
1	0.0581	0.1600	285.65
2	0.0672	0.2562	283.48
5	0.0297	0.1107	279.23

conditions. Generally, they seem to generate better paths (with smaller \bar{e}_i) consistently (with small σ) and faster (with smaller number of generations g). With added memory, the EP/N increased its efficiency for those cases where the memory was really helpful. However, Table 2 shows that memory did not improve the planner a lot, suggesting that memory is not always helpful (as it is the case with environment 2). This is also easy to understand.

What is interesting, but not apparent from these preliminary results, is the effect of the size of a memory buffer. Although it appears that the larger m is, the better the results are, the improvements on results are too small to make such conclusion definite. Hence, we have run a few additional experiments, which indicated that for the larger number of generations before the memory was recalled, i.e., before the robot encountered an unknown obstacle, the advantage of a larger memory became more significant.

V. CONCLUSIONS

This work aims to explore the role of memory in a system's adaptation to changes in an environment. Our experiments show that even a simple memory structure can have complex influence on the system performance. The role of a memory also depends on specific environments; for some of them it is much more significant than for others.

For future work, we would like to focus on gaining more insights about the role of memory structures in evolutionary systems. What kind of memory strategies are better? How can remembered information be recalled and used more efficiently? Which approach could decrease computational cost for memory operations? Should the size of memory vary during one run of the system? How good are multi-chromosome structures with a dominance function? We hope that further experiments with the EP/N would provide partial answers for the above questions.

ACKNOWLEDGMENTS

The research reported in this paper was partially supported by the grant 8T11C 010 10 from the Polish State Committee for Scientific Research.

REFERENCES

- [1] Arkin, R.C., "Motor Schema-based Mobile Robot Navigation", *Int. J. Robotics Research*, pp.92-112, Aug. 1989.
- [2] Bessiere, P., Ahuactzin, J.-M., Talbi, A.-G., and Mazer E., "The "Ariadne's Clew" Algorithm: Global Planning with Local Methods", *Proceedings of 1993 IEEE-IROS International Conference on Intelligent Robots and Systems, Yokohama, Japan, Sept. 1993.*

- [3] Borenstein, J., and Koren, Y., "The Vector Field Histogram — Fast Obstacle Avoidance for Mobile Robots", *IEEE Trans. Robotics and Automation*, 7(3), pp.278-287, June 1991.
- [4] Hocaoglu, C, and Sanderson, A.C., "Planning Multi-Paths using Speciation in Genetic Algorithms", Proceedings of the 1996 IEEE International Conference on Evolutionary Computation, Nagoya, Japan, pp.378-383, May 1996.
- [5] Foux, G., Heymann, M., Bruckstein, A., "Two-Dimensional Robot Navigation Among Unknown Stationary Polygonal Obstacles", *IEEE Transactions on Robotics and Automation*, vol.9, pp.96-102, 1993.
- [6] Glover, F., *Tabu Search — Part I*, ORSA Journal on Computing, Vol.1, No.3, pp.190-206, 1989.
- [7] Glover, F., *Tabu Search — Part II*, ORSA Journal on Computing, Vol.2, No.1, pp.4-32, 1990.
- [8] Latombe, J.C., *Robot Motion Planning*, Kluwer Academic Publishers, 1991
- [9] Lin, H.-S., Xiao, J., Michalewicz, Z., *Evolutionary Navigator for a Mobile Robot*, Proc. IEEE Int. Conf. Robotics and Automation, San Diego, May 1994, pp. 2199-2204
- [10] Lumelsky, V.J., "A Comparative Study on the Path Length Performance of Maze-Searching and Robot Motion Planning Algorithms", *IEEE Trans. Robotics and Automation*, 7(1), pp.57-66, Feb. 1991.
- [11] Lumelsky, V.J., and Stepanov, A.A., "Path Planning Strategies for a Point Mobile Automaton Moving amidst Unknown Obstacles of Arbitrary Shape", *Algorithmica*, vol.2, pp.403-430, 1987.
- [12] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, 3rd edition, 1996.
- [13] Michalewicz, Z., Xiao, J., Trojanowski, K., *Evolutionary Computation: One Project, Many Directions*, Proc. 9th International Symposium, ISMIS '96, Zakopane, June 9-13 1996, pp. 189-201
- [14] Michalewicz, Z., Xiao, J., *Evaluation of Paths in Evolutionary Planner/Navigator*, Proceedings of of the 1995 International Workshop on Biologically Inspired Evolutionary Systems, Tokyo, Japan, May 30-31, 1995, pp.45-52
- [15] Xiao, J., Michalewicz, Z., Zhang, L., *Operator Performance of Evolutionary Planner/Navigator*, Proceedings of of the 3rd IEEE ICEC, Nagoya, May 20-22, 1996,
- [16] Xiao, J., Michalewicz, Z., Zhang, L., and Trojanowski, K., *Adaptive Evolutionary Planner/Navigator for Mobile Robots*, submitted for publication, 1996.