A bit of everything, but not all of anything: sensory and computational tradeoffs for agents in collaborative collection tasks

Paul Schermerhorn  Matthias Scheutz
Artificial Intelligence and Robotics Laboratory
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556
{pscherm1,mscheutz}@nd.edu

Abstract

The multiagent object collection task requires a group of agents to collect all objects in an environment in the shortest time possible. This paper describes a simple prediction mechanism that improves performance on the collection task by avoiding objects that other agents are predicted to collect. Agents’ sensory range is also varied, providing two architectural enhancements that are compared in a subsequent cost-benefit analysis. In terms of “raw performance”, simple reactive agents are seen to perform similarly to deliberative agents for a wide range of sensor ranges. A hypothetical cost-benefit analysis is performed to determine the best agent architecture for each configuration of the collection task examined.

1. Introduction

Often when agents need to work together, complex cooperation mechanisms are implemented to ensure optimal performance. Communication is often involved to determine what other agents are doing and to communicate the agent’s intentions to others. However, in many cases it is possible to use much simpler strategies to accomplish acceptable performance (often only slightly worse than the expensive optimal solution). Determining the best architecture for a task involves comparing performance-cost tradeoffs for candidate architectures. Whereas for natural agents this comparison is conducted implicitly (i.e., efficient architectures are favored by selection), for artificial agents the comparison must be conducted explicitly by the designer.

This paper conducts an investigation into the performance-cost tradeoffs of four candidate architectures for multiagent object collection tasks in which a group of agents is required to collect a set of items randomly placed in an environment in the shortest time possible. Reactive and deliberative architectures are introduced and serve as bases for enhanced architectures using two extensions. The first enhancement is a simple physical extension of sensory range. Increasing the agent’s ability to detect remote items is an effective way of improving performance without incurring costly computation and communication costs. However, as the sensory range increases, the cost also increases, so there comes a point at which increasing the range is not worth the additional cost.

The second architectural enhancement employs a simple predictive mechanism to allow agents to select actions that are less likely to be in conflict with other agents’ actions. This reduces the amount of redundant work performed by individual agents, making their overall efficiency on the group task better (i.e., reducing the time it takes the group to complete the task). Adding a predictive component to the architecture also increases the cost, but by a constant factor, unlike range enhancements.

This paper proceeds as follows: we first define the task that forms the context in which the architectures will be evaluated. We then describe the base architectures as well as the architectural enhancements tested in our experiments. The experimental design and setup is given, and results are presented. This is followed by an example of the evaluation of the architectures in light of the experimental results using hypothetical values for component costs. Finally, we conclude with some discussion of the implications of this research.

2. Methods

The multiagent object collection task requires a group of $A$ agents to collect $C$ items in an environment while avoiding $B$ obstacles. Agents and items are randomly placed, and the goal is to minimize the total time required to collect all $C$ items, rather than to maximize the
number of items collected by any single agent. The optimal solution can be determined by exhaustively searching all assignments of items to agents and all collection orders within each assignment, but the task becomes computationally intractable with only relatively few agents and items. Furthermore, such an approach is a static, offline solution; if the environment is likely to change during the course of the task, the expense of recomputing the optimal solution must be incurred again.

What is needed is a flexible, online solution that inexpensively produces good (even if not optimal) behavior at a low cost. Simple reactive agents can be successful at foraging at a very low cost, employing a straightforward greedy collection procedure: if you see an item, collect it; if you see multiple objects, collect the closest. This can be improved upon by adding deliberative capacities that allow the agent to plan efficient routes around obstacles and to remember the locations of previously sensed items. The added performance for deliberative agents may be worth the additional cost in some environments.

The task applies directly to artificial agents attempting to gather items in an environment (McFarland, 1994, Carmenta and Hallam, 1999, Drogo and Ferber, 1992, Stergaard et al., 2001, Araujo and Grupen, 1996). If a number of landers were dropped onto another planet for the purpose of gathering a set of geologically interesting rock samples, for example, they would need to collect their targets as quickly as possible in order to complete their mission before running out of fuel. Furthermore, landing rovers on distant planets is an inexact science, with locations specifiable only in the most general terms and catastrophic failures a distinct possibility. It is, therefore, impossible to create a static plan ahead of time, ready to execute upon landing. Requiring the rovers to compute the optimal solution upon landing is impractical (or impossible), and the conditions may change during the course of the task (e.g., a rover may fail, leaving the others to complete its mission).

The lessons learned here also apply to biological agents (Koza et al., 1992, Schulz et al., 2003, Nishihawa and Nakano, 1998). When a group of animals is foraging, there is no notion of completing the task in the shortest amount of time; the animals are not working together to accomplish any task, but rather are operating independently to achieve their individual goals of survival and reproduction. However, the decision-making problem is much the same in this context as in the artificial task: of the food items I can sense, which would be the best for me to gather right now? To the extent that the solution to the artificial task relies on agents working in their own best interest (and as a side effect improving the overall group performance), it can be applied to understanding animal foraging. It is just as important for animals to avoid wasting time and resources in their foraging efforts—inefficient foragers are less likely to survive. Furthermore, successful foraging strategies are more likely to be passed on to subsequent generations than poor strategies, since their possessors are more likely to reproduce.

2.1 Reactive and Deliberative Agent Architectures

We employ two basic architectures in this study. The first is a reactive architecture that maps perceptions onto actions directly. All reactive agents process sensory information and produce behavioral responses using a motor schema-based approach (Arkin, 1989). Let $\text{Ent} = \{c, h, a\}$ be an index set of the three types of objects: items, obstacles and agents. For each object type $\text{Ent}$, a force vector $F_i$ is computed, which is the sum, scaled by $1/|v|^2$, of all vectors $v$ from the agent to the objects of type $t$ within the respective sensory range, where $|v|$ is the length of vector $v$. These perceptual schemas are mapped into motor space by the transformation function

$$T(x) = \sum_{t \in \text{Ent}} g_t \cdot F_t(x)$$

where the $g_t$ are the respective gain values of the perceptual schemas. The gain values simply scale the effect of sensory input, providing a means by which to prioritize certain inputs (e.g., if collecting items is especially important, the item gain value could be higher than the agent gain value, so that sensing an item has a greater impact on the direction chosen than sensing other agents). These gain values are initialized to values determined to be reasonable via a series of experiments, and are kept constant throughout the life of a reactive agent.

A collision detection mechanism involves an agent’s retreat reflex whenever it detects an impending collision with an obstacle or another agent (all collisions are fatal). The reflex works by inserting a very strong vector leading away from the site of the near collision. This vector is included for a random number of cycles between 5 and 15, and has the effect of moving the agent directly away from the object or agent. The reflex works well in most cases, although it is possible to fail in some situations (e.g., it may be possible to retreat into another obstacle in some circumstances).

Previous results (Schutz and Schmerhorn, 2003) demonstrated the need for a more effective way of dealing with cases in which no collection items are sensed. In these cases, agents simply continued on their current paths until they reached the edge of the world, and then would stay there. This behavior resulted in many experimental runs not finishing successfully, particularly in tests with fewer agents. For this study, a random walk mechanism was introduced that changes the heading of
the agents when they have not recently sensed any food items. Also, agents no longer stay near the edge of the world, but now will “bounce off”, as if the edge were a wall. These improvements dramatically reduce the number of task failures.

Reactive agents always behave in the same way, given that their gain values are constants: their positive $g_c$ makes them employ a greedy collection strategy (most of the time a “collect nearest” strategy (Spier and McFarland, 1998)), whereas their negative $g_a$ and $g_o$ values make them avoid obstacles and other agents. The effect of $g_a$ on the reactive agents’ behavior is to establish implicitly a “ranking” of who gets to collect an item first if multiple agents attempt to collect the same item: whoever is closest will be more strongly attracted to the item than repelled by the other agents, and hence be able to get to collect the item, whereas the other agents will be repelled more by the presence of agents than they are attracted to the item, and hence will move away. In a sense, $g_a$ implements a simple “coordination” strategy, if only one that is “negatively” determined.

The second basic architecture employed in this study is deliberative. Deliberative agents have several components that allow them to manipulate representations of collectible items in the environment. Most importantly, they have a route planner that can determine which item is closest to them and how they can best get to it. It is first and foremost this ability of being able to represent entities in the environment that opens up further possibilities such as storing and retrieving representations, using them in planning and plan execution, etc. None of these possibilities are available to reactive agents, which have access to sensed objects only in a holistic manner (via agglomerated force vectors).

The planner of the deliberative agents (based on a simplified version of the $A^*$ algorithm (Pearl, 1982)) is given a list of items known to the agent (i.e., stored in the agent’s memory), and returns a plan, which is a list of headings and distances, of how to get to the nearest reachable item. The plan is then passed to a plan execution mechanism, which ensures that plan steps are executed. When other agents cross a deliberative agent’s route and the reflex is triggered, “re-planning” is initiated, and the agent will continue by executing the new plan. Re-planning is also performed if the item chosen by the agent has been collected by another agent in the meantime. A further difference between deliberative and reactive agents is that, while the schema-based mechanism of the reactive agents will not pick out the most direct route to an item (because of the influence of other items and agents), and may even move away from the nearest goal item (because of a cluster of objects further away in the opposite direction, or a cluster of agents in the direction of the nearest item), deliberative agents will find the nearest item and plan a route directly to it (while avoiding other agents), thus saving time and energy. In the event that there are no items to be collected in a deliberative agent’s sensory range or in its memory, it will revert to reactive behavior to forage for items using the random walk mechanism described above.

2.2 Architectural Enhancements

We enhanced the basic architectures in two ways, one a physical modification and the other a control-system modification. The first enhancement is simply to extend the sensory range. All agents share the same sensor configuration, including smell, vision, and sonar. Reactive agents employ smell to create the vector for items to be collected and sonar to create the agent and obstacle vectors. Note that smell and sonar do not provide the location of any individual agent, but rather contribute to the summed influence of all items (in the case of smell). Normal reactive agents do not use vision, but the prediction extension described below uses it to locate individual agents and items. Deliberative agents use smell and sonar in the same way reactive agents do, although in most cases the deliberative layer suppresses their effects in order to execute a plan; only when there is no item in sensory range do the reactive mechanisms take effect. Vision is used by deliberative agents to locate individual items, agents, and obstacles in order to store them in memory and generate plans for collecting items.

Extending the sensory range of the agents is an intuitive way to improve their performance. For one thing, the more information an agent has about its surroundings, the better its decisions can be. Furthermore, with very small sensory range, it will often be the case that agents simply do not perceive any collection items, making it impossible for them to target any item for collection and requiring them to revert to the random walk behavior described above. However, sensor range increases with diminishing returns, especially as the range approaches the size of the environment, as later discussion will demonstrate.

The second enhancement we explore here is a primitive prediction mechanism. The goal of the mechanism is to prevent duplication of effort by trying to avoid more than one agent attempting to collect a single item. The prediction mechanism functions as a perceptual filter that makes “educated guesses” as to which items it would be fruitless to pursue and excluding those from influencing the agent’s behavior. For each agent in the current agent’s visual range, the closest item to that agent is located. If that item is closer to the other agent than to the current agent, it is excluded on the assumption that the other agent will attempt to collect it.

What is interesting about this prediction mechanism is that it directly benefits the individual agent to use it, however, to the extent that it works, it also contributes
to the group’s goal of collecting all items in the shortest possible time. The benefit to the individual is clear: if another agent is guaranteed to collect an item before the current agent arrives, selecting that item as a target for collection will lead to expense without the possibility of reward. Thus, it is in the best interest of the agent to ignore that item anyway. The emergent behavior produced is an implicit form of cooperation (i.e., the effect of the prediction mechanism is as though the agents are cooperating, even though from their perspective, they are operating in their own self-interest).

Adding the prediction mechanism to the base agent types results in four agent types: normal reactive, reactive-predictive, normal deliberative, and deliberative-predictive. Reactive-predictive agents prevent items identified as potentially targeted by other agents from being added to their perceptual schemas. Deliberative-predictive agents do not add these items to memory, preventing them from being identified as goals for the planner.

Note that this simple prediction is far from perfect. It will not always filter the items targeted by other agents. For example, if two other agents share the same closest item, one of them will (using the same prediction mechanism) filter out that item and pursue another item. Also, the mechanism may correctly filter out the closest item to another agent, but will not notice that the item the other agent will pursue after that should also be filtered because it, too, is closer to the other agent than to the current agent. Situations such as these will lead to duplication of effort even in predictive agents, but as we show below, even this imperfect prediction can improve performance substantially. More sophisticated prediction is, of course, possible, but would be significantly more costly in terms of computational resources.
3. Experiments and Results

In order to gauge whether increased sensory range or the prediction mechanism would improve performance (i.e., decrease the time to completion) for the collection task, we conducted a series of experiments in simulation. The artificial life simulator SWAGES under development in our lab was used as a platform in which implementations of the four architecture types (Reactive, Reactive-Predictive, Deliberative, and Deliberative-Predictive) were tested. The world in which the agents operate is 800 by 800, and is bounded. The number of agents \( A \) was varied from 1 to 5. The collection task was defined for ten items \( C = 0 \), and was conducted in environments free of obstacles \( B = 0 \) as well as environments containing five obstacles \( B = 5 \). Finally, the agents’ sensory range was varied from 100 to 800, with a step of 50.

Each experiment consists of 38 experimental runs using different randomly generated initial conditions. Each set of 38 initial placements of agents and items is used for all experiments with identical \( A \), and all obstacle environments share the same positions for obstacles. This allows us to compare directly between agent types and sensory ranges, as well as between environments with and without obstacles. The figures reported here are the average performance (1/average cycles to completion) of each experiment set for each architectural configuration. In addition, these averages are scaled to the performance of the optimal solution (where optimal performance is 1.0) in Figures 1 to 10. The performance of the optimal solution was calculated separately for each set of initial conditions via an exhaustive search of all possible assignments of items to agents and all permutations of order for each assignment. Unlimited sensory range is assumed in the optimal solution, making it the best possible performance on the task. The figures, therefore, depict the percentage of optimal performance obtained by agents of each architectural configuration. Note that the optimal performance is different for each value of \( A \), so while it may appear at first glance as though the performance of the agents is decreasing as \( A \) increases, in reality it is only performance relative to the optimal that is decreasing, reflecting the overhead (or inefficiency in agent distribution) incurred with additional agents (Scheutz and Schermerhorn, 2003).

Figures 1 to 5 contain performance data for 1 to 5 agents in environments with no obstacles. When there is only one agent, predictive versions perform identically to their base types. Increasing sensor range increases performance quickly at first, then tapers off at around 400 or 450, at which point increased sensory ranges do not contribute to better performance. Deliberative agents achieve better than 0.8 times the optimal performance, whereas reactive agents achieve just over 0.6. However, it is when two or more agents are working on the collection task that we see interesting patterns emerge. All agent types see their performance rise and then level off, but at different sensory ranges. Performance of deliberative types tends to level off at lower sensory range than performance of reactive types (see Figures 2 and 3). Similarly, non-predictive types level off at lower ranges than predictive types. Thus, we see reactive-predictive agents leveling off at around range 300 in two- and three-agent experiments, whereas other agent types level off at ranges between 250 and 400. This is an indication that reactive-predictive agents are better able to take advantage of extended sensor ranges, at least for these agent quantities. Alternatively, it could be argued that the other agent types utilize additional sensor range more efficiently, reaching their leveling-off point earlier than reactive-predictive agents.

It is interesting that reactive-predictive agents begin to outperform non-predictive deliberative agents immediately when the quantity of agents is more than one, and by the time three agents are working on the task, reactive-predictive agents are performing virtually identically to deliberative-predictive agents at sufficiently high sensor ranges. Also of note is the fact that with five agents, performance of normal reactive agents actually begins to decrease for sensor ranges higher than 300. One possible explanation for this is that more agents see the same item and go in the same direction, interfering with one another and distributing the agents in the environment disadvantageously.

Performance in environments with five obstacles is depicted by Figures 6 to 10. The patterns that emerge...
Figure 6: Average performance of one-agent experiments in five-obstacle environments for sensory ranges from 100 to 800 (scaled to optimal performance).

Figure 9: Average performance of four-agent experiments in five-obstacle environments for sensory ranges from 100 to 800 (scaled to optimal performance).

Figure 7: Average performance of two-agent experiments in five-obstacle environments for sensory ranges from 100 to 800 (scaled to optimal performance).

Figure 10: Average performance of five-agent experiments in five-obstacle environments for sensory ranges from 100 to 800 (scaled to optimal performance).

Figure 8: Average performance of three-agent experiments in five-obstacle environments for sensory ranges from 100 to 800 (scaled to optimal performance).
in these figures are very similar to those from the previous set. Once again, reactive agent types tend to level off at higher ranges than deliberative agent types, and predictive versions level off at higher ranges than their non-predictive base versions. Reactive-predictive agents again begin to outperform deliberative agents with just two agents, and with three or more are performing similarly to deliberative-predictive agents given sufficient sensor range. One interesting difference between zero- and five-obstacle environments is that normal reactive agents consistently underperform normal deliberative agents in five-obstacle environments, even if it is only by a small amount. This is a result of deliberative agents having the ability to plan efficient routes around obstacles, rather than having to take indirect routes around them as the reactive agents are forced to do by their schema-based approach.

4. Discussion
In terms of computational complexity, an ordering between the four agent types examined here looks like this: normal reactive agents have the lowest complexity.

Figure 11: Performance-cost ratio of one-agent experiments in zero-obstacle environments for sensory ranges from 100 to 800.

Figure 12: Performance-cost ratio of two-agent experiments in zero-obstacle environments for sensory ranges from 100 to 800.

Figure 13: Performance-cost ratio of three-agent experiments in zero-obstacle environments for sensory ranges from 100 to 800.

Figure 14: Performance-cost ratio of four-agent experiments in zero-obstacle environments for sensory ranges from 100 to 800.

Figure 15: Performance-cost ratio of five-agent experiments in zero-obstacle environments for sensory ranges from 100 to 800.
Figure 16: Performance-cost ratio of one-agent experiments in five-obstacle environments for sensory ranges from 100 to 800.

Figure 17: Performance-cost ratio of two-agent experiments in five-obstacle environments for sensory ranges from 100 to 800.

Figure 18: Performance-cost ratio of three-agent experiments in five-obstacle environments for sensory ranges from 100 to 800.

Figure 19: Performance-cost ratio of four-agent experiments in five-obstacle environments for sensory ranges from 100 to 800.

\(O(|A| + |B| + |C|)\), the complexity of summing the vectors of all entities, followed by reactive-predictive agents \(O(|A| \cdot |C|)\), the complexity of the predictive mechanism, which must examine each item for each agent in the worst case. Deliberative agents have by far the highest complexity as a result of the exponential cost of the \(A^*_s\) planner.

More complex computations will require more complex computational architectures, in turn leading to higher structural costs (i.e., costs associated with possessing and maintaining an architectural component). Given the significant difference in structural cost between deliberative and reactive architectures and the relatively similar performance on the collection task, it will be reasonable in at least some cases to select a reactive agent architecture for the collection task instead of a deliberative architecture.\(^2\) architectures will outperform their simpler opponents sufficiently to warrant their increased cost.) Furthermore, reactive and reactive-predictive architecture are easy to implement on robots in the laboratory. The following discussion will, therefore, focus on the tradeoffs between reactive agents of both kinds with varying sensor ranges.

Determining the best architecture for a task will involve examining the performance to cost ratio for each architecture under consideration. For reactive and reactive-predictive architectures, the structural cost will include some base cost \(a\) that is common to all configurations. The cost of additional architectural components can be expressed in terms of the base cost.

For example, the cost of increasing sensor range by a particular increment \(R\) could be expressed as \(\frac{d}{R}\) (where \(d\) is some constant), making the overall cost of increasing sensor range \(C_{\text{sensors}}(\text{range}) = \frac{d}{R} \cdot \text{range}\). Thus, increasing the sensor range of an architecture will increase its structural cost as a function of the amount of increase.

\(^2\)However, the collection task as defined here is very simple; there will be ecological niches in which more complex (e.g., deliberative)
The cost of the class of reactive architectures with various sensor ranges is $C_{\text{reactive}}(\text{range}) = a + \frac{a}{c} \cdot \frac{\text{range}}{R}$. (Note that for the purposes of this example, the increase in cost is linear on the increase in range; it may well be that the increase is instead quadratic or even higher order, in which case the results would look somewhat different.)

Adding the predictive mechanism to an architecture will also increase the structural cost. However, in this case, the additional cost is a constant amount, say, $\frac{\alpha}{c}$. That makes the cost of reactive-predictive agents $a + \frac{\alpha}{c}$. Coupling that with the potential for extended sensor ranges yields a structural cost function of $C_{\text{reactive-predictive}}(\text{range}) = a + \frac{\alpha}{c} + \frac{\beta}{d} \cdot \frac{\text{range}}{R}$.

Consider, for example, the case in which $a = 50$, $d = 10$, and $c = 10$. Furthermore, assume that the value of $R$ (i.e., the increment of increase in sensor range) is 50, as in the experiments described above, and that the cost of the initial range of 100 is included in $a$. This makes the cost function for normal reactive agents $C_{\text{reactive}}(\text{range}) = 50 + \frac{\text{range} - 100}{50} \cdot 5$. Similarly, reactive-predictive agents will incur structural costs of $C_{\text{reactive-predictive}}(\text{range}) = 50 + 5 + \frac{\text{range} - 100}{50} \cdot 5$.

Figures 11 to 20 plot the performance-cost ratios for reactive and reactive-predictive agents in the experiments described above using these assumed costs. In each case, the values plotted are the ratios divided by the number of agents working on the task. Determining the optimal range for a given agent type (reactive or reactive-predictive) is simply a matter of finding the maximum ratio for that type. For example, in Figure 12, the maximum for normal reactive agents is at range = 350, whereas for reactive-predictive agents it is at range = 300.

Determining which agent type (predictive or non-predictive) is best for a given range is a matter of finding the type whose ratio is higher at that range. For example, in Figures 11 and 16, normal reactive agents have a better performance-cost ratio for every value of range. This makes sense, because these are single-agent tests, so their performance is the same, whereas reactive-predictive agents incur an additional constant cost. In other cases, there may be a point (or more) at which the curves cross, meaning that for some values of range one agent type is better, while for others it is worse.

Finding the best combination of range and predictive capacity requires one to locate the global maximum of performance-cost ratio over all agent types and ranges. In the single-agent, zero-obstacle case (Figure 11), that maximum is achieved by normal reactive agents with a sensory range of 450. However, for the five-agent, five-obstacle case, the maximum is achieved by reactive-predictive agents with a range of 330 (Figure 20). In fact, for all multi-agent experiments, reactive-predictive agents with a medium range tend to have the best performance-cost tradeoff. The performance values used in computing the performance-cost ratios are unscaled, so it is possible to compare across figures to determine the best number of agents, as well. Given the cost functions used here, the best performance-cost ratio is obtained using five reactive-predictive agents with range = 330 (Figures 15 and 20).

This type of analysis will allow designers to choose the best combination of architectural features for a given task, making explicit the fact that raw performance does not tell the whole story, but cost must also be a consideration when selecting architectures and architectural components. Natural selection will perform the same type of analysis implicitly; biological agents that forage successfully but at a slow rate must also incur low costs if they wish to survive. Conversely, an agent type may forage very well, acquiring food sources at a very high rate, but if the cost of doing so is too high, those agents will not survive in the long run. When natural selection does find a maximum point (either local or global), it will be difficult for it to change. Removing the prediction component, to use the example explored here, will reduce the performance-cost ratio, making such agents less likely to survive than agents maintaining the component. Similarly, increasing or decreasing sensory range will lead to lower performance-cost ratios, again making it difficult to compete against agents whose range is at the level selected by the maximum.

5. Conclusion

In this paper we describe two agent architectures, reactive and deliberative, and two simple extensions to them. Predictably, increasing the sensory range yields increased performance, up to a point. Implicit cooperation emerges via the use of the primitive prediction mechanism tested here, and while its performance is not optimal, it increases performance significantly, al-

Figure 20: Performance-cost ratio of five-agent experiments in five-obstacle environments for sensory ranges from 100 to 800.
loving reactive-predictive agents to outperform nonpredictive deliberative agents and to perform on a par with deliberative-predictive agents. This is true even in environments containing obstacles, which previous research indicated favored deliberative agents in terms of raw performance (Schetz and Schermerhorn, 2002).

To account for architectural costs, we also conducted a performance-cost analysis of reactive and reactive-predictive agents using a hypothetical set of values for architectural costs. This illustration demonstrates the utility of such analyses, and points to a need for a more systematic treatment of cost that will allow designers to compare divergent architectures.

The lessons learned here do not apply only to artificial agents, however. The multiagent object collection task is (as previously mentioned) similar in many ways to animal foraging “tasks,” and the utility of the prediction mechanism would apply to them as well. The combination of structural (range) and processing (predictive) enhancements suggests that such mechanisms will coevolve in biological agents, with the eventual values depending on the interplay between the two and their environment. The simulation results reported in this paper demonstrate that there is no “black and white” answer to whether having prediction is better than not having it, or whether greater sensory range is more beneficial. For some sensory ranges, nonpredictive control is better than predictive control and vice versa. The cost-benefit analysis performed explicitly in this paper is implicitly performed by natural selection which favors individuals with higher performance-cost ratios.

References


