Proc. of IEEE International Conference
on Robotics and Automation, Pasadena, CA, 2008.

Towards Schema-Based, Constructivist Robot Learning: Validating an Evolutionary Search Algorithm for Schema Chunking

Yifan Tang and Lynne E. Parker
Distributed Intelligence Laboratory, Department of Electrical Engineering and Computer Science
The University of Tennessee
203 Claxton Complex, Knoxville, Tennessee 37996-3450
Email: {ytang, parker}@eecs.utk.edu

Abstract— **In this paper, we lay the groundwork for extending our previously developed ASyMTRe architecture to enable constructivist learning for multi-robot team tasks. The ASyMTRe architecture automatically configures schemas within, and across, robots to form the highest utility solution that achieves a given multi-robot team task. We believe that the schema-based approach used in ASyMTRe is a useful abstraction not only for forming heterogeneous coalitions, but also for enabling constructivist learning, in which chunks of schemas that solve intermediate subproblems are learned and then made available for future task solutions. However, the existing ASyMTRe search algorithm for finding configurations of schemas that completely solve given tasks (Centralized ASyMTRe – CA) is not well-suited for identifying useful chunks of schemas that could solve intermediate subtasks that may be useful in the future. Thus, in this current work, we explore an Evolutionary Learning (EL) technique for the offline learning of schema chunks that could be saved and used later in an online search (using the regular CA algorithm) for coalition configurations. However, we do not want to sacrifice solution quality in making use of the evolutionary search technique. Thus, we compare the solutions discovered by the EL algorithm with those that are found using CA, as well as with a third algorithm that randomizes the CA algorithm, called RA. Four different applications in simulation are used to evaluate the EL, CA, and RA techniques. Our results show that the EL approach indeed finds solutions of comparable quality to the CA technique, while also providing the added benefit of learning highly fit partial solutions, or schema chunks, that may be beneficial for future tasks via constructivist learning. We conclude by arguing that the combination of the online CA search for solving current multi-robot tasks can be combined with the offline EL approach that can identify intermediate solutions (or schema chunks) that may be useful for future team tasks. This combination should lead to an overall efficiency improvement for identifying coalition formations, as well as for continual learning.**

I. Introduction

In prior work, Tang and Parker [8], [11] developed the ASyMTRe approach to automatically generate robot team task solutions for coalitions[1] performing multi-robot tasks. The ASyMTRe approach is inspired by the theory of information invariants [5] and schema theory [1], and finds team task solutions by configuring the schema building blocks on each robot such that the resulting configuration achieves the specified task with the lowest cost possible. Because the challenge of locating a low-cost configuration of schemas across multiple robot team members is an NP-hard search problem [12] (which is also true for other task allocation problems), the ASyMTRe search algorithm that finds potential coalitions is based loosely on the findings of Shehory [10], who showed that for non-super-additive domains, better solutions consist of smaller coalition sizes. These concepts are implemented in ASyMTRe through heuristics that direct the search toward smaller team solutions first. Parker and Tang showed, through empirical evaluations, that the heuristic-based centralized ASyMTRe search algorithm[2] generates very good solutions quite quickly for several types of applications. A major benefit of this approach is that it enables robots to easily share sensory, computational, and effector capabilities in solving challenging multi-robot tasks.

In continuing work, our objective is to extend the ASyMTRe architecture to enable constructivist learning in the multi-robot team. Constructivist learning is a method for learning new knowledge and skills based upon past experience; this type of learning is recognized to be a common method used by humans from infancy to adulthood for lifelong learning [2]. Because much of human learning seems to be based on schema building blocks, our intent is to build upon our schema-based abstraction of ASyMTRe to enable constructivist robot learning. We believe that collections of schemas, called "chunks", analogous to the *Sensory Computational Systems (SCSs)* of Donald's information invariants theory [5], could be learned. Most of the chunks present intermediate solutions to the search problem. (In this paper, "schema chunk", "SCS", "intermediate solution", and "partial solution" are used as synonyms). Ultimately, our objective is to enable robot teams to learn and build up chunks constructively, in order to store knowledge from previous search processes, and to improve the efficiency for future searches.

However, the current solution search strategy of ASyMTRe does not construct chunks that would be amenable to this constructivist learning process. Thus, the goal of our current research is to determine if an alternative search strategy can have the benefit of facilitating constructivist learning in multi-robot teams, and if so, how its solution quality compares to other possible search techniques. The

[1] The coalition search problem is the problem of finding the appropriate combination of single-task robots that collectively perform multi-robot tasks using instantaneous assignment (taxonomized as ST-MR-IA, per [7]).

[2] Tang and Parker also implemented a distributed version of ASyMTRe [8], [13]. For simplicity, we focus on the centralized version in this paper.

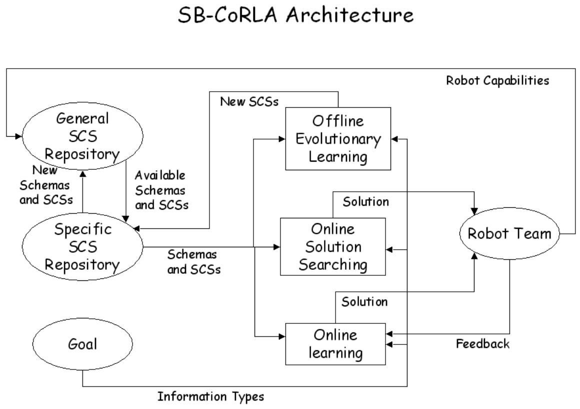


Fig. 1. SB-CoRLA Architectural Overview.

long term goal of our research is to develop a Schema-Based, Constructivist Robot Learning Architecture, which we call "SB-CoRLA", that can combine online solution searching and offline learning, in order to find a better solution more efficiently. The general idea is that the existing online ASyMTRe search technique, which we call CA (for Centralized ASyMTRe), will still be the main algorithm for finding online coalition solutions. However, a new offline learning process will be used to explore the schema base and identify chunks that can be used in later online search processes to speed up the search for solutions. Figure 1 shows the high level architecture of SB-CoRLA, which consists of three major processes: online searching, offline learning, and online learning. The general SCS repository is the knowledge base of the system that stores the original schemas and the more complex, learned SCSs (schema chunks). The specific SCS repository is a subset of the general SCS repository and is created based on each specific robot team's configurations and available capabilities of the current robots. When the robots are not busy performing tasks, the offline learning searches for highly-fit partial solutions (or chunks) based on the existing SCS repository. These partial solutions are then saved in the repository for future use. When the robot team has an immediate task to perform, the online search generates task solutions for the robot team based on the goal definition and the existing SCS repository. The robot team then carries out the task solution with the lowest cost. Online learning is the process of goal-directed feedback-based learning that modifies existing schemas to generate new schemas. This aspect of the approach is the subject of future research.

A search technique that we believe would be appropriate for offline learning of schema chunks is an evolutionary search technique. An evolutionary search technique could make use of a genetic algorithm to search the solution space by repeatedly combining highly-fit intermediate solutions to generate lower-cost complete solutions. We have developed such an evolutionary search technique called "EL" (Evolutionary Learning). EL is of particular interest to our constructivist learning objective, since we believe that the highly-fit intermediate solutions found in the evolutionary search can be beneficial as higher-level building blocks for constructivist learning.

However, we do not want to sacrifice solution quality in making use of the EL search technique. Thus, we must determine how the solution quality of the EL technique compares to alternative search techniques, including the original ASyMTRe search algorithm (CA), to ensure that we do not sacrifice solution quality. Therefore, the objective of this current paper is to determine the viability of the EL technique for finding reasonable partial solutions, as compared to the alternative search possibilities. To explore alternative search techniques, we compare the centralized version of the previously implemented ASyMTRe search algorithm (CA) with a Randomized ASyMTRe search algorithm (RA). RA makes use of the same fundamental search algorithm of ASyMTRe, but rather than making a greedy heuristic search of the potential multi-robot team task solutions, it randomly selects possible solutions. Because the partial solutions generated by EL will ultimately be used as higher-level building blocks by online search algorithms, we compare the cost of the partial solutions with the cost of the solutions generated by CA and RA. The cost of a solution is the sum of the costs of the active schemas used in the solution. If a robot is not assigned a task in a partial solution, no schema is activated for that robot, hence the cost for an unassigned robot is zero. Only if the solution cost of EL is comparable to (or lower than) CA or RA for the coalition formation problem does it make sense to us to use this technique as a foundation for constructivist learning in multi-robot coalitions.

The remainder of this paper is organized as follows. Section II presents the three search algorithms: the centralized ASyMTRe search algorithm (CA), the Randomized ASyMTRe search algorithm (RA), and the Evolutionary Learning search algorithm (EL). Section III describes the simulated applications and the settings used to study the alternative algorithms, followed by a discussion of the results in Section IV. Section V describes related work. Section VI concludes with some summary remarks.

II. The Algorithms

The CA, RA, and EL algorithms use the schema-based abstraction implemented in ASyMTRe [8], in which the search space consists of basic *schemas* [1], each of which requires and produces certain input(s) and output(s) called *information types*. The schemas represent basic robot capabilities and are categorized into perceptual schemas, motor schemas, and communication schemas. The inputs and outputs of schemas can be interconnected if their information types match. For instance, if schema S_a produces the global positions of a robot, and schema S_b needs the global positions of the robot to compute motor control commands such as speed and turning direction, then S_a can be connected to S_b via the information type "global position", to enable S_b to produce the information types "speed" and "turning direction". The input to perceptual schemas must come from sensors or other perceptual schemas. Communications schemas can pass information between perceptual and/or motor schemas across multiple robots, thus enabling robots to share information between them. The input to motor schemas comes from perceptual or communications schemas. The output from the motor schemas controls the robot's motors. The ASyMTRe process of automatically connecting the schemas through matching information types defines the information flow through the multi-robot system, thus generating the behavior control for the robot coalition.

The solutions for all three search algorithms consist of combinations and interconnections of active schemas on each robot in the robot team that allow the team to accomplish the task. This section describes the approaches used by each of the three search algorithms to find these solutions.

A. *Centralized ASyMTRe Search Algorithm (CA)*

The Centralized ASyMTRe algorithm (CA) is a two-step, anytime algorithm for searching for the proper connections of schemas to accomplish the goal task. The first step is to find all potential schema connections that can provide the required information types for a goal in an individual robot[3]. The second step is to instantiate a specific solution on each robot, by sequentially searching through permutation sequences of individual robots until a simultaneous solution for all robot team members is found. For n robots on the team, there are $n!$ permutation sequences that must be analyzed, hence the problem is NP-hard. The CA algorithm attempts to find reasonable cost solutions quickly by using a heuristic to guide its search that finds solutions for the less capable robots (i.e., robots that must be part of the solution, but which have fewer schema resources to work with) first, in order to avoid resource shortages. In this paper, we simplify the original CA cost function and define the solution cost as the cumulative sum of the individual schema costs[4].

The CA algorithm is designed to be an anytime algorithm, so that as soon as a valid solution is discovered, it is made available to the robot team. Thus, the CA approach is a greedy search approach that *theoretically* searches, in an anytime fashion, all $n!$ permutations of robots, selecting the best solution found. Although Tang and Parker showed two applications for which the CA algorithm computes the first solution very quickly (i.e., in a matter of seconds) and within a small multiple of the optimal solution (where it could be computed), it is unclear whether the solutions found are good approximations to the optimal solutions in general.

B. *Randomized ASyMTRe Search Algorithm (RA)*

The Randomized ASyMTRe algorithm (RA) uses the same two-step, anytime search algorithm used by CA. The RA approach first generates potential solutions for each of the robots and then performs a sequential search through each permutation arrangement of robots to assign solutions to individual robots. However, in contrast with the CA approach, the RA approach does not perform a greedy search when assigning solutions to robots. Instead, RA selects viable solutions randomly from among all possible solutions for each robot.

C. *Evolutionary Learning Search Algorithm (EL)*

The Evolutionary Learning (EL) approach makes use of a genetic algorithm that maintains a population of p individuals, each of which represents a configuration of schemas that may be a possible solution to the robot team coalition task or subtask. Algorithm 1 shows the details of the EL algorithm. Table I shows the various parameters that must be defined for EL, and their default values[5].

In the EL approach, an initial population is created that consists of individuals having random connections of schemas, with the following restrictions: first, schemas can only be connected if they have matching information types; and second, connections across different robots (which we call *inter-robot* connections) can only occur between communications schemas. As the initial population is built, the number of interconnections between schemas on different robots and between schemas on the same robot (which we call *intra-robot* connections) are governed by two connection rates specified by the user: the inter-robot connection rate, ρ, and the intra-robot connection rate, κ. Note that these individuals do not necessarily represent complete solutions, since they may not fully (or even partially) solve the task given to the robots. This maintenance of partial solutions during the search process is one of the principal ways in which the EL algorithm differs from the CA and RA algorithms. These partial solutions represents chunks of schemas that solve important subtasks.

As with any genetic algorithm, the fitness value of each individual, F, is determined after each new generation is created through either initialization or evolution. In our definition, F depends not only on the aggregated cost of the active schemas, c, (which is the criterion also used in CA and RA to calculate the cost of the solution), but also on the complexity of the solution, x, and the degree of goal achievement, q and u. The value of x is measured by the total number of schema connections for that solution, and is normalized to the range $[0, 1]$. The degree of goal achievement is measured in two ways: 1) by the percentage of information types that are required by the goal and that are fulfilled (q/q_{max}), and 2) by the percentage of robots that can fulfill their individual goals (u/n). F is calculated as a weighted sum of the normalized values of c, x, q, and u. The weight for each factor is domain-specific and determined by the user.

In our approach, the evolutionary process consists of fitness proportionate selection, single point crossover operations, and single point mutations. In the crossover operation, a crossover point is a randomly selected connection between two random schemas S_i and S_j in one randomly chosen individual solution. In the crossover process, another individual

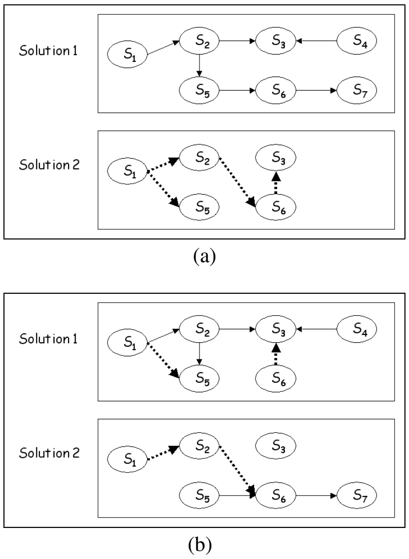
[3] These possible connections are called "potential solutions" for the robot in the ASyMTRe approach.

[4] Solution quality in the original ASyMTRe algorithm combined various factors in addition to cost, such as probability of schema success, in determining the utility of a solution.

[5] We will explore the impact of the mutation rate and the complexity in future work, therefore their default values are set to 0.

Algorithm 1 The Evolutionary Learning (EL) search algorithm.

```
(R, T, F): robot team composition, task, and fitness
n: number of robots in the team
m: number of different kinds of schemas in the system
p: number of individuals in a population
$g_{max}$: max. number of generations
$l_{max}$: max. number of generations without fitness improvement
c: aggregated cost of active schemas
x: complexity of an individual solution
q: number of required information types
u: number of robots that can achieve their individual goals
$c_{max}, q_{max}, u_{max}$: max. value for c, q, and u
1) Generate a list of $nm$ available schemas based on $R$.
2) Initialize the first population of size $p$ by connecting the schemas while respecting the following rules: $[O((nm^2 + n^2)p)]$
   • For each robot $R_i \in R$, randomly at rate $\kappa$, connect schema $S_a$ to $S_b$ if $S_a$'s output information type matches $S_b$'s input information type.
   • Between $R_i$ and each robot $R_j \in R$ ($i \neq j$), randomly at rate $\rho$, connect CS schema $CS_i$ to $CS_j$ if $CS_i$'s output information type matches $CS_j$'s input information type.
3) Calculate $F$ for each individual $p$ using the following formula: $[O(n^2 m^2 p)]$
      $F = w_c * (c/c_{max}) + w_x * x + w_q * (q/q_{max}) + w_u * (u/n)$
4) Repeat for $g_{max}$ generations
   • Select $\xi$ individuals using fitness-proportionate selection for reproduction. $[O(n)]$
   • Randomly at rate $\gamma$, perform pairwise crossover on $p$. $[O(n^2 m^2 p)]$
   • Randomly at rate $\delta$, perform single point mutation on $p$. $[O(nmp)]$
   • Prune each child individual, which is also a partial solution (or complete solution) for $R$. This process includes eliminating invalid and redundant connections, as well as calculating the fitness values. $[O(n^2 m^2 p)]$
   • Record the best solution if its fitness value is better than the best solution thus far.
   • Stop if:
     – Every robot can fulfill its individual goal;
     – Or, $l_{max}$ generations have been generated without fitness improvement;
     – Or, a time limit is reached.
```



(a)

(b)

Fig. 2. (a) Parents before crossover: Solution 1 and Solution 2; (b) Children after crossover: Solution 1′ and Solution 2′. The solid arrows indicate the uni-directional connections between schemas in the parent solution Solution 1, and the bold dashed arrows are the connections in Solution 2. The connection between $S1$ and $S5$ is randomly chosen to be the single crossover point. The connections $S_1 \rightarrow S_5$ and $S_6 \rightarrow S_3$ are swapped from Solution 2 to Solution 1 to create the child solution Solution 1′. The connections $S_5 \rightarrow S_6$ and $S_6 \rightarrow S_7$ are swapped from Solution 1 to Solution 2 to create the child solution Solution 2′. Note that because the connections are uni-directional, connection $S_3 \rightarrow S_4$ is not swapped.

solution is randomly chosen to be the other parent, and then the connection $S_i \rightarrow S_j$ and all the connections that schema S_j are connected to are swapped between the parents. The connections between schemas are uni-directional, indicating the direction of information flow. For example, Figure 2 shows one crossover process at the crossover point $S_1 \rightarrow S_5$. Mutation is the process of randomly adding or deleting a connection in an individual solution. Both crossover and mutation are unguided, random processes.

This evolutionary process is repeated over multiple generations until one of the following conditions is fulfilled: 1) every robot can fulfill its individual goal; 2) the solution quality has not improved in l_{max} generations; 3) a time limit is reached; 4) g_{max} generations have been created.

For n robots, m different kinds of schemas, and p individual solutions in each population, there are up to mn available schemas in the search space, and up to $(mn)^2$ possible ways of connecting the schemas. If we have a maximum of g generations, then the EL computational complexity for initializing a population, performing genetic operations, and pruning and evaluating the generated solutions is $O((mn)^2 pg)$.

III. Simulations

Four applications were implemented in simulation to compare the three algorithms: (A_1) multi-robot transportation, (A_2) box pushing, (A_3) robot formation, and (A_4) limited resource. In these applications, based on the available sensors, each robot possesses different combinations of perceptual schemas. The goal of each search algorithm is to determine which combination of sensors, distributed across which robots, constitutes the lowest cost solution that accomplishes the task.

TABLE I

EL PARAMETERS AND THEIR DEFAULT VALUES.

Name	Description	Default Value
p	population size	500
ξ	number of individuals selected for reproduction	50
γ	probability for crossover	90%
δ	probability for mutation	0
κ	intra-robot connection rate	50%
ρ	inter-robot connection rate	50%
w_c	weight for the aggregated cost of active schemas; used to calculate fitness	0.2
w_x	weight for the complexity; used to calculate fitness	0
w_q	weight for the percentage of information types required by the goal that are fulfilled; used to calculate fitness	0.2
w_u	weight for the percentage of robots that can achieve their goals; used to calculate fitness	0.6

Application A_1 requires robots to help each other (through sharing sensory information) in reaching their goal position. Various methods of sensor sharing are possible in this application, as implemented by Tang [11] on both physical and simulated robots. For example, if a robot has GPS and/or a laser, it can apply a perceptual schema to localize itself. If a robot has a laser and/or a camera, it can apply another perceptual schema to calculate the relative position of another robot within its sensing range.

Application A_2 requires robots to help each other push a box to a goal location. Again, various methods of sensor sharing are possible in this application, as implemented by Tang [11] on both physical and simulated robots. In this application, if a robot has a laser, it can apply a perceptual schema to measure the box's position relative to itself, or activate another perceptual schema to confirm contact with the box. If a robot has a camera, it can detect the goal location and use a third perceptual schema to calculate its push direction. With sonar, a robot can apply a fourth perceptual schema to detect the position of the box.

Tang's implementations on both applications A_1 and A_2 [11] made use of the CA algorithm. Her results showed that CA is capable of solving those two problems with highly satisfactory results.

Applications A_3 and A_4 are designed to test the limitations of CA. They are designed only as abstract applications, and are not implemented on physical robots. Application A_3 models n robots following each other in a long chain. The first robot is the leader of the formation and does not need any information from the other robots. The i-th robot needs a unique information type from the (i-1)-th robot, indicating the (i-1)-th robot's position, so that the i-th robot can maintain the formation.

Application A_4 consists of n robots. In this application, except for the last q robots (robots n, n-1, n-2, ..., n-q+1), all robots can accomplish their goals without help from other robots, i.e., without information communicated by other robots. However, if they can receive external help from other robots, then the cost of the solution can be lowered. The last q robots must receive external help in order to achieve their goals. Only the first h robots (robots 1, 2, ..., h), can offer this external help; hence the name "limited resource."

TABLE II

TIME (IN WALL CLOCK SECONDS) NEEDED FOR CA AND RA TO GENERATE THEIR FIRST SOLUTIONS FOR APPLICATIONS A_1, A_2, AND A_3, FOR A TEAM OF 25 ROBOTS. THE VALUES ARE AVERAGED OVER 10 RUNS.

	A_1	A_2	A_3
CA	0.077	0.151	0.033
RA	0.047	0.158	0.033

The three search strategies were tested using heterogeneous robot teams whose size varied from 5 to 25 robots. A robot can have three different sensors for application A_1 (GPS, laser, and camera), and three different sensors for application A_2 (laser, camera, and sonar). Applications A_3 and A_4 are theoretical tests with only abstract sensors. We compose heterogeneous robot teams by randomly choosing different available sensors and consequently different schemas for each robot. For the EL algorithm in these experiments, the default parameter values shown in Table I were used unless indicated otherwise.

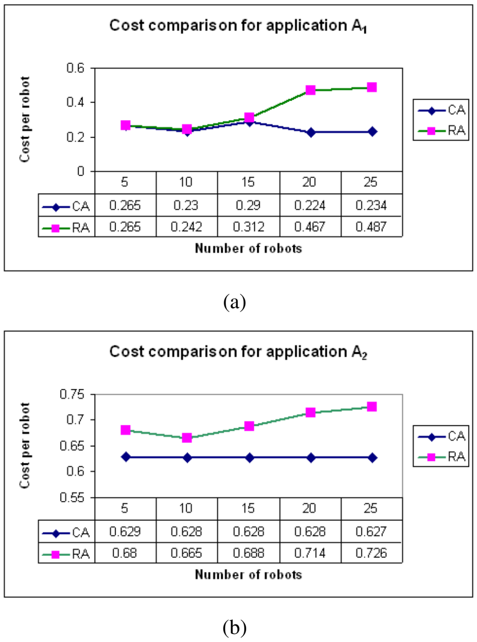
IV. RESULTS AND DISCUSSION

Using the simulation results, we compare the time CA and RA require to generate solutions, the solution quality of CA and RA, the solution improvement over time of EL, and the solution quality of CA, RA, and EL.

Since RA is developed as an alternative online solution search technique to CA, we first compare the times required by CA and RA to generate the first solutions. Table II shows the average time needed for CA and RA to generate their first solutions for a team of 25 robots. These results show that CA and RA are effectively equivalent in the amount of time required to generate a first solution for the applications A_1, A_2, and A_3.

Now comparing CA and RA in terms of cost, Figure 3 shows the cost comparisons for applications A_1 and A_2 for an average of 10 trials running 10 minutes each, and varying the robot team size from 5 to 25. In this figure, the comparison criterion is average cost per robot. These simulations show that the CA solutions always have a cost less than or equal to the solutions generated by RA, *if a solution can be found by CA*. (The comparison for application A_3 is similar to applications A_1 and A_2, and is omitted here for space reasons.)

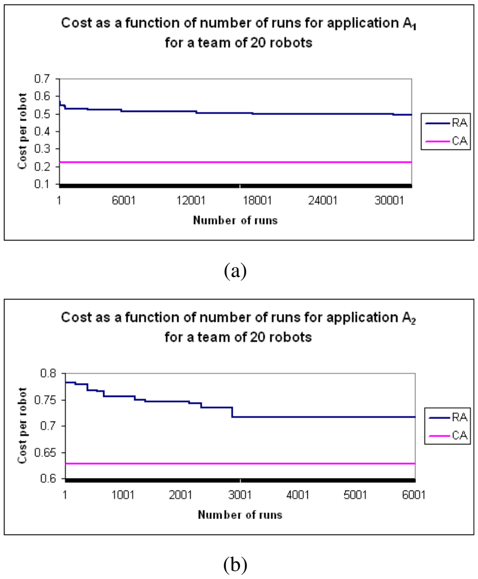
However, application A_4 poses a challenge for CA. Because of the nature of the limited resource requirements, a greedy search can only find the solution for specific sequences of robots. Because the heuristics of CA are designed to search all small solutions first, and because the number of possible solutions is exponential in the number of robots, CA is not able to find a solution for a team of 15 robots even after 50 hours of continuous running time (on a typical



(a)

(b)

Fig. 3. Comparison of solution costs per robot for robot teams with 5, 10, 15, 20, and 25 robots.



(a)

(b)

Fig. 4. Comparison of solution cost over time during a search for a team of 20 robots for applications A_1 and A_2.

present-day Linux PC). However, the RA approach is able to find a solution after 30 minutes.

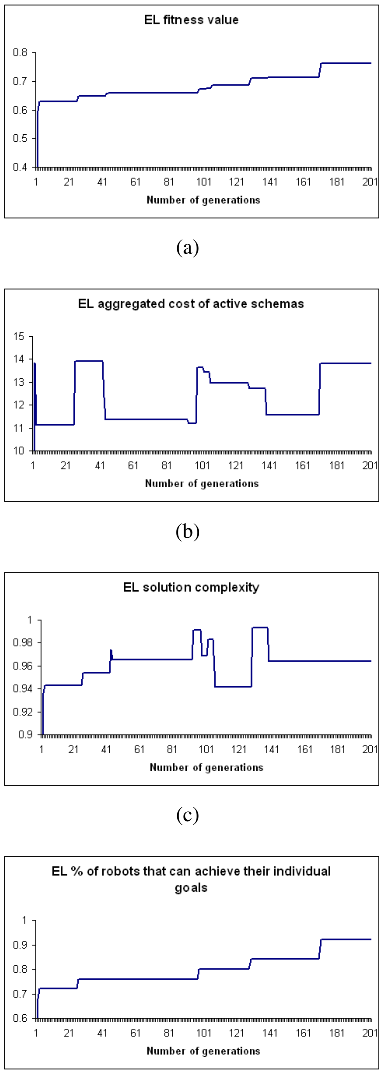
Figure 4 shows the change in cost values over time (during the search process) of CA and RA algorithms for a team of 20 robots for applications A_1 and A_2. In these simulation experiments, the first solution generated by CA using greedy search is always the best solution generated by CA. The later solutions do not improve. (This is not always the case, as shown in [8].) However, the first solution generated by RA is normally not the best solution, as the solution improves throughout the search process.

The EL algorithm is developed for offline constructivist learning. We examine its ability to improve its results over time. We also compare its result with CA via the cost of active schemas in the solution. EL generates both intermediate solutions (i.e., a partial solution in which only some of the robots can achieve the goal) as well as complete solutions. If EL generates good intermediate solutions in the beginning, then as the generations evolve and the fitness of the solution improves, more and more robots can accomplish the task, and eventually a complete solution can be found. However, sometimes, especially as the number of robots and schemas increases, EL does not guarantee a complete solution. (Presumably, a different set of EL parameter settings might overcome this problem, although in general, evolutionary techniques cannot guarantee convergence.) Figure 5 shows EL's simulation results for application A_1 with 25 robots ($\kappa = 0.3$, $\rho = 0.2$). This figure shows that the solution cost and complexity fluctuate during the search, while the percentage of the robots that can achieve a goal and the solution fitness increase. The cost and complexity do not decrease steadily because they change with the number of robots that can achieve their goal. The solution quality improvement can be credited to the evolving connections among *chunks* of schemas. Hence, EL provides a structural basis to further explore the search space, finding patterns of schema connections that cause the solution fitness to improve. We further compare the solution quality of CA and EL for application A_1 and A_2[6]. Figure 6 shows the cost for each robot in the solutions generated by CA and EL.

In summary, the simulations show that CA is usually very effective in finding solutions with low cost quickly. RA finds solutions in the same timely manner as CA, albeit with higher cost. However, CA cannot always find solutions in the time permitted, as discussed earlier for application A_4 that we developed to challenge the concept of CA. By contrast, RA was able to find solutions in this case. In RA, the cost of the discovered solution decreases throughout the search process.

Table III shows some key characteristics of the three search strategies. Compared with CA and RA, EL is more flexible because it can generate partial solutions with progressively improved quality by building upon previous discoveries. This ability of the EL algorithm to identify partial solutions and progressively improve the quality of the solutions without exhaustively searching the entire search space is important for helping us achieve our ultimate constructivist learning objectives. This ability is distinct from the CA and RA search processes, which do not have mechanisms for making use of these partial solutions, since they cannot compare the fitness of alternative partial solutions. Instead, the CA and RA algorithms discard these partial solutions (i.e., valid schema connections that lead to the provision of some, but not all, required information types) that have been

[6] Because CA's result is superior to RA for application A_1, we only compare EL with CA and omit the comparison between EL and RA.



(a)

(b)

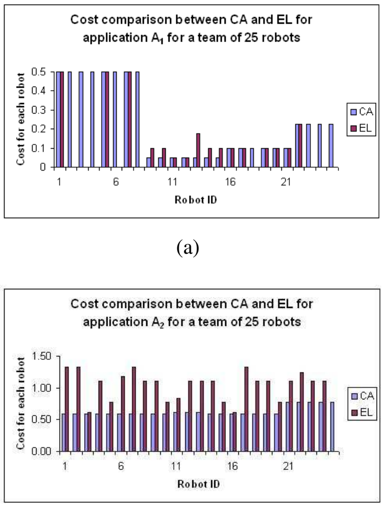
(c)

(d)

Fig. 5. These graphs show, during the search, the change over time of: (a) fitness value; (b) the total aggregated cost of all active schemas; (c) solution complexity; (d) the percentage of robots that can accomplish the task.

generated during the search process for each permutation arrangement of robots and starts the search process anew for the next robot sequence. Hence, the CA and RA approaches do not appear well-suited for our main research objective of achieving constructivist learning through schema chunking. In contrast, the simulation results show that the EL approach is competitive with the CA and RA approaches, and thus forms a solid foundation upon which to build constructivist learning techniques. Our goal is to enable online CA and RA search processes to use discovered schema chunks as if they were schemas, and in this way take advantage of these partial solutions generated by offline EL.

V. Related Work



(a)

(b)

Fig. 6. Comparison of solution cost for each robot in a team of 25 robots performing application A_1 and A_2. (a) EL generates a partial solution for the application A_1 that enables 16 robots to achieve their individual goals. From these 16 robots, 11 robots have solutions comparable to CA. (b) By increasing the connection rate between the schemas, EL generates a more complete team solution for the application A_2. However, most of the robots have higher costs than the CA solution. This result shows that EL is sensitive towards parameter settings. Another reason for the higher cost is that multiple solutions are generated for each robot. In the future, we are planning to separate those solutions into different chunks.

The schema-based building block approach, upon which all three of our search algorithms are based, is derived from the work of Arbib [1] and others. Arbib gives an overview of the theoretical aspects of schemas and explores schema theory from the neurological perspective. The definition of perceptual schema and motor schema originated from Arbib. Schemas are recursive in the sense that they can be divided into sub-schemas. Our research in constructivist learning is aimed at recursively build higher-level schemas based on existing schemas.

Having validated the evolutionary search process (EL) as appropriate for our ongoing constructivist learning work, we now briefly relate our schema-based foundation to other constructivist learning approaches. Drescher [6] and Chaput [3], [4] both developed schema-based constructivist learning models to emulate an infant exploring the environment using very basic perceptual schemas and motor schemas. Their work concentrated on the biological verification of the constructivist point of view using very basic level schemas that reflect the inherent abilities of an infant. Unlike their approach, our emphasis lies in automatically generating robot behaviors. We employ higher-level schemas, aiming for less computational complexity.

Our overall ideas for combining the offline search process for schema chunks together with the online use of these chunks to solve current tasks, in a constructivist learning model, originate from Piaget [9]. According to Piaget, the basic process of constructive intelligence development consists of two parts: accommodation and assimilation. Accommodation refers to modifying existing knowledge to adjust to the environment; assimilation refers to creating new knowledge by assembling existing knowledge. This paper

TABLE III

COMPARISON BETWEEN CENTRALIZED ASYMTRE SEARCH ALGORITHM (CA), RANDOMIZED ASYMTRE SEARCH ALGORITHM (RA), AND EVOLUTIONARY LEARNING SEARCH ALGORITHM (EL)

Algorithm	Computational Complexity	Solution Quality	Partial Solution	Learning
CA	Search for one robot sequence: $O(mn^2)$ Complete search for $n!$ permutation arrangements of robots: $O(mn!)$ m: number of potential solutions, n: number of robots	Locally optimal	No	No
RA	Search for one robot sequence: $O(mn^2)$ Complete search for $n!$ permutation arrangements of robots: $O(mn!)$ m: number of potential solutions, n: number of robots	Locally optimal	No	No
EL	Search for one generation: $O((mn)^2p)$ Search for g generations: $O((mn)^2pg)$ m: number of different schemas, n: number of robots p: number of individual solutions in the population	Locally optimal	Yes	Yes

proposed offline learning to accomplish assimilation. This offline learning process makes use of our EL search algorithm, validated in this paper, to generate chunks of schemas (SCSs). These chunks can then be used to generate new task solutions, with chunks treated in a similar manner as fundamental schemas. That is, chunks become more abstract schemas, which can then be treated the same as low-level schemas in the task solution generation process. The process iterates, with the objective of demonstrating constructivist learning over time.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have explored three different schema-based search strategies for generating robot coalitions to solve multi-robot team tasks — Centralized ASyMTRe (CA), Randomized ASyMTRe (RA), and Evolutionary Learning (EL). We compared these search strategies using four different simulated applications. The objective of these studies was to validate that our evolutionary learning technique can serve as the basis for constructivist learning without sacrificing solution quality.

RA is developed as alternative online solution search technique to CA. Our simulation results showed that none of CA and RA approach is always superior. For example, the heuristic CA approach can often find good solutions much more quickly than the other approaches. However, on one simulated application, CA was unable to find a solution in the allotted time whereas RA was able to find multiple solutions.

EL is developed for the purpose of offline constructivist learning. Our studies showed that the EL approach is comparable to the CA and RA approaches, and thus can serve as a valid foundation for continual robot learning. The main characteristic that EL has for this purpose is the ability to generate partial solutions, or chunks of schemas, that can be used to generate more complete solutions. This ability to generate partial solutions is especially important for constructivist learning, since we need a means to record and apply knowledge from previous searches to future problems. These partial solutions can be recorded as chunks of schemas, consisting of active schemas, connections among these schemas, and information types that the chunk, as a whole, requires and provides. A chunk of schemas can then be recorded in a schema repository and utilized for later search. For future studies of constructivist learning, we will apply the EL algorithm to develop a schema repository to hold learned chunks of connected schemas, and apply the chunks in subsequent search processes to solve new problems.

ACKNOWLEDGMENTS

This work is funded in part by NSF grant ISS-0631958. The authors thank Fang Tang and Balajee Kannan for their helpful discussions in the development of this research.

REFERENCES

[1] M. A. Arbib. Schema theory. In M. A. Arbib, editor, *The handbook of brain theory and neural networks*, pages 993–998. MIT Press, Cambridge, MA, USA, 2003.

[2] J. Bruner. *Acts of Meaning*. Harvard University Press, 1990.

[3] H. H. Chaput. *The constructivist learning architecture: A model of cognitive development for robust autonomous robots*. PhD thesis, The University of Texas at Austin, August 2004.

[4] H. H. Chaput, B. Kuipers, and R. Miikkulainen. Constructivist learning: A neural implementation of the schema mechanism. In *Proceedings of WSOM '03: Workshop on Self-Organizing Maps*, Kitakyushu, Japan, 2003.

[5] B. R. Donald. On information invariants in robotics. *Artificial Intelligence*, 72(1-2):217–304, 1995.

[6] G. Drescher. *Made-Up Minds: A Constructivist Approach to Artificial Intelligence*. MIT Press, 1991.

[7] B. P. Gerkey and M. J. Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *Intl. J. of Robotics Research*, 23(9):939–954, September 2004.

[8] L. E. Parker and F. Tang. Building multi-robot coalitions through automated task solution synthesis. *Proceedings of the IEEE, special issue on Multi-Robot Systems*, 94(7):1289–1305, July 2006.

[9] J. Piaget. *The Origins of Intelligence in Children*. London: Routledge and Kegan Paul, 1952.

[10] O. Shehory and S. Kraus. Feasible formation of coalitions among autonomous agents in non-super-additive environments. *Computational Intelligence*, 15(3), 1999.

[11] F. Tang and L. E. Parker. ASyMTRe: Automated synthesis of multi-robot task solutions through software reconfiguration. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, April 2005.

[12] F. Tang and L. E. Parker. Coalescing multi-robot teams through ASyMTRe: A formal analysis. In *Proceedings of IEEE International Conference on Advanced Robotics (ICAR)*, July 2005.

[13] F. Tang and L. E. Parker. Distributed multi-robot coalitions through ASyMTRe-D. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, August 2005.